What's New in Tableau

Browse summaries of new features for currently supported versions.

What's new in version 2018.2

Install and Deploy Tableau

- Install Tableau Bridge separately from Tableau Desktop on the next page

Connect to and Prepare Data

- Join spatial files on page 4
- Enhancements for Intuit QuickBooks Online connector on page 4

Design Views and Analyze Data

- Perform a nested sort with one click on page 4
- Extend functionality with dashboard extensions on page 5
- Visually align dashboard items with a grid on page 6
- Precisely position and size floating items with arrow keys on page 6
- See a preview while dragging floating dashboard items on page 7
- Float transparent filters, parameters, and highlighters over dashboards on page 7
- Automatically optimize dashboard layouts for mobile devices on page 8
- Show negative values on a log axis on page 8
- Use ISO 8601 date standard in calculated fields on page 9
Prepare, Author, and View Data on the Web

- Keyboard shortcuts for web authoring on page 9
- Create multi-connection data sources and cross-database joins on page 9
- Union your data on the web on page 10
- Pivot data on the web on page 10
- Copy data on the web on page 10
- Schema search on page 10
- Annotation improvements on page 11
- Edit Axis button on axes on page 12
- Better control over sorting on page 12

Collaborate with Tableau Online and Tableau Server

- Include colleagues on data conversations with @mentions on page 12

Save and Publish Data Sources and Workbooks

- Check how your workbook will look in an older version on page 13

Install and Deploy Tableau

Install Tableau Bridge separately from Tableau Desktop

Tableau Bridge is now a stand-alone client that you install separately from Tableau Desktop. For more information about Tableau Bridge requirements, compatibility with previous versions of Tableau Desktop, and the installation process, see Install Tableau Bridge in the Tableau Online Help.

Note: If you install Tableau Bridge on the same computer as Tableau Desktop, you can start Tableau Bridge from the Server menu like you did in previous versions.

Connect to and Prepare Data
Join spatial files

You can now join two spatial files, spatial database tables (Microsoft SQL Server only), or spatial fields across two data sources. For more information, see Join Spatial Files in Tableau on page 1922.

Enhancements for Intuit QuickBooks Online connector

There are several changes and enhancements to the Intuit QuickBooks Online connector that provide you with more flexibility. New transaction types are included in the Sales and Expenses Line Items table: Vendor Credit, Credit Memo, Journal Entry, Deposit, Estimate, Bill Payment, Payment, Purchase Order and Refund Receipt. Additionally, account information has been added to Credit Memo and Vendor Credit.

Individual transaction tables now have a new field called "Transaction Type" so you can join other tables to the Sales and Expenses Line Items table.

Design Views and Analyze Data

Perform a nested sort with one click

You can now use a nested sort operation without having to create calculated fields or combine dimensions. For more information, see Create a nested sort on page 1208
Extend functionality with dashboard extensions

Extensions increase dashboard functionality with custom objects created by third-party developers. With extensions, you can add unique features to dashboards or directly integrate them with applications outside Tableau.
Adding an extension is easy. In Tableau Desktop or the web-authoring mode of Tableau Online or Tableau Server, simply drag the Extension object to a dashboard. Then click either Extension Gallery to choose from all available extensions, or My Extensions to select an extension you previously downloaded. (To directly browse the Extension Gallery, go to extensiongallery.tableau.com.)

For more information, see Use Dashboard Extensions on page 2312.

Visually align dashboard items with a grid

To present a visually consistent design, arrange and size dashboard items over a grid. Choose Dashboard > Show Grid, or quickly toggle the grid on and off by pressing the G key.

For more information, see Size, position, and reorder individual dashboard items on page 2270.

Precisely position and size floating items with arrow keys

To precisely position a floating item, press arrow keys to move 1 pixel, or Shift+arrow keys to move 10 pixels. To resize items, add Alt (Windows) or Option (macOS) to the shortcuts above.
See a preview while dragging floating dashboard items

To give you a clear picture of how moved items will look in a new location, a preview appears while you drag them.

Float transparent filters, parameters, and highlighters over dashboards

To visually connect filters, parameters, and highlighters to related data, float these items, which are now transparent by default. Text always remains fully opaque, maintaining legibility.

For more information, see Tile or float dashboard items on page 2266.
Automatically optimize dashboard layouts for mobile devices

When you create a device layout, Tableau Desktop now automatically optimizes for the device type, adjusting the arrangement of dashboard items to best fit a phone or tablet. Often, these automatic layouts will address all the needs of your mobile users, but you can always customize a layout further.

For more information, see Create Dashboard Layouts for Different Device Types on page 2302.

Show negative values on a log axis

In Tableau Desktop and web authoring, when you select Logarithmic scale for an axis, you now have the option to specify Symmetric to display data that contains 0 or negative values on a log scale axis. For related details, see Change the axis scale to reversed or logarithmic on page 1846.
Use ISO 8601 date standard in calculated fields

When creating a date calculation, you now have the option to use the ISO 8601 international standard to resolve the calculation. For more information, see Date Functions on page 1298.

Prepare, Author, and View Data on the Web

Keyboard shortcuts for web authoring

Use keyboard shortcuts to quickly create and edit views on the web. For more information, see Shortcuts for web authoring.

Create multi-connection data sources and cross-database joins

Add a second connection to your data source and combine tables from different databases using a cross-database join. Your multi-connection data source can be saved for others in your
organization to use. To learn more about cross-database joins, see Join Your Data on page 657.

Union your data on the web

Can create new unions and add to unions from the same connection in Tableau Online and Tableau Server. To learn more, see Union Your Data on page 708.

Pivot data on the web

When working with file-based data sources on the web, you can pivot your data from crosstab format into columnar format. For more information, see Pivot Data from Columns to Rows on page 747.

Copy data on the web

Copy values in your data grid by selecting the values and then pressing Ctrl+C (Windows) or Command+C (macOS). Alternatively, to copy values in the metadata grid, select the values, right-click, and select Copy.

Schema search

Quickly locate data fields in the Data pane using schema search.
Annotation improvements

Create and delete annotations by right-clicking the view (in a worksheet or dashboard), and then selecting point or mark annotation. Right-click the annotation again to edit or remove the annotation.
Edit Axis button on axes

You now have two ways to open the Edit Axis dialog box. Hover near the top of the axis (vertical axis) or to the right (horizontal axis) and click the drop-down arrow that appears. Or, double-click the axis.

Better control over sorting

You can now access the Sort dialog box by right-clicking a dimension field on the Rows or Columns shelves in a view.

Collaborate with Tableau Online and Tableau Server

Include colleagues on data conversations with @mentions

To engage people in a conversation about compelling data you find in a view, @mention them in comments. As you begin to type a name, matching people with Tableau accounts at your organization appear in a list you can choose from. For more information, see Comment on Views on page 2673.
Quickly create data alerts on customized views

If you’ve changed a view (for example, by adjusting filters), you can now create alerts without first saving a custom view. For more information, see Send Data-Driven Alerts from Tableau Online or Tableau Server on page 2669.

Save and Publish Data Sources and Workbooks

Check how your workbook will look in an older version

It's important to know how your workbook will be affected before you export it to an older version of Tableau. The File > Export As command now gives you a clear picture, telling you if your workbook will change and which features are unsupported.

Added in version 2018.1

Tableau Licensing

- New user-based term licenses are now available on page 15

Install and Deploy Tableau
- Register Tableau Desktop from the command line on the next page
- Mac OS system requirements on page 16

Connect to and Prepare Data
- Support for remote config files for SAP NetWeaver Business Warehouse on page 16
- Connect to spatial columns in Microsoft SQL Server on page 16
- New connector name on page 16

Design Views and Analyze Data
- Apply step lines and jump lines on page 16
- Tips for analytics objects on page 17
- Filter hierarchical data on page 18
- Hide a Viz in Tooltip worksheet on page 19
- Refit saved clusters on page 20
- Synchronize dual axes for different numeric data types on page 20
- Navigate stories with arrows alone on page 20
- Create dual-axis (layered) maps from generated and custom latitude and longitude fields on page 20

Author and View Data on the Web
- Connect to data on the web on page 21
- Prepare your data on the web on page 21
- Resize headers on the web on page 21
- Add an accessible toolbar to an embedded view on page 22
- Other web authoring enhancements on page 22

Collaborate with Tableau Online and Tableau Server
- Quickly analyze cloud-based data with Dashboard Starters on page 23
Streamline discussion threads by deleting comments on page 23

Publish Data Sources and Workbooks

- Downgrade workbooks in Tableau Online on page 23

Tableau Licensing

New user-based term licenses are now available

Tableau now offers different types of user-based term licenses that grant a range of web authoring and other capabilities at various price points. This gives organizations more flexibility to fit licenses to the data analysis and visualization needs of different users. To learn more, see User-based licenses, or to see the capabilities of each license, see Tableau Pricing.

Install and Deploy Tableau

Register Tableau Desktop from the command line

When you install Tableau Desktop from the command line (for Windows) you can now register the product as part of the install without having to complete the registration for new users through the user interface or as a separate command line operation.

Add a single command line that includes the REGISTER="1" installer option. During the installation process, the installer will run the -register process and add the registration information. This process uses the registration information from the following location in the registry: HKEY_CURRENT_USER\Software\Tableau\Registration\Data.

For information about adding registration information to the registry and the property settings for the -register property, see the "Register Tableau Desktop" and the "Installer properties" sections in Deploy Tableau Desktop.
Mac OS system requirements

Tableau Desktop version 2018.1 for the Mac requires Mac OS X 10.11 El Capitan or later. All Tableau Desktop versions 10.0 through 10.5 are compatible with Mac OS X 10.10 Yosemite.

Connect to and Prepare Data

Support for remote config files for SAP NetWeaver Business Warehouse

Tableau now supports using landscape XML files located on a remote HTTP server.

Connect to spatial columns in Microsoft SQL Server

You can now use spatial columns when you connect to Microsoft SQL Server in Tableau. You can also use Custom SQL and RAWSQL to perform advanced spatial analysis in Tableau. For more information, see Connect to Spatial Data in Microsoft SQL Server on page 1908.

New connector name

Starting in Tableau 2018.1.4, the Cisco Information Server connector is renamed to TIBCO Data Virtualization.

Design Views and Analyze Data

Apply step lines and jump lines

Change the Line mark type to a step line or a jump line by clicking the Path property in the Marks card.
Step lines and jump lines can help you see and compare incremental patterns of change over time. Use these line types for numeric data that remains constant for periods of time, with noticeable changes or deltas – such as account balances, inventory levels, or interest rates. For details, see Draw paths between marks on page 1143.

Tips for analytics objects

In the Analytics pane, available analytics objects vary based on the current state of the data in a visualization. For example, you can't apply trend lines to the view when it contains stacked marks. Tooltips in the Analytics pane explain why an Analytics object isn't available, or provide a tip on what is required to use that object in the view.
Filter hierarchical data

When the data in your view contains hierarchies, filters for those fields can use the **All Values in Hierarchy** option. Your filtering experience now matches the parent/child relationships in the hierarchy.

For details on filtering options, see [Set options for filter card interaction and appearance](#) on page 1178.
Hide a Viz in Tooltip worksheet

You can now hide worksheets used as Viz in Tooltips, the same way you would hide worksheets used in stories or dashboards. In the target worksheet that is the Viz in Tooltip, click **Hide**.

To show the Viz in Tooltip worksheet again, in the source worksheet, click **Unhide All Sheets**.

For more details about Viz in Tooltip, see **Create Views in Tooltips (Viz in Tooltip)** on page 1773.
Refit saved clusters

Clusters are saved as groups so you can use them in other worksheets and workbooks, however, they don't automatically refresh. When the underlying data changes, you can now use the Refit option to refresh and recompute the data for your saved clusters. For related details, see Refit saved clusters on page 1687 in Find Clusters in Data on page 1680.

Synchronize dual axes for different numeric data types

In a chart with dual axes, you can now synchronize dual axes for numeric data types that don't match. For example, you can synchronize an axis that uses an integer data type and an axis that uses decimal data type.

Navigate stories with arrows alone

If you like a clean, simple navigator style, select the new arrows-only option. For more information, see Create a Story on page 2325.

Create dual-axis (layered) maps from generated and custom latitude and longitude fields

You can now create dual-axis maps from a combination of Tableau generated latitude and longitude fields, and custom latitude and longitude fields. For more information, see Create a dual-axis map from a combination of generated and custom latitude and longitude fields on page 2030.
Author and View Data on the Web

Connect to data on the web

With a Creator user license, you can connect to data directly in your browser. Upload Excel or text-based data sources, connect to data housed in a cloud database or on a server in your enterprise, or connect to published data saved to your site. For more information, see Creators: Connect to data on the web.

Prepare your data on the web

With a Creator user license, use the Data Sources tab to prepare your data for analysis. Create joins, add new data sources, clean your data using the Data Interpreter, and more. For more information, see Creators: Prepare Data on the Web.

Resize headers on the web

Resize the width of row headers and the height of column headers.
Add an accessible toolbar to an embedded view

Authors can enable an accessible toolbar so that all users can add comments, download the view, or share it with others. For more information, see Publish and embed the view in Author Views for Accessibility.

Other web authoring enhancements

- Show Me is now available for dashboards.
- Double-click an annotation to edit its text.
- Hide or show a Viz in Tooltip worksheet. In the target worksheet that is the Viz in Tooltip, click Hide. To show the Viz in Tooltip worksheet again, in the source worksheet, click Unhide All Sheets.
- Filter hierarchical data in views by using the All Values in Hierarchy filter option.
- Change the Line mark type to a step line or a jump line by clicking the Path property in the Marks card.
- Double-click an axis to edit axis settings.

Collaborate with Tableau Online and Tableau Server
Quickly analyze cloud-based data with Dashboard Starters

If you use Tableau Online, Dashboard Starters help you quickly author and analyze data from cloud-based systems like Salesforce, ServiceNow, Marketo, and Eloqua. Simply create a new workbook and choose from several beautiful, informative designs that are tailor-made for key business metrics. For more information, see Dashboard Starters for Cloud-based Data on page 2252.

Streamline discussion threads by deleting comments

If a comment is unnecessary or inaccurate, you can quickly delete it. Just click the X in the upper-right corner. For more information, see Comment on Views on page 2673.

Publish Data Sources and Workbooks

Downgrade workbooks in Tableau Online

If you create or edit a workbook in Tableau Online or Tableau Server, you can now downgrade it to a previous version so that users who are using Tableau Desktop 10.2 and later can still
open the workbook. To downgrade the workbook in Server, open a view and select Download > Tableau Workbook and select the version that you want to downgrade to. The file will download to your computer.

The downgraded workbook can be opened in the version selected and later. Any features not available in that previous version will be removed when the workbook is downgraded.

Added in version 10.5

Install and Deploy Tableau

- Activate Tableau Desktop from the command line on the next page
- Turn off driver downloads from the command line (Windows) on page 26
- Desktop operating system requirements (64-bit replaces 32-bit) on page 26

Connect to and Prepare Data

- Extracts use the new .hyper format on page 26
- Changes to the way values are computed on page 27
- Google authentication method update on page 27
- New Box connector on page 27
- Support for SSL authentication to IBM DB2 and BigInsights databases on page 27
- Use the Extract API on page 27
- SAP HANA support for failover on page 28
- SAP Sybase ASE support for SSO on page 28
- Support for OData V4 on page 28
- Oracle Eloqua connector update on page 28

Design Views and Analyze Data
Install and Deploy Tableau

Activate Tableau Desktop from the command line

When you install Tableau Desktop from the command line you can now activate the product without having to complete the activation process through the user interface. Instead add a single command line that includes the installer option ACTIVATE_KEY "<key>". During the installation process, the installer will run activate and apply the license key you provide to activate the product automatically.

For more information about installing Tableau from the command line, see Deploy Tableau Desktop.
Turn off driver downloads from the command line (Windows)

When you install Tableau Desktop the most common drivers are automatically installed for you. To prevent all or some of these drivers from installing automatically, you can turn off that option during install.

- To prevent all drivers from being automatically installed, do one of the following:
  - During install, from the Installer screen, click Customize. Then clear the Install database drivers: Microsoft SQL Server, PostreSQL and Amazon Redshift check box.
  - From the command line, set the DATABASEDRIVERS = "0"

- To prevent one or more of these drivers from being automatically installed, do the following:
  - Leave the Install database drivers: Microsoft SQL Server, PostreSQL and Amazon Redshift check box selected, or from the command line, set the DATABASEDRIVERS = "1"
  - Add a command line and set DRIVER_MSSQL, DRIVER_POSTGRES or DRIVER_REDSHIFT to "0" to disable an individual driver.

For more information about installing Tableau from the command line, see Deploy Tableau Desktop.

Desktop operating system requirements (64-bit replaces 32-bit)

Starting with version 10.5, Tableau Desktop, Tableau Reader, and Tableau Public (desktop) run only on 64-bit operating systems. Version 10.4 is the last version of Tableau Desktop, Tableau Reader, and Tableau Public to support 32-bit Windows operating systems.

Connect to and Prepare Data

Extracts use the new .hyper format

Beginning with version 10.5, when you create an extract, it uses the new .hyper format. Extracts in this new format take advantage of the improved data engine, which supports the same analytical and query performance as the data engine before it, but for even larger extracts.
Similarly, when you perform an extract task on a .tde extract using version 10.5, the extract is upgraded to a .hyper extract. For more information, see Extract Upgrade to .hyper Format on page 788.

Changes to the way values are computed

To improve data source efficiency and scalability, and produce results that are consistent with commercial databases, values in your data source can be computed differently in version 10.5. In some cases, these changes can cause differences with the values and marks in your view between version 10.4 (and earlier) and version 10.5 (and later). This applies to extract data sources, and can apply to multi-connection data sources, data sources that use live connections to file-based data, data sources that connect to Google Sheets data, cloud-based data sources, extract-only data sources, and WDC data sources. For more information, see Changes to values and marks in the view on page 775.

Google authentication method update

Starting in October 2017, Google's updated security requirements changed the connection workflow for some Tableau connectors. Now Tableau opens at tab in your external default browser when connecting to data that uses OAuth authentication. The connectors affected are Google Sheets on page 441, Google BigQuery on page 431, Google Analytics on page 426, Dropbox on page 418, OneDrive on page 503, and Box on page 407.

New Box connector

Use the Box connector to connect to Box data. For more information, see Box on page 407.

Support for SSL authentication to IBM DB2 and BigInsights databases

You can now connect to IBM DB2 and IBM BigInsights databases using SSL authentication. For more information, see IBM DB2 on page 453 and IBM BigInsights on page 451.

Use the Extract API

You can use the Extract API 2.0 to create .hyper extracts. For tasks that you previously performed using the Tableau SDK, such as publishing extracts, you can use the Tableau

- 27 -
Server REST API or the Tableau Server Client (Python) library. For refresh tasks, you can also use the Tableau Server REST API. For more information, see Tableau Extract API on page 817.

SAP HANA support for failover
Tableau supports connecting to a server that's configured for failover. For more information, see SAP HANA on page 541.

SAP Sybase ASE support for SSO
You can now connect to SAP Sybase ASE data using single sign-on (SSO). For more information, see SAP Sybase ASE on page 560.

Support for OData V4
Tableau now supports OData version 4.0. For more information, see OData on page 500.

Oracle Eloqua connector update
The Oracle Eloqua connector now supports incremental refreshes, validation for tables with more than 250 fields, and support for additional tables. For more information, see Oracle Eloqua on page 510.

Design Views and Analyze Data

Show visualizations in tooltips
As you craft views and look for ways to reveal more details about data to your audience, you can embed visualizations within tooltips—aka "Viz in Tooltip."
When you show related views in tooltips, you can help your audience engage with the data at a deeper level, while maximizing the space available for the current view. For details, see Create Views in Tooltips (Viz in Tooltip) on page 1773.

**Power trend lines**

When you add trend lines to a view, a Power option is now available. For related details, see Add Trend Lines to a Visualization on page 1661 and the “Power” model type in Trend Line Model Types on page 1669.
Map data updates and additions

In Tableau 10.5, additional cities are available for South East Asia, Central America, and South America.

Demographic data updates including 2018 to 2023 statistical projections are now available.

U.S. Core Based Statistical Areas (CBSA) and Metropolitan Statistical Areas (MSA) have been updated to 2015 definitions.

Updates are now available for U.S. postcodes.

Additionally, the following updates are now available for China:

- The areas of Beijing, Chongqing, Shanghai and Tianjin no longer contain city-level districts in the Administrative level 2/County geographic role.
- Administrative level 2 features in China (prefecture-level cities, prefectures, leagues, autonomous prefectures) are considered cities in Tableau. The center points of these areas are included in the City geographic role as well.
- Additional cities such as Tiemenguan are now included in the City geographic role.
- Names for Administrative level 1 and 2 features are now consistent with names published by the People’s Republic of China’s National Bureau of Statistics.

Extend dashboard dimensions to 10,000 pixels

To optimize designs for very large screens or vertical mobile layouts, you can now extend dashboard width and height to 10,000 pixels.

Publish Data Sources and Workbooks

Downgrade workbooks in Tableau Desktop

If you upgrade a workbook in Tableau Desktop to a newer version, you can now downgrade it to a previous version so that users who are using Tableau Desktop 10.2 and later can still open the workbook. To downgrade the workbook, in Tableau Desktop select **File > Export As** and select the version that you want to downgrade to. Then save it to your My Tableau Repository or a selected location.
The downgraded workbook can be opened in the version selected and later. Any features not available in that previous version will be removed when the workbook is downgraded.

**Note**: This feature is not available for workbooks that include a Tableau Data Extract that has been refreshed in 10.5. The extract will have been converted to use the new .hyper format and this format can’t be downgraded back to a .tde format.

## New in Web Authoring

### Edit axes

You can now edit axes on the web. To open the Edit Axis dialog box on the web, double-click an axis in the view. Options available from the Edit Axis dialog box include **Synchronize dual axes**, clearing the axis range (**Reset**), and editing tick marks. You can also enable or disable **Dual axis** in a field context menu (right-click a measure field on Rows or Columns shelf).

### Additional enhancements

- Add text objects to dashboards and edit them.
- Edit worksheet and dashboard titles. Double-click the title to open the **Edit Title** dialog box.
- Edit trend lines and view a description of the trend model. To view the trend model, hover
over any part of the trend line. To edit a trend line on the web, click a trend line and hold the cursor in place, and then click the **Edit** menu.

- Trend lines now include a Power option.
- Create hierarchies by dragging one dimension onto another in the Data pane.
- Set the default color property for a field.
- For quantitative color legends, advanced settings are available (dates not included).

**Collaborate with Tableau Online and Tableau Server**

**Upload custom images for user profiles**

By default, users and groups now have profile images that reflect their first two initials. To help you visually distinguish between profile types, user images are circular, while group images are square.

User profiles support custom images. To upload one, go to your [account settings](#) page, and click the current image or initials next to your name.

If your organization uses Active Directory, default profile images come from that system. But two-letter images may briefly appear until Active Directory next synchronizes with your Tableau server.

**Subscribe users in groups, and add custom messages**

To quickly subscribe several users, you can select entire groups created by your Tableau administrator. Each user in a group is subscribed independently, so the list of recipients remains fully customizable.
You can also add custom messages to subscription emails, clarifying their contents.

For more information, see **Subscribe to Views** on page 2665.

**Rename workbooks on Tableau Online or Tableau Server**

There's no need to republish workbooks just to rename them. Now you can rename workbooks directly on Tableau Online or Tableau Server. When you do, workbook URLs remain the same, so any links you’ve shared keep working.

Simply navigate to the main **Workbooks** area, click the pop-up menu in the upper-right corner of a workbook, and select **Rename**.
To rename a workbook, users need the Save permission. You'll have that by default for workbooks you own, but you might need to add it for other users. For details, see View or Edit Permission Rules in Help for Tableau site administrators.

Added in version 10.4

Install and Deploy Tableau

- Remove previous versions of Tableau Desktop during install (Windows only) on the next page
- Downgrade and publish workbooks to an older version of Tableau Server on page 36

Connect to and Prepare Data

- Certified and recommended data sources on page 37
- New Denodo connector on page 39
- Other Databases (ODBC) connector on the Mac on page 39
- Oracle connector supports SSL on page 40
- SAP GUI 7.4 support on page 40
- Create an SAP BW extract without a special license on page 40

Design Views and Analyze Data

- Precise dashboard spacing, with border and background color controls on page 40
- High DPI support for Windows on page 42
- Support for linear geometries on page 43
- Map data updates and additions on page 43
- MATLAB integration on page 44
- WCAG-conformant drop-down filters in embedded views on page 44

New in Web Authoring

- Turn on the View Toolbar on the web on page 45
- Edit groups on the web on page 46
- Create aliases on the web on page 46
- Format lines on page 46
- Rich text editing for dashboards and stories on page 47

Share and Collaborate

- Enhanced commenting on Tableau Server and Tableau Online on page 47
- Learn who has seen a view on Tableau Server and Tableau Online on page 48

Install and Deploy Tableau

Remove previous versions of Tableau Desktop during install (Windows only)

When you install the latest version of Tableau Desktop or Tableau Reader from the command line you can now automatically remove previous installed versions of those applications from the machine. Starting with version 10.4, you can set the REMOVEINSTALLEDAPP installer property setting to "1" and during install, the installer will detect and remove any previously installed versions back to version 9.3. If the machine has Tableau Desktop or Tableau Reader version 9.2 or prior installed, those versions won't be automatically uninstalled. Instead you must manually uninstall them.

Note: Tableau Public doesn't use this installer property as it always removes the previous version when you install a new version.
For more information about how to set the installer properties from the command line, see Deploy Tableau Desktop in the Tableau Desktop Deployment guide.

Downgrade and publish workbooks to an older version of Tableau Server

In previous releases, publishing workbooks to Tableau Server could be problematic when Tableau Desktop and Tableau Server were on different major versions of the application. For example, if you tried to publish a workbook authored in Tableau Desktop 10.3 to Tableau Server running on version 10.2 you received an error like this one.

![Error Message](image)

Now in Tableau Desktop 10.4 (Windows and Mac) when you publish a workbook to an older version of Tableau Server (10.2 and later), you receive a warning message instead that tells you that the workbook will be downgraded to the version that you are publishing to. Any features or functionality included in the workbook that aren't available in the older version is removed for compatibility.

**Note:** The new downgrade option is not available when publishing workbooks to Tableau Server 10.1 and earlier.
You can still open the downgraded workbook in the current version of Tableau Desktop, but you may need to add back the features that were removed when the workbook was downgraded. For more information about compatibility, see Version Compatibility Between Tableau Desktop and Tableau Server in the Tableau Desktop Deployment guide.

Connect to and Prepare Data

Certified and recommended data sources

Have you ever wondered if you could trust your underlying data or wished you could avoid creating a totally new data source? Author Tableau views with confidence and speed using certified and recommended data sources previously published by others.

Certified data sources are supported by both Tableau Online and Tableau Server. These data sources are carefully chosen by site administrators and project leaders and appear with a
unique certification badge. Hover over the badge to learn who certified the data source, read any descriptive notes they’ve provided, and confirm the connection type.

Here’s how certified data sources look when you connect to data in Tableau Desktop.

And here’s how you can identify certified data sources in Tableau Online or Tableau Server.

Recommended data sources are supported by Tableau Server. These data sources include personally certified ones and others automatically picked for you based on usage patterns at your organization. You can access recommended data sources on the right side of the data source setup page in Tableau Desktop, while you’re signed in to Tableau Server.
Recommended data sources are supported for a subset of connectors. For more information, see Use Certified and Recommended Data Sources and Tables on page 737.

New Denodo connector

Use the Denodo connector to connect to Denodo data. For more information, see Denodo on page 415.

Other Databases (ODBC) connector on the Mac

You can now use the Other Databases (ODBC) connector on the Mac to connect to a database when the database driver you want to connect with implements the ODBC standard. For more information, see Other Databases (ODBC) on page 600.
Oracle connector supports SSL
You can now connect to Oracle using SSL on premises, after you configure the Oracle client. For more information see Oracle on page 506.

SAP GUI 7.4 support
Tableau supports the new SAP UI Landscape XML format. For more information, see Support for SAPUILandscape.xml on page 552.

Create an SAP BW extract without a special license
Starting in 10.4, you can create an SAP BW extract without the need for a special license key. Prior to 10.4, SAP BW extracts were an unsupported beta feature. For more information, see Support for SAP BW extracts on page 555.

Design Views and Analyze Data

Precise dashboard spacing, with border and background color controls
If you previously used blank objects to refine dashboard layouts, you'll really appreciate the new spacing controls. Padding lets you precisely position items on dashboard, while borders and background colors let you visually highlight them.
A. Outer padding B. Blue border C. Inner padding with light blue background

To help you quickly produce attractive layouts, default outer padding is optimized for different types of dashboard items. Layout containers have no outer padding by default, sheets and objects have four pixels, and entire dashboards have eight pixels for trouble-free image and PDF export.

In the past, adjoining dashboard items were separated by a minimum of four pixels, but now you can create seamless designs by specifying zero outer padding.
Dashboard spacing is available in both Tableau Desktop and the web authoring workspace of Tableau Server and Tableau Online. For more details, see Add padding, borders, and background colors around items on page 227.

High DPI support for Windows

If DPI scaling is enabled on your Windows machines, you can now author your views with confidence, knowing that they will look as beautiful at 100% scaling as they do at 200% scaling.
Support for linear geometries

Starting with Tableau 10.4, you can connect to spatial files that contain linear geometries.

To create a map with linear geometries:

1. Connect to your spatial data.
2. Navigate to a worksheet.
3. In the Data pane, double-click the Geometry field.

You can encode size and color, as well as add additional levels of detail to the view.

For more information, see Create Tableau Maps from Spatial Files on page 1959.

Map data updates and additions

In Tableau 10.4, over 32,000 new U.S. cities are available. Additionally, the following NUTS codes are now supported:

- DE401
- DEB3I
- DEG0I
- DK012
- DK021
MATLAB integration

You can access MATLAB® models (deployed on MATLAB production servers) in Tableau using SCRIPT_ functions, or pre-process your data using MATLAB and persist that data into a Tableau data extract for further analysis.

For more information, see Put your MATLAB models and algorithms to work in Tableau and Pass Expressions to External Services on page 1623.

WCAG-conformant drop-down filters in embedded views

Single Value (drop-down) and Multiple Values (drop-down) filters are now WCAG-conformant when accessed in embedded views.
New in Web Authoring

Turn on the View Toolbar on the web

In prior releases, when authoring on the web, you could only see and use the View Toolbar when working with map views. Now you can turn on the View Toolbar for any view or dashboard in your workbook in web authoring.

From the top menu, select Worksheet > Show View Toolbar and select an option.

From a dashboard, select the zone on the dashboard where you want to show the View Toolbar, then do one of the following:

- From the top menu select Worksheet > Show View Toolbar.
- Select the drop-down arrow, then select the appropriate option from the context menu.
Edit groups on the web

Starting with Tableau version 10.4, you can edit groups.

To edit a group, in the Data pane, right-click a group field and select **Edit Group**.

In the Edit Group dialog box that opens, you can add or remove members from an existing group, as well as create new groups in the group field. You can also choose to Include an Other group.

Create aliases on the web

Starting with Tableau version 10.4, you can create aliases for members in a dimension so that their labels appear differently in the view.

To create aliases for a field:

1. In the Data pane, right-click a dimension and select **Aliases**.
2. In the Edit Aliases dialog box, under **Value (Alias)**, select a member and enter a new name.

Format lines

When you are authoring on the web, you can quickly change all the lines in your workbook. Select **Format > Workbook**, then expand the **Lines** section on the **Format Workbook** and make your selections. For more information, see **Format at the Workbook Level** on page 2356.
Rich text editing for dashboards and stories

You can now edit dashboard titles and zones, as well as story titles and descriptions on the web.

To do so, double-click them and then update the text in the dialog box that opens.

For more information, see Format Titles, Captions, Tooltips, and Legends on page 2382.

Share and Collaborate

Enhanced commenting on Tableau Server and Tableau Online

Commenting has been completely redesigned to inspire conversations about data discoveries. The new Comments pane appears to the right of views, rather than at the bottom, so you can see the discussion and the data at the same time. Profile images automatically appear too,
helping you quickly identify other users. To share filtered views with them, add interactive snapshots along with your comments to highlight the data you're describing.

Any comments you add in a desktop browser also appear in Tableau Mobile, and vice versa, so you can easily communicate with colleagues on the go. Comments remain with a view even if you revise the containing workbook (as long as the view name remains the same).

For more information, see Comment on Views on page 2673.

Learn who has seen a view on Tableau Server and Tableau Online

If you own a view, you can quickly find out who at your organization has seen it. At the top of a main site page, click Views. Then hover over the specific view you're interested in, and select
Who Has Seen This View? from the menu at upper-right.

<table>
<thead>
<tr>
<th>Views</th>
<th>Data Sources</th>
<th>Subscriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Top Viewed Threads**

230 views  0

Added in version 10.3

**Help Redesign**

- Combined help system on the next page

**Connect to and Prepare Data**

- Connect to .pdf files on page 51
- Union even more data on page 51
- Recommended tables and joins on page 51
- New connectors on page 52
- JSON schema updates on page 53

**Design Views and Analyze Data**
- Double-click to zoom on page 53
- Tooltip selection on page 53
- Latest date presets for discrete date filters on page 54
- Apply table calculation filters to totals on page 56
- Map data updates and additions on page 57
- Evenly distribute a layout container's items on page 57
- Easy access to story actions on page 58
- Use numbers for story navigation on page 58

New in Web Authoring

- Edit quantitative color legends on page 58
- Edit colors on separated legends on page 59
- Customize how people interact with your map on page 60
- Format numbers on the web on page 60
- Create stories on page 61
- Create and edit bins on page 62
- Drag fields to Show Me in the view on page 62
- Drill up and down a continuous hierarchy in the view on page 63
- Save data source on page 64

Share and Collaborate

Data-driven alerts from Tableau Online and Tableau Server on page 65

Help Redesign

Combined help system

Starting with version 10.3, Tableau Desktop Help is now known as Tableau Help. Tableau Help contains all help topics related to analyzing and consuming data in Tableau Desktop, Tableau
Server, and Tableau Online. This help is for people who create workbooks or data sources and publish them, and for people who want to see, interact with, and share views in Tableau.

We welcome your feedback on this change, as well as any feedback or ideas you might have on how to make our content better. Please use the feedback bar on the top of any page ("Was this page helpful?") to open the comment field and submit your feedback.

Use Tableau on the Web on page 2554 provides links to the topics that used to be located in Tableau Server and Tableau Online help.

Connect to and Prepare Data

Connect to .pdf files

Now you can use the PDF File connector to connect to tables in your .pdf files. For more information, see PDF File on page 344.

Union even more data

You can union tables from a .pdf file as well as Aster Database, Cloudera Hadoop, Hortonworks Hadoop, IBM DB2, IBM PDA (Netezza), Pivotal Greenplum Database, SAP Sybase ASE, SAP Sybase IQ, and Teradata. For more information about union, see Union Your Data on page 708.

Recommended tables and joins

When you’re setting up the Tableau data source or modifying an existing one, it can be helpful to see what tables are popular with other people at your organization. When you connect to a database while simultaneously signed in to Tableau Server (from Tableau Desktop), in the left pane a Recommended list shows the tables used most often in data sources and workbooks published to Tableau Server.
After you add a recommended table to the canvas, others tables that are frequently joined to it appear in the Recommended list. Double-click one to automatically create a recommended join between the tables, using the most popular fields and join type by default. Note: If foreign keys are specified in the database, they override the recommended joins.

Recommended tables and joins are supported for a subset of connectors. For more information, see Use Certified and Recommended Data Sources and Tables on page 737.

**New connectors**

**New Amazon Athena connector**

Use the Amazon Athena connector to connect to Amazon Athena data. For more information, see Amazon Athena on page 382.

**New Dropbox connector**

Use the Dropbox connector to connect data on Dropbox. For more information, see Dropbox on page 418.

**New MongoDB BI connector**

Use the MongoDB BI connector to connect to MongoDB BI data. For more information, see MongoDB BI Connector on page 493.
New OneDrive connector
Use the OneDrive connector to connect to data on OneDrive. For more information, see OneDrive on page 503.

New ServiceNow ITSM connector
Use the ServiceNow ITSM connector to connect to ServiceNow ITSM data. For more information, see ServiceNow ITSM on page 565.

JSON schema updates
Tableau scans the data in the first 10,000 rows of the JSON file and infers the schema from that process. Sometimes, more fields exist in rows that were not scanned to create the inferred schema. Now, Tableau indicates when additional fields are detected that you may want to include in your analysis. For more information, see JSON File on page 338.

Design Views and Analyze Data

Double-click to zoom
In prior releases, when you double-clicked on a view that used continuous measures for both the X and Y axis (also known as quantitative/quantitative views, for example scatter plots) and the View toolbar was turned off, Tableau would automatically zoom in.

This resulted in a confusing user experience. Now for these quantitative/quantitative non-map view types, Tableau no longer zooms in by default when you double-click on the view. This change applies to new workbooks and upgraded workbooks. The double-click to zoom functionality is also turned off by default for table and cross-tab chart types.

Tooltip selection
This feature enables you to select similar marks in a view from the tooltip by category. Any discrete dimension or measure that is included in the tooltip becomes an active link that you can click on to select other marks in the view that have the same value.

The example below shows how clicking on Machines in the tooltip selects all marks in the view that have the same sub-category, in this case, Machines.
This feature works on Tableau Desktop, Tableau Server and Tableau Mobile. It is turned on by default for new worksheets and turned off by default for upgraded worksheets. To turn the feature off or on, at the worksheet level click **Tooltip** on the marks card to open the **Edit Tooltip** dialog box, and then clear or select the **Allow selection by category** check box.

For more information, see **Highlight data by category in tooltips** on page 2597.

**Latest date presets for discrete date filters**

Discrete date filters can be set to filter to the latest date value in the data source. You can use this setting to ensure that the most recent date in the data source is selected in the filter. For
details, see Latest Date Preset for "Filter Discrete Dates" under "Filter Dates" (Desktop) in Filter Data from Your Views on page 1162.
On Tableau Server and Tableau Online, presets are applied when the view first loads in the browser, but not when the browser or data is refreshed.

Apply table calculation filters to totals

When you show totals and you want a table calculation filter to apply to the totals, you can now select **Apply to totals** in the drop-down menu for that filter (on the Filters shelf). This option lets you decide when a table calculation filter should be applied to totals. For details, see "Filter table calculations" in **Filter Data from Your Views** on page 1162.
Map data updates and additions

The following data has been updated for Tableau version 10.3:

- French region names were updated to the final 2016 naming scheme.
- NUTS (Nomenclature of Territorial Units for Statistics) codes and Airport codes no longer require country as a parent to geocode on a map.
- Netherlands landen and provincies were updated.
- Australian LGAs were updated with 2016 data.

Additionally, Rwanda districts (uturere) are now supported by Tableau, and can be geocoded using the County geographic role. Polygons for New Zealand postcodes were also added. For more information about additional map data supported by Tableau, see Location Data that Tableau Supports for Building Map Views on page 1905.

Evenly distribute a layout container's items

After you add a layout container to a dashboard, you can make the items it contains evenly spaced using the Distribute Evenly command (on the desktop and in web authoring). For details, see Size and Lay Out Your Dashboard on page 2257.
Easy access to story actions

The new story toolbar appears when you mouse-over the story navigation area. Use it to revert story changes, apply updates to a story point, delete a story point, or create a new story point out of the current, customized one.

Use numbers for story navigation

The story Layout tab includes a new navigation option: Numbers. Use this option to make the story navigator display a sequence of numbers instead of caption boxes or dots. For details, see Create a Story on page 2325.

New to Web Authoring

Web authoring topics for Tableau Server and Tableau Online are now located in Tableau Help, starting with version 10.3. To see web authoring features from past releases, see What's New in Tableau Server.

Edit quantitative color legends

You can now edit the color palette for continuous color legends in web authoring. You can open the Edit Colors dialog box from the Marks card or by clicking the drop-down arrow on the legend.
You can also set a custom color for the start and end colors by entering the Hex value.

Edit colors on separated legends

When you create separate color legends for measures in your view, you are no longer restricted to using only the default color palette or the color palette assigned to each color legend when the view was published from Tableau Desktop.
In web authoring mode, you can now select different color palettes for each legend. Click the drop-down arrow on the legend to open the Edit Colors dialog box and then select the color palette. You can also set custom start and end colors using Hex values.

Customize how people interact with your map

In web authoring mode, you can now customize how your audience interacts with your map view in the following ways using the Map Options dialog box:

- Show a map scale
- Hide map search
- Hide the view toolbar
- Turn off pan and zoom

For more information, see Customize How People Interact with your Map on page 2069.

Format numbers on the web

You can now specify basic number formatting for measures in the view on Tableau Server and Tableau Online. For more information, see Format Numbers and Null Values on Tableau Server or Tableau Online.
Create stories

In web authoring, you can now create a story to tell a data narrative, provide context, demonstrate how decisions relate to outcomes, or to simply make a compelling case. See Create a Story on page 2325 for details.
Create and edit bins

Create bins from continuous measures and edit them in Tableau Server and Tableau Online, similar to how you do it in Tableau Desktop. For details, see Create Bins from a Continuous Measure on page 956.

Drag fields to Show Me in the view

Select and drag dimensions and measures of interest to the view area. A "Show Me" view is automatically created. You can then click other Show Me options to try different view types.
Drill up and down a continuous hierarchy in the view

In a view with a continuous hierarchy, hover near the headers on a continuous axis to display the + and - controls. Click to drill down or up.
Save data source

You now have the option to save a data source that is embedded with a published workbook, as a separate, published data source on Tableau Server and Tableau Online that other users can connect to. When you save the data source, you can choose to update the workbook to connect to the newly saved data source.

Share and Collaborate
Data-driven alerts from Tableau Online and Tableau Server

When data reaches important thresholds for your business, data-driven alerts automatically send email notifications to key people you specify. You can set alerts on all chart types except Gantt charts and maps.

To begin, select a continuous numeric axis of a chart:

Then click Alert in the toolbar:

When the Create Alert pop-up window appears, set the data condition and threshold value that triggers alert emails. In the view, a red line shows where the threshold falls relative to current data. Simply specify the email subject line, schedule, and recipients to start alerting your organization about critical metrics.
To manage alerts you own or receive, click your name at the top right corner of a Tableau Online or Tableau Server page, click My Content, and then click Alerts.

For more details, see Send Data-Driven Alerts from Tableau Online or Tableau Server on page 2669.

Added in version 10.2

Install and Deploy Tableau

- Install Wizard for Tableau Desktop on the Mac on the next page
- Check for product maintenance updates on page 68

Connect to and Prepare Data

- Salesforce support for cross-database joins on page 69
- Union more data on page 69
- Improved date recognition on page 72
- Connect to more Presto data on page 73
- New SharePoint Lists connector on page 73
• New Apache Drill connector on page 73
• Connect to Aster Database on a Mac on page 73

Design Views and Analyze Data

• Expandable marks cards on page 69
• Legends per measure on page 70
• Connect to Spatial files on page 72
• Show a scale on your map on page 73
• Geocode Nomenclature of Territorial Units for Statistics (NUTS) codes on page 73
• Map data updates and additions on page 74
• Customize joins using calculations to resolve format mismatches between fields on page 75
• New correlation and covariance functions on page 76
• TLS mutual authentication on Mac on page 76
• Use numbers for story point navigation on page 76
• Format lines workbook-wide on page 77
• Support for WCAG-conformant views on the web on page 77

Publish Data Sources and Workbooks

• Automatically save your workbooks on the next page
• Quickly share content with Tableau Online on page 75
• Export your data to CSV on page 75
• Support for SAP BW Single Sign-On on page 78

Install Wizard for Tableau Desktop on the Mac

The new customizable install wizard guides you through the installation process. It installs the most commonly used database drivers by default (Amazon Redshift, MySQL, Oracle, PostgreSQL, Simba Spark and Simba SQL) and creates a desktop shortcut.
For more information about installing Tableau Desktop, see the Tableau Desktop Deployment Guide.

Check for product maintenance updates

Instead of waiting for a notification that a product maintenance update is available to download, or dismissing the prompt and then wishing you hadn’t, you can now check for these updates at any time. Simply select Check for Product Updates from the Help menu in Tableau Desktop and Tableau will check to see if a more current maintenance release version is available. One of the following options will display:

- If new updates are available, a dialog box displays to enable you to download and install the latest updates.
- If no new updates are available, a message is shown indicating that no new updates were found.
- If an update is already in progress, a message is shown indicating that the update is in progress.

The same 3-day waiting period that applies to the automatic update process applies to this feature. If automatic updates are turned off for Tableau, then this option is not available. For more information about how to manage automatic updates for your product, see Control Product Updates in the Tableau Desktop Deployment guide.

Automatically save your workbooks

Tableau Desktop now automatically saves your work for you every few minutes - no more losing hours of work if Tableau closes unexpectedly.

The new Autosave feature is turned on by default. If Tableau crashes, a recovered version of the workbook is automatically created with a .twbr extension and saved in the same location as the original file or in your My Tableau Repository/Workbooks folder. New workbooks are saved with the name “Book1” plus a numeric ID. When you reopen Tableau, a recovery dialog box shows a list of the recovered files that you can select and open to continue in your flow. You can also delete unwanted files from this same dialog box.
Users can turn this feature on or off from the Help menu in Tableau Desktop. Administrators can also turn this feature off from the command line by changing the AUTOSAVE installer property setting. If the Administrator turns this feature off, then the Autosave option is not available from the Help menu.

For more information about how to turn the Autosave feature on or off using the command line, see Deploy Tableau Desktop in the Tableau Desktop Deployment guide.

Salesforce support for cross-database joins

Salesforce supports combining tables using a cross-database join. For more information, see the "Combine tables from different databases" section in Join Your Data on page 657.

Union more data

You can union tables from Amazon Redshift, Google BigQuery, HP Vertica, Microsoft SQL, MySQL, Oracle, and PostgreSQL databases. You can also union JSON files. For more information about union, see Union Your Data on page 708.

Expandable marks cards

If you ever work with views that need a large number of fields on the marks card but are frustrated because you can't see them all without scrolling, the marks card has now been enhanced to address this. The marks card now automatically expands to fill all of the empty space that is available.
Legends per measure

When you include the **Measure Values** and **Measure Names** fields in your view, you can now color code your data by measure, enabling you to further differentiate sections of your data.

In prior releases, when you used these fields and added **Measure Values** to Color on the Marks card, Tableau would create a single color legend that applied to all the marks in the view. Now, when you drag **Measure Values** to Color on the Marks card, you can select the option to create separate color legends for each measure in the view and then select a unique color palette for each one.

You can also easily recombine the separate color legends at any time by selecting the option to combine legends. Tableau even remembers your color palettes when you toggle between combined legends and separate legends.

**Single combined legend**

The example below shows the default behavior when you drag **Measure Values** to Color on the Marks card. Tableau creates a single color legend that applies to all marks in the view.
On the Marks card, click the drop-down arrow on the **Measure Values** field, then select **Use Separate Legends** from the context menu to create a legend for each measure. Tableau assigns the default color palette to each legend, but you can edit it as needed, as shown in the example below.

**Multiple legends**

To recombine the separate legends into a single legend for all measures, on the Marks card, click the drop-down arrow on the **Measure Values** field and select **Combine Legends** from the context menu.
Improved date recognition

When your date values are interpreted as strings or integers, change the data type of the field to **Date** or **Date & Time** instead of creating a calculation to change its type. For more information, see Convert a Field to a Date Field on page 740.

Connect to Spatial files

You can now connect to Shapefiles, MapInfo tables, KML (Keyhole Markup Language) files, and GeoJSON files in Tableau Desktop. For more information, see the Create Tableau Maps from Spatial Files on page 1959 topic and the Spatial File on page 1906 example connector topic.
Connect to more Presto data

In addition to connecting to Presto 141t from Teradata, Tableau now connects to Presto on-premises and Amazon EMR Presto instances for Presto versions 0.148. For more information about the Presto connector, see Presto on page 528.

New SharePoint Lists connector

Use the SharePoint Lists connector to connect to SharePoint lists. For more information, see SharePoint Lists on page 570.

New Apache Drill connector

Use the Apache Drill connector to connect to Apache Drill data. For more information, see Apache Drill on page 401.

Connect to Aster Database on a Mac

You can connect to an Aster Database on a Mac. You can get the driver for the Mac from the Driver Download page on the Tableau website.

Show a scale on your map

You can now display a scale on your maps so that your audience can understand distances between your data points. For more information, see Show a map scale on page 2069.

Geocode Nomenclature of Territorial Units for Statistics (NUTS) codes

You can now geocode NUTS 1 - 3 level codes in Tableau using the NUTS Europe geographic role.

To assign the NUTS Europe geographic role to a field:
- In the Data pane, click the data type icon next to the field, select **Geographic Role**, and then select **NUTS Europe** from the list.

For more information, see *Types of geographic roles in Tableau* on page 1944.

**Map data updates and additions**

India districts and postcodes, New Zealand postcodes, Australia LGAs, and Saudi Arabia cities data has been updated.

Additionally, second-level administrative divisions (county-equivalents) for the following cities and countries are now automatically recognized by Tableau, and can be geocoded using the County geographic role. For more information, see *Location Data that Tableau Supports for Building Map Views* on page 1905.

- Berlin
- Burkina Faso
- Ethiopia
- Guinea
- Guinea-Bissau
- Indonesia
- Ivory Coast
- Malaysia
- Mali
- Myanmar
- Nigeria
- Norway
- Philippines
- Senegal
- Sierra Leone
- The Gambia
- Togo
- Zambia
Customize joins using calculations to resolve format mismatches between fields

A format mismatch between fields used in a join can often cause a broken join or prevent you from being able to create a join in the first place. In many cases, you can resolve format mismatches between fields by customizing your join using a calculation. For more information, see Join Your Data on page 657.

Quickly share content with Tableau Online

To quickly share content with others, sign up for a free trial to Tableau Online: click the Share button in the toolbar, and then click Create Site. (If you’re already signed in to Tableau Online or Tableau Server, clicking Share displays the standard publishing options.)

Export your data to CSV

Export to a CSV file all the data that you combine and prepare in Tableau without the need to create or use a baseline view first. For more information, see Export Data on page 2477.
New correlation and covariance functions

Three new aggregate functions are available:

- **CORR**: Returns the Pearson correlation coefficient of two expressions.
- **COVAR**: Returns the sample covariance of two expressions.
- **COVARP**: Returns the population covariance of two expressions.

See Tableau Functions (Alphabetical) on page 1449 for details.

In addition, three related table calculation functions are available:

- **WINDOW_CORR**: Returns the Pearson correlation coefficient of two expressions within the defined window.
- **WINDOW_COVAR**: Returns the sample covariance of two expressions within the defined window.
- **WINDOW_COVARP**: Returns the population covariance of two expressions within the defined window.

See Table Calculation Functions on page 1347 for details.

TLS mutual authentication on Mac

Tableau Desktop for Mac now supports TLS mutual authentication with Tableau Server. If your organization uses client certificates for computer or user authentication, and mutual authentication has been enabled on Tableau Server, then you can authenticate as you would with any other application that supports TLS mutual authentication. Tableau Desktop on Windows also supports TLS mutual authentication.

Use numbers for story point navigation

You can now specify numbers as a way for your audience to step through your story. This can be helpful if you have a large number of individual story points and want to avoid scroll bars on the navigator.
For more information on story layout and formatting, see Create a Story on page 2325.

Format lines workbook-wide

You can quickly change line formatting for every view in your workbook in one spot by clicking Format > Workbook. When you make changes to your workbook's line settings, a gray dot appears next to the setting in the Format Workbook pane.

For more details, see Format at the Workbook Level on page 2356.

Support for WCAG-conformant views on the web

Embedded views that you author in Tableau Desktop, publish to Tableau Server or Tableau Online, and embed in WCAG-conformant web pages, now meet WCAG 2.0 AA conformance
and section 508 standards.

Views that are made available to users in Tableau Desktop, Tableau Reader, or Tableau Public are not yet supported for WCAG conformance. Views created or edited on the web in Tableau Server or Tableau Online are not yet supported for WCAG conformance.

The following parts of the embedded view are WCAG-conformant:

- Sheet tabs
- The view area of the sheet
- Titles and captions
- Categorical legends
- Single Value (list) filters, which are radio buttons, and Multiple Values (list) filters, which are check boxes.
- The View Data window

Some aspects of the Tableau interface in the embedded views are not yet supported, such as other filter types, parameters, and the view toolbar. Using these controls in your visualizations might make it difficult for users who use assistive technologies to fully understand your views.

Keyboard navigation, programmatic context for assistive technologies (using ARIA roles), alternative text for non-text elements and contrast standards, and authentication when signing into Tableau Server or Tableau Online are also supported for embedded views.

For more information about creating accessible views, see Best Practices for Designing Accessible Views on page 2119 and Author Views for Accessibility on page 2127.

Support for SAP BW Single Sign-On

You can configure single sign-on (SSO) support from Tableau to SAP NetWeaver Business Warehouse (SAP BW). For more information, see Enable Single Sign-On for SAP BW in the Tableau Server Help.

Added in version 10.1
Changes to the Installer below

Join tables using right and full outer join types below

Union data using wildcard search on the next page

Geocode airports on the next page

Map data updates on the next page

New JSON file connector on page 81

New Anaplan connector on page 81

New Oracle Eloqua connector on page 81

Added initial SQL parameter support on page 81

Teradata support for encryption on page 82

IBM PDA support for SSL on page 81

SAP HANA SQL query-based prompts support on page 82

Google BigQuery feature enhancements on page 82

SAP NetWeaver Business Warehouse feature enhancements on page 82

Changes to the Installer

When you install Tableau Desktop, Tableau Reader or Tableau Public, you might want to perform a quiet installation. When you perform a quiet installation, you must accept the End User License Agreement (EULA) by setting the ACCEPTEULA option to "1".

For example: `tableau.exe /quiet ACCEPTEULA="1"`.

For more information, see Deploy Tableau Desktop in the Tableau Desktop Deployment guide.

When you install a new version of Tableau Public, the installer will overwrite or remove any previous versions that you have on your computer. You can have only one instance of Tableau Public on your computer at one time.

Join tables using right and full outer join types

Use right and full outer join types when joining Excel tables to Excel tables, text file tables to text file tables, or tables in a cross-database join. **Note:** For cross-database joins, right and full
outer joins are supported only if the underlying database or file supports right and full outer join types.

**Union data using wildcard search**

In addition to unioning tables manually, you can now union tables automatically by specifying a wildcard search. For more information, see *Union Your Data* on page 708.

**Geocode airports**

You can now geocode airports in Tableau using the International Air Transport Association (IATA) and International Civil Aviation Organization (ICAO) airport codes.

To assign the Airport geographic role to a field:

- In the Data pane, click the data type icon next to the field, select *Geographic Role*, and then select the *Airport* geographic role.

For more information, see *Types of geographic roles in Tableau* on page 1944 and *Assign a geographic role to a field* on page 1942.

**Map data updates**

Netherlands municipalities are now automatically recognized by Tableau and can be geocoded with the County geographic role. For more information, see *Location Data that Tableau Supports for Building Map Views* on page 1905.

Postcodes for South Africa and Thailand are now automatically recognized by Tableau and can be geocoded with the Zip Code/Post Code geographic role. For more information, see *Location Data that Tableau Supports for Building Map Views* on page 1905.

Postcode data has been updated in the following countries:

- Belgium
- Czech Republic
- Denmark
- France
- Germany
- Italy
- Netherlands
- Spain
- Slovakia
- Switzerland
- Turkey
- United States

New JSON file connector

Use the JSON file connector to connect to a local JSON file. For more information, see JSON File on page 338.

New Anaplan connector

Starting in 10.1.2, use the Anaplan connector to connect to Anaplan data. For more information, see Anaplan on page 397.

New Oracle Eloqua connector

Starting in 10.1.2, use the Oracle Eloqua connector to connect to Oracle Eloqua data. For more information, see Oracle Eloqua on page 510.

Added initial SQL parameter support

Initial SQL parameter support is added for a number of connectors, including Amazon EMR, MapR Hadoop Hive, Cloudera Hadoop, HP Vertica, and Spark SQL. For more information about initial SQL, see Run Initial SQL on page 643.

IBM PDA support for SSL

IBM PDA (Netezza) supports connecting to an SSL server. For more information, see IBM PDA (Netezza) on page 457.
Teradata support for encryption

You can now use encryption with your connections to Teradata. For more information, see Teradata on page 580.

SAP HANA SQL query-based prompts support

You can now use your SQL query-based SAP HANA prompts in Tableau. For more information, see SAP HANA on page 541.

Google BigQuery feature enhancements

Support has changed from BigQuery legacy SQL (BQL) to standard SQL. Your workbooks will upgrade when you open them in Tableau. With standard SQL comes other benefits for BigQuery connections including:

- Support for level of detail calculations.
- Improved speed of creating an extract.
- Faster validation of custom SQL.
- The ability to select a billing project for each connection.

SAP NetWeaver Business Warehouse feature enhancements

- Tableau now provides a default schema selection.
- Search functionality is added for selecting a catalog or InfoProvider, and a cube or query.

Added in version 10.0

- Retain connection details on the next page
- New connectors on page 84
• **Size marks on an axis** on the next page
• **New table calculation experience** on page 85
• **Display titles in worksheets** on page 86
• **Combine data with cross-database joins** on page 86
• **Highlight data points in context** on page 86
• **LOD expressions: Dimension expressions supported in dimensionality declaration** on page 87
• **Find clusters in data** on page 88
• **Oracle support for table functions and for Kerberos** on page 88
• **SQL Server support for contained databases** on page 89
• **Use JDBC to connect to SAP HANA on a Mac** on page 89
• **Create device layouts for a dashboard** on page 89
• **Format at the workbook level** on page 90
• **Create territories on a map** on page 91
• **Measure distance in metric or imperial (U.S.) units** on page 92
• **Map data updates** on page 92
• **Update to the default map server address** on page 93
• **Filter data across multiple primary data sources** on page 93
• **On-demand connections** on page 93

### Retain connection details

After you connect to a database, Tableau 10.0 stores every connection detail that you enter in the Connection dialog box except the password. The next time you connect to that database, the only thing you have to enter is the password. These settings are stored per user, so shared computers do not share connection settings.
New connectors

New Google Sheets connector
Use the Google Sheets connector to connect to a Google Sheets data source. For more information, see Google Sheets on page 441.

New Presto connector
Use the Presto connector to connect to Presto 141t from Teradata. For more information, see Presto on page 528.

New MemSQL connector
Use the MemSQL connector to connect to MemSQL data. MemSQL uses a MySQL driver, so if you already have a MySQL driver installed, you can connect to MemSQL. For more information, see MemSQL on page 478.

New QuickBooks Online connector
Use the QuickBooks Online connector to connect to QuickBooks Online data. For more information, see Intuit QuickBooks Online on page 459.

Size marks on an axis
For views where the mark type is Bar and there are continuous fields on both Rows and Columns, Tableau 10.0 adds options for sizing the bar marks on the axis where the bars are anchored.

- The bar marks in histograms are continuous by default (with no spaces between the marks), and are sized to match the size of the bins. See Build a Histogram on page 881 for an example.
- When there is a field on Size, you can determine the width of the bar marks on the axis where the bars are anchored using the field on Size. To do this, click the Size card and choose Fixed.
- When there is no field on Size, you can specify the width of the bar marks on the axis
where the bars are anchored in axis units. To do this, click the **Size** card, choose **Fixed**, and then type a number in the **Width in axis units** field.

- When there is a continuous date field on the axis where the bars are anchored, the width of the marks is set to match the level of the date field. For example, if the level of the continuous date field is MONTH, the bars are exactly one month wide—that is, slightly wider for 31-day months than for 30-day months. You can configure the width of the bars by clicking the **Size** card, choosing **Fixed**, and then typing a number in the **Width in days** field, but the resulting bar widths don't take account of the varying lengths of time units, such as months and years.

### New table calculation experience

Tableau 10.0 enhances the experience of creating table calculations:

- When you work in the Table Calculation dialog box to configure a calculation, the view updates in real time to show the results of your choices.

- **Highlighting** shows you the scope and direction of your calculation by setting off part of the view with a colored background. As you change the **Compute Using** option in the Table Calculation dialog box, the highlighting reflects your changes.

See [Transform Values with Table Calculations](#) on page 1524 for details.
Display titles in worksheets

Titles are now turned on automatically in worksheets and appear as part of the view. You can create polished presentations using your worksheet without creating a separate dashboard.

Combine data with cross-database joins

Combine data from different databases using a cross-database join. For example, you can combine related data from tables stored in an Excel workbook and a SQL Server database. For more information, see Join Your Data on page 657.

Highlight data points in context

The Highlighter is a new interactive exploration tool that you can use to quickly find and highlight specific data points while maintaining the context of the data. It is simple and powerful.

To use this new tool, do the following:

1. Select Show Highlighter from the context menu on a discrete field that is used in the view and that affects the level of detail in the view.

2. Search for the data you are interested in (a specific mark or group of marks) using a keyword search, or select a mark from the drop-down list.

   The marks are highlighted in the view.
You can also perform ad hoc comparisons with instant highlighting when you hover over items in the drop-down list.

**LOD expressions: Dimension expressions supported in dimensionality declaration**

You can now use any expression that evaluates as a dimension, including Date expressions, in the dimensionality declaration for a level of detail expression.

This example excludes a discrete date bin in the view:

```
{EXCLUDE YEAR(Date) : SUM([Sales])}
```

This example excludes the YEAR(Date), which is a discrete date bin in the view:

```
{EXCLUDE DATEPART('year',Date) : SUM([Sales])}
```

This example excludes the YEAR(Date), which is a continuous date field in the view:

```
{EXCLUDE DATETRUNC('year', Date): SUM([Sales])}
```

For information about level of detail expression syntax, see *How Level of Detail Expressions Work in Tableau* on page 1585.
Find clusters in data

You can now find meaningful clusters in your data by dragging **Cluster** from the **Analytics** pane and dropping it in the view. The result is a statistical object that Tableau places on **Color**. If there is already a field on **Color**, Tableau moves that field to **Label**.

Cluster analysis partitions marks in the view into clusters, where the data values within each cluster are more similar to one another than they are data values in other clusters.

You can drag a cluster from **Color** to the **Data** pane to make it a group dimension in which the individual members (Cluster 1, Cluster 2, etc.) contain the marks that the cluster algorithm has determined are more similar to each other than they are to other marks.

See **Find Clusters in Data** on page 1680 for details.

Oracle support for table functions and for Kerberos

One way to simplify data access in Oracle is to create table functions, which return tables while hiding complex logic and exposing parameters. When you connect to Oracle, you can use table
functions, which appear under **Stored Procedures** on the data source page.

Tableau supports Kerberos for Oracle. Enterprise environments that use Active Directory with Tableau can now have a seamless single sign-on experience with Oracle and Tableau.

For more information, see [Oracle](#) on page 506.

**SQL Server support for contained databases**

Microsoft SQL Server and Azure SQL support a new feature called *contained databases*. When a database is *contained*, user information is stored in the database rather than in the database management system, which makes the database more portable. In Tableau 10.0, you can connect to contained databases by entering the database name in the connection dialog box. See [Microsoft SQL Server](#) on page 487.

**Use JDBC to connect to SAP HANA on a Mac**

You can now connect to SAP HANA on a Mac using JDBC drivers. For more information, see [SAP HANA](#) on page 541.

**Create device layouts for a dashboard**

After you’ve built a dashboard you can create layouts for it that are specific to particular devices.

When you publish the dashboard to Tableau Server or Tableau Online, people who interact with it experience a dashboard expressly designed for their screen display, whether it’s a phone, tablet, or desktop. For more information, see [Create Dashboard Layouts for Different Device Types](#) on page 2302.
Format at the workbook level

You can quickly change how fonts and titles look in every view in a workbook by specifying format settings at the workbook level instead of at the worksheet level. See Format at the Workbook Level on page 2356 for more information.
Create territories on a map

You can create territories on a map view by grouping existing locations (location points or polygons) together. For more information, see Create Territories on a Map on page 2061.
Measure distance in metric or imperial (U.S.) units

You can now choose to measure in meters and kilometers or feet and miles when you measure distances on a map with the Radial tool. For more information, see Change the units of measurement on page 2093.

Map data updates

Hong Kong districts; Singapore URA subzones, South Korea Sigungu (Metropolitan Areas); and Taiwan cities and districts are now automatically recognized by Tableau and can be geocoded with the County geographic role. For more information, see Location Data that Tableau Supports for Building Map Views on page 1905.

Postcodes for Argentina, Brazil, Costa Rica, India, Mexico, Singapore, and Taiwan are now automatically recognized by Tableau and can be geocoded with the Zip Code/Post Code geographic role. For more information, see Location Data that Tableau Supports for Building Map Views on page 1905.
Update to the default map server address

When configuring Tableau Desktop and Tableau Server proxy settings, use maps.tableau.com to connect to the default map server. For more information, see Internet Access Requirements in the Tableau Desktop Deployment Guide.

Filter data across multiple primary data sources

You can apply a filter across multiple primary data sources. For more information, see Filter Data Across Multiple Data Sources on page 1185.

On-demand connections

When you load a workbook, Tableau connects only to the data sources that are required to display your data. This means that if there are multiple sheets in a workbook, Tableau only connects to the data sources for the current sheet so that you can view your data as soon as possible. For data sources that require authentication, now you only have to enter your credentials for the current sheet.

Added in version 9.3

- Desktop installer improvements on the next page
- Stay connected to Tableau Server or Tableau Online on page 95
- Forecasting improvements on page 96
- New data source support for Tableau functions and aggregations on page 97
- Add color to sheets in sorter view on page 97
- Workbook updates to use published data source on page 97
- Union your data on page 98
- Data grid enhancements on page 98
- Zooming improvements for maps on page 98
- WMS Support for Web Mercator on page 99
- New default tool for maps on page 99
- View tools remain active on page 99
Desktop installer improvements

To streamline the install experience for Tableau Desktop for Windows, the four most common database drivers are now included as part of the setup process and are automatically installed by default. The following drivers are included:

- MySQL (version 5.3.6)
- Microsoft SQL Server (version 2008 R2 SP3)
- PostgreSQL (version 9.3.400)
- Amazon Redshift (version 1.2.1)

**Note:** Microsoft Visual C++ Redistributable is also automatically installed as a prerequisite for MySQL.

To view the list of drivers, or to clear the option to install these drivers, accept the license agreement on the installer page, and then click **Customize**.
You can clear the option to install the drivers in either the user interface or from the command line.

If you want to perform a quiet install for Tableau Desktop 9.3, use the installer (.exe) file. The installer package (.msi) is included as part of the installer file and can no longer be extracted separately.

For more information about how to perform a quiet install of Tableau Desktop, see Deploy Tableau Desktop in the Tableau Desktop Deployment guide.

Stay connected to Tableau Server or Tableau Online

When you connect to Tableau Server or Tableau Online, Tableau Desktop saves each connection from session to session if you don’t sign out. Next time you launch Tableau Desktop, you are signed in to your most recent server connection. You can also sign into different servers and easily switch between your available server and site connections.
Forecasting improvements

Forecasting in Tableau Desktop has been improved in the following ways:

- You can now forecast values ordered by an integer dimension. Formerly it was only possible to forecast measures that are ordered by time. For more information, see Forecasting When No Date is in the View on page 1717.

- Automatic selection of the best season length for a forecast has been improved. For more information, see the Seasonality section in How Forecasting Works in Tableau on page 1706.

- Tableau forecasting will now skip null or missing values when estimating model parameters. However, the nulls remain in the series so the indexes of the values remain the same. For example, for a daily time series that is missing weekend days, the algorithm computes the model, potentially with parameters for a weekly cycle, but without using the missing values. But because the missing values are in the series, the cycle length is still seven days.

Due to this change:

- Forecasting now supports irregular time series without zero fill. So no error will result if you do not select Fill in missing values with zeros in the Forecast
Options dialog box.

- The Fill in missing values with zeros option is no longer selected by default in the Forecast Options dialog box.

New data source support for Tableau functions and aggregations

For Tableau 9.3, more functions and aggregations are available for more data sources:

- The PERCENTILE function and aggregation are now available for Cloudera Hive, Hortonworks Hadoop Hive, and EXASolution data sources.
- The MEDIAN function and aggregation are now available for Cloudera Hive, Hortonworks Hadoop Hive, and EXASolution data sources.
- Regular expression functions (beginning with REGEXP_) are now available for HP Vertica, Pivotal Greenplum, and Teradata (version 14.1 and above) data sources. See Additional Functions on page 1373. In addition, REGEXP_MATCH is now supported for Impala 2.3.0 and higher in Cloudera Hadoop data sources.
- The SPLIT function is now supported for Impala 2.3.0 and higher in Cloudera Hadoop data sources. Split functionality in the Data source pane is also now supported for Impala 2.3.0 and higher in Cloudera Hadoop data sources. See Split a Field into Multiple Fields on page 761.

Add color to sheets in sorter view

To make it easier to identify or group worksheets, you can now add color to the sheet tabs in the sheet sorter.

Workbook updates to use published data source

When you publish a data source to Tableau Server or Tableau Online, the workbook you’re publishing it from is updated to connect to the published data source. At the same time the local data source is closed. To continue using the local data source, clear the Update workbook to
use the published data source check box in the Publish data source to Tableau Server dialog box.

If you click Undo after publishing a data source, Tableau will revert to using the local data source, but it will not "un-publish" the data source.

Tableau does not replace the local data source when you publish a cube (multidimensional) data source.

For more information, see Publish a Data Source on page 2516.

Union your data

In your Tableau data source, you can now append rows of data from one table to another table thereby creating unions. If columns in the union do not align correctly, you can merge the columns whose names don't match. For more information, see Union Your Data on page 708.

Data grid enhancements

From the grid on the Data Source page, you can now do the following:

- See extract data and extract-only calculations, including data from the Web Data Connector
- Review data after applying extract filters and aggregating extracts
- Create groups and bins
- Join on pivot columns
- Join on merged columns
- See sheet and table name information for unioned tables

Zooming improvements for maps

Zooming in map views just got easier. You can now scroll to zoom in and out of a point on a map.
WMS Support for Web Mercator

Tableau now supports WMS servers that use Web Mercator. For more information, see Supported Spatial Reference Systems on page 2081 in the Use WMS Servers topic.

New default tool for maps

When you click and drag in a map view, you can now automatically select marks with the rectangular tool. This is a change in behavior to Tableau Desktop version 9.2, where clicking and dragging in a map view would automatically allow you to pan across the view instead.

View tools remain active

When you select a tool from the view toolbar (the pan tool, zoom area tool, radial tool, rectangular tool, or lasso tool), it remains active until you select to use another tool. The same is true if you use keyboard shortcuts to switch between tools. This is a change in behavior to Tableau Desktop version 9.2, where a tool would revert back to the default after one use.

The last tool you select will be saved with the workbook and available when you reopen it or publish it to the web.

Totals are no longer included in color encoding

Totals are no longer color encoded by default when you add them to the view.

To include totals in color encoding, click Color on the Marks card and then select Edit Colors. In the Edit Colors dialog box, select Include Totals.

| Default | Totals included in color encoding |
New Snowflake data connector

Use the Snowflake connector to connect to a Snowflake data warehouse. For more information, see **Snowflake** on page 573.

New Kognitio data connector

Starting in 9.3.1, use the Kognitio connector to connect to a Kognitio database. For more information, see **Kognitio** on page 464.

New Cisco Information Server data connector

Starting in 9.3.1, use the Cisco Information Server connector to connect to a Cisco Information Server virtual database. For more information, see **TIBCO Data Virtualization** on page 591.
Data preview for Web Data Connectors

When you connect using a Web Data Connector, Tableau now opens on the Data Source page so that you can prepare your data (for example, change data types or hide columns) before you start your analysis.

Kerberos support for PostgreSQL and Teradata

Kerberos support has been added for PostgreSQL and Teradata connections. For more information, see PostgreSQL on page 525 or Teradata on page 580. For information about configuring Tableau Server for Kerberos, see Kerberos in the Tableau Server Help.

OAuth support for Salesforce in Tableau Desktop

You can now use OAuth when you connect to Salesforce in Tableau Desktop. After you provide your Salesforce credentials and allow Tableau access to the data, Salesforce.com creates an OAuth access token that is used to connect to the data. Instead of having to embed your credentials in data sources or workbooks, you can use the access token.

Initial SQL support added to more data sources

Oracle, Pivotal Greenplum, and Microsoft SQL Server now support initial SQL statements.

Initial SQL parameter support

For data source connections that support initial SQL, you can now perform parameter substitution for a set of useful parameters, such as ServerUser and WorkbookName. For more information, see Run Initial SQL on page 643.

Added in version 9.2
- Google Analytics query enhancements below
- Recent colors save to the application below
- Data grid enhancements on the next page
- Data Interpreter enhancements on the next page
- Data pane enhancements on the next page
- Match mark label color to mark color on page 104
- Label most recent marks on page 105
- Fix one or both ends of an axis on page 106
- Move totals to the top or left of the view on page 107
- New Map Options dialog box on page 108
- Support for Mapbox maps on page 108
- SAP HANA column labels on page 109
- Query improvements on page 109
- Show Quick Filter has changed to Show Filter on page 109

Google Analytics query enhancements

By default, Tableau returns all data in a Google Analytics query to avoid returning sampled data. See Google Analytics on page 426.

![Query returns: All data. Sample data](image)

Recent colors save to the application

From any Color drop-down control, you can select a recently used color. Up to eight recently used colors save to the application and are available in any workbook.

![More Colors...](image)
Data grid enhancements

The following functionality is now available from the grid on the Data Source page:

- Sort columns and rows
- Double-click the name of the column to rename the field
- Reset the name of a single column or select multiple columns to reset the column names at once
- Create calculations based on existing fields in the data source
- Copy values in the data source using Ctrl+C (Command-C on a Mac).

For more information, see Edit Data Sources on page 817.

Data Interpreter enhancements

The Data Interpreter now detects subtables in your Excel data. For more information, see Excel on page 321.

Data pane enhancements

The following functionality is now available in the Data pane:

- Click the icon to the left of a field in the Data pane to change the field’s data type or geographic role:
- Click a field and hold down the mouse button to rename the field:

- Enhanced Search options in the Data pane. Click the search icon at the top of the Data pane or press Ctrl + F (Command-F on a Mac) to initiate a search. As you type in the search box, search now filters the contents of the Data pane to show all fields that contain the typed string. Search remains open until you click the search icon or press Ctrl + F again.

- The zones apportioned for Dimensions, Measures, Parameters, and Sets in the Data pane automatically size so that scroll bars are not shown unless there are more fields than can be displayed in the available space.

**Match mark label color to mark color**

You can select to closely match the color of each label to the color of its mark. See Show and Hide Mark Labels on page 2180 in the Show and Hide Mark Labels topic.
Label most recent marks

When there is a date field in the view, you can select to label the most recent marks in the view. Labels are based on the level of detail of fields in the view. See Show and Hide Mark Labels on page 2180.
Fix one or both ends of an axis

You can specify the values for the start or end of an axis. See Edit Axes on page 1838 in the Edit Axes topic.
Move totals to the top or left of the view

You can select to show totals at the top or left of the view. See **Move totals** on page 1762.
New Map Options dialog box

Use the new Map Options dialog box to customize how your audience interacts with your map view. Use the Map Layers pane (previously the Map Options pane) to customize the appearance of your map. See Customize How People Interact with your Map on page 2069.

Support for Mapbox maps

You can now connect to Mapbox maps in Tableau. See Use Mapbox Maps on page 2073.
SAP HANA column labels

By default, the SAP HANA column names now display the column label, rather than the column name.

Query improvements

The following query improvements have been added for Tableau 9.2.

- **Filtering on categorical aggregates**
  You can now create a filter that aggregates a measure and use it as a domain filter.

- **Numerical bins and combined fields in level of detail expressions**
  You can now refer to a numerical bin or a combined field in a level of detail expression.

- **MIN and MAX supported for boolean data type**
  These aggregations can now be used with boolean fields.

- **Additional optimizations**
  Tableau can now detect “always false” predicates and avoid issuing queries for them.
  For known domains such as boolean, we short circuit to return {true, false} or {true, false, null}. Other rewrites include folding case expressions with constant results to IN SET or NOT IN set.

Show Quick Filter has changed to Show Filter

In Tableau 9.2, the option to Show Quick Filter for a field has changed to Show Filter. For more information, see Filter Data from Your Views on page 1162

Added in Version 9.1

The following new features and enhancements have been added to Tableau Desktop for version 9.1.
Product Updates

To ensure that you always have the most up-to-date features, security resolutions, and corrected issues, Tableau Desktop includes a product update feature. See Turn Product Updates Off or On on page 2740.

Single Sign-On for SAP HANA

When SAP HANA is configured to support single sign-on (SSO), after you sign in to the SAP HANA server, you can access data, and publish data sources and workbooks to Tableau Server, without having to re-enter your user name and password. See Support for SAP HANA single sign-on (SSO) on page 548. Note: Tableau Desktop requires SAP HANA driver version 1.00.85 and later to support SSO for SAP HANA.

Web Data Connector

You can use a web data connector to connect to data that is accessible over HTTP and that doesn't already have a connector. You can create your own web data connector or use one that has been created by someone else. See Web Data Connector on page 596.

SAP Prompts

With SAP HANA and SAP Netweaver Business Warehouse, you can prompt for a variable when a workbook is opened. See SAP HANA on page 541 or SAP NetWeaver Business Warehouse on page 549.
Bin Size Optimization

When you create discrete bins from a continuous measure, Tableau can compute an optimal bin size. See Create Bins from a Continuous Measure on page 956.

Auto-complete Availability

You can now take advantage of auto-completion for formulas in additional locations where you can create formulas—for example, when you add conditions or limits to filters, or when you create a set based on a condition. See (for example) Filter Data from Your Views on page 1162.

Confidence Intervals for Reference Lines

You can now add and configure confidence intervals (bands) for a reference line. See Add a Reference Line on page 1639 in the Reference Lines, Bands, Distributions, and Boxes article for information on the available options. You can no longer add confidence intervals for reference distributions.

Measure Distance with the Radial Tool

Use the Radial tool to measure approximate distances between your data and the locations or landmarks in a map view. For more information see, Measure Distances Between Data Points and Locations in a Map on page 2089.

Turn Off Pan and Zoom

You can turn off pan and zoom to control how your audience interacts with your map view or background image. For more information see, Customize How People Interact with your Map on page 2069

Added in Version 9.0
The following new features and enhancements have been added to Tableau Desktop for version 9.0.

- New start page experience below
- Pivot below
- Split fields into multiple fields below
- Metadata area on the next page
- Data Interpreter on the next page
- New data sources and enhancements to existing data sources on the next page
- Analytics pane on page 115
- Recalculated lines on page 115
- New calculation features on page 115
- Story point formatting on page 117
- Advanced selection tools on page 117
- Map search on page 117
- Configurable tooltip behavior on page 117

New start page experience

The start page in Tableau Desktop is a central location from which you can connect to your data, access most recently used workbooks, and explore content produced by the Tableau community. For more information, see Start Page on page 134.

Pivot

Pivot your Microsoft Excel and text file data sources from crosstab format to columnar format. For more information, see Pivot Data from Columns to Rows on page 747.

Split fields into multiple fields

Split existing string fields into new fields. For more information, see Split a Field into Multiple Fields on page 761.

SPLIT is also available as a Tableau function, for use in calculated fields. For more information, see String Functions on page 1288.
Metadata area

Perform bulk management tasks, like hiding multiple fields at once, using the metadata area. For more information, see Data Source Page on page 139.

Data Interpreter

Detect and remove unique formatting and extraneous information in your Excel data sources. For more information, see Excel on page 321.

New data sources and enhancements to existing data sources

New: Amazon Aurora

Use the Amazon Aurora connector to connect to an Amazon Aurora data source. For more information, see Amazon Aurora on page 386.

New: Spark on Azure HDInsight

Use the Spark SQL connector to connect to a Spark on Azure HDInsight data source. For more information, see Spark SQL on page 576.

New: Microsoft Azure SQL Database

Connect to Microsoft Azure SQL Database data using the Microsoft SQL Server connector. For more information, see Microsoft SQL Server on page 487.

New: Google Cloud SQL

Use the Google Cloud SQL connector to connect to a Google Cloud SQL database instance. For more information, see Google Cloud SQL on page 438.
New: Microsoft Azure SQL Data Warehouse
Connect to Microsoft Azure SQL Data Warehouse data using the Microsoft SQL Server connector. For more information, see Microsoft SQL Server on page 487.

New: Amazon EMR
Use the Amazon EMR connector to connect to an Amazon Elastic MapReduce (EMR) database. For more information, see Amazon EMR Hadoop Hive on page 389.

New: Spark SQL
Use the Spark SQL connector to connect to a Spark SQL cluster (requires Apache Spark 1.2.1 or later). For more information, see Spark SQL on page 576.

New: Statistical File
Use the Statistical File connector to connect to SAS (*.sas7bdat), SPSS (*.sav), and R (*.rdata, *.rda) data files. For more information, see Statistical File on page 357.

Enhanced: MapR Hadoop Hive
MapR Hadoop Hive now supports connectivity for the Mac.

Enhanced: MySQL
SSL encryption is supported. Mac drivers are now included in the TableauDrivers.dmg file. For more information, see MySQL on page 497.

Enhanced: Salesforce
The Salesforce connector now supports more flexible joins. You can create joins on any string fields in addition to fields that are constrained references between tables. For more information, see Salesforce on page 533.
Enhanced: SAP NetWeaver Business Warehouse (BW)

The SAP NetWeaver Business Warehouse connector has been updated to include support for variables. For more information, see SAP NetWeaver Business Warehouse on page 549.

Tableau Data Extracts

Tableau Data Extracts are no longer listed on the Connect pane. Use Other Files to connect to a Tableau Data Extract. For more information, see Other Files on page 372.

Windows Azure Marketplace

To connect to Windows Azure Marketplace, use the OData connector. Workbooks created in earlier versions of Tableau that used the Windows Azure Marketplace DataMarket connector will work as expected. For more information, see OData on page 500.

Analytics pane

The Analytics pane, on the left side of the Tableau workspace, provides quick and easy access to common analytic objects in Tableau. You can drag reference lines, forecasts, trend lines, and other objects into your view from the Analytics pane.

For more information, see Apply Advanced Analysis to a View (Analytics Pane) on page 174.

Recalculated lines

Recalculated lines help you gain insight into how a subset of your data compares to the overall data in your view. For more information, see Explore and Analyze Data in a View on page 2595.

New calculation features

The following features were added to Tableau Desktop to enhance the experience of creating and using calculated fields.
Non-modal calculation editor

The calculation editor has been redesigned to provide interactive editing, intelligent formula-completion, and drag-and-drop support. The calculation editor is now also available when you’re editing a view in Tableau Server or Tableau Online.

For more information, see Get Started with Calculations in Tableau on page 1221.

Ad-hoc calculations

Ad-hoc calculations are calculations that you can create and update on a shelf in the view. Ad-hoc calculations can be useful for testing a hunch, trying a what-if scenario, or debugging a complex calculation.

For more information, see Ad-Hoc Calculations on page 1618.

Auto-completion for formulas

As you type a formula, either in the calculation editor or in an ad-hoc calculation, Tableau displays a list of options for completing the formula.

Level of detail expressions

Level of detail expressions support aggregation using dimensions other than those in the view. With level of detail expressions, you can attach one or more dimensions to any aggregate expression.

For more information, see Create Level of Detail Expressions in Tableau on page 1567.

Functions for regular expressions

Four new functions have been added to the calculation language to support regular expressions.

For more information, see functions beginning with REGEXP in Additional Functions on page 1373.
Functions for hexagonal binning

Hexagonal binning is a technique for clustering data in a two-dimensional plane. Two new functions have been added to the calculation language to support hexagonal binning. These functions support binning data in a single dimension using histograms, or by a geographic level of detail in a map.

For more information, see functions beginning with HEXBIN in Number Functions on page 1275.

Story point formatting

You can now re-size the captions in your story, change the shading and font of the navigator, and select to fit a dashboard to the exact size of a story in your workbook. For more information, see Create a Story on page 2325.

Advanced selection tools

Use the Radial, Rectangular, and Lasso tools on the view toolbar to select multiple marks in the view. For more information, see View Toolbar on page 2610.

Map search

Use map search to find locations on a map view so you can quickly explore and inspect data. For more information, see Search for Locations in Your Map on page 2088.

Configurable tooltip behavior

Show tooltips instantly as you move the mouse over the marks in a view or configure tooltips to display only after resting the mouse on a mark.
Get Started

This section gives you the basics on getting started with Tableau.

- For a walkthrough of how to get started, see Build a Basic View to Explore Your Data below.
- For an in-depth tutorial on how to use Tableau, see Get Started with Tableau Desktop.
- For examples of how you can build basic types of charts and views, see Build Common Chart Types in Data Views on page 844.
- For information on how to build and use maps, see Geographic Data Analysis in Tableau on page 1862.
- If you are using Tableau Online and Tableau Server to explore data and edit views, see Use Tableau on the Web on page 2554.

Watch a video: To see related concepts demonstrated in Tableau, watch Getting Started, a 25-minute free training video. Use your tableau.com account to sign in.

To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Build a Basic View to Explore Your Data

This topic walks through how to create basic views and explore your data. It shows how your view of data in Tableau evolves through your process of exploration.

If you are using Tableau Online and Tableau Server to explore data and edit views, see Use Tableau on the Web on page 2554.

In this article

- Connect to your data
- Create a basic view
Connect to your data

The first step is to connect to the data you want to explore. This example shows how to connect to Sample - Superstore data in Tableau Desktop.

1. Open Tableau. On the start page, under Connect / Saved Data Sources, click Excel.

2. In the Open dialog box, navigate to the Sample - Superstore Excel file on your computer. Go to /My Documents/My Tableau Repository/Datasources/version number/[language]. Select Sample - Superstore, and then click Open.

After you connect to the Excel data, the data source page shows the sheets in your data. Drag the "Orders" to the canvas to start exploring that data.
Depending on how your data is structured, you might need to do more data preparation and integration before you start exploring it in Tableau. For more details on connecting to your data, see Connect to and Prepare Data on page 319 and Tips for Working with Your Data on page 325.

3. Click the sheet tab to go to the new worksheet and begin your analysis.

About the Data pane

In the worksheet, the columns from your data source are shown as fields on the left side in the Data pane. The Data pane contains a variety of different kinds of fields; for now the most important ones are dimensions and measures. Dimensions typically hold categorical data such as product types and dates, while measures hold numeric data such as sales and profit. To learn more, see Dimensions and Measures, Blue and Green on page 250.
If you have more than one data source in a workbook, click the data source connection name in the Data pane to select it for use. For more details, see Navigating Data Sources in the Data Pane on page 169.

For details on the many ways you can customize the fields in the data pane, see Organize and Customize Fields in the Data Pane on page 962, Edit Default Settings for Fields on page 973, and Work with Data Fields in the Data Pane on page 160.

If you have related dimensions, sometimes you might want to group them in a folder, or as a hierarchy. For example, in this data source, Country, State, City, and Postal Code are grouped into a hierarchy named Location. You can drill down into a hierarchy by clicking the + in a field, or drill back up by clicking the - sign in a field.
Build the view

Every view that you build in Tableau should start with a question. What do you want to know?

Every time you drag a field into the view or onto a shelf, you are asking a question about the data. The question will vary depending on where you drag various fields, the types of field, and the order in which you drag fields into the view.

For every question you ask, the view changes to represent the answer visually - with marks (shapes, text, hierarchies, table structures, axes, color).

Different ways to start building a view

When you build a view, you add fields from the Data pane. You can do this in different ways. For example:

- Drag fields from the Data pane and drop them onto the cards and shelves that are part of every Tableau worksheet.
- Double-click one or more fields in the Data pane.
- Select one or more fields in the Data pane and then choose a chart type from Show Me, which identifies the chart types that are appropriate for the fields you selected. For details, see Use Show Me to Start a View on page 2170.
- Drop a field on the Drop field here grid, to start creating a view from a tabular perspective.

For more details on building visualizations with fields, see Start Building a Visualization by Dragging Fields to the View on page 1087.
As you start exploring data in Tableau, you will find there are many ways to build a view. Tableau is extremely flexible, and also very forgiving. As you build a view, if you ever take a path that isn’t answering your question, you can always undo to a previous point in your exploration.

- To undo or redo, click undo ← or redo → on the toolbar.

You can undo all the way back to the last time you opened the workbook. You can undo or redo an unlimited number of times.

---

Build a view from scratch

These steps show how to build a basic view that shows year-by-year profit.

1. From the **Dimensions** area in the **Data** pane, drag the **Order Date** field to the **Columns** shelf. You might need to expand the Order hierarchy to see Order Date.

   When you drag a field over a shelf, a plus sign indicates that the shelf can accept the field.
The resulting table has four columns and one row. Each column header represents a member of the Order Date field (the default date level is YEAR). Each cell contains an “Abc” label, which indicates that the current mark type for this view is text.

![Table Image]

Notice that the field is colored blue, which indicates that it is discrete.

The default date level is determined by the highest level that contains more than one distinct value (for example, multiple years and multiple months). That means that if [Order Date] contained data for only one year but had multiple months, the default level
would be month. You can change the date level using the field menu.

If you're wondering why there are two sets of date levels (from Year down to Day), the first set is for discrete date levels and the second list is for continuous date levels. For more on these concepts, see Convert Fields between Discrete and Continuous on page 980.

2. From Measures in the Data pane, drag the Profit field to the Rows shelf.
The **Profit** field is colored green on the **Rows** shelf, indicating that it is continuous. Also, the field name changes to **SUM(Profit)**. That’s because measures are automatically aggregated when you add them to the view, and the default aggregation for this measure is SUM. For more information on what happens when you add a measure to a view—and why—see *Dimensions and Measures, Blue and Green* on page 250.

Tableau transforms the table into a line chart and creates a vertical axis (along the left side) for the measure.

A line chart is a great way to compare data over time and identify trends effectively.

This line chart shows profit over time. Each point along the line shows the sum of profit for the corresponding year.
The next step is to drill a little deeper into the time dimension.

Drill into the data

This step shows how you can modify the view to show quarters in addition to years. As you drill down into the hierarchy, the view changes to become a nested table.

You can show [Order Date] by quarters using either of the following methods:

- Click the plus button on the left side of the field YEAR(Order Date) field on Columns.
Drag the **Order Date** field (again) from the Data pane and drop it on the **Columns** shelf to the right of **Year(Order Date)**.

The new dimension divides the view into separate panes for each year. Each pane, in turn, has columns for quarters. This view is called a nested table because it displays multiple headers, with quarters nested within years. The word "headers" might be a bit misleading because while the year headers remain at the "head" of the view (that is, the top), the quarter headers are at the foot (that is, the bottom). So it might be more accurate to call them **footers**.

---

**Increase the level of detail with small multiples**

This step shows how you can modify the nested table view to add customer segment. This will create a small multiples view.

Drag the **Segment** dimension from the **Data** pane. Drop it just to the left of **SUM(Profit)** on Rows.
The field is added to the Rows shelf and row headers are created. Each header represents a member of the Segment field.

You could achieve the same result by dropping Segment just to the left of the Profit axis in the view (show below). Tableau often supports multiple ways to add fields to the view.

**Note:** Tableau does not allow you to place a dimension to the right of a measure on either the Rows or Columns shelves because that visual structure would not make sense in the view.
The new dimension divides the view into 12 panes, one for each combination of year and segment. This view is a more complex example of a nested table. Any view that contains this sort of grid of individual charts is referred to as a small multiples view.

![Graph](image)

At this point you probably don't want to make the view any more granular; in fact, you may want to simplify the view by removing (that is, filtering) some data.

**Filter the view to focus your exploration**

This section shows how you can focus your exploration by only showing a subset of the data, such as modifying the view to show only data for orders in 2012 and 2013.

1. Drag the **Order Date** measure from the **Data** pane and drop it on the **Filters** shelf.
2. In the Filter Field dialog box, choose the date level you want to filter on—**Years**. Then click **Next**.

3. In the next pane, clear any two years that you do not want to include in the view.

4. When you are finished, click **OK**.

The view updates to only show data rows where Order Date is 2012 or 2013. Tableau can now allocate more space to the data that interests you.
Next, you will increase the detail in your data exploration by dragging a field to Color on the Marks card.

**Use the Marks card to add depth to your analysis**

This step shows how you can modify the view to color the marks based on the region.

1. Drag the **Region** dimension from the **Data** pane and drop it on **Color**.

   Placing a dimension on **Color** separates the marks according to the members in the dimension, and assigns a unique color to each member. The color legend displays each member name and its associated color.
Each pane now has four lines, one for each region. This view now shows profit for each customer segment and region for 2012 and 2013.
For more details on the Marks card and level of detail, see *Shelves and Cards Reference* on page 185, *Marks* on page 213, and *How dimensions affect the level of detail in the view* on page 255.

To try your hand at building more views, see *Build Common Chart Types in Data Views* on page 844 and *Edit Tableau Views on the Web* on page 2586.

For an in-depth tutorial on how to use Tableau, see *Get Started with Tableau Desktop*.

If you are using Tableau Online and Tableau Server to explore data and edit views, see *Use Tableau on the Web* on page 2554.

**Tour the Tableau Environment**

This section provides information on Tableau pages and its workspace, including panes, shelves, icons, and other elements in Tableau Desktop.

- **The Tableau Workspace** on page 142
- **Start Page** below
- **Data Source Page** on page 139

If you are using Tableau on the web, see *Creators: Get Started with Web Authoring* on page 2566 and *Tour your Tableau Site* on page 2557.

**Watch a video:** To see related concepts demonstrated in Tableau, watch *Getting Started*, a 25-minute free training video. Use your tableau.com account to sign in.

**Start Page**

The start page in Tableau Desktop is a central location from which you can do the following:

- Connect to your data
- Open your most recently used workbooks, and
- Discover and explore content produced by the Tableau community.

The start page consists of three panes: **Connect, Open**, and **Discover**.
Connect

Connect to data and open saved data sources.
On the **Connect** pane, you can do the following:

- **Connect to data**: Under **To a File**, connect to data stored in Microsoft Excel files, text files, Access files, Tableau extract files, and statistical files, such as SAS, SPSS, and R. Under **To a Server**, connect to data stored in databases like Microsoft SQL Server or Oracle. The server names listed in this section change based on which servers you connect to and how often.

- **Open saved data sources**: Quickly open data sources that you have previously saved to your My Tableau Repository directory. Also, Tableau provides sample saved data sources that you can use to explore Tableau Desktop functionality. To follow along with examples in the Tableau Desktop documentation, you'll usually use the **Sample – Superstore** data source.

Open recent workbooks, pin workbooks to the start page, and explore sample workbooks.
On the Open pane, you can do the following:

- **Open recently opened workbooks**: When you open Tableau Desktop for the first time, this pane is empty. As you create and save new workbooks, the most recently opened workbooks appear here. Click the workbook thumbnail to open a workbook, or if you don’t see a workbook thumbnail, click the Open a Workbook link to find other workbooks that are saved to your computer.

- **Pin workbooks**: You can pin workbooks to the start page by clicking the pin icon that appears in the top-left corner of the workbook thumbnail. Pinned workbooks always appear on the start page, even if they weren’t opened recently. To remove a recently opened or pinned workbook, hover over the workbook thumbnail, and then click the “x” that appears. The workbook thumbnail is removed immediately but will show again with
your most recently used workbooks the next time you open Tableau Desktop.

- **Explore sample workbooks**: Open and explore sample workbooks.

Discover

See popular views in Tableau Public, read blog posts and news about Tableau, and find training videos and tutorials to help you get started.
Data Source Page

Before or during your analysis, you may want to make changes to the Tableau data source. You can do that on the data source page. Tableau takes you to the data source page after you
establish the initial connection to your data. You can also access the data source page by clicking the **Data Source** tab from any location in the workbook.

**Watch a video:** To see related concepts on connecting to data and using the Data pane, watch *Getting Started with Data*, a 6-minute free training video. Use your tableau.com account to sign in.

In this article

- **Left pane**
- **Canvas**
- **Data grid**
- **Metadata grid**

Although the look of the page and the options available vary depending on the type of data that you are connected to, the data source page generally consists of four main areas: left pane, canvas, data grid, and metadata grid.
For a 6-minute walkthrough of the Tableau workspace, watch "Getting Started with Data", a free Connecting to Data training video. Use your tableau.com account to sign in. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Left pane

The left pane of the data source page displays details about your data, that is the data that Tableau Desktop is connected to. For file-based data, the left pane might display the file name and the worksheets in the file. For relational-based data, the left pane might display the server, the database or schema, and the tables in the database.

You can also use the left pane to add more connections to the data source to create cross-database joins.

The left pane does not display for cube (multidimensional) data.

Canvas

When connected to most relational and file-based data, you can drag one or more tables to the canvas area to set up your Tableau data source.

When connected to cube data, the top of the data source page displays the available catalog or queries and cubes to select from to set up your Tableau data source.

Data grid

Use the data grid to review the fields and the first 1,000 rows of the data contained in the Tableau data source. You can also use the data grid to make general modifications to the Tableau data source like sorting or hiding fields; renaming fields or resetting field names; creating calculations; changing the column or row sort; or adding aliases. For the Web Data Connector, file- and relational-based data sources in Extract mode, you can see extract data in the grid, including extract only calculations.

To select multiple fields in the grid, click a column, and then drag the mouse to select more columns. To select all fields, click the area in the upper-left corner of the grid like in the example, below.
The grid does not display for cube (multidimensional) data.

Metadata grid

Depending on the type of data that you are connected to, click the metadata grid button to navigate to the metadata grid. The metadata grid displays the fields in your data source as rows so that you can quickly examine the structure of your Tableau data source and perform routine management tasks, such as renaming fields or hiding multiple fields at once.

When connected to cube or some extract only data, the metadata grid displays by default.

The Tableau Workspace

The Tableau workspace consists of menus, a toolbar, the Data pane, cards and shelves, and one or more sheets. Sheets can be worksheets, dashboards, or stories. For details on dashboard or story workspaces, see Create a Dashboard on page 2248 or The Story Workspace on page 2318.
**Watch a video:** To see related concepts demonstrated in Tableau, watch The Tableau interface in Getting Started, a 4-minute free training video. Use your tableau.com account to sign in.

If you are using Tableau on the web, see **Creators: Get Started with Web Authoring** on page 2566 and **Tour your Tableau Site** on page 2557.

In this article

**Workspace area below**

**Tableau Toolbar Button Reference** on the next page

**Show and Hide the Side Bar (Data pane)** on page 150

**Status Bar Information** on page 151

Workspace area

A. Workbook name. A workbook contains sheets. A sheet can be a worksheet, a dashboard, or a story. For more information, see **Workbooks and Sheets** on page 226.

B. **Cards and shelves** - Drag fields to the cards and shelves in the workspace to add data to your view.

C. **Toolbar** - Use the toolbar to access commands and analysis and navigation tools.
D. **View** - This is the workspace where you create your data visualizations.

E. **Goes to the start page. For more information, see Start Page on page 134.**

F. **Side Bar** - In a worksheet, the side bar area contains the **Data pane** and the **Analytics pane.**

G. **Go to the data source page. For more information, see Data Source Page on page 139.**

H. **Status bar** - Displays information about the current view.

I. **Sheet tabs** - Tabs represent each sheet in your workbook. This can include worksheets, dashboards, and stories. For more information, see Workbooks and Sheets on page 226.

---

**Tableau Toolbar Button Reference**

When you are creating or editing a view, you can use the toolbar at the top of the view to perform common actions.

In Tableau Desktop, you can hide or display the Tableau toolbar by selecting **Window > Show Toolbar.**

The table below explains the functions of each toolbar button. Some buttons are not available in all Tableau products.

<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Tableau icon](image) | **Tableau icon**: Navigates to the start page. For more information, see Start Page on page 134.  
**Note**: Tableau Desktop only. |
<p>| ⬅️ | <strong>Undo</strong>: Reverses the most recent action in the workbook. You can undo an unlimited number of times, back to the last time you opened the workbook, even after you have saved. For more information, see Undo and Redo on page 2616. |
| ➡️ | <strong>Redo</strong>: Repeats the last action you reversed with the Undo button. You can redo an unlimited number of times. |</p>
<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Save Icon](image) | **Save:** In Tableau Desktop, saves the changes made to the workbook. For more information, see [Save Your Work on page 2460](#).  
In Tableau Server or Tableau Online, click **File > Save** or **File > Save As** to save your changes. |
| ![New Data Source Icon](image) | **New Data Source:** In Tableau Desktop, opens the **Connect** pane where you can create a new connection or open a saved connection. For more information, see [Connect to Your Data on page 319](#).  
In Tableau Server or Tableau Online, opens the **Connect to a Data Source** page, where you can connect to a published data source. For more information, see [Connect to published data sources while web editing on page 376](#). |
| ![Pause Auto Updates Icon](image) | **Pause Auto Updates:** Controls whether Tableau updates the view when changes are made. Use the drop-down menu to automatically update the entire sheet or just use filters. For more information, see [Refresh Data or Pause Automatic Updates on page 2649](#). |
| ![Run Update Icon](image) | **Run Update:** Runs a manual query of the data to update the view with changes when automatic updates are turned off. Use the drop-down menu to update the entire worksheet or just use filters.  
**Note:** Tableau Desktop only. |
| ![New Worksheet Icon](image) | **New Worksheet:** Creates a new blank worksheet, use the drop-down menu to create a new worksheet, dashboard, or story.  
For more information, see [Create new worksheets](#), |
<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>dashboards, or stories</strong> on page 227.</td>
</tr>
<tr>
<td><img src="image" alt="Duplicate" /></td>
<td><strong>Duplicate</strong>: Creates a new worksheet containing the same view as the current sheet. For more information, see <strong>Duplicate a sheet</strong> on page 229.</td>
</tr>
<tr>
<td><img src="image" alt="Clear" /></td>
<td><strong>Clear</strong>: Clears the current worksheet. Use the drop-down menu to clear specific parts of the view such as filters, formatting, sizing, and axis ranges.</td>
</tr>
<tr>
<td><img src="image" alt="Swap" /></td>
<td><strong>Swap</strong>: Moves the fields on the <strong>Rows</strong> shelf to the <strong>Columns</strong> shelf and vice versa. The <strong>Hide Empty Rows</strong> and <strong>Hide Empty Columns</strong> settings are always swapped with this button.</td>
</tr>
<tr>
<td><img src="image" alt="Sort Ascending" /></td>
<td><strong>Sort Ascending</strong>: Applies a sort in ascending order of a selected field based on the measures in the view. For more information, see <strong>Sort Data in a Visualization</strong> on page 1202.</td>
</tr>
<tr>
<td><img src="image" alt="Sort Descending" /></td>
<td><strong>Sort Descending</strong>: Applies a sort in descending order of a selected field based on the measures in the view. For more information, see <strong>Sort Data in a Visualization</strong> on page 1202.</td>
</tr>
<tr>
<td><img src="image" alt="Totals" /></td>
<td><strong>Totals</strong>: You can compute grand totals and subtotals for the data in a view. Select from the following options:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Show Column Grand Totals</strong>: Adds a row showing totals for all columns in the view.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Show Row Grand Totals</strong>: Adds a column showing totals for all rows in the view.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Row Totals to Left</strong>: Moves rows showing totals to the left of a crosstab or view.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Column Totals to Top</strong>: Moves columns showing</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>totals to the top of a crosstab or view.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Add All Subtotals</strong>: Inserts subtotal rows and columns in the view, if you have multiple dimensions in a column or row.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Remove All Subtotals</strong>: Removes subtotal rows or columns.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Tableau Server and Tableau Online only. In Tableau Desktop, click <strong>Analysis &gt; Totals</strong>. For more information, see <a href="#">Show Totals in a Visualization</a> on page 1754.</td>
</tr>
<tr>
<td><img src="highlight.png" alt="Highlight Button" /></td>
<td><strong>Highlight</strong>: Turn on highlighting for the selected sheet. Use the options on the drop-down menu to define how values are highlighted. For more information, see <a href="#">Highlight Toolbar Button</a> on page 1803.</td>
</tr>
<tr>
<td><img src="group_members.png" alt="Group Members Button" /></td>
<td><strong>Group Members</strong>: Creates a group by combining selected values. When multiple dimensions are selected, use the drop-down menu to specify whether to group on a specific dimension or across all dimensions. For more information, see <a href="#">Correct Data Errors or Combine Dimension Members by Grouping Your Data</a> on page 1002.</td>
</tr>
<tr>
<td><img src="mark_labels.png" alt="Show Mark Labels Button" /></td>
<td><strong>Show Mark Labels</strong>: Switches between showing and hiding mark labels for the current sheet. For more information, see <a href="#">Show and Hide Mark Labels</a> on page 2180.</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image" alt="Fix Axes" /></td>
<td><strong>Fix Axes</strong>: switches between a locked axis that only shows a specific range and a dynamic axis that adjusts the range based on the minimum and maximum values in the view. For more information, see <a href="#">Edit Axes on page 1838</a>. <strong>Note</strong>: Tableau Desktop only.</td>
</tr>
<tr>
<td><img src="image" alt="Format Workbook" /></td>
<td><strong>Format Workbook</strong>: Open the Format Workbook pane to change how fonts and titles look in every view in a workbook by specifying format settings at the workbook level instead of at the worksheet level. <strong>Note</strong>: Tableau Server and Tableau Online only. In Tableau Desktop, click <strong>Format &gt; Workbook</strong>. For more information, see <a href="#">Format at the Workbook Level on page 2356</a>.</td>
</tr>
<tr>
<td><img src="image" alt="Fit" /></td>
<td><strong>Fit</strong>: Specifies how the view should be sized within the window. Select Standard, Fit Width, Fit Height, or Entire View. <strong>Note</strong>: This menu is not available in geographic map views. The <strong>Cell Size</strong> commands have different effects depending on the type of visualization. To access the <strong>Cell Size</strong> menu in Tableau Desktop click <strong>Format &gt; Cell Size</strong>.</td>
</tr>
<tr>
<td><img src="image" alt="Show/Hide Cards" /></td>
<td><strong>Show/Hide Cards</strong>: Shows and hides specific cards in a worksheet. Select each card that you want to hide or show on the drop-down menu. In Tableau Server and Tableau Online, you can show and hide cards for the Title, Caption, Filter and Highlighter only.</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image" alt="Presentation Mode" /></td>
<td><strong>Presentation Mode</strong>: Switches between showing and hiding everything except the view (i.e., shelves, toolbar, Data pane). For more information, see <a href="#">Reorganizing the Workspace on page 238</a>.</td>
</tr>
</tbody>
</table>
| ![Download](image) | **Download**: Use the options under **Download** to capture parts of your view for use in other applications.  
- **Image**: Displays the view, dashboard, or story as an image in a new browser tab.  
- **Data**: Displays the data from the view in a new browser window with two tabs: **Summary**, showing aggregated data for the fields shown in the view, and **Underlying**, showing underlying data for the selected marks in the visualization. If the new window does not open, you may need to disable your browser’s popup blocker.  
- **Crosstab**: Saves the underlying data for the selected marks in the visualization to a CSV (comma-separated values) file which can then be opened in Microsoft Excel.  
- **PDF**: Opens the current view as a PDF in a new browser window. From there you can save it to a file. If the new window does not open, you may need to disable your browser’s pop-up blocker. |
| ![Share Workbook With Others](image) | **Share Workbook With Others**: Publish your workbook to Tableau Server or Tableau Online. For more information, see [Simple Steps to Share a Workbook on page 2497](#). |

*Note:* Tableau Desktop only.

*Note:* Tableau Server and Tableau Online only.
Show and Hide the Side Bar (Data pane)

The Side Bar contains the Data pane and the Analytics pane when you are editing a worksheet. Different panes are visible depending on what you are doing in the view (Data, Analytics, Story, Dashboard, Layout, Format). The most important thing to know about the Side Bar is that you can expand and collapse this area in the workspace.

To hide the side bar in Tableau Desktop, click the collapse arrow in the side bar.

To hide the side bar on the web, click the collapse arrow in the side bar.

To show the side bar on Tableau Desktop, click the expand arrow in the bottom-left of the workspace (in the status bar).
To show the side bar on the web, click the expand arrow in the side bar.

Status Bar Information

The status bar is located at the bottom of the Tableau workspace. It displays descriptions of menu items as well as information about the current view. For example, the status bar below
shows that the view has 143 marks shown in 3 rows and 12 columns. It also shows that the SUM(Sales) for all the marks in the view is $2,297,201.

You can hide the status bar by selecting Window > Show Status Bar.

Occasionally, Tableau will display warning icons in the bottom right corner of the status bar to indicate errors or warnings. Below are the possible warning icons and what they mean.

<table>
<thead>
<tr>
<th>Warning Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Warning Icon" /></td>
<td><strong>Cancel query indicator:</strong> When you cancel multiple queries, an indicator appears to show you how many queries are still running on the database and using resources.</td>
</tr>
<tr>
<td>Warning Icon</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>When you cancel a query in Tableau, the database is told to stop processing the query. However, some databases do not support cancel (MS Excel, MS Access, Essbase, Microsoft Analysis Services 2000). If you cancel a query using one of these types of data sources, the query is abandoned by Tableau but is still running in the background.</td>
</tr>
<tr>
<td>Warning Icon</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>and using resources. When you have abandoned queries, an indicator appears in the bottom right corner of the workbook showing the number of queries still running. As queries in the background complete, the number will go down. It is important to monitor the number of queries running and not let the number get too high, otherwise you will see performance issues.</td>
</tr>
<tr>
<td>Warning Icon</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>degradation of both Tableau and the underlying database.</td>
</tr>
</tbody>
</table>

Note:
- Text, Microsoft Excel, and Microsoft Access data sources may be temporarily unavailable after canceling a query because of a lock perfor...
<table>
<thead>
<tr>
<th>Warning Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Warning Icon" /></td>
<td>med internally. You may have to wait until the abandoned query has completed before reconnecting.</td>
</tr>
<tr>
<td><img src="image" alt="Precision Warning Icon" /></td>
<td><strong>Precision warning:</strong> Some fields are more precise in the database than Tableau can model. When you add a field to a view that</td>
</tr>
<tr>
<td>Warning Icon</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image" alt="Warning Icon" /></td>
<td>contains values with more precision than Tableau can model, a warning icon is displayed in the bottom right corner of the status bar. For example, a value in the database may have 22 decimal places but Tableau only supports up to 15. When you add that field to the view, you get a precision warning. If you click on the warning, you can read more details</td>
</tr>
<tr>
<td>Warning Icon</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>including the number of decimal places that have been truncated in the view.</td>
</tr>
<tr>
<td></td>
<td>Remember that the precision of the data displayed in Tableau will always first be dependent on the data in your database. If the values in your database exceed 15 decimal places, when you add them to the view, the value is truncated and a precision warning appears.</td>
</tr>
<tr>
<td>Warning Icon</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image" alt="1 unknown" /></td>
<td><strong>Special values indicator</strong>: If your data contains null values, unknown geographic locations, or negative or zero values on a logarithmic axis; the values are shown with an indicator in the lower right corner of the view. Click the indicator for options for handling these values. See <a href="#">Handling Null and Other Special Values</a> on page 2205 to learn more about</td>
</tr>
</tbody>
</table>
### Warning Icon | Description
--- | ---
 | this indicator and how to handle these values.

---

## Work with Data Fields in the Data Pane

Tableau displays data source connections and data fields for the workbook in the **Data** pane on the left side of the workspace.

**Note:** For details on how to start creating visualizations, see [Get Started](#) on page 118. For details on the many ways you can customize the fields in the Data pane, see [Organize and Customize Fields in the Data Pane](#) on page 962 and [Edit Default Settings for Fields](#) on page 973.

### In this article

- **Areas of the Data pane**
  - Columns become fields in the view
  - Fields that Tableau automatically creates: Measure Names and Measure Values, Number of Records, Longitude and Latitude
  - Perform common tasks in the Data pane
  - Change data fields to fit your needs
  - Relational data versus cube data

### Areas of the Data pane

After you connect to your data and set up the data source with Tableau, the data source connections and fields appear on the left side of the workbook in the Data pane. For details on connecting to data, see [Connect to and Prepare Data](#) on page 319.

Current data source connections appear at the top of the **Data** pane. When you have more than one connection available, click a connection to select it and start working with that data.
You build visualizations by adding fields from the Data pane to the view. For details, see Start Building a Visualization by Dragging Fields to the View on page 1087.

Below the data source connections in the Data pane are the fields that are available in the currently selected data source. You can toggle between the Data and Analytics panes in a worksheet. For details on the Analytics pane, see Apply Advanced Analysis to a View (Analytics Pane) on page 174.
The Data pane includes these areas:

- **Dimensions**
  - Customer
    - Customer Name
    - Segment
  - Order
    - Order Date
    - Order ID
    - Ship Date
    - Ship Mode
  - Location
  - Product
    - Profit (bin)
    - Region
    - Measure Names

- **Measures**
  - Discount
  - Profit
  - Profit Ratio
  - Quantity
  - Sales
    - Latitude (generated)
    - Longitude (generated)
    - Number of Records
    - Measure Values

- **Sets**
  - Top Customers by Profit

- **Parameters**
  - Profit Bin Size
  - Top Customers
- Dimensions – Fields that contain qualitative values (such as names, dates, or geographical data). You can use dimensions to categorize, segment, and reveal the details in your data. Dimensions affect the level of detail in the view. Examples of dimensions include dates, customer names, and customer segments.

- Measures – Fields that contain numeric, quantitative values can be measured. You can apply calculations to them and aggregate them. When you drag a measure into the view, Tableau applies an aggregation to that measure (by default). Examples of measures: sales, profit, number of employees, temperature, frequency.

- Sets – Subsets of data that you define. Sets are custom fields based on existing dimensions and criteria that you specify. For more information, see Create Sets on page 1004.

  Named sets from an MS Analysis Services server or from a Teradata OLAP connector also appear in Tableau in this area of the Data pane. You can interact with these named sets in the same way you interact with other custom sets in Tableau.

- Parameters – Values that can be used as placeholders in formulas, or replace constant values in calculated fields and filters. For more information, see Create Parameters on page 1031.

**Note:** For cube data sources, fields are explicitly defined as dimensions or measures when the database is created. For relational data sources, Tableau automatically organizes the fields. By default, fields containing text, date or boolean values are dimensions, while fields containing numerical values are measures.

By default the field names defined in the data source are displayed in the Data pane. You can rename fields and member names, create hierarchies, and organize the fields into groups and folders. For details, see Edit Default Settings for Fields on page 973, Organize and Customize Fields in the Data Pane on page 962, and Create Hierarchies on page 987.
For cube (multidimensional) data sources, the fields are determined by the dimensions and measures of a cube. In Tableau, cube data sources are supported only in Windows.

Here's an example of fields from an Excel worksheet.

When you start building a visualization in a worksheet, these columns are available as fields in the Data pane. For details, see Areas of the Data pane on page 160.

Each field has a data type (that you can change if needed), and a role: discrete dimension, continuous dimension, discrete measure, or continuous measure. For details, see Data Types on page 263 and Dimensions and Measures, Blue and Green on page 250.

Each field also includes some default settings, such as a default aggregation of SUM or AVG, depending on the structure of the current view. For details, see Edit Default Settings for Fields on page 973 and Data Aggregation in Tableau on page 279.

Fields that Tableau automatically creates

The Data pane can also contain a number of fields that do not come from your original data: Measure Names and Measure Values, Number of Records, Latitude and Longitude.
Measure Names and Measure Values

- The **Measure Values** field always appears at the bottom of the Measures area in the **Data** pane and contains all the measures in your data, collected into a single field with continuous values. Drag individual measure fields out of the Measure Values card to remove them from the view.

- The **Measure Names** field always appears at the bottom of the Dimensions area in the **Data** pane and contains the names of all measures in your data, collected into a single field with discrete values.

For more details about how to use Measure Values and Measure Names in visualizations, see **Measure Values and Measure Names** on page 2196. To see measure names and measure values in action, watch the 5-minute **Measure Names and Measure Values** training video. Use your **tableau.com** account to sign in. To view more training and introductory videos, go to **Free Training Videos** on the Tableau website.

Number of Records

The **Number of Records** field is an automatically generated, calculated field that is simply set to the number 1. That number gets associated with each row in the data source. If you add the Number of Records field to the view, you will see the summed count of all the rows in a data source (the number of records). You can use the Number of Records field to get quick counts of various dimensions’ values. Viewing the Number of Records can help you to see if you data joins are working the way you expect them to.

Latitude and Longitude (generated)

When Tableau interprets fields to be geographic fields that can be used with maps, it automatically geocodes the data and includes **Latitude (generated)** and **Longitude (generated)** fields. You can use these fields to overlay your data on live maps. For more information on how to use these fields, and best practices for building maps in Tableau, see **Geographic Data Analysis in Tableau** on page 1862, **Assign Geographic Roles** on page 1941, **Location Data that Tableau Supports for Building Map Views** on page 1905.
Perform common tasks in the Data pane

**To select a data source connection for analysis,** click the data source connection name in the Data pane. For more details, see *Navigating Data Sources in the Data Pane* on page 169.

![Data pane screenshot](image)

**To view a context menu for the data source,** click **Data** in the top menu and then click on the data source in the menu list. For more information, see *Edit Data Sources* on page 817.

**To search for fields in the Data pane,** click the magnifying class icon and then type in the text box.

**To see the underlying data,** click the **View Data** icon at the top of the Data pane. For more information, see *View Underlying Data* on page 2616.
In cases where Tableau has misclassified a field as a dimension or a measure, possibly because of the data type, you can convert it and change its role.

**To convert a measure to a dimension**, drag the measure and drop it into the **Dimensions** area in the **Data** pane. For more details, see [Convert a Measure to a Dimension](#) on page 981.

---

**Change data fields to fit your needs**

When you drag a field into the view, it will have certain default settings and characteristics. You can customize a field that is already in the view, just for that instance of the field. Or you can change its settings in the Data pane to make the field use those settings going forward.

You can control the definition of a field in the view, depending on how you want to work with that field data.

---

**Note:** To change the default settings for a field before you drag it into the view, right-click it (Control-click on a Mac). You can then edit its settings and default properties from its context menu.

For details on the many ways you can customize the fields in the Data pane, see [Organize and Customize Fields in the Data Pane](#) on page 962 and [Edit Default Settings for Fields](#) on page 973.

---

**Relational versus cube data**

The Data pane for a relational and cube data source are shown below. Note that the panes look essentially the same for both data sources in that the fields are organized into dimensions and measures. However, the cube data source contains hierarchies for dimensions. For example, notice that the Employee dimension in the cube Data pane contains hierarchical members such as Manager Name and Employee Dept.

Relational data sources don’t have built-in hierarchies. However, relational data sources often have related dimensions that have an inherent hierarchy. For example, a data source may have fields for Country, State, and City. These fields could be grouped into a hierarchy called Location. You can assemble relational hierarchies by dragging and dropping in the Data pane. For details, see [Create Hierarchies](#) on page 987.
Note: In Tableau, cube (multidimensional) data sources are supported only in Windows.

Data pane with relational data (left image) versus cube data (right image)

You can expand or collapse hierarchies in both relational and cube Data panes by clicking the arrow. You can hide the Data pane all together by clicking the minimize button ☰ in the upper-right corner of the Data pane.

For information on cube data sources, see Cube Data Sources on page 2216.
Navigating Data Sources in the Data Pane

The top of the Data pane lists all of the data sources in a given workbook. Click the data source you want to use to select it. The Data pane updates to show the corresponding fields in that data source.

You can resize the data source list area in the Data pane to save space. When you resize to a limited vertical height, you can scroll down to a data source.

Each data source has an icon to indicate its type. For example, the icon can indicate whether the data source is relational, cube (multidimensional), or an extract. In Tableau, cube data sources are supported only in Windows.
You can right-click (control-click on a Mac) the data source to access the commands that are on the Data menu. For example, you can right-click (control-click on a Mac) a data source and rename, export, or close it.
For details on the many ways you can customize and work with the fields in the Data pane, see Organize and Customize Fields in the Data Pane on page 962, Edit Default Settings for Fields on page 973, and Work with Data Fields in the Data Pane on page 160.

Navigate Between the Start Page and Workspace

When Tableau Desktop is already open, you can navigate to the start page by clicking the Tableau icon in the upper-left corner of the Tableau Desktop workspace.

To go back to the Tableau workspace, click the Tableau icon in the start page.

Handling Null and Other Special Values

Some data requires special handling in Tableau. Specifically:
null values
unrecognized or ambiguous geographic locations
negative or zero values when working with a logarithmic scale
negative or zero values when working with treemaps

In this article

About special values
Null numbers and dates, and negative values on log axes
Unknown geographic locations
Zero and negative values in treemaps

About special values
When your data contains any of these special values, Tableau cannot plot them in the view. Instead, it displays an indicator in the lower right corner of the view. Click the indicator to see options for how to handle these values.
Null numbers and dates, and negative values on log axes

When you drag a measure or continuous date to the view, the values are shown along a continuous axis. If the field contains null values or if there are zeroes or negative values on a logarithmic axis, Tableau cannot plot them. Tableau displays an indicator in the lower right corner of the view. Click the indicator and choose from the following options:

- **Filter Data** - exclude the null values from the view using a filter. When you filter data, the null values are also excluded from any calculations used in the view.

- **Show Data at Default Position** - show the data at a default location on the axis. The null values will still be included in calculations. The default position depends on the data type. The table below defines the defaults.

<table>
<thead>
<tr>
<th>Type of Data</th>
<th>Default Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numbers</td>
<td>0</td>
</tr>
<tr>
<td>Dates</td>
<td>12/31/1899</td>
</tr>
<tr>
<td>Negative Values on a Log Axis</td>
<td>1</td>
</tr>
<tr>
<td>Unknown Geographic Location</td>
<td>(0, 0)</td>
</tr>
</tbody>
</table>

**Note:** See Logical Functions on page 1314 for functions, such as ISNULL() and IFNULL(), that you can use to handle null values with a calculated field.

Unknown geographic locations

When working with maps and geographic fields, unknown or ambiguous locations are identified by the indicator in the lower right corner of the view. Click the indicator and choose from the following options:

- **Edit Locations** - correct the locations by mapping your data to known locations.

- **Filter Data** - exclude the unknown locations from the view using a filter. The locations will not be included in calculations.

- **Show Data at Default Position** - show the values at the default position of (0, 0) on the
Zero and negative values in treemaps

When working with treemaps, any null or zero values display in the indicator in the lower right corner of the view. Click the indicator and choose from the following options:

- **Filter Data** - exclude the unknown locations from the view using a filter. The locations will not be included in calculations.

- **Use Absolute Values** - use the absolute value to determine the size of the corresponding area in the view. For example, both values of 5 and -5 are shown as the same size.

If you don’t know how to handle the values, you can choose to leave the special values indicator. Generally, you should continue to show the indicator so that you know there is data that is not being shown in the view. However, to hide the indicator, right-click (control-click on a Mac) it and select **Hide Indicator**.

Apply Advanced Analysis to a View (Analytics Pane)

Drag reference lines, box plots, trend lines forecasts, and other items into your view from the **Analytics** pane, which appears on the left side of the workspace. Toggle between the **Data** pane and the **Analytics** pane by clicking the tabs at the top of the side bar.
Tableau Desktop Analytics pane

In Tableau Desktop, options for adding Analytics objects to the view are available the Analytics pane or menu, or in context in the view. For example, reference lines and bands are available when you edit an axis, and trend lines and forecasts are available from the Analysis menu.

The Analytics pane provides drag-and-drop access for the various options.

On the web, most Analytics objects are available from the Analytics pane.

Watch a video: To see related concepts on visual analytics in demonstrated in Tableau, watch the Getting Started with Visual Analytics (6-minutes) and Getting Started with Calculations (3-minutes), free training videos. Use your tableau.com account to sign in.
In this article

Add analytics objects to the view
Delete an analytics object from the view
Edit an analytics object in the view

Analytics objects definitions

- **Constant Line** on page 179
- **Average Line** on page 180
- **Median with Quartiles** on page 180
- **Box Plot** on page 180
- **Totals** on page 181
- **Average with 95% CI** on page 182
- **Median with 95% CI** on page 182
- **Trend Line** on page 183
- **Forecast** on page 183
- **Custom Reference Line** on page 183
- **Custom Reference Band** on page 184
- **Custom Distribution Band** on page 184
- **Custom Box Plot** on page 184

Add an analytics object to the view

To add an item from the **Analytics** pane, drag it into the view. When you drag an item from the **Analytics** pane, Tableau shows the possible destinations for that item. The range of choices varies depending on the type of item and the current view.

In a simple case, the drop target area would offer these three options:
The terms **Table**, **Pane** and **Cell** define the scope for the item:

For a more complicated view—for example, if the view contained a line chart with multiple or dual axes—Tableau would show you a drop target area that looked like this:

If you dropped the item in one of the three larger boxes in the header at the top of the drop target area—for example, the Table box—a separate median with quartiles would be added for each axis:
But if you drop the item in any of the six lower boxes aligned with a specific measure, the median with quartiles would only be added on the corresponding axis, with the specified scope.

Delete an analytics object from the view

You can delete an analytics object from the view by clicking Undo, or drag the object off the view to delete it.

You can also click on an item and choose Remove from the tooltip.

**Note:** Some Analytics pane items (Median with Quartiles and Average with 95% CI) add both a reference line and a reference distribution. Unless you are using Undo, you would need to delete these items separately.

Edit an analytics object in the view

To edit an item you have added from the Analytics pane, click on the item and select Edit from the tooltip. For additional editing options, see the section for a particular item type under Analytics object definitions, below.
Analytics object definitions

The following items can be dragged from the Analytics pane and dropped in the view. If an analytics object cannot be applied to the current configuration of fields in the view, it isn’t available.

Constant Line

Adds one or more constant lines to the view. You can add a constant line for a specific measure, for all measures, or for date dimensions. When you add a constant line, Tableau displays a Value prompt where you specify the value for the constant:

In Tableau Desktop, the Value prompt for a date value is a calendar control:

You can click on a resulting constant line and choose Edit or Remove. In Tableau Desktop there is a third option: Format. Choosing Edit opens the Edit Reference Line dialog box. For details, see Edit Existing Reference Lines, Bands, and Distributions on page 1660 in the Reference Lines, Bands, Distributions, and Boxes article. Another way to edit a line in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose Edit Reference Line.
Average Line

Adds one or more average lines to the view. You can add an average line for a specific measure or for all measures.

You can click on a resulting average line and choose a different aggregation, such as Total or Sum. You can also choose Edit or Remove. In Tableau Desktop there is a third option: Format. Choosing Edit opens the Edit Reference Line dialog box. For details, see Edit Existing Reference Lines, Bands, and Distributions on page 1660 in the Reference Lines, Bands, Distributions, and Boxes article. Another way to edit a line in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose Edit Reference Line.

---

Median with Quartiles

Adds one or more sets of median lines and distribution bands to the view. You can add a median with quartiles for a specific measure or for all measures.

The distribution bands are computed as quartiles; the middle two quartiles are shaded.

You can click on a resulting median line or distribution and choose Edit or Remove. In Tableau Desktop there is a third option: Format. Median lines and distributions must be edited, formatted, or removed separately. Choosing Edit opens the Edit Reference Line dialog box. You must click on the outer edge of a distribution band to see the options--clicking in the middle of the band has no effect. For details, see Edit Existing Reference Lines, Bands, and Distributions on page 1660 in the Reference Lines, Bands, Distributions, and Boxes article. Another way to edit a line or distribution in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose Edit Reference Line. A submenu will offer you two choices: Quartiles and Median.

For information on distribution types, including quartiles, see Add Reference Distributions on page 1651 in the Reference Lines, Bands, Distributions, and Boxes article.

---

Box Plot

Adds one or more box plots to the view. You can add box plots for a specific measure or for all measures. The scope for a box plot is always Cell (and never Table or Pane).
Click or hover over any of the horizontal lines in the box plot to see statistical information about the whiskers, quartiles, and median.

You can also choose Edit or Remove when you click on a line. In Tableau Desktop there is a third option: Format. Choosing Edit opens the Edit Reference Line dialog box. Another way to edit a box plot in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose Edit Reference Line.

Note: In Tableau Desktop, there are two items named Box Plot in the Analytics pane. For the Box Plot option in the Summarize section, Tableau will automatically add a box plot for the specified target. For the Box Plot option in the Custom section, Tableau will open the Edit Reference Line, Band, or Box dialog box after you specify a target.

### Totals

Adds totals to the view. When you add totals, the drop options are Subtotals, Column Grand Totals, and Row Grand Totals.

For details, see Show Totals in a Visualization on page 1754.

To remove totals, click the relevant column or row header and choose Remove.

In Tableau Desktop, you can also click on a totals column or row header after adding totals and set the aggregation for that row or column from the tooltip:
Average with 95% CI

Adds one or more sets of average lines with distribution bands; the distribution bands are configured at a 95% confidence interval. You can add these items for a specific measure or for all measures.

The confidence interval distribution bands shade the region in which the population average will fall 95% of the time.

You can click on a resulting average line or distribution and choose Edit or Remove. In Tableau Desktop there is a third option: Format. Choosing Edit opens the Edit Reference Line dialog box. The average lines and distributions must be edited, formatted, or removed separately. You must click on the outer edge of a distribution band to see the options--clicking in the middle of the band has no effect. Another way to edit a line or distribution in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose Edit Reference Line. A submenu will offer you two choices: Average and 95% Confidence Interval.

You can also remove lines and bands by dragging them off the view.

--

Median with 95% CI

Adds one or more sets of median lines with distribution bands; the distribution bands are configured at a 95% confidence interval. You can add these items for a specific measure or for all measures.

The confidence interval distribution bands shade the region in which the population median will fall 95% of the time.

You can click on a resulting median line or distribution and choose Edit, Format, or Remove. In Tableau Desktop there is a third option: Format. Choosing Edit opens the Edit Reference Line dialog box. The median lines and distributions must be edited, formatted, or removed separately. You must click on the outer edge of a distribution band to see the options--clicking in the middle of the band has no effect. Another way to edit a line or distribution in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose Edit Reference Line. A submenu will offer you two choices: Median and 95% Confidence Interval.

You can also remove lines and bands by dragging them off the view.

--
Trend Line

Adds one or more trend lines to the view. When you add trend lines, the drop options identify the trend line model types available in Tableau: Linear, Logarithmic, Exponential, and Polynomial. For some views, only a subset of these options is available.

For details, see Trend Line Model Types on page 1669

Click on a trend line to remove or edit it, or to see a statistical definition. You can also remove a trend line by dragging it off the view.

Forecast

Adds a forecast to the view. This option is only available in Tableau Desktop—not when you edit a view on the web. Forecasting is only possible when there is at least one measure in the view.

Forecasting is not supported for views based on multidimensional data sources. In addition, the view cannot contain any of the following:

- Table calculations
- Disaggregated measures
- Percent calculations
- Grand Totals or Subtotals
- Date values with aggregation set to Exact Date

A time series containing null values also imposes constraints.

For details, see Forecasting on page 1705.

To remove, edit, or read a description of the current forecast, go to the Analysis menu and choose Forecast.

Custom Reference Line

You can add reference lines for a specific measure or for all measures in the view.

After you drag a reference line from the Analytics pane and drop it on a target, Tableau automatically opens an edit dialog box. See Add a Reference Line on page 1639 in the
Reference Lines, Bands, Distributions, and Boxes article for information on the available options. To return to this dialog box later, click on the line and choose Edit.

Custom Reference Band

You can add reference bands for a specific measure or for all measures in the view.

After you drag a reference band from the Analytics pane and drop it on a target, Tableau automatically opens the Edit Reference Line, Band, or Box dialog box. See Add Reference Bands on page 1646 in the Reference Lines, Bands, Distributions, and Boxes article for information on the available options. To return to this dialog box later, click on the band and choose Edit. You must click on the outer edge of a reference band to see the options—clicking in the middle of the band has no effect.

Custom Distribution Band

You can add reference distributions for a specific measure or for all measures in the view.

After you drag a reference distribution from the Analytics pane and drop it on a target, Tableau automatically opens Edit Reference Line, Band, or Box dialog box. See Add Reference Distributions on page 1651 in the Reference Lines, Bands, Distributions, and Boxes article for information on the available options. To return to this dialog box later, click on the band and choose Edit. You must click on the outer edge of a distribution band to see the options—clicking in the middle of the band has no effect.

Custom Box Plot

In Tableau Desktop—but not when you edit a view on the web—you can drag a box plot from the Custom section of the Analytics pane and drop it on a target. (But keep in mind that on the web, you can add a box plot from the Summarize section on the Analytics pane.) When you drag Box Plot from the Custom section, Tableau automatically opens Edit Reference Line, Band, or Box dialog box. See Add a Box Plot on page 1656 in the Reference Lines, Bands, Distributions, and Boxes article for information on the available options. The scope for a box plot is always Cell (and never Table or Pane).
Click any of the horizontal lines in the box plot to see statistical information about the whiskers, quartiles, and median.

To return to this dialog box later, click on the band and choose **Edit**.

---

**Shelves and Cards Reference**

Every worksheet in Tableau contains shelves and cards, such as Columns, Rows, Marks, Filters, Pages, Legends, and more.

By placing fields on shelves or cards, you:

- Build the structure of your visualization.
- Increase the level of detail and control the number of marks in the view by including or excluding data.
- Add context to the visualization by encoding marks with color, size, shape, text, and detail.

Experiment with placing fields on different shelves and cards to find the optimal way to look at your data.

**In this article**

- Options for starting a view
  - Columns and Rows shelves
  - Hide rows and columns
  - Marks card
  - Filters shelf
  - Pages shelf
  - Additional shelves, legends, cards, and controls

**Options for starting a view**

If you aren’t sure where to place a field, you can get let Tableau help you determine the best way to display the data.

- You can drag fields from the **Data** pane and drop them onto the cards and shelves that are part of every Tableau worksheet.
- You can double-click one or more fields in the **Data** pane.
- You can select one or more fields in the **Data** pane and then choose a chart type from **Show Me**, which identifies the chart types that are appropriate for the fields you selected. For details, see **Use Show Me to Start a View** on page 2170.
- You can drop a field on the **Drop field here** grid, to start creating a view from a tabular perspective.

![Drop field here grid](image)

---

**Columns and Rows shelves**

Drag fields from the Data pane to create the structure for your visualizations.

The **Columns** shelf creates the columns of a table, while the **Rows** shelf creates the rows of a table. You can place any number of fields on these shelves.

When you place a dimension on the **Rows** or **Columns** shelves, headers for the members of that dimension are created. When you place a measure on the **Rows** or **Columns** shelf, quantitative axes for that measure are created. As you add more fields to the view, additional headers and axes are included in the table and you get an increasingly detailed picture of your data.

In the view shown below, the members of the **Segment** dimension are displayed as column headers, while the **Profit** measure is displayed as a vertical axis.
Tableau displays data using marks, where every mark corresponds to a row (or a group of rows) in your data source. The inner fields on the Rows and Columns shelves determine the default mark type. For example, if the inner fields are a measure and a dimension, the default mark type is a bar. You can manually select a different mark type using the Marks card drop-down menu. For more information, see Change the Type of Marks in the View on page 1101.

Adding more fields to the Rows and Columns shelves adds more rows, columns, and panes to the table.
Hide rows and columns

Generally you will add dimensions and measures to create the rows and columns of the table and you'll either include all data or add filters to only show a subset. However, when you filter data it is also excluded from calculations and other computations performed on the summarized data in the table. Instead of filtering the data, you can hide the row or column so it doesn't display in the view but it is still included in calculations.

Hiding columns is especially useful when using table calculations that compare to previous or next. In that case, there is always a row or column that doesn’t show data because there is no data to compare to. You can simply hide the empty column without modifying the table calculation.

For example, when calculating year-over-year growth, the first year doesn't have a previous year to compare to, so the column is left blank. Filtering the first year will remove it from the view but it will also remove it from the calculation so now the second year doesn't have a previous year to compare to and is left blank. Instead of filtering, you can hide the column that you don't want to show without changing the calculation.

**To hide a row or column:**

Right-click (control-click on Mac) the row or column you want to hide, and then select **Hide**.
To show hidden data:

Open the field menu for a field that has hidden columns or rows and select **Show Hidden Data**.
Marks card

The Marks card is a key element for visual analysis in Tableau. As you drag fields to different properties in the Marks card, you add context and detail to the marks in the view. You use the Marks card to set the mark type (see Change the Type of Marks in the View on page 1101), and to encode your data with color, size, shape, text, and detail.
In this example, three different fields have been dragged to different properties in the Marks card. Segment is on Color, Region is on Shape, and Quantity is on Size.

After you add a field to the Marks card, you can click the icon next to the field to change the property it is using. You can also click the property buttons in the Marks card to change those settings.
Many properties can have multiple fields. For example, you can add multiple fields to Label, Detail, Tooltip, and Color. Size and Shape can only have one field at a time. For more details, see Control the Appearance of Marks in the View on page 1123.

**Note:** By default, dragging a new field to **Color** replaces the existing fields. To add a new field to color without replacing the existing field, hold the SHIFT key on your keyboard while dragging a new field to **Color** on the Marks card.

---

**Filters shelf**

The Filters shelf allows you to specify which data to include and exclude. For example, you might want to analyze the profit for each customer segment, but only for certain shipping containers and delivery times. By placing fields on the Filters shelf, you can create such a view.

**Note:** This section presents a brief overview of filtering. For more information on filtering, see Filter Data from Your Views on page 1162.

You can filter data using measures, dimensions, or both at the same time. Additionally, you can filter data based on the fields that make up the columns and rows of the table. This is called an internal filter. You can also filter data using fields that don’t contribute headers or axes to the table. This is called an external filter. All filtered fields display on the Filters shelf.

To illustrate the basic concepts of filtering, consider the following view.
Suppose you are not interested in the Home Office data. You can remove this column from the view by filtering the Segment dimension. To do so, select Filter on the field menu or drag the Segment dimension to the Filters shelf. The Filter dialog box opens. By default all members are selected. Clear the check box for Home Office to exclude it from the view. All selected members will be included.
As shown below, the view updates and the Home Office column is removed. The filter is indicated by the **Segment** field on the **Filters** shelf.
Suppose you want to only view profit for a category of the products. Even though the **Category** field is not used on the **Rows** and **Columns** shelves or on the Marks card, you can still add a filter. Drag the **Category** dimension to the **Filters** shelf. This is an example of an external filter because Category is not part of the view.

The Filter dialog box automatically opens. By default, none of the members are selected. Select the members you want to keep as part of the view. All cleared members are excluded. In this example, Office Supplies is selected.

The modified data view is shown below. The mark label shows that the sum of the profit for the Consumer segment has decreased to $56,330. This number is derived by summing all the rows in the data source that are associated with the Corporate market and are part of the Office Supplies category.
The order of fields placed on the Filters shelf does not affect the data view because the filters are independent. In other words, the result of filtering by customer segment, and then by container is the same as filtering by container and then by customer segment. For more information about the filtering order of operations, see Filter Data from Your Views on page 1162.

Pages shelf

The Pages shelf lets you break a view into a series of pages so you can better analyze how a specific field affects the rest of the data in a view. When you place a dimension on the Pages
shelf you are adding a new row for each member in the dimension. When you place a measure on the Pages shelf, Tableau automatically converts the measure into a discrete measure.

The Pages shelf creates a set of pages, with a different view on each page. Each view is based on a member of the field you placed on the Pages shelf. You can easily flip through the views and compare them on a common axis, using the controls that get added to the view when you move a field to the Pages shelf. For example, the view below shows the Profit vs. Sales by Region for each day throughout the month. The image below shows days 1, 2, 3, and 4. You would have to scroll down to see other days in the month.

![Tableau view with Pages shelf](image)

To make this view more user-friendly, move DAY(Order Date) to the Pages shelf and use the associated control to flip through the pages (one for each day). You can quickly discover hidden insights. In this example, it is interesting that the 19th is an especially big day in terms of sales and profit in the Western region.
When you add a field to the **Pages** shelf, a page control is automatically added to the right of your view.

Use this control to navigate through the pages. There are three ways to navigate through the pages in a view:

- Jump to a specific page
- Manually advance through the pages
- Automatically advance through pages

**Jump to a specific page**

Select the member or value you want to view from the drop-down list to display a specific page.
Manually advance through the pages

You can manually advance through the sequence of pages by doing any of the following:

- Use the forward and back buttons on either side of the drop-down list to navigate through the pages one at a time.
- Use the Page Slider to quickly scroll forward and backward in the sequence of pages.
- Use any of the following keyboard shortcuts to scroll forward and backward in the sequence of pages.

<table>
<thead>
<tr>
<th>Keyboard Shortcut</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F4</td>
<td>Starts and stops forward playback</td>
</tr>
<tr>
<td>SHIFT + F4</td>
<td>Starts and stops backward playback</td>
</tr>
<tr>
<td>CTRL + .</td>
<td>Skip forward one page</td>
</tr>
<tr>
<td>CTRL + ,</td>
<td>Skip backward one page</td>
</tr>
</tbody>
</table>

On a Mac, these are the equivalent keyboard shortcuts.
### Play Controls

<table>
<thead>
<tr>
<th>Key Combination</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F4</td>
<td>Starts and stops forward playback</td>
</tr>
<tr>
<td>Shift-F4</td>
<td>Starts and stops backward playback</td>
</tr>
<tr>
<td>Command-period</td>
<td>Skip forward one page</td>
</tr>
<tr>
<td>Command-comma</td>
<td>Skip backward one page</td>
</tr>
</tbody>
</table>

#### Automatically advance through pages (Tableau Desktop only)

Use the playback controls to watch a slide show of the pages in the view. You can play forward or backward, and stop the playback at any time. You can control the speed of playback with the speed controls in the bottom right corner of the control. The smallest bar indicates the slowest playback speed.

![Playback Controls](image)

#### Show the Page history

Show page history using the Show History check box. With page history, marks from previous pages are shown on the current page.

*In Tableau Desktop only:* Open the drop-down control for history to specify what marks to show and when to show them.
The history drop-down control has the following options:

- **Marks to show history for** – select whether you want to show history for just selected marks, highlighted marks, marks that you’ve manually selected to show history for, or all marks. To manually show history for marks, right-click (control-click on Mac) the mark in the view and select an option on the Page History menu.

- **Length** – select the number of pages to show in the history.

- **Show** – specify whether to show the historical marks, a line tracing through the previous values (trails), or both.

- **Marks** – format the historical marks including the color and how much to fade them if the color is set to automatic, the marks will either use the default mark color or the color encoding on the Color shelf.

- **Trails** – format the lines that are drawn through the historical marks. This option is only available if Trails is selected in the Show options.

Page trails may not display if there are multiple marks per color on a page. Make sure that the level of detail for the view is less than or equal to the level of detail on the **Pages** shelf and on
the Color target. Also, trails are only supported for discrete mark types such as squares, circles, or shapes. They are not supported when the mark type is Automatic.

Pages on dashboards (Tableau Desktop only)

When a dashboard contains multiple views that use the same field on the Pages shelf, you can control all of the views with a single page control by selecting the Synchronized option. This option is only available on the page control shown on a dashboard.

Additional Shelves, Legends, Cards, and Controls

Some shelves, legends, cards, or controls are only displayed as a result of things that you do as you work with views. For example, the Color legend is only displayed when there is a field on Color.
Tableau provides controls for moving or otherwise customizing these elements of the view.

The following list describes each such shelf, legend, card, or control.

- **Measure Values Shelf** – Measure Values is a special field that always appears at the bottom of the Measures area of the Data pane and contains all the measures of your data collected into one field. Tableau automatically adds Measure Values to the view when multiple measures are sharing the same axis. When Measure Values is in the view, Tableau displays a Measure Values shelf that shows which measures are being included. You can add measures to or remove measures from this card. See Measure Values and Measure Names on page 2196.

- **Color Legend** – Shows how colors are allocated when there is a field on Color.

- **Shape Legend** – Shows how shapes are allocated when there is a field on Shape.

- **Size Legend** – Shows how sizes are allocated when there is a field on Size.

- **Map Legend** – Shows the legend for the symbols and patterns on a map. The map legend is not available for all map providers.

- **Parameter Controls** – A separate parameter control is available for every parameter in the workbook. For more information, see Create Parameters on page 1031.

- **Title** – A title is displayed by default for every view. The default title is the sheet name. Double-click a title (Control-click on a Mac) to edit it.

- **Caption** – Choose Show caption from the Worksheet menu to display a caption for the view.

- **Summary Card** – Choose Show summary from the Worksheet menu to display a summary card for the view. For more information, see Summary Card on page 1747.

- **Page Control** – Provides options for navigating through pages when there is a field on the Pages shelf.
Parts of the View

This section describes the basic elements of views that you can create in Tableau. You can show or hide parts of the view as needed (described below). Every view has a table in some form, which may include rows, columns, headers, axes, panes, cells, and marks. Views can optionally include tooltips, titles, captions, field labels, and legends.

In this article

<table>
<thead>
<tr>
<th>The View area (image with callouts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers</td>
</tr>
<tr>
<td>Axes</td>
</tr>
<tr>
<td>Panes</td>
</tr>
<tr>
<td>Cells</td>
</tr>
<tr>
<td>Marks</td>
</tr>
<tr>
<td>Tooltips</td>
</tr>
<tr>
<td>Titles</td>
</tr>
<tr>
<td>Captions</td>
</tr>
<tr>
<td>Field Labels</td>
</tr>
<tr>
<td>Legends</td>
</tr>
<tr>
<td>Legends Per Measure</td>
</tr>
</tbody>
</table>

The View area

Data views are displayed in a table on every worksheet. A table is a collection of rows and columns, and consists of the following components: Headers, Axes, Panes, Cells, and Marks. In addition to these, you can choose to show or hide Titles, Captions, Field Labels, and Legends.
A. **Field Labels** on page 219 - The label of a discrete field added to the row or column shelf that describes the members of that field. For example, Category is a discrete field that contains three members; Furniture, Office Supplies and Technology.

B. **Titles** on page 216 - The name that you give your worksheet, dashboard, or story. Titles display automatically for worksheets and stories and you can turn them on to display them in your dashboards.

C. **Marks** on page 213 - The data that represents the intersection of the fields (dimensions and measures) included in your view. Marks can be represented using lines, bars, shapes, maps and so on.

D. **Legends** on page 221 - A key that describes how the data is encoded in your view. For example if you use shapes or colors in your view, the legend describes what each shape or color represents.

E. **Axes** on page 209 - Created when you add a measure (fields that contain quantitative, numerical information) to the view. By default, Tableau generates a continuous axis for this data.

F. **Headers** on the next page - The member name of a field.

G. **Captions** on page 218 - Text that describes the data in the view. Captions can be automatically generated and can be toggled on and off.

Also see information on **Cells** on page 212 and **Panes** on page 212.
**Headers**

Headers are created when you place a dimension or discrete field on the **Rows** shelf or the **Columns** shelves. The headers show the member names of each field on the shelves. For example, in the view below the column headers show the members of the **Order Date** field and the row headers show the members of the **Sub-Category** field.

![Table](image)

You can show and hide row and column headers at anytime.

**To hide headers:**

---

**Sheet 1**

<table>
<thead>
<tr>
<th>Sub-Category</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
<th>2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories</td>
<td>$25,014</td>
<td>$40,524</td>
<td>$41,896</td>
<td>$59,946</td>
</tr>
<tr>
<td>Appliances</td>
<td>$15,314</td>
<td>$23,241</td>
<td>$26,050</td>
<td>$42,927</td>
</tr>
<tr>
<td>Art</td>
<td>$6,058</td>
<td>$6,237</td>
<td>$5,910</td>
<td>$8,914</td>
</tr>
<tr>
<td>Binders</td>
<td>$43,488</td>
<td>$37,453</td>
<td>$49,485</td>
<td>$72,986</td>
</tr>
<tr>
<td>Bookcases</td>
<td>$20,037</td>
<td>$38,544</td>
<td>$26,275</td>
<td>$30,024</td>
</tr>
<tr>
<td>Chairs</td>
<td>$77,242</td>
<td>$71,735</td>
<td>$83,919</td>
<td>$95,554</td>
</tr>
<tr>
<td>Copiers</td>
<td>$10,850</td>
<td>$26,179</td>
<td>$49,599</td>
<td>$62,899</td>
</tr>
<tr>
<td>Envelopes</td>
<td>$3,856</td>
<td>$4,512</td>
<td>$4,730</td>
<td>$3,379</td>
</tr>
<tr>
<td>Fasteners</td>
<td>$661</td>
<td>$545</td>
<td>$960</td>
<td>$858</td>
</tr>
<tr>
<td>Furnishings</td>
<td>$13,826</td>
<td>$21,090</td>
<td>$27,874</td>
<td>$28,915</td>
</tr>
<tr>
<td>Labels</td>
<td>$2,841</td>
<td>$2,956</td>
<td>$2,827</td>
<td>$3,861</td>
</tr>
<tr>
<td>Machines</td>
<td>$62,023</td>
<td>$27,764</td>
<td>$55,907</td>
<td>$43,545</td>
</tr>
<tr>
<td>Paper</td>
<td>$14,835</td>
<td>$15,288</td>
<td>$20,638</td>
<td>$27,718</td>
</tr>
<tr>
<td>Phones</td>
<td>$77,391</td>
<td>$68,314</td>
<td>$78,660</td>
<td>$105,643</td>
</tr>
<tr>
<td>Storage</td>
<td>$50,329</td>
<td>$45,048</td>
<td>$58,632</td>
<td>$69,834</td>
</tr>
<tr>
<td>Supplies</td>
<td>$14,394</td>
<td>$1,952</td>
<td>$14,278</td>
<td>$16,049</td>
</tr>
<tr>
<td>Tables</td>
<td>$46,088</td>
<td>$39,150</td>
<td>$60,833</td>
<td>$60,894</td>
</tr>
</tbody>
</table>
Right-click (control-click on Mac) the headers in the view and select **Show Header**.

To show headers:

- Select the field in the view whose headers you want to show and select **Show Header** on the field menu.
Hiding headers can be useful when you are working with multiple measures. For example, the view below shows both the sales and profit for each region along a single axis. You can see the view looks cluttered with the Measure Names headers showing. Because Measure Names is also indicated by the mark color, you can hide the excess headers to clean up the view.
Axes

Axes are created when you place a measure or continuous field on the **Rows** or **Columns** shelves. By default, the values of the measure field are displayed along a continuous axis.
You can show and hide axes at any time.

**To hide axes:**

- Right-click (control-click on Mac) the axis in the view and select *Show Header* to clear the check mark next to this option.
To show axes:

- Right-click (control-click on Mac) the measure in the view whose axis you want to show and select **Show Header** on the field menu.
Panes

Panes are created by the intersection of rows and columns in a table. An example of a pane is highlighted in the view below.

Cells

Cells are the basic components of any table you can create in Tableau. For a text table, the cell is the intersection of a row and a column, and is where the text is displayed. For example, one of the 68 cells is highlighted in the view below.
Marks

When you drag fields to the view, the data are displayed using marks. Each mark represents the intersection of all of the dimensions in the view.

For example, in a view with Region and Year dimensions, there is a mark for every combination of those two dimensions (East 2011, East 2012, West 2011, West 2012, etc.). In this case, the mark type is set to Text, so the Abc represents the location where the value for the text mark will appear—once a measure such as Sales is added to the view.
Marks can be displayed in many different ways including lines, shapes, bars, maps, and so on. You can show additional information about the data using mark properties such as color, size, shape, labels, etc. The type of mark you use and the mark properties are controlled by the Marks card. Drag fields to the Marks card to show more data. For example, the same view above is shown again below but this time with **Profit** on Color. With this additional information, it is clear that the West region had the highest profit in 2014.

Control the marks in the view using the Marks card. Use the drop-down menu to specify the type of mark to show. Drag fields to the Marks card and use the drop-down controls to add more information to the view and control the color, shape, size, labels, and number of marks in the view.

**Tooltips**

Tooltips are additional data details that display when you hover over one or more marks in the view. When you select one or more marks and hover, tooltips also include options to filter marks (exclude or keep only), display marks that have the same values, create groups, create sets, or display the underlying data. If you don’t want users to be able to access tooltip commands, you can disable them.
For details on formatting tooltips and other tooltip settings, also see Format Titles, Captions, Tooltips, and Legends on page 2382, Add tooltips to marks on page 1133, and Create Views in Tooltips (Viz in Tooltip) on page 1773. For details on analysis options in tooltips, see Explore and Analyze Data in a View on page 2595.

Tooltip command buttons for exploring data in a viz

The top of the tooltip lists commands for filtering data, creating a group, sorting the selection, and view the underlying data. For example, you can use the tooltip to quickly remove an outlier in a scatter plot. Each of the commands are described below.

To see tooltip commands, hover over a mark and then keep the cursor still. The

- **Keep Only** - creates a filter that removes all other data. See Select to keep or exclude data points in your view on page 1163 to learn more.
- **Exclude** - creates a filter that removes the selected data. See Select to keep or exclude data points in your view on page 1163 to learn more.
- **Group Members** - creates a group based on the selection. If the selection contains multiple dimensions, you can group on one dimension or all dimensions. See Correct Data Errors or Combine Dimension Members by Grouping Your Data on page 1002 to learn more.
- **Create Set** (Tableau Desktop only) - creates a new set containing the selected members. You can create a new set or add members to an existing set. See Create Sets on page 1004 to learn more.
- **View Data** - opens a window displaying the data. You can view the summarized data or the underlying data. See View Underlying Data on page 2616 to learn more.

These commands are visible by default. You can disable the commands in the Edit Tooltip dialog box.
Disable tooltip commands

If you don't want users to be able to access tooltip commands, you can disable them.

1. Click Tooltip on the Marks card or select Worksheet > Tooltip.
2. In the Edit Tooltip dialog box, clear the Include command buttons check box.

Tooltip settings apply to the active worksheet and can be different for each sheet in the workbook.

Body text and markup in tooltips

The body of a tooltip contains details about a specific mark or a selection of multiple marks. For example, in a bar chart showing sales by region, the tooltip body may include the actual sales amount and the region name. The default tooltip is based on the fields used in the view. You can customize what is shown in the tooltip by dragging fields to Tooltip on the Marks card.

To customize a tooltip and its formatting, click Tooltip on the Marks card.

Alternatively, you can select Worksheet > Tooltips.

Action links

If the sheet has any actions, the action links are listed below the body of the tooltip. An action adds context and interactivity to your data through filters, highlighting, and links to external resources. See Actions on page 1789 to learn more about adding actions to your workbook.

Titles

You can show titles on any worksheet, dashboard, or story. For worksheets and stories, a title is displayed by default, but you can remove it. For dashboards, you can add a title. By default, the title is the name of the sheet, but you can edit the title to change the text and include dynamic values such as page number and sheet name. For more information about how to format titles, see Format Titles, Captions, Tooltips, and Legends on page 2382.

Show and hide titles in worksheets

Titles are shown by default for worksheets and are included as part of the worksheet, shown at the top of the view. You can move the title to the sides or the bottom of the view. However, when you move the title from the top of the view, it becomes a Title card and displays like any other card in the view.
Note: If you move a title from the top position and then hide it; when you show the title again, it appears back at the top of the worksheet in its default position.

To show or hide titles in a worksheet

- From the toolbar menu, click Worksheet > Show Title.
- On the toolbar, click the drop-down arrow on the Show/Hide Cards button and select Title from the context menu.

  Toggle the check mark on or off to show or hide the title.

Show and hide titles in dashboards

You can turn on titles for dashboards. The title appears as part of the dashboard.

To show or hide titles on a dashboard, from the toolbar menu, select Dashboard > Show Title.
When you add worksheets to the dashboard, the title of the worksheet automatically shows, even if you turned off the title on the worksheet itself. To turn off the title for the worksheet on the dashboard, do the following steps:

1. In the dashboard, select the worksheet to highlight it.
2. In the top right corner of the highlighted worksheet, click the drop-down arrow and select Title in the context menu to clear the check mark.

Show and hide titles in stories

Story titles are displayed by default. To toggle story titles on or off, in the top menu, select Story > Show Title to add or remove the check mark.

Captions

All views can have a caption that is either automatically generated or manually created. The caption is displayed on the Caption card.

To show a caption in a worksheet, select it on the Show/Hide Cards toolbar menu or select Worksheet > Show Caption.
The caption is automatically generated by default.

To edit the caption, double-click the Caption area in the view. In the Edit Caption dialog box, you can use change the font, size, color, and alignment and style.

Click the Insert menu to add automatic text such as page number, sheet name, and field and parameter values.

The caption can optionally be included when printing, printing to PDF, and publishing to Tableau Server. When you export the view as an image to another application like Microsoft PowerPoint, you can optionally include the caption.

Field Labels

Placing discrete fields on the rows and column shelves creates headers in the view that display the members of the field. For example, if you place a field containing products on the rows shelf, each product name is shown as row headers.
In addition to showing these headers, you can show field labels, which are labels for the headers. In this example, the rows are labeled **Category**, to indicate that the discrete category names are members of the **Category** field.

Field labels apply only to discrete fields. When you add continuous fields to the view, Tableau creates an axis. The axis is labeled with a header.

By default, field labels are shown.

To hide or show field labels, select **Analysis > Table Layout > Show Field Labels for Rows** or **Show Field Labels for Columns**.
You can format the fonts, alignment, shading, and separators for field labels.

Legends

When you add fields to Color, Size, and Shape on the Marks card, a legend displays to indicate how the view is encoded with relation to your data.

**Color Legend**

<table>
<thead>
<tr>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central</td>
</tr>
<tr>
<td>East</td>
</tr>
<tr>
<td>South</td>
</tr>
<tr>
<td>West</td>
</tr>
</tbody>
</table>

**Size Legend**
Not only do legends help you understand encodings, you can also use legends to sort, filter, and highlight specific sets of data. For more information, see Legend Highlighting on page 1797.

Measure values and color legends

If you include the Measure Values and Measure names fields in your views, you can create either a single combined color legend or separate color legends for your measures. If you drag the Measure Values field to Color on the Marks card, by default Tableau creates a single color legend that applies one color palette to all marks in the view. If you want to differentiate certain measures in the view you can create separate color legends for the measures and assign a unique color palette to each legend.

For more information about Measure Values and Measure Names, see Measure Values and Measure Names on page 2196.

The following example shows how to create separate color legends. This example uses the Sample Superstore data set.

1. Connect to the Sample-Superstore data set.

2. From the Data pane, drag Order Date to the Columns shelf and Category and Sub-
Category to the Rows shelf.

3. Under Dimensions in the Data pane, drag Measure Names to the columns shelf and drop it to the right of Order Date.

4. Under Measures in the Data pane, drag Measure Values to Color on the Marks card.

5. In the Measure Values card, drag measures off the card so that you keep SUM(Sales) and SUM(Profit) only.

6. Click Label on the Marks card and select Show mark labels to show the measure values in the view.

   When you drag Measure Values to Color on the Marks card, Tableau creates a single color legend and adds it to the view. Your view should look something like this.

7. To create separate legends for each measure on the view, click the drop-down arrow on the Measure Values field in the Marks card and select Use Separate Legends from the context menu.
Tableau creates an individual color legend for each measure in the view using the default color palette.

To assign a different color palette to the color legend, do one of the following:

- In Tableau Desktop, click the drop-down arrow in the top right corner for each color legend and select **Edit Colors**. Then select a color from the Palette drop-down list.
In Tableau Server or Tableau Online, click the drop-down arrow in the top right corner for each color legend. Then select a color from the Palette drop-down list.

Your view might look something like the following example:

8. To combine the separate legends back to a single legend, click the drop-down arrow on the Measure Values field on the Marks card and select Combine Legends from the context menu.
Workbooks and Sheets

Tableau uses a workbook and sheet file structure, much like Microsoft Excel. A workbook contains sheets. A sheet can be a worksheet, a dashboard, or a story.

- A **worksheet** contains a single view along with shelves, cards, legends, and the Data and Analytics panes in its side bar. For details on the worksheet workspace, see *The Tableau Workspace* on page 142.

- A **dashboard** is a collection of views from multiple worksheets. The Dashboard and Layout panes are available in its side bar. For more details about creating dashboards, see *Dashboards* on page 2239.

- A **story** contains a sequence of worksheets or dashboards that work together to convey information. The Story and Layout panes are available in its side bar. For more details about creating stories, see *Stories* on page 2318.

For details on organizing sheets, see *Navigate and Organize Sheets* on page 234. For details on creating and opening workbooks, see *Create or Open Workbooks* on page 231.

In this article

About sheets
Create a new sheet
Undo, redo, or clear sheets
Duplicate a sheet
Rename a sheet
See the underlying data in a sheet
Delete sheets

About sheets

Each workbook can contain different types of sheets: views (also known as worksheets), dashboards, and stories.

- A worksheet is where you build views of your data by dragging and dropping fields onto shelves.

- A dashboard is a combination of several views that you can arrange for presentation or
A story is a sequence of views or dashboards that work together to convey information.

The sheets display along the bottom of the workbook as tabs. In this section you’ll learn how to create, open, duplicate, hide, and delete sheets. You’ll also learn how to organize sheets in a workbook.

Within a workbook, you can create new sheets, clear an entire worksheet, duplicate sheets, hide or show a worksheet, and delete a sheet. Tableau has several ways to view and organize the sheets in your workbook.

Create new worksheets, dashboards, or stories

There are several ways to create new sheets in a workbook, dashboard, or a story. You can create as many sheets in a workbook as you want.

To create a new worksheet, dashboard, or story, click the New Worksheet, New Dashboard, or New Story button at the bottom of the workbook.

To rename a new worksheet, dashboard, or story, right-click (Ctrl-click on a Mac) the tab and then select the Rename command.

More ways to create new worksheets

Create a new worksheet by doing one of the following:

- Select Worksheet > New Worksheet.
- Right-click any open tab in the workbook, and select New Worksheet from the menu.
- On the toolbar, click the drop-down arrow on the New Worksheet button and then select New Worksheet.
- Press Ctrl + M on your keyboard (Command-M on a Mac).
More ways to create new dashboards

Create a new dashboard by doing one of the following:

- Select **Dashboard > New Dashboard**.
- Click the **New Dashboard** button at the bottom of the workbook.
- Right-click on any open tab in the workbook, and select **New Dashboard** from the menu.
- On the toolbar, click the drop-down arrow on the New Worksheet button and then select **New Dashboard**.

More ways to create new stories

Create a new story by doing one of the following:

- Select **Story > New Story**.
- Click the **New Story** button at the bottom of the workbook.
- Right-click on any open tab in the workbook, and select **New Story** from the menu.
- On the toolbar, click the drop-down arrow on the New Worksheet button and then select **New Story**.

---

Undo, redo, or clear sheets

Every Tableau workbook contains a history of steps you have performed on the worksheets, dashboards, and stories in that workbook for the current work session.
To move backward through the history, click **Undo** \[←\] on the toolbar or press Ctrl + Z on your keyboard (Command-Z on a Mac).

To move forward through the history, click **Redo** \[→\] on the toolbar or press Ctrl + Y (Command-Y on a Mac) on your keyboard.

To remove all fields, formatting, sizing, axis ranges, filters, sorts, and context filters in the sheet, click **Clear Sheet** \[🗑️\] on the toolbar.

To clear specific aspects of the view, use the Clear Sheet drop-down menu.

**Note:** Using the clear commands on the toolbar does not clear the history. If you decide that you didn’t want to clear the sheet, click the **Undo** button.

---

Duplicate a sheet

When you want to use an existing sheet as a jumping off point for more exploration, you can duplicate that sheet. The duplicated sheet contains all of the same fields and settings as a starting point for new analysis.

**Duplicate** creates a new version of a worksheet, dashboard, or story you can modify without effecting the original sheet.

To duplicate the active sheet, right-click the sheet tab (control-click on Mac) and select **Duplicate**.

**Note:** When you duplicate a dashboard, a new version of the dashboard is created, but it still references the original worksheets that were used to create the dashboard.

Duplicate as Crosstab

A crosstab (sometimes referred to as a Pivot Table) is a table that summarizes data in rows and columns of text. It is a convenient way to display the numbers associated with the data view.

To create a new cross-tab sheet based on the data in the current sheet, right-click the sheet tab (control-click on Mac) and select **Duplicate as Crosstab**. Or select **Worksheet > Duplicate as Crosstab**.
This command inserts a new worksheet into your workbook and populates the sheet with a cross-tab view of the data from the original worksheet. Dashboards and stories cannot be duplicated as crosstabs.

Rename sheets

To rename the active sheet, right-click (control-click on Mac) the sheet in the sheet tab along the bottom of the workbook, and then click Rename Sheet.

Or, double-click the name of the sheet in a sheet tab, type a new name, and then click Enter.

See the underlying data in a sheet

To see the numbers behind the marks in your view

Hover over or click a mark to display the associated data in a tooltip. With the tooltip open, click the View Data command at the top of the tooltip to view underlying data.

Right-click the view (with no marks selected), and then click View Data.

You can copy and paste the data into Excel or another application. To export the underlying data, click Export All.

Delete sheets

Deleting a sheet removes it from the workbook. There must always be at least one worksheet in a workbook.

To delete the active sheet, right-click (control-click on Mac) the sheet in the sheet tab along the bottom of the workbook, and then click Delete.

Worksheets used in a dashboard or story cannot be deleted, but they can be hidden.

A worksheet used as Viz in Tooltip can be hidden or deleted.
Create or Open Workbooks

A workbook contains one or more sheets, each of which can be a worksheet, dashboard, or story. You can use workbooks to organize, save, share, and publish your results.

- Create or open a workbook below
- Open a workbook that has an unsupported data connection on the Mac below
- Open a published workbook from the server on the next page

Create or open a workbook

When you open Tableau, a new, blank workbook opens automatically.

To create a new workbook, select File > New

To open an existing workbook, click the thumbnail image of the workbook on the start page.

You can open multiple workbooks simultaneously. Each workbook appears in its own window.

Note: When you open multiple workbooks in Tableau Desktop on the Mac, multiple instances of the application are created, each with its own icon in the Dock. This differs from typical Mac application behavior, where one instance of the application handles all open documents.

Open a workbook that has an unsupported data connection on the Mac

When working in Tableau Desktop on the Mac, you might need to open a Windows workbook that contains data sources that are not supported in Tableau Desktop on the Mac. To do so, follow the steps below:

1. In Tableau Desktop on Windows, save the workbook as an extract. To create and save an extract, on the Data menu, select a data source, and then select Extract Data.
2. Open the extract file (.hyper) you saved in Tableau Desktop on your Mac.

Open a published workbook from the server

If you have been granted the **Download/Web Save As** permission for a published workbook, you can use Tableau Desktop to open the workbook from the server. When you open a workbook from the server and make changes, you can save it to your computer or, if you have been allowed the **Write/Web Save** permission, you can save the changes directly on the server.

**To open a workbook from the server**

1. Select **Server > Open Workbook**.

2. If you are not already signed in to Tableau Server or Tableau Online, do so at the prompt.

If you’re not sure how to sign in to the server you use, see **Sign in to Tableau Server or Online** on page 310.
3. In the Open Workbook from Tableau Server dialog box, select the workbook you want to open, and then click **Open**.

![Open Workbook from Tableau Server dialog box](image)

**Tip:** You can use the **Find** drop-down list or the Search box to browse or search for the workbook you want.

You can search all workbooks on the server or find by tags, publisher, project, or workbooks that you published.
Navigate and Organize Sheets

You can navigate and organize sheets in a workbook using the sheet tabs, the filmstrips, or the sheet sorter.

In this article

- Options for navigating, viewing, and organizing sheets
  - Navigate with sheet tabs
  - Show thumbnails in the filmstrip
  - Sheet commands

Options for navigating, viewing, and organizing sheets

There are three ways to navigate and view the sheets in a workbook:

- Tabs at the bottom of the workbook
- In the filmstrip view
- In the sheet sorter view

The tabs are useful when you want to quickly navigate between a small number of sheets. If your workbook has a large number of sheets, you can use the sheet sorter to easily navigate them all.

You can also drag and drop to do the following:

- Reorder the sheets
- Create new sheets
- Duplicate or delete existing sheets from any of the views
Navigate with sheet tabs

Each sheet is represented as a tab along the bottom of the workbook. Select any tab to open the corresponding worksheet.

In the bottom right corner of the application window, there are several controls that you can use to advance through each sheet or quickly jump to the first or last sheet in the workbook. These controls are only available when there are too many sheet tabs to show across the bottom of the application window.

You can also navigate between sheets using the window menu or move through the multiple worksheets by pressing the left or right arrow keys on your keyboard.

To navigate through multiple worksheets, select a worksheet tab at the bottom of the workbook first.

**Note:** Navigating within a story is a different matter—there, you use the Navigator to move between story points. See [Create a Story](#) on page 2325.

Show thumbnails with the filmstrip

Similar to the sheet tabs, the filmstrip displays along the bottom of the workbook. However, instead of just sheet names, the filmstrip also shows a thumbnail image of each sheet. The filmstrip is useful when you are using Tableau to present your analysis and works well when you are working in Presentation mode.
Open the filmstrip by clicking the **Filmstrip** button on the status bar (bottom right corner) of the workbook. Just like with the tabs, select the thumbnail image for the sheet you want to open. You can right-click the images to specify commands that apply to each sheet.

**Note:** When viewed from a Retina display, workbooks that are created on a standard resolution device will show only the first thumbnail in the filmstrip in high resolution. To display all thumbnails in the filmstrip in high resolution, resave the workbook on a computer with Retina display.

---

Manage many sheets with sheet sorter

The sheet sorter shows all sheets in a workbook as thumbnail images on a single page. The sheet sorter is useful when you have a large number of sheets in a workbook. Open the sheet sorter by clicking the sheet sorter button on the status bar (bottom right corner of the workbook).
Right-click a sheet (Control-click on the Mac) to see these commands. You can also right-click (Control-click on the Mac) to refresh the thumbnail image of a particular sheet or Refresh All Thumbnails at once. If you have a large number of sheets, refreshing your view can take some time.

Sheet commands

Use sheet commands to manage and organize your worksheets. For example you can create new sheets, duplicate sheets, copy formatting, apply color, or delete the sheet entirely.

You can access sheet commands on the right-click menu (Control-click on a Mac) in the worksheet, sheet sorter, or the filmstrip view. To apply commands to multiple sheets at once, press the Ctrl key (Shift key on a Mac), and then select the sheets.

To make it easier to identify and group sheets, you can assign a color to sheets. You can select from seven different colors. Selecting None clears the color.

To assign a color to sheets, select one or more sheets, right-click the sheets (Control-click on the Mac), select Color, and then pick a color.
The color strip appears on the bottom of the tab or sheet.

Reorganizing the Workspace

Every worksheet contains a variety of cards, shelves, legends, and so on. You can reorganize the workspace by rearranging cards, hiding and showing specific parts of the workspace, and hiding everything but the view using Presentation Mode.
Rearranging Cards

A worksheet contains several cards that contain shelves, legends, and other controls. Each card can be rearranged to create a custom workspace. To move a card, point the cursor at the title area of the card you want to move. When the cursor becomes the move symbol, click and drag the card to a new position. As you drag the card around the worksheet, the possible positions for it are highlighted with a black bar.

You can restore the worksheet windows to their default state by selecting **Reset Cards** on the Show/Hide Cards toolbar control.

Showing and Hiding Parts of the Workspace

Just about everything in the workspace can be turned on and off so you can avoid cluttering the worksheet with unnecessary cards, shelves, etc.
To show and hide the toolbar or status bar, select **Window** and then select what you want to hide.

To show and hide the window on the left side, which is either the Data pane (for worksheets), the Dashboard pane (for dashboards), or the Dashboards and Sheets pane (for stories); click the Minimize button in the upper right corner of the pane. The pane is minimized to the bottom left corner of the workbook. Click the same button again to restore the pane.

To show or hide a card click **Show/Hide Cards** on the toolbar and then select the card you want to show or hide.

You can restore the worksheet windows to their default state by selecting **Reset Cards** on the **Show/Hide Cards** toolbar control.

**Presentation Mode**

Sometimes you may want to use Tableau to present your findings. Rather than hiding each card or shelf one at a time, you can switch to presentation mode. Presentation mode hides everything on the sheet except for the view and its associated legends, filter cards, parameter controls, and worksheet tabs.

To switch in and out of presentation mode, click the Presentation Mode button on the toolbar or select **Window > Presentation Mode**.

Use the presentation mode controls in the bottom right corner to move between sheets and more. Each presentation mode control is described below.

| ![ ] | Show Filmstrip - shows the sheets as thumbnails at the bottom of the workspace. |
| ![ ] | Show Tabs - shows the sheet tabs at the bottom of the workspace. |
| ![ ] | Previous/Next Sheet - advances forward or backward through the sheets in a workbook. |
| ![ ] | Enter/Exit Full Screen - switches between expanding the workbook to fill the entire screen and showing it in a window. |
Tableau File Types and Folders

You can save your work using several different Tableau specific file types: workbooks, bookmarks, packaged data files, data extracts, and data connection files. Each of these file types are described below. For related details, see Save Your Work on page 2460.

- **Workbooks (.twb)** – Tableau workbook files have the .twb file extension. Workbooks hold one or more worksheets, plus zero or more dashboards and stories.

- **Bookmarks (.tbm)** – Tableau bookmark files have the .tbm file extension. Bookmarks contain a single worksheet and are an easy way to quickly share your work. For more information, see Save a bookmark on page 2462.

- **Packaged Workbooks (.twbx)** – Tableau packaged workbooks have the .twbx file extension. A packaged workbook is a single zip file that contains a workbook along with any supporting local file data and background images. This format is the best way to package your work for sharing with others who don’t have access to the original data. For more information, see Packaged Workbooks on page 2465.

- **Extract (.hyper or .tde)** – Depending on the version the extract was created in, Tableau extract files can have either the .hyper or .tde file extension. Extract files are a local copy of a subset or entire data set that you can use to share data with others, when you need to work offline, and improve performance. For more information, see Extract Your Data on page 773.

- **Data Source (.tds)** – Tableau data source files have the .tds file extension. Data source files are shortcuts for quickly connecting to the original data that you use often. Data source files do not contain the actual data but rather the information necessary to connect to the actual data as well as any modifications you've made on top of the actual data such as changing default properties, creating calculated fields, adding groups, and so on. For more information, see Save Data Sources on page 832.

- **Packaged Data Source (.tdsx)** – Tableau packaged data source files have the .tdsx file extension. A packaged data source is a zip file that contains the data source file (.tds) described above as well as any local file data such as extract files (.hyper or .tde), text files, Excel files, Access files, and local cube files. Use this format to create a single file
that you can then share with others who may not have access to the original data stored locally on your computer. For more information, see **Save Data Sources** on page 832.

These files can be saved in the associated folders in the My Tableau Repository directory, which is automatically created in your My Documents folder when you install Tableau. Your work files can also be saved in other locations, such as your desktop or a network directory.

### Changing the Repository Location

You can specify a new location for the Tableau repository if you are not using the default location in your Documents folder. For instance, if you are required to have your data on a network server instead of on your local machine, you can point Tableau at the remote repository.

1. Select **File > Repository Location**.
2. Select a new folder that will act as the new repository location in the Select a Repository dialog box.
3. Restart Tableau so that it uses the new repository.

Changing the repository location does not move the files contained in the original repository. Instead, Tableau creates a new repository where you can store your files.

### Language and Locale

Tableau Desktop is localized into several languages.

When you first run Tableau, it recognizes your computer locale and uses the appropriate language if it is supported. If you are using an unsupported language, the application defaults to English.

You can configure Tableau to display the user interface (menus, messages, etc.) by choosing **Help > Choose Language**. After you change this setting, you'll need to restart the application for the changes to take effect. You do not need to change this setting for every workbook.

To configure date and number formatting, choose **File > Workbook Locale**. By default, the locale is set to **Automatic**, which means the locale will match the locale when the workbook is opened. This can be useful if you are authoring a workbook that will be viewed in many different languages and you want the dates and numbers to update accordingly. When you select a specific locale, the workbook will not change regardless of who opens it.

Tableau checks the following, in order, to determine the workbook locale:
- Workbook Locale (explicit setting)
- Windows Locale or Mac language
- Tableau Language

If none of the above is set, then the workbook locale defaults to English.

**Day of the Week Sorting**

If you are working in a language for which Tableau does not provide a local version, set your workbook locale to assure that Tableau can sort the days of the week in the correct chronological order. Otherwise, Tableau will sort the names of the days alphabetically. If none of the supported locales is appropriate, you can sort the days of the week manually. See [Sort Data in a Visualization](#) on page 1202.

**Visual Cues and Icons in Tableau Desktop**

Tableau provides many visual cues to help you evaluate the type of data that's displayed in the Data pane and the state of a data view.

**Data sources in the Data pane**

The following table explains each of the icons used to describe the type of data sources in the Data pane. Each icon in the table can be modified by one of two indicators.

- A blue check mark indicates that the data source is the primary data source in the workbook.
- An orange check mark indicates that the data source is the secondary data source in the workbook.

<table>
<thead>
<tr>
<th>Visual Cue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![icon]</td>
<td>The workbook is directly connected to a relational data source or file.</td>
</tr>
</tbody>
</table>
### Visual Cue Description

<table>
<thead>
<tr>
<th>Visual Cue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image]</td>
<td>The workbook is connected to a cube (multidimensional) data source. In Tableau, cube data sources are supported only in Windows.</td>
</tr>
<tr>
<td>![Image]</td>
<td>The workbook is connected to an extract that still references the underlying data.</td>
</tr>
<tr>
<td>![Image]</td>
<td>The workbook is connected to a data source that has been published to Tableau Server.</td>
</tr>
</tbody>
</table>

### Fields in the Data Pane

The following table explains each of the icons displayed in the Data pane. Each icon in the table can be modified by one of four indicators.

- Blue icons indicate that the field is discrete.  
- Green icons indicate that the field is continuous.  
- Icons preceded by the equal sign (=) indicate that the field is a user-defined calculation or a copy of another field.
<table>
<thead>
<tr>
<th>Visual Cue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abc</td>
<td>The field contains text values.</td>
</tr>
<tr>
<td>#Abc</td>
<td>The field contains numeric values.</td>
</tr>
<tr>
<td>##</td>
<td>The field is a calculation defined on the server.</td>
</tr>
<tr>
<td>#</td>
<td>The field contains only date values.</td>
</tr>
<tr>
<td>#</td>
<td>The field contains both date and time values.</td>
</tr>
<tr>
<td>#</td>
<td>The field contains geographical data and has been assigned a geographic role. Use these fields when building map views. See <a href="#">Geographic Data Analysis in Tableau</a> on page 1862.</td>
</tr>
<tr>
<td>#</td>
<td>The field contains geographical data from an active custom geocoding file. See <a href="#">Geocode Locations Tableau Does Not Recognize and Plot Them on a Map</a> on page 1925.</td>
</tr>
<tr>
<td>#f #F</td>
<td>The field contains boolean (true or false) values.</td>
</tr>
<tr>
<td>#AaBb</td>
<td>The field is a calculation that is defined in the database by an administrator. These fields are marked with a cylinder icon and are not available for all data sources.</td>
</tr>
<tr>
<td>Visual Cue</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>The field is a user-defined set. See <strong>Create Sets</strong> on page 1004.</td>
</tr>
<tr>
<td></td>
<td>The field is a server named set.</td>
</tr>
<tr>
<td></td>
<td>The field is a set that was automatically created as a result of an action.</td>
</tr>
<tr>
<td></td>
<td>The field is a user filter, used when publishing to the web. See <strong>Restrict Access at the Data Row Level</strong> on page 2541.</td>
</tr>
<tr>
<td></td>
<td>The field is a numeric bin. See <strong>Create Bins from a Continuous Measure</strong> on page 956.</td>
</tr>
<tr>
<td></td>
<td>The field is a group. See <strong>Correct Data Errors or Combine Dimension Members by Grouping Your Data</strong> on page 1002.</td>
</tr>
<tr>
<td></td>
<td>The field is a relational hierarchy. See <strong>Create Hierarchies</strong> on page 987.</td>
</tr>
<tr>
<td></td>
<td>A folder that contains one or more fields. Folders are used to organize fields in the Data pane. See <strong>Organize the Data Pane</strong> on page 962.</td>
</tr>
<tr>
<td></td>
<td>The field is an attribute of a cube (multidimensional) data source. In Tableau, cube data sources are supported only in Windows.</td>
</tr>
<tr>
<td></td>
<td>The field is a varying attribute of a cube (multidimensional) data source.</td>
</tr>
<tr>
<td></td>
<td>The field is a level in a multidimensional hierarchy. Levels greater than five are shown without numbers.</td>
</tr>
<tr>
<td></td>
<td>The field is blended with a field from another data source. See <strong>Primary and secondary data sources</strong> on page 683</td>
</tr>
</tbody>
</table>
### Visual Cue Description

<table>
<thead>
<tr>
<th>Visual Cue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c/o</td>
<td>The field is not blended with a field from another data source. See <a href="#">Primary and secondary data sources</a> on page 683</td>
</tr>
</tbody>
</table>

## Fields on Shelves

Fields placed on shelves use a combination of icons, colors, and text styles as visual cues.

<table>
<thead>
<tr>
<th>Visual Cue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region</td>
<td>A blue field on a shelf indicates a discrete field. In most cases, adding a dimension to a shelf results in a blue field. Blue fields are discrete—they contain a finite number of values. Adding a blue field to a shelf creates headers. For details, see <a href="#">Headers</a> on page 206.</td>
</tr>
<tr>
<td>SUM(Sales)</td>
<td>A green field on a shelf indicates a continuous field. In most cases, adding a measure to a shelf results in a green field. Green fields are continuous—they contain an infinite number of values. Adding a green field to a shelf creates an axis. For details, see <a href="#">Axes</a> on page 209.</td>
</tr>
<tr>
<td>Company</td>
<td>The Sort icon indicates a field that has either a computed or manual sort order applied. See <a href="#">Sort Data in a Visualization</a> on page 1202.</td>
</tr>
<tr>
<td>Date.Fiscal</td>
<td>The sigma icon indicates a slicing filter in a multidimensional (cube) data source. See <a href="#">Create Slicing Filters</a> on page 2222.</td>
</tr>
<tr>
<td>Core Product Gro..</td>
<td>The Venn diagram icon indicates a set. See <a href="#">Create Sets</a> on page 1004.</td>
</tr>
<tr>
<td>Visual Cue</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>A field name shown in italics indicates a filtered set.</td>
<td></td>
</tr>
<tr>
<td>A gray field on the Filters shelf indicates a context filter. See Improve View Performance with Context Filters on page 1194.</td>
<td></td>
</tr>
<tr>
<td>The delta icon indicates that the field is a table calculation. See Transform Values with Table Calculations on page 1524.</td>
<td></td>
</tr>
<tr>
<td>The plus and minus controls appear when the field is part of a hierarchy that you can traverse.</td>
<td></td>
</tr>
<tr>
<td>The arrow icon indicates that a forecast is being displayed for the field. This icon is also used on the Forecast Indicator field, which is an automatic field used to distinguish between actual and forecast values. See Forecasting on page 1705.</td>
<td></td>
</tr>
<tr>
<td>The field is from a secondary data source. See Blend Your Data on page 682.</td>
<td></td>
</tr>
<tr>
<td>The field is assigned to a specific worksheet.</td>
<td></td>
</tr>
<tr>
<td>The field is assigned to all worksheets with the same data source.</td>
<td></td>
</tr>
<tr>
<td>The field is incompatible with one or more other fields in the view.</td>
<td></td>
</tr>
</tbody>
</table>

Fields on the Marks card

Fields placed on the Marks card use specific icons to describe how they appear in the view. For more details, see Control the Appearance of Marks in the View on page 1123.
### Visual Cue Description

<table>
<thead>
<tr>
<th>Visual Cue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✍️ Segment</td>
<td>The field is applied to Color on the Marks card.</td>
</tr>
<tr>
<td>☐ ☑ SUM(Profit)</td>
<td>The field is applied to Size on the Marks card.</td>
</tr>
<tr>
<td>☑ ☑ AVG(Discou..)</td>
<td>The field is applied to Label on the Marks card.</td>
</tr>
<tr>
<td>☑ ☑ Sub-Category</td>
<td>The field is applied to Shape on the Marks card.</td>
</tr>
<tr>
<td>✍️ SUM(True Average)</td>
<td>The field is applied to Detail on the Marks card.</td>
</tr>
<tr>
<td>☑ ☑ AVG(Math)</td>
<td>The field is applied to Tooltip on the Marks card.</td>
</tr>
<tr>
<td>☐ ☑ YEAR(Date)</td>
<td>The field is applied to Path on the Marks card. Path is only available when the Line or Polygon mark type is selected from the Marks drop-down menu.</td>
</tr>
</tbody>
</table>

### Sheets in the Dashboards and Worksheets pane

The following table explains each of the icons used to describe the type of sheet that can be placed in a story. A blue check mark indicates that a sheet is being used in one or more story points.

<table>
<thead>
<tr>
<th>Visual Cue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐ ☑</td>
<td>The sheet is a worksheet.</td>
</tr>
<tr>
<td>☑ ☑</td>
<td>The sheet is a dashboard.</td>
</tr>
</tbody>
</table>
Tableau Concepts

Why are some fields dimensions and others measures?

Why is the background color blue for some fields, and green for others?

How does adding a filter affect my viz?

How does Tableau aggregate data in a view?

The topics in this section cover concepts that answer these questions to help you understand why Tableau does what it does.

If you're new to Tableau Desktop, also see other topics in Get Started on page 118.

If you're new to Tableau Public, also see Tableau Public Resources.

Watch a video: To see related concepts demonstrated in Tableau, watch the free training videos listed under Why is Tableau Doing That?. Use your tableau.com account to sign in.

Dimensions and Measures, Blue and Green

When you connect to a new data source, Tableau assigns each field in the data source to either the Dimensions area or the Measures area of the Data pane, depending on the type of data the field contains. You use these fields to build views of your data.

Watch a Video: To see related concepts demonstrated in Tableau, watch Understanding Pill Types, a 5-minute free training video. Use your tableau.com account to sign in.

In this article

- About data fields (dimensions, measures, blue, green)
- Examples of continuous and discrete fields used in a view
- Dimension fields in the view
- How dimension fields affect the level of detail
- Measure fields in the view and default aggregation
About data field roles and types

Data fields are made from the columns in your data source. Each field is automatically assigned a data type (such as integer, string, date), and a role: Discrete Dimension or Continuous Measure (more common), or Continuous Dimension or Discrete Measure (less common).

- **Dimensions** contain qualitative values (such as names, dates, or geographical data). You can use dimensions to categorize, segment, and reveal the details in your data. Dimensions affect the level of detail in the view.

- **Measures** contain numeric, quantitative values that you can measure. Measures can be aggregated. When you drag a measure into the view, Tableau applies an aggregation to that measure (by default).

Blue versus green fields

Tableau represents data differently in the view depending on whether the field is discrete (blue), or continuous (green). **Continuous** and **discrete** are mathematical terms. Continuous means "forming an unbroken whole, without interruption"; discrete means "individually separate and distinct."

- Green measures **SUM(Profit)** and dimensions **YEAR(Order Date)** are continuous. Continuous field values are treated as an infinite range. Generally, continuous fields add axes to the view.

- Blue measures **SUM(Profit)** and dimensions **Product Name** are discrete. Discrete values are treated as finite. Generally, discrete fields add headers to the view.

Possible combinations of fields in Tableau

This table shows examples of what the different fields look like in the view. People sometimes call these fields "pills", but we refer to them as "fields" in Tableau help documentation.
Continuous Dimensions (dimensions with a data type of String or Boolean cannot be continuous)

Discrete Measures

Continuous Measures

Note: With a cube (multidimensional) data source, the options for changing data roles are limited. In Tableau Desktop, cubes are supported only on Windows.) You can change some measures from continuous to discrete, but in general, you cannot change data roles for fields in cube data sources. For related details, see Cube Data Sources on page 2216.

A visual cue that helps you know when a field is a measure is that the field is aggregated with a function, which is indicated with an abbreviation for the aggregation in the field name, such as:

SUM(Profit)

To learn more about aggregation, see List of Predefined Aggregations in Tableau on page 283 and Aggregate Functions in Tableau on page 1320.

But there are exceptions:

- If the entire view is disaggregated, then by definition no field in the view is aggregated. For details, see How to Disaggregate Data on page 289.
- If you are using a multidimensional data source, fields are aggregated in the data source and measures fields in the view do not show that aggregation.

Note: You can set the default aggregation and other properties and settings for fields. For details on the many ways you can customize the fields in the Data pane, see Organize and Customize Fields in the Data Pane on page 962, Edit Default Settings for Fields on page 973, and Work with Data Fields in the Data Pane on page 160.
Examples of continuous and discrete fields used in a view

In the example on the right (below), because the Quantity field is set to **Continuous**, it creates a horizontal axis along the bottom of the view. The green background and the axis help you to see that it's a continuous field.

In the example on the right, the Quantity field has been set to **Discrete**. It creates horizontal headers instead of an axis. The blue background and the horizontal headers help you to see that it's discrete.

In both examples, the Sales field is set to **Continuous**. It creates a vertical axis because it is continuous and it's been added to the Rows shelf. If it was on the Columns shelf, it would create a horizontal axis. The green background and aggregation function (in this case, SUM) help to indicate that it's a measure.

The absence of an aggregation function in the Quantity field name help to indicate that it's a dimension.
Dimension fields in the view

When you click and drag a discrete dimension field from the Dimensions area to Rows or Columns, Tableau creates column or row headers.

In many cases, fields from the Dimension area will initially be discrete when you add them to a view, with a blue background. Date dimensions and numeric dimensions can be discrete or continuous, and all measures can be discrete or continuous.

After you drag a dimension to Rows or Columns, you can change the field to a measure just by clicking the field and choosing Measure. Now the view will contain a continuous axis instead of column or row headers, and the field's background will become green:
Date dimensions can be discrete or continuous. Dimensions containing strings or Boolean values cannot be continuous.

Tableau does not aggregate dimensions. For a discussion of the different types of aggregation Tableau can perform, see List of Predefined Aggregations in Tableau on page 283.

In Tableau queries, dimensions in the view are expressed in SQL as "Group By" clauses.

For details on converting fields between continuous and discrete, see Convert Fields between Discrete and Continuous on page 980.

How dimensions affect the level of detail in the view

The level of detail in a view refers to how granular the data is given the dimension and measure data in the view.

As you add dimensions to Rows or Columns, the number of marks in the view increases.

To understand why adding dimensions increases the number of marks in the view, do the following:

1. Drag Segment to Columns.
   
   The status bar at the bottom of the Tableau window shows you that there are now three marks in the view:

   ![Data Source Sheet 1](image)

   Those marks just contain placeholder text, Abc, because you are only building the view's structure at this point.

2. Drag Region to Columns.
   
   Now there are 12 marks. Three values in Segment multiplied by four values in Region is 12.

3. Drag [Ship Date] to Rows.
   
   The total is now 57 marks (three segments by four regions by five years is 60, but there are three combinations of the dimensions in the view for which there is no data in the data source).
We could continue adding dimensions to **Rows** and **Columns** and observe as the number of total marks continues to increase. Dragging a dimension to a location on the Marks card such as Color or Size will also increase the number of marks, though it will not increase the number of headings in the view. The process of adding dimensions to the view to increase the number of marks is known as setting the *level of detail*.

Adding a dimension to any of the following locations in Tableau affects the level of detail:

4. The view now contains 57 separate instances of Abc—the view is all structure and no content. Rectify this by dragging **Sales** to Text. The view can now be considered complete:
Notes

- In some cases, adding a measure to the view can increase the number of marks in the view. For example, if you dropped Sales on Rows in the view above, the number of marks would be 57. But if you then also dropped Profit on Rows, the number of marks would increase to 114. But this is not the same as changing the view’s level of detail.

- The number of marks in the view is not guaranteed to correspond to the number you would get by multiplying the number of dimension values in each of the dimensions that make up the level of detail. There are multiple reasons why the number of marks could be lower. To increase the number of marks in this view from 57 to 60 in the view above, right-click (Control-click on a Mac) on one of the Date headers in the view and the date or bin headers and choose Show Missing Values. For more information about how to show missing values, see Show or Hide Missing Values or Empty Rows and
Measure fields in the view

When you drag a measure to the view, it is aggregated by default. The type of aggregation will vary depending on the type of view. You should always check the aggregation and change it if necessary. For details, see "Change the default aggregation" in Edit Default Settings for Fields on page 973. For more details about aggregation, see Data Aggregation in Tableau on page 279.

When you drag a continuous field from the Measures area to Rows or Columns, Tableau creates a continuous axis for that field.
If you click the field and change it to **Discrete**, the values become column headers.

Tableau continues to aggregate values for the field, because even though the field is now discrete, it is still a measure, and Tableau aggregates measures by default.

In cases where Tableau has misclassified a field as a dimension or a measure, possibly because of the data type, you can convert it and change its role. If a measure contains numbers that don't need to be aggregated (such as a field that contains date values), you may want to convert it to be a dimension.

For related details, see **Convert a Measure to a Dimension** on page 981.
For details on converting fields between continuous and discrete, see Convert Fields between Discrete and Continuous on page 980.

---

**How continuous and discrete fields change the view**

*Continuous* and *discrete* are mathematical terms. Continuous means "forming an unbroken whole, without interruption"; discrete means "individually separate and distinct."

In Tableau, fields can be either continuous or discrete. When you drag a field from the **Measures** area to **Columns** or **Rows**, the values are continuous by default and Tableau creates an axis. When you drag a field from the **Dimensions** area of the **Data** pane to **Columns** or **Rows**, the values are discrete by default and Tableau creates column or row headers.

**Continuous fields produce axes**

If a field has values that are numbers that can be added, averaged, or otherwise aggregated, Tableau assigns that field to the **Measures** area of the **Data** pane when you first connect to a data source. Tableau is assuming that the values are continuous.

Tableau displays an axis when you drag a continuous field to **Rows** or **Columns**. An axis is a measuring line that shows values between a minimum and a maximum. Rulers and analog thermometers are examples of physical objects that display axes.

Tableau must be able to show a range of actual and potential values, because in addition to the initial values in the data source, it is always possible that new values will emerge as you work with a continuous field in the view.

While there are value labels on a continuous axis (0, 0.5, ... 3.0 in the following image), actual marks don't have to align with these labels as they would with column headers. For example, in the following image, the blue bar actually extends to a value of 6.940 on the horizontal axis, not 7.0 exactly.
The number of potential values for continuous fields is impossible to anticipate. For example, if you have a field named Ratings and the initial values are 1, 3, 3.5, 3.6, and 4, that's five distinct values. But if you drop Ratings on Rows, Tableau automatically aggregates that value as SUM (which you would then immediately change to AVG, because it's more logical to average grades than to add them), and that would then create a sixth value (3.02) that didn't exist until you added the field to the view. And if you then applied a filter that eliminated two of the initial values, the average would change as well, so that would be yet another value. And then if you changed the aggregation, ... You get the idea. The number of potential values is, if not infinite, then certainly immense.

The fact that a field contains numbers does not automatically indicate that those values are continuous. Postal codes are the classic example: though they are often composed entirely of numbers, they are actually string values which shouldn't be added or averaged. If Tableau assigns such a field to the Measures area, you should drag it up to the Dimensions area.

Discrete fields create headers

If a field contains values that are names, dates, or geographical locations—anything other than numbers—Tableau assigns that field to the Dimensions area of the Data pane when you first connect to a data source. Tableau treats the values as discrete.

Tableau creates headers when you drag a discrete field to Columns or Rows. The individual values for a discrete field become the row or column headings. Because these types of values are never aggregated, no new field values are created as you work with your view, so there is no need for an axis.
Discrete versus continuous fields on filters

- When you drop a discrete dimension field on the Filters shelf, Tableau prompts you to choose which "members" of the discrete field to include in the view.

![Filter dialog box](image)

- When you drop a Date field on Filters, the result can be a discrete filter or a continuous filter. For more information, see Filter dates on page 1171.

- When you drop a continuous measure on Filters, Tableau first prompts you to choose an aggregation for the filter, and then prompts you to specify how to filter the continuous range of values.

- When you drop a continuous dimension on Filters (other than a Date), Tableau prompts you to specify how to filter the continuous range of values.

For more on filtering various types of fields, see Drag dimensions, measures, and date fields to the Filters shelf on page 1165.

Discrete versus continuous fields on color

When you drop a discrete field on Color in the Marks card, Tableau displays a categorical palette and assigns a color to each value of the field.
When you drop a continuous field on Color, Tableau displays a quantitative legend with a continuous range of colors.

For more information about color palettes, see *Color Palettes and Effects* on page 1145.

---

**Data Types**

All fields in a data source have a data type. The data type reflects the kind of information stored in that field, for example integers (410), dates (1/23/2015) and strings (“Wisconsin”). The data type of a field is identified in the Data pane by one of the icons shown below.

**Data type icons in Tableau**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>🍃</td>
<td>Text (string) values</td>
</tr>
<tr>
<td>🕒</td>
<td>Date values</td>
</tr>
<tr>
<td>🕒🌟</td>
<td>Date &amp; Time values</td>
</tr>
<tr>
<td>📊</td>
<td>Numerical values</td>
</tr>
<tr>
<td>🟢</td>
<td>Boolean values (relational only)</td>
</tr>
<tr>
<td>🌍</td>
<td>Geographic values (used with maps)</td>
</tr>
</tbody>
</table>

You can change the data type for a field either on the **Data Source** page or in the **Data** pane.

**Change the data type for a field in the Data Source page**

Sometimes Tableau incorrectly interprets the data type of a field. For example, Tableau might interpret a field that contains dates as an integer data type, rather than a date data type.
You can change the data type for a field that was part of the original data source (as opposed to a calculated field created in Tableau) on the Data Source page.

1. Click the data type icon for the field (as shown in the table above).

2. Choose a new data type from the drop-down list:

   ![Data Type Dropdown]

   **Tip:** Be sure to change data types before you create an extract. Otherwise, your data may not be accurate. For example, if a floating-point field in the original data source is interpreted as an integer by Tableau, and you create your extract before you change the field's data type, the resulting floating-point field in Tableau will have some of its precision truncated.

For information on changing data types on the Data Source page, see Data Source Page on page 139.

Change the data type for a field in the Data pane

To change the data type of a field in the Data pane, click the icon to the left of the field name, and then choose a new data type from the drop-down list.
Change the data type for a field in the view

To change a field's data type in a view, right-click (control-click on a Mac) the field in the Data pane, choose Change Data Type, and then select the appropriate data type from the drop-down list.
Note: Sometimes the data in your database is more precise than Tableau can model. When you add these values to the view, a precision warning appears in the right corner of the status bar. See Status Bar Information on page 151.

Mixed data types in data from files

Most columns in a Microsoft Excel, Microsoft Access, or CSV (comma-separated value) file contain values that are all of the same data type (Booleans, dates, numbers, or text). When you connect to the file, Tableau creates a field in the appropriate area of the Data pane for each column. Dates and text values are dimensions, and numbers are measures.

However, files that you connect to might include columns that have a mixture of data types, such as numbers and text, or numbers and dates. When you connect to the file, the mixed-value column is mapped to a field with a single data type in Tableau. Therefore, a column that contains numbers and dates might be mapped as a number data type (making it a measure) or it might be mapped as a date data type (in which case Tableau treats it as a dimension).

Tableau determines how to map mixed-value columns to data types by the data types of the first 10,000 rows in an Excel data source, and the first 1,024 rows in a CSV data source. For example, if most of the first 10,000 rows are text values, the entire column is mapped to use the text data type.

Note: Empty cells also create mixed-value columns because their formatting is different from text, dates, or numbers.

When Tableau determines a data type for each field, if the values in a field don’t match that data type, Tableau handles the field in one of several different ways, depending on the data type. For example, sometimes Tableau populates those fields with Null values, as shown in the following table:

<table>
<thead>
<tr>
<th>Mapped data type</th>
<th>Treatment of other data types in the field.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Dates and numbers are treated as text. Nulls are not created.</td>
</tr>
<tr>
<td>Dates</td>
<td>Text is treated as Null. A number is treated as the day in numeric order from 1/1/1900.</td>
</tr>
<tr>
<td>Numbers</td>
<td>Text is treated as Null. A date is treated as the number of days since</td>
</tr>
<tr>
<td>Mapped data type</td>
<td>Treatment of other data types in the field</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>1/1/1900.</td>
</tr>
<tr>
<td>Boolean</td>
<td>Text, dates, and numbers are treated as Null.</td>
</tr>
</tbody>
</table>

If using fields that are based on mixed-value columns introduces difficulties when analyzing your data, you can do one of the following:

- Format empty cells in your underlying data source so that they match the data type of the column.
- Create a new column that does not contain the mixed values.

### Tableau's Order of Operations

The order of operations in Tableau, sometimes called the query pipeline, is the order in which Tableau performs various actions. Actions are also known as operations. Many operations apply filters, which means that as you build a view and add filters, those filters always execute in the order established by the order of operations.

**Watch a Video:** To see related concepts demonstrated in Tableau, watch Understanding Tableau’s Order of Operations for Smarter Analytics, a 42-minute free presentation.

This article includes two scenarios for updating a view to correct problems resulting from the order of operations: converting a dimension filter to a context filter, and converting a table calculation to a FIXED level of detail expression.

### In this article

- **About the order of operations (aka query pipeline) on** the next page
- **Example 1: Convert a Dimension Filter to a Context Filter** on page 270
- **Example 2: Convert a Table Calculation to a FIXED Level of Detail Expression** on page 273
About the order of operations (aka query pipeline)

Sometimes, you might expect Tableau to execute filters in one order, but the order of operations dictates that they be executed in a different order, which gives you unexpected results. When this happens, you can sometimes change the order in which operations are executed in the pipeline.

The Tableau order of operations includes all the elements in the following illustration. Filters are shown in blue; other operations, which are mostly calculations, are shown in black.
Note: In the order of operations, the latest date filter is global to the workbook, while context filters apply per worksheet. The latest date is determined just after the workbook opens for first use, after data source filters, but before context filters. At that point the date is set, and the latest date preset is used as a dimension filter.
Example 1: Convert a Dimension Filter to a Context Filter

This and the following example use the Sample — Superstore data source provided with Tableau Desktop.

In this example, the view addresses the following question: Who are the top 10 customers, by total sales, in New York City?

The view contains two dimension filters, one that you create on the General tab in the Filters dialog box, and the other on the Top N tab. The problem is that these filters are executing simultaneously, whereas you would like to general filter to be applied before the top n filter, so that the top n filter can act on the results as previously filtered by the general filter. The solution is to redefine one of the filters as a context filter so that a clear order of precedence is established.

Here are the steps for building this view.

1. Drag Sales to Columns.
2. Drag City and [Customer Name] to Rows.
3. Drag City from the Data pane again, this time to Filters. On the General tab in the Filter dialog box, set the filter to show just a single value: New York City. Do this by clicking None and then choosing New York City.

   This creates a general dimension filter.

4. Click the Sort Descending button ( ) on the toolbar. Your view now looks like this:

![Image]

Note the first few names in the list: Ashbrook, Fuller, Vernon, etc.
5. Now drag [Customer Name] from the Data pane to Filters, and create a Top 10 Filter, to show only the top 10 customers by total sales:

![Filter [Customer Name] dialog box showing top 10 filter settings]

After you apply this second filter, the view looks right, but notice that the names shown are no longer the same as before:

![Tableau view showing top 10 customers]

What happened to Peter Fuller, formerly in second place? The goal was to show the top 10 customers in New York City, but now the view is actually showing the top 10 customers overall.

The problem is that top and general dimension filters are applied simultaneously—they are both dimension filters, and they appear in the same place in the Tableau order of operations:
The solution is to add the general dimension filter (on City) to context—that is, by turning it into a context filter, which is executed before any other filter that you create in a worksheet.

For details, see *Improve View Performance with Context Filters* on page 1194.

6. Right-click **City** on the **Filters** shelf (Control-click on a Mac) and choose **Add to Context**. As a context filter, this filter now takes precedence over the dimension filter, and so the view now shows what it’s supposed to:
Example 2: Convert a Table Calculation to a FIXED Level of Detail Expression

In this example, the view addresses the following question: What is the percent of total sales by product sub-category?

The view contains a dimension filter and a table calculation. Tableau applies the dimension filter before executing the table calculation. To invert the order of these operations, use a FIXED level of detail expression instead of a table calculation.

Here are the steps for building this view.

1. In a new worksheet, drag Sales to Columns.
2. Drag Sub-Category to Rows.
3. Right-click SUM(Sales) on Columns and select a quick table calculation – Percent of Total.
4. Click the Sort Descending button (▲▼) on the toolbar to sort the categories from most to least.
5. Click the Show Mark Labels button (□) on the toolbar to display measure values in the view.

Your view now looks like this:
Note the percentages for the first few items: 14.37%, 14.30%, etc.

6. Right-click **Sub-Category** on **Rows** and choose **Show Filter**.

7. Clear the check mark for **Chairs** in the filter.
In the view, the percentages are now different—the highest percentage is now over 16%. In some cases, this may be just the result that you want (that is, for percentages to be recalculated as you work with the quick filter). But in other cases you may want the percentages to hold steady even as you filter items in or out. That’s what we want in this case.

In the order of operations, a dimension filter is applied before a table calculation:
To have Tableau calculate the percentages before it acts on the quick filter, you create a FIXED level of detail expression, and then use that instead of the table calculation.

FIXED level of detail expressions compute a value using the specified dimensions, without reference to the dimensions in the view. In this case you’ll use it to establish percentages for the various sub-categories—percentages that won’t be affected by your
general dimension filter. Why? Because FIXED level of detail expressions are computed before dimension filters are applied.

For details, see Create Level of Detail Expressions in Tableau on page 1567.

8. The FIXED level of detail expression must divide the sum of Sales (for a particular measure value) by the total sum of Sales for the view. Because the numerator is aggregated, the denominator must be as well, so the expression you write is:

$$\text{SUM}([\text{Sales}]) / \text{SUM}([\text{FIXED : SUM}([\text{Sales}])])$$

9. Save that expression as FixedSumOfSales and then drag it from the Data pane to Columns, dropping it to the right of the existing SUM(Sales) field that uses the table calculation. (Keep them both in the view for comparison.) Here is what your view now looks like:

![Chart showing percentage numbers consistent across different fields with quick filter applied.](image)

The percentage numbers in the chart on the right are now consistent, regardless of which fields you select or don’t select with the quick filter. All that remains is to format the values for FixedSumOfSales so that they show as percentages.

10. Right-click FixedSumOfSales on Columns and choose Format. In the Format pane, choose Numbers and then Percentage:
This gives you the final view:

As you select or clear items in the Sub-Category quick filter, the percentages in the bar chart on the left change, but the percentages in the bar chart on the right do not.
Data Aggregation in Tableau

In Tableau, you can aggregate measures or dimensions, though it is more common to aggregate measures. Whenever you add a measure to your view, an aggregation is applied to that measure by default. The type of aggregation applied varies depending on the context of the view.

Watch a Video: To see related concepts demonstrated in Tableau, watch Aggregation, Granularity, and Ratio Calculations, a 4-minute free training video. Use your tableau.com account to sign in.

In this article

List of Predefined Aggregations in Tableau on page 283
Set the Default Aggregation for a Measure on page 287
Change the Aggregation of a Measure in the View below
Aggregating Dimensions on the next page
How to Disaggregate Data on page 289
Example: Scatter Plots, Aggregation, and Granularity on page 290

Change the Aggregation of a Measure in the View

When you add a measure to the view, Tableau automatically aggregates its values. Sum, average, and median are common aggregations; for a complete list, see List of Predefined Aggregations in Tableau on page 283.

The current aggregation appears as part of the measure’s name in the view. For example, Sales becomes SUM(Sales). Every measure has a default aggregation which is set by Tableau when you connect to a data source. You can view or change the default aggregation for a measure—see Set the Default Aggregation for a Measure on page 287.

You can aggregate measures using Tableau only for relational data sources. Multidimensional data sources contain data that is already aggregated. In Tableau, multidimensional data sources are supported only in Windows.
You can change the aggregation for a measure in the view from its context menu:

![Context Menu with Measure Aggregation Options]

**Aggregating Dimensions**

You can aggregate a dimension in the view as **Minimum**, **Maximum**, **Count**, or **Count (Distinct)**. When you aggregate a dimension, you create a new temporary measure column, so the dimension actually takes on the characteristics of a measure.
Note: The Count (Distinct) aggregation is not supported for Microsoft Access data sources, and for Microsoft Excel and Text File data sources using the legacy connection. If you are connected to one of these types of data sources, the Count (Distinct) aggregation is unavailable and shows the remark "Requires extract." If you save the data source as an extract, you will be able to use the Count (Distinct) aggregation.

Another way to view a dimension is to treat it as an Attribute. Do this by choosing Attribute from the context menu for the dimension. The Attribute aggregation has several uses:

- It can ensure a consistent level of detail when blending multiple data sources.
- It can provide a way to aggregate dimensions when computing table calculations, which require an aggregate expression.
- It can improve query performance because it is computed locally.

Tableau computes Attribute using the following formula:

\[
\text{IF MIN([dimension]) = MAX([dimension]) THEN MIN([dimension]) ELSE "*" END}
\]

The formula is computed in Tableau after the data is retrieved from the initial query. The asterisk (*) is actually a visual indicator of a special type of Null value that occurs when there are
multiple values. See Troubleshoot Data Blending on page 696 to learn more about the asterisk.

Below is an example of using Attribute in a table calculation. The table shows sales by market, market size, and state. Suppose you wanted to compute the percent of total sales each state contributed to the market. When you add a Percent of Total quick table calc (see Quick Table Calculations on page 1559) that computes along State, the calculation computes within the red area shown below. This is because the Market Size dimension is partitioning the data.

When you aggregate Market Size as an Attribute, the calculation is computed within the Market (East, in the following image), and the Market Size information is used purely as a label in the display.
List of Predefined Aggregations in Tableau

Sometimes it is useful to look at numerical data in an aggregated form such as a summation or an average. The mathematical functions that produce aggregated data are called aggregation functions. Aggregation functions perform a calculation on a set of values and return a single value. For example, a measure that contains the values 1, 2, 3, 3, 4 aggregated as a sum returns a single value: 13. Or if you have 3,000 sales transactions from 50 products in your data source, you might want to view the sum of sales for each product, so that you can decide which products have the highest revenue.

You can use Tableau to set an aggregation only for measures in relational data sources. Multidimensional data sources contain aggregated data only.

Note: Using floating-point values in combination with aggregations can sometimes lead to unexpected results. For details, see Understanding data types in calculations on page 1273.

Tableau provides a set of predefined aggregations that are shown in the table below. You can set the default aggregation for any measure that is not a calculated field that itself contains an
aggregation, such as \(\text{AVG}([\text{Discount}])\). See Set the Default Aggregation for a Measure on page 287. You can also set the aggregation for a field already in the view. For details, see Change the Aggregation of a Measure in the View on page 279.

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>Description</th>
<th>Result for measure that contains 1, 2, 2, 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Returns the value of the given expression if it only has a single value for all rows in the group, otherwise it displays an asterisk (*) character. Null values are ignored. This aggregation is particularly useful when aggregating a dimension. To set a measure in the view to this aggregation, right-click (control-click on Mac) the measure and choose Attribute. The field then changes to show the text ATTR:</td>
<td>N/A</td>
</tr>
</tbody>
</table>

| Dimension   | Returns all unique values in a measure or dimension.                                                                                                                                                      | 3 values (1, 2, 3)                            |
| Sum         | Returns the sum of the numbers in a measure. Null values are ignored.                                                                                                                                    | 1 value (8)                                  |
| Average     | Returns the arithmetic mean of the numbers in a measure. Null values are ignored.                                                                                                                     | 1 value (4)                                  |
| Count (Distinct) | Returns the number of unique values in a measure or dimension. When applied to a dimension, Tableau creates a new temporary column that is a measure because the result of a count is a number. You can count numbers, dates, booleans, and strings. Null values are ignored in all cases. This aggregation is not available for the following types of workbooks: | 1 value (3)                                  |

- Workbooks created before Tableau Desktop 8.2 and that use Microsoft Excel or Text File data.
<table>
<thead>
<tr>
<th>Aggregation</th>
<th>Description</th>
<th>Result for measure that contains 1, 2, 2, 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count (Distinct)</td>
<td>Sources.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Workbooks that use legacy connections.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Workbooks that use Microsoft Access data sources.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If you are connected to a workbook that uses one of these types, Count (Distinct) is unavailable and Tableau shows the message &quot;Requires extract.&quot; To use this aggregation, extract your data. See Extract Your Data on page 773.</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>Returns the smallest number in a measure or continuous dimension. Null values are ignored.</td>
<td>1 value (1)</td>
</tr>
<tr>
<td>Maximum</td>
<td>Returns the largest number in a measure or in the given expression based on a sample population. Null values are ignored. Returns a Null if there are fewer than 2 members in the sample that are not Null. Use this function if your data represents a sample of the population.</td>
<td>1 value (0.8165)</td>
</tr>
<tr>
<td>Std. Dev (Pop.)</td>
<td>Returns the standard deviation of all values in the given expression based on a biased population. Assumes that its arguments consist of the entire population. Use this function for large sample sizes.</td>
<td>1 value (0.7071)</td>
</tr>
<tr>
<td>Variance</td>
<td>Returns the variance of all values in the given expression based on a sample. Null values are ignored. Returns a Null if there are fewer than 2 members in the sample that are not Null. Use this function if your data represents a sample of the population.</td>
<td>1 value (0.6667)</td>
</tr>
<tr>
<td>Variance (Pop.)</td>
<td>Returns the variance of all values in the given expression based on a biased population. Assumes</td>
<td>1 value (0.5000)</td>
</tr>
<tr>
<td>Aggregation</td>
<td>Description</td>
<td>Result for measure that contains 1, 2, 2, 3</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>Disaggregate</td>
<td>Returns all records in the underlying data source. To disaggregate all measures in the view, select <strong>Aggregates Measures</strong> from the Analysis menu (to clear the check mark). Tableau allows you to view data in disaggregated form (relational databases only). When data are disaggregated, you can view all of the individual rows of your data source. For example, after discovering that the sum of sales for rubber bands is $14,600, you might want to see the distribution of individual sales transactions. To answer this question, you need to create a view that shows individual rows of data. That is, you need to disaggregate the data (see <strong>How to Disaggregate Data</strong> on page 289). Another way to look at disaggregated data is to view the underlying data for all or part of a view. For more details, see <strong>View Underlying Data</strong> on page 2616.</td>
<td>4 values (1, 2, 2, 3)</td>
</tr>
</tbody>
</table>

You can also define custom aggregations as described in **Aggregate Functions in Tableau** on page 1320. Depending on the type of data view you create, Tableau will apply these aggregations at the appropriate level of detail. For example, Tableau will apply the aggregation to individual dimension members (the average delivery time in the East region), all members in a given dimension (the average delivery time in the East, West, and Central regions), or groups of dimensions (the sum of sales for all regions and for all markets).
Set the Default Aggregation for a Measure

You can set the default aggregation for any measure that is not a calculated field that itself contains an aggregation, such as \( \text{AVG}([\text{Discount}]) \). A default aggregation is a preferred calculation for summarizing a continuous or discrete field. The default aggregation is automatically used when you drag a measure to a view.

To change the default aggregation:

Right-click (control-click on Mac) a measure in the Data pane and select Default Properties > Aggregation, and then select one of the aggregation options.
Note: You can use Tableau to aggregate measures only with relational data sources. Multidimensional data sources contain aggregated data only.

You cannot set default aggregations for published data sources. The default aggregation
is set when the data source is initially published. Create a Local Copy of the published data source to adjust the default aggregation.

How to Disaggregate Data

Whenever you add a measure to your view, an aggregation is applied to that measure by default. This default is controlled by the Aggregate Measures setting in the Analysis menu.

If you decide you want to see all of the marks in the view at the most detailed level of granularity, you can disaggregate the view. Disaggregating your data means that Tableau will display a separate mark for every data value in every row of your data source.

To disaggregate all measures in the view:

- Clear the Analysis >Aggregate Measures option. If it is already selected, click Aggregate Measures once to deselect it.

When Aggregate Measures is selected, Tableau will attempt to aggregate measures in the view by default. This means that it collects individual row values from your data source into a single value (which becomes a single mark) adjusted to the level of detail in your view.

The different aggregations available for a measure determine how the individual values are collected: they can be added (SUM), averaged (AVG), or set to the maximum (MAX) or minimum (MIN) value from the individual row values.

For a complete list of the available aggregations, List of Predefined Aggregations in Tableau on page 283.
The level of detail is determined by the dimensions in your view—for information about the concept of level of detail, see How dimensions affect the level of detail in the view on page 255.

Disaggregating your data can be useful for analyzing measures that you may want to use both independently and dependently in the view. For example, you may be analyzing the results from a product satisfaction survey with the Age of participants along one axis. You can aggregate the Age field to determine the average age of participants or disaggregate the data to determine at what age participants were most satisfied with the product.

Disaggregating data can be useful when you are viewing data as a scatter plot. See Example: Scatter Plots, Aggregation, and Granularity below.

**Note:** If your data source is very large, disaggregating the data can result in a significant performance degradation.

---

**Example: Scatter Plots, Aggregation, and Granularity**

If you place one measure on the **Rows** shelf and another measure on the **Columns** shelf, you are asking Tableau to compare two numerical values. Typically, Tableau chooses a scatter plot as the default visualization in such cases. The initial view will most likely be single mark, showing the sum for all values for the two measures. This is because you need to increase the level of detail in the view.

Start building the scatter plot below
Use dimensions to add detail on the next page
Try adding more fields to the rows and columns shelves on page 293
Try disaggregating the data on page 295

Start building the scatter plot

There are various ways to add detail to a basic scatter plot: you can use dimensions to add detail, you can add additional measures and/or dimensions to the Rows and Columns shelves to create multiple one-mark scatter plots in the view, or you can disaggregate the data. And,
you can also use any combination of these options. This topic looks at these alternatives using the Sample-Superstore data source.

To create the initial view, follow these steps:

1. Place the **Sales** measure on the **Columns** shelf.
2. Place the **Profit** measure on the **Rows** shelf.

The measures are automatically aggregated as sums. The default aggregation (SUM) is indicated in the field names. The values shown in the tooltip show the sum of sales and profit values across every row in the data source.

Follow the steps below to use dimensions to add detail to the view and to disaggregate data.

### Use dimensions to add detail

Follow these steps to develop the scatter plot view you created above by adding dimensions to show additional levels of detail.
1. Drag the **Category** dimension to **Color** on the Marks card.

   This separates the data into three marks—one for each dimension member—and encodes the marks using color.

   ![Diagram showing separation of data into three marks by color]

2. Drag the **State** dimension to **Detail** on the Marks card.

   Now there are many more marks in the view. The number of marks is equal to the number of distinct states in the data source multiplied by the number of categories.

   ![Diagram showing increased number of marks with state detail]
Although more marks are now displayed, the measures are still aggregated. So regardless of whether there is one row in the data source where State = North Dakota and Category= Furniture, or 100 such rows, the result is always a single mark.

Maybe this process is developing the view in a direction you find useful, or maybe you prefer to go in a different direction—for example, by adding a time dimension to the view, or by introducing trend lines or forecasting. You decide what questions to ask.

Try adding more fields to the rows and columns shelves

Revert to the original one-mark view and follow these steps to develop the scatter plot view by adding fields to the **Rows** and **Columns** shelves.
1. Drag the **State** dimension to the **Columns** shelf.

Even if you drop **Continent** to the right of **SUM(Sales)**, Tableau moves it to the left of **SUM(Sales)**. This is because you cannot insert a dimension within a continuous axis. Instead, your view shows a separate axis for each member of the dimension.

2. Drag the **Segment** dimension to the **Rows** shelf.

You now have a view that provides an overview of Sales and Profit across states and customer segments. It can be interesting to hover over the marks in the view to see tooltip data for various segments:
Try disaggregating the data

Another way to modify your original one-mark scatter plot to display more marks is by disaggregating the data.

Clear the **Analysis > Aggregate Measures** option. If it is already selected, click **Aggregate Measures** once to deselect it.
What you have actually done is to dis-aggregate the data, because this command is a toggle that was originally selected (check mark present). Tableau aggregates data in your view by default.

Now you see a lot of marks—one for each row in your original data source:

![Disaggregated Data](image)

When you disaggregate measures, you no longer are looking at the average or sum for the values in the rows in the data source. Instead, the view shows a mark for every row in the data source. Disaggregating data is a way to look at the entire surface area of the data. It’s a quick way to understand the shape of your data and to identify outliers. In this case, the disaggregated data shows that for many rows in the data, there is a consistent relationship between sales income and profit—this is indicated by the line of marks aligned at a forty-five degree angle.
How to Set the Default Aggregation for a Measure

You can set the default aggregation for any measure that is not a calculated field that itself contains an aggregation, such as \( \text{AVG}([\text{Discount}]) \). A default aggregation is a preferred calculation for summarizing a continuous or discrete field. The default aggregation is automatically used when you drag a measure to a view. To change the default aggregation, right-click (control-click on Mac) a measure in the Data pane and select Default Properties > Aggregation and then select one of the options.

**Note:** You can use Tableau to aggregate measures only with relational data sources. Multidimensional data sources contain aggregated data only.

**Note:** You cannot set default aggregations for published data sources. The default aggregation is set when the data source is initially published. Create a Local Copy of the published data source to adjust the default aggregation.
Aggregating Data

In Tableau, you can aggregate measures or dimension, though it is more typical to aggregate measures.
Aggregating Measures

When you add a measure to the view, Tableau automatically aggregates its values. Sum, average, and median are common aggregations; for a complete list, see Data Aggregation in Tableau on page 279. The current aggregation appears as part of the measure’s name in the view. For example, Sales becomes SUM(Sales). Every measure has a default aggregation which is set by Tableau when you connect to a data source. You can view or change the default aggregation for a measure—see How to Set the Default Aggregation for a Measure on page 297.

You can aggregate measures using Tableau only for relational data sources. Multidimensional data sources contain data that is already aggregated. In Tableau, multidimensional data sources are supported only in Windows.

You can change the aggregation for a measure in the view from its context menu:
Aggregating Dimensions

You can aggregate a dimension in the view as Minimum, Maximum, Count, or Count (Distinct). When you aggregate a dimension, you create a new temporary measure column, so the dimension actually takes on the characteristics of a measure.
Note: The Count (Distinct) aggregation is not supported for Microsoft Access data sources, and for Microsoft Excel and Text File data sources using the legacy connection. If you are connected to one of these types of data sources, the Count (Distinct) aggregation is unavailable and shows the remark "Requires extract." If you save the data source as an extract, you will be able to use the Count (Distinct) aggregation.

Another way to view a dimension is to treat it as an Attribute. Do this by choosing Attribute from the context menu for the dimension. The Attribute aggregation has several uses:

- It can ensure a consistent level of detail when blending multiple data sources.
- It can provide a way to aggregate dimensions when computing table calculations, which require an aggregate expression.
- It can improve query performance because it is computed locally.

Tableau computes Attribute using the following formula:

```
IF MIN([dimension]) = MAX([dimension]) THEN MIN([dimension]) ELSE "*" END
```

The formula is computed in Tableau after the data is retrieved from the initial query. The asterisk (*) is actually a visual indicator of a special type of Null value that occurs when there are
multiple values. See Troubleshoot Data Blending on page 696 to learn more about the asterisk.

Below is an example of using Attribute in a table calculation. The table shows sales by market, market size, and state. Suppose you wanted to compute the percent of total sales each state contributed to the market. When you add a Percent of Total Quick Table Calculations on page 1559 that computes along State, the calculation computes within the red area shown below. This is because the Market Size dimension is partitioning the data.

<table>
<thead>
<tr>
<th>Market</th>
<th>Market Size</th>
<th>State</th>
<th>Sales</th>
<th>% of Total Sales along State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central</td>
<td>Major Market</td>
<td>Colorado</td>
<td>$567,197</td>
<td>31.58%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Illinois</td>
<td>$698,653</td>
<td>45.80%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ohio</td>
<td>$54,517</td>
<td>22.62%</td>
</tr>
<tr>
<td></td>
<td>Small Market</td>
<td>Iowa</td>
<td>$54,750</td>
<td>48.93%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Missouri</td>
<td>$24,647</td>
<td>21.92%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wisconsin</td>
<td>$33,069</td>
<td>29.40%</td>
</tr>
<tr>
<td>East</td>
<td>Major Market</td>
<td>Florida</td>
<td>$37,343</td>
<td>27.08%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Massachusetts</td>
<td>$29,066</td>
<td>21.67%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New York</td>
<td>$70,862</td>
<td>51.25%</td>
</tr>
<tr>
<td></td>
<td>Small Market</td>
<td>Connecticut</td>
<td>$26,429</td>
<td>33.07%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New Hampshire</td>
<td>$14,887</td>
<td>16.93%</td>
</tr>
<tr>
<td>South</td>
<td>Major Market</td>
<td>Texas</td>
<td>$37,410</td>
<td>100.00%</td>
</tr>
<tr>
<td></td>
<td>Small Market</td>
<td>Louisiana</td>
<td>$23,161</td>
<td>34.92%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New Mexico</td>
<td>$16,882</td>
<td>23.03%</td>
</tr>
</tbody>
</table>

When you aggregate Market Size as an Attribute, the calculation is computed within the Market (East, in the following image), and the Market Size information is used purely as a label in the display.
Disaggregating Data

To disaggregate all measures in the view, select Analysis > Aggregate Measures.

Tableau typically aggregates measures in your view, which means that it collects individual row values from your data source into a single value (which becomes a single mark) adjusted to the level of detail in your view. The different aggregations available for a measure determine how the individual values are collected: they can be added (SUM), averaged (AVG), or set to the maximum (MAX) or minimum (MIN) value from the individual row values. For a complete list of the available aggregations, Data Aggregation in Tableau on page 279. The level of detail is determined by the dimensions in your view—for information about the concept of level of detail, see How dimensions affect the level of detail in the view on page 255.
Disaggregating your data means that Tableau will display a separate mark for every row data value in your data source. This can be useful for analyzing measures that you may want to use both independently and dependently in the view. For example, you may be analyzing the results from a product satisfaction survey with the Age of participants along one axis. You can aggregate the Age field to determine the average age of participants or disaggregate the data to determine at what age participants were most satisfied with the product.

Disaggregating data can be useful when you are viewing data as a scatter plot. See Example – Scatter Plots and Aggregation below.

**Note:** If your data source is very large, disaggregating the data can result in a significant performance degradation.

**Example - Scatter Plots and Aggregation**

If you place one measure on the **Rows** shelf and another measure on the **Columns** shelf, you are asking Tableau to compare two numerical values. Typically, Tableau chooses a scatter plot as the default visualization in such cases. The initial view will most likely be single mark, showing the sum for all values for the two measures. This is because you need to increase the level of detail in the view.

In this article

1. **Start building the scatter plot**
2. **Use dimensions to add detail**
3. **Add more fields to the rows and columns shelves**
4. **Disaggregate the data**

Start building the scatter plot

There are various ways to add detail to a basic scatter plot: **you can use dimensions to add detail**, you can add additional measures and/or dimensions to the Rows and Columns shelves to create multiple one-mark scatter plots in the view, or you can **disaggregate the data**. And, you can also use any combination of these options. This topic looks at these alternatives using the **Sample-Superstore** data source.

To create the initial view, follow these steps:
1. Place the **Sales** measure on the **Columns** shelf.

2. Place the **Profit** measure on the **Rows** shelf.

The measures are automatically aggregated as sums. The default aggregation (SUM) is indicated in the field names. The values shown in the tooltip show the sum of sales and profit values across every row in the data source.

Follow the steps below to use dimensions to add detail to the view and to disaggregate data.

---

**Use dimensions to add detail**

Follow these steps to develop the scatter plot view you created above by adding dimensions to show additional levels of detail.

1. **Drag the Category** dimension to **Color** on the Marks card.

   This separates the data into three marks—one for each dimension member—and encodes the marks using color.
2. Drag the **State** dimension to **Detail** on the Marks card.

Now there are many more marks in the view. The number of marks is equal to the number of distinct states in the data source multiplied by the number of categories.
Although more marks are now displayed, the measures are still aggregated. So regardless of whether there is one row in the data source where State = North Dakota and Category = Furniture, or 100 such rows, the result is always a single mark.

Maybe this process is developing the view in a direction you find useful, or maybe you prefer to go in a different direction—for example, by adding a time dimension to the view, or by introducing trend lines or forecasting. You decide what questions to ask.

Add more fields to the rows and columns shelves

Revert to the original one-mark view and follow these steps to develop the scatter plot view by adding fields to the **Rows** and **Columns** shelves.
1. Drag the **State** dimension to the **Columns** shelf.

   Even if you drop **Continent** to the right of **SUM(Sales)**, Tableau moves it to the left of **SUM(Sales)**. This is because you cannot insert a dimension within a continuous axis. Instead, your view shows a separate axis for each member of the dimension.

![Diagram with State and SUM(Sales) dimensions](image)

2. Drag the **Segment** dimension to the **Rows** shelf.

   You now have a view that provides an overview of Sales and Profit across states and customer segments. It can be interesting to hover over the marks in the view to see tooltip data for various segments:

![Diagram with Segment and SUM(Sales) dimensions](image)
Disaggregate the data

Another way to modify your original one-mark scatter plot to display more marks is by disaggregating the data.

Select **Analysis > Aggregate Measures**.

What you have actually done is to dis-aggregate the data, because this command is a toggle that was originally selected (check mark present). Tableau aggregates data in your view by default.
Now you see a lot of marks—one for each row in your original data source:

![Graph showing sales and profit data](image)

When you disaggregate measures, you no longer are looking at the average or sum for the values in the rows in the data source. Instead, the view shows a mark for every row in the data source. Disaggregating data is a way to look at the entire surface area of the data. It’s a quick way to understand the shape of your data and to identify outliers. In this case, the disaggregated data shows that for many rows in the data, there is a consistent relationship between sales income and profit—this is indicated by the line of marks aligned at a forty-five degree angle.

**Sign in to Tableau Server or Online**

To access content on Tableau Server or Tableau Online, sign in via Tableau Desktop or a web browser.
Note: If you sign in via a web browser, it must be configured to allow first-party cookies.

In this article

Sign in from Tableau Desktop below
Automatically keep Tableau Desktop connected to Tableau Server or Online on page 313
Switch user accounts from Tableau Desktop (Kerberos authentication only) on page 315
Sign in with another user account in a browser on page 316
Sign in to a specific site in a browser (Tableau Server only) on page 317

Sign in from Tableau Desktop

Tableau Server

1. In Tableau Desktop, select Server > Sign In, and enter your server, and username, and password.
2. If you have access to multiple sites, select the one you want to use.

Tableau Online

1. In Tableau Desktop, select Server > Sign In, select the Tableau Online button in the bottom left corner, and enter your username and password.
2. If you have access to multiple sites, select the one you want to use.

Automatically keep Tableau Desktop connected to Tableau Server or Online

When you connect to Tableau Server or Tableau Online, Tableau Desktop keeps you signed in from session to session, if you don’t sign out. Next time you launch Tableau Desktop, you are automatically signed in to your most recent server connection. You can easily switch to other available server and site connections.

You can see the server and site you are signed into, as well as who you are signed in as, in two places:

- The **Server** menu shows the server and site that you are signed in to. **Sign out** shows you are signed in as.
• The status bar also provides this information. Hover over the username to see the current server and site.

Switch servers, switch sites

To switch servers, on the Server menu, click **Sign in to another server**, and then enter the server name and your sign-in credentials.

If you have access to multiple sites, on the Server menu, click **Sign in to another site**.
Sign out or clear all saved connections

When you sign in to a server and site, Tableau Desktop stores your credentials in a secure token that remembers your connection. After this token is in place, you can access the server directly, without having to sign in.

Click **Server > Sign out** to clear your sign-in credentials for the current connection. To remove all of your existing server connections, click **Help > Settings and performance > Clear saved server sign-ins**.

If you never want server sign-ins to be saved, a Tableau Server administrator can change the server settings to disallow connected clients. Be aware that this setting also affects other Tableau clients like Tableau Mobile. For more information, see **Authentication for Connected Devices** in the Tableau Server Help.

Switch user accounts from Tableau Desktop (Kerberos authentication only)

If your organization uses Tableau Desktop with Kerberos authentication and it doesn't succeed, you're prompted to provide a user name and password. To sign in as a different user for testing purposes, complete these steps:
1. On the Server menu, select **Switch User**.

![Server menu with Switch User selected](image)

2. In the Tableau Server Sign In dialog box, provide the new user name and password.

To later sign in using your normal credentials, select **Server > Switch to Self**.

![Server menu with Switch to Self selected](image)

**Sign in with another user account in a browser**

If you are signed in to the wrong account, you can switch to another one.

1. Click **Sign Out** on the user menu in the upper-right corner of the page.

![User menu with Sign Out selected](image)
Note: If you do not see a Sign Out command on the Tableau Server user menu, you must use the sign-out interface provided by your organization's external identity provider (IdP). For more information, contact your administrator.

2. Type your user name and password, and then click Sign In

Sign in to a specific site in a browser (Tableau Server only)

If you are a member of multiple sites, you’re prompted to select one when you sign in to the server. Either click a visible site name or search for a name to select from a list of matching sites.
The name of the current site is displayed on the site menu.
Connect to and Prepare Data

Before you begin your analysis, you must connect to your data and then set up the data source. There are many optional configurations that you can make on the Data Source page that Tableau uses to interpret and interact with your data.

The topics in this section describe how to connect to your data and use these configurations in order to optimize your data source for analysis.

Connect to Your Data

Before you can build a view and analyze your data, you must first connect Tableau to your data. Tableau supports connecting to a wide variety of data, stored in a variety of places. For example, your data might be stored on your computer in a spreadsheet or a text file, or in a big data, relational, or cube (multidimensional) database on a server in your enterprise. Or, you might connect to public domain data available on the web such as U.S. Census Bureau information, or to a cloud database source, such as Google Analytics, Amazon Redshift, or Salesforce.

In this article

Tableau Desktop below
Tableau Online and Tableau Server web authoring on
the next page
Request a new connector on the next page

Tableau Desktop

When you launch Tableau Desktop, the data connectors that are available to you are listed on the Connect pane, which is the left pane on the Start page. File types are listed first, then common server types, or servers that you've recently connected to. Click More to see the complete list of data connectors you can use.

For supported files and databases, Tableau provides built-in connectors that are built for and optimized for those types of data. If your file or database type is listed under Connect, use this named connector to connect to your data. If your file or database type is not listed, you might
have the option of creating your own connection using Other Databases (ODBC) or Web Data Connector. Tableau provides limited support for connections that you create using either of these options.

The data connectors supported by your copy of Tableau Desktop are determined by the version you purchased. For more information, see the list of data connectors on the Tableau website. After you've connected to data, you can save the connections to have them show up under the Saved data sources section on the Connect pane.

You supply different information for each data connection that you want to make. For example, for most data connections, you'll need to supply a server name and your sign-in information. With some data connections, you can Run Initial SQL on page 643 statements, and SSL-enabled servers require that you select the Require SSL check box when you connect. The following sections discuss the specific information you need to provide for each type of data you want to connect to.

Tip: You can quickly create a data source in Tableau by copying and pasting data using the clipboard. For more information, see Create a Data Source or Add a New Connection with Clipboard Data on page 641.

Tableau Online and Tableau Server web authoring

When you sign into your Tableau site and select New Workbook, the data connectors that are available to you are shown when you’re in the Connect to Data window. You can upload files using the Files tab, connect to server or cloud data sources with Connectors, or use published data sources with On this Site. If you’re signed into Tableau Online, Dashboard Starters are also available. For more information, see Creators: Connect to data on the web on page 2580.

The data connectors supported by your Tableau site are determined by your site’s server and your license level. For more information, see What can I do with a Tableau site? on page 2555 After you’ve connected to data, you can save the connections to have them show up in the Data Sources section of your site.

Request a new connector

If Tableau doesn't have a built-in connector for your data, consider requesting one on Tableau Community. Use Ideas on Community to search for your connector to see if it's been
requested, and if it has been, vote for it. If it's not listed, add it. Tableau regularly reviews Ideas on Community to help determine what features should be added to the product.

**Supported Connectors**

Follow the link below for information on how to connect to your specific data. Connectors are listed in the order that they appear on the Connect pane.

**Excel**

This article describes how to connect Tableau to Microsoft Excel file data and set up the data source. Tableau connects to .xls and .xlsx files.

To connect to a .csv file, use the Text file connector.

In this article

- Make the connection and set up the data source below
- Get more data on page 323
- Set Excel table options on page 324
- Use Data Interpreter to clean your data on page 324
- About .ttde and .hhyper files on page 324
- Changes to the way values are computed on page 324

Make the connection and set up the data source

1. After you open Tableau, under **Connect**, click **Excel**.

2. Select the Excel workbook you want to connect to, and then click **Open**.

   Alternatively, in Tableau Desktop on Windows, to connect to the Excel file using the Microsoft Jet-based connection, in the Open dialog box, click the **Open** drop-down menu, and then select **Open with Legacy Connection**. The Data Source page appears.

3. On the Data Source page, do the following:

   1. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
2. If your Excel file has one table, click the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

You can also connect to a named range or an Excel table (also known as an Excel list) in the same way you connect to a worksheet. Both the named range and Excel table function as a table in Tableau.

You create named ranges in Excel by selecting a range of cells and then selecting Define Name on the Formulas tab. Similar to named ranges, you can create an Excel table in Excel by selecting a range of cells, and then selecting Insert > Table. When you connect to a named range or Excel table in Tableau, an icon appears next to the sheet in the Data Source page as shown below.

You can connect to multiple Excel workbooks at the same time as long as each connection in the data source has a unique name.

Note: Tableau does not support pivot tables in Excel.

Microsoft Excel data source example

Here is an example of a Microsoft Excel data source is shown below.
Note: If the Excel file contains columns that are more than 254 characters wide, Tableau Desktop can't use these fields for workbooks that were created before Tableau Desktop 8.2. Also, you cannot use the legacy connection to connect to this data. Either remove the columns, modify them to fit within 254 characters prior to connecting in Tableau Desktop, or upgrade the Excel data source. For more information about upgrading data sources, see Upgrade Data Sources on page 834.

Get more data

Get more data into your data source by adding more tables or connecting to data in a different database.

- **Add more data from the current file:** From the left pane, drag additional tables to the canvas to combine data using a join or union. For more information, see Join Your Data on page 657 or Union Your Data on page 708.

- **Add more data from different databases:** In the left pane, click Add next to Connections. For more information, see Join Your Data on page 657.

  If a connector you want is not listed in the left pane, select Data > New Data Source to add a new data source. For more information, see Blend Your Data on page 682.
Set Excel table options

Excel table options are scoped to the connection. To change the table options, on the canvas, click the table drop-down arrow and then specify whether the data includes field names in the first row. If so, these names will become the field names in Tableau. If field names are not included, Tableau generates them automatically. You can rename the fields later.

Use Data Interpreter to clean your data

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter on page 752.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer’s directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can’t be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

Changes to the way values are computed

Starting from version 10.5, when you are working with extract data sources as well as data sources that use live connections to file-based data like Excel, the values in your data can be computed differently from previous versions of Tableau. This change means that you might see differences between the data and the marks in your view between version 10.4 (and earlier) and version 10.5 (and later). The purpose of this change is to improve the efficiency and scalability of your Excel data source. For more information, see Changes to values and marks in the view on page 775.

In the case of an Excel data source, one example of this change is with case sensitivity. In version 10.4 (and earlier), for comparing and sorting purposes, string values are treated as case insensitive and therefore treated the same and stored as a single string value. In version 10.5 (and later), for sorting and comparing purposes, values remain case insensitive. However,
values are case sensitive for storing purposes. This becomes evident when values are displayed on the data source page.

For example, suppose you have a column in your data that contains the values "House," "HOUSE," and "houSe." You see the following string values depending on the version of Tableau you are using:

- In version 10.4 and earlier, both on the data source page and view, you see: "House," "House," and "House."
- In version 10.5 and later, on the data source page you see: "House," "HOUSE," and "houSe." But in the view you see: "House," "House," and "House."

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Data Prep with Text and Excel Files - Watch the Tableau video (registration or sign in required).

Tips for Working with Your Data

Data can be organized in various ways. To take advantage of Tableau Desktop, Tableau recommends that you connect to data that is formatted for analysis. Specifically, data that is:

- as granular as possible rather than aggregated (such as daily weather data rather than monthly averages)
- organized like a database table (rather than a column-oriented table such as a crosstab)
- stripped of extraneous information (anything that’s not the data and its headers)

When data is structured for analysis, it's much easier to ask and answer questions. Tableau can aggregate raw data to the desired level, rather than being restricted by the aggregations already present in the data. Groups and hierarchies can be created as needed, and calculations can be performed in the flow of analysis.

Tableau Desktop has basic cleaning options and the Data Interpreter. Tableau Prep may be necessary for more complex formatting issues.

The following Tableau Desktop-specific sections highlight and provide suggestions for resolving some common formatting or issues that can make analyzing your data difficult.
In this article:

- **Pivot data in crosstab format** below
- **Remove pre-aggregated data** on the next page
- **Remove or exclude introductory text** on the next page
- **Flatten hierarchical headers to a single row** on page 328
- **Remove blank rows** on page 330
- **Add missing headers** on page 330

**Pivot data in crosstab format**

When data is formatted as in crosstab format, the table is column oriented. In a column oriented table, the variables are stored as column headers. However, Tableau Desktop is optimized for row oriented data. In a row-oriented table, the variables are stored in the row values.

For example, suppose you have column-oriented table, which shows math, science, and history scores for grade school students.

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>School</th>
<th>Math</th>
<th>Science</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M</td>
<td>West</td>
<td>90</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>South</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>M</td>
<td>Central</td>
<td>50</td>
<td>80</td>
<td>80</td>
</tr>
</tbody>
</table>

Tableau Desktop is optimized to connect to row-oriented tables, where math, science, and history values are organized under a column called "Subject" and the scores for each student are organized under a column called "Score." You can pivot the columns in the rows, by manually editing your Excel data. Alternatively, connect to your Excel data from Desktop and then use the pivot option. For more information about the pivot option, see **Pivot Data from Columns to Rows** on page 747.
Remove pre-aggregated data

Data can often come pre-aggregated. That is, data can contain sums, averages, medians, etc. A common example of pre-aggregated data comes in the form of subtotals and grand totals. Subtotals and grand totals data are computed from the raw data, but is not part of raw data itself.

For example, suppose you have a table that contains a row of subtotals information.

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>School</th>
<th>Math</th>
<th>Science</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M</td>
<td>West</td>
<td>90</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>South</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>M</td>
<td>Central</td>
<td>50</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>M</td>
<td>Central</td>
<td>100</td>
<td>90</td>
<td>80</td>
</tr>
<tr>
<td>5</td>
<td>F</td>
<td>West</td>
<td>90</td>
<td>100</td>
<td>80</td>
</tr>
</tbody>
</table>

Subtotals | 380 | 400 | 370 |

In this case, pre-aggregated data needs to be removed. To use subtotals and grand totals in your analysis, manually remove this type of information from your table. Then, connect to your Excel data from Desktop and calculate subtotals and totals using the totals option. For more information, see Show Totals in a Visualization on page 1754. Alternatively, connect to your Excel data from Desktop, turn on Data Interpreter, and then use the totals option. For more information, see Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter on page 752.

Remove or exclude introductory text

Excel data that is delivered as a report can contain titles or blocks of introductory text. Because Desktop expects either column headers or row values in the first row of a table, this information can cause problems during your analysis.

For example, suppose you have a table that contains a report title and date.
In this case, the title and date information needs to be removed. To use a title and date for a report, do one of the following:

- Manually remove this information from your Excel data. Then connect to your Excel data from Desktop and add a report title using the title option. For more information, see Format Titles, Captions, Tooltips, and Legends on page 2382.
- Connect to your Excel data from Desktop, turn on Data Interpreter, and then use the title option. For more information, see Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter on page 752.
- If you cannot remove this information from your Excel data, create a named range and connect to the named range from Desktop. For more information, see Excel on page 321.

### Excel Report

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>School</th>
<th>Math</th>
<th>Science</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M</td>
<td>West</td>
<td>90</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>South</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>M</td>
<td>Central</td>
<td>50</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>M</td>
<td>Central</td>
<td>100</td>
<td>90</td>
<td>80</td>
</tr>
<tr>
<td>5</td>
<td>F</td>
<td>West</td>
<td>90</td>
<td>100</td>
<td>80</td>
</tr>
</tbody>
</table>

**Flatten hierarchical headers to a single row**

In general, Tableau Desktop expects only the first row in your Excel data to contain column headers. Data that contains multiple layers of column headers can cause problems during your analysis.

For example, suppose you have a table that contains one major header and multiple sub-headers.
In this case, the hierarchy of headers must be flattened or removed. To do this, you can manually create a new column for each header in the hierarchy directly in your Excel data. Alternatively, connect to your Excel data from Tableau Desktop and then turn on Data Interpreter. Verify that your headers are flattened correctly. For more information about Data Interpreter, see Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter on page 752.

Make sure there are no blank cells

If you create new columns for your hierarchical headers, make sure that each cell in the new columns contains values.
While you might repeat the same value for each row, it's important that each row contains the data that associates it with the data that was stored in the hierarchical header. You must manually remove blank cells from your Excel data.

<table>
<thead>
<tr>
<th>ID</th>
<th>State</th>
<th>Gender</th>
<th>School</th>
<th>Math</th>
<th>Science</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>California</td>
<td>M</td>
<td>West</td>
<td>90</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>California</td>
<td>F</td>
<td>South</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>California</td>
<td>M</td>
<td>Central</td>
<td>50</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>California</td>
<td>M</td>
<td>Central</td>
<td>100</td>
<td>90</td>
<td>80</td>
</tr>
<tr>
<td>5</td>
<td>California</td>
<td>F</td>
<td>West</td>
<td>90</td>
<td>100</td>
<td>80</td>
</tr>
<tr>
<td>6</td>
<td>Oregon</td>
<td>M</td>
<td>North</td>
<td>70</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>7</td>
<td>Oregon</td>
<td>F</td>
<td>East</td>
<td>80</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>8</td>
<td>Oregon</td>
<td>F</td>
<td>West</td>
<td>50</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>9</td>
<td>Oregon</td>
<td>F</td>
<td>West</td>
<td>100</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>Oregon</td>
<td>M</td>
<td>West</td>
<td>80</td>
<td>80</td>
<td>90</td>
</tr>
</tbody>
</table>

Remove blank rows

Make sure that there are no blank rows in your data. To fix blank rows, you must remove the blank rows from your Excel data.

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>School</th>
<th>Math</th>
<th>Science</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M</td>
<td>West</td>
<td>90</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>South</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>M</td>
<td>Central</td>
<td>50</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>M</td>
<td>Central</td>
<td>100</td>
<td>90</td>
<td>80</td>
</tr>
<tr>
<td>5</td>
<td>F</td>
<td>West</td>
<td>90</td>
<td>100</td>
<td>80</td>
</tr>
<tr>
<td>22</td>
<td>M</td>
<td>North</td>
<td>70</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>23</td>
<td>F</td>
<td>East</td>
<td>80</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>24</td>
<td>F</td>
<td>West</td>
<td>50</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>25</td>
<td>F</td>
<td>West</td>
<td>100</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>26</td>
<td>M</td>
<td>West</td>
<td>80</td>
<td>80</td>
<td>90</td>
</tr>
</tbody>
</table>

Add missing headers

Make sure that there are no missing column headers. To fix missing headers, you must manually add the missing headers directly to your Excel data.
This article describes how to connect Tableau to text file data and set up the data source. Tableau connects to delimited text files (*.txt, *.csv, *.tab, *.tsv).

In this article

- Make the connection and set up the data source below
- Optional settings on the next page
- Best practices on page 335
- About .ttde and .hhyper files on page 335

Make the connection and set up the data source

1. After you open Tableau, under Connect, click Text File.

2. Select the file you want to connect to, and then click Open.

   Alternatively, when using Tableau Desktop on Windows, to connect to the text file using the Microsoft Jet-based connection, in the Open dialog box, click the Open drop-down menu, and then select Open with Legacy Connection. The data source page appears.

3. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to. The default name is automatically generated based on the file name.
   b. Click the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.
Note: For text files, custom SQL is available only when using the legacy connection or in workbooks that were created before Tableau Desktop 8.2.

Text file data source example

Here is an example of a text file data source:

Optional settings

You can set the following options before building the view.

Get more data

Get more data into your data source by adding more tables or connecting to data in a different database.

- **Add more data from the current file**: From the left pane, drag additional tables to the canvas to combine data using a join or union. For more information, see Join Your Data on page 657 or Union Your Data on page 708.

- **Add more data from different databases**: In the left pane, click Add next to Connections. For more information, see Join Your Data on page 657.
If a connector you want is not listed in the left pane, select Data > New Data Source to add a new data source. For more information, see Blend Your Data on page 682.

Clean your data with Data Interpreter

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter on page 752.

Set text file options

On the canvas, click the table drop-down arrow and then select whether the first row contains column names. This option is selected by default. Alternatively, you can have Tableau generate names when you connect. These names can be changed later. You can also select Text file properties to specify the following:

- Select the character that is used to separate the columns. Select from the list of characters or select Other to type in a custom character.
- Select the text qualifier that encloses values in the text file.
- Select a character set that describes the text file encoding. The available encodings are based on the operating system you are using. For example, on Windows, ANSI is listed as windows-1252 and OEM is listed as 437.

  Note: In workbooks created prior to Tableau Desktop 8.2 or that use the legacy connection, you can select ANSI, OEM, UTF-8, UTF-16, or Other. When you select Other, you must specify the character set in the provided text field. This value is verified when the connection is attempted.

- Select the locale by which the file should be parsed. This option tells Tableau which decimal and thousands separator to use.

Review the data, pivot, split, and create calculations

The first 1,000 rows of the data in the data source are automatically displayed below the canvas in the data grid. If you add tables, remove tables, or make changes to the join conditions, the data grid updates with your changes. You can also do the following in the data grid:
- Change the data type or geographical role of a column by clicking the data type icon.
- Hide a field by clicking the column drop-down arrow and selecting Hide.
- Rename a field by double-clicking the field name.
- Reset a field name by clicking the column drop-down arrow and selecting Reset Name.
- Sort fields in both the data grid and metadata grid by selecting a sort option from the Sort fields drop-down list.
- Sort rows in the data grid by clicking the sort button next to the column name.
- Pivot fields to transform data in a crosstab format into a columnar format. This is only available for non-legacy connection types. For more information, see Pivot Data from Columns to Rows on page 747.
- Split a string field into multiple fields. This is only available for non-legacy connection types. For more information, see Split a Field into Multiple Fields on page 761.
- Create new calculations based on existing field in the Tableau data source. Click the column drop-down arrow and select Create Calculated Field.
- Copy values in the grid by selecting the values and then pressing Ctrl+C (Command-C on a Mac). Alternatively, to copy values in the metadata grid, select the values, right-click (Control-click on a Mac), and then select Copy.

Examine the data source structure and perform management tasks

Use the metadata area to quickly examine the general structure of the Tableau data source and its fields. If you are working with a particularly large data source, use the metadata area to perform routine data management tasks such as hide multiple fields at once or quickly rename or reset fields.

Connect live or use an extract

At the top of the Data Source page, select Live or Extract mode. If you choose to create an extract, the Edit link displays. Click Edit to set up filters that define a subset of the data that you want to include in the extract. For more information, see Extract Your Data on page 773.

Add data source filters

At the top of the data source page, click Add to add data source filters to control what data is used in included in the data source and therefore restrict the visibility and use of the fields in the data source.
Best practices

Collect files in a single directory

Collect all relevant text files for a multi-table connection in a single directory, with nothing else in that directory, so that users cannot inadvertently select a file that is not appropriate for the connection.

Text file column width and size considerations

If the text file contains columns that are more than 254 characters wide, Tableau cannot use these fields for workbooks that were created before Tableau Desktop 8.2 or that use the legacy connection. Either remove the columns, modify them to fit within 254 characters prior to connecting in Tableau, or upgrade the text file data source.

Large text files often perform poorly as a data source, because the queries can take a long time.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Access

This article describes how to connect Tableau to a Microsoft Access file (*.mdb, *.accdb) and set up the data source. Tableau supports all Access data types except OLE Object and Hyperlink.
In this article

**Before you begin below**

**Make the connection and set up the data source below**

Before you begin

Before you begin, gather this connection information:

- Access file name.
- If the file is password protected, you need the database password.
- If the file has workgroup security, you need workgroup security credentials:
  - Workgroup file name
  - User
  - Password

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Access**, select the Access file that you want to connect to, and then select **Sign In**.

   - Password protected - If the Access file is password protected, select **Database Password**, and then enter the password.
   - Workgroup security - If the Access file is protected by workgroup security, select **Workgroup Security**, and then enter the **Workgroup File name**, **User**, and **Password** into the corresponding text fields.

2. On the data source page, do the following:
a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to. The default name is automatically generated based on the file name.

b. Drag a table to the canvas. You can drag a table or query.

c. Select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Access data source example

Here is an example of an Access data source:
**Note:** If the Access file contains columns that are more than 254 characters wide, Tableau can't use these fields. Either remove the columns from the table or modify them to fit within 254 characters prior to connecting with Tableau.

See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**JSON File**

This article describes how to connect Tableau to a local JSON file and set up the data source.

In this article

- Make the connection and set up the data source below
- Select schema levels on the next page
- Change schema levels on page 341
- Union JSON files on page 341
- How dimension folders are organized for hierarchical JSON files on page 341
- Why measures are calculated in hierarchical JSON files on page 342
- Tips for working with JSON data on page 343
- About .ttde and .hhyper files on page 344

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **JSON File**. Then do the following:
   
   a. Select the file you want to connect to, and then select **Open**.
   
   b. In the Select Schema Levels dialog box, select the schema levels you want to view and analyze in Tableau, and then select **OK**. For more information, see **Select schema levels** on the next page.

2. On the data source page, do the following:
a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. Select the sheet tab to start your analysis.

JSON file data source example

Here is an example of a JSON file as a data source using Tableau Desktop on a Windows computer:

Select schema levels

When you connect Tableau to a JSON file, Tableau scans the data in the first 10,000 rows of the JSON file and infers the schema from that process. Tableau flattens the data using this inferred schema. The JSON file schema levels are listed in the Select Schema Levels dialog box.

The schema levels that you select in the dialog box determine which dimensions and measures are available for you to view and analyze in Tableau. If you select a child schema level, the parent level is also selected.

For example, here’s a snippet of a

The JSON file generates these schema levels:
Detect new fields

Sometimes, more fields exist in rows that were not scanned to create the inferred schema. If you notice that a field you need is missing under **Schema**, you can choose to do one of the following:

- Scan the entire JSON document. It may take a long time for the scan to complete.
- Select schema levels from the schema listed and then select **OK**. Tableau reads your entire document and if more fields are found, they are listed in the Select Schema Levels dialog box.
Whenever Tableau detects that new fields are available, for example, during an extract refresh or when Tableau creates an extract after you’ve selected the schema levels, either an information icon near the file name or a notification on the Select Schema Levels dialog box will indicate that additional fields have been found.

Change schema levels

You can change the schema levels you selected by going to the data source page and selecting Data > [JSON file name] > Select Schema Level. Or, hover over the file name on the canvas and select the drop-down menu > Select Schema Level.

Union JSON files

You can union JSON data. To union a JSON file, it must have a .json, .txt, or .log extension. For more information about union, see Union Your Data on page 708.

When you union JSON files, the schema is inferred from the first 10,000 rows of every file in the union.

You can change the schema levels after you union files. For more information, see Change schema levels above.

How dimension folders are organized for hierarchical JSON files

After you select the sheet tab, the selected schema levels of your JSON file show under Dimensions on the Data pane. Each folder corresponds with the schema level you selected, and the attributes associated with that schema level are listed as children of the folder.

For example, in the following image, Stars is a dimension under the schema level Stars rated folder, and Day is a dimension under the schema level Visits folder. Category is also a schema level, but because it is a list of values and not a hierarchy of data, it doesn't require its own folder, but is instead grouped under a parent folder, Example Business. Note that schema levels in the Select Schema Levels dialog box do not map directly to the folder structure in the Data pane. Folders in the Data pane are grouped by object so that you can easily navigate to fields and still have context for where the fields come from.
For each document, a unique index is generated and stored in the flattened representation of data. An index is also generated for each level in the schema.

For example, in the image above, in addition to the Document Index (generated) entry, the Category, Stars rated, and Visits schema levels all have generated indexes.

Why measures are calculated in hierarchical JSON files

When a hierarchical JSON file is flattened, data might be duplicated. To keep measures consistent with their schema levels, Tableau creates level of detail (LOD) calculations to accurately represent the data at the schema level. The original measures are located in the Source Measures folder and you can use them, but we recommend that you use the calculated measures.

On the Data pane, calculated measures are labeled as =#<measure name> per <parent name>.
To view the LOD calculation for a measure, follow these steps:

1. Select the measure.
2. Select the drop-down arrow, and then select Edit.

The following example shows the LOD calculation for Revenue per Document. The formula selects the maximum revenue for each document index value.

The use of LOD calculations means that you can select multiple schema levels and be confident that the measures are not over-counted.

Tips for working with JSON data

These tips can help you work with your JSON data in Tableau.

- Do not exceed the 10x10 limit for nested arrays.
  A high number of nested arrays creates a lot of rows. For example, 10x10 nested arrays result in 10 billion rows. When the number of rows Tableau can load into memory is exceeded, an error displays. In this case, use the Select Schema Levels dialog box to reduce the number of selected schema levels.

- A data source that contains more than 100 levels of JSON objects can take a long time to
A high number of levels creates a lot of columns, which can take a long time to process. As an example, 100 levels can take more than two minutes to load the data. As a best practice, reduce the number of schema levels to just the levels that you need for your analysis.

- A single JSON object cannot exceed 128 MB.
  When a single object top-level array exceeds 128 MB, you must convert it to a file where the JSON objects are defined one per line.

- The pivot option is not supported.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer’s directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can’t be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

PDF File

This article describes how to connect Tableau to .pdf file data and set up the data source.

In this article

Make the connection and scan your document for tables on the next page
Make the connection and scan your document for tables

1. After you open Tableau, under **Connect**, click **PDF File**.
2. Select the file you want to connect to, and then click **Open**.
3. In the Scan PDF File dialog box, specify the pages in the file that you want Tableau to scan for tables. You can choose to scan for tables in all pages, just a single page, or a range of pages.

   **Note:** The scan counts the first page of the file as page 1, similar to most PDF readers. When you scan for tables, specify the page number that the PDF reader displays and not the page number that might be used in the document itself, which may or may not start from page 1.

   For example, suppose you want to use "Table 1" from the image below. The PDF reader displays a number, and the .pdf file displays a different number. To correctly scan for this table, specify the page number that the PDF reader displays. In this example, you specify page 15.

4. On the data source page, do the following:
a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to. The default name is automatically generated based on the file name.

b. If your file contains one table, click the sheet tab to start your analysis. Otherwise, from the left pane drag a table onto the canvas and then click the sheet tab to start your analysis.

**About the tables in the left pane**

Tables that are identified in the .pdf file are given unique names and are displayed in the left pane after a scan. For example, you might see a table name like "Page 1, Table 1." The first part of the table name indicates the page in the .pdf file the table came from. The second part of the table name indicates the order the table was identified. If Tableau has identified more than one table on a page, the second part of the table name can indicate one of two things:

- Tableau has identified another unique table or sub-table on the page.
- Tableau has interpreted the table on the page in another way. Tableau might provide multiple interpretations of a table depending on how the table is presented in your .pdf file.
PDF file data source example

Here is an example of a PDF file data source:
Get more data

Get more data into your data source by adding more tables or connecting to data in a different database.

- **Add more data from the current file:**
  - From the left pane, drag additional tables to the canvas to combine data using a join or union. For more information, see *Join Your Data* on page 657 or *Union Your Data* on page 708.
  
  - If the pages that were scanned in step 3 of the procedure listed above do not produce the tables that you need in the left pane, click the drop-down arrow next to the PDF File connection, and click *Rescan PDF file*. This option allows you to create a new scan so that you can specify different pages in the .pdf file to scan for tables.

- **Add more data from a different database:** In the left pane, click *Add* next to Connections. For more information, see *Join Your Data* on page 657.
  
  If a connector you want is not listed in the left pane, select *Data > New Data Source* to add a new data source. For more information, see *Blend Your Data* on page 682.

Set table options

You can set table options. On the canvas, click the table drop-down arrow and then specify whether the data includes field names in the first row. If so, these names will become the field
names in Tableau. If field names are not included, Tableau generates them automatically. You can rename the fields later.

Use Data Interpreter to clean your data

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter on page 752.

Union tables in your .pdf files

You can union tables in your file. For more information about union, see Union Your Data on page 708.

When you use wildcard search to union tables, the result is scoped to the pages that were scanned in the initial file you connected to. For example, suppose you have three files: A.pdf, B.pdf, and C.pdf. The first file you connect to is A and you limit the scan for tables to page 1. When you use wildcard search to union tables from files B and C, the additional tables included in the union can only come from page 1 of B and page 1 of C.

Tips for working with .pdf files

The following tips can help you work with your .pdf files in Tableau.

- **Use PDF File connector to identify just the tables in your .pdf file.**

  The primary goal of the PDF File connector is to find and identify tables in your .pdf file. Therefore, it ignores any other information in the file that does not appear to be part of a table, including titles, captions, and footnotes. If related data is stored in one of these areas, such as in the table title, you can use Tableau to first export the .pdf file data into a .csv file, manually add the data that was stored in the table title, and then connect to the .csv file instead. For more information, see Export your data to .csv file on page 2478.

- **Use standard tables.**

  In general, Tableau works best with standard tables that use a tabular format. Ideally, the tables in your .pdf file have column headers on a single line and have rows values on a single line as demonstrated in the example below.
Colors and shading used in or around the tables can affect how the tables are identified.

Tables that have unique formatting might require some cleanup or manual editing outside of Tableau. Unique formatting can include hierarchical headers, header names that span multiple lines, row values that span multiple lines, angle headers, and stacked tables as demonstrated in the examples shown below.

Note: Tableau does not support connections to .pdf files generated by scanning (optical character recognition) software.

- **Validate the data.**

  Make sure that you validate the data in the tables that Tableau identifies in your .pdf file. You can validate the data by using either the data grid or if you used the Data Interpreter, the results workbook.

- **Avoid tables that span across pages.**
If your .pdf file contains a table that spans across pages, Tableau interprets that table as multiple tables. To resolve this issue, use a union to combine the tables. For more information, see Union Your Data on page 708.

- **Rename .pdf files whose file names contain unicode characters.**
  After connecting to a .pdf file that contains unicode characters in its file name, you might see the following error.

To resolve this issue, rename the file using non-unicode characters, and connect to your .pdf file again.

- **Do not use password protected .pdf files.**
  After connecting to and scanning a .pdf file for tables, you might see the following error.

Tableau shows this error when your .pdf file is password protected and unable to access its contents. Tableau is unable to support connections to password protected .pdf files.

- **Alias values that are interpreted differently or incorrectly.**
  In the data grid you might notice that some values are interpreted differently from the .pdf file. You can correct this interpretation by using aliases to rename specific values within a field.
For example, suppose you see the following table after connecting to your .pdf file. Some state abbreviations are being interpreted in lowercase form, which are highlighted in blue.

<table>
<thead>
<tr>
<th>City</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>College Station, tx</td>
<td>20</td>
</tr>
<tr>
<td>Tempe, az</td>
<td>20</td>
</tr>
<tr>
<td>Oakland, ca</td>
<td>20</td>
</tr>
<tr>
<td>Detroit, MI</td>
<td>1</td>
</tr>
<tr>
<td>Pittsburgh, PA</td>
<td>2</td>
</tr>
<tr>
<td>Cincinnati, OH</td>
<td>3</td>
</tr>
<tr>
<td>Portland, or</td>
<td>4</td>
</tr>
</tbody>
</table>

You can resolve this issue by using aliases to change the lowercase abbreviations to uppercase abbreviations. To do this, click the drop-down arrow next to the column name and select **Aliases**.

- **Resolve column headers that are interpreted as table values.**

  In the data grid you might also notice that some column headers in your .pdf file are interpreted as table values instead. This can occur if your .pdf file contains tables with unique formatting or hierarchical headers. In this scenario, try the Data Interpreter first. If Data Interpreter doesn’t resolve the issue, consider manually renaming the columns to their appropriate names and filtering header names that are being treated as values by using data source filters.

  For example, suppose you see the following table after connecting to your .pdf file. The table headers from the .pdf file are being interpreted as table values, which are highlighted in blue.
One way you can resolve a header issue like this is to follow steps similar to the following:

1. Double-click the column name, and then rename F1 to Year. Repeat this step for F2 through F4 for Coal, Gas, and Oil.

2. Click the data type icon for the Year column and change it to a number data type. This causes the non-numerical values in this column to convert to null values.

3. In the upper-right corner of the data source page, click **Add**, click the **Add** button, and then select the Year field.

4. In the Filter dialog box, select both the **Null** and **Exclude** check boxes.
The rows in the Year column that contain null values are removed from the data grid, which affect the rows from the other columns in the table.

<table>
<thead>
<tr>
<th>#</th>
<th>Abc</th>
<th>Abc</th>
<th>Abc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>Coal</td>
<td>Gas</td>
<td>Oil</td>
</tr>
<tr>
<td>1960</td>
<td>318.8</td>
<td>0.1</td>
<td>42.5</td>
</tr>
<tr>
<td>1961</td>
<td>316.5</td>
<td>0.1</td>
<td>40.2</td>
</tr>
<tr>
<td>1962</td>
<td>315.6</td>
<td>0.2</td>
<td>43.3</td>
</tr>
</tbody>
</table>

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

### Spatial File

This topic describes how to connect Tableau Desktop to Shapefiles, MapInfo tables, KML (Keyhole Markup Language) files, and GeoJSON files.

**Note:** Connecting to spatial data is supported in Tableau Desktop version 10.2 and later.

If you have Tableau Desktop version 10.1 or earlier, and would like to use shapefile data to create map views, see **Create Tableau Maps from Shapefiles** in the Tableau Desktop 10.1 Help.

**In this article**

**Before you connect** below

**Make the connection and set up the data source** on the next page

**Work with the Geometry column** on the next page

**About .ttde and .hhyper files** on page 357

**Before you connect**

Before you can connect to spatial files, make sure to include all of the following files in the same directory:

- **For Esri shapefiles:** The folder must contain a .shp, .shx, and .dbf file.
- **For MapInfo tables:** The folder must contain a .TAB, .DAT, .MAP and .ID or .MID/.MIF file.
- **For KML files:** The folder must contain the .kml file. (No other files are required.)
- **For GeoJSON files:** The folder must contain the the .geojson file. (No other files are required.)
Note: You can only connect to point geometries, linear geometries, and polygons in current versions of Tableau. You cannot connect to mixed geometry types.

Also note that if your data does not display diacritics (accent marks on characters) properly, check to make sure the file is UTF-8 encoded.

Make the connection and set up the data source

Start Tableau and under Connect, select Spatial file. Then do the following:

1. Navigate to the folder that contains your spatial data and select the spatial file you want to connect to.
2. Select Open.

Spatial file data source example

Here is an example of a spatial file data source using Tableau Desktop on a Mac computer:

Work with the Geometry column

There are many tasks that you can optionally perform on your data before you start your analysis, such as hiding or renaming fields. Note, however, the following restrictions apply when working with the Geometry column:
- You can’t hide the Geometry column.
- You can’t split the Geometry column.
- On the data source page, you can’t create a calculated field using the Geometry column.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer’s directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can’t be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

See also
Create Tableau Maps from Spatial Files on page 1959

Statistical File

This article describes how to connect Tableau to a statistical file and set up the data source. Tableau connects to SAS (*.sas7bdat), SPSS (*.sav), and R (*.rdata, *.rda) data files.

In this article

Make the connection and set up the data source on the next page
R data file object and format support on page 359
Change the character encoding for a statistical file on page 359
About .ttde and .hhyper files on page 360
Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Statistical File**, select the file that you want to connect to, and then select **Open**.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Select the sheet tab to start your analysis.

For information about connecting to more than one table, see [Join Your Data](#) on page 657.

**Statistical file data source example**

Here is an example of a statistical file data source using Tableau Desktop on a Windows computer:

![Statistical file data source example](image)

**Note the following:**

- The Statistical File connector does not support value labels.
- The Statistical File connector supports only one table per statistical file.
If an error message appears, "An error occurred while communicating with the data source," make sure that your statistical file contains only one object, and that the object is a data frame or a matrix. R files may contain hidden objects in what appears to be a clean workspace. To check for hidden objects, run `ls(environment(), all.names=TRUE)` from RStudio.

- You can change the character encoding for a statistical file. For more information, see Change the character encoding for statistical files.

R data file object and format support

R data files should contain only one of the following types of objects:

- Two-dimensional matrices
- Vectors
- Factors
- Data frames

Note that if the R data file contains more than one object, Tableau will connect to the first one.

Supported R file formats are binary or ASCII. Compressed files are not supported in Tableau Desktop. You must decompress files first with another program, such as Gzip or WinZip, before you connect with Tableau.

Change the character encoding for a statistical file

Tableau reads the data in a statistical file (for example, an SAS or R file) based on the file's character encoding or on information in an input file. R files typically use the character encoding of the operating system, compared to SAS and SPSS files, which include the character encoding information in the file. Sometimes, you might need to specify a different encoding. For example, if a colleague sends you a statistical file with Greek character encoding, then you must specify a Greek character set to use the file with Tableau. If you need to use a different character set when reading from a statistical file, you can create a Tableau data source customization (TDC) file and specify the encoding to use.

Create the TDC file

A .tdc file is an XML file that applies to a single data source and contains vendor and driver name information of the data source provider. For the statistical file connector, the vendor and driver name is stat-direct.

To create a TDC file:
1. Open a plain text editor, such as Notepad.

2. Copy the information from the sample provided below, paste it into your text file, and then specify the `source-charset` value. (For a list of encodings, see User-defined Encodings Supported by the Statistical File Connector on the next page.)

3. Save the file with a .tdc extension—for example, `r-statsfile.tdc`—to the My Tableau Repository\Datasources folder.

Sample TDC file sets the source-charset value

The following TDC file example sets the `source-charset` value to `shift-jis` for a statistical file data source.

```xml
<connection-customization class='stat-direct' enabled='true' version='10.0'>
  <vendor name='stat-direct' />
  <driver name='stat-direct' />
  <customizations>
    <customization name='source-charset' value='shift-jis' />
  </customizations>
</connection-customization>
```

Important: Tableau does not test or support TDC files. These files should be used as a tool to explore or occasionally address issues with your data connection. Creating and maintaining TDC files requires careful manual editing, and there is no support for sharing these files.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.
In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

User-defined Encodings Supported by the Statistical File Connector

This topic lists character encodings supported by Tableau Statistical File connector. The list includes single-byte, multi-byte, and Unicode user-defined encodings, as well as single-byte and multi-byte encodings that cannot currently be mapped to the corresponding SAS encodings.

---

**In this article:**

- Single-byte user-defined encodings
- Multi-byte user-defined encodings
- Unicode user-defined encodings
- Single-byte encodings that can't be mapped to SAS encodings
- Multi-byte encodings that can't be mapped to SAS encodings

---

**Single-byte user-defined encodings**

ASCII
CSASCII
US-ASCII
US
ISO_646.IRV:1991
ISO646-US
646
ISO-IR-6
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM367</td>
<td></td>
</tr>
<tr>
<td>CP367</td>
<td></td>
</tr>
<tr>
<td>ANSI_X3.4-1986</td>
<td></td>
</tr>
<tr>
<td>ANSI_X3.4-1968</td>
<td></td>
</tr>
<tr>
<td>ISO-8859-1</td>
<td></td>
</tr>
<tr>
<td>CSISOLATIN1</td>
<td></td>
</tr>
<tr>
<td>LATIN1</td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td></td>
</tr>
<tr>
<td>ISO_8859-1:1987</td>
<td></td>
</tr>
<tr>
<td>ISO8859-1</td>
<td></td>
</tr>
<tr>
<td>ISO-IR-100</td>
<td></td>
</tr>
<tr>
<td>ISO-8859-1</td>
<td></td>
</tr>
<tr>
<td>IBM819</td>
<td></td>
</tr>
<tr>
<td>CP819</td>
<td></td>
</tr>
<tr>
<td>ISO-8859-15</td>
<td></td>
</tr>
<tr>
<td>LATIN-9</td>
<td></td>
</tr>
<tr>
<td>ISO_8859-15</td>
<td></td>
</tr>
<tr>
<td>ISO8859-15</td>
<td></td>
</tr>
<tr>
<td>ISO-IR-203</td>
<td></td>
</tr>
<tr>
<td>IBM850</td>
<td></td>
</tr>
<tr>
<td>CSPC850MULTILINGUAL</td>
<td></td>
</tr>
<tr>
<td>CP850</td>
<td></td>
</tr>
<tr>
<td>850</td>
<td></td>
</tr>
<tr>
<td>WINDOWS-1252</td>
<td></td>
</tr>
<tr>
<td>MS-ANSI</td>
<td></td>
</tr>
<tr>
<td>CP1252</td>
<td></td>
</tr>
</tbody>
</table>
ISO-8859-7
CSISOLATINGGREEKISO_8859-7:1987
ISO_8859-7
ISO-IR-126
ISO-8859-7
GREEK8
GREEK
ELOT_928
ECMA-118
WINDOWS-1253
MS-GREEK
CP1253
ISO-8859-10
CSISOLATIN6
LATIN6
L6
ISO_8859-10:1992
ISO_8859-10
ISO8859-10
ISO-IR-157
WINDOWS-1257
WINBALTRIM
CP1257
ISO-8859-2
CSISOLATIN2
LATIN2
L2
ISO_8859-2:1987
ISO_8859-2
ISO8859-2
ISO-IR-101
IBM852
CSPCP852
CP852
852
WINDOWS-1250
MS-EE
CP1250
ISO-8859-5
CSISOLATINCYRILLIC
ISO_8859-5:1988
ISO_8859-5
ISO8859-5
ISO-IR-144
CYRILLIC
WINDOWS-1251
MS-CYRL
CP1251
CP866
CSIBM866
IBM866
866
TIS-620
TIS620.2533-1
TIS620.2533-0
TIS620.2529-1
TIS620-0
TIS620
ISO-IR-166
ISO-8859-11
CP874
CSISOLATIN5
LATIN5
L5
ISO_8859-9:1989
ISO_8859-9
ISO8859-9
ISO-8859-9
ISO-IR-148
CSIBM857
IBM857
CP857
857
WINDOWS-1254
MS-TURK
CP1254
CP1129
VPS
WINDOWS-1258
CP1258
ISO-8959-6
CSISOLATINARABIC
ISO_8859-6:1987
ISO_8859-6
ISO8859-6
ISO-IR-127
ECMA-114
ASMO-708
ARABIC
WINDOWS-1256
MS-ARAB
CP1256
ISO-8859-8
CSISOLATINHEBREW
ISO_8859-8:1988
ISO_8859-8
ISO8859-8
ISO-IR-138
HEBREW
IBM864
CSIBM864
CP864
WINDOWS-1255
MS-HEBR
CP1255
IBM862
CSPC862LATINHEBREW0x2E
CP862
862

Multi-byte user-defined encodings
CP936
WINDOWS-936
MS936
GBK
GB2312
CSISO58GB231280
ISO-IR-58
GB_2312-80
CHINESE
ISO-2022-CN
CP950
windows-950
ms-950
ms950
CSBIG5
CN-BIG5
BIGFIVE BIG5
BIG-FIVE
BIG-5
BIG5HKSCS
BIG5-HKSCS
EUC-TW
CSEUCTW
EUCTW
EUC-JP
CSEUCPKDFMTJAPANESE
EXTENDED_UNIX_CODE_PACKED_FORMAT_FOR_JAPANESE
EUCJP
EUC-JP
ISO-2022-JP
CSISO2022JP
ISO-2022-JP
CSSHIFTJIS
SJIS
SHIFT_JIS
SHIFT-JIS
MS_KANJII
CP932
EUC-KR
CSEUCKR
EUCKR
EUC-KR
UHC
CP949
EUC-CN
CSGB2312
GB2312
EUCCN
CN-GB
Unicode user-defined encodings
TF-8
UCS-2
UCS-2BE
UCS-2LE
UCS-4
UCS-4BE
UTF-16
UTF-16BE
UTF-16LE
UTF-32
UTF-32LE
UTF-32BE
UTF-7

Single-byte encodings that can't be mapped to SAS encodings

MACROMAN
CSMACINTOSH
MACINTOSH
MAC
ISO-8859-14
LATIN8
L8
ISO_8859-14:1998
ISO_8859-14
ISO8859-14
ISO-IR-199
ISO-CELTIC
MACGREEK
MACICELAND
ISO-8859-3
CSISOLATIN3
LATIN3
L3
ISO_8859-3:1988
ISO_8859-3
ISO8859-3
ISO-IR-109
ISO-8959-4
CSISOLATIN4
LATIN4
L4
ISO_8859-4:1988
ISO_8859-4
ISO8859-4
ISO-IR-110
ISO-8959-13
LATIN7
L7
ISO_8859-13
ISO8859-13
ISO-IR-179
ISO-8859-13
MACCENTRALEUROPE
MACCROATIAN
IBM855
CSIBM855
CP855
855
KOI8-R
CSKOI8R
MACCYRILLIC
KOI8-U
CSKOI8R
MACUKRAINIAN
ISO-8859-16
LATIN10
L10
ISO_8859-16:2001
ISO_8859-16
ISO8859-16
ISO-IR-226
MACROMANIAN
ARMSCII-8
GEORGIAN-ACADEMY
MACTURKISH
TCVN
VISCII
CSVISCII
VISCII1.1-1
MACARABIC
MACHEBREW
WINDOWS-874

Multi-byte encodings that can't be mapped to SAS encodings
GB18030
Other Files

This topic describes how to connect Tableau to supported file types, including Tableau Data Extract files and Tableau Workbooks.

1. Start Tableau and under Connect, select Other Files.

2. In the Open dialog box, navigate to and select a file.

3. Select Open.

4. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to
connect to.

5. Select the sheet tab to start your analysis.

Tableau Data Extract data source example

Here is an example of a Tableau Data Extract file data source using Tableau Desktop on a Windows computer:

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Connect to a Published Data Source on Tableau Online or Tableau Server

You can connect to published data sources from either Tableau Desktop or a web browser. The process differs slightly for each.
In this article

**Connect to a published data source from Tableau Desktop below**

**Connect to published data sources while web editing** on page 376

Connect to a published data source from Tableau Desktop

1. Start Tableau Desktop, and under **Connect**, select **Tableau Server**.
2. To connect to Tableau Online, select **Tableau Online** under **Quick Connect**.
   
   To connect to Tableau Server, enter the name of the server and then select **Connect**.

   ![Tableau Server Sign In](image)

   **Tip**: If you don't sign out, Tableau Desktop saves your server connection, so you can skip step 3. You can also skip it if Kerberos is enabled on Tableau Server and your computer has valid credentials. For more information, see **Automatically keep Tableau Desktop connected to Tableau Server or Online** on page 313.

3. To sign in:
   
   - For Tableau Server, enter your user name and password.
   - For Tableau Online, enter your email address and password.

4. Select a data source from the list of published data sources.
You can sort the list of data sources by selecting the column headers. Alternatively, search for a data source by using the search box. Select the refresh button to refresh the list and show any new data sources.

![Data Sources Table](image)

**Note:** If you select a cube (multidimensional) data source, the Create Local Copy dialog box appears, and you must create a local copy of the data before you can start your analysis.

5. Select the sheet tab to start your analysis. Tableau Server or Tableau Online data sources are shown in the **Data** pane with a Tableau icon.
You may want to download a local copy of the data source so that you can, for example, work offline or make changes to a data source without modifying the original. To download a local copy, on the Data menu, select the data source, and then select Create Local Copy. A duplicate of the data source is added to the Data pane.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

Connect to published data sources while web editing

Tableau Desktop users who create and customize data connections for Tableau use can publish their data sources to Tableau Server or Tableau Online. Publishing enables sharing data among colleagues, including those who don't use Tableau Desktop, but have permission to edit workbooks in the web editing environment.

The steps below describe how to connect to published data sources when you’re signed in to Tableau Server or Tableau Online. For details on how to edit workbooks and view on the web, see Build Views on the Web and Using Tableau on the Web.
Note: To connect to data, you need permission to create and edit views.

1. On a Tableau Server or Tableau Online site, select a view to edit.

2. In editing mode, click the New Data Source icon.

3. In the Connect to Data Source dialog box, select a published data source from the list, and then click Add.

You can also connect to a published data source when creating a new workbook:

1. On a Tableau Server or Tableau Online site, navigate to the Content page, and select Data Sources.

2. In the list of data sources, select the check box next to the one you want to use, and then click Actions and select New Workbook.

Actian Matrix

This article describes how to connect Tableau to an Actian Matrix (formerly ParAccel) database and set up the data source.

In this article

Before you begin below
Make the connection and set up the data source on the next page

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.
Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Actian Matrix. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database and the name of the database that you want to connect to.
   b. Enter the user name and password, and then select Sign In.
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Schema drop-down list, select a schema.
   c. Under Table, select a table or use the text box to search for a table by name.
   d. Drag a table to the canvas, and then select the sheet tab to start your analysis.
      Use custom SQL to connect to a specific query rather than the entire data source.
      For more information, see Connect to a Custom SQL Query on page 723.

Actian Matrix data source example

Here is an example of an Actian Matrix data source:
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**Actian Vector**

This article describes how to connect Tableau to an Actian Vector database and set up the data source.
In this article

Before you begin below
Make the connection and set up the data source below

Before you begin

Before you begin, gather this connection information:

- Virtual node name for the database you want to connect to
- Database name
- Authentication method: Authentication defined in the virtual node, or user name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Actian Vector. For a complete list of data connections, select More under To a Server. Then do the following:
   
   a. Enter the name of the virtual node for the database and the name of the database you want to connect to.

   b. Specify whether to use authentication defined in the virtual node, or a specific user name and password.

   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.

   d. Select Sign In.
If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Schema drop-down list, select a schema or use the text box to search for a schema by name.
   c. Under Table, select a table or use the text box to search for a table by name.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source.
For more information, see Connect to a Custom SQL Query on page 723.

Actian Vector data source example

Here is an example of an Actian Vector data source:
See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Amazon Athena

This article describes how to connect Tableau to Amazon Athena data and set up the data source.
In this article

Before you begin

Before you begin below

Make the connection and set up the data source below

Customize JDBC connections on page 385

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Name of the S3 staging directory, for example, s3://aws-athena-query-results-123456785678-us-eastexample-2/
- Amazon Web Services (AWS) access keys (access key ID and secret access key). For more information, see Access keys on the AWS website.
- (Optional) Initial SQL statement to run every time Tableau connects

You must have Java installed on the computer that runs Tableau Desktop. For more information, see Driver Download on the Tableau website.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Amazon Athena. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server.
   b. Enter the name of the S3 staging directory.
   c. Enter your AWS access key ID in the Username field.
   d. Enter your AWS secret access key in the Password field.
   e. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of
every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.

f. Select **Sign In.**

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. Select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

**Amazon Athena data source example**

Here is an example of an Amazon Athena data source using Tableau Desktop on a Windows computer:
Customize JDBC connections

Amazon Athena uses a JDBC connection, which you can customize using a properties file. For more information, see Customize JDBC Connections Using a Properties File in Tableau Community.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Connect to your S3 data with the Amazon Athena connector in Tableau 10.3 (and later) - Tableau blog post with links to Amazon Athena resources.
Amazon Aurora

This article describes how to connect Tableau to an Amazon Aurora database and set up the data source.

In this article

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Amazon Aurora. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the user name and password.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.
Select the **Require SSL** check box when connecting to an SSL server.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the **Database** drop-down list, select a database or use the text box to search for a database by name.
   c. Under **Table**, select a table or use the text box to search for a table by name.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 723.

Amazon Aurora data source example

Here is an example of an Amazon Aurora data source using Tableau Desktop on a Windows computer:
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.
Amazon EMR Hadoop Hive

This article describes how to connect Tableau to an Amazon EMR (Elastic MapReduce) Hadoop Hive database and set up the data source.

**Note:** Starting in version 2018.2, Tableau supports Amazon EMR Hadoop Hive only, not Impala. Amazon no longer provides Impala drivers.

In this article

- **Before you begin below**
- **Make the connection and set up the data source on the next page**
- **Sign in on a Mac on page 392**
- **Work with Hadoop Hive data on page 393**

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to and port number
- Authentication method:
  - No Authentication
  - Kerberos
  - Username
  - Username and password
  - Microsoft Azure HDInsight Service (starting in version 10.2.1)
- Transport options depend on the authentication method you choose and can include the following:
  - Binary
  - SASL
  - HTTP
- Sign-in credentials depend on the authentication method you choose and can include the following:
- User name
- Password
- Realm
- Host FQDN
- Service Name
- HTTP Path

- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

**Note:** Make sure you use the latest available drivers. To get the latest drivers, see Amazon EMR Hadoop Hive on the Tableau Driver Download page.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Amazon EMR Hadoop Hive**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database and the port number to use.
   b. In the **Authentication** drop-down list, select the authentication method to use. The information you are prompted to provide depends on the authentication method you choose.
   c. If the **Transport** drop-down list is available, select the type of transport to use.
   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   e. Select **Sign In**.
Select the **Require SSL** option when connecting to an SSL server.

If Tableau can't make the connection, verify that your credentials are correct. If you still can’t connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   
a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.

c. In the **Table** text box, select the search icon or enter the table name and select the search icon, and then select the table.

d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see [Connect to a Custom SQL Query](#) on page 723.

**Amazon EMR Hadoop Hive data source example**

Here is an example of an Amazon EMR Hadoop Hive data source connecting to server using Tableau Desktop on a Windows computer:
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.
Work with Hadoop Hive data

Work with date/time data

Tableau supports TIMESTAMP and DATE types natively. However, if you store date/time data as a string in Hive, be sure to store it in ISO format (YYYY-MM-DD). You can create a calculated field that uses the DATEPARSE or DATE function to convert a string to a date/time format. Use DATEPARSE() when working with an extract, otherwise use DATE(). For more information, see **Date Functions** on page 1298.

For more information about Hive data types, see **Dates** on the Apache Hive website.

NULL value returned

A NULL value is returned when you open a workbook in Tableau 9.0.1 and later and 8.3.5 and later 8.3.x releases that was created in an earlier version and has date/time data stored as a string in a format that Hive doesn't support. To resolve this issue, change the field type back to **String** and create a calculated field using DATEPARSE() or DATE() to convert the date. Use DATEPARSE() when working with an extract, otherwise use the DATE() function.

High latency limitation

Hive is a batch-oriented system and is not yet capable of answering simple queries with very quick turnaround. This limitation can make it difficult to explore a new data set or experiment with calculated fields. Some of the newer SQL-on-Hadoop technologies (for example, Cloudera’s Impala and Hortonworks’ Stringer project) are designed to address this limitation.

See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

Amazon Redshifft

This article describes how to connect Tableau to an Amazon Redshift database and set up the data source.
In this article

Before you begin below
Make the connection and set up the data source below
Sign in on a Mac on page 396

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Amazon Redshift. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database and the name of the database you want to connect to.
   b. Enter the user name and password.
      Select the Require SSL check box when connecting to an SSL server.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.
If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   
a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the Schema drop-down list, select a schema or use the text box to search for a schema by name.

c. Under Table, select a table or use the text box to search for a table by name.

d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Amazon Redshift data source example

Here is an example of an Amazon Redshift data source using Tableau Desktop on a Windows computer:
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.
Anaplan

This article describes how to connect Tableau to Anaplan data and set up the data source.

In this article

- Before you begin below
- Make the connection and set up the data source below
- Date range selections can impact performance on page 399
- Groom your data on page 400
- Refresh your data on page 401

Before you begin

Before you begin, gather this information for connecting:

- One of the following Anaplan credentials:
  - An email address and password for your Anaplan account.
  - A certificate that Anaplan generates for you. To create a certificate, see the Certificates tab on the Anaplan website.

- The Anaplan workspace, model, and any exports (in CSV format) that you want to use. An Anaplan export action must be created in Anaplan before you can use the export in Tableau. For more information about creating a module and export action in Anaplan, see Tableau Connector for Anaplan on the Anaplan Community website.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Anaplan. For a complete list of data connections, select More under To a Server. Then do the following:
a. Select how you want to sign in to Anaplan:
   - **Credentials** – Enter your email address and password.
   - **Certificate** – Select **Upload** to upload your certificate and then follow the prompts to sign in.

For single sign-on (SSO) support, ask your Anaplan Administrator to set up single sign-on. For more information, see **Single Sign-on (SSO)** on the Anaplan Community website. Note that if SSO is enabled, you must be an exception user or use a certificate to authenticate.

b. Select a workspace, model, and one or more exports.
   **Note:** The export data must be in CSV format.

c. Select **Connect**.

   If Tableau can't make the connection, verify that your credentials or certificate are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network or Anaplan administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Under **Table**, select a table and drag it to the top of the canvas. If you have only one table, it appears on the canvas.

   c. Select the sheet tab to go to the worksheet.

      After you select the sheet tab, Tableau imports the data by creating an extract.

      Creating extracts may take some time depending on the amount of data that is included.

**Anaplan data source example**

Here is an example of an Anaplan data source using Tableau Desktop on a Windows computer:
Date range selections can impact performance

It’s tempting to gather as much data as possible when you do an analysis, however, retrieving records from Anaplan can be time-consuming. Tableau doesn’t know how much data there is in a particular date range until it retrieves the data. For this reason, you should restrict your date range at first, and then expand after you evaluate performance.
To give you a rough idea of how much time it might take to retrieve data from Anaplan, tests were conducted using a high-speed connection. This table shows how long it took in the test environment to retrieve a given number of records.

<table>
<thead>
<tr>
<th>Number of records</th>
<th>Time to retrieve</th>
</tr>
</thead>
<tbody>
<tr>
<td>100,000</td>
<td>1.7 minutes</td>
</tr>
<tr>
<td>250,000</td>
<td>4 minutes</td>
</tr>
<tr>
<td>500,000</td>
<td>8.5 minutes</td>
</tr>
</tbody>
</table>

Groom your data

All data from Anaplan is returned as type **String**. To groom your data, convert the fields to the proper data type, for example, **Number** or **Date**. You can also change **Geographic Role**, and you can convert a dimension to a measure. For information about changing the data type, see **Data Types** on page 263.

To convert a dimension to a measure, in the **Data** pane on the worksheet, select the drop-down arrow next to the field name and select **Convert to Measure**.
For information about dimensions and measures, see *Dimensions and Measures, Blue and Green* on page 250.

### Refresh your data

Tableau supports only extract connections for Anaplan, not live connections. Refreshing extracts on Tableau Server that were saved using certificate sign-on is not supported. You can update the data by refreshing the extract; incremental refreshes are not supported. For more information, see [Refresh Extracts](#) on page 798. For information about refresh schedules, see [Schedule Extract Refreshes as You Publish a Workbook](#) on page 2540.

**Note:** When there is a scheduled refresh of the Tableau extract on Tableau Server or Tableau Online, an export action is called in Anaplan. The export action creates a blocking operation that locks the model in Anaplan.

### See also

- [Set Up Data Sources](#) on page 646 – Add more data to this data source or prepare your data before you analyze it.
- [Build Charts and Analyze Data](#) on page 843 – Begin your data analysis.

### Apache Drill

This article describes how to connect Tableau to Apache Drill data and set up the data source.

### In this article

#### Before you begin below

- [Make the connection and set up the data source](#) on the next page
- [Sign in on a Mac](#) on page 403

#### Before you begin

Before you begin, gather this connection information:
- Type of connection:
  - Direct: server name
  - Zookeeper: quorum and cluster ID
- Authentication:
  - No authentication
  - User name and password

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Apache Drill. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Select the Connection method:
      - Direct - Enter the Server name.
      - Zookeeper - Enter the Quorum and Cluster ID.
   b. Select the Authentication method:
      - No Authentication
      - Username and password - Enter the user name and password.
   c. Select Sign In.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under Schema, select a schema from the drop-down list.
   c. Under Table, select a table and drag it to the canvas.
For information about connecting to more than one table, see Join Your Data on page 657.

d. Select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Apache Drill data source example

Here is an example of an Apache Drill data source using Tableau Desktop on a Windows computer:

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.
Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences > Network > Advanced**, and then open the **DNS** tab.

See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**Aster Database**

This article describes how to connect Tableau to Aster Database data and set up the data source.

In this article

**Before you begin**

**Make the connection and set up the data source** on the next page

**Sign in on a Mac** on page 406

**Use initial SQL to improve performance** on page 406

**Before you begin**

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

**Driver required**

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the **Driver Download** page where you can find driver links and installation instructions.
Make the connection and set up the data source

1. Start Tableau and under Connect, select Aster Database. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database and the name of the database that you want to connect to.
   b. Enter the user name and password.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Schema drop-down list, select a schema or use the text box to search for a schema by name.
   c. Under Table, select a table or use the text box to search for a table by name.
   d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Aster Database data source example

Here is an example of an Aster Database data source:
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

Use initial SQL to improve performance

In Aster, initial SQL can be used to generate an output table, which can improve performance of subsequent database access. Initial SQL in Aster also supports SQL-MapReduce, a framework created by Teradata Aster to enable developers to write powerful and highly expressive SQL-MapReduce functions. For more information, see SQL/MapReduce: A Practical Approach, a whitepaper on the Teradata website.
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**Box**

This article describes how to connect Tableau to Box data and set up the data source.

In this article

- **Before you begin** below
- **Make the connection and set up the data source** below
- **Use Data Interpreter to clean your data** on page 410
- **Limitations and known issues** on page 410

**Before you begin**

Before you begin, get the email address and password for your Box account.

**Make the connection and set up the data source**

1. On the start page, under **Connect**, click **Box**. For a complete list of connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:
   a. Enter your email and password and then click **Authorize**.
   b. Click **Grant access to Box**.
   c. Close the browser window when notified to do so.
   d. Search for or select the file to connect to, and then click **Connect**.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out
which data source to connect to.

b. Click the sheet tab to start your analysis.

Box data source example

Here is an example of a Box data source using Tableau Desktop on a Windows computer:
Use Data Interpreter to clean your data

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter on page 752.

Adding Tableau to your Box account

You can officially enable Tableau as an approved app for your Box account. For information on enabling this feature, please see Adding Apps to Your Box Account on the Box website.

Limitations and known issues

The Box connector limits connecting to a single Excel, JSON, or text-based file.

Known issues

- If you click Grant Access to Box repeatedly during the authentication phase an error results.
- Long file names cause the column to be pushed out of alignment.
- The file window does not resize.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Cloudera Hadoop

This article describes how to connect Tableau to a Cloudera Hadoop database and set up the data source.

In this article

Before you begin on the next page
Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to and port number
- Type of database: Hive Server 2 or Impala
- Authentication method:
  - No Authentication
  - Kerberos
  - User Name
  - User Name and Password
  - Microsoft Azure HDInsight Service (starting in version 10.2.1)
- Transport options depend on the authentication method you choose and can include the following:
  - Binary
  - SASL
  - HTTP
- Sign-in credentials depend on the authentication method you choose and can include the following:
  - User name
  - Password
  - Realm
  - Host FQDN
  - Service name
  - HTTP path
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

**Note:** Make sure you use the latest available drivers. To get the latest drivers, see Cloudera Hadoop on the Tableau Driver Download page.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Cloudera Hadoop**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database and the port number to use. If you are connecting using Cloudera Impala, you must use port 21050; this is the default port if you are using the 2.5.x driver (recommended).
   b. In the **Type** drop-down list, select the type of database to connect to. Depending on the version of Hadoop and the drivers you have installed, you can connect to one of the following:
      - **Hive Server 2**
      - **Impala**
   c. In the **Authentication** drop-down list, select the authentication method to use.
   d. Enter the information that you are prompted to provide. The information you are prompted for depends on the authentication method you choose.
   e. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   f. Select **Sign In**.
      
      Select the **Require SSL** option when connecting to an SSL server.
If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Schema drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.
   c. In the Table text box, select the search icon or enter the table name and select the search icon, and then select the table.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Note: This database type supports only equal (=) join operations.

Cloudera Hadoop data source example

Here is an example of a Cloudera Hadoop data source using Tableau Desktop on a Windows computer:
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.
Work with Hadoop Hive data

Work with date/time data

Tableau supports TIMESTAMP and DATE types natively. However, if you store date/time data as a string in Hive, be sure to store it in ISO format (YYYY-MM-DD). You can create a calculated field that uses the DATEPARSE or DATE function to convert a string to a date/time format. Use DATEPARSE() when working with an extract, otherwise use DATE(). For more information, see Date Functions on page 1298.

For more information about Hive data types, see Dates on the Apache Hive website.

NULL value returned

A NULL value is returned when you open a workbook in Tableau 9.0.1 and later and 8.3.5 and later 8.3.x releases that was created in an earlier version and has date/time data stored as a string in a format that Hive doesn't support. To resolve this issue, change the field type back to String and create a calculated field using DATEPARSE() or DATE() to convert the date. Use DATEPARSE() when working with an extract, otherwise use the DATE() function.

High latency limitation

Hive is a batch-oriented system and is not yet capable of answering simple queries with very quick turnaround. This limitation can make it difficult to explore a new data set or experiment with calculated fields. Some of the newer SQL-on-Hadoop technologies (for example, Cloudera’s Impala and Hortonworks' Stringer project) are designed to address this limitation.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Denodo

This article describes how to connect Tableau to Denodo and set up the data source.
In this article

**Before you begin below**
**Make the connection and set up the data source below**

**Before you begin**

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Name of the database
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

**Driver required**

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

**Update 20170515 of Denodo 6.0 required**

If you can’t connect Tableau to your Denodo data, you might need to ask your database administrator to apply update 20170515 to the Denodo 6.0 database. For more information about update 20170515 for Denodo 6.0, see the Denodo website. (Sign in required.)

**Make the connection and set up the data source**

1. Start Tableau and under **Connect**, select **Denodo**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the name of the database.
   c. Enter your user name and password.

   Select the **Require SSL** check box when connecting to an SSL server.

   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of
every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.

e. Select Sign In.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Under Table, select a table or use the text box to search for a table by name.

   c. Click the sheet tab to start your analysis.

Denodo data source example

Here is an example of a Denodo data source using Tableau Desktop on a Windows computer:
See also

**Set Up Data Sources on page 646** – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data on page 843** – Begin your data analysis.

**Dropbox**

This article describes how to connect Tableau to Dropbox data and set up the data source.
In this article

**Before you begin below**

**Make the connection and set up the data source below**

**Use Data Interpreter to clean your data on page 421**

Before you begin

Before you begin, get the email address and password for your Dropbox account.

**Make the connection and set up the data source**

1. Start Tableau and under **Connect**, select **Dropbox**. For a complete list of data connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:

   a. Enter your email and password and then select **Sign in**.

   b. Select **Allow** to let Tableau Desktop access your Dropbox files and folders.

   c. Close the browser window when notified to do so.

   d. Search for or select the file to connect to, and then select **Connect**.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Select the sheet tab to start your analysis.

**Note:** Tableau doesn't support pivot tables in Dropbox.

**Dropbox data source example**

Here is an example of a Dropbox data source using Tableau Desktop on a Windows computer:
Use Data Interpreter to clean your data

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter on page 752.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

EXASOL

This article describes how to connect Tableau to data stored in the EXASOL platform and set up the data source. Tableau can connect to EXASOL version 4.2 and later.

In this article

Before you begin below
Make the connection and set up the data source on the next page
Sign in on a Mac on page 423

Before you begin

Before you begin, gather this connection information:

- Name of the server you want to connect to
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau
displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select EXASOL. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that you want to connect to.
   b. Enter the user name and password.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Schema drop-down list, select a schema or use the text box to search for a schema by name.
   c. Under Table, select a table or use the text box to search for a table by name.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

EXASOL data source example

Here is an example of an EXASOL data source using Tableau Desktop on a Windows computer:
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.
Firebird

This article describes how to connect Tableau to a Firebird database and set up the data source.

In this article

Before you begin below
Make the connection and set up the data source below
Sign in on a Mac on the next page
Upgrade a Firebird extract on page 426

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Location of the database
- User name and password

Make the connection and set up the data source

1. Start Tableau and under Connect, select Firebird. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the database, or browse to the location of the database.
   c. Enter the user name and password, and then select Sign In.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Select a table, drag it to the canvas, and then select the sheet tab to start your
analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Firebird data source example

Here is an example of a Firebird data source using Tableau Desktop on a Windows computer.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.
Upgrade a Firebird extract

Beginning with version 10.5, Tableau has changed the format of extracts to the .hyper format. This format change requires Firebird extracts be opened and saved using an earlier version of Tableau before it can either be used with the current version of Tableau or be upgraded to the .hyper format. When you open and save the Firebird extract in an earlier version of Tableau, the Firebird extract is upgraded to a .tde format. To upgrade a Firebird extract, follow the steps listed below.

1. Get an earlier version of Tableau Desktop. To get an earlier version of Tableau Desktop, see the Tableau Alternate Downloads Site page.
2. Install the earlier version of Tableau Desktop, and then use it to open the Firebird extract.
3. Save the extract.
4. Open the upgraded Firebird extract using the current version of Tableau Desktop. The extract should work as expected.

**Note:** You can upgrade your extract from a .tde to a .hyper format by selecting **Data > Extract > Upgrade**. For more information, see Extract Upgrade to .hyper Format on page 788.

See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

Google Analytics

This article describes how to connect Tableau to Google Analytics (GA) and set up the data source.

In this article

**Before you begin** on the next page
Before you begin

Before you begin, gather this connection information:

- GA email address and password

Make the connection and set up the data source

1. Start Tableau and under Connect, select Google Analytics. For a complete list of data connections, select More under To a Server. In the tab Tableau opens in your default browser, do the following:
   a. Sign in to GA using your email or phone, and then select Next to enter your password. If multiple accounts are listed, select the account that has the GA data you want to access, and enter the password, if you are not already signed in.
b. Select **Allow** so that Tableau Desktop can access your GA data.

c. Close the browser window when notified to do so.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Follow the steps at the top of the data source page to complete the connection.

      **Step 1** – Select an Account, Property, and Profile using the drop-down menus.

      **Step 2** – Select filters for a date range and a segment.
For **Date Range**, you can select one of the predefined date ranges or select specific dates. When selecting a date range, GA can provide complete data only up to the previous full day. For example, if you choose Last 30 days, data will be retrieved for the last 30-day period ending yesterday.

For **Segment**, select a segment to filter your data. Segments are preset filters that you can set for a GA connection. Default Segments are defined by Google, and Custom Segments are defined by the user on the GA website. Segments also help prevent sampling by filtering the data as defined by the segment. For example, with a segment, you can get results for a specific platform, such as tablets, or for a particular search engine, such as Google.

**Note:** GA restricts the amount of data that it returns in a query. When you try to retrieve more data than GA allows in a single query, GA returns sampled data instead. If Tableau detects that your GA query might return sampled data, Tableau attempts to bypass the query restriction to return all data instead. For more information, see All data vs. sampled data returned from a query on the next page below.

**Step 3** – Add dimensions and measures by using the **Add Dimension** and **Add Measure** drop-down menus, or select a predefined set of measures from the **Choose a Measure Group** drop-down menu. Some dimensions and measures cannot be used together. For more information, see Dimensions & Metrics Reference Guide on the Google developer website.

c. Select the sheet tab to start your analysis. After you select the sheet tab, Tableau imports the data by creating an extract. Note that Tableau Desktop supports only extracts for Google Analytics. You can update the data by refreshing the extract. For more information, see Extract Your Data on page 773.

**Google Analytics data source example**

Here is an example of a Google Analytics data source connection using Tableau Desktop on a Windows computer:
All data vs. sampled data returned from a query

GA restricts the amount of data that it returns from a query and provides sampled data instead. Sampled data is a random subset of your data. When performing analysis on sampled data, you can miss interesting outliers, and aggregations can be inaccurate. If Tableau detects that your query might return sampled data, by default, Tableau creates multiple queries from your query, and then combines the results from the queries to return all data.

You see the following message when Tableau returns all data.

Query returns: All data. Sample data

If the query stays within the boundaries of the query restrictions, GA doesn’t return sampled data and you do not see the above message.

Troubleshoot issues with returning all data

If your query continues to return sampled data, consider the following:

- **Missing date dimension** – You must use the date dimension in your query to return all data.
- **Too much data** – Your query might contain too much data. Reduce the date range.
Note that the minimum date range is one day.

- **Non-aggregatable dimensions and measures** – Some dimensions and measures cannot be separated into multiple queries. If you suspect a problematic dimension or measure in your query, hover over the **All data** button to see the tooltip that shows which dimensions or measures to remove from your query.

- **Legacy workbooks** – Workbooks created in Tableau Desktop 9.1 and earlier cannot return all data. Open the legacy workbook in Tableau Desktop 9.2 and later and save the workbook.

**Return sampled data**

In cases when workbook performance is critical or there are specific dimensions and measures you want to use in your query that are not supported by Tableau’s default query process, use sampled data instead. To return sampled data, select the **Sample data** button.

![Sample data button](image)

See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**5 Tips to Get More from Google Analytics** - Read the Tableau whitepaper (registration or sign in required).

**Google BigQuery**

This article describes how to connect Tableau to Google BigQuery and set up the data source.
In this article

Before you begin below
Make the connection and set up the data source below
Use customization attributes to improve query performance on page 434
Check if your workbook uses standard SQL or legacy SQL on page 437

Before you begin

Before you begin, gather this connection information:

- Google BigQuery email or phone, and password

Make the connection and set up the data source

1. Start Tableau and under Connect, select Google BigQuery. For a complete list of data connections, select More under To a Server. In the tab Tableau opens in your default browser, do the following:

   a. Sign in to Google BigQuery using your email or phone, and then select Next to enter your password. If multiple accounts are listed, select the account that has the Google BigQuery data you want to access and enter the password, if you're not already signed in.
b. Select **Accept** so that Tableau Desktop can access your Google BigQuery data.

c. Close the browser window when notified to do so.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. (Optional) From the **Billing Project** drop-down list, select a billing project. If you don’t select a billing project, **EmptyProject** appears in the field after you have selected the remaining fields.

   c. From the **Project** drop-down list, select a project. Alternatively, select **publicdata** to connect to sample data in BigQuery.

   d. From the **Dataset** drop-down list, select a data set.

   e. Under **Table**, select a table.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 723.
**Note:** Google BigQuery has changed support from BigQuery legacy SQL (BQL) to standard SQL. Your workbooks will upgrade to support standard SQL when you open them in Tableau.

Google BigQuery data source example

Here is an example of a Google BigQuery data source using Tableau Desktop on a Windows computer:

![Google BigQuery data source example](image)

**Note:** Because of the large volume of data in BigQuery, Tableau recommends that you connect live.

Use customization attributes to improve query performance

You can use customization attributes to improve the performance of large result sets returned from BigQuery to Tableau Online and Tableau Server, and on Tableau Desktop.

You can have the customization attributes included in your published workbook or data source, as long as you specify the attributes before you publish the workbook or data source to Tableau Online or Tableau Server.
Use Google BigQuery customization attributes

Customization attributes accept integer values and affect both live queries and extract refreshes for the specified connection.

The following attributes help the most to increase performance of large result sets:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bq-fetch-tasks</td>
<td>Number of parallel background tasks to use when fetching data using HTTP. The default is 10.</td>
</tr>
<tr>
<td>bq-large-fetch-rows</td>
<td>Number of rows to fetch in each batch for spool queries. The default is 50000.</td>
</tr>
</tbody>
</table>

The following attributes are also available and are primarily used for small queries:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bq-fetch-rows</td>
<td>Number of rows to fetch in each batch for non-spool queries. The default is 10000.</td>
</tr>
<tr>
<td>bq-response-rows</td>
<td>Number of rows returned in non-spool non-batched queries. The default is 10000.</td>
</tr>
</tbody>
</table>

This capability setting accepts yes or no values and can be useful when testing:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAP_BIGQUERY_FORCE_SPOOL_JOB</td>
<td>Force all queries to use the temp table approach. The default value is “no.” Change the value to “yes” to turn this attribute on.</td>
</tr>
</tbody>
</table>

How Tableau returns rows from Google BigQuery

Tableau uses two approaches to return rows from BigQuery: the default non-spool approach, or the temp table (spool) approach:

- On the first attempt, queries are executed using the default, non-spool query, which uses the bq-fetch-rows setting.
- If the result set is too large, the BigQuery API returns an error and the Tableau BigQuery connector retries the query by saving the results into a BigQuery temp table. The BigQuery connector then reads from that temp table, which is a spool job that uses the bq-large-fetch-rows setting.
How to specify the attributes

You can specify attributes in one of two ways: in a Tableau Datasource Customization .tdc file, or in the workbook or data source XML.

Specify attributes in a .tdc file

To specify customization attributes during a publish workbook or publish data source operation from Tableau Desktop, follow these steps:

1. Create an XML file that contains the customization attributes.
2. Save the file with a .tdc extension, for example, BigQueryCustomization.tdc.
3. Save the file to the My Tableau Repository\Datasources folder.

The customization attributes in the .tdc file are read and included by Tableau Desktop when the data source or workbook is published to Tableau Online or Tableau Server.

Important: Tableau does not test or support TDC files. These files should be used as a tool to explore or occasionally address issues with your data connection. Creating and maintaining TDC files requires careful manual editing, and there is no support for sharing these files.

Example of a .tdc file with recommended settings for large extracts

```xml
<connection-customization class='bigquery' enabled='true' version='8.0'>

  <vendor name='bigquery'/>

  <driver name='bigquery'/>

  <customizations>

    <customization name='bq-fetch-tasks' value='10'/>

    <customization name='bq-large-fetch-rows' value='10000'/>

  </customizations>

</connection-customization>
```
Manually embed attributes in the XML of the workbook or data source file

You can manually embed customization attributes inside the 'connection' tag in the workbook .twb file or the data source .tds file. The BigQuery customization attributes are bold in the following example to make them easier for you to see.

Example of manually embedded attributes

```xml
<connection CATALOG='publicdata' EXECCEATALOG='some-project-123' REDIRECT_URI='some-url:2.0:oob' SCOPE='https://www.googleapis.com/auth/bigquery
https://www.googleapis.com/auth/userinfo.profile
https://www.googleapis.com/auth/userinfo.email'
authentication='yes' bq-fetch-tasks='10' bq-large-fetch-rows='10000' bq_schema='samples' class='bigquery' connection-dialect='google-bql' connection-protocol='native-api' login_title='Sign in to Google BigQuery' odbc-connect-string-extras='' project='publicdata' schema='samples' server='googleapis.com/bigquery' server-oauth='' table='wikipedia' username=''>
```

Check if your workbook uses standard SQL or legacy SQL

In 2016, Google updated the BigQuery APIs to support standard SQL, in addition to still supporting BigQuery SQL (now called legacy SQL). Starting in Tableau 10.1, the Google BigQuery connector has been upgraded to support standard SQL, and also still supports legacy SQL. Standard SQL enables users of the BigQuery connector to use level of detail expressions, get faster metadata validation, and select a billing project with the connection.

Now, when you create a new workbook, Tableau supports standard SQL by default. Tableau also supports legacy SQL using the Use Legacy SQL option on the Data pane. For example, when you open a workbook that was created using a previous version of Tableau Desktop, and if your workbook uses legacy SQL, the Use Legacy SQL option is selected.

You might want to configure the Use Legacy SQL option for the following reasons:

- You have an existing workbook that you want to update to use standard SQL in order to write level of detail expressions or to take advantage of other improvements. In this case, make sure that the Use Legacy SQL option is not selected.
- You are creating a new workbook that needs to connect to a legacy SQL view. You can’t
mix legacy SQL with standard SQL, so you need to select the **Use Legacy SQL** option for the workbook to function.

In Google BigQuery, views are written in standard SQL or legacy SQL. You can join views written in standard SQL to views written in standard SQL, or views written in legacy SQL to views written in legacy SQL, and you can join views written in either version of SQL to a table. But you can’t join views written in standard SQL and views written in legacy SQL in one workbook. When you join views, you must set the **Use Legacy SQL** check box to correspond to the SQL type used in the view you are connecting to.

**Note:** Tableau Desktop provides limited support for nested data when you use legacy SQL or standard SQL. For example, if a table contains nested data and you’re using legacy SQL or standard SQL, on the data source page, **Update Now** won’t work.

For more information about migrating from legacy SQL to standard SQL, see [Migrating from legacy SQL](https://cloud.google.com/bigquery/docs/migrating-from-legacy-sql) on the Google Cloud Platform website.

See also

- **Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 843 – Begin your data analysis.
- Google BigQuery & Tableau: Best Practices - Read the Tableau whitepaper (registration or sign in required)

**Google Cloud SQL**

This article describes how to connect Tableau to a Google Cloud SQL database instance and set up the data source.

In this article

- **Before you begin** on the next page
- **Make the connection and set up the data source** on the next page
- **Sign in on a Mac** on page 440
Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Google Cloud SQL**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the user name and password, and then select **Sign In**.
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the **Database** drop-down list, select a database or use the text box to search for a database by name.
   c. Under **Table**, select a table or use the text box to search for a table by name.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 723.
Google Cloud SQL data source example

Here is an example of a Google Cloud SQL data source using Tableau Desktop on a Windows computer.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**Google Sheets**

This article describes how to connect Tableau to Google Sheets and set up the data source.

In this article

Before you begin below
Make the connection and set up the data source below
About .ttde and .hhyper files on page 444
Troubleshoot Google Sheets issues on page 445

Before you begin

Before you begin, gather this connection information:

- Google email address and password

**Note:** The Google Sheets connector does not support Google Team Drive accounts.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Google Sheets**. For a complete list of data connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:

   a. Sign in to Google Sheets using your email or phone, and then select **Next** to enter your password. If multiple accounts are listed, select the account that has the Google Sheets data you want to access and enter the password, if you’re not already signed in.
b. Select **Allow** so that Tableau Desktop can access your Google Sheets data.

c. Close the browser window when notified to do so.

d. Select a Google Sheet from the list or use the text box to search for a Google Sheet by name or by URL, and then select **Connect**.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data figure out which data source to connect to.
   b. If your Google Sheets file has one table, select the sheet tab to start your analysis.

   **Note:** Tableau doesn't support pivot tables in Google Sheets.

Select Your Google Sheet dialog box functionality

The Select Your Google Sheet dialog box includes the following functionality:

- The list of sheets that you can select from includes your private sheets, sheets shared with you, and the public sheets that you've accessed in the past.
- If you search by URL and the URL doesn't exist or you don't have access to it, an error displays.
- You can select the **Name** and **Last opened by me** column names to sort the Google Sheets, and when you select a sheet you can preview it in the right pane. You cannot sort by **Owned by**.

Google Sheets data source example

Here is an example of a Google Sheets data source:
Connect to more data

You can connect to more than one table by using join. For more information, see Join Your Data on page 657.

You can also connect to a named range the same way you connect to a worksheet. The named range functions as a table in Tableau.

You create named ranges in Google Sheets by highlighting a range of cells and then selecting Data > Named ranges. When you connect to a named range in Tableau, an icon appears next to the sheet in the Data Source tab as shown below.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.
In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

Troubleshoot Google Sheets issues

Data limit in Google Drive

You can store up to 2 million cells for spreadsheets that are created in or converted to Google Sheets. For more information, see Files you can store in Google Drive in the Google Drive Help.

Error message: Internal Error - An unexpected error occurred and the operation could not be completed.

If there are errors in your Google Sheet, such as #DIV/0! or #N/A, Tableau is unable to create an extract and an error message will appear. To resolve this issue, wrap the function with `iferror()` and have it return a blank, or any value that's appropriate.

For example, the sheet below includes a #DIV/0! error.

The solution is to wrap the calculation in an `iferror()` calculation.
See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Hortonworks Hadoop Hive

This article describes how to connect Tableau to a Hortonworks Hadoop Hive database and set up the data source.

In this article

Before you begin below
Make the connection and set up the data source on the next page
Sign in on a Mac on page 449
Work with Hadoop Hive data on page 450

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Authentication method:
  - No Authentication
  - Kerberos
  - User Name
  - User Name and Password
  - Microsoft Azure HDInsight Service (starting in version 10.2.1)
- Transport options depend on the authentication method you choose and can include the following:
  - Binary
  - SASL
  - HTTP
• Sign-in credentials depend on the authentication method you choose and can include the following:
  • User name
  • Password
  • Realm
  • Host FQDN
  • Service name
  • HTTP path
• Are you connecting to an SSL server?
  • (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

**Note:** Make sure you use the latest available drivers. To get the latest drivers, see Hortonworks Hadoop Hive on the Tableau Driver Download page.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Hortonworks Hadoop Hive**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of server that hosts the database.
   b. In the **Authentication** drop-down list, select the authentication method to use.
   c. Enter the information that you are prompted to provide. The information you are prompted for depends on the authentication method you choose.
   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 643.
Select **Sign In**.

Select the **Require SSL** option when connecting to an SSL server.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

**2. On the data source page, do the following:**

- **a. (Optional)** Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

- **b.** From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.

- **c.** In the **Table** text box, select the search icon or enter the table name and select the search icon, and then select the table.

- **d.** Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 723.

**Note:** This database type only support equal (=) join operations.

**Hortonworks Hadoop Hive data source example**

Here is an example of a Hortonworks Hadoop Hive data source using Tableau Desktop on a Windows computer:
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.
Work with Hadoop Hive data

Work with date/time data

Tableau supports TIMESTAMP and DATE types natively. However, if you store date/time data as a string in Hive, be sure to store it in ISO format (YYYY-MM-DD). You can create a calculated field that uses the DATEPARSE or DATE function to convert a string to a date/time format. Use DATEPARSE() when working with an extract, otherwise use DATE(). For more information, see Date Functions on page 1298.

For more information about Hive data types, see Dates on the Apache Hive website.

NULL value returned

A NULL value is returned when you open a workbook in Tableau 9.0.1 and later and 8.3.5 and later 8.3.x releases that was created in an earlier version and has date/time data stored as a string in a format that Hive doesn't support. To resolve this issue, change the field type back to String and create a calculated field using DATEPARSE() or DATE() to convert the date. Use DATEPARSE() when working with an extract, otherwise use the DATE() function.

High latency limitation

Hive is a batch-oriented system and is not yet capable of answering simple queries with very quick turnaround. This limitation can make it difficult to explore a new data set or experiment with calculated fields. Some of the newer SQL-on-Hadoop technologies (for example, Cloudera’s Impala and Hortonworks’ Stringer project) are designed to address this limitation.

Truncated columns in Tableau

The default string column length for Hortonworks Hadoop Hive is 255 characters. For more information about Hortonworks Hive ODBC driver configuration options, and specifically about DefaultStringColumnLength, see the Hive ODBC Driver User Guide from Hortonworks.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.
IBM BigInsights

This article describes how to connect Tableau to an IBM BigInsights database.

In this article

Before you begin below
Make the connection and set up the data source on the next page

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to and port number
- Database name
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

SSL requirements

This connector enables you to connect to your server using SSL authentication. If you work in an SSL environment, your computer is probably already configured to support SSL. If you have trouble making an SSL connection with Tableau, make sure that the following IBM software is installed on your computer:

- v10.5fp3 connector for IBM DB2
- GSK8 SSL Library

Note that the v10.5fp3 connector and the GSK8 SSL Library must have the same bitness. For example, both must be either 32-bit or 64-bit. For more information, see Installation of the GSK8 SSL Library on the IBM website.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau
displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select IBM BigInsights. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database and the port number to use.
   b. Enter the name of the database you want to connect to.
   c. Enter your user name and password.
   d. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   e. Select Sign In.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Schema drop-down list, select a schema or use the text box to search for a schema by name.
   c. Under Table, select a table or use the text box to search for a table by name.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

IBM BigInsights data source example

Here is an example of an IBM BigInsights data source:
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**Using Tableau with IBM BigInsights Hadoop** - Tableau blog post.

**IBM DB2**

This article describes how to connect Tableau to an IBM DB2 database and set up the data source. You can also use this connector to connect to an IBM DB2 for z/OS database. See the **Technical Specifications** to confirm which DB2 databases are supported.
In this article

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to (Host name) and port number
- Database name
- User name (User ID) and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Port settings

The port number is dependent on the type of server you are connecting to and whether you are connecting to an encrypted port. Generally, use 50000 for a non-encrypted port and 60000 for an encrypted port. It is possible that your server is configured to use a non-standard port. Contact your administrator if you don’t know which port to connect to.

SSL requirements

This connector enables you to connect to your server using SSL authentication. If you work in an SSL environment, your computer is probably already configured to support SSL. If you have trouble making an SSL connection with Tableau, make sure that the following IBM software is installed on your computer:

- v10.5fp3 connector for IBM DB2
- GSK8 SSL Library

Note that the v10.5fp3 connector and the GSK8 SSL Library must have the same bitness. For example, both must be either 32-bit or 64-bit. For more information, see Installation of the GSK8 SSL Library on the IBM website.
Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select IBM DB2. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database (the Host name), the port number, and the name of the database that you want to connect to.
   b. Enter your user name (User ID) and password.
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Schema drop-down list, select a schema or use the text box to search for a schema by name.
   c. Under Table, select a table or use the text box to search for a table by name.
   d. Drag a table to the canvas, and then select the sheet tab to start your analysis.
      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.
IBM DB2 data source example

Here is an example of an IBM DB2 data source:

Note: Tableau doesn't support the DECFLOAT data type.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Connecting Tableau to a Db2 database - Review connection instructions on the IBM website.
IBM PDA (Netezza)

This article describes how to connect Tableau to an IBM PDA (PureData System for Analytics) database and set up the data source.

In this article

- Before you begin below
- Make the connection and set up the data source below

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select IBM PDA (Netezza). For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the name of the database that you want to connect to.
   c. Enter the user name and password.
      Select the Require SSL option when connecting to an SSL server.
   d. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of
every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.

e. Select Sign In.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Under Table, select a table or use the text box to search for a table by name.

   c. Drag a table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

IBM PDA (Netezza) data source example

Here is an example of an IBM PDA (Netezza) data source:
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**Intuit QuickBooks Online**

This article describes how to connect Tableau to Intuit QuickBooks Online data and set up the data source.
Before you begin

Before you begin, gather this connection information:

- Your email address or user ID and password for your Quickbooks Online account.

**Note:** To connect Tableau to QuickBooks Online data, you must be a Company Administrator on the QuickBooks Online company. Only one administrator at your company can connect Tableau to QuickBooks Online.

QuickBooks Online doesn't support retrieving the Inventory Adjustment transactions. For information about when Inventory Adjustment transactions will be available in the QuickBooks Online API, see the responses to this question on the Intuit Developer website.

Make the connection and set up a data source

1. Start Tableau and under **Connect**, select **Intuit QuickBooks Online**. For a complete list of data connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:
   a. Enter your email address or user ID and password for your QuickBooks Online account.
   b. Select **Sign In**.
      
      QuickBooks Online periodically prompts for two-step verification. If this happens, follow the prompts to get a code to confirm your Intuit account.
   c. If more than one company is associated with your account, select the company you want to connect to.
   d. Select **Authorize** to authorize Intuit to securely share your data with Tableau Desktop.
If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Under Table, select a table and drag it to the top of the canvas.

   c. Select the Sheet 1 tab to start your analysis.

      After you select the sheet tab, Tableau imports the data by creating an extract. Note that Tableau Desktop supports only extracts for QuickBooks Online. You can update the data by refreshing the extract. For more information, see Extract Your Data on page 773.

      Creating extracts may take some time depending on the amount of data that is included.

**QuickBooks Online data source example**

Here is an example of a QuickBooks Online data source using Tableau Desktop on a Windows computer:
Troubleshoot QuickBooks Online errors

You might see one of the following errors when you try to connect Tableau to your QuickBooks Online data.

Workbooks saved in previous versions of Tableau

Workbooks and data sources created prior to version 2018.2 of Tableau will not be able to access tables or fields added in later versions of the connector. To resolve this, open a new workbook and create a new connection to your QuickBooks Online data source. Copy and paste the worksheets you want to keep from the previous connection, or recreate them in Tableau.

Sorry, only administrators can buy apps...

QuickBooks Online requires that each user who connects an app to QuickBooks Online be a Company Administrator. The following Intuit error appears if you are not an administrator on the company account:

Sorry, only administrators of <email address>’s Company can buy apps in the company. Please contact administrator in the company or else choose another company.

To resolve this issue, ask your company account owner to make you a Company Administrator on the account.

Error Code: app_already_purchased

Only one Company Administrator is allowed to connect per application, for example to Tableau Desktop. If someone in your company has already connected Tableau to your QuickBooks Online account, the following Intuit error appears:

Oops! An error has occurred. Please close this window and try again.

Error Code: app_already_purchased

Message: The application has already been subscribed to by another user for this company. Please contact <email address> to make changes to this subscription.

To resolve this issue, you need to ask that Company Administrator to give up the application privileges for Tableau Desktop and provide them to you. If that’s not possible, for example, because the administrator actively uses Tableau, another option is for the administrator to
publish the data source to Tableau Server or Tableau Online, so that anyone with permission can access the data.

Changing a Connection Opens the My Apps Page

If you need to access another QuickBooks Online company, after you sign in (and possibly go through the two-step authentication process), Intuit might show the My Apps page, rather than the “select the company to connect to” page. To resolve this issue, follow these steps:

1. Close the My Apps page.
2. On the Tableau Connect pane, select Intuit QuickBooks Online.
3. If more than one company is associated with your account, select the company you would like to connect to.
4. Select Authorize to open the Tableau data source page.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Kognitio

This article describes how to connect Tableau to a Kognitio database and set up the data source.

In this article

Before you begin below
Make the connection and set up the data source on the next page
Sign in on a Mac on page 466

Before you begin

Before you begin, gather this connection information:
- Name of the server that hosts the database you want to connect to
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Kognitio. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database you want to connect to.
   b. Enter the user name and password.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.

Select the Require SSL check box when connecting to an SSL server.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Schema drop-down list, select a schema or use the text box to search for a schema by name.
c. Under **Table**, select a table or use the text box to search for a table by name.
d.
Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 723.

**Kognitio Data Source Example**

Here is an example of a Kognitio data source using Tableau Desktop on a Windows computer:

![Kognitio Data Source Example](image)

**Sign in on a Mac**

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.
Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

MapR Hadoop Hive

This article describes how to connect Tableau to a MapR Hadoop Hive database and set up the data source.

In this article

Before you begin below
Make the connection and set up the data source on the next page
Sign in on a Mac on page 470
Work with Hadoop Hive data on page 471

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Authentication method:
  - No Authentication
  - Kerberos
  - User Name
  - User Name and Password
  - Microsoft Azure HDInsight Service (starting in version 10.2.1)
- Transport options depend on the authentication method you choose and can include the
following:

- Binary
- SASL
- HTTP

- Sign-in credentials depend on the authentication method you choose and can include the following:
  - User name
  - Password
  - Realm
  - Host FQDN
  - Service Name
  - HTTP Path

- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

**Note:** Make sure you use the latest available drivers. To get the latest drivers, see MapR Hadoop Hive on the Tableau Driver Download page.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **MapR Hadoop Hive**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. In the **Authentication** drop-down list, select the authentication method to use.
   c. Enter the information that you are prompted to provide. The information you are
prompted for depends on the authentication method you choose.

d. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.

e. Select Sign In.

Select the Require SSL option when connecting to an SSL server.

If Tableau can’t make the connection, verify that your credentials are correct. If you still can’t connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the Schema drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.

c. In the Table text box, select the search icon or enter the table name and select the search icon.

d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Note: This database type supports only equal (=) join operations.

MadR Hadoop data source example

Here is an example of a MapR Hadoop Hive data source using Tableau Desktop on a Windows computer.
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences > Network > Advanced**, and then open the **DNS** tab.
Work with Hadoop Hive data

Work with date/time data

Tableau supports TIMESTAMP and DATE types natively. However, if you store date/time data as a string in Hive, be sure to store it in ISO format (YYYY-MM-DD). You can create a calculated field that uses the DATEPARSE or DATE function to convert a string to a date/time format. Use DATEPARSE() when working with an extract, otherwise use DATE(). For more information, see Date Functions on page 1298.

For more information about Hive data types, see Dates on the Apache Hive website.

NULL value returned

A NULL value is returned when you open a workbook in Tableau 9.0.1 and later and 8.3.5 and later 8.3.x releases that was created in an earlier version and has date/time data stored as a string in a format that Hive doesn't support. To resolve this issue, change the field type back to String and create a calculated field using DATEPARSE() or DATE() to convert the date. Use DATEPARSE() when working with an extract, otherwise use the DATE() function.

High latency limitation

Hive is a batch-oriented system and is not yet capable of answering simple queries with very quick turnaround. This limitation can make it difficult to explore a new data set or experiment with calculated fields. Some of the newer SQL-on-Hadoop technologies (for example, Cloudera's Impala and Hortonworks' Stringer project) are designed to address this limitation.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Marketo

This article describes how to connect Tableau to Marketo data and set up the data source.
In this article

Before you begin below
Make the connection and set up the data source below
Use Dashboard Starters on page 474
Date range selections can impact performance on page 475

Before you begin

Before you begin, gather this connection information:

- Custom Service Endpoint
- Client ID
- Client Secret

For details about how to create the custom service information you need from Marketo to connect, see the Custom Service topic on the Marketo website.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Marketo. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter your Custom Service Endpoint, Client ID, and Client Secret.
   b. Select Sign In.
   c. Select Filter Type: Relative date range or Fixed date range, and then select or specify the range.
   d. Select Connect to authorize Marketo to securely share your data with Tableau Desktop.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network or Marketo administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data
source naming convention that helps other users of the data source figure out which data source to connect to.

b. Under **Table**, select a table and drag it to the top of the canvas.

c. Select the sheet tab to start your analysis.

After you select the sheet tab, Tableau imports the data by creating an extract. Note that Tableau Desktop supports only extracts for Marketo. You can update the data by refreshing the extract. For more information, see **Extract Your Data** on page 773.

Creating extracts may take some time depending on the amount of data that is included.

**Marketo data source example**

Here is an example of a Marketo data source using Tableau Desktop on a Windows computer:
Use Dashboard Starters

If you are connecting to your data through Tableau Online, you can use a Dashboard Starter to quickly build informative dashboard designs made specifically for key business metrics. For more information, see Dashboard Starters for Cloud-based Data on page 2252
Date range selections can impact performance

It's tempting to gather as much data as possible when you do an analysis, however, retrieving records from Marketo can be time-consuming. Tableau doesn’t know how much data there is in a particular date range until it retrieves the data. For this reason, you should restrict your date range at first, and then expand after you evaluate performance.

To give you a rough idea of how much time it might take to retrieve data from Marketo, tests were conducted using a high-speed connection. This table shows how long it took in the test environment to retrieve a given number of records.

<table>
<thead>
<tr>
<th>Number of records</th>
<th>Time to retrieve</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td>4.5 seconds</td>
</tr>
<tr>
<td>10,000</td>
<td>45 seconds</td>
</tr>
<tr>
<td>100,000</td>
<td>8 minutes</td>
</tr>
<tr>
<td>1,000,000</td>
<td>75 minutes</td>
</tr>
</tbody>
</table>

See also

**Set Up Data Sources on page 646** – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data on page 843** – Begin your data analysis.

MarkLogic

This article describes how to connect Tableau to a MarkLogic database and set up the data source.

In this article

**Before you begin below**

*Make the connection and set up the data source on the next page*

Before you begin

Before you begin, gather this connection information:
- Name of the server that hosts the database you want to connect to
- Port number for the ODBC server process
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select MarkLogic. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database you want to connect to.
   b. Enter the port number for the ODBC server process of the database you want to connect to.
   c. Enter your user name and password.
   d. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   e. Select Sign In.
      If the connection is unsuccessful, verify that your user name and password are correct and that the port number correctly identifies the MarkLogic database configured by your database administrator. If the connection continues to fail, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data
source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Schema** drop-down list, select a schema or use the text box to search for a schema by name.

c. Under **Table**, select a table or use the text box to search for a table by name.

d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see [Connect to a Custom SQL Query](#) on page 723.

MarkLogic data source example

Here is an example of a MarkLogic data source:
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**Tableau and MarkLogic Visualize Unstructured Data** - Read the Tableau blog post.

**MemSQL**

This article describes how to connect Tableau to a MemSQL database and set up the data source.

In this article

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

**Driver required**

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the **Driver Download** page where you can find driver links and installation instructions.
Make the connection and set up the data source

1. Start Tableau and under Connect, select MemSQL. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the user name and password.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.
      Select the Require SSL check box when connecting to an SSL server.
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Database drop-down list, select a database or use the text box to search for a database by name.
   c. Under Table, select a table or use the text box to search for a table by name.
   d. Drag a table to the canvas, and then select the sheet tab to start your analysis.
      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

MemSQL data source example

Here is an example of a MemSQL data source using Tableau Desktop on a Windows computer:
Note: In order to use your net services definitions in Tableau, you must set either TNS_ADMIN or ORACLE_HOME as an environment variable. To set TNS_ADMIN as the environment variable use the full path of the directory that contains the tnsnames.ora file. To set ORACLE_HOME as an environment variable use the path to the main Oracle directory.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.
Microsoft Analysis Services

This topic describes how to connect Tableau to a Microsoft Analysis Services database and set up a data source.

Before you begin below
Make the connection and set up the data source below
Incompatible Measures and Dimensions on page 483

Before you begin

Before you begin, gather this connection information:

- For a remote cube file: Server name, or URL if connecting to the server using HTTP
- For a local cube file: File name
- Authentication method: Windows Authentication or user name and password

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Microsoft Analysis Services. For a complete list of data connections, select More under To a Server. Then do the following:
a. Select whether to connect to a remote cube file on a server or to a local cube file.

To connect to a remote cube file, select Server and enter the name of the server in the text box. If you are connecting to the server using HTTP, you can enter the URL as the server name.

To connect to a local cube file, select Local cube file and select Browse to navigate to the cube file on your computer.

b. Select how you want to sign in to the server. Specify whether to use Windows Authentication or a specific user name and password. If the cube is password protected, and you are not in a Kerberos environment, you must enter your user name and password.

Specify whether to use Windows Authentication or a specific user name and password.

c. Select Sign In.

If Tableau can’t make the connection, verify that your credentials are correct. If you still can’t connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. Select a database.

c. Select a cube from the database.

d. Select the sheet tab to start your analysis.

Microsoft Analysis Services data source example

Here is an example of a Microsoft Analysis Services data source:
Incompatible Measures and Dimensions

When you build views in Tableau using a Microsoft Analysis Services Cube you might see some fields highlighted in gray, or you might see a caution symbol on a field in the view with the message, "This measure is incompatible with one or more dimensions in this view." This occurs because it is possible to have measures and dimensions that don’t make a lot of sense when
placed in the view together. For example, you may have a measure for Sales Quota. It won’t make sense to place that measure against a dimension containing products if products don’t have sales quotas.

Tableau helps you figure out the dimensions and measure that can be used together in meaningful ways by highlighting unrelated dimensions and measures in gray. So in the last example, when we place Sales Quota onto a shelf, the products dimensions are highlighted in gray. Highlighted dimensions are not disabled and can still be added to the view. When you add an incompatible measure to the view, the measure is marked with a caution symbol. For more information on working with cubes, see Cube Data Sources on page 2216.

See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**Microsoft PowerPivot**

This article describes how to connect Tableau to a Microsoft PowerPivot database.

In this article

- **Before you begin** below
- **Make the connection and set up the data source** on the next page
- **Use PowerPivot perspectives** on page 486

**Before you begin**

Before you begin, gather this connection information:
- SharePoint URL, or SharePoint, UNC, or local Excel file name

If you plan to publish the workbook to Tableau Server, make sure to connect to a PowerPivot file on SharePoint. Tableau Server does not support connections to local PowerPivot files.

**Note:** Tableau Desktop supports Microsoft PowerPivot 2010 and 2013.

Use this connector with Tableau Desktop on a Windows computer.

**Driver required**

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

**Make the connection and set up the data source**

1. Start Tableau and under **Connect**, select **Microsoft PowerPivot**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Select whether to connect to a PowerPivot file using a SharePoint URL, a SharePoint UNC (file path), or a local Excel file.
   b. Select **Sign In**.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. The file name appears under **Select a File**. Note that there is one file per connection. Search for or select a perspective available in that file.
   c. Select the sheet tab to start your analysis.

**Microsoft PowerPivot data source example**

Here is an example of a Microsoft PowerPivot data source:
Use PowerPivot perspectives

PowerPivot data may contain one or more perspectives. Perspectives are subsets of objects from the model that define sets of data. Typically perspectives are defined for a particular group of users or business scenario. Using perspectives can make it easier to navigate large data sources.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.
Build Charts and Analyze Data on page 843 – Begin your data analysis.

Microsoft SQL Server

This article describes how to connect Tableau to a Microsoft SQL Server database and set up the data source.

**Note:** Use the Microsoft SQL Server connector to connect to Microsoft SQL Server Parallel Data Warehouse (PDW), Microsoft Azure SQL Data Warehouse, or Microsoft Azure SQL Database.

In this article

**Before you begin below**

Make the connection and set up the data source on the next page

Sign in on a Mac on page 490

Resolve "Worksheet Unavailable" error message on page 490

Before you begin

Before you begin, gather this connection information:

- Name of the server you want to connect to
- (Optional) Port number if you want to connect to a non-default port
- (Optional) Database name if you want to connect to a contained database
- Authentication method: Windows Authentication or user name and password
- Are you connecting to an SSL server?
- Do you want to set the database isolation level to read uncommitted data?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau
displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Microsoft SQL Server. For a complete list of data connections, select More under To a Server. Then do the following:

   a. Enter the name of the server you want to connect to.
      
      To connect to a non-default port, use the `<server name>, <port number>` format when entering the server name. For example, ExampleServer, 8055.

   b. (Optional) Enter a database name if you want to connect to a contained database.

   c. Select how you want to sign in to the server. Specify whether to use Windows Authentication or a specific user name and password. If the server is password protected, and you are not in a Kerberos environment, you must enter the user name and password.
      
      Select the Require SSL check box when connecting to an SSL server.

   d. Specify whether to Read uncommitted data. This option sets the database isolation level to Read Uncommitted. Long queries from Tableau, including extract refreshes, can lock the database and delay transactions. Select this option to allow queries to read rows that have been modified by other transactions even when they have not been committed yet. When this option is cleared, Tableau uses the default isolation level specified by the database.

   e. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.

   f. Select Sign In.
      
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data
source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Database** drop-down list, select a database or use the text box to search for a database by name.

c. Under **Table**, select a table or use the text box to search for a table by name.

You can also specify a stored procedure in the database. For more information about stored procedures, including a list of constraints specific to SQL Server databases, see **Use a Stored Procedure** on page 733.

Starting with Tableau version 2018.1, you can connect to spatial columns. For more information, see **Connect to Spatial Data in Microsoft SQL Server** on page 1908.

d. Drag the table or stored procedure to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 723.

Starting with Tableau version 2018.1, you can use Custom SQL to perform advanced spatial analysis on spatial columns in Microsoft SQL Server. For more information, see **Use Custom SQL and RAWSQL to perform advanced spatial analysis** on page 1910.

**Note:** Tableau Desktop does not support the Microsoft SQL Server TIME data type. Fields of this data type are not imported and do not appear in Tableau Desktop. If included in stored procedures, TIME data type fields will not appear in Tableau Desktop.

For more information, see **Use a Stored Procedure** on page 733.

**Microsoft SQL Server data source example**

Here is an example of a Microsoft SQL Server data source using Tableau Desktop on a Windows computer:
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

Resolve "Worksheet Unavailable" error message

When you open a workbook, you may see a “Worksheet Unavailable” error message and then, when you select Edit Connection, you are prompted to provide your sign-in credentials. This error typically displays under one of two circumstances:
• You don’t have access to a data connection used in the workbook.
• Your credentials are invalid, for example, your password has expired.

To resolve, verify that your sign-in credentials are correct. If they are, contact the Tableau administrator to ask about access to the data connection.

See also
* **Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.
* **Build Charts and Analyze Data** on page 843 – Begin your data analysis.
* **Connect to Spatial Data in Microsoft SQL Server** on page 1908

**MonetDB**

This article describes how to connect Tableau to a MonetDB database and set up the data source.

In this article

**Before you begin** below

**Make the connection and set up the data source** on the next page

**Before you begin**

Before you begin, gather this connection information:

• Name of the server that hosts the database you want to connect to
• Database name
• User name and password
• (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.
Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select MonetDB. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database and the name of the database that you want to connect to.
   b. Enter the user name and password.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.
   
   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Schema drop-down list, select a schema.
   c. Under Table, select a table or use the text box to search for a table by name.
   d. Drag a table to the canvas, and then select the sheet tab to start your analysis.
   
   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.
MonetDB data source example

Here is an example of a MonetDB data source:

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

MongoDB BI Connector

This article describes how to connect Tableau to MongoDB Business Intelligence (BI) data and set up the data source.
In this article

Before you begin below
Make the connection and set up the data source below
Sign in options on page 496

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password, if MongoDB authentication is enabled
- Database name associated with the user’s credentials, if MongoDB authentication is enabled
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

You must install and run the MongoDB Connector for BI, version 2.1 or later, before you can connect Tableau to your MongoDB BI data.

To get the MongoDB Connector for BI, sign in to the MongoDB website and download the connector. Follow the Installation instructions on the MongoDB website.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select MongoDB BI Connector. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server.
   b. If MongoDB authentication is enabled, enter your user name with associated database name, and password. For information about the correct format to use, see Sign in options on page 496. If MongoDB authentication isn't enabled,
proceed to step c.

Select the Require SSL check box when connecting to an SSL server. For more information about connecting with SSL to MongoDB, see Connecting Tableau to MongoDB on the MongoDB website.

c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.

d. Select Sign In.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Select the sheet tab to start your analysis.

MongoDB BI Connector data source example

Here is an example of a MongoDB BI Connector data source using Tableau Desktop on a Windows computer:
Sign in options

User accounts in MongoDB are associated with specific logical databases in MongoDB. So when users log into MongoDB, they need to specify the database name associated with the user’s credentials. This is done by passing special options in the username string. For example, if user henrywilson is associated with the example database, he enters the following in the Username field:

henrywilson?source=example

You can use the mechanism option to define the authentication mechanism if you want to use a mechanism other than the default, SCRAM-SHA-1. For example, if user henrywilson wants to connect to the example database using challenge/response as the authentication mechanism, he enters the following in the Username field:

henrywilson?source=example,mechanism=MONGODB-CR

For more information about authentication options, see the MongoDB website.
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**MySQL**

This article describes how to connect Tableau to a MySQL database and set up the data source.

In this article

Before you begin below

**Make the connection and set up the data source** on the next page

**Sign in on a Mac** on page 499

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the **Driver Download** page where you can find driver links and installation instructions.
Make the connection and set up the data source

1. Start Tableau and under Connect, select MySQL. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the user name and password.
      Select the Require SSL option when connecting to an SSL server.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Database drop-down list, select a database or use the text box to search for a database by name.
   c. Under Table, select a table or use the text box to search for a table by name.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.
      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

MySQL data source example

Here is an example of a MySQL data source using Tableau Desktop on a Windows computer:
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.
OData

This article describes how to connect Tableau to an OData data source. Tableau connects to OData V1 - V4.

In this article

**Before you begin** below

**Make the connection and set up the data source** below

**OData connector support** on page 502

Before you begin

Before you begin, gather this connection information:

- Server URL for the data you want to connect to
- Sign-in credentials (user name and password), if required,

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **OData**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the server URL for the data you are connecting to.
   b. If necessary, enter authentication information (user name and password).
   c. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
Note: If you publish the data source to Tableau Server or Tableau Online, the data source name is limited to 255 characters. Longer names cause a RepositoryException error.

b. Select the sheet tab to start your analysis.

After you select the sheet tab, Tableau imports the data by creating an extract. For more information about extracts, see Extract Your Data on page 773. Note that Tableau Desktop supports only extracts for OData.

OData data source example

Here is an example of an OData data source:
OData connector support

- The OData connector supports only extracts, which means that some OData V4 functionality is not supported, for example, interactive query arguments such as $expand or $select.
- The OData connector doesn't support browsing OData service documents.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.
OneDrive

This article describes how to connect Tableau to OneDrive data and set up the data source.

In this article

Before you begin below
Make the connection and set up the data source below
Use Data Interpreter to clean your data on page 506

Before you begin

Before you begin, get the email address and password for your OneDrive account.

Make the connection and set up the data source

1. Start Tableau and under Connect, select OneDrive. For a complete list of data connections, select More under To a Server. In the tab Tableau opens in your default browser, do the following:
   a. Enter your email address and then select Next.
   b. Enter your password and then select Sign In.
   c. Select Yes to grant Tableau access to your OneDrive files.
   d. Close the browser window when notified to do so.
   e. Search for or select the file to connect to, and then select Connect.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Select the sheet tab to start your analysis.

Note: Tableau doesn't support pivot tables in OneDrive.
OneDrive data source example

Here is an example of a OneDrive data source using Tableau Desktop on a Windows computer:
Use Data Interpreter to clean your data

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see **Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter** on page 752.

See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

Oracle

This article describes how to connect Tableau to an Oracle database and set up the data source.

In this article

- **Before you begin** below
- **Make the connection and set up the data source** on the next page
- **Sign in on a Mac** on page 509
- **Use net services definitions in Tableau** on page 510

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to and the Oracle service name and port, or the TNS name
- Authentication method: Integrated Authentication or user name and password
- Are you connecting to an SSL server? You must configure the Oracle client before you can use SSL with Tableau. For more information, see **Configure the Oracle Client to Use SSL with Tableau** on Tableau Community.
Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Oracle. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the server name or the TNS name.
      If you enter the server name, you must enter the Oracle service name and port number.
      If you enter the TNS name, do not enter the Oracle service name and port number.
   b. Select how you want to sign in to the server. Specify whether to use Integrated Authentication or a specific user name and password.
      On Windows only, select the Require SSL check box when connecting to an SSL server.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the Schema drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema. **Note:** Search is case-sensitive.

c. Under Table, select the search icon or enter the table name and select the search icon, and then select the table. **Note:** Search is case-sensitive.

Tableau Desktop also supports connecting to Oracle table functions, which appear under Stored Procedures in the left pane of the Data Source page.

d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Oracle data source example

Here is an example of an Oracle data source using Tableau Desktop on a Windows computer:
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.
Use net services definitions in Tableau

In order to use your net services definitions in Tableau, you must set either TNS_ADMIN or ORACLE_HOME as an environment variable. To set TNS_ADMIN as the environment variable use the full path of the directory that contains the tnsnames.ora file. To set ORACLE_HOME as an environment variable use the path to the main Oracle directory.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Troubleshooting Oracle Connection Errors - Review the Tableau Knowledge Base article about how to troubleshoot errors.

Oracle Eloqua

This topic describes how to connect Tableau to Oracle Eloqua data and set up the data source. It also describes how date range selections can impact performance.

In this article

Before you begin below
Make the connection and set up the data source on the next page
Date range selections can impact performance on page 513
Eloqua limits on page 514

Before you begin

Before you begin, gather and confirm this information for connecting:

- You must be a member of the following Eloqua security groups:
  - Advanced Users - Marketing
  - API Users
  - Company Name
Make the connection and set up the data source

1. Start Tableau and under Connect, select Oracle Eloqua. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter Company Name, Username, and Password, and then select Sign In.
   b. Select Filter Type: Relative date range or Fixed date range, and then select or specify the range.
   c. Select Connect.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network or Eloqua administrator.

   After your credentials are verified, the Eloqua connector will only prompt you to authenticate when necessary. The Sign Out option will start the authentication process again.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data
source naming convention that helps other users of the data source figure out which data source to connect to.

b. Under **Table**, select a table and drag it to the top of the canvas.

c. Select the sheet tab to start your analysis.

After you select the sheet tab, Tableau imports the data by creating an extract. Note that Tableau Desktop supports only extracts for Eloqua. You can update the data by refreshing the extract. For more information, see [Extract Your Data](#) on page 773.

Creating extracts may take some time depending on the amount of data that is included.

**Oracle Eloqua data source example**

Here is an example of an Oracle Eloqua data source using Tableau Desktop on a Windows computer:
Date range selections can impact performance

It’s tempting to gather as much data as possible when you do an analysis, however, retrieving records from Eloqua can be time-consuming. Tableau doesn’t know how much data there is in
a particular date range until it retrieves the data. For this reason, you should restrict your date range at first, and then expand after you evaluate performance. You can modify the date range on the Data Source page. Under Connections, select the drop-down arrow for the connection and select Edit connection.

To give you a rough idea of how much time it might take to retrieve data from Eloqua, tests were conducted using a high-speed connection. This table shows how long it took in the test environment to retrieve a given number of records. The connector retrieves 1000 records per request.

<table>
<thead>
<tr>
<th>Number of records</th>
<th>Time to retrieve</th>
</tr>
</thead>
<tbody>
<tr>
<td>50,662 records</td>
<td>3 minutes 17 seconds</td>
</tr>
<tr>
<td>95,214 rows</td>
<td>6 minutes 14 seconds</td>
</tr>
<tr>
<td>194,679 rows</td>
<td>12 minutes 55 seconds</td>
</tr>
</tbody>
</table>

Eloqua limits

Field limits

Some Eloqua tables can be customized to have more than 250 fields. Tableau currently retrieves the first 250 fields and ignores the rest. When you select an Events, Custom Objects, Contracts, or Accounts table, Tableau will display a warning if there are more fields than Tableau can retrieve.

Daily table request limits

Each table requested by Tableau counts against a daily Eloqua request limit of 2000. When the limit is exceeded, Eloqua operations might choose to disable Tableau, in which case, they will contact the account owners. You can monitor your daily usage in Eloqua using the Eloqua Marketing Operations Center.

Incremental Refresh

For all the activity labels (with the exception of "Activities- all"), you can now implement incremental refreshes. For more information, please see Refresh Extracts on page 798.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.
Build Charts and Analyze Data on page 843 – Begin your data analysis.

Oracle Essbase

This article describes how to connect Tableau to an Oracle Essbase database.

In this article

Before you begin below
Make the connection and set up the data source below
Set an accounts dimension on page 517
Build views with Oracle Essbase on page 518

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Oracle Essbase. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter your user name and password to sign in to the server, and then select Sign In.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Search for or select an application.
   c. Search for or select a database from your application.
   d. Select the sheet tab to start your analysis.

Oracle Essbase data source example

Here is an example of an Oracle Essbase data source:
Set an accounts dimension

In some case, the accounts dimension for your data source can appear in the Dimensions area of the Data pane. This might occur if there is an error in the cube and another field is identified as the accounts dimension or there is no accounts dimension set at all. The accounts dimension defines the fields that are included as measures. To correct this error, right-click the field and then select **Set as Accounts Dimension** from the context menu.
Build views with Oracle Essbase

When Tableau is connected to an Oracle Essbase data source, there are three important features that you should know about:

Generations and Levels

In Tableau, you can work with either the generations or the levels of a dimension. The generations of a dimension are all members that are an equal distance from the root of the dimension. The levels are all members that are an equal distance from the leaves of the dimension. For balanced dimensions, you'll typically want to work with generations. However, if your dimension is ragged, then it may make more sense to navigate using levels.

By default, the generations of each dimension are listed in the Data pane. When you drag a dimension to a shelf, all generations that are ancestors of the selected generation (all generations that are above it in the hierarchy) are automatically included in the placement.

If you would rather navigate using the levels of a dimension, right-click the name of the dimension and then select **Hierarchy > Levels**.
If you are using the same dimension in multiple worksheets, you can use levels in one worksheet and generations in another worksheet simultaneously. Furthermore, you can mix generations and levels from different dimensions in the same worksheet.

Shared Members

Shared members are dimension members that appear in more than one place in a hierarchy. For example, Diet Coke might be part of the product generation. But it might be shared by both the diet colas branch and the colas branch of the product hierarchy above it. In the database, however, the data about Diet Coke is stored just once.

By default, Tableau includes shared members in all generations (or levels) of a dimension. This means that a shared member might appear multiple times in a table. If you choose to exclude shared members, they will appear only once in a table. By default, shared members are included for all dimensions. To exclude shared members for a given dimension hierarchy, right-click the dimension name in the Data pane and select Include Shared Members from the menu.

The figure below shows part of a data view where shared members are included (left) and excluded (right). Notice that diet drinks are shared members.
Includes Shared Members

The diet drinks are listed in both the Diet Drinks hierarchy and their respective drink hierarchies.

Does Not Include Shared Members

The diet drinks are only listed once, in their respective drink hierarchies.

Setting the Default Member

All multidimensional data sources have default members that are set when the data source is first built. If you find that you are creating filters all the time to look at the same specific data, you may find it useful to change the default member. For example, if you are the regional manager for the Western region in a company and you only want to look at your region’s numbers, you can set the default member to the Western region.

To change the default member in Tableau, right-click a dimension hierarchy and select Set Default Member.
In the subsequent dialog box, select from the following options:

- Default member defined on cube – uses the default member that was defined when the cube was built. This is the default setting in Tableau.

- (All) member for the hierarchy – uses the ALL member for the selected hierarchy as the default member.

- Selected member – uses the member that you select in the bottom half of the dialog box as the default member.

The default member determines how you view the cube and so is much more powerful than applying filters. All fields will be calculated based on the default member you select. In addition, these default member settings are saved with the connection.

See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.
Pivotal Greenplum Database

This article describes how to connect Tableau to a Pivotal Greenplum Database and set up the data source.

In this article

Before you begin below
Make the connection and set up the data source below
Sign in on a Mac on page 524

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Pivotal Greenplum Database. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database and the name of the database you want to connect to.
   b. Enter the user name and password.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run
Initial SQL on page 643.

d. Select Sign In.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Select and drag a table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Pivotal Greenplum Database data source example

Here is an example of a Pivotal Greenplum Database data source using Tableau Desktop on a Windows computer.
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.
PostgreSQL

This article describes how to connect Tableau to a PostgreSQL database and set up the data source.

In this article

Before you begin below
Make the connection and set up the data source below
Sign in on a Mac on page 527

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- Authentication method: Integrated Authentication or user name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select PostgreSQL. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database that you want to connect to.
   b. Enter the name of the database.
   c. Select how you want to sign in to the server. Specify whether to use Integrated Authentication or User Name and Password. If the server is password protected, and you are not in a Kerberos environment, you must enter the user
name and password.

**Note:** If you're using a Mac, and it is not attached to the domain correctly, the Mac won't know that Kerberos is being used in the domain, and the Authentication drop-down list won't be available.

Select the **Require SSL** check box when connecting to an SSL server.

d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.

e. Select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Under **Table**, select a table or use the text box to search for a table by name.

   c. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

**PostgreSQL data source example**

Here is an example of a PostgreSQL data source using Tableau Desktop on a Windows computer.
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences > Network > Advanced**, and then open the **DNS** tab.

See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.
Presto

This article describes how to connect Tableau to a Presto database and set up the data source.

Tableau connects to Presto 141t from Teradata and to Presto on-premises and Amazon EMR Presto instances for Presto version 0.148.

In this article

Before you begin below
Make the connection and set up the data source on the next page
Sign in on a Mac on page 530

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Catalog
- Authentication method:
  - Username
  - LDAP

  Note: LDAP authentication is available from Tableau Desktop version 10.3.2 forward.

- Sign-in credentials. Your choices depend on the authentication method you choose, and can include the following:
  - User name
  - Password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects
Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Presto**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server you want to connect to.
   b. Enter the catalog name.
   c. Select the **Authentication** method: **Username** or **LDAP**, and enter the credentials you're prompted for.
      Select the **Require SSL** check box when connecting to an SSL server.
   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 643.
   e. Select **Sign In**.
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.
   c. In the **Table** text box, select the search icon or enter the table name and select the search icon, and then select the table.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.
Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Presto data source example

Here is an example of a Presto data source using Tableau Desktop on a Windows computer:

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.
Progress OpenEdge

This article describes how to connect Tableau to a Progress OpenEdge database and set up the data source.

In this article

Before you begin below
Make the connection and set up the data source below

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Progress OpenEdge. For a complete list of data connections, select More under To a Server. Then do the following:
a. Enter the name of the server that hosts the database you want to connect to.

b. Enter the name of the database.

c. Enter the user name and password.

d. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.

e. Select Sign In.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can’t connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Under Table, select a table or use the text box to search for a table by name.

   c. Drag the table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Progress OpenEdge data source example

Here is an example of a Progress OpenEdge data source:
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**Salesforce**

This article describes how to connect Tableau to Salesforce.com data and set up a data source.

In this article

**Before you connect** on the next page
Before you connect

Before you begin, gather this connection information:

- Salesforce user name and password

Use this connector with Tableau Desktop on a Windows computer.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Salesforce. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter your user name and password for Salesforce.com.
   b. Select Log In.
   c. In the Allow Access dialog box, select Allow.

   If the connection is unsuccessful, verify that the authentication information is correct. If the connection continues to fail, your computer is having trouble locating the server or you may not have permission to access the data. Contact your network administrator or Salesforce administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Select a standard connection or create your own custom connection.

   Under Standard Connection, you can select from a list of predefined queries, which represent commonly used objects, and drag it to the canvas. Or, under Table, you can select a single table, which includes any custom objects your
company has created, and drag it to the canvas.

c. Select the sheet tab to start your analysis.

Refresh Salesforce data

After you click the sheet tab, Tableau imports the data by creating an extract. Tableau Desktop supports only extracts for Salesforce.

**Note:** The initial extract may take some time depending on the amount of data that is included.

After the initial extract, you can do a full or incremental refresh of the extract. Incremental refreshes for Salesforce data include any Salesforce object that has been added or changed since the last incremental refresh. Results returned are limited to the previous 30 days.

To refresh, select **Data** > [data source name] > **Extract** > **Refresh (Incremental)** | **Refresh (Full)**.

Salesforce data source example

Here is an example of a Salesforce data source:
Use Dashboard Starters

If you are connecting to your data through Tableau Online, you can use a Dashboard Starter to quickly build informative dashboard designs made specifically for key business metrics. For more information, see Dashboard Starters for Cloud-based Data on page 2252
Use joins in Salesforce

When you set up the data source on the data source page, when you drag a table to the canvas, the list of tables you can select from to join is filtered. You can create joins on any string fields and on fields that are constrained references between tables. (Only left and inner joins are supported.) You can also add joins later. For more information, see Join Your Data on page 657.

Cross-database join limitations with Salesforce

Salesforce also supports combining tables from different databases using a cross-database join. For more information, see the "Combine tables from different databases" section in Join Your Data on page 657.

When a Salesforce connection is included in a multi-connection data source, the data source won't support the following features:

- Live connections
- Extract filters
- Incremental updates to the extract
- Viewing your data in the data grid

Troubleshoot Salesforce connections

Tableau leverages the various force.com APIs to connect to Salesforce.com, Force.com, and Database.com data. These services have some restrictions on the type of data and the amount of data that you can access at a given time. If you are having trouble connecting, the following list of common restrictions may help you find a solution.

Company account and user profile configuration

In order for Tableau to connect to Salesforce data, all of the following must be enabled on both your company’s account and your user profile:

- SOAP API for signing in
- REST API for getting meta data
- BULK API for downloading objects
- REST API for downloading objects that the BULK API does not support
- Replication SOAP APIs for retrieving changes in the data
In order to optimize for performance and ensure the APIs are available for all of their customers, Salesforce.com balances the load by limiting the number of concurrent API requests as well as limiting the total number of API requests overall. An error occurs if these limits are reached while connected using Tableau. See the Salesforce.com developer documentation to learn more about these limits.

**Note:** Some editions of Salesforce may not allow API access at all. Contact your Salesforce administrator to verify that your account has access to the above list of APIs.

**Errors during extract**

When you connect to Salesforce using Tableau, the data is automatically extracted into a Tableau Data Extract file. In some cases, certain fields cannot be extracted because of character limits. Specifically, text fields that are greater than 4096 characters and calculated fields will not be included in the extract. If you have calculated fields in your data, you will need to recreate them in Tableau after creating the extract.

In addition, the Force.com API restricts queries to 10,000 total characters.

**See also**

- **Set Up Data Sources on page 646** – Add more data to this data source or prepare your data before you analyze it.

- **Build Charts and Analyze Data on page 843** – Begin your data analysis.

- **5 Ways to Maximize Your Salesforce Data** - Read the Tableau whitepaper (registration or sign in required).

**Splunk**

This article describes how to connect Tableau to Splunk data and set up a data source.

In this article

**Before you begin** on the next page

**Make the connection and set up the data source on the next page**

**Join limitations with Splunk** on page 540
Before you begin

Before you begin, gather this connection information:

- Server URL, including the name of the database, if there are multiple databases
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Splunk. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the server URL. You should use HTTPS to connect to Splunk data.
      If your server contains multiple databases, enter the name of a specific database on the server.
      Note: Database names are case sensitive.
   b. Enter your user name and password.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. Under **Saved Search**, enter a saved search name in the text box or select a saved search from the list, and drag it to the box at the top of the data source page.

c. Select the sheet tab to start your analysis.

**Splunk data source example**

Here is an example of a Splunk data source:

![Splunk data source example](image)

**Join limitations with Splunk**

While you cannot create a join between Splunk tables, you can combine Splunk data from multiple tables by doing one of the following:
• **Set up and connect to a saved search**: Set up a saved search in Splunk that returns all the tables you need in a single search result. Then connect to this saved search from Tableau Desktop.

• **Use data blending**: Set up a data source for each Splunk table you need, then use data blending to combine the data. For more information, see [Blend Your Data](#) on page 682.

• **Create an extract**: Create an extract of each Splunk table and then join the extracts together using a cross-database join. For more information, [Extract Your Data](#) on page 773 and [Combine tables from different databases](#) on page 664.

However, you can create a cross-database join between Splunk data and data from different databases (or files) with one exception. After adding a new connection to the Tableau data source, you must double-click the table in the left pane to add it to the canvas. You cannot drag the table from the left pane to the canvas as you normally would when creating a cross-database join. For more information about cross-database joins, see [Combine tables from different databases](#) on page 664.

Note: Creating a cross-database join between Splunk tables is not supported. A cross-database join between Splunk tables causes all the connections in the Tableau data source to disappear except for the last Splunk connection.

See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**SAP HANA**

This article describes how to connect Tableau to an SAP HANA database and set up the data source.

In this article

**Before you begin** on the next page
Before you begin

Before you begin, gather this connection information:

- Are you making a single-node or multi-node connection?
  - SingleNode: Name of server that hosts the database you want to connect to, and the port number if you’re using a non-default port
  - MultiNode: Names of servers and port numbers that host the databases you want to connect to

- Authentication method:
  - Windows: Windows Authentication or user name and password
  - Mac: Kerberos or user name and password

- Are you connecting to an SSL server?

- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.
Make the connection and set up the data source

1. Start Tableau and under Connect, select SAP HANA. For a complete list of data connections, select More under To a Server. Then do the following:

   a. Select the type of connection:

      - **SingleNode** - Enter the name of the server that hosts the database you want to connect to, and the port number if you're using a non-default port.
      - **MultiNode** - Enter the host name and port number of each server, separated by a comma. For example:

        host1:30015,host2:30015,host3:30015

        For more information about SAP HANA support for failover, see Configuring Clients for Failover on the SAP Help Portal.

   b. Specify how you want to sign in to the server:

      **On Windows:**

      - Select **Use Windows Authentication** if your environment supports single sign-on (SSO). For more information about single sign-on and environment requirements, see Support for SAP HANA single sign-on (SSO) on page 548.
      - Or, select **Use a specific username and password**. If the server is password protected and you are not in a Kerberos environment, you must enter the user name and password.

      **On Mac:** Select **Kerberos** or select **Use a specific username and password**.

      Select the **Require SSL** option when connecting to an SSL server.

   c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.

   d. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.

c. In the **Table** text box, select the search icon or enter the table name and select the search icon, and then select the table.

d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

   By default, column labels are displayed instead of column names.

   Use custom SQL to connect to a specific query rather than the entire data source.

   For more information, see **Connect to a Custom SQL Query** on page 723.

   **Note:** When you create a self-join with an analytic view, if one of the tables contains variables, make sure that the table with variables is on the left. Otherwise, the join may not return the expected results.

SAP HANA data source example

Here is an example of an SAP HANA data source using Tableau Desktop on a Windows computer.
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

Pass an option to the JVM on a Mac

To connect Tableau Desktop on a Mac to SAP HANA, you must have Java installed on your Mac. For more information, see SAP HANA on the Driver Download page.
If you’re familiar with how to pass a system property to the Java virtual machine (JVM), and you are using an Oracle JVM, you can use the environment variable `JAVA_OPTIONS` to override the default options.

The `JAVA_OPTIONS` variable takes a string, which is equivalent to command-line JVM arguments. The following example sets the maximum heap size to 256 MB, and specifies the path to the Kerberos configuration file:

```
export JAVA_OPTIONS="-Xmx256m -Dsun.security.krb5.conf=/etc/krb5.conf"
```

Tableau starts the JVM with the following default options:

- `-Xmx512m`

  Sets the maximum heap size to 512 MB

- `sun.security.jgss.native=true`

  Forces the JVM to delegate Kerberos authentication to the operating system library

If the default options don’t work, you can use one of the following options to configure Kerberos authentication:

- `sun.security.krb5.conf`
- `java.security.krb5.realm`
- `java.security.krb5.kdc5.conf`
- `sun.security.jgss.native`

### Customize JDBC connections on a Mac

On a Mac, SAP HANA uses a JDBC connection, which you can customize using a properties file. For more information, see Customize JDBC Connections Using a Properties File in Tableau Community.

### Install trusted SSL certificates on a Mac

If you want to use SSL/TLS to encrypt communications between Tableau Desktop and your database, you may need to add certificates to the Java certificate trust store. This is required if your database uses self-signed certificates.
1. Start the Terminal application, and navigate to your Java Home directory, which is usually located at /Library/Internet Plug-Ins/JavaAppletPlugin.plugin/Contents/Home, unless you have installed the JDK or set the JAVA_HOME environment variable.

2. Run the following command. (You might need to use the `sudo` Terminal command, which is used by administrators to execute commands as a different user, such as root).

   ```bash
   bin/keytool -importcert -keystore cacerts -alias <certificate name> -file <path to certificate file>
   ```

   You can now connect Tableau Desktop to your database using SSL/TLS.

For more information on installing trusted certificates, consult the documentation for your Java Runtime Environment.

**Select variables and input parameters**

If the table you use includes required or optional variables or parameters, the Variables and Input Parameters dialog box opens.

- Required variables and parameters display their current value or *Required.
- Optional variables and parameters display their current value or are blank.
• Prompt for the variable when the workbook is opened by selecting the check box under Prompt.

Select a variable or parameter and type or select a value for it. Repeat for all required values and any optional values that you want to include, and then select OK.

**Note:** You can use SQL query-based SAP HANA prompts.

**Support for SAP HANA single sign-on (SSO)**

When SAP HANA is configured to support single sign-on (SSO), after you sign in to the SAP HANA server, you can access data, and publish data sources and workbooks to Tableau Server, without having to re-enter your user name and password. And, you can publish a data source or workbook so that other users with SSO can access the published data sources and workbooks without having to enter their user names and passwords.

**Important:** Your environment must be correctly configured to support SSO for SAP HANA:

- Tableau Desktop requires SAP HANA driver version 1.00.85 and later.
- Tableau Server must be configured to support SSO for SAP HANA. For information, see Configure SAP HANA for Single Sign-On in the Tableau Server Help.

**Sign in to the server**

When SSO is set up in your environment, to sign in to SAP HANA, simply enter the server name, select **Use Windows Authentication**, and then click **Sign In**.

**Publish to the server**

In an SSO environment, users don’t have to enter their credentials to access a data source or a workbook if you publish it to Tableau Server with the correct authentication mode.

To enable SSO when you publish a data source or workbook to Tableau Server, follow these steps:

1. Select **Server > Publish Data Source** or **Server > Publish Workbook**.
2. Under Data Sources, click **Edit**.
3. In the Manage Data Sources dialog box, do the following:
   1. Select the **Publish Type: Embedded in workbook** or **Published separately**.
2. Select Viewer credentials for Authentication.

4. Click Publish.

How to refresh HANA extracts when you use SSO

When you use single sign-on (SSO) to connect to SAP HANA, because of authentication restrictions, you can't schedule an extract refresh when you publish the data source or workbook to a server. You can, however, use the Tableau Data Extract Utility to automate the extract refresh task.

For example, the following command refreshes an SAP HANA extract named "mydatasource" that has been published to Tableau Server. This command specifies the following:

- The name of your Tableau Server
- The name of the data source to refresh

C:\Program Files\Tableau\Tableau 10.2\bin>tableau refreshextract --server https://mytableauserver --datasource mydatasource

For more information about the utility, see Tableau Data Extract Command-Line Utility on page 806.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

SAP NetWeaver Business Warehouse

This article describes how to connect Tableau to SAP NetWeaver Business Warehouse (BW) data and set up the data source.

In this article

Install SAP software on the next page
Before you connect on the next page
Use this connector with Tableau Desktop on a Windows computer.

Install SAP software

Before you can connect Tableau to SAP BW data, you must install SAP software on your computer. For instructions on how to install SAP software, see Connecting to SAP BW on Tableau Community.

Before you connect

Before you connect, gather this connection information:

- Connection name
- Authentication: User name and password, or integrated authentication
- (Optional) Client for BW system
- (Optional) Language

If you want to use single sign-on, be sure that your SAP personal security environment (PSE) is configured for single sign-on.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select SAP NetWeaver Business Warehouse. For a complete list of data connections, select More under To a Server. Then do the
following:

a. Select a connection from the drop-down list.

A connection appears in the drop-down list if a system entry exists for it. You can create a new system entry using the SAP Logon utility, or contact your database administrator.

b. Select Authentication method:

**Username and Password** - Enter the user name and password to sign in to the server.

**Integrated Authentication** - Select this option if your environment is configured for single sign-on (SSO). This option is configured by the Tableau Server administrator. For more information, see Enable Single Sign-On for SAP BW in the Tableau Server Help.

c. Optionally, enter the **Client ID** for the BW system and the **Language**.

d. Select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can’t connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. Search for or select a catalog or InfoProvider.

c. Search for or select a cube or query.

d. Select the sheet tab to start your analysis.

**SAP BW data source example**

Here is an example of an SAP BW data source:
Support for SAPUIILandscape.xml

Prior to version 7.40 of the SAP GUI tools, SAP BW connections are defined on client computers in the saplogon.ini file. When you add a connection definition using SAP Logon, that definition is added to the saplogon.ini file. Tableau reads from that file to get the list of SAP BW systems to show in the **Select connection** drop-down list in the Tableau connection dialog box.
Starting in version 7.40 of the SAP GUI tools, the format of the connection definition file has changed to XML, and this connection information is now stored in SAPUILandscape.xml and SAPUILandscapeGlobal.xml.

Tableau supports using landscape XML files located on a remote HTTP server.

Connections in drop-down list might not work

If you update to version 7.40 of the SAP GUI tools and then have issues with the Select connection drop-down list where connections don’t work in Tableau, it might be because Tableau doesn’t support Messageserver elements in the SAPUILandscape.xml and SAPUILandscapeGlobal.xml file. If a connection definition references a message server, the connection will display in the Select connection drop-down list, but it won’t connect.

Select variables and input parameters

If the table you use includes required or optional variables or parameters, the Variables and Input Parameters dialog box opens.

- Required variables and parameters display their current value or *Required.
- Optional variables and parameters display their current value or are blank.
• Prompt for the variable when the workbook is opened by selecting the check box under **Prompt**.

Select a variable or parameter and type or select a value for it. Repeat for all required values and any optional values that you want to include, and then select **OK**.

**Select the alias for a dimension**

When you connect to SAP NetWeaver Business Warehouse (SAP BW) databases, you have the option to select which alias to use for dimensions. SAP BW offers a number of different options including long name, short name, medium name, and key. Sometimes the alias used by default is not very useful. You can right-click the dimension and change it to another option such as **Short name**, which changes the labels to something more meaningful.
Support for SAP BW extracts

Starting in 10.4, you can create an extract of SAP BW data without the need for a special license key from Tableau. Prior to 10.4, SAP BW extracts were an unsupported beta feature. If you’re using this beta feature, license keys will be extended through 2017, however, once the keys expire or to get support for this feature, you must upgrade to version 10.4.

When you create an extract from an SAP BW data source, Tableau converts data from the cube into a relational extract. This conversion can sometimes take a while based on the size of the cube. In addition, much of the cube metadata cannot be represented in the relational data model, so there are some limitations when working with extracts created from an SAP BW data source in Tableau Desktop. (Note that beta extract functionality was not supported in Tableau Desktop 9.0.)

SAP BW extract features

The following features are included in SAP BW extract functionality:

- The behavior of empty user filters has changed to be consistent with the behavior on other data sources. Using an empty user filter no longer hides the dimension.
- You can use Hide to hide dimensions before you extract data or refresh an extract. This reduces the size of the extract and the amount of time to perform an extract.
- Hiding a dimension takes priority over creating a user filter on the dimension.
- You can rename dimensions and measures in the data grid before extracting data.
- Tableau modifies the extract query slightly to fix some of the [cube contains no data to transform] errors.

SAP BW extract limitations

- Tableau supports only infocubes, or InfoProviders that contain only infocubes. A composite source that contains other BW object types is not supported.
- For large extracts, Tableau automatically partitions the query if the query fails or times out.
- BEx queries that contain more than 49 dimensions are not supported by SAP BW.
- You can’t switch between a live connection and the extract. When you connect to an SAP BW data source, you’ll have the option to Connect Live or Import the data. If you choose to Connect Live, the Extract options that are typically available for a connection are not an option. Further, if you choose to Import that data and create an extract, you cannot
You can't replace an extract with a live connection to SAP BW.

Incremental extracts are not supported.

When creating the extract, you can add filters to limit the data included in the extract, but these filters can be based only on hierarchies.

Tableau doesn't support extracting the following SAP BW Objects:

- Calculated sets.
- Custom formatting and cell properties, except aliases.
- User hierarchies. Hierarchies will collapse.
- Pre-aggregated data.
- Exception aggregation.
- Compounding attribute.

Data displays differently for extract and live connections

In addition to the above limitations, you may notice some differences between an extract connection and live connection to SAP BW data. Specifically, when you connect to SAP BW directly, the dimensions, hierarchies, and levels are represented in the Data pane. After you create an extract, all of the fields are shown in a flat list.

| Live connection | Extract connection |
When you connect to an SAP BW data source and create an extract or connect to a Tableau Data Source file (.tds) that connects to an SAP BW extract, the fields display in the Data pane as expected. However, if you connect directly to the extract file, you will see extra fields that represent the alias options. All aliases are included in the extract and are typically hidden. You can hide the alias variants that you are not using by right-clicking the field and selecting Hide.

Aliases hidden

Aliases showing
Frequently asked questions

Requirements

What versions do I need?

- For the client machine where Tableau Desktop or Tableau Server is installed: SAP GUI 7.30 or later for Windows.
- For the server machine: SAP BW 7.40 or later.
- Tableau Desktop or Tableau Server, any supported version. For information about which versions are supported, see Supported Versions on the Tableau website.

Do I need any special licenses like OpenHub to use Tableau with SAP BW?

Tableau uses the SAP OLE DB for OLAP provider (part of the Open Analysis Interfaces) to interact with SAP BW. This provider is a mechanism to do 'reporting' and retrieving 'reasonable' amounts of data. Unlike OpenHub, the interface is not designed for moving large amounts of data. You should verify with SAP that you have licensed SAP BW correctly for your environment and that your licensing terms permit the use of the OLE DB for OLAP provider for tools like Tableau.

Does the SAP BW connector require HANA?

No. The Tableau SAP BW connector does not require HANA. The connector will work with any of the supported SAP BW backend databases such as Oracle and SQL Server. However, you may use HANA if you prefer.
Connection

Is the connection between Tableau and SAP BW a live connection or is it a Tableau extract?

Tableau supports live and extract connections to SAP BW data.
With a live connection you will always see the latest data in your workbooks and dashboards.

Does Tableau support connecting directly to the underlying database tables that support the BW instance?

No. Tableau does not support this method for connecting to the SAP BW system.

Performance

How does performance compare to existing tools we use?

System performance should be comparable to existing tools. If performance is an issue in your environment, you might want to consider adopting SAP HANA in your environment. For more information, see SAP HANA on the SAP website.

Security

How does Tableau handle user security?

Tableau leverages all the security policies already implemented in the SAP BW server. There is no need for you to duplicate this in Tableau. All user authentication and authorization is performed by SAP BW. Tableau passes your credentials to the SAP BW system to be authenticated. Once authenticated, you will see only the objects and data that your profile or role on the SAP BW system has been enabled to access.

How does user security work with Tableau Server when I publish a dashboard?

By default, worksheets and dashboards connected to SAP BW data require you to log into SAP BW with your SAP user name and password. However, Tableau Server can be configured to embed credentials so that you are taken directly to the published views and dashboards. In addition, Tableau Server offers an additional layer of security if desired. Tableau Server administrators can configure user authentication and authorization, independent and in
addition to the SAP BW authentication and authorization. For more information about Tableau Server security, refer to the Security section of the Tableau Server Online Help.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

SAP Sybase ASE

This article describes how to connect Tableau to an SAP Sybase ASE database and set up the data source.

Tableau connects to Sybase ASE version 15.2 and later.

In this article

Before you begin below
Make the connection and set up the data source on the next page

Before you begin

Before you begin, gather this connection information:

- Name of the server you want to connect to, port number, and if you use single sign-on, service name
- Authentication method: integrated authentication (single sign-on), or user name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.
Make the connection and set up the data source

1. Start Tableau and under Connect, select SAP Sybase ASE. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server you want to connect to, and specify the port to use. Optionally, enter the service name. If you use single sign-on, you must enter the service name.
   b. Select the Authentication method:
      
      **Integrated Authentication** - Select this option if your environment is configured for single sign-on (SSO).
      
      **Username and Password** - Enter the user name and password to sign in to the server.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.
      
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Database drop-down list, select a database or use the text box to search for a database by name.
   c. From the Schema drop-down list, select a schema or use the text box to search for a schema by name.
   d. Under Table, select a table or use the text box to search for a table by name.
      
      You can also specify a stored procedure in the database. To access stored procedures, the database must be appropriately configured with a linked server,
which could be a remote server or a pointer to the database itself (known as *loopback*). For more information, see Use a Stored Procedure on page 733. 

e. Drag a table or stored procedure to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

**Note:** Kerberos authentication for SAP Sybase ASE doesn’t support delegation. This means that if you selected **Integrated Authentication**, you can’t use **Viewer credentials** as the **Authentication** method when you publish a workbook or data source to Tableau Server; you can only use **Server Run As account**, and the Tableau Desktop Run As User account must have an account in the SAP Sybase ASE database.

### SAP Sybase ASE data source example

Here is an example of an SAP Sybase ASE data source:
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**SAP Sybase IQ**

This article describes how to connect Tableau to an SAP Sybase IQ database and set up the data source.

In this article

<table>
<thead>
<tr>
<th>Before you begin below</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Make the connection and set up the data source</strong> on the next page</td>
</tr>
</tbody>
</table>

**Before you begin**

Before you begin, gather this connection information:

- Name of the host that hosts the database you want to connect to

- (Optional) Server name. Tableau supports Sybase IQ multiplexes, so a server name is not required.

- Authentication method: Windows Authentication or user name and password

- (Optional) Initial SQL statement to run every time Tableau connects

**Use this connector with Tableau Desktop on a Windows computer.**

**Driver required**

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the **Driver Download** page where you can find driver links and installation instructions.
Make the connection and set up the data source

1. Start Tableau and under Connect, select SAP Sybase IQ. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the host name that hosts the database that you want to connect to.
   b. (Optional) Enter the server name.
   c. Select how you want to sign in to the server. Specify whether to use Windows Authentication or a specific user name and password. If the server is password protected, you must enter the user name and password.
   d. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   e. Select Sign In.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Database drop-down list, select a database or use the text box to search for a database by name.
   c. Under Table, select a table or use the text box to search for a table by name.
   d. Drag a table to the top area of the data source page, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

SAP Sybase IQ data source example

Here is an example of an SAP Sybase IQ data source:
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**ServiceNow ITSM**

This article describes how to connect Tableau to ServiceNow IT Service Management (ITSM) data and set up the data source.
Before you begin below
Make the connection and set up the data source below
Use Dashboard Starters on the next page
Key considerations on the next page
Use a local ServiceNow account on page 568
Date range selections can impact performance on page 568
Troubleshoot data access on page 569

Before you begin

Before you begin, gather this connection information:

- ServiceNow instance you want to connect to
- User name and password for a local ServiceNow account

Connection requirements

To connect Tableau to ServiceNow, make sure that the following requirements are met:

- **Use a local ServiceNow account.** The ServiceNow API doesn’t support single sign-on (SSO), so the Tableau connector can only authenticate using a local ServiceNow account. For more information, see **Use a local ServiceNow account** on page 568.

- **Have the correct permissions.** Access control lists (ACLs) must provide the user the permission required to access the data. For more information see **Troubleshoot data access** on page 569.

- **For on premises, CORS is configured.** Before you connect to an on premises instance of ServiceNow, you must configure CORS for Tableau in ServiceNow. For more information, see Configure CORS for Tableau on ServiceNow in Tableau Community.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **ServiceNow ITSM**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Select **Instance Location: Cloud-based** or **On Premises**.
   b. Enter your ServiceNow instance, user name, and password for a local ServiceNow
account.

Note: When you select On Premises, you must enter a full URL for the ServiceNow instance.

c. Select Sign In.

d. Select Filter Type: Relative date range or Fixed date range, and then select or specify the range.

e. Select Connect.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network or ServiceNow administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Under Table, select a table and drag it to the top of the canvas.

   c. Select the sheet tab to start your analysis.

   After you select the sheet tab, Tableau imports the data by creating an extract. Note that Tableau Desktop supports only extracts for ServiceNow. You can update the data by refreshing the extract. For more information, see Extract Your Data on page 773.

   Creating extracts may take some time depending on the amount of data that is included.

Use Dashboard Starters

If you are connecting to your data through Tableau Online, you can use a Dashboard Starter to quickly build informative dashboard designs made specifically for key business metrics. For more information, see Dashboard Starters for Cloud-based Data on page 2252

Key considerations

ServiceNow enforces an account's access permissions

ServiceNow accounts that have the Admin role can retrieve encrypted password values from the User table. This file will be null for accounts with less permissions. Because ServiceNow
enforces the access permissions assigned to the account, be aware that there might be other sensitive fields in your data that an Admin can access.

Custom tables not supported

The Tableau connector for ServiceNow ITSM supports the ITSM service, which doesn't include custom tables.

Display value fields aren't included in the data source

When you connect to a table that contains fields with display values (fields with a dv_ prefix), those fields aren't included when you create a data source. You can add the display values by joining to the appropriate related tables that contain those values.

Use a local ServiceNow account

You must use a local account to connect Tableau to your data. If you don't have a local ServiceNow account, we recommend that you ask your administrator to implement one of the following solutions:

- Create a local account and share it.
- Create a local account and temporarily share it long enough to extract the data and publish it to Tableau Online or Tableau Server. Then the administrator can save the new password and schedule refreshes.
- Create a local account on a non-production (“staging”) instance and use it to create a data source. Give the data source to the administrator to edit the connection to point to the production instance, publish the data source to Tableau Online or Tableau Server, and schedule for data refreshes.
- Ask the administrator to publish the data sources you need.

Date range selections can impact performance

When you apply a date filter, it's tempting to gather as much data as possible when you do an analysis, however, retrieving records from ServiceNow ITSM can be time-consuming. Tableau doesn't know how much data there is in a particular date range until it retrieves the data. For this reason, you should restrict your date range at first, and then expand after you evaluate performance. Date filters are applied to these tables: metric_instance, change_request, incident, task, problem.
To give you a rough idea of how much time it might take to retrieve data from ServiceNow ITSM, tests were conducted using a high-speed connection. This table shows how long it took in the test environment to retrieve a given number of records.

<table>
<thead>
<tr>
<th>Number of records</th>
<th>Time to retrieve</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td>11 seconds</td>
</tr>
<tr>
<td>10,000</td>
<td>2 minutes</td>
</tr>
<tr>
<td>100,000</td>
<td>18 minutes</td>
</tr>
</tbody>
</table>

Troubleshoot data access

For ServiceNow ITSM users

If you see an error when you try to access data, for example, "You are not authorized to access this resource...," contact your ServiceNow administrator.

For ServiceNow administrators

ServiceNow uses access control lists (ACLs) to specify who has access to which data. Often the default permissions that are set for ServiceNow fulfillers are adequate for Tableau users to access the content they need to analyze ServiceNow ITSM data.

Tableau users need to access metadata about tables in ServiceNow, which is available in the following tables:

- sys_dictionary table (metadata about columns)
- sys_db_object table (metadata about tables)
- sys Glide_object table (metadata about field types)

If a Tableau user can’t access ServiceNow ITSM data, make sure that the ACLs provide the user table level and field level read access to the tables above, and that there are no scripts that are locking down access.

To grant access to the metadata tables, as a best practice, you might consider creating a role with the ACLs needed to access the metadata tables, and then assign the role to a Tableau User group. Then, you can assign Tableau users to the group so that users can access the data needed for analysis in Tableau.

For information on the creation of an ACL, please see the ServiceNow Access control list rules on the ServiceNow website.
See also

**Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

**Build Charts and Analyze Data** on page 843 – Begin your data analysis.

**SharePoint Lists**

This article describes how to connect Tableau to SharePoint lists and set up the data source.

In this article

- **Before you connect** below
- **Make the connection and set up the data source** on the next page
- **Connection Tips** on page 572
- **Sign in on a Mac** on page 573
- **Refresh your data** on page 573

**Before you connect**

Before you begin, gather this connection information:

- The URL to the SharePoint site that contains the lists you want to connect to
- The edition of your SharePoint server: on premise or SharePoint Online
- The method of authentication: user name and password, third-party SSO, or integrated authentication
- The user name or email address, password, and (optionally) SSO domain, if needed

**Driver required**

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the **Driver Download** page where you can find driver links and installation instructions.
Make the connection and set up the data source

1. Start Tableau and under Connect, select SharePoint Lists. For a complete list of data connections, select More under To a Server. Then do the following:
   a. In the SharePoint Site text box, enter the SharePoint URL. Don’t include the list name in the URL.
   b. Select the Edition: On Premise or SharePoint Online.
   c. Select the Authentication method: Username and Password, Third-party SSO, or Integrated Authentication.
   d. If the fields are available, enter the Username and Password, and optionally, enter SSO Domain.
   e. Select Sign In.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under Table, select a table and drag it to the top of the canvas.
   c. Select the sheet tab to start your analysis.

SharePoint lists data source example

Here is an example of a SharePoint lists data source using Tableau Desktop on a Windows computer:
Connection Tips

- If you use Kerberos, select **Integrated Authentication** as your **Authentication** type.
- If you use NTLM, select **Username and Password** as your **Authentication** type.
- If you select **Third-party SSO** as your **Authentication** type, enter your credentials, which might be your user name or email address, and your password. Optionally, enter the **SSO Domain**.

**Note**: Tableau currently supports only **Okta** and **OneLogin** for Third-party SSO connections.

- Any list you connect to will be the existing default SharePoint view only. This means that if any rows of data are filtered on the default view in SharePoint, you will not be able to
retrieve the filtered data. If your goal is to access everything in the SharePoint list then you will need to adjust the default view to show all rows.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

Both Authentication methods on a Mac require you to enter your user name and password to connect.

Refresh your data

Tableau supports only extract connections for SharePoint lists, not live connections. You can update the data by refreshing the extract; incremental refreshes are not supported. For more information, see Refresh Extracts on page 798. For information about refresh schedules, see Schedule Extract Refreshes as You Publish a Workbook on page 2540.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Snowflake

This article describes how to connect Tableau to a Snowflake data warehouse and set up the data source.

In this article

Before you begin on the next page
Make the connection and set up the data source on the next page
Before you begin

Before you begin, gather this connection information:

- Name of the server name that you want to connect to
- Authentication method:
  - Username and Password
  - SAML IdP
- Sign in credentials depend on the authentication method you choose and can include the following:
  - User name and password
  - URL for the SAML IdP server
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Snowflake. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that you want to connect to.
   b. Select the Authentication method: Username and Password or SAML IdP.
   c. Enter the information that you are prompted to provide. The information you are prompted for depends on the authentication method you choose. If you select SAML IdP, in the SAML IdP (Okta) field, enter the URL for the SAML IdP server.
   d. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run...
Initial SQL on page 643.

e. Select Sign In.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the Warehouse drop-down list, select the warehouse or use the text box to search for a warehouse by name.

c. From the Database drop-down list, select a database or use the text box to search for a database by name.

d. From the Schema drop-down list, select a schema or use the text box to search for a schema by name.

e. Under Table, select a table or use the text box to search for a table by name.

f. Drag a table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.
Build Charts and Analyze Data on page 843 – Begin your data analysis.

Best Practices for Using Tableau with Snowflake on the Snowflake website (registration required).

Spark SQL

This article describes how to connect Tableau to a Spark SQL database and set up the data source. Tableau can connect to Spark version 1.2.1 and later.

You can use the Spark SQL connector to connect to a Spark cluster on Azure HDInsight, Azure Data Lake, Databricks, or Apache Spark.

In this article

Before you begin below
Make the connection and set up the data source on the next page
Sign in on a Mac on page 579

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to and port number
- Authentication method:
  - No Authentication
  - Kerberos
  - User Name
  - User Name and Password
  - Microsoft Azure HDInsight Service
- Transport. Your choices depend on the authentication method you choose, and include the following:
  - Binary
  - SASL
  - HTTP
- Sign-in credentials. Your choices depend on the authentication method you choose, and can include the following:
  - User name
  - Password
  - Realm
  - Host FQDN
  - Service Name
  - HTTP Path
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

**Driver required**

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

**Make the connection and set up the data source**

1. Start Tableau and under Connect, select Spark SQL. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database and the port number to use.
   b. Connect to the database using SparkThriftServer. Note that the legacy SharkServer and SharkServer2 connections are provided for your use, but are not supported by Tableau.
   c. Select the Authentication method. Then, based on your selection, enter the information you are prompted for.
      - If the server is password protected, and you are not in a Kerberos environment, you must enter the user name and password.
   d. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
e. Select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.

c. In the **Table** text box, select the search icon or enter the table name and select the search icon, drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 723.

**Note:** Kerberos authentication for Spark SQL doesn't support delegation. This means that you can't use **Viewer credentials** as the **Authentication** method when you publish a workbook or data source to Tableau Server; you can only use **Server Run As account**.

Spark SQL data source example

Here is an example of a Spark SQL data source using Tableau Desktop on a Windows computer.
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.
Teradata

This topic describes how to connect Tableau to a Teradata database or a Teradata Unity server and set up the data source.

In this article

Before you begin

Before you begin, gather this connection information:

- For Teradata database: Name of the server you want to connect to
- For Teradata Unity server: URL of the server you want to connect to
- Authentication method: Teradata Database, LDAP, or Integrated Authentication
- Depending on the environment, the user name and password
- Do you need an encrypted connection?
- (Optional) Query banding and initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Teradata. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database you want to connect to.

   **Note:** You can connect to a Teradata Unity server by entering the appropriate URL in the Server text box.
b. Select how you want to sign in to the server. Specify whether to use the built-in Teradata Database authentication, LDAP, or Integrated Authentication. If the server is password protected, and you are not in a Kerberos environment, you must select Teradata Database or LDAP and enter the user name and password.

c. Select the Require Encryption check box if you need an encrypted connection.

d. (Optional) Select Query Banding and Initial SQL.

e. Select Sign In.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the Database drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the database.

c. In the Table text box, select the search icon or enter the table name and select the search icon, and then select the table. You can also specify a stored procedure in the database. For more information and for a list of constraints specific to Teradata databases, see Use a Stored Procedure on page 733.

d. Drag the table or stored procedure to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Teradata data source example

Here is an example of a Teradata data source using Tableau Desktop on a Windows computer.
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences > Network > Advanced**, and then open the **DNS** tab.

Use query bands to increase performance

When you connect to a Teradata database, you can optionally define query band statements that run during connection. These statements can increase performance, and they take advantage of the built-in security rules of the database.

Using query bands, you can pass parameters into the Teradata environment. Use these to set up a workbook to filter the data based on security rules that exist in the database. For example,
you can pass in the Tableau Server user name for the current user so that when the view is loaded it only shows the data specific to that user. Query bands can also be used to improve performance. When connecting to Teradata, you can define a map between the name of the attributes passed into the query band and the corresponding values from Tableau.

**To set up query banding:**

1. On the data source page, select Data > Query Banding and Initial SQL.
2. In the subsequent dialog box, specify name/value pairs in the top text box labeled Query Banding. You can use the Insert drop-down menu to add Tableau values. The Tableau values are described in the table below.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;TableauMode&gt;</code></td>
<td>The mode Tableau is operating in when generating queries. This value will either be “Connect” when retrieving metadata or “Analytical” when retrieving actual data.</td>
<td>Connect or Analytical</td>
</tr>
<tr>
<td><code>&lt;LoginUser&gt;</code></td>
<td>The user name of the person signed in to the database.</td>
<td>jsmith</td>
</tr>
<tr>
<td><code>&lt;ServerUser&gt;</code></td>
<td>The signed-in Server user. Does not include domain name.</td>
<td>jsmith</td>
</tr>
</tbody>
</table>

**Note:**

ProxyUser returns the same value
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>as</td>
<td>ServerUser. However, ProxyUser sets up impersonation and stores the Tableau Server user in the query band parameter. If you are using query banding for security purposes, you should use ProxyUser instead. This will ensure that query results are not shared between different users. ServerUser should only be used for auditing purposes.</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
</tbody>
</table>
| `<ServerUserFull>` | The signed-in server user including the domain name (if the server is using Active Directory).  

**Note:**  
ProxyUser Full returns the same value as ServerUser Full. However, ProxyUser Full sets up impersonation and stores the Tableau Server user in the query band parameter. If you are using query banding for security purposes, you should use ProxyUser Full. | domain.lan\jsmith |
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>instead. This will ensure that query results are not shared between different users.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ServerUserFull should only be used for auditing purposes.</td>
<td></td>
</tr>
<tr>
<td><code>&lt;ProxyUser&gt;</code></td>
<td>Used when setting up impersonation on the server. Provides the username of the current server user.</td>
<td>jsmith</td>
</tr>
<tr>
<td><code>&lt;ProxyUserFull&gt;</code></td>
<td>Used when setting up impersonation on the server. Provides the username and domain name of the current server user.</td>
<td>domain.lan\jsmith</td>
</tr>
<tr>
<td><code>&lt;TableauApp&gt;</code></td>
<td>The name of the Tableau application.</td>
<td>Tableau Desktop Professional or Tableau Server</td>
</tr>
<tr>
<td><code>&lt;TableauVersion&gt;</code></td>
<td>The version of the Tableau application.</td>
<td>6100.11.0428.0300</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>&lt;WorkbookName&gt;</td>
<td>The name of the workbook</td>
<td>Financial-Analysis</td>
</tr>
</tbody>
</table>

An example query band statement is shown below. This example passes the user name for the current server user.

```
ApplicationName=<TableauApp>
Version=<TableauVersion>
ProxyUser=<ProxyUser>
TableauMode=<TableauMode>
```

Tableau checks the statement for errors as you type. When it is valid a green check mark displays at the bottom of the text box.
Use initial SQL commands

When you connect to a Teradata database, you can optionally specify a SQL command that will be run once upon connection. See Run Initial SQL on page 643 to learn more about adding these commands to your connection.

Note: If Tableau detects that the Initial SQL payload cannot be delivered in one query, it attempts to split the query into separate statements, where each statement must end with a semicolon followed by a newline character, with no additional characters (such as an extra space) in between. This requirement is important because if a semicolon exists within the interior of a statement, for example within a string literal, it may not be safe to split the statement at that point. If the Initial SQL code that you specify in Tableau returns an error, and the same Initial SQL is deemed valid by the Teradata databases, it may be that Tableau has split the query incorrectly. If this happens, try reformatting the SQL code in Tableau manually.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Tableau and Teradata: The visual approach to the active warehouse - Read the Tableau whitepaper (registration or sign in required).

Teradata OLAP Connector

This article describes how to establish a Teradata OLAP connection.

In this article

Before you begin on the next page
Make the connection and set up the data source on the next page
Work with Teradata OLAP data on page 591
Before you begin

Before you begin, gather this connection information:

- Connection name
- User name and password

Use this connector with Tableau Desktop on a Windows computer.

When you connect to Teradata OLAP data, you select from a list of connections, each representing a system data source name (DSN). A connection appears in the drop-down list if a system DSN exists for that connection. You can use the Windows ODBC Data Source Administrator utility to create a system DSN, or contact your database administrator.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under Connect, select Teradata OLAP Connector. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Select a connection from the drop-down list.
   b. Enter your user name and password to log on to the server.
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
   c. Select Sign In.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Search for or select a catalog.
c. Search for or select a cube from the catalog.

d. Select the sheet tab to start your analysis.

Teradata OLAP connection example

Here is an example of a Teradata OLAP connection:
Work with Teradata OLAP data

Named sets from a Teradata OLAP data source are displayed in the Sets area of the Data pane in Tableau. You can interact with these named sets in the same way you interact with other custom sets in Tableau. For more information, see Create Sets on page 1004. You can view underlying data for Teradata OLAP data sources, provided the database administrator has enabled this functionality. For more information, see View Underlying Data on page 2616.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

TIBCO Data Virtualization

This article describes how to connect Tableau to a TIBCO Data Virtualization (formerly called Cisco Information Server) virtual database and set up the data source.

In this article

Before you begin below
Make the connection and set up the data source on the next page

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the data you want to connect to
- (Optional) Domain name
- Datasource name
- Authentication method: LDAP or user name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.
Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **TIBCO Data Virtualization**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the data you want to connect to.
   b. (Optional) Enter the name of the domain.
   c. Enter the name of the **Datasource** you want to connect to.
   d. Select how you want to sign in to the server. Specify whether to use **LDAP** or a user name and password.
   e. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 643.
   f. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the **Catalog** drop-down list, enter the catalog name in the text box, or select the catalog from the list.
   c. From the **Schema** drop-down list, enter the schema name in the text box, or select the schema from the list.
   d. Under **Table**, enter the table name in the text box, or select the table from the list.
e. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.

Vertica

This article describes how to connect Tableau to a Vertica database and set up the data source.

In this article

Before you begin
Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.
Make the connection and set up the data source

1. Start Tableau and under Connect, select Vertica. For a complete list of data connections, select More under To a Server. Then do the following:
   a. Enter the name of the server that hosts the database and the name of the database that you want to connect to.
   b. Enter the user name and password.
   c. (Optional) Select Initial SQL to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see Run Initial SQL on page 643.
   d. Select Sign In.
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the Schema drop-down list, select a schema or use the text box to search for a schema by name.
   c. Under Table, select a table or use the text box to search for a table by name.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.
      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see Connect to a Custom SQL Query on page 723.

Vertica data source example

Here is an example of a Vertica data source using Tableau Desktop on a Windows computer:
Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to System Preferences > Network > Advanced, and then open the DNS tab.

See also

Set Up Data Sources on page 646 – Add more data to this data source or prepare your data before you analyze it.

Build Charts and Analyze Data on page 843 – Begin your data analysis.
Vertica & Tableau Solution Overview - Read the Tableau whitepaper (registration or sign in required).

Vertica Integration with Tableau: Tips and Techniques: Updated for Tableau 10! - Read the Vertica knowledge base article on the Vertica website.

Web Data Connector

This article describes web data connectors, what to do before you use a connector, and how to connect Tableau to a web data connector.

In this article

- Use a web data connector below
- Test and vet the web data connector below
- How to use a web data connector on Tableau Server and Tableau Online below
- Connect to the data source on the next page
- Optional settings on page 599
- Tableau support for Web Data Connector on page 600

Use a web data connector

You can use a web data connector to connect to data that is accessible over HTTP and that doesn't already have a connector. A web data connector is an HTML file that includes JavaScript code. You can create your own web data connector or use one that has been created by someone else. The web data connector must be hosted on a web server running locally on your computer, on a third-party web server, or on Tableau Server.

Test and vet the web data connector

You should use a web data connector that you trust. If you are unfamiliar with the web data connector, you or your Tableau Server administrator should test and vet the web data connector before you use it. For more information, see Testing and Vetting Web Data Connectors in the Tableau Server Help.

How to use a web data connector on Tableau Server and Tableau Online

When you use a web data connector, Tableau creates an extract of the data that the connector accesses. You can refresh the extract in Tableau Desktop.
When you publish the data source or workbook to Tableau Server, you cannot refresh the extract on Tableau Server unless the web data connector is added to the safe list on Tableau Server. If the connector requires credentials to sign in to the web-based data source, you need to ensure that the credentials are embedded with the data source. For more information, see Web Data Connectors in Tableau Server in the Tableau Server Help.

After the web data connector has been tested and vetted, the server administrator can add the URL for the connector to the safe list on the server. The server administrator should give this URL to you and to anyone else who wants to use the connector in a workbook.

If you open a workbook on Tableau Server that was created using a web data connector, but the connector has not been added to the safe list on Tableau Server, and you want to be able to refresh the extract on Tableau Server, follow the process for testing, vetting, and adding the connector to the safe list. If the connector requires credentials to sign in, you need to ensure that the credentials are embedded with the data source. You can then refresh your data on Tableau Server.

When you publish to Tableau Online, as a security measure, Tableau Online can't connect to or refresh an extract created by a web data connector. To refresh some web data connector extracts, you can use Tableau Bridge. For more information, see When to use Tableau Bridge to keep data fresh in the Tableau Online Help.

When you publish to Tableau Public, because you can't add a web data connector to Tableau Public, you can't refresh web data connector extracts directly on Tableau Public.

Connect to the data source

1. Start Tableau and under Connect, select Web Data Connector. For a complete list of data connections, select More under To a Server.

2. Enter the URL of the web data connector that you want to connect to, and then press Enter.

   **Note:** If you enter a URL to a website, rather than to a web data connector, an error message will appear.

3. If the connector displays a webpage, enter any information that you're prompted for, and
then submit the page.

4. Wait while the connector retrieves your data and imports it into Tableau as an extract.

5. Select the sheet tab to start your analysis.

Web data connector data source example

An example of a web data connector data source is shown below. The first time you connect to a web data connector, the connection page outlines how to connect to the web data connector.

After you connect to a web data connector, on the connection page, you can do the following:

- View a history of all the web data connectors that you've connected to, or clear the history, in the drop-down list in the address bar.
- View your five most recently used connectors under Recent Connectors.
- Select links at the bottom on the window to learn more about web data connectors:
  - **What's this?** links to an overview of this feature.
  - **Use a connector** links to this topic.
  - **Build a connector** links to the Web Data Connector SDK, which provides tools and documentation for building a web data connector.
Optional settings

After you connect, you can make the following configuration changes to the data source before you start your analysis.

- **Sort fields and rows** – From the **Sort fields** drop-down list, select how you want to sort the columns in the grid or metadata grid. Sort the row values by selecting the sort
button next to the column name.

- **Rename or hide columns** – Select the column header drop-down arrow and select the option you want.

- **Split columns** – Depending on how the data is structured, you can split the columns in your data into new fields. For more information, see *Split a Field into Multiple Fields* on page 761.

- **Create calculations** – Create a new calculation based on an existing field in the data source.

- **Copy values** – Copy values in the grid by selecting the values and then pressing Ctrl+C. Alternatively, to copy values in the metadata grid, select the values, right-click, and then select *Copy*.

- **Add new or edit other data sources** – Select the drop-down arrow next to the data source.

Tableau support for Web Data Connector

Tableau provides limited customer support for connections using the **Web Data Connector**. Tableau does not provide support for connectors or for other programs written to interface with the WDC API. However, you can submit questions and ask for help on the Tableau developer community forums.

Tableau *does* provide support for the WDC library and SDK though. If you find an issue with the WDC library, the simulator, or any of the developer samples, submit an issue on Github.

See also

- **Set Up Data Sources** on page 646 – Add more data to this data source or prepare your data before you analyze it.

- **Build Charts and Analyze Data** on page 843 – Begin your data analysis.

Other Databases (ODBC)

When you start Tableau, under **Connect**, you can see the file and database types that are supported by Tableau Desktop. Select **More** to see the complete list. For supported files and databases, Tableau provides built-in connectors that are built for and optimized for those data
sources. If your file or database type is listed here, use this named connector to connect to your data.

If your file or database type is not listed on under Connect, you might be able to connect to it using the Other Databases (ODBC) connector. ODBC (Open Database Connectivity) is a standard way to connect to a database. If the database driver you want to connect with implements the ODBC standard, you can connect Tableau to your data using the ODBC driver for your database and the Tableau Other Databases (ODBC) connector.

In this article

- **Make the connection** below
- Configure for publishing and cross-platform portability below
- Tableau support for Other Databases (ODBC) connections on page 603

**Make the connection**

To connect to your data using the ODBC driver, do the following:

1. Start Tableau and under Connect, select Other Databases (ODBC). For a complete list of data connections, select More under To a Server.

2. Enter the information required to make the connection.

   If table or schema names are not listed, click the search icon or enter the name in the text box and click the search icon, and then select the name.

**Important:** When you use Other Databases (ODBC) to connect to a database, the outcome may vary and compatibility with Tableau Desktop features is not guaranteed.

If your connection works as you need it to, you can begin your analysis. If, however, you need to customize your ODBC connection to improve functionality and performance, refer to the articles under See also on page 603.

**Configure for publishing and cross-platform portability**

When you work with a generic ODBC connection, additional configuration is required to make sure that the workbooks and data sources you create can be used on different computers and platforms.
For publishing and for cross-platform portability:

- We recommend that you use data source names (DSNs) to accommodate different drivers and configurations between systems.
- The driver and DSN must be installed and configured to match the workbook or data source connection.

As an example, Ashley Garcia creates an **Other Databases (ODBC)** data source that connects using a DSN for which she has the associated ODBC driver installed on her Windows computer. Here is an example of the data source configuration, including the data source name, that Ashley created:

![My ODBC Connector Data Source Configuration](image)

Ashley can do a couple of things with this data source to make it available for others to use.

- She can publish it to Tableau Server. When she does this, Ashley needs to ask the Tableau Server administrator to create the same DSN and install the associated ODBC driver on the Tableau Server computer so that the data source can be accessed by
Ashley and by others.

- She can email the data source to coworkers. Ashley needs to tell the coworkers to install the associated ODBC driver and create the same DSN on their computers so that they can open the data source in Tableau Desktop.

When you try to open a workbook or data source created using the Other Databases (ODBC) connector and your workstation isn't configured correctly (you don't have the workbook or data source's DSN configured, or you don't have the ODBC driver installed), you'll see an error message that begins Generic ODBC requires additional configuration. To resolve this error, contact the workbook or data source publisher or the Tableau Server administrator to get the DSN and ODBC driver information that matches the workbook or data source. Then install the driver and configure the DSN on your computer.

Create a DSN

The method you use to create a DSN depends on your operating system.

- **Windows.** Create a DSN using the ODBC Driver Manager.

- **OS X.** Create a DSN using an ODBC Manager utility. If you don't have the utility installed on your Mac, you can download one from, for example, http://www.odbcmanager.net/. Or you can manually edit the odbc.ini file.

Review the documentation for your operating system for specific instructions on how to create a DSN.

Tableau support for Other Databases (ODBC) connections

Tableau provides limited customer support for connections using Other Databases (ODBC). Tableau supports the Other Databases (ODBC) connector functionality, for example, Support will help you with the sample code provided in Tableau documentation, or if your connection works in Tableau Desktop, but doesn't work in Tableau Server. However, Tableau does not support creating or debugging custom TDC files. For help, you can review the documentation listed under See also below, and you can submit questions and ask for help on the Tableau Developer Community forums.

See also

**Tableau and ODBC on the next page** – Provides background information about ODBC, describes how Tableau determines the functionality of an ODBC driver, and lists frequently asked questions.
Customize and Tune ODBC Connections on page 609 – Describes how to fine tune the ODBC connection information for improved functionality and performance.

Tableau Capability Customizations Reference on page 619 – Lists customizations you can use to define which Tableau capabilities are supported by the data source.

ODBC/SQL Customizations Reference on page 635 – Lists customizations that represent the parts of the ODBC and SQL standards that the ODBC driver reports supporting.

Tableau and ODBC

Tableau includes built-in connectors for many databases. These connectors take advantage of capabilities and optimizations specific to these data sources, and they're developed and tested to ensure they are robust and provide good performance.

But you might be working with a database that Tableau doesn't have a connection for. In that case, you still might be able to connect Tableau to the database. Tableau has a connector that uses the ODBC standard. By using ODBC, you can access any data source that supports the SQL standard and implements the ODBC API.

In this article

ODBC basics below
How Tableau determines the capabilities of the ODBC driver on the next page
Tune ODBC connector performance on page 607
Tableau support for ODBC connections on page 607
ODBC FAQ on page 607

ODBC basics

ODBC (Open Database Connectivity) is an industry standard that enables a wide variety of software to access data. The basis of ODBC is a standard syntax for SQL queries that is used by software applications to connect to and request data from databases. The ODBC driver, usually provided by the database vendor, accepts requests in this standard syntax and converts the request to the native format that the target database prefers. In effect, the ODBC driver is a translation layer to convert from a general-purpose request to the database-specific request.
For example, the ODBC specification details that dates should be specified in SQL statements by using the syntax: `{d 'yyyy-mm-dd'}. This format is then translated by the driver to the proper date syntax of the actual database. If you use the ODBC connector to connect to an Oracle database, the ODBC connector sends a request to the Oracle ODBC driver in this format:

```
select name from emp where birthdate > {d '1987-12-29'}
```

The driver converts your request to the actual format the Oracle database requires, which is this:

```
select name from emp where birthdate > '29-DEC-87'
```

The ODBC specification includes syntax for making function requests, join syntax, data types, and data type conversions. The SQL language itself supports complex concepts such as nested queries, correlated subqueries, temporary tables, and a variety of functions that can be used in select, where, group by, join clauses, and more. The ODBC driver is responsible for the proper conversion of all these requests to the syntax of the target database.

How Tableau determines the capabilities of the ODBC driver

There are differences in how each database vendor implements capabilities of the ODBC standard. Tableau uses a feature-discovery API in ODBC to question a database driver about the functionality it supports. One example of how Tableau changes behavior based on what the driver reports is the list of functions available when creating a calculated field. Fewer driver limitations means that more functions are available.

In some cases Tableau will require you to create a data extract from the data returned by the ODBC connector. And there are some ODBC drivers and databases that Tableau can't connect to.

ODBC discovery

During the initial connection, Tableau performs requests to the driver and SQL discovery queries in order to determine the capabilities of the driver.

Some examples of the capabilities discovery that Tableau performs include:

- What scalar and aggregate functions are available.
- What date and time manipulation functions are available.
- Can temporary tables be created and can if select into statements can be used.
- Are subqueries supported.
- Are top and limit queries supported.
- What join styles are supported (outer, inner, full).
- What data types are supported.

Based on the outcome of the capabilities discovery, Tableau will classify the current connection as being in one of four categories:

- **Fully Functional.** This is a driver that supports all the functions and capabilities that Tableau uses.
- **Minor limitations.** This is a driver that has a small number of non-critical limitations. An example would be a driver that doesn't support the full set of numeric, string, or date functions that Tableau normally makes available. Tableau will modify its behavior to account for these limitations and proceed.
- **Major limitations.** This is a driver that has a large number of limitations or critical capabilities that Tableau relies upon but that are not supported. However, even with these major restrictions, Tableau might be able to connect well enough to create a Tableau extract to retrieve the data into Tableau for further work offline from the database. In this case, Tableau will recommend that you create an extract. When you create the extract, you might not be able to create a filter on the data source to reduce the amount of data in the extract. (For more information, see Filter Data from Data Sources on page 766.) After the data has been extracted, full Tableau functionality is available when working with the extract.
- **Fatal limitations.** This is a driver that does not support even the most minimal set of capabilities that Tableau needs in order to connect and run the basic queries to create extract files. Therefore, Tableau is unable to proceed any further with this driver.

After the connection is established, if Tableau has determined that the capabilities available classify this connection as anything less than fully functional, a message is displayed to inform you of the limitations that have been detected. For example, a connection to an ODBC driver with minor limitations displays this message:
For more critical limitations, the warning dialog box will advise you to create a Tableau extract in order to proceed.

Tune ODBC connector performance

Tableau supports the ability to customize your ODBC data connection, which can improve the connection experience. For more information, see Customize and Tune ODBC Connections on page 609.

Tableau support for ODBC connections

Tableau provides no guarantee or warranty that using Tableau with any particular ODBC driver or database will be able to successfully connect and query data. Some ODBC drivers will support the full Tableau interactive experience, while others may work only for creating extracts. Some ODBC drivers might not work with Tableau.

Note: Tableau will provide reasonable levels of customer support to assist in troubleshooting connections with ODBC drivers, but can’t create or customize a connector to work with a specific ODBC driver.

ODBC FAQ

What's a typical use case for the ODBC connector?

The most common use case for an ODBC connection is to access data specifically to retrieve it into a Tableau extract. Bringing this data into a Tableau extract then allows the full Tableau capabilities to work with the data. Many ODBC drivers support the needed functionality to connect and perform the simpler queries required to perform an extract. Whether you extract all of the data or just select a few columns and set some filters to retrieve a relevant subset, this use of ODBC is a good scenario to keep in mind.
Where do I get ODBC drivers for my database?

Most database vendors distribute ODBC drivers for use with their databases. Contact your database vendor to obtain the drivers. Additionally there are a number of third party manufacturers of ODBC drivers that can provide drivers for a variety of common databases.

What ODBC version driver do I need?

Be sure your ODBC driver is a version 3 or higher driver, meaning it implements the ODBC specification version 3. Each driver provider will have their own version numbering system for their driver versions, which will likely be different from the ODBC version number it implements. The ODBC version 3 specification was introduced in 1995, so it is likely your database provider has a driver that implements this level of compliance. We have seen a number of Tableau customers with drivers that are not version 3 compliant (which fail to connect), who then upgraded to a newer driver and were able to proceed.

Has Tableau tested database [x]?

Tableau has tested ODBC connections with many data sources, but due to the volume of ODBC drivers available on the market we haven’t tested against every possible driver, nor can Tableau perform this breadth of testing. The best recommendation for any particular driver is to give it a try, and let us know how it works.

What do I do if it doesn't work?

First check the driver version. If you look at the Tableau logs we will echo back the ODBC version level the driver reports. Search the logs for an entry like the following:

```
ODBCProtocol: driver ODBC version: 03.52
```

The number at the end indicates the ODBC version level. If it is less than 03.00, the driver needs to be upgraded.

If you get warnings when you connect about capabilities not supported, check with your database vendor to see if they have an updated driver that supports more capabilities. Not all drivers will work with Tableau.

Should I use the named database connector or the ODBC connector?

If you are connecting to a database for which Tableau has a named connection option, then use the named connector. The named connectors are optimized for the particular database.
See also

**Other Databases (ODBC) on page 600** – Describes how to connect to your data using the ODBC connector.

**Customize and Tune ODBC Connections** below – Describes how to fine tune the ODBC connection information for improved functionality and performance.

**Tableau Capability Customizations Reference on page 619** – Lists customizations you can use to define which Tableau capabilities are supported by the data source.

**ODBC/SQL Customizations Reference on page 635** – Lists customizations that represent the parts of the ODBC and SQL standards that the ODBC driver reports supporting.

**Customize and Tune ODBC Connections**

When you use a connector for a database that supports SQL, Tableau generates SQL statements that are tuned for that database. Because Tableau has no representation of the SQL dialect used by an ODBC data source, it must infer the supported syntax through a variety of tests. If a driver reports incorrect or incomplete information about the SQL dialect it supports, you can use Tableau customizations to fine tune this connection information in order to improve functionality and performance.

In this article, you will make an ODBC connection, examine the resulting .tds file and use part of it to create a Tableau Datasource Customization (.tdc) file that you can then use to customize your ODBC connection. Before you begin, you should be familiar with the content in **Tableau and ODBC on page 604**.

**Note:** Tableau will provide reasonable levels of customer support to assist in troubleshooting connections with ODBC drivers, but can't create or customize a connector to work with a specific ODBC driver.

In this article

- Make an ODBC connection on the next page
- Review the XML structure of a TDS file on page 611
- Make customizations global with a .tdc file on page 612
- Customize your ODBC connection on page 616
- ODBC documentation reference on page 618
Make an ODBC connection

This section shows you how to create an ODBC connection using an example. In the example, you connect to a SQLite database using the SQLite ODBC driver.

Prerequisites

The ODBC connection in this article is based on SQLite (http://www.sqlite.org/), an open-source database.

You need to download the following two items:

Disclaimer: This information refers to a third-party product. This example is not an endorsement of this product over any other competing products.

- The SQLite 32-bit ODBC driver (required for 32-bit or 64-bit Windows). Download and install the following (select the SQLite 2 drivers):
  http://www.ch-werner.de/sqliteodbc/sqliteodbc.exe
- A sample SQLite database created from the Microsoft example Northwind database. Download and expand the following:
  http://download.vive.net/Northwind.zip

Create a connection

To create an ODBC connection, you connect to the Northwind database using the SQLite3 ODBC driver, and then save the connection as a Tableau Data Source (.tds) file.

1. Open Tableau Desktop.
2. On the start page under Connect, click Other Databases (ODBC).
3. Under Connect Using, select Driver, and then from the drop-down list, select SQLite3 ODBC Driver.
4. Click Connect.
5. Next to the Database Name text box, click Browse, navigate to the location of your Northwind.sl3 file, and then click Open.
6. Click OK to close the dialog box, and then click Sign In.
7. On the data source page, in the **Table** text box, enter **Orders**.

8. Drag the **Orders** table to the canvas, and then click the sheet tab.

   A "Tableau identified limitations for the ODBC data source" dialog box opens. You can review the details or dismiss the dialog box and continue. For more information, see [How Tableau determines the capabilities of the ODBC driver](#).

9. Select **Data > Orders [path to Northwind.sl3] > Add to Saved Data Sources**.

10. Click **Save**.

11. Close the workbook. When asked to save changes, click **No**.

Now you can open the .tds file to examine the SQLite connection.

**Review the XML structure of a TDS file**

Open the saved data source (.tds file) in a text editor to view the XML structure of the .tds file. By default, the file you created above is named **Northwind.sl3.tds** and is in the following path:

`Users\[your name]\Documents\My Tableau Repository\Datasources`

or

`Users\[your name]\Documents\My Tableau Repository (Beta)\Datasources`

This is an XML document describing the SQLite connection to the Northwind Orders table. Within the `<connection>` section, find the `<connection-customization>` element, which contains `<customization>` elements that you can edit.

**Vendor and driver name**

The connection customization section begins with the names Tableau detected for the ODBC driver and the database vendor. Tableau uses this information to associate a specific connection customization with a single type of ODBC data source. The section looks like this:

```
<connection-customization class='genericodbc' enabled='false' version='10.1'>

/vendor name='SQLite' />
```
<driver name='SQLite3 ODBC Driver' />

<customizations>

Types of customizations

Tableau permits two types of customizations: Tableau-specific capabilities, and ODBC API calls to SQLGetInfo. These customizations are made of name/value pairs, with the names following a convention of CAP_ for Tableau capabilities and SQL_ for the SQLGetInfo API calls.

The saved data source file contains examples of both types of customizations. These customizations represent the values that Tableau was able to detect by querying the driver when you connected. The list of customizations might be incomplete or incorrect. You can use customization to modify and shape Tableau's behavior when connecting to an ODBC data source.

The following articles contain a complete reference for the customizations. You should have an idea of how you need to adjust your ODBC connection to perform as you expect in your environment. Review the lists to get an idea of the customizations you can use to tune your ODBC connection.

- Tableau Capability Customizations Reference on page 619
- ODBC/SQL Customizations Reference on page 635

Format of customization values

- The customization values are all represented as strings for each name/value pair.
- Tableau capabilities are all Boolean values represented by yes or no.
- The SQLGetInfo values can be long-integer, short-integer, or string data, depending on the expected return value of SQLGetInfo for the given identifier.
- Many of the integer fields are bitmasks that represent a collection of capabilities.

Make customizations global with a .tdc file

To make your customization changes apply to all connections for a given ODBC data source, you need to create a Tableau Datasource Customization (.tdc) file. This file contains only the <connection-customization> section and is applied to any new Tableau connection that matches the database vendor name and driver name described in the .tdc file (as noted in Vendor and driver name on the previous page). Any existing workbook or data source file
that already has an enabled customization section will use only the customizations that it supplies, not the .tdc file.

**Important:** Tableau does not test or support TDC files. These files should be used as a tool to explore or occasionally address issues with your data connection. Creating and maintaining TDC files requires careful manual editing, and there is no support for sharing these files.

Create a .tdc file

When you create a .tdc file, you need to save it to the right location so that it can be used by your connection. If you create a workbook that you want to publish to Tableau Server, then you need to save the .tdc file to the server as well. For more information, see Using a .tdc File with Tableau Server in Tableau Knowledge Base.

**Tableau Desktop**

1. **Using a text editor,** copy and paste the entire `<connection-customization>` section of your saved data source file. For reference, see the Sample SQLite TDC file.

2. **Name the file** `odbc-sqlite.tdc` and **save it to** Documents\My Tableau Repository\Datasources.

   **Note:** The file must be saved using a .tdc extension, but the name does not matter.

3. **Restart Tableau Desktop** to apply the change.

4. **Create a new connection** to SQLite as described in Create a connection.

5. **Go to and open the Tableau Desktop log file,** and look for a record similar to the example below to verify that this customization file was applied to your new connection.

<table>
<thead>
<tr>
<th>Log File Location</th>
<th>Example of Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>By default, C:\User-s&lt;user&gt;\Documents\My Tableau Repository\Logs</td>
<td>Found matching TDC 'C:\User-s&lt;name&gt;\Documents\My Tableau Repository\Datasources\odbc-sqlite.tdc' for class='genericodbc',</td>
</tr>
<tr>
<td>Log File Location</td>
<td>Example of Record</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td></td>
<td>vendor='SQLite', and driver='SQLite3 ODBC Driver'</td>
</tr>
</tbody>
</table>

**Tableau Server**

1. Using a text editor, copy and paste the entire `<connection-customization>` section of your saved data source file. For reference, see the Sample SQLite TDC file.

2. Name the file `odbc-sqlite.tdc` and save it to the following location:
   - For Windows:
     In Tableau Server versions prior to 2018.2:
     Program Files\Tableau\Tableau Server\<version>\bin
     Starting in Tableau Server version 2018.2:
     Program Files\Tableau\Tableau Server\packages\bin.<build number>
   - Or for all Tableau Server versions:
     ProgramData\Tableau\Tableau Server\data\tabsvc\vizqlserver\Datasources
   To save the file, you must be a Windows Administrator on the server computer.
   - For Linux:
     /var/opt/tableau/tableau_
     server/data/tabsvc/vizqlserver/Datasources/

   **Note:** The file must be saved using a `.tdc` extension, but the name does not matter. The `.tdc` file must be saved on all server nodes for consistent behavior.

3. Restart Tableau Server to apply the change.

4. In Tableau Desktop, create a new connection to SQLite as described in Create a connection.

5. Go to and open the Tableau Server log file, and look for a record similar to the example below to verify that this customization file was applied to your new connection.
### Log File Location vs. Example of Record

<table>
<thead>
<tr>
<th>Log File Location</th>
<th>Example of Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\ProgramData\Tableau\TableauServer\data\tabsvc\vizqlserver\Logs</td>
<td>Found matching TDC 'C:\ProgramData\Tableau\Tableau Server\data\tabsvc\vizqlserver\Datasources\odbc-sqlite.tdc for class='genericodbc', vendor='SQLite', and driver='SQLite3 ODBC Driver'</td>
</tr>
</tbody>
</table>

## Sample SQLite .tdc file

```xml
<?xml version='1.0' encoding='utf-8' ?>

<connection-customization class='genericodbc' enabled='true' version='7.8'>

  <vendor name='SQLite' />

  <driver name='SQLite3 ODBC Driver' />

  <customizations>
    
    <customization name='CAP_CREATE_TEMP_TABLES' value='yes' />
    
    <customization name='CAP_QUERY_BOOLEXPR_TO_INTEXPR' value='yes' />
    
    <customization name='CAP_QUERY_GROUP_BY_ALIAS' value='no' />
    
    <customization name='CAP_QUERY_GROUP_BY_DEGREE' value='yes' />
    
    <customization name='CAP_QUERY_JOIN_ACROSS_SCHEMAS' value='no' />
    
    <customization name='CAP_QUERY_JOIN_REQUIRES_SCOPE' value='no' />
    
    <customization name='CAP_QUERY_SUBQUERIES' value='yes' />

</customizations>
```

-615-
Customize your ODBC connection

Let's assume that you've made an ODBC connection to your data source and verified that you can get the metadata and data you need. Now you need to determine if creating an extract or if using the live connection as is gives you the functionality you need. If it does, the .tdc file you created is sufficient, and there is no need to customize your connection.

If the .tdc file is not performing the way you want it to, you can customize the connection. You should have an idea of how you need to adjust your ODBC connection to perform as you expect in your environment. Review the lists to see the customizations you can use to tune your ODBC connection.

- **Tableau Capability Customizations Reference** on page 619
- **ODBC/SQL Customizations Reference** on page 635

Common customizations for improving functionality

The following customizations can help improve functionality for data sources with default capabilities that are not optimal.

- **CAP_SUPPRESS_DISCOVERY_QUERIES** – Setting this value to true prevents Tableau from performing any SQL queries at the beginning of the connection to determine the driver's capabilities. You will need to consider which other capabilities should be explicitly enabled, since Tableau will not be able to determine those automatically.

- **SQL_SQL_CONFORMANCE** – This setting indicates which level of the SQL standard is
fully supported by the data source. Tableau works best with at least entry-level conformance, so this value should be at least 1 if the data source supports it.

- **SQL_AGREGATE_FUNCTIONS** – This setting indicates which aggregate functions are supported, such as MIN, MAX, SUM. A value of '127' indicates support for all standard SQL aggregate functions.

### Handle severely limited data sources

Some data sources are so severely limited that Tableau is unable to complete the steps of creating a connection. Occasionally this is due to crashes within the driver, which cause Tableau to cease working. You can use a global TDC file to prevent Tableau from issuing queries or checking for capabilities that might be associated with the instability.

To create this file, you need to know the database vendor name and ODBC driver name.

1. Create a new connection in Tableau, open your log file, and then look for a line like the following:

   `GenericODBCProtocol::Connect: Detected vendor: 'SQLite' and driver: 'SQLite3 ODBC Driver'

2. Create a .tdc file with the listed vendor name and driver name. See Vendor and driver name on page 611 for an example.

3. Use customizations, such as `CAP_SUPPRESS_DISCOVERY_QUERIES`, to constrain Tableau’s interaction with a limited data source.

### Configure ODBC driver settings

ODBC drivers provide dialog boxes for supplying connection details such as the server, user name, and password. Many offer advanced options for controlling the connection behavior. You should consider these options when exploring ways to improve the functionality or performance of your ODBC connection. In particular, look for settings that control the items listed below, as these have been the cause of past issues with ODBC connections in Tableau:

- **Transaction Isolation** - Prefer READ, COMMITTED, or SERIALIZABLE to ensure that queries do not include data from pending transactions in their results.

- **Cursors** - Prefer Holdable or Scrollable over Streaming cursors. Since Tableau closes a transaction after every read-only analytical query, streaming cursors may become
truncated and lead to incomplete results displayed in Tableau.

- Row/Resultset Buffers - Prefer larger buffer sizes to improve the performance of fetching numerous rows, which can greatly improve the speed of creating extracts. This is sometimes called the cache size or response size.
- Character Set - In decreasing order of preference: UTF-16, UTF-8, ASCII.

Advanced SQLite customizations

Tableau has special customizations for SQLite that are built in to the product. These customizations take precedence over any connection customizations for SQLite in the workbook file, data source file, or global .tds file. In order to make advanced changes to SQLite connection customizations, you need to trick Tableau into ignoring its own SQLite dialect. You can do so by changing the reported name of the database vendor (<customization name='SQL_DBMS_NAME' value='SQLite' />) to a different value such as SQLite-Tableau.

ODBC documentation reference

Microsoft's MSDN has documentation for the ODBC standard. The ODBC Appendixes are the most applicable as resources for the article information, specifically:

- Appendix C: SQL Grammar - especially the SQL Minimum Grammar and ODBC Escape Sequences
- Appendix D: Data Types
- Appendix E: Scalar Functions

See also

Other Databases (ODBC) on page 600 – Describes how to connect to your data using the ODBC connector.

Tableau and ODBC on page 604 – Provides background information about ODBC, describes how Tableau determines the functionality of an ODBC driver, and lists frequently asked questions.

Tableau Capability Customizations Reference on the next page – Lists customizations you can use to define which Tableau capabilities are supported by the data source.
ODBC/SQL Customizations Reference on page 635 – Lists customizations that represent the parts of the ODBC and SQL standards that the ODBC driver reports supporting.

Tableau Capability Customizations Reference

You can set the following capability customizations in the Tableau Datasource Customization (TDC) file to define which Tableau capabilities are supported by the ODBC connection. For more information, see Customize and Tune ODBC Connections on page 609.

Many of these customizations influence the type of SQL queries that Tableau issues. For settings that are not defined, Tableau attempts to determine the proper values for each capability by issuing various forms of SQL queries to experimentally verify which forms are supported, as described in How Tableau determines the capabilities of the ODBC driver.

<table>
<thead>
<tr>
<th>Capability Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAP_CREATE_TEMP_TABLES</td>
<td>Set to 'yes' if Tableau can create temporary tables needed for certain complex or optimized queries. See also: CAP_SELECT INTO.</td>
</tr>
<tr>
<td>CAP_CONNECT_STORED_PROCEDURE</td>
<td>Set to 'yes' to allow support for connecting to a stored procedure.</td>
</tr>
<tr>
<td>CAP_FAST_METADATA</td>
<td>Set to 'yes' if you have small to moderate size schemas. This capability controls whether Tableau should enumerate all of the objects immediately when you connect. Set the value to &quot;yes&quot; to enable this capability for better performance when creating new connections. Disable this capability to allow search for specific schemas or tables instead of retrieving all objects. You can search for all objects by using an empty string. This capability is available in 9.0 and later.</td>
</tr>
<tr>
<td>CAP_ISOLATION_LEVEL_READ_COMMITTED</td>
<td>Set to 'yes' to force the transaction isolation level to Read Committed if the data source supports it. Only one of the four transaction isolation levels should be set to 'yes'. See also: CAP_SET_ISOLATION_LEVEL_VIA_SQL, CAP_SET_ISOLATION_LEVEL_VIA_ODBC_API.</td>
</tr>
<tr>
<td>CAP_ISOLATION_LEVEL_READ</td>
<td>Set to 'yes' to force the transaction isolation level</td>
</tr>
<tr>
<td>UNCOMMITTED</td>
<td>to Read Uncommitted if the data source supports it. Only one of the four transaction isolation levels should be set to 'yes'. This capability can improve speed by reducing lock contention, but may result in partial or inconsistent data in query results. See also: CAP_SET_ISOLATION_LEVEL_VIA_SQL, CAP_SET_ISOLATION_LEVEL_VIA_ODBC_API.</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>--</td>
</tr>
<tr>
<td>CAP_ISOLATION_LEVEL_REPEATABLE_READS</td>
<td>Set to 'yes' to force the transaction isolation level to Repeatable Reads if the data source supports it. Only one of the four transaction isolation levels should be set to 'yes'. See also: CAP_SET_ISOLATION_LEVEL_VIA_SQL, CAP_SET_ISOLATION_LEVEL_VIA_ODBC_API.</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>--</td>
</tr>
<tr>
<td>CAP_ISOLATION_LEVEL_SERIALIZABLE</td>
<td>Set to 'yes' to force the transaction isolation level to Serializable if the data source supports it. Only one of the four transaction isolation levels should be set to 'yes'. This is a very conservative setting that may improve stability at the expense of performance. See also: CAP_SET_ISOLATION_LEVEL_VIA_SQL, CAP_SET_ISOLATION_LEVEL_VIA_ODBC_API.</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>--</td>
</tr>
<tr>
<td>CAP_SET_ISOLATION_LEVEL_VIA_ODBC_API</td>
<td>Set to 'yes' to force Tableau to set the transaction isolation level for the data source using the ODBC API. CAP_SET_ISOLATION_LEVEL_VIA_ODBC_API must be set to 'yes' when any one of the four CAP_ISOLATION_LEVEL capabilities has been set to 'yes'.</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>--</td>
</tr>
<tr>
<td>CAP_SET_ISOLATION_LEVEL_VIA_SQL</td>
<td>Set to 'yes' to force Tableau to set the transaction isolation level for the data source using a SQL query. CAP_SET_ISOLATION_LEVEL_VIA_SQL must be set to 'yes' when any one of the four CAP_ISOLATION_LEVEL capabilities has been set to 'yes'.</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>--</td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CAP_MULTIPLE_CONNECTIONS_FROMSAME_IP</td>
<td>Set to 'no' to prevent Tableau from creating more than one active connection to the database. This is a conservative setting that may increase stability at the expense of performance.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_DETECT_ALIAS_CASE_FOLDING</td>
<td>Set to 'yes' to allow Tableau to detect and recover from an ODBC data source that reports the field names in a result set using only upper-case or lower-case characters, instead of the expected field names.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_BOOL_AS_WCHAR_01LITERAL</td>
<td>Set to 'yes' to bind a Boolean data type as a WCHAR containing values '0' or '1'.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_BOOL_AS_WCHAR_TFLITERAL</td>
<td>Set to 'yes' to bind a Boolean data type as WCHAR containing values 't' or 'f'.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_FORCE_DATE_AS_CHAR</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to bind date values as CHAR.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_FORCE_DATETIME_AS_CHAR</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to bind datetime values as CHAR.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_FORCE_MAX_STRING_BUFFERS</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to use maximum-sized buffers (1MB) for strings instead of the size described by metadata.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_FORCE_MEDIUM_STRING_BUFFERS</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to use medium-sized buffers (1K) for strings instead of the size described by metadata.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_FORCE_SMALL_STRING_BUFFERS</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to use small buffers for strings instead of the size described by metadata.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_FORCE_SIGNED</td>
<td>Set to 'yes' to force binding integers as signed.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_PRESERVE_BOM</td>
<td>Set to 'yes' to preserve BOM when present in strings. Hive will return BOM and treat strings containing it as distinct entities.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_SUPPRESS_COERCE_TO_STRING</td>
<td>Set to 'yes' to prevent the Tableau native ODBC protocol from binding non-string data as strings (i.e. requesting driver conversion).</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_SUPPRESS_INT64</td>
<td>Set to 'yes' to prevent the Tableau native ODBC protocol from using 64-bit integers for large numeric data.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_SUPPRESS_PREFERRED_CHAR</td>
<td>Set to 'yes' to prevent the Tableau native ODBC protocol from preferring a character type that differs from the driver default.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_SUPPRESS_PREFERRED_TYPES</td>
<td>Set to 'yes' to prevent the Tableau native ODBC protocol from binding any data according to its preferred wire types. With this capability set, Tableau will only bind according to the data types described by the ODBC driver via metadata.</td>
</tr>
<tr>
<td>CAP_ODBC_BIND_SUPPRESS_WIDE_CHAR</td>
<td>Set to 'yes' to prevent the Tableau native ODBC protocol from binding strings a WCHAR. Instead they will be bound as single-byte CHAR arrays, and processed locally for any UTF-8 characters contained within.</td>
</tr>
<tr>
<td>CAP_ODBC_CONNECTION_STATE_VERIFY_FAST</td>
<td>Set to 'yes' to check if a connection is broken with a fast ODBC API call.</td>
</tr>
<tr>
<td>CAP_ODBC_CONNECTION_STATE_VERIFY_PROBE</td>
<td>Set to 'yes' to check if a connection is broken with a forced probe.</td>
</tr>
<tr>
<td>CAP_ODBC_CONNECTION_STATE_VERIFY_PROBE_IF_STALE</td>
<td>Set to 'yes' to check if a connection is broken with a forced probe only if it is &quot;stale&quot; (i.e., unused for about 30 minutes).</td>
</tr>
<tr>
<td>CAP_ODBC_CONNECTION_STATE_VERIFY_PROBE_PREPARED_QUERY</td>
<td>Set to 'yes' to check if a connection is broken using a prepared query.</td>
</tr>
<tr>
<td>CAP_ODBC_CURSOR_DYNAMIC</td>
<td>Set to 'yes' to force the Tableau native ODBC</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CAP_ODBC_CURSOR_FORWARD_ONLY</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to set the cursor type for all statements to Forward-only (non-scrollable).</td>
</tr>
<tr>
<td>CAP_ODBC_CURSOR_KEYSET_DRIVEN</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to set the cursor type for all statements to Keyset-driven (scrollable, detects changes to values within a row).</td>
</tr>
<tr>
<td>CAP_ODBC_CURSOR_STATIC</td>
<td>Set to 'yes' to force Tableau to set the cursor type for all statements to Static (scrollable, does not detect changes).</td>
</tr>
<tr>
<td>CAP_ODBC_ERROR_IGNORE_FALSE_ALARM</td>
<td>Set to 'yes' to allow the Tableau native ODBC protocol to ignore SQL_ERROR conditions where SQLSTATE is '00000' (meaning &quot;no error&quot;).</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_ALLOW_CHAR_UTF8</td>
<td>Set to 'yes' to allow the use of single-byte char data type for binding Unicode strings as UTF-8.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_BIND_FORCE_TARGET_METADATA</td>
<td>Set to 'yes' to force binding for export based on all of the metadata from the target table instead of the ODBC metadata for the parameterized insert statement.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_BIND_PREFER_TARGET_METADATA</td>
<td>Set to 'yes' to prefer binding for export based on specific types of metadata from the target table instead of the ODBC metadata for the parameterized insert statement.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_BUFFERS_RESIZABLE</td>
<td>Set to 'yes' to allow export buffers to be reallocated after the first batch to improve performance.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_BUFFERS_</td>
<td>Set to 'yes' to ignore the width of a single row</td>
</tr>
<tr>
<td>Setting</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SIZE_FIXED</td>
<td>when computing the total rows to insert at a time.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_BUFFERS_SIZE_LIMIT_512KB</td>
<td>Set to 'yes' to limit export buffers to 512 KB. This is an uncommon setting.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_BUFFERS_SIZE_MASSIVE</td>
<td>Set to 'yes' to force the use of large buffers for insert. If CAP_ODBC_EXPORT_BUFFERS_RESIZABLE is not set or disabled, a fixed row count is used.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_BUFFERS_SIZE_MEDIUM</td>
<td>Set to 'yes' to force the use of medium-sized buffers for insert. If CAP_ODBC_EXPORT_BUFFERS_RESIZABLE is not set or disabled, a fixed row count is used.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_BUFFERS_SIZE_SMALL</td>
<td>Set to 'yes' to force the use of small buffers for insert. If CAP_ODBC_EXPORT_BUFFERS_RESIZABLE is not set or disabled, a fixed row count is used.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_CONTINUE_ON_ERROR</td>
<td>Set to 'yes' to continue data insert despite errors. Some data sources report warnings as errors.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_DATA_BULK</td>
<td>Set to 'yes' to allow the use of ODBC bulk operations for data insert.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_DATA_BULK_VIA_INSERT</td>
<td>Set to 'yes' to allow the use of ODBC bulk operations based on 'INSERT INTO' parameterized queries.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_DATA_BULK_VIA_ROWSET</td>
<td>Set to 'yes' to allow the use of ODBC bulk operations based on a rowset cursor.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_FORCE_INDICATE_NTS</td>
<td>Set to 'yes' to force the use of indicator buffers for identifying null-terminated strings (NTS).</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_FORCE_SINGLE_ROW_BINDING</td>
<td>Set to 'yes' to force the use of a single row for binding export buffers to insert data.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_FORCE</td>
<td>Set to 'yes' to force the use of a single row for</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SINGLE_ROW_BINDING_WITH_TIMESTAMP</td>
<td>Binding export buffers when dealing with timestamp data. This is required for some versions of Teradata.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_FORCE_STRING_WIDTH_FROM_SOURCE</td>
<td>Set to 'yes' to force the use of the source string width (from Tableau metadata), overriding the destination string width (from insert parameter metadata).</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_FORCE_STRING_WIDTH_USING_OCTET_LENGTH</td>
<td>Set to 'yes' to force the use of the source string width from the octet length.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_SUPPRESS_STRING_WIDTH_VALIDATION</td>
<td>Set to 'yes' to suppress validating that the target string width can accommodate the widest source strings.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_BATCH_MASSIVE</td>
<td>Set to 'yes' to commit in massive batches of INSERT statements (~100,000). This may be useful with single-row export binding.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_BATCH_MEDIUM</td>
<td>Set to 'yes' to commit in medium-sized batches of INSERT statements (~50). A single statement may be bound to multiple records.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_BATCH_SMALL</td>
<td>Set to 'yes' to commit in small batches of INSERT statements (~5). A single statement may be bound to multiple records.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_BYTES_MASSIVE</td>
<td>Set to 'yes' to commit in massive batches of data (~100 MB).</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_BYTES_MEDIUM</td>
<td>Set to 'yes' to commit in medium batches of data (~10 MB).</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_BYTES_SMALL</td>
<td>Set to 'yes' to commit in small batches of data (~1 MB).</td>
</tr>
<tr>
<td>SMALL</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_EACH_STATEMENT</td>
<td>Set to 'yes' to commit after executing each INSERT statement. A single statement may be bound to multiple records.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_INTERVAL_LONG</td>
<td>Set to 'yes' to commit in long intervals of elapsed time (~100 seconds).</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_INTERVAL_MEDIUM</td>
<td>Set to 'yes' to commit in medium intervals of elapsed time (~10 seconds).</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_INTERVAL_SHORT</td>
<td>Set to 'yes' to commit in short intervals of elapsed time (~1 seconds).</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_ONCE_WHEN_COMPLETE</td>
<td>Set to 'yes' to commit only once at the end after the export is complete.</td>
</tr>
<tr>
<td>CAP_ODBC_EXPORT_TRANSLATE_DATA_PARALLEL</td>
<td>Set to 'yes' to use parallel loops to translate Tableau DataValues to wire buffers on export.</td>
</tr>
<tr>
<td>CAP_ODBC_FETCH_ABORT_FORCE_CANCEL_STATEMENT</td>
<td>Set to 'yes' to cancel the statement handle upon interrupting SQLFetch with a cancel exception.</td>
</tr>
<tr>
<td>CAP_ODBC_FETCH_BUFFERS_RESIZABLE</td>
<td>Set to 'yes' to allow buffers to be reallocated after fetch to improve performance or handle data truncation.</td>
</tr>
<tr>
<td>CAP_ODBC_FETCH_BUFFERS_SIZE_FIXED</td>
<td>Set to 'yes' to ignore the width of a single row when computing the total rows to fetch.</td>
</tr>
<tr>
<td>CAP_ODBC_FETCH_BUFFERS_SIZE_MASSIVE</td>
<td>Set to 'yes' to force the use of large buffers. If CAP_ODBC_FETCH_BUFFERS_SIZE_FIXED is enabled, a fixed row count is used.</td>
</tr>
<tr>
<td>CAP_ODBC_FETCH_BUFFERS</td>
<td>Set to 'yes' to force the use of medium-sized buffers.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SIZE_MEDIUM</td>
<td>Set to 'yes' to force the use of small buffers. If CAP_ODBC_FETCH_BUFFERS_SIZE_FIXED is enabled, a fixed row count is used.</td>
</tr>
<tr>
<td>CAP_ODBC_FETCH_BUFFERS_SIZE_SMALL</td>
<td>Set to 'yes' to force the use of small buffers. If CAP_ODBC_FETCH_BUFFERS_SIZE_FIXED is enabled, a fixed row count is used.</td>
</tr>
<tr>
<td>CAP_ODBC_FETCH_CONTINUE_ON_ERROR</td>
<td>Set to 'yes' to allow the Tableau native ODBC protocol to continue resultset fetch despite errors (some data sources report warnings as errors).</td>
</tr>
<tr>
<td>CAP_ODBC_FETCH_IGNORE_FRACTIONAL_SECONDS</td>
<td>Set to 'yes' to allow the Tableau native ODBC protocol to ignore the fractional seconds component of a time value when fetching query result set data. This is useful when working with data sources that do not follow the ODBC specification for fractional seconds, which must be represented as billionths of a second.</td>
</tr>
<tr>
<td>CAP_ODBC_FETCH_RESIZE_BUFFERS</td>
<td>Set to 'yes' to allow the Tableau native ODBC protocol to automatically resize buffers and fetch again if data truncation occurred.</td>
</tr>
<tr>
<td>CAP_ODBC_FORCE_SINGLE_ROW_BINDING</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to use a single row for result set transfers instead of the more efficient bulk-fetch.</td>
</tr>
<tr>
<td>CAP_ODBC_IMPORT_ERASE_BUFFERS</td>
<td>Set to 'yes' to reset the contents of data buffers before fetching each block.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_FORCE_LENGTH_AS_PRECISION</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to use the column &quot;length&quot; as the numeric precision. This is an uncommon setting.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_FORCE_NUM_PREC_RADIX_10</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to assume the numeric precision is reported in base-10 digits. This is an uncommon setting.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_FORCE_UNKNOWN_AS_STRING</td>
<td>Set to 'yes' to force the Native ODBC Protocol to treat unknown data types as string instead of ignoring the associated column.</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_FORCE_UTF8_IDENTIFIERS</td>
<td>Set to 'yes' to force the protocol to treat identifiers as UTF-8 when communicating with the driver.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_SKIP_DESC_TYPE_NAME</td>
<td>Set to 'yes' to remove the check for the SQL_DESC_TYPE_NAME attribute with the SQLOctetLength API.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_STRING_LENGTH_UNKNOWN</td>
<td>Set to 'yes' to prevent Tableau from allocating memory based on the driver-reported string length, which may not be known or reported properly. Instead, Tableau will use a fixed-sized string length, and will reallocate as needed to handle string data that is too large for the fixed-size buffer.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_STRING_TRUST_OCTET_LENGTH</td>
<td>Set to 'yes' to use the octet length reported by the driver for strings instead of computing it from the number of characters.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_SUPPRESS_EXECUTED_QUERY</td>
<td>Set to 'yes' to prevent Tableau from executing a query as a means of reading metadata. While Tableau typically includes a row-limiting clause in such metadata queries (e.g., 'LIMIT', or 'WHERE 1=0'), this may not help when used with a Custom SQL connection for database systems with poor query optimizers. Note that this capability may prevent Tableau from determining the connection metadata properly.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_SUPPRESS_PREPARED_QUERY</td>
<td>Set to 'yes' to prevent Tableau from using a prepared query as a means of reading metadata. A prepared query is often the fastest way to accurately read metadata. However, not all database systems are capable of reporting</td>
</tr>
</tbody>
</table>
metadata for a prepared query without actually executing the query. Note that certain metadata -- for example from connections using Custom SQL -- cannot be retrieved if this capability and CAP_ODBC_METADATA_SUPPRESS_EXECUTED_QUERY are both set.

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAP_ODBC_METADATA_SUPPRESS_SELECT_STAR</td>
<td>Set to 'yes' to prevent reading metadata using a 'select *' query.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_SUPPRESS_SQLCOLUMNS_API</td>
<td>Set to 'yes' to prevent Tableau from using older, less accurate API for reading metadata from ODBC data sources. Setting this capability allows Tableau to read metadata by issuing a full 'select *' query, which is expensive but may enable connectivity for extremely limited or unstable data sources.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_SUPPRESS_SQLFOREIGNKEYS_API</td>
<td>Set to 'yes' to prevent Tableau from attempting to read metadata describing foreign key constraints. Despite the simple nature of this ODBC API, some drivers may have unstable behavior or produce inaccurate results. Setting this capability may force Tableau to generate less efficient queries involving multi-table joins.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_SUPPRESS_SQLPRIMARYKEYS_API</td>
<td>Set to 'yes' to prevent Tableau from reading primary key metadata using the SQLPrimaryKeys API or an equivalent query. This capability is available in Tableau 9.1 and later.</td>
</tr>
<tr>
<td>CAP_ODBC_METADATA_SUPPRESS_SQLSTATISTICS_API</td>
<td>Set to 'yes' to prevent reading unique constraints and table cardinality estimates using the SQLStatistics API or an equivalent query. This capability is available in Tableau 9.0 and later.</td>
</tr>
<tr>
<td>CAP_ODBC_REBIND_SKIP_UNBIND</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to rebind a column directly and skip</td>
</tr>
<tr>
<td>Setting</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CAP_ODBC_SUPPRESS_INFO_SCHEMA_STORED_PROCS</td>
<td>Set to 'yes' to prevent the INFORMATION.SCHEMA schema from being queried when enumerating stored procedures.</td>
</tr>
<tr>
<td>CAP_ODBC_SUPPRESS_PREPARED_QUERY_FOR_ALL_COMMAND_QUERIES</td>
<td>Set to 'yes' to execute all commands directly (i.e., no prepared statement).</td>
</tr>
<tr>
<td>CAP_ODBC_SUPPRESS_PREPARED_QUERY_FOR_DDL_COMMAND_QUERIES</td>
<td>Set to 'yes' to execute DDL commands (e.g. CREATE TABLE) directly (i.e., no prepared statement).</td>
</tr>
<tr>
<td>CAP_ODBC_SUPPRESS_PREPARED_QUERY_FOR_DML_COMMAND_QUERIES</td>
<td>Set to 'yes' to execute DML commands (e.g. INSERT INTO) directly (i.e., no prepared statement).</td>
</tr>
<tr>
<td>CAP_ODBC_SUPPRESS_SYS_SCHEMA_STORED_PROCS</td>
<td>Set to 'yes' to explicitly add the &quot;SYS&quot; schema to the schema exclusions when enumerating stored procedures.</td>
</tr>
<tr>
<td>CAP_ODBC_TRANSACTIONS_COMMIT_INVALIDATES_PREPARED_QUERY</td>
<td>Set to 'yes' to indicate that a transaction will invalidate all prepared statements and close any open cursors.</td>
</tr>
<tr>
<td>CAP_ODBC_TRANSACTIONS_SUPPRESS_AUTO_COMMIT</td>
<td>Set to 'yes' to prevent the Native ODBC Protocol from using default auto-committing transaction behavior in ODBC. This capability cannot be used with CAP_ODBC_TRANSACTIONS_SUPPRESS_EXPLICIT_COMMIT.</td>
</tr>
<tr>
<td>CAP_ODBC_TRANSACTIONS_SUPPRESS_EXPLICIT_COMMIT</td>
<td>Set to 'yes' to prevent the Native ODBC Protocol from explicitly managing transactions. This capability cannot be used with CAP_ODBC_TRANSACTIONS_SUPPRESS_AUTO_COMMIT.</td>
</tr>
<tr>
<td>CAP_ODBC_TRIM_CHAR_LEAVE_</td>
<td>Set to 'yes' to leave whitespace padding at the unbinding, which reduces ODBC API calls when resizing buffers to refetch truncated data.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>PADDING</td>
<td>end of a character or text data type. Most data sources will trim this whitespace automatically, but the behavior depends on the driver.</td>
</tr>
<tr>
<td>CAP_ODBC_TRIM_VARCHAR_PADDING</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to trim trailing whitespace from VARCHAR columns which the driver has erroneously padded.</td>
</tr>
<tr>
<td>CAP_ODBC_UNBIND_AUTO</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to unbind and deallocate columns automatically, which can reduce ODBC API calls.</td>
</tr>
<tr>
<td>CAP_ODBC_UNBIND_BATCH</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to unbind and deallocate columns in a single batch operation, which can reduce ODBC API calls.</td>
</tr>
<tr>
<td>CAP_ODBC_UNBIND_EACH</td>
<td>Set to 'yes' to force the Tableau native ODBC protocol to unbind and deallocate columns individually, which may improve stability.</td>
</tr>
<tr>
<td>CAP_ODBC_UNBIND_PARAMETERS_BATCH</td>
<td>Set to 'yes' to unbind all parameters in a single batch operation.</td>
</tr>
<tr>
<td>CAP_ORACLE_SHOW_ALL_SYNONYMOwners</td>
<td>Set to 'yes' to list all the owners in the all_synonyms view for Oracle. This capability is available in 9.0 and later.</td>
</tr>
<tr>
<td>CAP_QUERY_BOOLEANEXPR_TO_INTEXP</td>
<td>Set to 'yes' if Tableau must coerce any Boolean expressions to an integer value in order include in a result set.</td>
</tr>
<tr>
<td>CAP_QUERY_FROM_REQUIRES_ALIAS</td>
<td>Set to 'yes' if the FROM clause must provide an alias for the given table.</td>
</tr>
<tr>
<td>CAP_QUERY_GROUP_ALLOW_DUPLICATES</td>
<td>Set to 'no' if SQL queries cannot contain duplicate expressions in the GROUP BY clause (this is uncommon).</td>
</tr>
<tr>
<td>CAP_QUERY_GROUP_BY_ALIAS</td>
<td>Set to 'yes' if SQL queries with aggregations can reference the grouping columns by their corresponding alias in the SELECT list, e.g. GROUP BY &quot;none_ShipCountry_nk&quot;.</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CAP_QUERY_GROUP_BY_DEGREE</td>
<td>Set to 'yes' if SQL queries with aggregations can reference the grouping columns by the ordinal position of each column, e.g. GROUP BY 2, 5. See also: CAP_QUERY_SORT_BY_DEGREE</td>
</tr>
<tr>
<td>CAP_QUERY_HAVINGQUIRES_GROUP_BY</td>
<td>Set to 'yes' if Tableau must use an artificial grouping field for any query which has a HAVING clause but no grouping columns.</td>
</tr>
<tr>
<td>CAP_QUERY_HAVINGUNSUPPORTED</td>
<td>Set to 'yes' if the SQL syntax for HAVING is unsupported. Tableau may be able to work around this using subqueries. See also: CAP_QUERY_SUBQUERIES.</td>
</tr>
<tr>
<td>CAP_QUERY_INCLUDEGROUP_BY_COLUMNS_IN_SELECT</td>
<td>Set to 'yes' to require all GROUP BY expressions to also appear in the SELECT expression list.</td>
</tr>
<tr>
<td>CAP_QUERY_JOIN ACROSS SCHEMAS</td>
<td>Set to 'yes' if SQL queries can express joins between tables located in different schemas.</td>
</tr>
<tr>
<td>CAP_QUERY_JOIN_ASSUMECONSTRAINED</td>
<td>Set to 'yes' to cull inner joins even if the database tables does do not have FK-PK relationships.</td>
</tr>
<tr>
<td>CAP_QUERY_JOIN_PUSH_DOWNCONDITION_EXPRESSIONS</td>
<td>Set to 'yes' to rewrite joins to simplify the ON clause conditions to simple identifier comparisons.</td>
</tr>
<tr>
<td>CAP_QUERY_JOINQUIRESSCOPE</td>
<td>Set to 'yes' if SQL queries must scope each join clause within parentheses to ensure a proper order of evaluation.</td>
</tr>
<tr>
<td>CAP_QUERY_JOINREQUIRESSUBQUERY</td>
<td>Set to 'yes' to force join expressions involving more than two tables to be composed with subqueries.</td>
</tr>
<tr>
<td>CAP_QUERY_NULL_REQUIRES_CAST</td>
<td>Set to 'yes' if the data source requires that all NULL literals are cast to an explicit data type.</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CAP_QUERY_SELECT_ALIASES_SORTED</td>
<td>Set to 'yes' if Tableau must impose a deterministic order on the SELECT expressions (sorted by alias) to ensure that query results can be properly matched with each field in the Tableau visualization. This is only required for data sources which do not preserve the aliases of the SELECT expressions when returning metadata with the query results.</td>
</tr>
<tr>
<td>CAP_QUERY_SORT_BY_DEGREE</td>
<td>Set to 'yes' if SQL queries can reference the sorting columns by the ordinal position of each column, e.g. ORDER BY 2, 5. See also: CAP_QUERY_GROUP_BY_DEGREE.</td>
</tr>
<tr>
<td>CAP_QUERY_SUBQUERIES</td>
<td>Set to 'yes' if the data source supports subqueries.</td>
</tr>
<tr>
<td>CAP_QUERY_SUBQUERIES_WITH_TOP</td>
<td>Set to 'yes' if the data source supports a TOP or LIMIT row-limiting clause within a subquery.</td>
</tr>
<tr>
<td>CAP_QUERY_SUBQUERY_DATASOURCE_CONTEXT</td>
<td>Set to 'yes' to use subquery filtered query context to implement data source filters. This capability is available in Tableau 8.0 through Tableau 9.3 only.</td>
</tr>
<tr>
<td>CAP_QUERY_SUBQUERY_QUERY_CONTEXT</td>
<td>Set to 'yes' to force Tableau to use a subquery for context filters instead of a temporary table or locally cached results.</td>
</tr>
<tr>
<td>CAP_QUERY_TOP_N</td>
<td>Set to 'yes' if the data source supports any form of row-limiting clause. The exact forms supported are described below.</td>
</tr>
<tr>
<td>CAP_QUERY_TOPSTYLE_LIMIT</td>
<td>Set to 'yes' if the data source uses LIMIT as the row-limiting clause.</td>
</tr>
<tr>
<td>Capability</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CAP_QUERY_TOPSTYLE RowNum</td>
<td>Set to 'yes' if the data source supports an Oracle-style filter on ROWNUM as the row-limiting clause.</td>
</tr>
<tr>
<td>CAP_QUERY_TOPSTYLE Top</td>
<td>Set to 'yes' if the data source uses TOP as the row-limiting clause.</td>
</tr>
<tr>
<td>CAP_QUERY_USE_QUERY Fusion</td>
<td>Set to 'no' to prevent Tableau from combining multiple individual queries into a single combined query. Turn off this capability for performance tuning or if the database is unable to process large queries. This capability is enabled by default and is available in Tableau 9.0 and later for all data sources except Tableau data extracts. Support for this capability in Tableau data extracts is available in Tableau 9.0.6.</td>
</tr>
<tr>
<td>CAP_SELECT INTO</td>
<td>Set to 'yes' if Tableau can create a table on the fly from the resultset of another query. See also: CAP_CREATE_TEMP_TABLES.</td>
</tr>
<tr>
<td>CAP_SELECT TOP INTO</td>
<td>Set to 'yes' if Tableau can use a TOP or LIMIT row-limiting clause when creating a table from a query resultset.</td>
</tr>
<tr>
<td>CAP_STORED_PROCEDURE PREFER Temp Table</td>
<td>Set to 'yes' to use a temporary table to support remote queries over the stored procedure result set.</td>
</tr>
<tr>
<td>CAP_STORED_PROCEDURE REPAIR Temp Table Strings</td>
<td>Set to 'yes' to attempt to compute actual string widths if metadata indicates no width or non-positive width.</td>
</tr>
<tr>
<td>CAP_STORED_PROCEDURE Temp Table From Buffer</td>
<td>Set to 'yes' to populate the temporary table from a result set buffered in entirety.</td>
</tr>
<tr>
<td>CAP_STORED_PROCEDURE Temp Table From New Protocol</td>
<td>Set to 'yes' to populate the temporary table from a separate protocol created for just this operation.</td>
</tr>
</tbody>
</table>
See also

Other Databases (ODBC) on page 600 – Describes how to connect to your data using the ODBC connector.

Tableau and ODBC on page 604 – Provides background information about ODBC, describes how Tableau determines the functionality of an ODBC driver, and lists frequently asked questions.

Customize and Tune ODBC Connections on page 609 – Describes how to fine tune the ODBC connection information for improved functionality and performance.

ODBC/SQL Customizations Reference below – Lists customizations that represent the parts of the ODBC and SQL standards that the ODBC driver reports supporting.

ODBC/SQL Customizations Reference

You can set the following customizations in the Tableau Datasource Customization (TDC) file to define parts of the ODBC and SQL standards that the ODBC driver supports.

In this article

Documentation resources for SQLGetInfo below
SQLGetInfo long-integer values on the next page
SQLGetInfo short-integer values on page 639
SQLGetInfo string values on page 639

Documentation resources for SQLGetInfo

The names of these customizations come from the identifiers used as parameters to SQLGetInfo.

For more information, see the MSDN documentation and the source code header file sqlext.h for the numeric and bit-mask values associated with each customization.
The Microsoft documentation for the ODBC API function SQLGetInfo:

A sample header file, sqlext.h, for the ODBC API identifier values used with SQLGetInfo:
http://cpansearch.perl.org/src/MIMER/DBD-Mimer-1.00/sqlext.h

SQLGetInfo long-integer values

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL_ODBC_INTERFACE_CONFORMANCE</td>
<td>Currently unused.</td>
</tr>
<tr>
<td>SQL_SQL_CONFORMANCE</td>
<td>Integer bitmask. Defines the level which the data source conforms to the SQL standard: '1' for entry-level SQL-92 conformance, '2' for FIPS 127-2 transitional, '4' for Intermediate and '8' for Full conformance.</td>
</tr>
<tr>
<td>SQL_CATALOG_USAGE</td>
<td>Integer bitmask. Defines the SQL statements in which a catalog identifier can be used.</td>
</tr>
<tr>
<td>SQL_SCHEMA_USAGE</td>
<td>Integer bitmask. Defines the SQL statements in which a schema identifier can be used.</td>
</tr>
<tr>
<td>SQL_AGGREGATE_FUNCTIONS</td>
<td>Integer bitmask. Defines which standard SQL aggregation forms are supported.</td>
</tr>
<tr>
<td>SQL_NUMERIC_FUNCTIONS</td>
<td>Integer bitmask. Defines which SQL scalar numeric functions are supported.</td>
</tr>
<tr>
<td>SQL_STRING_FUNCTIONS</td>
<td>Integer bitmask. Defines which SQL scalar string functions are supported.</td>
</tr>
<tr>
<td>SQL_TIMEDATE_FUNCTIONS</td>
<td>Integer bitmask. Defines which SQL scalar date / time functions are supported.</td>
</tr>
<tr>
<td>SQL_TIMEDATE_ADD_INTERVALS</td>
<td>Integer bitmask. Defines which date / time intervals are supported with the TIMESTAMPADD scalar function.</td>
</tr>
<tr>
<td>SQL_TIMEDATE_DIFF_INTERVALS</td>
<td>Integer bitmask. Defines which date / time intervals are supported with the TIMESTAMPDIFF scalar function.</td>
</tr>
<tr>
<td>SQL_DATETIME_LITERALS</td>
<td>Integer bitmask. Defines which SQL-92 literals are supported for representing DATE / TIME constants and INTERVALs.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SQL_SYSTEM_FUNCTIONS</td>
<td>Integer bitmask. Defines support for special SQL system scalar functions: IFNULL, DBNAME and USERNAME.</td>
</tr>
<tr>
<td>SQL_SQL92_VALUE_EXPRESSIONS</td>
<td>Integer bitmask. Defines which logical functions are supported for testing and manipulating values: CASE, CAST and NULLIF.</td>
</tr>
<tr>
<td>SQL_SQL92_NUMERIC_VALUE_FUNCTIONS</td>
<td>Integer bitmask. Defines which functions can produce a numeric value from non-numeric data, including: EXTRACT (for date / time part extraction), CHAR_LENGTH, CHARACTER_LENGTH and POSITION(.. IN ..).</td>
</tr>
<tr>
<td>SQL_SQL92_STRING_FUNCTIONS</td>
<td>Integer bitmask. Defines which string manipulation functions are supported.</td>
</tr>
<tr>
<td>SQL_SQL92_DATETIME_FUNCTIONS</td>
<td>Integer bitmask. Defines which date / time manipulation functions are supported for determining the current date, time or timestamp.</td>
</tr>
<tr>
<td>SQL_OJ_CAPABILITIES</td>
<td>Integer bitmask. Defines which type of outer joins are supported.</td>
</tr>
<tr>
<td>SQL_SQL92_RELATIONAL_JOIN_OPERATORS</td>
<td>Integer bitmask. Defines which types of JOIN operators are supported, e.g. INNER, OUTER.</td>
</tr>
<tr>
<td>SQL_SQL92_PREDICATES</td>
<td>Integer bitmask. Defines which predicates are supported for logical tests of values, e.g. IS NULL, LIKE, IN.</td>
</tr>
<tr>
<td>SQL_CONVERT_FUNCTIONS</td>
<td>Integer bitmask. Defines which ODBC scalar functions are supported for CASTing or CONVERTing one data type to another.</td>
</tr>
<tr>
<td>SQL_CONVERT_TINYINT</td>
<td>Integer bitmask. Determines which other data types that this named type can be converted to using the ODBC scalar function CONVERT.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SQL_CONVERT_SMALLINT</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_INTEGER</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_BIGINT</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_REAL</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_FLOAT</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_DOUBLE</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_CHAR</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_VARCHAR</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_LONGVARCHAR</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_DECIMAL</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_NUMERIC</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_BIT</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_GUID</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_BINARY</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_VARBINARY</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_LONGVARBINARY</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_DATE</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_TIME</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_TIMESTAMP</td>
<td>Integer bitmask. Same as above.</td>
</tr>
<tr>
<td>SQL_CONVERT_INTERVAL_DAY_TIME</td>
<td>Integer bitmask. Same as above.</td>
</tr>
</tbody>
</table>
### SQLGetInfo short-integer values

<table>
<thead>
<tr>
<th>SQL_CURR_COMMIT_BEHAVIOR</th>
<th>Short integer value. “0” if the prepared statements are closed upon commit for the data source.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL_MAX_IDENTIFIER_LEN</td>
<td>Integer value. Defines the maximum number of characters that can be used in an identifier. Tableau leaves room for one extra character as the string terminator.</td>
</tr>
<tr>
<td>SQL_TXN_CAPABLE</td>
<td>Short integer value. “0” if the data source does not support the transaction.</td>
</tr>
<tr>
<td>SQL_QUOTED_IDENTIFIER_CASE</td>
<td>Integer bitmask.</td>
</tr>
</tbody>
</table>

### SQLGetInfo string values

<table>
<thead>
<tr>
<th>SQL_COLUMN_ALIAS</th>
<th>Boolean value. “Y” if the data source supports using aliases for columns listed in the SELECT clause.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL_IDENTIFIER_QUOTE_CHAR</td>
<td>String value. Indicates the character which can be used for quoting identifiers. Because the connection customization is an XML document, any entities must properly be encoded. For example, double-quotes will be ’&quot;’. Additionally this character is assumed to work as the opening and closing character around identifiers, so some data sources which require ‘[]’ to enclose identifiers will not be supported.</td>
</tr>
<tr>
<td>SQL_CATALOG_NAME_SEP_ATOR</td>
<td>Character value. Indicates the separator character to use between identifiers when qualifying them with a catalog, schema or table name. This is typically the period character.</td>
</tr>
<tr>
<td>SQL_SPECIAL_CHARS</td>
<td>String value. Indicates the special characters which are allowed in identifier strings.</td>
</tr>
<tr>
<td>SQL_CATALOG_TERM</td>
<td>String value. This is the descriptive term for a database</td>
</tr>
<tr>
<td>SQL_SCHEMA_TERM</td>
<td>String value. This is the descriptive term for a database schema, which appears in the Tableau connection dialog for this ODBC data source.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SQL_TABLE_TERM</td>
<td>String value. This is the descriptive term for a database table, which appears in the Tableau connection dialog for this ODBC data source.</td>
</tr>
<tr>
<td>SQL_DRIVER_NAME</td>
<td>String value. This is the name of the database ODBC driver.</td>
</tr>
<tr>
<td>SQL_DRIVER_VER</td>
<td>String value. This is the version number of the ODBC driver.</td>
</tr>
<tr>
<td>SQL_DRIVER_ODBC_VER</td>
<td>String value. This is the version of the ODBC API which the driver supports.</td>
</tr>
<tr>
<td>SQL_ODBC_VER</td>
<td>String value. This is the version of ODBC which the Windows ODBC Driver Manager supports. This should not need to be customized.</td>
</tr>
<tr>
<td>SQL_DBMS_NAME</td>
<td>String value. This is the name of the database vendor.</td>
</tr>
<tr>
<td>SQL_DBMS_VER</td>
<td>String value. This is the version of the database system.</td>
</tr>
<tr>
<td>SQL_SERVER_NAME</td>
<td>String value. This is the named network address of the database server.</td>
</tr>
<tr>
<td>SQL_USER_NAME</td>
<td>String value. This is the name of the currently authenticated user.</td>
</tr>
</tbody>
</table>

**See also**

Other Databases (ODBC) on page 600 – Describes how to connect to your data using the ODBC connector.
Tableau and ODBC on page 604 – Provides background information about ODBC, describes how Tableau determines the functionality of an ODBC driver, and lists frequently asked questions.

Customize and Tune ODBC Connections on page 609 – Describes how to fine tune the ODBC connection information for improved functionality and performance.

Tableau Capability Customizations Reference on page 619 – Lists customizations you can use to define which Tableau capabilities are supported by the data source.

Create a Data Source or Add a New Connection with Clipboard Data

Sometimes you want to pull in data from an outside source for some quick analysis. Rather than create a whole data source and then connect in Tableau, you can copy and paste the data directly into your workbook. Tableau automatically creates a data source that you can begin analyzing.

When you paste data on the data source page, Tableau creates a new connection in the existing data source.

When you paste data on the sheet, Tableau creates a new data source that you can begin analyzing. When you paste the data as a data source, the data source is saved as a text file to your Tableau Repository when you save the workbook.

You can copy and paste data from a variety of office applications including Microsoft Excel and Microsoft Word. You can also copy and paste HTML tables from web pages. Tables that are copied as comma separated values or tab delimited can be pasted into Tableau.

**Note:** Not all applications use these formats when copying.

1. Select the data you want and copy it to the clipboard.

2. Open Tableau Desktop and do one of the following:
   1. On the data source page, select **Data > Paste Data as Connection** or **Paste Data as Data Source**.
2. On the sheet, select **Data > Paste** to paste the data as a data source.

3. Select **File > Save** to save the data source.

When you save the workbook, the data source either becomes a part of the existing data source or is added to your repository, depending on which of the methods you choose. If you paste the data as a data source, the data source is saved with the workbook when you save the workbook as a packaged workbook (.twbx).
Run Initial SQL

When connecting to some databases, you can specify an initial SQL command that will run when a connection is made to the database, for example, when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. Initial SQL is not run when your refresh your view. Note that this initial SQL is different than a custom SQL connection. A custom SQL connection defines a relation (or table) to issue queries against. For more information, see Connect to a Custom SQL Query on page 723.

You can use this command to:

- Set up temporary tables to use during the session.
- Set up a custom data environment.

You have the option to add an initial SQL command in the Server Connection dialog box or on the Data Source page.

Note: If your data source supports running an initial SQL statement, an Initial SQL link appears in the lower-left corner of the Server Connection dialog box. For information about your data source, see Supported Connectors on page 321.

To use initial SQL

1. In the Server Connection dialog box, click Initial SQL. Or, on the Data Source page, select Data > Initial SQL or Data > Query Banding and Initial SQL depending on the database you connect to.

2. Enter the SQL command into the Initial SQL dialog box. You can use the Insert drop-
down menu to pass parameters to your data source.

![Initial SQL dialog box](image)

**Note:** Tableau does not examine the statement for errors. This SQL statement is simply sent to the database when you connect.

Your software license may restrict you from using initial SQL with your connection. If you publish to Tableau Server, the server must be configured to allow Initial SQL statements. By default, the server software is configured to allow these statements to run when the workbook is loaded in a web browser. Administrators can disable the functionality on the Data Connections tab of the Tableau Server Configuration utility. If the server does not allow initial SQL statements, the workbook opens, but the initial SQL commands are not sent.

**Parameters in an initial SQL statement**

You can pass parameters to your data source in an initial SQL statement. There are several reasons why this is useful:

- You can configure impersonation using the `TableauServerUser` or `TableauServerUserFull` parameters.
- If your data source supports it, you can set up row-level security (for example, for Oracle VPD or SAP Sybase ASE) to make sure that users see only the data that they are authorized to see.
- You can provide more details in logging, for example, the Tableau version or the workbook name.
The following parameters are supported in an initial SQL statement:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example of returned value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TableauServerUser</td>
<td>The user name of the current server user. Use when setting up impersonation on the server. Returns an empty string if the user is not signed in to Tableau Server.</td>
<td>asmith</td>
</tr>
<tr>
<td>TableauServerUserFull</td>
<td>The user name and domain of the current server user. Use when setting up impersonation on the server. Returns an empty string if the user is not signed in to Tableau Server.</td>
<td>domain.lan\asmith</td>
</tr>
<tr>
<td>TableauApp</td>
<td>The name of the Tableau application.</td>
<td>Tableau Desktop Professional Tableau Server</td>
</tr>
<tr>
<td>TableauVersion</td>
<td>The version of the Tableau application.</td>
<td>9.3</td>
</tr>
<tr>
<td>WorkbookName</td>
<td>The name of the Tableau workbook. Use only in workbooks with an embedded data source.</td>
<td>Financial-Analysis</td>
</tr>
</tbody>
</table>

Examples

The following examples show different ways you can use parameters in an initial SQL statement.

- This example sets the security context on Microsoft SQL Server:
  
  ```sql
  EXECUTE AS USER = [TableauServerUser] WITH NO REVERT;
  ```

- This example shows how, on a DataStax data source, you can use parameters to add detail to logging or to set up a session variable to keep track of the data:
  
  ```sql
  SET TABLEAUVERSION [TableauVersion];
  ```

- This example can be used to help set up row-level security for Oracle VPD:
  
  ```sql
  begin
  ```
DBMS_SESSION.SET_IDENIFIER([TableauServerUser]);

eend;

Note: Oracle PL/SQL blocks require a trailing semicolon to terminate the block. Consult Oracle documentation for the proper syntax.

Defer execution to the server

You can defer an initial SQL statement so that it is executed only on the server. One reason to defer execution to the server is if you don’t have permission to execute the commands that set up impersonation. Use <ServerOnly></ServerOnly> tags to enclose the commands to be executed only on the server.

Example:

CREATE TEMP TABLE TempTable(x varchar(25));

INSERT INTO TempTable VALUES(1);

<ServerOnly>INSERT INTO TempTable VALUES(2);</ServerOnly>

Security and impersonation

If you use the TableauServerUser or TableauServerUserFull parameter in an initial SQL statement, you will create a dedicated connection that can’t be shared with other users. This will also restrict cache sharing, which can enhance security, but may also slow performance.

Set Up Data Sources

After you connect to your data, use the Data Source page to set up the data source and prepare your data for analysis. There are many optional configurations that you can make before you begin your analysis. The configurations that you make on the Data Source page creates the data source that Tableau uses to interpret and interact with your data.

The topics in this section describe how to use these configurations in order to optimize your data source for analysis.
Plan the Data Source

In this article

- What is a Tableau data source? below
- Know where your data is coming from on the next page
- Ways to combine your data in Tableau on page 649
- Relationships between data sets can impact your setup on page 651
- How do you want to use the Tableau Data Source? on page 656

At the center of Tableau is your data. How successful you are with exploring your data, building views, and answering questions about your data depends on how you set up the Tableau data source.

The data source is the conduit to your data and Tableau—the data source tells Tableau how to interpret and interact with your data. Even though you can set up the data source with just a few clicks, it’s helpful to know a few things before you begin that affect how you set up the data source.

What is a Tableau data source?

A Tableau data source is the link between your data and Tableau. It is essentially the sum of your data, the connection information, and the customizations that you make on top of data to work with it in Tableau. The data source can contain:

- Information about where the data is located, such as a file name and path or a network location. Or, details on how to connect to your data, such as database server name and server sign-in information.

- The names of any tables in the connection, as well as information about how the tables relate to each other.

- A layer of customizations that you make on top of your data but that are not part of the original data itself, such as calculations, groups, and renamed fields. For details, see Organize and Customize Fields in the Data Pane on page 962.

After you connect to your data and select the first table to interact with from your database, a basic data source is created for you.
Because your data can be captured and stored in so many different ways, you might need to do some additional set up of the Tableau data source before you begin your analysis.

If your data is stored in one table, there’s typically less set up required for your data source. However, you still might need to prepare the data for analysis. For details, see Tips for Working with Your Data on page 325.

When working with multiple tables that contain data that you want, you might need to combine the data. How you combine the data varies based on whether the tables are from the same or different databases (including the same or different Excel workbooks or folders).

Know where your data is coming from

When you set up a data source, it’s important to know whether your data is stored in one table or in multiple tables.

Data comes from a single table

If your data comes from one table, you can connect to your data, drag a table onto the canvas to create the data source, and then start building your view. For more information, see Connect to Your Data on page 319.

Data comes from multiple tables from the same database

If your data comes from multiple tables in the same database, you can connect to your data and then combine the tables in Tableau. In most cases, when your data is coming from a single database, the data source you set up in Tableau contains a single connection to a database that has all the tables that you need for your analysis.

When your data is stored in multiple tables from the same database, the following methods are available to combine your data depending on the data you’re working with:

- Union
- Join
- Data blending

Data comes from multiple tables from different databases

If your data comes from multiple tables in different databases, you also can connect to your data and then combine the tables in Tableau. But before combining the tables, you must decide between setting up:
• an individual data source for each connection to the database that contains the tables that you need for your analysis, or
• a single data source with multiple independent connections to all the tables you need for your analysis.

When your data is stored in tables from different databases, you can choose from the following methods to combine data:

• Data blending – supports one connection per data source, and one data source per database (or Excel or text file)
• Cross-database join – allows two or more connections per data source

**Ways to combine your data in Tableau**

In general, there are three methods for combining data in Tableau: join, blend, or union.

**Join**

Joining is a method for combining tables related by common fields (that is, common columns). The result of combining data using a join is a virtual table that extends horizontally by adding columns of data.

Tableau supports joins between tables in the same database and between tables in different databases, which is also known as a cross-database join.

For more information about how to join data in Tableau, see **Join Your Data** on page 657.
Data blending

Data blending is also another method that lets you combine data. When you use data blending to combine your data, you combine data in what is called a primary data source with common fields from one or more secondary data sources.

Data blending is useful when the data sources you’re working with contain data that's at different levels of detail. This means, when one data set captures data using greater or lesser granularity than the other data set. For example, when you want to combine a data source that has daily transactional data with a table that contains quarterly data.

Data blending is also useful when you want to join data but the databases you're using don’t allow joins—for example, data in cube data sources.

As with joins, the result of combining data using data blending is a virtual table that extends horizontally by adding columns of data.

For more information about how data blending works and how to blend your data in Tableau, see Blend Your Data on page 682.

Union

Unioning is a method for appending values (that is, rows) to tables. You can union tables if they have the same columns. The result of combining data using a union is a virtual table that has the same columns but extends vertically by adding rows of data.
You can union tables in one of two ways: manually or using wildcard search. For more information about how to union data using one of these methods in Tableau, see Union Your Data on page 708.

Relationships between data sets can impact your set up

Whether you are working with multiple tables of data from the same or different databases, you must consider the relationship between the tables you want to combine in conjunction with your analysis goals.

When thinking about the relationship one table has with another, ask yourself questions like “Is one table significantly larger than the other?” Or, “Does one table contain a different level of detail than the other?”

In many cases, it's important to ask these questions up front, before you set up the data source. But sometimes the problems aren't highlighted until you've set up the data source.

Before you set up the data source

If you haven't set up the data source, consider the following relationship scenarios and recommendations.

**When you need to enrich an existing table (star schema).** Use a join to combine data when you need to add more dimensions to a fact table, or a table with the quantitative information, with dimensions from other tables in the same database. This type of relationship between the tables is sometimes referred to as a star schema.

For example, suppose you have the following tables: A, B, C, D, and E.

**Table A (sales)**

<table>
<thead>
<tr>
<th>Date</th>
<th>Customer ID</th>
<th>Store ID</th>
<th>Product ID</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/01</td>
<td>1002</td>
<td>South1</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>12/13</td>
<td>3456</td>
<td>South2</td>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>12/18</td>
<td>1810</td>
<td>North1</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>
Table B (customer)

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Name</th>
<th>Phone number</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Henry Wilson</td>
<td>555-0100</td>
</tr>
<tr>
<td>02</td>
<td>Michelle Kim</td>
<td>555-0887</td>
</tr>
<tr>
<td>03</td>
<td>Ashley Garcia</td>
<td>555-0698</td>
</tr>
</tbody>
</table>

Table C (neighborhood)

<table>
<thead>
<tr>
<th>Store ID</th>
<th>Parking</th>
<th>Exits</th>
</tr>
</thead>
<tbody>
<tr>
<td>West1</td>
<td>Yes</td>
<td>2</td>
</tr>
<tr>
<td>West2</td>
<td>Yes</td>
<td>2</td>
</tr>
<tr>
<td>East1</td>
<td>Yes</td>
<td>1</td>
</tr>
</tbody>
</table>

Table D (time)

<table>
<thead>
<tr>
<th>Date</th>
<th>Open</th>
<th>Close</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/03</td>
<td>7:33</td>
<td>6:30</td>
</tr>
<tr>
<td>11/04</td>
<td>7:20</td>
<td>6:28</td>
</tr>
<tr>
<td>11/05</td>
<td>8:31</td>
<td>7:30</td>
</tr>
</tbody>
</table>

Table E (product)

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Color</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>White</td>
<td>Plastic</td>
</tr>
<tr>
<td>02</td>
<td>Rust</td>
<td>Fiberglass</td>
</tr>
<tr>
<td>03</td>
<td>Moss</td>
<td>Fiberglass</td>
</tr>
</tbody>
</table>

Table A contains sales data; the quantitative data in this table is the Quantity value. Table B contains customer data, table C contains neighborhood data, table D contains time data, and table E contains product data. All tables are related to the fact table, table A, but none of the dimension tables (B, C, D, and E) are related to each other.
When all tables have a chain of relationships (snowflake schema). Use a join to combine data when table C relates to table A via table B. The relationship between the tables can be described as a snowflake schema.

For example, suppose you have the following tables: A, B, and C.

Table A (sales)

<table>
<thead>
<tr>
<th>Date</th>
<th>Order ID</th>
<th>Customer Name</th>
<th>Product Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/01</td>
<td>134</td>
<td>Michelle</td>
<td>Main</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kim</td>
<td></td>
</tr>
<tr>
<td>12/13</td>
<td>135</td>
<td>Henry</td>
<td>Main</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wilson</td>
<td></td>
</tr>
<tr>
<td>12/18</td>
<td>136</td>
<td>Ashley</td>
<td>Holiday</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Garcia</td>
<td></td>
</tr>
</tbody>
</table>

Table B (store)

<table>
<thead>
<tr>
<th>Order ID</th>
<th>Branch ID</th>
<th>Requires Parking Validation</th>
<th>Staff</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>01</td>
<td>Yes</td>
<td>5</td>
</tr>
<tr>
<td>100</td>
<td>03</td>
<td>No</td>
<td>3</td>
</tr>
<tr>
<td>102</td>
<td>01</td>
<td>Yes</td>
<td>5</td>
</tr>
<tr>
<td>103</td>
<td>02</td>
<td>Yes</td>
<td>3</td>
</tr>
</tbody>
</table>
Table C (geographic)

<table>
<thead>
<tr>
<th>Branch ID</th>
<th>Neighborhood</th>
<th>County</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Central</td>
<td>King</td>
<td>Seattle</td>
</tr>
<tr>
<td>02</td>
<td>Downtown</td>
<td>King</td>
<td>Seattle</td>
</tr>
<tr>
<td>03</td>
<td>Marina</td>
<td>King</td>
<td>Kirkland</td>
</tr>
<tr>
<td>04</td>
<td>SoDo</td>
<td>King</td>
<td>Seattle</td>
</tr>
</tbody>
</table>

Table A contains sales data. Table B contains store data. Finally, table C contains geographic data. In this example, table A and table B relate to each by the columns highlighted in pink. Table B and table C relate to each other by the columns highlighted in orange. Tables A and C relate to each other because of their relationship to table B.

When tables have a different level of detail – multi-fact table: Use data blending when the measure values in the tables of data you need to combine are at a different level of detail. Different level of detail means that one data set captures the data using either greater or lesser granularity than the other data set. The relationship between the tables of data can be described as multi-fact tables.

For example, suppose you have the following tables: 1, 2, A, B, C, D, and E. Table 1 contains sales data and table 2 contains purchase data. Table B and C contain retailer and area data. Tables D and E contain supplier and region information. In this case, table 1 and its dimension tables capture transactional data at a different level of detail than table 2 and its dimension tables.
When each table has a large number of records: Cardinality refers to the uniqueness of data contained in a column. Cardinality plays a role when you’re working with tables that have a large number of records. When the tables you want to analyze contain many rows of data, queries can be slow (and performance of the overall data source is affected) so Tableau recommends that you choose a method for combining data based on the cardinality of the related columns between tables.

- **When related columns have a lot of repeated data – low cardinality**: For example, a table called Student might contain a Gender column that contains only two values: female and male. When related columns between tables have low cardinality, use data blending to combine data.

- **When related columns have highly unique data – high cardinality**: For example, a table called Policy Holders might contain an ID column that contains a unique value for every subscriber. When related columns between tables have high cardinality, use joins or cross-database joins to combine your data.

When you need to analyze both summary and detail information together: If you need to see both the summary of a calculation and the breakdown of the calculation in the same view, use data blending. For more information, see **Blend on Summary Data** on page 704.

After you set up the data source

If you’ve set up the data source and have already chosen a method for combing your data, consider the following scenarios and alternative recommendations in case your current method doesn’t produce the results you expect:
• **When you have created multiple connections to the same database:** Use data blending if a join doesn't produce the expected results. Unexpected results from a join can result from rows being counted twice because the data sets are at different levels of detail. Or, you're trying to create a view that shows both summary and detail information together. An alternative to using blending in this case is to use LOD calculations. For more information, see Level of Detail Expressions and Aggregation.

• **When you have created multiple data sources for tables in the same database:** Make sure you use a single connection and join tables that belong to the same database. Ensuring that the data source in your workbook uses one connection per database reduces the number of queries sent to the database, which improves the overall performance of the data source, views, and workbook.

• **When you need to add more data to an existing data source:** Use a join if the data is stored in the same database.

After you understand the relationship your data has to each other, also consider your analysis goals.

---

**How do you want to use the Tableau Data Source?**

When thinking about the data source that you're setting up to use for your analysis, ask yourself questions like, "Does my data need to be published to Tableau Server so that it can be shared?" Or "Does my data, stored in different tables, need to be refreshed in real-time?" Or "Can the different tables of data be updated at different intervals?"

Because the table combining methods that you can use are done at either the data source or workbook level, answers to above questions can also help you determine the best method for combining tables so that you can optimize your efforts when setting up the data sources.

<table>
<thead>
<tr>
<th>Tables in the same database</th>
<th>Tables in different databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Join</td>
<td>Union</td>
</tr>
<tr>
<td>Data blending</td>
<td>Cross-data-base join</td>
</tr>
</tbody>
</table>

**Repeatability**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Joins can</td>
<td>Unions can</td>
<td>Blends can be created for each sheet in the work-</td>
<td>Cross-data-base join</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Use the data source as a template to reuse within the current workbook. 

be created for each connection in the data source.

be created for each connection in the data source.

book.

base joins can be created for each data source in the workbook.

<table>
<thead>
<tr>
<th><strong>Reuse</strong></th>
<th>Data sources that contain joins can be shared or published for different analysis.</th>
<th>Data sources that contain unions can be shared or published to be used for different analysis.</th>
<th>Because blends are saved to the workbook (and not to the data source), if you need to use a blend, duplicate the sheet that uses the blend to reuse for different analysis.</th>
<th>Data sources that contain cross-data-base joins can be shared or published.</th>
</tr>
</thead>
</table>

| **Single source of truth** | Data sources that contain joins can be used as a single source of truth. | Data sources that contain unions can be used as a single source of truth. | Because blends are comprised of two or more data sources, the workbook (and not the data sources) that contains the blend can be shared as the single source of truth. | Data sources that contain cross-data-base joins can be used as a single source of truth. |

**Join Your Data**

In this article

- Overview of join types on page 659
- Combine tables from the same database on page 660
The data that you analyze in Tableau is often made up of a collection of tables that are related by specific fields (that is, columns). Joining is a method for combining the related data on those common fields. The result of combining data using a join is a virtual table that is typically extended horizontally by adding columns of data.

**Note:** When joining tables, the fields that you join on must have the same data type. If you change the data type after you join the tables, the join will break.

For example, suppose you are analyzing data for a publisher. The publisher might have two tables. The first table contains ID numbers, first name, last name, and publisher type. The second table contains ID numbers, price, royalty, and title of published books. The related field between the two tables might be ID.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Table 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ID</strong></td>
<td><strong>Book Title</strong></td>
</tr>
<tr>
<td>20-034 Adam Davis</td>
<td>Weather in the Alps</td>
</tr>
<tr>
<td>20-165 Ashley Garcia</td>
<td>My</td>
</tr>
</tbody>
</table>
In order to analyze these two tables together, you can join the tables on ID to answer questions like, "How much was paid in royalties for authors from a given publisher?". By combining tables using a join, you can view and use related data from different tables in your analysis.

<table>
<thead>
<tr>
<th>ID</th>
<th>First Name</th>
<th>Last Name</th>
<th>Publisher Type</th>
<th>Book Title</th>
<th>Price</th>
<th>Royalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>20034</td>
<td>Adam</td>
<td>Davis</td>
<td>Independent</td>
<td>The Magic Shoe Lace</td>
<td>15.99</td>
<td>7,000</td>
</tr>
<tr>
<td>20165</td>
<td>Ashley</td>
<td>Garcia</td>
<td>Big</td>
<td>Weather in the Alps</td>
<td>19.99</td>
<td>5,000</td>
</tr>
</tbody>
</table>

**Overview of join types**

In general, there are four types of joins that you can use to combine your data in Tableau: inner, left, right, and full outer. The tables you can join and the different join types you can use depend on the database or file you connect to. You can tell which join types your data supports by checking the join dialog after you've connected to your data and have at least two tables on the canvas.

<table>
<thead>
<tr>
<th>Join Type</th>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
</table>
| Inner     | When you use an inner join to combine tables, the result is a table that contains values that have matches in both tables.
| **Left** | When you use a left join to combine tables, the result is a table that contains all values from the left table and corresponding matches from the right table. When a value in the left table doesn't have a corresponding match in the right table, you see a null value in the data grid. |
| **Right** | When you use a right join to combine tables, the result is a table that contains all values from the right table and corresponding matches from the left table. When a value in the right table doesn't have a corresponding match in the left table, you see a null value in the data grid. |
| **Full outer** | When you use a full outer join to combine tables, the result is a table that contains all values from both tables. When a value from either table doesn't have a match with the other table, you see a null value in the data grid. |
| **Union** | Though union is not a type of join, union is another method for combining two or more tables by appending rows of data from one table to another. Ideally, the tables that you union have the same number of fields, and those fields have matching names and data types. For more information about union, see *Union Your Data* on page 708. |

**Combine tables from the same database**

If the tables you need to analyze are from the same database, or workbook (for Excel), or directory (for text) then use the following procedure to combine tables. Combining tables that are from the same database require only a single connection in the data source. Typically, joining tables from the same database yields better performance. This is because querying
data that is stored on the same database takes less time and leverages the native capabilities of the database to perform the join.

**Note:** Depending on the level of detail of the tables you want to combine, you might consider data blending instead. For more information, see Blend Your Data on page 682.

To join tables

1. **In Tableau Desktop:** on the start page, under Connect, click a connector to connect to your data. This step creates the first connection in the Tableau data source.

   **In web authoring:** Select New Workbook and connect to your data. This step creates the first connection in the Tableau data source.

2. Select the file, database, or schema, and then double-click or drag a table to the canvas.

   ![Tableau Desktop screenshot](image)

   **Note:** If you're authoring on the web or signed in to Tableau Server (from Tableau Desktop) while you are setting up the data source, you have access to recommended tables to help make combining your data easier. For more information, see Use Certified and Recommended Data Sources and Tables on page 737.

3. Double-click or drag another table to the canvas, and then click the join relationship to
add join clauses and select your join type.

4. Add one or more join clauses by selecting a field from one of the available tables used in the data source, a join operator, and a field from the added table. Inspect the join clause to make sure it reflects how you want to connect the tables.

For example, in a data source that has a table of order information and another for returns information, you could use an inner join to combine the two tables based on the **Order ID** field that exists in both tables.

Note: You can delete an unwanted join clauses by clicking the "x" that displays when you hover over the right side of the join clause.

5. When you are finished, close the Join dialog.
After you’ve created a join, review the data grid to make sure that the join produces the results that you expect. For more information, see Review join results in the data grid on page 671. To troubleshoot your join, see Troubleshoot joins on page 677.

Continue to prepare your data source for analysis. You can rename and reset fields, create calculations, clean your data with Data Interpreter, change the data types of fields, and so on.

About null values in join keys

In general, joins are performed at the database level. If the fields used to join tables contain null values, most databases return data without the rows that contain the null values. However, if you’ve set up your single-connection data source to use an Excel, text, or Salesforce connection, Tableau provides an additional option to allow you to join fields that contain null values with other fields that contain null values.

To join on null values

- After you’ve set up your data source, on the data source page, select Data > Join null values to null values.

For example, suppose you have two tables of data that you want to join: Orders_June and Orders_July.

<table>
<thead>
<tr>
<th>ID</th>
<th>Location</th>
<th>ID</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>New York</td>
<td>1</td>
<td>New York</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Miami</td>
<td>3</td>
<td>Miami</td>
</tr>
</tbody>
</table>

If you join on both the ID and Location fields, most databases return the following table of data:

Join (of Orders_June and Orders_July)

<table>
<thead>
<tr>
<th>ID</th>
<th>Location</th>
<th>ID(Orders_July)</th>
<th>Location (Orders_July)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>New York</td>
<td>1</td>
<td>New York</td>
</tr>
<tr>
<td>3</td>
<td>Miami</td>
<td>3</td>
<td>Miami</td>
</tr>
</tbody>
</table>
If you are using a single Excel, text, or Salesforce connection in your data source, select **Data > Join null values to null values** to return the following table:

<table>
<thead>
<tr>
<th>ID</th>
<th>Location</th>
<th>ID(Orders_July)</th>
<th>Location (Orders_July)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>New York</td>
<td>1</td>
<td>New York</td>
</tr>
<tr>
<td>2</td>
<td>null</td>
<td>2</td>
<td>null</td>
</tr>
<tr>
<td>3</td>
<td>Miami</td>
<td>3</td>
<td>Miami</td>
</tr>
</tbody>
</table>

**Note:** This option is available for single-connection data sources that use text, Excel, and Salesforce connections. If you add a second connection to a data source that uses this option, the join reverts back to the default behavior of excluding rows with null values.

**Combine tables from different databases**

Beginning with Tableau version 10.0, if the tables you need to analyze are stored in different databases, or workbooks (for Excel), or directories (for text), use the following procedure to combine tables using a **cross-database join**.

Cross-database joins require that you first set up a multi-connection data source—that is, you create a new connection to each database before you join tables. When you connect to multiple databases, a data source becomes a multi-connection data source. Multi-connection data sources can be advantageous when you need to analyze data for an organization that uses different internal systems or when you need to work with data that is managed separately by both internal and external groups.

**Note:** In many cases, using a cross-database join is the primary method for combining your data. However, there are some cases that you might need to combine your data using data blending instead. For more information, see **Blend Your Data on page 682**.

After you’ve combined tables using a cross-database join, Tableau colors the tables in the canvas and the columns in the data grid to show you which connection the data comes from.
To join tables from different databases

1. In **Tableau Desktop**: On the Start page, under **Connect**, click a connector to connect to your data. This step creates the first connection in the Tableau data source.

   In **web authoring**: Select New Workbook and connect to your data. This step creates the first connection in the Tableau data source.

2. Select the file, database, or schema, and then double-click or drag a table to the canvas.

3. In the left pane, under **Connections**, click the **Add** button (+ in web authoring) to add a new connection to the Tableau data source. A new connection is required if you have related data stored in another database.
Note: If the connector you want is not available from the Connect list, cross-database joins are not supported for the combination of sources that you want to join. This includes connections to cube data (e.g., Microsoft Analysis Services), most extract-only data (e.g., Google Analytics and OData), and Tableau Server data sources. Instead of joining tables, consider using data blending. For more information, see Blend Your Data on page 682.

4. Add one or more join clauses by selecting a field from one of the available tables used in the data source, a join operator, and a field from the added table. Inspect the join clause to make sure it reflects how you want to connect the tables.

For example, in a data source that has a table of order information and another table of returns information, you could join the two tables based on the Order ID field that exists in both tables. Select the type of join.
Note: You can delete an unwanted join clause by clicking the "x" that displays when you hover over the right-side of the join clause.

5. When you are finished, close the Join dialog box.

Tables and columns are colored to show you which connection the data comes from.

After you've created a cross-database join, continue to prepare your multi-connection data source for analysis. You can rename and reset fields, create calculations, clean your data with Data Interpreter, change the data types of fields, and so on.

To troubleshoot your join, see Troubleshoot joins on page 677.

About working with multi-connection data sources
Working with multi-connection data sources is just like working with any other data source, with a few caveats, discussed in this section.

Union data from within a connection
To union data, you must use text tables or Excel tables from the same connection. That is, you cannot union tables from different databases. In Tableau Desktop, you can union tables across different Excel workbooks and files in different folders. For more information, see the Union tables using wildcard search (Tableau Desktop) on page 712.

If you need to union data from different databases, use Tableau Prep.
Collation

Collation refers to the rules of a database that determine how string values should be compared and sorted. In most cases, the collation is handled by the database. However, when you work with cross-database joins, you might join columns that have different collations.

For example, suppose your cross-database join used a join key comprised of a case-sensitive column from SQL Server and a case-insensitive column from Oracle. In cases like this, Tableau maps certain collations to others to minimize interpreting values incorrectly.

The following rules are used in cross-database joins:

- If a column uses collation standards of the International Components for Unicode (ICU), Tableau uses the collation of the other column.
- If all columns use collation standards of the ICU, Tableau uses the collation of the column of the left table.
- If no columns use collation standards of the ICU, Tableau uses a binary collation. A binary collation means the locale of the database and data type of the columns determine how string values should be compared and sorted.

**Note:** Collation of Japanese characters, that is, Kana-sensitivity, depends on the database that you are connected to.

Calculations and multi-connection data sources

Only a subset of calculations can be used in a multi-connection data source.

**In Tableau Desktop:** You can use a specific calculation if it is both:

- Supported by all the connections in the multi-connection data source
- Supported by Tableau extracts.

**In web authoring (Tableau Online and Tableau Server):** You can use a specific calculation if it is supported by all the connections in the multi-connection data source.

Stored procedures

Stored procedures are not available for multi-connection data sources.
Pivot data from within a connection

To pivot data, you must use text columns or Excel columns from the same connection. That is, you cannot include columns from different databases in a pivot.

Make extract files the first connection (Tableau Desktop only)

When connecting to extract files in a multi-connection data source, make sure that the connection to the extract (.tde or .hyper) file is the first connection. This preserves any customizations that might be a part of the extract, including changes to default properties, calculated fields, groups, aliases, etc.

**Note:** If you need to connect to multiple extract files in your multi-connection data source, only the customizations in the extract in the first connection are preserved.

Extracts of multi-connection data sources that contain connections to file-based data (Tableau Desktop only)

If you're publishing an extract of a multi-connection data source that contains a connection to file-based data such as Excel, selecting the **Include external files** option puts a copy of the file-based data on the server as part of the data source. In this case, a copy of your file-based data can be downloaded and its contents accessed by other users. If there is sensitive information in the file-based data that you have intentionally excluded from your extract, **do not** select **Include external files** when you publish the data source.

For more information about publishing data sources, see **Publish a Data Source** on page 2516.
About queries and cross-database joins

For each connection, Tableau sends independent queries to the databases in the join. The results are stored in a temporary table, in the format of an extract file.

For example, suppose you create connections to two tables, dbo.listings and reviews$. These tables are stored in two different databases, SQL Server and Excel. Tableau queries the database in each connection independently. The database performs the query and applies customizations such as filters and calculations, and Tableau stores the results for each connection in a temporary table. In this example, FQ.Temp_1 is the temporary table for the connection to SQL Server and FQ.Temp_2 is the temporary table for the connection to Excel.

When you perform a cross-database join, the temporary tables are joined together by Tableau Desktop. These temporary tables are necessary for Tableau to perform cross-database joins.

After the tables have been joined, "topn" filter is applied to limit the number of values shown in the data grid to the first 1,000 rows. This filter is applied to help maintain responsiveness of the data grid and the overall performance of the Data Source page.

Joined tables
Review join results in the data grid

After you have created a join on the canvas, review the data grid to make sure the join produces the results that you expect. If the data grid displays data that you don't expect, you might need to modify the join.

Results in the data grid

- **No data**: If no data displays in the data grid, you might need to change the join type or a join field used in the join condition. If you suspect a mismatch between fields in the join, use a calculation instead. For more information, see Use calculations to resolve mismatches between fields in a join below.

- **Duplicate data**: If you see duplicate data, there a few things you can do. Consider changing the aggregation of the measure that use in your analysis, use a calculation, or use data blending instead. For more information about data blending, see Blend Your Data on page 682.

- **Missing data**: If some data is missing from the data grid, you might need to change the join type or a join field used in the join condition. Again, if you suspect a mismatch between fields in the join, use a calculation instead. For more information, see Use calculations to resolve mismatches between fields in a join below.

- **Many null values**: If you see many null values that you do not expect, you might need to change the join type from the full outer type to the inner type.

- **All null values for one table**: If all values for one table is null, there are no matches between the tables that you are joining. If this is not expected, consider changing the join type.

Use calculations to resolve mismatches between fields in a join

When the fields in a join condition don't match—that is a mismatch between the values in the fields used in a join condition, the data grid can show little or no data at all. A mismatch between fields can occur for several reasons but often caused by the differences in format of the string values or date values in the fields. In many cases, you can resolve mismatches between the fields in your join by using a calculation.

Most functions are available for you to use in a calculation to create and replace a field in the join condition, with the exception of aggregate functions and table calculation functions.
**String mismatch**

A common mismatch scenario when working with string data occurs when one of the fields on one side of the join condition is equivalent to two or more fields on the other side of the join condition. In this case, you can use a calculation to combine the two fields so that its format matches the other field in the join condition.

For example, suppose you want to join two tables that contain the following data:

<table>
<thead>
<tr>
<th>Patron</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>First name</strong></td>
<td><strong>Last name</strong></td>
</tr>
<tr>
<td>Alan</td>
<td>Wang</td>
</tr>
<tr>
<td>Andrew</td>
<td>Smith</td>
</tr>
<tr>
<td>Ashley</td>
<td>Garcia</td>
</tr>
<tr>
<td>Fred</td>
<td>Suzuki</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Member number</strong></th>
<th><strong>Emergency contact</strong></th>
<th><strong>Relationship</strong></th>
<th><strong>Emergency number</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam Davis</td>
<td>555-0324</td>
<td>Ellen Davis</td>
<td>Partner</td>
<td>555-0884</td>
</tr>
<tr>
<td>Alan Wang</td>
<td>555-0356</td>
<td>Jean Wilson</td>
<td>Mother</td>
<td>555-0327</td>
</tr>
<tr>
<td>Fred Suzuki</td>
<td>555-0188</td>
<td>Jim Suzuki</td>
<td>Brother</td>
<td>555-3188</td>
</tr>
<tr>
<td>Henry Wilson</td>
<td>555-0100</td>
<td>Laura Rodriguez</td>
<td>Partner</td>
<td>555-0103</td>
</tr>
</tbody>
</table>
The common fields between the two tables appear to be name. However, in the Patron table the first and last names are in separate columns and in the Contact table the first and last names are in the same column. To join the tables on names, you can use a calculation in the left side of the join condition to merge the first name and last name columns together.

The result is a calculated field on the left side of the join condition that is accessible only from the join dialog. This calculation converts the field in the Patron table into a format that now matches the format of the field in the Contact table on the right side of the join condition.
Using the calculation in the join produces the following combined table:

<table>
<thead>
<tr>
<th>First name</th>
<th>Last name</th>
<th>Branch</th>
<th>Member since</th>
<th>Units borrowed</th>
<th>Fees</th>
<th>Suggested limit</th>
<th>Name</th>
<th>Phone number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alan</td>
<td>Wang</td>
<td>North</td>
<td>2000</td>
<td>1</td>
<td>0</td>
<td>15</td>
<td>Alan Wang</td>
<td>555-0356</td>
</tr>
<tr>
<td>Fred</td>
<td>Suzuki</td>
<td>North</td>
<td>2000</td>
<td>52</td>
<td>.90</td>
<td>15</td>
<td>Fred Suzuki</td>
<td>555-0188</td>
</tr>
</tbody>
</table>

**Date mismatch**

A common mismatch scenario when working with date data occurs when the date values in one field of the join condition are captured at a different level of detail than the other field in the join condition. In this case you can use a calculation in the join condition to change the format of the field on one side of the join condition so that its format matches the other field in the join condition.

For example, suppose you have the following two tables of data:
<table>
<thead>
<tr>
<th>Date</th>
<th>Reservation type</th>
<th>Requester ID</th>
<th>ID</th>
<th>First name</th>
<th>Last name</th>
<th>Branch</th>
<th>Member since</th>
<th>Units borrowed</th>
<th>Fees</th>
<th>Suggested limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/29/2002</td>
<td>Non-profit</td>
<td>23344-8888</td>
<td>23250-2870</td>
<td>Alan</td>
<td>Wang</td>
<td>North</td>
<td>2000</td>
<td>1</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>5/5/2003</td>
<td>Non-profit</td>
<td>33401-5476</td>
<td>23344-8978</td>
<td>Andrew</td>
<td>Smith</td>
<td>North</td>
<td>2000</td>
<td>36</td>
<td>3.50</td>
<td>15</td>
</tr>
<tr>
<td>3/12/2004</td>
<td>Non-profit</td>
<td>33401-5476</td>
<td>23344-5566</td>
<td>Ashley</td>
<td>Garcia</td>
<td>South</td>
<td>2000</td>
<td>243</td>
<td>11.30</td>
<td>15</td>
</tr>
<tr>
<td>3/15/2006</td>
<td>City</td>
<td>21189-6980</td>
<td>90000-5122</td>
<td>Brian</td>
<td>Frank</td>
<td>East</td>
<td>2011</td>
<td>12</td>
<td>.10</td>
<td>10</td>
</tr>
<tr>
<td>7/8/2007</td>
<td>K-12</td>
<td>33401-5476</td>
<td>92149-1769</td>
<td>Elizabeth</td>
<td>Johnson</td>
<td>West</td>
<td>2010</td>
<td>19</td>
<td>.5</td>
<td>10</td>
</tr>
<tr>
<td>1/4/2008</td>
<td>Individual</td>
<td>56049-5523</td>
<td>23344-5566</td>
<td>Fred</td>
<td>Suzuki</td>
<td>North</td>
<td>2000</td>
<td>52</td>
<td>.90</td>
<td>15</td>
</tr>
<tr>
<td>2/14/2014</td>
<td>Non-profit</td>
<td>23344-5566</td>
<td>93950-2870</td>
<td>Jane</td>
<td>Johnson</td>
<td>West</td>
<td>2017</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>12/21/2015</td>
<td>Non-profit</td>
<td>23344-5566</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2/10/2016</td>
<td>Non-profit</td>
<td>23344-5566</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To find out more information about new patron behavior, joining the Patron table to the Projector Rental table might provide some insight about which library services motivate new memberships. The common fields between the two tables appear to be "Date" and "Member since." However, the date values in each field are captured at different levels of detail. To join
these tables on their respective date fields, use a combination of DATE functions in a
calculation on each side of the join condition to make the level of detail in each field match.

Using the calculation in the join produces the following combined table:

<table>
<thead>
<tr>
<th>Date</th>
<th>Reservation type</th>
<th>Requester ID</th>
<th>ID</th>
<th>First name</th>
<th>Last name</th>
<th>Branch</th>
<th>Member since</th>
<th>Units borrowed</th>
<th>Fees</th>
<th>Suggested limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/2000</td>
<td>Individual</td>
<td>233445-589</td>
<td>232502-870</td>
<td>Alan</td>
<td>Wang</td>
<td>North</td>
<td>2000</td>
<td>1</td>
<td>0.0-0</td>
<td>15</td>
</tr>
<tr>
<td>1/1/2000</td>
<td>Individual</td>
<td>233445-589</td>
<td>233445-589</td>
<td>Fred</td>
<td>Suzuki</td>
<td>North</td>
<td>2000</td>
<td>52</td>
<td>0.9-0</td>
<td>15</td>
</tr>
<tr>
<td>1/1/2000</td>
<td>Individual</td>
<td>233445-589</td>
<td>233445-566</td>
<td>Ashley</td>
<td>Garcia</td>
<td>South</td>
<td>2000</td>
<td>243</td>
<td>11.30</td>
<td>15</td>
</tr>
<tr>
<td>1/1/2000</td>
<td>Individual</td>
<td>233445-589</td>
<td>233448-978</td>
<td>Andrew</td>
<td>Smith</td>
<td>North</td>
<td>2000</td>
<td>36</td>
<td>3.5-0</td>
<td>15</td>
</tr>
</tbody>
</table>

To determine if a patron rented the projector in the same year he or she started his or her
membership, add one more clause to the join based on ID.
The result of the additional join condition shows that only one patron might have started his membership to rent a projector.

<table>
<thead>
<tr>
<th>Date</th>
<th>Reservation type</th>
<th>Requester ID</th>
<th>ID</th>
<th>First name</th>
<th>Last name</th>
<th>Branch</th>
<th>Joined</th>
<th>Units borrowed</th>
<th>Fees</th>
<th>Suggested limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/2000</td>
<td>Individual</td>
<td>233445-589</td>
<td>233445-589</td>
<td>Fred</td>
<td>Suzuki</td>
<td>North</td>
<td>2000</td>
<td>52</td>
<td>0.9-0</td>
<td>15</td>
</tr>
</tbody>
</table>

Troubleshoot joins

Over-counting values or duplicated data

When you connect to and join multiple tables together, you set up a denormalized version of the data. In some cases, Tableau treats the multiple tables as one table. When multiple tables are treated as one table, after the tables are joined, all tables are queried. This can cause values to be over-counted.

For example, suppose you have two tables. The first table contains employee information, such as employee ID and salary. The second table contains organizational information, such as department names.
If you join these tables on **Employee ID**, an employee's salary is counted more than once because the employee is associated with multiple departments.

<table>
<thead>
<tr>
<th>Employee ID</th>
<th>Name</th>
<th>Salary</th>
<th>Department</th>
<th>Hire Date</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>20107</td>
<td>Wilson, Henry</td>
<td>61,000</td>
<td>Support</td>
<td>7/28/2015</td>
<td>Operations</td>
</tr>
<tr>
<td>20107</td>
<td>Wilson, Henry</td>
<td>61,000</td>
<td>Sales</td>
<td>8/25/2016</td>
<td>Operations</td>
</tr>
</tbody>
</table>

To help resolve this issue, you can do one of the following:

- **Change the aggregation**: Depending on your analysis, you can use the MIN or AVG aggregation to remove over-counting.
  
  For example, if you change the aggregation of **Salary** from SUM to MIN or from SUM to AVG.

- **Create a calculated field that removes the duplicate values**: You can create a calculation that divides the sum of the field being duplicated by the number of instances of the field that is causing the duplication.
  
  For example, the Salary values are being duplicated by the number of instances of Employee ID for Wilson, Henry. In this case, in the view replace Salary with a calculated
field that uses the following formula: \( \text{SUM ([Salary])} / \text{COUNT ([Employee ID])} \).

You could also use a Level of Detail Expression to delete the duplicate data. For more information, see Removing Duplicate Data with LOD Calculations in the Tableau Knowledge Base.

- **Blend the data:** Instead of creating a join, which might duplicate your data, you can blend the data on a common field instead. For more information, see Blend Your Data on page 682.

- **Use custom SQL:** Tableau supports using custom SQL for connecting to multiple tables in a data source. You can pre-aggregate the tables before joining them with a GROUP BY clause. This option requires some knowledge about writing SQL queries and assistance from a database expert if possible. For more information about how to connect to a custom SQL query from Tableau, see Connect to a Custom SQL Query on page 723.

**Broken cross-database joins**

Before you join tables using a cross-database join, make sure that the data types of the join keys match. If the data types of the join keys don't match, the join breaks, which is indicated by a red exclamation point. To fix a broken join, use one of the following suggestions:

- You can change the format of a field (including its data type) in the join dialog to make the join keys match by creating a join calculation. A join calculation supports a subset of calculations that lets you modify the format of the join key in one or more tables that you want to combine. To create a join calculation, click the join relationship between the tables that have a broken join, click the field whose format needs to be modified, and then select Create Join Calculation. For more information, see Use calculations to resolve mismatches between fields in a join on page 671.

- For text or Excel-based data, modify the data type of one of the text or Excel fields in the join key using the data type menu in the data grid.

- **Tableau Desktop only:** For most connections, you can use the Convert to custom SQL option to change the data type of one of the fields in the join key, using a function like CAST (). The Convert to custom SQL option is available only when the data source contains only one connection. In this case, remove the second connection and then select Data > Convert to custom SQL.
Assuming Referential Integrity

In some cases, you can improve query performance by selecting the option to **Assume Referential Integrity** from the **Data** menu. When you use this option, Tableau will include the joined table in the query only if it is specifically referenced by fields in the view.

Using this setting is appropriate when you know that your data has referential integrity (see definition below) but your database is not enforcing or cannot enforce referential integrity. If you have the option of configuring referential integrity in your database that is a better option than using this setting because it can improve performance both in the database and in Tableau.

The **Assume Referential Integrity** option in Tableau can only affect performance on Tableau’s end. If your data does not have referential integrity and you turn on this setting, query results may not be reliable.

To understand what referential integrity is, imagine connecting to sales data that has two tables: Sales and Product Catalog. These two tables are shown below:

<table>
<thead>
<tr>
<th>Sales</th>
<th>Product Catalog</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product ID</strong></td>
<td><strong>Product ID</strong></td>
</tr>
<tr>
<td>(Foreign Key)</td>
<td>(Primary Key)</td>
</tr>
<tr>
<td><strong>Sale Amount</strong></td>
<td><strong>Product Name</strong></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>10 Inch Tablet</td>
</tr>
<tr>
<td>1</td>
<td>10/1/2012</td>
</tr>
<tr>
<td>2000</td>
<td>10/2/2012</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Definition:** Referential integrity is a database constraint that requires a relationship between two entities, ensuring that certain values of the primary key are used in one entity only when corresponding values of the foreign key are used in another entity.
<table>
<thead>
<tr>
<th>Product ID</th>
<th>Product Name</th>
<th>Product ID</th>
<th>Sale Amount</th>
<th>Transaction Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 Inch Tablet</td>
<td>1</td>
<td>100</td>
<td>10/1/2012</td>
</tr>
<tr>
<td>1</td>
<td>10 Inch Tablet</td>
<td>1</td>
<td>2000</td>
<td>10/2/2012</td>
</tr>
<tr>
<td>2</td>
<td>Smart Phone</td>
<td>2</td>
<td>50</td>
<td>9/30/2012</td>
</tr>
<tr>
<td>3</td>
<td>Desk Lamp</td>
<td>3</td>
<td>10</td>
<td>8/21/2012</td>
</tr>
<tr>
<td>4</td>
<td>Memory Stick</td>
<td>2</td>
<td>50</td>
<td>9/30/2012</td>
</tr>
</tbody>
</table>

Because all products that are sold must have a listing in the Product Catalog, every row in the Sales table has a matching row in the Product Catalog table. When these two tables are joined on Product ID, you end up with a table that looks like this:

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Product Name</th>
<th>Product ID</th>
<th>Sale Amount</th>
<th>Transaction Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 Inch Tablet</td>
<td>1</td>
<td>100</td>
<td>10/1/2012</td>
</tr>
<tr>
<td>1</td>
<td>10 Inch Tablet</td>
<td>1</td>
<td>2000</td>
<td>10/2/2012</td>
</tr>
<tr>
<td>2</td>
<td>Smart Phone</td>
<td>2</td>
<td>50</td>
<td>9/30/2012</td>
</tr>
<tr>
<td>3</td>
<td>Desk Lamp</td>
<td>3</td>
<td>10</td>
<td>8/21/2012</td>
</tr>
</tbody>
</table>

Now let's say you build a view to look at Sale Amount by Region. By default, dragging the Sale Amount field to the view may create a query like this:

```
SELECT SUM([Sales Amount]) FROM [Sales] S INNER JOIN [Product Catalog] P ON S.ProductID = P.ProductID
```

By selecting **Assume Referential Integrity**, you tell Tableau that the joined tables have referential integrity. In other words, you are confirming that the Sales table will always have a matching row in the Product Catalog table. Because that is true, Tableau doesn’t need any information from the Product Catalog table in order to return these results. When you drag the Sale Amount field into the view, Tableau can simplify the query to:

```
SELECT SUM([Sales Amount]) FROM [Sales]
```

This simplified query can often return quicker results because it removes the join operation. This option impacts only inner joins and does not affect data sources with a single table.
Blend Your Data

In this article

Prerequisites for data blending on the next page
Differences between joins and data blending on page 684
Blend your data on page 686
Data blending limitations on page 689

Data blending is a method for combining data that supplements a table of data from one data source with columns of data from another data source.

Usually you use joins to perform this kind of data combining, but there are times, depending on factors like the type of data and its granularity, when it's better to use data blending.

For example, suppose you have transactional data stored in Salesforce and quota data stored in an Excel workbook. The data you want to combine is stored in different databases, and the granularity of the data captured in each table is different in the two data sources, so data blending is the best way to combine this data.

Data blending is useful under the following conditions:

- **You want to combine data from different databases that are not supported by cross-database joins.**

  Cross-database joins do not support connections to cubes (for example, Oracle Essbase) or to some extract-only connections (for example, Google Analytics). In this case, set up individual data sources for the data you want to analyze, and then use data blending to combine the data sources on a single sheet.

- **Data is at different levels of detail.**

  Sometimes one data set captures data using greater or lesser granularity than the other data set.

  For example, suppose you are analyzing transactional data and quota data. Transactional data might capture all transactions. However, quota data might aggregate transactions at the quarter level. Because the transactional values are captured at different levels of detail in each data set, you should use data blending to combine the data.
Use data blending instead of joins under the following conditions:

- **Data needs cleaning.**
  
  If your tables do not match up with each other correctly after a join, set up data sources for each table, make any necessary customizations (that is, rename columns, change column data types, create groups, use calculations, etc.), and then use data blending to combine the data.

- **Joins cause duplicate data.**
  
  Duplicate data after a join is a symptom of data at different levels of detail. If you notice duplicate data, instead of creating a join, use data blending to blend on a common dimension instead.

- **You have lots of data.**
  
  Typically joins are recommended for combining data from the same database. Joins are handled by the database, which allows joins to leverage some of the database’s native capabilities. However, if you’re working with large sets of data, joins can put a strain on the database and significantly affect performance. In this case, data blending might help. Because Tableau handles combining the data after the data is aggregated, there is less data to combine. When there is less data to combine, generally, performance improves.

  **Note:** When you blend on a field with a high level of granularity, for example, date instead of year, queries can be slow.

**Prerequisites for data blending**

Your data must meet the following requirements in order for you to use data blending.

**Primary and secondary data sources**

Data blending requires a primary data source and at least one secondary data source. When you designate a primary data source, it functions as the main table or main data source. Any subsequent data sources that you use on the sheet are treated as a secondary data source. Only columns from the secondary data source that have corresponding matches in the primary data source appear in the view.

Using the same example from above, you designate the transactional data as the primary data source and the quota data as the secondary data source.
**Note:** Cube (multidimensional) data sources must be used as the primary data source. Cube data sources cannot be used as a secondary data source.

Defined relationship between the primary and secondary data sources

After designating primary and secondary data sources, you must define the common dimension or dimensions between the two data sources. This common dimension is called the *linking field*.

Continuing the example from above, when you blend transactional and quota data, the date field might be the linking field between the primary and secondary data sources.

- If the date field in the primary and secondary data sources have the same name,
  
  Tableau creates the relationship between the two fields and shows a link icon (🔗) next to the date field in the secondary data source when the field is in the view.

- If the two dimensions don’t have the same name, you can define a relationship that creates the correct mapping between the date fields in the primary and secondary data sources.

Differences between joins and data blending

Data blending simulates a traditional left join. The main difference between the two is *when* the join is performed with respect to aggregation.

**Left join**

When you use a left join to combine data, a query is sent to the database where the join is performed. Using a left join returns all rows from the left table and any rows from the right table that has a corresponding row match in the left table. The results of the join are then sent back to and aggregated by Tableau.

For example, suppose you have the following tables. If the common columns are **User ID** and **Patron ID**, a left join takes all the data from the left table, as well as all the data from the right table because each row has a corresponding row match in the left table.
Data blending

When you use data blending to combine data, a query is sent to the database for each data source that is used on the sheet. The results of the queries, including the aggregated data, are sent back to and combined by Tableau. The view uses all rows from the primary data source, the left table, and the aggregated rows from the secondary data source, the right table, based on the dimension of the linking fields. Dimension values are aggregated using the ATTR aggregate function, which means the aggregation returns a single value for all rows in the secondary data source. If there are multiple values for the rows, an asterisk (*) is shown. Measure values are aggregated based on how the field is aggregated in the view.

You can change the linking field or add more linking fields to include different or additional rows of data from the secondary data source in the blend, changing the aggregated values.

For example, suppose you have the following tables. If the linking fields are User ID and Patron ID, blending your data takes all of the data from the left table, and supplements the left table with the data from the right table. In this case, not all values can be a part of the resulting table because of the following:

- A row in the left table does not have a corresponding row match in the right table, as indicated by the null value.
- There are multiple corresponding values in the rows in the right table, as indicated by the asterisk (*).
Suppose you have the same tables as above, but the secondary data source contains a new field called Fines. Again, if the linking fields are User ID and Patron ID, blending your data takes all of the data from the left table, and supplements it with data from the right table. In this case, you see the same null value and asterisks in the previous example in addition to the following:

- Because the Fines field is a measure, you see the row values for the Fines field aggregated before the data in the right table is combined with the data in the left table.
- As with the previous example, a row in the left table does not have corresponding row for the Fines field, as indicated by the second null value.

Blend your data

You can use data blending when you have data in separate data sources that you want to analyze together on a single sheet. The following example demonstrates how to blend data from two data sources: an Excel data source and an SQL Server data source.

To blend your data

Step 1: Connect to your data and set up the data sources

1. Connect to a set of data and set up the data source on the data source page. This
example uses the **Sample - Superstore** data source.

2. Select **Data > New data source**, connect to the second set of data, and then set up the data source. This example uses a SQL Server data source that contains information about forecasted sales, called Sales Plan.

3. Click the sheet tab to start building your view.

**Step 2: Designate a primary data source**

Drag at least one field from your primary data source into the view to designate it as the primary data source.

1. In the **Data** pane, click the data source that you want to designate as the primary data source. In this example, **Sample - Superstore** is selected.

2. Drag the fields you want to use from the data source into the view. In this example, a view is created that shows Sales by Segment and Category.

**Step 3: Designate a secondary data source**

Any fields used in the view from data sources that are not the primary data source or active links automatically designate subsequent data sources as the secondary data source.

1. In the **Data** pane, click the data source that you want to designate as the secondary data source. In this example, the Sales Plan data source is selected.

   When you complete this step, an orange bar displays down the left side of the Data pane. The orange bar indicates the secondary data source. In addition to the orange bar, broken link icons display next to potential linking fields.

2. Click a broken link icon (🔗) to establish a relationship between the secondary and primary data sources. This is the field that determines the level of detail that Tableau should aggregate to. In this example, Segment is the linking field.

   If a broken link icon does not appear next to the field that should be the linking field or no broken links appear, see **Step 4: (Optional) Define or edit relationships** on the next page.
3. Drag the fields you want to use from the secondary data source into the view. In this example, Sales Plan field is used on the Detail card to change the level of detail of the view.

Step 4: (Optional) Define or edit relationships

Tableau detects when a field from the primary data source also exists in a secondary data source, and indicates that the fields are potential linking fields by marking them with a broken link icon in the Data pane. You click a broken link icon to establish a relationship between the primary and secondary data sources, and have Tableau blend data from both data sources on a single sheet.

You must have at least one linked field in order to use data from the secondary data source.

1. Select Data > Edit Relationships.
2. In the Relationships dialog box, verify that the primary data source is selected from the Primary data source drop-down list. In this example, the Sample - Superstore data source is selected.
3. Select the secondary data source in the Secondary data source pane, select Custom in the relationships list, and then click the Add button. In this example, the Sales Plan data source is selected.
4. In the Add/Edit Field Mapping dialog box, do the following:
   1. Select a field from the primary data source.
   2. Select a field from the secondary data source to establish the linking field or the relationship between the data sources even though the fields do not have the same name.
   3. Click OK.

   In this example, a relationship between Segment field in the Superstore – Sample data source is selected and Customer Segment field in the Sales Plan data source is selected. You can map these two fields to create a relationship even though they don’t have the same name.
5. (Optional) Continue to add and remove as many relationships as necessary and then click OK.
The related fields are shown in the secondary data source as potential linking fields.

6. Click the broken link icon (🔗) next to these fields in the data pane to make the relationship between the primary and secondary data sources active. In this case, the broken link icon next to Customer Segment is made into an active link icon (🔗).

If the related field from the primary data source is used in the view, the link becomes active automatically.

When data blending, the relationship matches values based on the member aliases. You can fix fields that don’t match by editing the aliases. For example, when you map a Segment field in the primary data source to the Segment field in the secondary data source, “Small Business” will not map correctly to “S. Business”. You must edit the aliases in one of the data sources. For more information, see Create Aliases to Rename Members in the View on page 977.

You can also use a secondary data source to re-alias the field values in a primary data source. For more information, see Alias Field Values Using Data Blending below.

Data blending limitations

There are some data blending limitations around non-additive aggregates, such as COUNTD, MEDIAN, and RAWSQLAGG. For more information, see Troubleshoot Data Blending on page 696.

(Back to top)

Alias Field Values Using Data Blending

Data blending is a method for combining data. Data blending works by supplementing the data in the primary data source with the data in the secondary data source.

Aliasing is the alternative name that you can assign to a value in a dimension field. You can use aliases to rename specific values within a dimension. This can be useful when you want to show more relevant or descriptive dimension values in your view than what the original data provides.

You can use data blending as a method for re-aliasing values in a data source by using field values from another data source. To do this, there must be a field in the secondary data source that contains aliases for a field in the primary data source.
For example, suppose you have a primary data source that contains information on fruit, including fruit names and code values that the fruit are associated with. A secondary data source contains a field with more descriptive code values.

<table>
<thead>
<tr>
<th>Primary data source</th>
<th>Secondary data source</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fruit</strong></td>
<td><strong>Fruit ID</strong></td>
</tr>
<tr>
<td>Apple</td>
<td>A</td>
</tr>
<tr>
<td>Banana</td>
<td>B</td>
</tr>
</tbody>
</table>

When you re-alias a field in your primary data source, you can create a view that replaces the original code values with the more descriptive code values.

**Note:** After you have used the secondary data source to provide aliases for the primary data source, you no longer need the secondary data source and can close it.

**To alias field values**

1. Set up the primary and secondary data sources and establish a relationship between the data sources. For more information, see Step 4: (Optional) Define or edit relationships on page 688. The data sources used in this example are Fruit Stand and Alternate ID.

2. Select the primary data source in the **Data** pane, and then drag the field that you want to alias to the view.

   For this example, Fruit Stand is the primary data source.

3. Select the secondary data source in the **Data** pane, and drag the field that contains the alias information to the view.

   For this example, Alternate ID is the secondary data source. **Note:** The field you are using to alias the field values in the primary data source cannot be the linking field you are using to link the primary and secondary data sources.
In this example, Alternate ID field in the secondary data source contains the aliasing information needed by the Fruit Stand field in the primary data source.

<table>
<thead>
<tr>
<th>Fruit ID</th>
<th>Alternate ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>APP</td>
</tr>
<tr>
<td>B</td>
<td>BAN</td>
</tr>
<tr>
<td>C</td>
<td>CAN</td>
</tr>
<tr>
<td>D</td>
<td>DUR</td>
</tr>
<tr>
<td>E</td>
<td>EMB</td>
</tr>
</tbody>
</table>

4. Right-click (control-click on a Mac) the field from the secondary data source and select **Edit Primary Aliases**.

Values in the primary data source must have corresponding alias values in the secondary data source. An asterisk under **Has Alias** confirms that the data sources have corresponding alias values.
Note: If asterisks don't appear under Has Alias, then either the secondary data source doesn't have the corresponding aliases to match the field values in the primary data source, or there is more than one field value in the primary data source that has the same alias in the secondary data source. In cases like this, aliasing field values in the primary data source with data from the secondary data source is not possible.

5. Click OK. The alias values in the secondary data source replace the field values in the primary data source. In this example, the more descriptive code values from Alternate ID field replaces the values in the Fruit ID field.

6. (Optional) Select the secondary data source from the Data menu, and select Close.

To watch a video that demonstrates aliasing field values using data blending, see Cleaning Data by Bulk Re-Aliasing.

Bring a Field into the Primary Data Source

Data blending is a method for combining data. Data blending works by supplementing the data in the primary data source with the data in the secondary data source.

In cases where you might only need just some of the data in the secondary data source, you can create a primary group to bring only the fields that you need into the primary data source. This eliminates the need for the secondary data source, which reduces the size of the workbook and can improve workbook performance.

For example, suppose you have two data sources: Superstore and Population. The Superstore data source contains lots of data, but most importantly, it contains data about states and the region each of the states is associated with. The Population data source contains state and population data, but doesn't contain any data about regions. In this case, you can blend the Population and Superstore data sources to create a primary group. Then so that you can pull in
the Region field from the Superstore data source into the Population data source, thereby eliminating the need for the Superstore data source all together.

To blend two data sources and create a primary group

1. Download and open the following workbook from Tableau Public.
   
   
   2. Click Download Workbook in the upper-right corner and then open the workbook.

   In the workbook, the State dimension is already on the Rows shelf, making it the first field in the view and establishing the Population data source as the primary data source for the sheet. The Superstore data source is also included in the workbook and functions as the secondary data source.

2. In the Data pane, select Superstore, and then drag the Region dimension to the Rows shelf and place it in front of State.

   The view shows the regions from the Superstore data source (secondary) and the states in the Population data source (primary) that are associated with those regions. There are two states, Alaska and Hawaii, from Population data source (primary) that are not associated with a specific region. These two states are assigned to a "null" region.
3. Right-click the **Region** field in the **Rows** shelf from the Superstore data source (secondary), and select **Create Primary Group**.

4. In the Edit Group dialog box, drag Alaska and Hawaii (from the "null" region) to the **West** region. This task assigns the states in the Population data source (primary) in the "null" region into a region.
5. Click **OK** to close the Edit Group dialog box.

6. In the **Data** pane, select the State data source (secondary). Now you see a new group field listed in the Population data source (primary) called **State (group)**.

7. Drag the **State (group)** dimension over the Region dimension on the **Rows** shelf to replace it.

8. Drag the **2016 Population** measure to the **Columns** shelf to see Population based on the regions from the Superstore (secondary) data source.

9. (Optional) You can close the Superstore data source (secondary), and then continue with your analysis. You can also publish the data source or workbook without the secondary data source. This is because the field you were using from the secondary data source, **Region**, is replaced by the primary group field that you created in the primary data source.
To watch a video that demonstrates another example of the Create a Primary Group functionality, see Cleaning Data by Bulk Re-Aliasing.

Troubleshoot Data Blending

In this article

Common warnings and errors when blending data sources on the next page
Asterisks show in the sheet on page 699
Null values appear after blending data sources on page 701
Blending issues after publishing data sources on page 702
Blending with a cube (multidimensional) data source on page 703
Other data blending issues on page 703

When you use data blending to combine your data from different data sources, there are a few common issues that might arise. This topic lists them and describes how you can respond to each situation.
Common warnings and errors when blending data sources

No relationship to the primary data source

When you drag a field from a secondary data source to the view, you might see a warning that says:

*Fields cannot be used from the [name of secondary data source] data source, because there is no relationship to the primary data source. In the Data pane, switch to the [name of secondary data source] data source, and click at least one link icon to blend these data sources.*

This warning occurs when you have no active links in the secondary data source

For example, suppose you have two data sources that are related by the **State** and **Date** fields. At least one of these fields must have the active link icon (🔗) next to it in the secondary data source. You can make the link active by clicking the link icon in the **Data** pane or by using the related field from the primary data source in the view.

The secondary data source may not have any relationships to the primary data source. Look in the **Data** pane for the link icon. Tableau automatically links fields that have the same name. If your fields do not have the same name, you must define a relationship between them. For more information, see Step 4: (Optional) Define or edit relationships on page 688.

Primary and secondary connections are from tables in the same data source

When you drag a field from a secondary data source to the view, you might see a warning that says:

*The primary and secondary connections are from tables in the same data source. Instead of linking the connections, use the Data menu to join the data. Joins can integrate data from many tables and may improve performance and filtering.*

This warning occurs when the workbook contains separate data sources that connect to the same database. Though you can combine data in this way, Tableau recommends that you use a join to combine data from the same database instead. Joins are typically handled by the database, which means that joins leverage some of the database's native performance capabilities.
Cannot blend the secondary data source because one or more fields use an unsupported aggregation

Data blending has some limitations regarding non-additive aggregates such as COUNTD, MEDIAN, and RAWSQLAGG. Non-additive aggregates are aggregate functions that produce results that cannot be aggregated along a dimension. Instead, the values have to be calculated individually. All Number functions, except for MAX and MIN, are non-additive aggregates. For more information, see Tableau Functions (by Category) on page 1380.

These limitations cause certain fields in the view to become invalid under certain circumstances. If you hover your mouse cursor over one of these invalid fields, you see the following error:

*Cannot blend the secondary data source because one or more fields use an unsupported aggregation.*

This error can occur for one of the following reasons:

- **Non-additive aggregates from the primary data source**: Non-additive aggregates are only supported in the primary data source if the data in the data source comes from a relational database that allows the use of temporary tables. To work around this issue,
consider creating an extract of your data source. Extracts support temporary tables.

- **Non-additive aggregates from the secondary data source:** Non-additive aggregates are only supported in the secondary data source if the linking field from the primary data source is included in the view and no non-aggregated dimensions from the secondary data source is included in the view. Some number functions can still be used if they include an additive aggregation. For example, use \( \text{ROUND}(	ext{SUM}([\text{Sales}]), 1) \) instead of \( \text{ROUND}([\text{Sales}], 1) \)

- **Non-additive aggregates from a multi-connection data source that uses a live connection:** Multi-connection data sources that connect to data using a live connection do not support temporary tables. Therefore, using a multi-connection data source that connects to data using a live connection prohibits the use of blending functionality with non-additive aggregates. To work around this issue, consider creating an extract of your multi-connection data source. Extracts support temporary tables.

- **LOD expressions from the secondary data source:** This error can also appear when you use a level of detail expression in a view that uses data blending. To resolve the error, make sure the linking field in the primary data source is in the view before you use an LOD expression from the secondary data source and remove any dimensions, including dimension filters, from the secondary data source.

- **Published data sources as the primary data source:** Because certain versions of Tableau Server does not support temporary tables, there are some limitations around non-additive aggregates.
  - For Tableau 8.3 and earlier, Tableau Server does not support temporary tables. Therefore, using a published data source as your primary data source prohibits the use of the blending functionality with non-additive aggregates.
  - For Tableau 9.0 and later, you can use COUNTD and MEDIAN with blending functionality in a published data source that is used as the primary data source. However, the other limitations listed above still apply.

**Asterisks show in the sheet**

When you blend data, make sure that there is only one matching value in the secondary data source for each mark in the primary data source. If there are multiple matching values, you see an asterisk in the view that results after you blend data.

For example, suppose you have two data sources: Population and Superstore. The primary data source, Population, has a field called **State**. The secondary data source, Superstore,
contains fields called **State** and **Segment**. In the secondary data source, each state has multiple segments. Alabama, for example, has three segments: Consumer, Corporate, and Home Office.

### Primary data source

<table>
<thead>
<tr>
<th>Data</th>
<th>Analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pop</td>
<td>Sup</td>
</tr>
</tbody>
</table>

### Secondary data source

When you blend the two data sources on the **State** field, you create a relationship where individual state values (in the primary data source) can have multiple segment values (in the secondary data source). In this case, multiple values for segments in the secondary data source for each corresponding state value in the primary data source cause asterisks to appear in the view, as demonstrated by the images below.

**Blended data**
Resolve asterisks in the sheet

Avoid asterisks in the sheet by making sure that there is only one matching value in the secondary data source for each mark in the primary data source. Though the way you ensure this depends on the data and the view you're trying to create, consider one of the following suggestions to resolve asterisks in the sheet.

- Add field with a higher level of granularity from the primary data source to the sheet.
  For example, suppose you have a view similar to the example used above. Consider adding a field such as City into the sheet because its values are at a higher level of granularity than a field like State.

- Rebuild the view to switch the primary and secondary data sources with each other. In general, you should make the data source whose values have a higher level of granularity the primary data source.
  For example, suppose you have a view similar to the example used above. Consider making Superstore the primary data source and Population the secondary data source.

Null values appear after blending data sources

Null values can sometimes appear in place of the data you want in the view when you use data blending. Null values can appears for a few reasons:

- The secondary data source does not contain values for the corresponding values in the primary data source.
- The data types of the fields you are blending on are different.
- The values in the primary and secondary data sources use different casing.

Data blending works by supplementing the data in the primary data source with data from the secondary data source based on the linking field. This means Tableau takes all the data in the primary data source, and only the corresponding matches in the secondary data source. If Tableau cannot identify at least one matching value in the secondary data source for the corresponding value in the primary data source, null values appear.

You can resolve this issue by doing the following:

- **Insert data in the secondary data source**: Insert missing data in the secondary data source so that all records in the primary data source have a match. For example, suppose your primary data source contains values for all 50 U.S. states, but your
secondary data source only has values for 30 U.S. states. To resolve this issue, you can add values for the remaining 20 U.S. states to the underlying data set of the secondary data source. Alternatively, you might be able to Bring a Field into the Primary Data Source on page 692 or Alias Field Values Using Data Blending on page 689 to work around null values if you cannot modify the underlying data.

- **Verify data types in the primary and secondary data sources match**: For example, both data sources should use string data for the fields you are blending on, or both fields should contain numbers. You can't blend fields where one field contains numbers and the other contains strings.

- **Verify the casing of the values in the primary and secondary data sources match**: If one data source uses a mix of upper- and lowercase names, and the other uses only uppercase, you'll see null values. For more information, see Nulls Show When Data Blending.

### Blending issues after publishing data sources

When you make a field-level customization that blends two data sources, the customization is based on one of the data sources, the primary data source. Then, when the primary and secondary data sources are published to Tableau Server or Tableau Online, the primary data source, which contains the customization (i.e., the reference to the secondary data source) no longer works. The customization no longer works because the relationship between the primary and secondary data sources is lost. This is because the relationship between the primary and secondary data sources is defined at the workbook level.

For example, suppose you have two data sources: Store - Main and Store - West. You create a field-level customization, like a calculation, that combines the sales totals from the primary data source, Store - Main, with the sales totals from the secondary data source, Store - West. As soon as you publish the data sources to Tableau Server or Tableau Online, the calculation in Store - Main no longer works. This is because the calculation refers to the secondary data source, Store - West, whose location is now undetermined.

To work around this scenario, do the following:

1. Before creating any field-level customizations, publish each data source first.
2. Connect to each data source.
3. Create a workbook that contains the field-level customizations that you need.
4. Publish the workbook to Tableau Server.
You can use this workbook as a "template" that can be shared and downloaded by you and others.

Blending with a cube (multidimensional) data source

Cube data sources can only be used as the primary data source for blending data in Tableau. They cannot be used as secondary data sources.

Other data blending issues

For other issues that emerge while blending data, see the following links.

Sort is not available

- For issues sorting on a calculated field, see Sorting by Fields is Unavailable for Data Blended Measures.
- For issues with a computed sort, see Sort Options Not Available from Toolbar When Data Blending.

Actions do not behave as expected

- Fields from the secondary data source cannot be added to a URL action. See Fields from Blended Data Source Unavailable for URL Actions.
- Action filters are not behaving as expected. See Action Filters with Blended Data Not Working as Expected.

Unexpected values and field changes

- Invalid fields when using COUNTD, MEDIAN, and RAWSQLAGG. See COUNTD Invalid in Published Data Sources When Blending.
- Duplicate totals after every date value in the view. See Issues with Blending on Date Fields.
- Latitude and longitude fields are grayed out. See Error "Invalid field formula" Creating
Map.

- Underlying data shows different values than blended data. See Underlying Data from Secondary Data Source Not Displayed or Consistent with Blended Data.

**Blend on Summary Data**

Data blending is a method for combining data. Data blending works by supplementing the data in the primary data source with the data in the secondary data source.

When one of the data sources that you're working with is large, you can reduce query time by using data blending to combine data in the primary data source with summary data from the secondary (large data source).

For example, suppose you have two data sources: Store - North and Store - South. Store - North is the primary data source and Store - South is the large secondary data source. To reduce query times when working with a large data source like Store - South, you can use data blending to blend only on the summary data of the large secondary data source (Store - South) with the data in the smaller and presumably faster primary data source (Store - North).

To blend on summary data, you can do the following:

1. Create a relationship between data sources.
2. Bring summary data from the secondary data source into the primary data source.
3. Compare primary data source data to summary data in the secondary data source.

**Step 1: Create a relationship between the primary and secondary data sources**

1. Download and open the following from Tableau Public.
   1. Go to
   2. Click **Download Workbook** in the upper-right corner.

   In the workbook, the Order Date dimension is already on the Columns shelf, making it the first field in the view and establishing *Store - North* as the primary data source for the sheet. The *Store - South* data source is also included in the workbook and functions as the large secondary data source.

2. Select **Data > Edit Relationships**.
   1. In the Relationships dialog box, select **Custom**, and then click **Add**.
4. In the Add/Edit Field Mapping dialog box, in both lists, click the arrow next to Order Date, select \texttt{Year(Order Data)}, and then click \texttt{OK}.

5. In the Relationships dialog box, click \texttt{Add} again.

6. In the Add/Edit Field Mapping dialog box, in both lists, click \texttt{Prod Type 1}, and then click \texttt{OK}.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{diagram.png}
\caption{Add/Edit Field Mapping dialog box.}
\end{figure}

Step 2: Bring summary data from the secondary data source into the primary data source

1. In the Data pane, select the Store - North data source.

2. Select \texttt{Analysis > Create Calculated Field}.

3. In the calculation dialog box, do the following:
   \begin{enumerate}
   \item In the name text box, enter \texttt{Sales for Store - South}.
   \item In the formula text box, type the \texttt{ZN} function, the \texttt{SUM} function, and the Sales Totals field from the Store - South data source.
   \end{enumerate}

   Your formula should look like the following:\texttt{ZN(SUM([Store - South]. [Sales Total]))}
This formula pulls in just the sales data from the large secondary data source.

\[
\text{ZN(SUM([Store - South].[Sales Total]))}
\]

4. Select **Analysis > Create Calculated Field**.

5. In the calculation dialog box, do the following:

   1. In the name text box, enter **Total Sales (North and South)**.

   2. In the formula text box, type the following:

   ```
   IF ISNULL([Sales for Store - South]) THEN 0 ELSE [Sales for Store - South] END
   +
   IF ISNULL (SUM([Sales Total])) THEN 0 ELSE SUM([Sales Total]) END
   ```

This formula adds the sales data from the large secondary data source to the sales data in the primary data source to get combined sales for Store - South and Store - North.
Step 3: Compare primary data source data to summary data of secondary data source

1. From the Data pane, drag **Order Date** to the Columns shelf and **Prod 1** to Rows shelf.
2. Drag **Measure Names** to the Rows shelf, and then to the Filters shelf.
3. In the Filter dialog box, click **None**, and then select **Sales Total**, **Sales for Store - South**, **Total Sales (North and South)** check boxes.

4. From the Data pane, drag **Measure Values** to Text.

The final result is a view that summarizes primary data source and secondary data source data on the Order Year and product category.
Note: The example sales data in both Store - North and Store - South are identical and therefore yield same results for both Sales Total and Sales for Store - South fields.

Union Your Data

In this article

- Union tables manually on page 711
- Union tables using wildcard search (Tableau Desktop)
  on page 712
  Expand search to find more Excel, text, JSON, .pdf data on page 714
- Rename, modify, or remove unions on page 717
- Matching field names or field ordering on page 718
- Metadata about unions on page 719
- Merge mismatched fields in the union on page 720
You can union your data to combine two or more tables by appending values (rows) from one table to another. To union your data in Tableau data source, the tables must come from the same connection.

If your data source supports union, the **New Union** option displays in the left pane of the data source page after you connect to your data. You can also refer to the following list to verify that your data source supports union:

**Tableau Desktop**

- Excel
- Text File
- Google Sheets
- JSON File
- PDF File
- Amazon Redshift
- Aster Database
- Cloudera Hadoop
- Google BigQuery
- Hortonworks Hadoop
- Vertica
- IBM DB2
- IBM PDA (Netezza)
- Microsoft SQL
- MySQL
- Oracle
- PostgreSQL
- Pivotal Greenplum Database
- SAP Sybase ASE
- SAP Sybase IQ
- Teradata

**Web authoring (Tableau Online and Tableau Server)**

- Excel
- Text File
- Amazon Aurora
- Amazon Redshift
- Google Cloud SQL
- IBM BigInsights
- IBM DB2
- IBM PDA (Netezza)
- MemSQL
- Microsoft SQL
- MySQL
- Pivotal Greenplum Database
- PostgreSQL
- SAP Sybase ASE
- SAP Sybase IQ
- Vertica
For best results, the tables that you combine using a union must have the same structure. That is, each table must have the same number of fields, and related fields must have matching field names and data types.

For example, suppose you have the following customer purchase information stored in three tables, separated by month. The table names are "May2016," "June2016," and "July2016."

<table>
<thead>
<tr>
<th>May2016</th>
<th>June2016</th>
<th>July2016</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Day</strong></td>
<td><strong>Customer</strong></td>
<td><strong>Purchases</strong></td>
</tr>
<tr>
<td>4</td>
<td>Lane</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>Chris</td>
<td>6</td>
</tr>
<tr>
<td>28</td>
<td>Juan</td>
<td>1</td>
</tr>
</tbody>
</table>

A union of these tables creates the following single table that contains all rows from all tables.

**Union**

<table>
<thead>
<tr>
<th><strong>Day</strong></th>
<th><strong>Customer</strong></th>
<th><strong>Purchases</strong></th>
<th><strong>Type</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Lane</td>
<td>5</td>
<td>Credit</td>
</tr>
<tr>
<td>10</td>
<td>Chris</td>
<td>6</td>
<td>Credit</td>
</tr>
<tr>
<td>28</td>
<td>Juan</td>
<td>1</td>
<td>Credit</td>
</tr>
<tr>
<td>1</td>
<td>Lisa</td>
<td>3</td>
<td>Credit</td>
</tr>
<tr>
<td>28</td>
<td>Isaac</td>
<td>4</td>
<td>Cash</td>
</tr>
<tr>
<td>28</td>
<td>Sam</td>
<td>2</td>
<td>Credit</td>
</tr>
<tr>
<td>Day</td>
<td>Customer</td>
<td>Purchases</td>
<td>Type</td>
</tr>
<tr>
<td>-----</td>
<td>----------</td>
<td>-----------</td>
<td>------</td>
</tr>
<tr>
<td>2</td>
<td>Mario</td>
<td>2</td>
<td>Credit</td>
</tr>
<tr>
<td>15</td>
<td>Wei</td>
<td>1</td>
<td>Cash</td>
</tr>
<tr>
<td>21</td>
<td>Jim</td>
<td>7</td>
<td>Cash</td>
</tr>
</tbody>
</table>

**Union tables manually**

Use this method to manually union distinct tables. This method allows you to drag individual tables from the left pane of the Data Source page and into the Union dialog box.

**To union tables manually**

1. On the data source page, double-click **New Union** to set up the union.

2. Drag a table from the left pane to the Union dialog box.

3. Select another table from the left pane and drag it directly below the first table.
4. Click Apply or OK to union.

**Union tables using wildcard search (Tableau Desktop)**

Use this method to set up search criteria to automatically include tables in your union. Use the wildcard character, which is an asterisk (*), to match a sequence or pattern of characters in the Excel workbook and worksheet names, Google Sheets workbook and worksheet names, text file names, JSON file names, .pdf file names, and database table names.

When working with Excel, text file data, JSON file, .pdf file data, you can also use this method to union files across folders, and worksheets across workbooks. Search is scoped to the selected connection. The connection and the tables available in a connection are shown on the left pane of the Data source page.
To union tables using wildcard search

1. On the data source page, double-click **New Union** to set up the union.

2. Click **Wildcard (automatic)** in the Union dialog box.

3. Enter the search criteria that you want Tableau to use to find tables to include in the union.
For example, you can enter *2016 in the Include text box to union tables in Excel worksheets that end with "2016" in their names. Search criteria like this will result in the union of May2016, June2016, and July2016 tables (Excel worksheets), from the selected connection. In this case, the connection is called Sales, and the connection made to the Excel workbook containing the worksheets you wanted was in the quarter_3 folder in the sales directory (e.g., Z:\sales\quarter_3).

4. Click Apply or OK to union.

Expand search to find more Excel, text, JSON, .pdf data

The tables initially available to union are scoped to the connection you've selected. If you want to union more tables that are located outside of the current folder (for Excel, text, JSON, .pdf files) or in a different workbook (for Excel worksheets), select one or both check boxes in the Union dialog box to expand your search.

For example, suppose you want to union all Excel worksheets that end with "2016" in its name outside of the current folder. The initial connection is made to an Excel workbook located in the same directory in the above example, Z:\sales\quarter_3.
• **Include**: If you enter *2016* in the **Include** text box and leave the remaining search criteria of the dialog as is, Tableau looks for all Excel worksheets that end with “2016” in its name inside the current folder. In the diagram below, the yellow highlighted item represents the current location, that is, the Excel workbook that you created a connection to in the "quarter_3". The green box represents the tables belonging to workbooks and sheets that are unioned as result of this search criteria.

• **Include + Expand search to subfolders**: If you enter *2016* in the **Include** text box and select the **Expand search to subfolders** check box, Tableau does the following:
  - Looks for all Excel worksheets that end with "2016" in their names inside the current folder.
  - Looks for additional Excel worksheets that end with "2016" in their names that are located in Excel workbooks in subfolders of the "quarter_3" folder.

In the diagram below, the yellow highlighted item represents the current location, that is, the Excel workbook that you created a connection to in the "quarter_3" folder. The green box represents the tables belonging to workbooks and worksheets that are unioned as a result of this search criteria.
• **Include + Expand search to parent folder**: If you enter *2016* in the Include text box and select the Expand search to parent folder check box, Tableau does the following:
  
  - Looks for all Excel worksheets that end with "2016" in their names inside the current folder, "quarter_3."
  
  - Looks for additional Excel worksheets that end with "2016" in their names that are located in parallel folders of the "quarter_3" folder. In this example, "quarter_4" is the parallel folder.

In the diagram below, the yellow highlighted item represents current location, that is, the Excel workbook that you created a connection to in the "quarter_3" folder. The green boxes represent the tables belonging to the workbook and worksheets that are unioned as a result of this search criteria.

• **Include + Expand search to subfolders + Expand search to parent folder**: If you enter *2016* in the Include text box and select both the Expand search to subfolders and Expand search to parent folder check boxes, Tableau does the following:
  
  - Looks for all Excel worksheets that end with "2016" in their names inside the current folder, "quarter_3."
  
  - Looks for additional Excel workbooks that are located in the subfolders of the current folder, "quarter_3."
  
  - Looks for additional Excel workbooks that are located in parallel folders and subfolders of the "quarter_3" folder. In this example, "quarter_4" is the parallel folder.

In the diagram below, the yellow highlighted item represents the current location, that is, the Excel workbook that you created a connection to. The green box represents the tables belonging to the workbook and worksheets that are unioned as a result of this search criteria.
Note: When working with Excel data, wildcard search includes named ranges but excludes tables found by Data Interpreter.

Rename, modify, or remove unions

Perform basic union tasks directly in the canvas of the Data Source page.

To rename a union

1. Double-click the union table on the canvas.
2. Enter a new name for the union.

To add or remove tables in the union

1. Click the union drop-down arrow and then select Edit Union.
2. You can drag additional tables that you want to union from the left pane, or hover over a table until the remove icon displays and then click the icon to remove the table.
3. Click **Apply** or **OK** to complete the task.

To remove a union

- Click the union drop-down arrow and select **Remove**.

Matching field names or field ordering

Tables in a union are combined by matching field names. When working with Excel, Google Sheets, text file, JSON file or .pdf file data, if there are no matching field names (or your tables do not contain column headers), you can tell Tableau to combine tables based on the order of the fields in the underlying data by creating the union and then selecting **Generate field names automatically** option from the union drop-down menu.
Metadata about unions

After you create a union, additional fields about the union are generated and added to the grid. The new fields provide information about where the original values in the union come from, including the sheet and table names. These fields are useful when unique information that is critical to your analysis is embedded in the sheet or table name.

For example, the tables used in the example above have unique month and year information stored in the table name instead of in the data itself. In this case, you can use the Table Name field that is generated by the union to access this information and use it in your analysis.

If a named range is used in a union, null values display under the Sheet field.
Note: You can use the fields generated by a union, such as `Sheet or Table Name`, as join keys. You can use a unioned table in a join with another table or unioned table.

Merge mismatched fields in the union

When field names in the union do not match, fields in the union contain null values. You can merge the non-matching fields into a single field using the merge option to remove the null values. When you use the merge option, the original fields are replaced by a new field that displays the first non-null value for each row in the non-matching fields.

You can also create your own calculation or, if possible, modify the underlying data to combine the non-matching fields.

For example, suppose a fourth table, "August2016", is added to the underlying data. Instead of the standard "Customer" field name, it contains an abbreviated version called "Cust."

**August2016**

<table>
<thead>
<tr>
<th>Day</th>
<th>Cust.</th>
<th>Purchases</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Maria</td>
<td>2</td>
<td>Credit</td>
</tr>
<tr>
<td>9</td>
<td>Kathy</td>
<td>1</td>
<td>Credit</td>
</tr>
<tr>
<td>18</td>
<td>Vijay</td>
<td>7</td>
<td>Cash</td>
</tr>
</tbody>
</table>

A union of these tables creates a single table that contains all rows from tables, with several null values. You can use the merge option to combine the related customer fields into a single field.

**Union (with null values)**

<table>
<thead>
<tr>
<th>Day</th>
<th>Customer</th>
<th>Purchases</th>
<th>Type</th>
<th>Cust.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Lane</td>
<td>5</td>
<td>Credit</td>
<td>null</td>
</tr>
<tr>
<td>10</td>
<td>Chris</td>
<td>6</td>
<td>Credit</td>
<td>null</td>
</tr>
</tbody>
</table>

**Union (with columns that have been merged)**

<table>
<thead>
<tr>
<th>Day</th>
<th>Purchases</th>
<th>Type</th>
<th>Customer, Cust.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>Credit</td>
<td>Lane</td>
</tr>
</tbody>
</table>
| Day | Customer | Purchases | Type  | Cus- 
|-----|----------|-----------|-------|---
| 28  | Juan     | 1         | Credit| null |
| 1   | Lisa     | 3         | Credit| null |
| 28  | Isaac    | 4         | Cash  | null |
| 28  | Sam      | 2         | Credit| null |
| 2   | Mario    | 2         | Credit| null |
| 15  | Wei      | 1         | Cash  | null |
| 21  | Jim      | 7         | Cash  | null |
| 7   | null     | 2         | Credit| Maria |
| 9   | null     | 1         | Credit| Kathy |
| 18  | null     | 7         | Cash  | Vijay |

<table>
<thead>
<tr>
<th>Day</th>
<th>Purchases</th>
<th>Type</th>
<th>Customer, Cust.</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6</td>
<td>Credit</td>
<td>Chris</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>Credit</td>
<td>Juan</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Credit</td>
<td>Lisa</td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>Cash</td>
<td>Isaac</td>
</tr>
<tr>
<td>28</td>
<td>2</td>
<td>Credit</td>
<td>Sam</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Credit</td>
<td>Mario</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>Cash</td>
<td>Wei</td>
</tr>
<tr>
<td>21</td>
<td>7</td>
<td>Cash</td>
<td>Jim</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>Credit</td>
<td>Maria</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>Credit</td>
<td>Kathy</td>
</tr>
<tr>
<td>18</td>
<td>7</td>
<td>Cash</td>
<td>Vijay</td>
</tr>
</tbody>
</table>

After you merge fields, you can use the field generated from the merge in a pivot or split, or use the field as a join key. You can also change the data type of the field generated from a merge.

To merge mismatched fields
1. Select two or more columns in the grid.
2. Click the column drop-down arrow, and then select **Merge mismatched fields**.

**To remove a merge**

- Click the column drop-down arrow of the merged field and select **Remove merge**.

**At a glance: Working with unions**

**Tableau Desktop and web authoring (Tableau Online and Tableau Server)**

- A unioned table can be used in a join.
- A unioned table can be used in a join with another unioned table.
- The fields generated by a union, **Sheet** and **Table name**, can be used as the join key.
- If a named range is used in union, null values display under the **Sheet** field.
- The field generated from a merge can be used in a pivot.
- The field generated from a merge can be used as a join key.
- The data type of the field generated from a merge can be changed.
- Union tables from within the same connection. That is, you cannot union tables from different databases.

**Tableau Desktop only**

- When working with Excel data, wildcard search includes named ranges but excludes tables found by Data Interpreter.
- The field generated from a merge can be used in a pivot or split.
- To union a JSON file, it must have a `.json`, `.txt`, or `.log` extension. For more information about working with JSON data, see **JSON File** on page 338.
- When using wildcard search to union tables in a `.pdf` file, the result of the union is scoped to the pages that were scanned in the initial `.pdf` file you connected to. For more information about working with data in `.pdf` files, see **PDF File** on page 344.
- Stored procedures cannot be unioned.
- When working with database data, you can convert your union into custom SQL.
Connect to a Custom SQL Query

In this article

Connect to a custom SQL query below
Examples of custom SQL queries on the next page
Edit a custom SQL query on page 730
Use parameters in a custom SQL query on page 730

For most databases, you can connect to a specific query rather than the entire data set. Because databases have slightly different SQL syntax from each other, the custom SQL you use to connect to one database might be different from the custom SQL you might use to connect to another. However, using custom SQL can be useful when you know exactly the information you need and understand how to write SQL queries.

Though there are several common reasons why you might use custom SQL, you can use custom SQL to union your data across tables, recast fields to perform cross-database joins, restructure or reduce the size of your data for analysis, etc.

**Note:** For Excel and text file data sources, this option is available only in workbooks that were created before Tableau Desktop 8.2 or when using Tableau Desktop on Windows with the legacy connection. To connect to Excel or text files using the legacy connection, connect to the file, and in the Open dialog box, click the **Open** drop-down menu, and then select **Open with Legacy Connection**.

Connect to a custom SQL query

1. After connecting to your data, double-click the **New Custom SQL** option on the Data Source page.

2. Type or paste the query into the text box.
3. When finished, click **OK**.

When you finish making the connection to your data, only relevant fields from the custom SQL query display in the data grid of the Data Source page.

**Examples of custom SQL queries**

**Combine your tables vertically (union)**

If you need to append data to each other, you can use the union option directly in Tableau. In some cases your database does not support this option, so you can use custom SQL instead.

For example, suppose you have the following two tables: November and December.

<table>
<thead>
<tr>
<th>November</th>
<th>December</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Guest ID</strong></td>
<td><strong>Party Size</strong></td>
</tr>
<tr>
<td>3005</td>
<td>4</td>
</tr>
<tr>
<td>3006</td>
<td>2</td>
</tr>
<tr>
<td>3007</td>
<td>2</td>
</tr>
<tr>
<td>3008</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
You can use the following custom SQL query to append the second table, December, to the first table, November:

```
SELECT * FROM November UNION ALL SELECT * FROM December
```

The result of the query looks like this in the data grid:

<table>
<thead>
<tr>
<th>Guest ID</th>
<th>Party Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>3005</td>
<td>4.00000</td>
</tr>
<tr>
<td>3006</td>
<td>2.00000</td>
</tr>
<tr>
<td>3007</td>
<td>2.00000</td>
</tr>
<tr>
<td>3008</td>
<td>2.00000</td>
</tr>
<tr>
<td>3009</td>
<td>2.00000</td>
</tr>
<tr>
<td>3010</td>
<td>2.00000</td>
</tr>
<tr>
<td>3011</td>
<td>4.00000</td>
</tr>
<tr>
<td>3012</td>
<td>5.00000</td>
</tr>
<tr>
<td>3013</td>
<td>3.00000</td>
</tr>
<tr>
<td>3014</td>
<td>4.00000</td>
</tr>
<tr>
<td>3015</td>
<td>2.00000</td>
</tr>
</tbody>
</table>

For more information about the union option, see Union Your Data on page 708.

**Change the data type of a field to do a cross-database join**

When you want to perform a join between two tables, the data type of the fields you join on must be the same. In cases when the data type of the fields are not the same, you can use custom SQL to change the data type (cast) the field before performing the join.

For example, suppose you want to join two tables, Main and Sub, using the Root and ID fields, respectively. The Root field is a number type and the ID field is a string type. You can use the following custom SQL query to change the data type of Root from a number to a string so that you can join the Main and Sub tables using the Root and ID fields.
SELECT [Main].[Root] AS [Root_Number]
CAST([Main].[Root] AS INT) AS [Root_String]
FROM [Main]

The result of this query shows the original Root field and the Root field cast as a string.

<table>
<thead>
<tr>
<th>#</th>
<th>Custom SQL Query</th>
<th>Atc</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Root_Number</td>
<td>Root_String</td>
</tr>
<tr>
<td>1</td>
<td>7,981.00</td>
<td>7981</td>
</tr>
<tr>
<td>2</td>
<td>740.00</td>
<td>740</td>
</tr>
<tr>
<td>3</td>
<td>741.00</td>
<td>741</td>
</tr>
<tr>
<td>4</td>
<td>742.00</td>
<td>742</td>
</tr>
<tr>
<td>5</td>
<td>1,760.00</td>
<td>1760</td>
</tr>
</tbody>
</table>

For more information about joins and cross-database joins, see Join Your Data on page 657.

**Reduce the size of your data**

When working with very large data sets, sometimes you can save time while working with your data if you reduce its size first.

For example, suppose you have a large table called FischerIris. You can use the following custom SQL query to retrieve the specified columns and records thereby reducing the size of the data set that you connect to from Tableau.

```sql
SELECT
    [FischerIris].[Species] AS [Species],
    [FischerIris].[Width] AS [Petal Width],
    COUNT([FischerIris].[ID]) AS [Num of Species]
FROM [FischerIris]
WHERE [FischerIris].[Organ] = 'Petal'
AND [FischerIris].[Width] > 15.0000
GROUP BY [FischerIris].[Species], [FischerIris].[Width]

Restructure your data (pivot)
In some cases, you might be working with a table that needs to be restructured before analysis. Though this type of task can be done directly in Tableau by using options like pivot, your database might not support it. In this case, you can use custom SQL instead.

For example, suppose you have the following table:

<table>
<thead>
<tr>
<th>Season ID</th>
<th>Items - Don't like</th>
<th>Items - Defective</th>
<th>Items - Too small</th>
<th>Items - Too big</th>
</tr>
</thead>
<tbody>
<tr>
<td>R000151493</td>
<td>1</td>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>R000151493</td>
<td>2</td>
<td>44</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>R000151495</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>R000151495</td>
<td>0</td>
<td>17</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>R000151497</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>R000151497</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>R000151789</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>R000151789</td>
<td>0</td>
<td>12</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>R000151813</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>R000151813</td>
<td>3</td>
<td>0</td>
<td>53</td>
<td>1</td>
</tr>
<tr>
<td>R000151815</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R000151815</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R000151855</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>R000151855</td>
<td>59</td>
<td>0</td>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td>R000151857</td>
<td>4</td>
<td>0</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>R000151857</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>R000153013</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>R000153013</td>
<td>9</td>
<td>1</td>
<td>19</td>
<td>0</td>
</tr>
</tbody>
</table>

To change its structure and optimize your data for analysis in Tableau, you can use the following custom SQL query:

```sql
SELECT Table1.Season ID AS [Season ID],
Table1.Items - Don't like AS [Quantity],
"Don't Like" AS [Reason]
FROM Table1
UNION ALL
SELECT Table1.Season ID AS [Season ID],
Table1.Items - Defective AS [Quantity],
"Defective" AS [Reason]
FROM Table1
UNION ALL
SELECT Table1.Season ID AS [Season ID],
Table1.Items - Too big AS [Quantity],
"Too Big" AS [Reason]
FROM Table1
UNION ALL
SELECT Table1.Season ID AS Season ID,
```
Table1.Items - Too small AS [Quantity]  
"Too Small" AS [Reason]  
FROM Table1

The result of the query looks like this in the data grid:

<table>
<thead>
<tr>
<th>Abc</th>
<th>Custom SQL Query</th>
<th>Abc</th>
<th>Custom SQL Query</th>
<th>Abc</th>
<th>Custom SQL Query</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Season ID</td>
<td></td>
<td>Quantity</td>
<td></td>
<td>Reason</td>
</tr>
<tr>
<td>R000151493</td>
<td>null</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151493</td>
<td>null</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151495</td>
<td>null</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151495</td>
<td>null</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151497</td>
<td>null</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151497</td>
<td>null</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151789</td>
<td>null</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151789</td>
<td>null</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151813</td>
<td>3</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151813</td>
<td>3</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151815</td>
<td>0</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151815</td>
<td>0</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151855</td>
<td>0</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
<tr>
<td>R000151855</td>
<td>59</td>
<td></td>
<td>null</td>
<td></td>
<td>Don’t Like</td>
</tr>
</tbody>
</table>

For more information about the pivot option, see Pivot Data from Columns to Rows on page 747.

**Combine (join) and aggregate your data**

If you need to combine tables and aggregate your data, you can use both a join and default aggregation type options directly in Tableau. In some cases you might need to use custom SQL instead.

For example, suppose you have the following two tables: Orders and Vendors.
You can use the following custom SQL query to find a count on the number of orders and do a left join on the Orders and Vendors tables:

```sql
SELECT Vendors.Name, COUNT(Orders.Order) AS Number Of Orders
FROM Orders
LEFT JOIN Vendors
ON Orders.VendorID=Vendors.VendorID
GROUP BY Name;
```

The result of the query looks like this:

<table>
<thead>
<tr>
<th>Vendors</th>
<th>Number Of Orders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packing You</td>
<td>1</td>
</tr>
<tr>
<td>ShipWise Express</td>
<td>1</td>
</tr>
<tr>
<td>Tminus Shipping</td>
<td>1</td>
</tr>
</tbody>
</table>

For more information about joins, see Join Your Data on page 657.

Errors when duplicate columns are referenced

If your custom SQL query references duplicate columns, you may get errors when trying to use one of the columns in your analysis in Tableau. This will happen even if the query is valid. For example, consider the following query:

```sql
SELECT * FROM authors, titleauthor WHERE authors.au_id = titleauthor.au_id
```

The query is valid, but the `au_id` field is ambiguous because in this case it exists in both the “authors” table and the “titleauthor” table. Tableau will connect to the query but you will get an
error anytime you try to use the \texttt{au\_id} field. This is because Tableau doesn’t know which table you are referring to.

\begin{quote}
\textbf{Note:} It is a best practice to define column aliases with an AS clause whenever possible in a Custom SQL Query. This is because each database has its own rules when it comes to automatically generating a column name whenever an alias is not used.
\end{quote}

\section*{Edit a custom SQL query}

\subsection*{To edit a custom SQL query}

1. On the data source page, in the canvas, hover over the custom SQL table until the arrow displays.

2. Click the arrow and then select \textit{Edit Custom SQL}.

3. In the dialog box, edit the custom SQL query.

\section*{Use parameters in a custom SQL query}

You can use parameters in a custom SQL query statement to replace a constant value with a dynamic value. You can then update the parameter in the workbook to modify the connection. For example, you may connect to a custom SQL query that provides web traffic data for a particular page that is specified by a pageID. Instead of using a constant value for the pageID value in the SQL query, you can insert a parameter. Then after finishing the connection, you can show a parameter control in the workbook. Use the parameter control to switch out the pageID and pull in data for each page of interest without having to edit or duplicate the connection.

You can create a parameter directly from the Custom SQL dialog box or use any parameters that are part of the workbook. If you create a new parameter, it becomes available for use in the workbook just like any other parameter. See \textit{Create Parameters} on page 1031 to learn more.

\subsection*{To add a parameter to a custom SQL query}

1. On the data source page, in the canvas, hover over the table until the edit icon displays, and then click the edit button.

2. At the bottom of the dialog box, click \textit{Insert Parameter}.

3. Select a constant value in the SQL statement and then, from the \textit{Insert Parameter} drop-down menu select the parameter you want to use instead. If you have not created a
parameter yet, select **Create a new parameter**. Follow the instructions in *Create Parameters* on page 1031 to create a parameter.

**Note:** Parameters can only replace literal values. They cannot replace expressions or identifiers such as table names.

In the example below, the custom SQL query returns all orders that are marked as Urgent priority. In the custom SQL statement, the order priority is the constant value. If you want to change the connection to see the High priority orders, you would have to edit the data source.

Instead of creating and maintaining many variations of the same query, you can replace the constant order priority value with a parameter. The parameter should contain all of the possible values for Order Priority.
After you create a parameter, you can insert it into the SQL statement to replace the constant value.

After you finish editing the connection, the new parameter is listed in the Parameters area at the bottom of the Data pane and the parameter control displays on the right side of the view. As you select different values, the connection updates.
Note: If you are using an extract, you must refresh the extract in order to reflect changes to the parameter. Publishing a data source that uses Custom SQL parameters includes the parameters. The parameters are transferred to any workbooks that connect to the data source.

See Also
Use Custom SQL and RAWSQL to perform advanced spatial analysis on page 1910

Use a Stored Procedure

A stored procedure is a subroutine available to applications that access a relational database system. When you connect to a SAP Sybase ASE, Microsoft SQL Server, or Teradata database with Tableau, you can use a stored procedure to define the connection.

For Oracle, you can use a table function to define the connection. Oracle table functions act similarly to stored procedures and are listed under Stored Procedures on the Data Source page.
When you create a data source using one of these data types, available procedures are listed under **Stored Procedures**, as shown in the Microsoft SQL Server example:

From the left pane, drag the procedure to the canvas or double-click one of the listed procedures. If parameters are available in the procedure, the Parameters dialog box automatically displays.

Instead of entering a value, you can use an existing Tableau parameter, or create a new Tableau parameter for the value:
If you then expose the Tableau parameter in the view, users are able to change the value of the parameter in the procedure interactively.

Notes on Stored Procedures

If you use stored procedures to define a data source for Tableau, keep the following in mind:

- If a stored procedure returns more than one result set, Tableau reads the first one and ignores the rest.
- If a stored procedure has output parameters, Tableau filters out the stored procedure.
- Stored procedures that have parameters of a non-scalar type are excluded.
- Result set columns that don't have matching types in Tableau (such as varbinary, geometry, and hierarchyid) are logged. If all result set columns map to unknown data types, Tableau displays a message:
  "The result set... has no usable columns."
- Stored procedures that return no result sets are listed on the data source page but fail if selected.
- If no value is provided for a parameter that the stored procedure requires, an error occurs. Tableau cannot determine in advance whether parameters are required.
- Tableau does not perform any transaction management for stored procedures. That is, stored procedure writers must not depend on Tableau to start transactions before invoking stored procedures, or to commit them afterward.
- Column names must be unique for stored procedures to work. If two columns have the same name, or if no name is provided, the procedure can result in an error.
- If there are multiple queries in a stored procedure (for example, to read values from
another table or to hold temporary combinations) each of the queries must return the same sets of columns in the same order (same names and data types). To ensure that column order and names match in the query results, you may need to explicitly \texttt{CAST} to ensure the data type is correct, for example \texttt{CAST(Username as VARCHAR(20))}, and explicitly name the columns. If a stored procedure does not follow these guidelines, an error message can result:

"InsertData: unbound column error"

- If there are multiple queries in a stored procedure (for example, to read values from another table or to hold temporary combinations) and the procedure is generating an error, try adding \texttt{SET NOCOUNT ON} to the top of the procedure. This prevents the message which shows the count of number of rows affected by a Transact-SQL statement from being returned as part of the result set for a query.

In addition, the following constraints apply for specific databases.

**Stored Procedure Constraints for Teradata Databases**

The following constraints apply for stored procedures on Teradata databases.

- Values must be provided for every parameter. If the user does not provide a value for one or more parameters, Tableau displays a Teradata database error stating there are too few values provided for the stored procedure.

**Stored Procedure Constraints for SQL Server Databases**

The following constraints apply for stored procedures on SQL Server databases.

- If the result set for a stored procedure contains columns of type IMAGE or TEXT, the stored procedure will fail with an "Incorrect syntax" error message.

- If the total width of the result set (number of bytes in each row) exceeds 8060, the stored procedure fails. This can occur with very wide tables (hundreds of columns) or with tables having large text columns, intended to hold thousands of characters of text.

- Tableau does not display stored procedures from schema "sys".

- If the user does not provide a value for one or more parameters that the procedure requires, Tableau displays a SQL Server database error in the form "The procedure requires a value for parameter @x but one was not provided."

- Stored procedures that contain multiple queries should follow the guidelines listed in Notes on Stored Procedures (above).
- Tableau Desktop does not support the Microsoft SQL Server TIME data type. When fields of this type are included in a stored procedure on a Microsoft SQL Server database, Tableau Desktop will not import them.

**Stored Procedure Constraints for SAP Sybase ASE Databases**

The following constraints apply for stored procedures on SAP Sybase ASE databases (Windows only).

- The database must have a properly configured remote server.
- If the user does not provide a value for one or more parameters that the procedure requires, Tableau displays a Sybase ASE database error in the form "The procedure requires a value for parameter @x but one was not provided."

**Use Certified and Recommended Data Sources and Tables**

To speed up the process of validating and setting up data, use certified and recommended data sources and tables. Here's how these options differ:

- *Certified data sources* are carefully chosen by site administrators and project leaders.
- *Recommended data sources* include personally certified ones and others automatically picked for you based on usage patterns at your organization.
- *Recommended tables* are database tables that are used frequently in data sources and workbooks published to your Tableau server.

By taking advantage of relevant content that's already available, you can avoid duplicating work and spend more time on your analysis.

**Note:** To access these features in Tableau Desktop, you need to sign in to Tableau Server or Tableau Online from the Server menu.

In this article

**Certified data sources published to Tableau Server or Tableau Online** on the next page

**Recommended data sources published to Tableau Server or Tableau Online** on page 739
Certified data sources published to Tableau Server or Tableau Online

Certified data sources appear with a unique certification badge. Hover over the badge to learn who certified the data source, read any descriptive notes they’ve provided, and confirm the connection type.

Here’s how certified data sources look when you connect to data in Tableau Desktop.

And here’s how you can identify certified data sources in Tableau Server or Tableau Online.
Recommended data sources published to Tableau Server or Tableau Online

On the right side of the data source setup page, click the **Recommended Data Sources** button to quickly access popular data sources at your organization.

![Recommended Data Sources](image)

**Note:** Tableau Server administrators can turn off recommendations features.

**Recommended tables**

Recommended tables appear in the left pane of the data source page. After you add a recommended table to the canvas, the Recommended list shows tables frequently joined to the table you added. If you join a recommended table, Tableau uses the most popular fields and join type by default. (Any foreign keys specified in a database override recommended joins).
Recommended data sources and tables are supported for a subset of data connectors. If you don't see the Recommended list while signed into Tableau Online or Tableau Server from Tableau Desktop, the current data connector doesn't support recommendations.

**Convert a Field to a Date Field**

Each database stores Date fields in slightly different ways. Tableau does its best in interpreting the date fields, but sometimes a field will be imported to Tableau as a text string or numerical field. If this happens, there are a few steps we can take to resolve the issue.

**Verifying Date Fields**

The date field might appear as a string when connecting to the Data Source... Or as a string in the Dimensions Pane.
When dates are interpreted as strings, you will lose all of the features and conveniences of working with date fields, such as drilling down, using date calculations, and switching between Continuous and Discrete measures.

If your dates aren't interpreted correctly, take the following steps in order:

1. **Change the data type of the field** below
2. **Create a calculation using the DATEPARSE function** on the next page
3. **Create a calculation using the DATE function** on page 746

**Change the data type of the field**

The first step in resolving a date field interpretation issue is to make sure the data type is set to **Date** or **Date & Time**.

1. In the **Data Source** or **Dimensions** pane, click the data type icon and change the data type to **Date** or **Date & Time**
2. Inspect the data in the view or Data Source pane. If you see many Null values, return the data type to String and proceed to Create a calculation using the DATEPARSE function below to correct the issue.

Create a calculation using the DATEPARSE function

Dates are stored in a nearly infinite array of formats. Some date fields have years before months, others separate the parts of the date with periods, and still others use a combination of formats. When Tableau cannot interpret a date field, it might be because the particular format cannot be translated.
The DATEPARSE function lets you clearly define which parts of your field are which parts of a date. In essence, you are creating a map that Tableau can use to translate the string into a date field. This map is referred to as the format.

**Note:** This function is available through the following connectors: non-legacy Excel and text file connections, Amazon EMR Hadoop Hive, Cloudera Hadoop, Google Sheets, Hortonworks Hadoop Hive, MapR Hadoop Hive, MySQL, Oracle, PostgreSQL, and Tableau extracts. Some formats may not be available for all connections.

### Creating the DATEPARSE calculation

1. Inspect the current format of your date field. Note where months, years, hours, and days are displayed in your field. You will need this information for the DATEPARSE function.

   - A. Day of month
   - B. Month
   - C. Year
   - D. Hour, minute, second, millisecond, period

2. Right click your date field in the Dimensions pane and select **Create > Calculated Field**.

3. In the dialog, write the DATEPARSE function. The DATEPARSE function has two parts: the format and the string. The string is field you wish to convert, which must be a string data type.

   - **A. Format**
   - **B. String**

The **format** is the guide Tableau will use to interpret the string as a date. Each part of a date or time string has a matching symbol, as seen in the table below. The format must
exactly duplicate the way that the date is displayed. For example, if a single year code ("Y") is used when the string has a 2-digit year code ("97"), the calculation might return null values.

**Note:** The exception to the format exactly duplicating the displayed string is that long form parts of dates ("September") can be formatted with four symbols ("September" = "MMMM").

The format must include all spaces, hyphens, and other non-alphanumeric symbols in order to interpret the string correctly.

**Date Field Symbols**

<table>
<thead>
<tr>
<th>Date Part</th>
<th>Symbol</th>
<th>Example String</th>
<th>Example Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>Y</td>
<td>2016, 97, 2</td>
<td>YYYY,YYY,YY,Y</td>
</tr>
<tr>
<td>Era</td>
<td>G</td>
<td>AD, Anno Domini</td>
<td>GGGG</td>
</tr>
<tr>
<td>Month</td>
<td>M</td>
<td>9, 12, Sep, September</td>
<td>M, MM, MMM, MMM</td>
</tr>
<tr>
<td>Week of year (1-52)</td>
<td>w</td>
<td>8,27</td>
<td>w, ww</td>
</tr>
<tr>
<td>Day of Month</td>
<td>d</td>
<td>1, 15</td>
<td>d, dd</td>
</tr>
<tr>
<td>Day of Year (1-365)</td>
<td>D</td>
<td>23, 143</td>
<td>DDD,DDD</td>
</tr>
<tr>
<td>Period</td>
<td>a</td>
<td>AM, am, PM</td>
<td>aa, aaaa</td>
</tr>
<tr>
<td>Hour (1-12), Hour (0-24)</td>
<td>h, H</td>
<td>1, 16, 03</td>
<td>h, HH, hh</td>
</tr>
<tr>
<td>Minute</td>
<td>m</td>
<td>8,59</td>
<td>m, mm</td>
</tr>
<tr>
<td>Second, Millisecond</td>
<td>s, A</td>
<td>24, 2, 34532</td>
<td>ss, s, AAAAAA</td>
</tr>
</tbody>
</table>

For a more complete list of date symbols, please see the [Formatting Date and Times Overview on the International Components](http://www.unicode.org) page.
Note: Some date formats are not supported by all databases and file types.

4. Check your format against the string displayed in Tableau. If the symbols and formatting is correct, select OK to create the new calculated field. The calculated field will act as a date field in your view.

Hyper extracts

For .hyper extract data sources, the symbols are defined by the Unicode Consortium. For more information, go to Format syntax in DATEPARSE function for extract data sources on page 1079 to see a the subset of field types and symbols that can be used with the DATEPARSE function in .hyper extracts.

Locale considerations

The DATEPARSE function relies on the locale specified by your computer settings to interpret and then display the strings that you want to convert. More specifically, the locale will affect whether a certain format can be recognized. This means that if a format is not supported by the locale, then you might see a null value or no value returned. For example, suppose you have the following string in your data:

12Sep2016:9:8:8.6546

The values returned from the DATEPARSE function for this string are different based on locale. In the English locale you will get a certain value, but for the Japanese locale you will get no value. In this case, no value is returned because the Japanese locale does not recognize “Sep.”

Datetime value

<table>
<thead>
<tr>
<th>English locale</th>
<th>Japanese locale</th>
</tr>
</thead>
<tbody>
<tr>
<td>#9/12/2016 9:08:09 AM#</td>
<td>-</td>
</tr>
</tbody>
</table>

As with all dates, after you have used the DATEPARSE function to convert a string to a datetime type, by default Tableau will display the datetime value in the default format of your locale. If the
locale changes, the result of the DATEPARSE function might display your new datetime value in a different format.

**Note:** Because of a Jet limitation, the DATEPARSE function cannot correctly identify locale information specified by your computer settings in extracts created from Access data sources. To resolve this issue, consider exporting your Access data to Excel.

Create a calculation using the DATE function

If the DATEPARSE function is not available for the data that you're working with, or the field you are trying to convert is a number data type, you can use the DATE function instead.

The DATE function converts a number, string or date expression to a date type. When you create a calculation that uses the DATE function, Tableau creates a new field in your Tableau data source that allows you to interact with your date data as a date. To successfully produce date values from a number, string, or date expression using the DATE function, Tableau needs to be able to interpret the components of the string into date parts. After the components of the date are identified, Tableau uses the computer locale to determine the default format of the date.

For example, suppose the table you're working with contains a column of date data that is called "Original Date." The "Original Date" column is a string type.

<table>
<thead>
<tr>
<th>Original Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>03Jan2017</td>
</tr>
<tr>
<td>05Jan2017</td>
</tr>
<tr>
<td>07Mar2017</td>
</tr>
<tr>
<td>19Mar2017</td>
</tr>
<tr>
<td>30Apr2017</td>
</tr>
</tbody>
</table>

In this case, you can create a calculated field called "New Date" that uses an expression in a DATE function to convert the string values in the "Original Date" field into date values.

For this example, the date expression is comprised of the LEFT function to isolate the day component, the MID function to isolate the month component, and the RIGHT function to isolate the year component.
DATE (LEFT([Original Date], 2) + "/" + MID([Original Date],3,3) + "/" + RIGHT([Original Date],4))

The "New Date" calculation produces the following column:

<table>
<thead>
<tr>
<th>New Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/3/2017</td>
</tr>
<tr>
<td>1/5/2017</td>
</tr>
<tr>
<td>3/7/2017</td>
</tr>
<tr>
<td>3/19/2017</td>
</tr>
<tr>
<td>4/30/2017</td>
</tr>
</tbody>
</table>

In this example, the new date values are based on an English locale and default formatting.

See Also

- Custom Date Formats on page 1071
- Format Dates Using ISO-8601 Weeks and Years on page 1082
- Date Functions on page 1298
- All Functions (by Category): Date
- STR() Function Ignores the Default Date and Number Formatting (Tableau Knowledge Base)

Pivot Data from Columns to Rows

Sometimes, analyzing data that is stored in a crosstab format can be difficult in Tableau. When working with Microsoft Excel, text file, Google Sheets, and .pdf data sources, you can use the pivot your data from crosstab format into columnar format. If you are working with other data sources, you can Pivot using custom SQL (Tableau Desktop) on page 749.

For example, suppose you have the number of devices sold by quarter for three vendors in three separate fields. You can pivot your data so that the vendor is in one field and the number of devices sold is in another field.
Pivot the data

After you have set up the data source, in the grid, select two or more columns. Click the drop-down arrow next to the column name, and then select **Pivot**. New columns called "Pivot field names" and "Pivot field values" are created and added to the data source. The new columns replace the original columns that you selected to create the pivot.

![Pivot the data](image)

Add to the pivot

To add more data to the pivot, select another column, click the drop-down arrow next to the column name, and then select **Add Data to Pivot**. Make sure that the pivot columns and values look as expected before you begin your analysis.
To remove a pivot, click the drop-down arrow next to the name of a pivot column, and then select **Remove Pivot**.

**Troubleshooting pivots**

- **Red fields in the view and fields with exclamation points in the Data pane:** Because the original fields are replaced with new pivot fields, any references to the original fields in the view will no longer work. They cause fields to become red in the view or show a red exclamation point next to the field in the Data pane.

- **Null values in the grid:** If all of the original fields used in the pivot are removed, for example in an extract refresh, null values display in the pivot fields.

- **No pivot option:** Pivot appears when you select two or more columns in a single Microsoft Excel, text file, Google Sheets, and .pdf data source. If using a different data source in Tableau Desktop, you can use custom SQL to pivot.

**Pivot using custom SQL (Tableau Desktop)**

You can also use custom SQL to pivot your data, even if you aren't working Excel, text file, Google Sheets, and .pdf data sources. When you use the UNION ALL operator in a custom SQL query, you can take values from distinct columns and put them into a new column.

For example, suppose you have a table called **Contest**.

**Contest**
To optimize your analysis of this data in Tableau, you can use the following custom SQL query to pivot the "Start Time" and "End Time" columns so that their values are in a single column.

Select [Runner]
   , 'Start' as [Action]
   , [Start Time] as [Time]
From [Contest]
Union ALL
Select [Runner]
   , 'End' as [Action]
   , [End Time] as [Time]
From [Contest]

The above custom SQL query does the following:

- Pivots the **Start Time** column header into a string value called **Start** and adds that value to a new column called **Action**.
- Pivots the **End Time** column header into a string value called **End** and adds that value to a new column called **Action**.
- Pivots the **Start Time** and **End Time** columns so that their values are in a new column called **Time**.

The following table shows the results of this custom SQL query.

<table>
<thead>
<tr>
<th>Runner</th>
<th>Action</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amanda</td>
<td>Start</td>
<td>9/3/2016 3:04 PM</td>
</tr>
<tr>
<td>Runner</td>
<td>Action</td>
<td>Time</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>------------------</td>
</tr>
<tr>
<td>Oscar</td>
<td>Start</td>
<td>9/3/2016 3:04 PM</td>
</tr>
<tr>
<td>William</td>
<td>Start</td>
<td>9/3/2016 3:04 PM</td>
</tr>
<tr>
<td>Amanda</td>
<td>End</td>
<td>9/3/2016 3:25 PM</td>
</tr>
<tr>
<td>Oscar</td>
<td>End</td>
<td>9/3/2016 3:21 PM</td>
</tr>
<tr>
<td>William</td>
<td>End</td>
<td>9/3/2016 3:16 PM</td>
</tr>
</tbody>
</table>

To pivot data using custom SQL

1. Connect to your data.

2. Double-click the New Custom SQL option in the left pane. For more information, see Connect to a Custom SQL Query on page 723.

3. In the Edit Custom SQL dialog box, copy and paste the following custom SQL query and replace the contents with information about your table:

   ```sql
   Select [Static Column], 'New Value (from Column Header 1)' as [New Column Header], [Pivot Column Values 1] as [New Values] From [Table] Union ALL
   Select [Static Column], 'New Value (from Column Header 2)' as [New Column Header], [Pivot Column Values 2] as [New Values] From [Table] Union ALL
   Select [Static Column]
   ```
, 'New Value (from Column Header 3' as [New Column Header]
, [Pivot Column Values 3] as [New Values]

From [Table]

Where the following is true:

- **Static Column**: A comma-delimited list of the columns from Table, both dimensions and measures, which should not be included in the pivot.
- **New Value (from Column Header 1-3)**: New names that you give to the original column headers, which are used as row values in the pivot.
- **Pivot Column Values 1-3**: The columns whose values need to be pivoted into a single column.
- **New Column Header**: The name you give the new column that contains the new row values from New Value (from Column Header 1-3).
- **New Values**: The name give the new column that contains the original values from Pivot Column Values 1-3.
- **Table**: The table that you connected to.

4. Click OK.

**Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter**

When you track data in Excel spreadsheets, you create them with the human interface in mind. To make your spreadsheets easy to read, you might include things like titles, stacked headers, notes, maybe empty rows and columns to add white space, and you probably have multiple tabs of data too.

When you want to analyze this data in Tableau, these aesthetically pleasing attributes make it very difficult for Tableau to interpret your data. That's where Data Interpreter can help.

**Tip**: Though Tableau's Excel add-in is no longer supported, Data Interpreter can help you reshape your data for analysis in Tableau.
In this article

What does Data Interpreter do? below
Turn on Data Interpreter and review results below
When Data Interpreter is not available on page 760

What does Data Interpreter do?

Data Interpreter can give you a head start when cleaning your data. It can detect things like titles, notes, footers, empty cells, and so on and bypass them to identify the actual fields and values in your data set.

It can even detect additional tables and sub-tables so that you can work with a subset of your data independently of the other data.

After Data Interpreter has done its magic, you can check its work to make sure it captured the data that you wanted and identified it correctly. Then, you can make any necessary adjustments.

After you select the data that you want to work with, you might also need to do some additional cleaning steps like pivoting your data, splitting fields, or adding filters to get the data in the shape you want before starting your analysis.

Note: If your data needs more cleaning than what Data Interpreter can help you with, try Tableau Prep.

Turn on Data Interpreter and review results

1. From the Connect pane, connect to an Excel spreadsheet or other connector that supports Data Interpreter such as Text (.csv) files, PDF files or Google sheets.

2. Drag a table to the canvas (if needed), then on the Data Source page, in the left pane, select the Use Data Interpreter check box to see if Data Interpreter can help clean up your data.
Note: When you clean your data with Data Interpreter, Data Interpreter cleans all the data associated with a connection in the data source. Data Interpreter does not change the underlying data.

3. In the Data pane, click the **Review the results** link to review the results of the Data Interpreter.

A copy of your data source opens in Excel on the **Key for the Data Interpreter** tab. Review the key to find out how to read the results.
4. Click each tab to review how Data Interpreter interpreted the data source.

If Data Interpreter found additional tables, also called found tables or sub-tables, they are identified in the <sheet name>_subtables tab by outlining their cell ranges. A separate tab is also included for each sub-table, color coded to identify the header and data rows.

If Data Interpreter does not provide the expected results, clear the **Cleaned with Data Interpreter** check box to use the original data source.

5. To replace the current table with any of the found tables, drag the current table off the canvas and then drag the found table that you want to use to the canvas.
If Data interpreter has misidentified the range of the found table, after you drag the found table to the canvas, click the drop-down arrow on that table, and then select **Edit Found Table** to adjust the corners of the found table (the top-left cell and bottom-right cell of the table).

![Edit Found Table](image)

6. After you have the data that you want to work with, you can apply any additional cleaning operations to your data so that you can analyze it.

**Data Interpreter Example**

In this example we are connecting to an Excel spreadsheet with violent crime data by city and state for the year 2016. This spreadsheet includes multiple tables on one sheet and some extra formatting.

<table>
<thead>
<tr>
<th>City</th>
<th>State</th>
<th>Violent Crimes (2015)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlanta</td>
<td>Georgia</td>
<td>214</td>
</tr>
<tr>
<td>Austin</td>
<td>Texas</td>
<td>26</td>
</tr>
<tr>
<td>Baltimore</td>
<td>Maryland</td>
<td>109</td>
</tr>
<tr>
<td>Boston</td>
<td>Massachusetts</td>
<td>229</td>
</tr>
<tr>
<td>Chicago</td>
<td>Illinois</td>
<td>111</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Ohio</td>
<td>84</td>
</tr>
<tr>
<td>Dallas</td>
<td>Texas</td>
<td>36</td>
</tr>
<tr>
<td>Denver</td>
<td>Colorado</td>
<td>65</td>
</tr>
<tr>
<td>Detroit</td>
<td>Michigan</td>
<td>116</td>
</tr>
<tr>
<td>Houston</td>
<td>Texas</td>
<td>37</td>
</tr>
<tr>
<td>Indianapolis</td>
<td>Indiana</td>
<td>56</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>California</td>
<td>265</td>
</tr>
<tr>
<td>Miami</td>
<td>Florida</td>
<td>147</td>
</tr>
<tr>
<td>New York</td>
<td>New York</td>
<td>194</td>
</tr>
<tr>
<td>Seattle</td>
<td>Washington</td>
<td>89</td>
</tr>
</tbody>
</table>

A. Title
B. Merged header cells
C. Extra white space
D. Sub-tables

The extra formatting in this spreadsheet makes it difficult for Tableau to determine what the field headers and values are.

Instead, it reads the data vertically and assigns each column the default value F1, F2, F3 (Field 1, Field 2, Field 3) and so on. Blank cells are read as null values.

To see if Data Interpreter can help clean this data set, we select **Use Data Interpreter**.

Data Interpreter detected the proper headings for the fields, removed the extra formatting and found several sub-tables. The sub-tables are listed in the **Sheets** section in the Data pane and are named using the original sheet name and the cell ranges for each sub-table.

In this example there are three sub-tables: **Crimes 2016 A4:H84**, **Crimes 2016 K5:L40**, and **Crimes 2016 O5:P56**.

To examine the results of the Data Interpreter more closely, we click the **Review the results** link in the Data pane to view an annotated copy of the spreadsheet.
Here we see a copy of the original data, color coded to identify which data was identified as header data and which data was identified as field values.

The next tab shows us the sub-tables that Data Interpreter found, outlined by the cell ranges.

In this example the first sub-table, **Crimes 2016 A4:H84**, has the main data that we want to work with. To use this table as our data table, we can simply drag the original table off the canvas and then drag the new table to the canvas.
Once we have the data that we want to work with in the canvas, we can do some additional clean up on the data. For example we can:

- Change the field names so that they represent city, state, and month names.
- Pivot the months fields.
- Drag in the third sub-table **Crimes 2016 o5:P56** and join it to our first sub-table on the **State** field to include state populations for our analysis.
- Hide any duplicate fields that were added as a result of the join.

The results might look something like this:
Now we are ready to start analyzing our data in Tableau.

When Data Interpreter is not available

The Data Interpreter option might not be available for the following reasons:

- **The data source is already in a format that Tableau can interpret**: If Tableau Desktop doesn't need extra help from Data Interpreter to handle unique formatting or extraneous information, the Data Interpreter option is not available.

- **Many rows or many columns**: The Data Interpreter option is not be available when your data has the following attributes:
  - Data contains more than 2000 columns.
  - Data contains more than 3000 rows and more than 150 columns.

- **The data source is not supported**: Data Interpreter is only available for Microsoft Excel, Text (.csv) files, PDF files and Google Sheets. For Excel, your data must be in the .xls or .xlsx format.
Split a Field into Multiple Fields

If you have string fields in your data that contain multiple units of information, for example, the first and last name of a customer, it might be easier to analyze the data if you split the values in that field into separate fields. You can use split or custom split options in Tableau to separate the values based on a separator or a repeated pattern of values present in each row of the field. In this example, the common separator is a space character ( ).

The new fields created by the split or custom split are added to the data source as calculated fields.

Splits and custom splits are based on the SPLIT string function. For more information, see String Functions on page 1288.

You can tell if your data supports the SPLIT function by checking the field drop-down menu:

- On the data source page, check the menu for Split and Custom Split.
- On the Data pane in the worksheet, check the menu for Transform > Split and Custom Split.

Because different connection types support different functionality in Tableau, you may need to check both places to determine if your data supports the split and custom split options.
Split fields automatically

A string field can be split automatically based on a common separator that Tableau detects in the field.

Depending on the connection type, a split can automatically separate a field’s values in up to ten new fields. The data type of the new fields generated by the split can vary depending on the pattern combination that Tableau detects in the separator that is used to split the original field.

**Note:** In some cases, if the split takes too long to generate new fields or Tableau cannot find a common separator, a custom split dialog box displays. For more information, see the **Custom split on the next page** section.

To split a field automatically

1. On the Data Source page, in the grid, click the drop-down arrow next to the field name.
2. Select **Split**.

   ![Data Source Grid Example](image)

   **Note:** You can also use the split option from the Data pane in the worksheet. In the Data pane, right-click the field you want to split, and then select **Transform > Split**.

If you do not like the results of the split, you can go to the Data pane and edit the calculated fields that are created by the split. Alternatively, you can click undo in the Tableau Desktop toolbar or remove the split.
To remove fields created by a split

1. On the Data Source page, in the grid, click the drop-down arrow next to the field name.
2. Select **Delete**.
3. Repeat steps 1-2 to remove all the fields created by the split.

Alternatives to automatic split

Sometimes, using automatic split is not the best option. The following are example of cases when you should not split fields automatically:

- **Values contain different number of separators**: Fields cannot be split automatically if the number of separators varies from value to value. For example, suppose a field has the following values:
  
  jsmith| accounting | north
dnguyen | humanresources
lscott | recruiting| west
karnold | recruiting | west

  In cases like this, consider using a custom split. For more information, see the **Custom split** below section.

- **Values contain mixed separators**: Fields cannot be split automatically if the separator types are different. For example, suppose a field contains the following values:
  
  smith.accounting
dnguyen-humanresources
lscott_recruiting
karnold_recruiting

  In cases like this, consider using regular expressions to create new fields. For more information, see **Additional Functions** on page 1373.

Custom split

You can use the custom split option to specify a common separator for the split. Like the split option, a custom split can separate a field’s values in up to ten new fields. In addition, you can choose to split the values at the first n occurrences of the separator, the last n occurrences of
the separator, or at all occurrences of the separator. The data type of the new fields generated by the custom split always results in a string data type.

To use a custom split

1. On the Data Source page, in the grid, click the drop-down arrow next to the field name.
2. Select **Custom Split**.

<table>
<thead>
<tr>
<th>Orders</th>
<th>Abc</th>
<th>Abc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Claire Gute</td>
<td></td>
<td>Rename</td>
</tr>
<tr>
<td>Claire Gute</td>
<td></td>
<td>Copy Values</td>
</tr>
<tr>
<td>Darrin Van Huff</td>
<td></td>
<td>Hide</td>
</tr>
<tr>
<td>Sean O’Donnell</td>
<td></td>
<td>Aliases...</td>
</tr>
<tr>
<td>Sean O’Donnell</td>
<td></td>
<td>Create Calculated Field...</td>
</tr>
<tr>
<td>Brosina Hoffman</td>
<td></td>
<td>Create Group...</td>
</tr>
<tr>
<td>Brosina Hoffman</td>
<td></td>
<td>Split</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Custom Split...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pivot (select multiple fields)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Describe...</td>
</tr>
</tbody>
</table>

**Note:** You can also access the custom split option from the Data pane. In the Data pane, right-click the field you want to split, and then select **Transform > Custom Split**.

If you do not like the results of the split, you can go to the Data pane and edit the calculated fields that are created by the split. Alternatively, you can click undo in the toolbar or remove the split.

To remove fields created by a custom split

1. On the Data Source page, in the grid, click the drop-down arrow next to the filed name.
2. Select **Delete**.
3. Repeat steps 1-2 to remove all the fields created by the custom split.

At a glance: Working with splits and custom splits

The following is a list of additional notes you might need to know about splits and custom splits.
• New fields generated from a split or custom split cannot be used as keys to join tables, but can be used to blend data sources.

• New fields generated from a split or custom split cannot be used in a pivot.

• Split and custom split options are not supported for sets, groups, parameters, and bins.

• Microsoft SQL Server only allows up to four split fields.

• To generate more than ten new fields, consider using a split or custom split on the field that was previously generated by a split or custom split.

Troubleshooting splits and custom splits

The following is a list of issues you might experience when using splits and custom splits:

• **Split and custom split options missing for a supported data source type:** Split and custom split options are available only for fields that are a string data type.

• **Null values or empty cells:** After creating a split or a custom split, new fields might contain null values or no values at all. Null values or empty cells occur when there are no values for all of the expected new fields.

• **Data has been removed:** Tableau might use portions of the field’s values as a separator. If a portion of a field’s values is used as a separator, those values no longer appear in the new fields. For example, suppose a field contains the following values:

  ZIP-98102
  ZIP-98109
  ZIP-98119
  ZIP-98195

  In this case, the split will create a new field with the following values:

  98102
  98109
  98119
  98195
The split will not create a separate field for “ZIP-” because the split uses it as a separator.

**Filter Data from Data Sources**

You can create filters on a data source, thereby reducing the amount of data in the data source.

If you create an extract from a data source that already has data source filters in place, those filters are automatically recommended as extract filters, and will appear in the Extract dialog. Those recommended filters are not required to be part of the Extract filter list, and can safely be removed without affecting the existing set of data source filters.

Data source filters can be useful for restricting the data users can see when you publish a workbook or data source. When you publish a data source to Tableau Server, the data source and any associated files or extracts are transported in entirety to the Server. As you publish a data source you can define access permissions for downloading or modifying the data source, and you can also choose the users and groups who can remotely issue queries through Tableau Server against that data source. When users have query permission and no download permission, you can share a rich data model having calculated fields, aliases, groups, sets and more—but only for querying.

Furthermore, users who query published data source will never be able to see or modify any data source filters present on the originally published data source, but all of the users’ queries will be subject to those data source filters. This is a great way to offer a restricted subset of your data, for example by filtering dimensions for specific users and groups, or by defining data source filters based on a fixed or relative date range. This is often useful for data security, and it also allows you to manage performance of the remote database which Tableau Server will ultimately query on a user’s behalf. For systems that rely heavily on partitions or indexing, data source filters may yield tremendous control over the performance of queries issued by Tableau.

**Create a data source filter**

The primary way to create a data source filter is from the data source page.

**To create a data source filter**

1. On the data source page, click **Add** in the Filters section in the upper-right corner of the page.
To create a data source filter on a worksheet, right-click (control-click on a Mac) the data source and choose **Edit Data Source Filters**.

Whether you start from the Data Source page or from a worksheet, you see an Edit Data Source Filter dialog box, listing any existing data source filters.

2. Click **Add** to open an Add Filter dialog box listing all fields in the data source.
3. Click to select a field to filter; then specify how the field should be filtered, just as you would for a field on the Filters shelf.

To add an additional data source filter, repeat this procedure.

**Global filters and data source filters**

When you create a data source filter, any global filters that use that data source are displayed automatically in the **Edit Data Source Filters** dialog box to make it easy for you to promote a global filter to be a data source filter. To promote the global filter to be a data source filter, click **OK**.

If you promote a global filter to be a data source filter, that global filter will no longer be visible in worksheets of the workbook (because it becomes a data source filter).
**Important:** Be aware that you do not need to select a global filter in the **Edit Data Source Filters** dialog box to promote it. When you click **OK**, all global filters in the list will be promoted.

To prevent a global filter from being promoted to a data source filter, select the global filter in the **Edit Data Source Filters** dialog box, and then click **Remove**.

---

**Understand Field Type Detection and Naming Improvements**

Tableau uses a collection of commonly used patterns to detect and transform your data to make it easier for you to interact with. For example, when you work with fields in the Data pane, Tableau takes the data from your data source, detects its type, and divides it into dimensions and measures.

Tableau can also detect fields that should be treated as dimensions and clean up field names by making them more readable. You see these improvements only when the field names, which come from column headers in your data source, meet the patterns or conditions described below.

**Note:** Tableau never changes your underlying data.

If you prefer to maintain the field types and field names from the underlying data, see the **Undo changes made by Tableau** section at the bottom of this article.
Fields that are treated as a dimension

Field names that contain certain keywords are treated as dimensions, even if the values for those fields are numeric.

Keywords Code, Key, and ID

Field names that contain the following keywords and meet the conditions listed are treated as dimensions instead of measures.

Conditions:

- Contains keywords Code, ID, or Key.
- Keywords are either separated from other text in the field name by non-letter characters, all capitalized, or the first letter is capitalized in a field name that otherwise has mixed casing.
- Keywords are at the beginning or end of the field name, with leading or trailing non-letter characters. In traditional Chinese, Japanese, and Korean, the key word must be located at the end of the field name.

<table>
<thead>
<tr>
<th>Language</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>Code, Id, Key</td>
</tr>
<tr>
<td>Chinese (Traditional)</td>
<td>Id</td>
</tr>
<tr>
<td>French</td>
<td>Cle, Clé, Identificação, Identificador</td>
</tr>
<tr>
<td>German</td>
<td>Chiffre, Kennung, Kennnummer, Id, Identität, Schlüssel</td>
</tr>
<tr>
<td>Japanese</td>
<td>Id</td>
</tr>
<tr>
<td>Korean</td>
<td>Id</td>
</tr>
<tr>
<td>Portuguese</td>
<td>Chave, ID</td>
</tr>
<tr>
<td>Spanish</td>
<td>Clave, Id, Identificación</td>
</tr>
</tbody>
</table>

Keywords Number, Num, and Nbr

Field names that end with Number, Num, or Nbr are treated as a dimension instead of a measure. For example, “Record Number” is treated as a dimension but “Number of Records” is not. In addition, the Korean field name must be four or fewer characters long.
<table>
<thead>
<tr>
<th>Language</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>Number, Num, Nbr</td>
</tr>
<tr>
<td>Chinese (Traditional)</td>
<td>-</td>
</tr>
<tr>
<td>French</td>
<td>Nombre, Num, Nb</td>
</tr>
<tr>
<td>German</td>
<td>Nr, Num, Nummer, Zahl</td>
</tr>
<tr>
<td>Japanese</td>
<td>番号</td>
</tr>
<tr>
<td>Korean</td>
<td>번호</td>
</tr>
<tr>
<td>Portuguese</td>
<td>Número, Num, Nº</td>
</tr>
<tr>
<td>Spanish</td>
<td>Núm, N.º, Nro, Número</td>
</tr>
</tbody>
</table>

Keywords related to dates

Field names containing keywords that are recognized as date parts are treated as dimensions. These field names can only contain only one additional word unrelated to dates to qualify. For example, “Fiscal Year” is treated as a dimension but “Fiscal Year Information” is not. In addition, traditional Chinese field names must be four or fewer characters long and cannot contain digits. Japanese and Korean field names must be four or fewer characters long.

<table>
<thead>
<tr>
<th>Language</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>Year, Yr, Day, Day of Week, Week, Wk, Month, Quarter, Fy</td>
</tr>
<tr>
<td>Chinese (Traditional)</td>
<td>年, 月</td>
</tr>
<tr>
<td>French</td>
<td>An, AF, Année, Année, Jour de la semaine, Jour, Mois, Semaine, Trimestre</td>
</tr>
<tr>
<td>German</td>
<td>BJ, FJ, EJ, GJ, Jahr, Jr, Quart, Quartal, Monat, Tag de Woche, Tag, Viertel, Woche, Wo</td>
</tr>
<tr>
<td>Japanese</td>
<td>年度</td>
</tr>
<tr>
<td>Korean</td>
<td>년도</td>
</tr>
</tbody>
</table>
Field name clean-up

Field names that contain specific characters or capitalized in a certain way are renamed. Field values that include square brackets are automatically converted to parentheses.

Field names with underscore and space characters

Non-leading and trailing underscore (_) characters in field names are converted into space ( ) characters. Carriage return or line feed characters in a field name are removed. However, field names that contain leading underscore characters remain unchanged. In addition, leading and trailing spaces in field names are removed. This type of field name clean up applies only to English, French, German, Portuguese, and Spanish languages.

Example 1: The field name “Country_Name” is converted to “Country Name”.
Example 2: The field name “_Days_On_Market” is converted to “_Days On Market”.

Field names with multiple underscore characters in a row remain unchanged. For example, “Country___Name” remains as “Country___Name”.

Field names and capitalization

Field names that use all capital letters with non-letter characters are converted to all lower-case letters except for the characters immediately after the non-letter character.

Additionally, field names that use all lower-case letters are converted so the first letter in the field name is capitalized.

This type of field name clean up applies only to English, French, German, Portuguese, and Spanish languages.

Example 1: The field name “PC1” is converted to “Pc1.” The field name “COUNTRY_NAME” is converted to “Country Name”. However, “Budget COGS” remains as “Budget COGS”.
Example 2: The field name “FDA” does not change. However, the field name
“FDA_Sales” is converted to “Fda Sales”.

**Example 3:** The field name “age” is converted to “Age”.

Space characters are added to the case boundaries of field names that contain mixed casing with non-letter characters.

**Example:** The field name “ThisCase” is converted to “This Case”.

### Field names that are two or three letters long

Field names that are two or three letters long that are part of a multiple word phrase are converted so that each letter in the word is capitalized, unless the two or three letter word contains a vowel (i.e., a, e, i, o, or u).

**Example:** The field name "Unit Qty" is converted to "Unit QTY". However, the field name "Sales Amt" remains as "Sales Amt".

Several three-letter acronyms are converted to all uppercase. Examples of three-letter acronyms include the following:

<table>
<thead>
<tr>
<th>CIF</th>
<th>FDA</th>
<th>MPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMA</td>
<td>FOB</td>
<td>MSA</td>
</tr>
<tr>
<td>DOB</td>
<td>FTE</td>
<td>SKU</td>
</tr>
<tr>
<td>EIN</td>
<td>KPI</td>
<td>UPC</td>
</tr>
<tr>
<td>ESP</td>
<td>LOB</td>
<td>URL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USD</td>
</tr>
</tbody>
</table>

The following field names are converted to all lowercase letters unless the name occurs at the beginning of the field name:

<table>
<thead>
<tr>
<th>1st</th>
<th>as</th>
<th>down</th>
<th>of</th>
<th>th (suffix following a number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2nd</td>
<td>at</td>
<td>for</td>
<td>on</td>
<td>than</td>
</tr>
<tr>
<td>3rd</td>
<td>but</td>
<td>in</td>
<td>off</td>
<td>the</td>
</tr>
</tbody>
</table>
Undo changes made by Tableau

If you prefer to maintain the field types and field names specified by the underlying data, you can undo changes made by Tableau either manually, at the data source level, or automatically, at the application level.

To undo changes manually

1. Open your workbook and go to the Data Source page.
2. Click the metadata grid button.
3. Press Ctrl+Click (Windows) or Command+Click (Mac) to select the rows with changes you want to undo.
4. Click the drop-down arrow for one of the rows selected, and select Reset name.

To automatically undo changes made by Tableau at the application level, follow the procedure in the Overriding Automatic Field Renaming article. This method affects all data sources that you access from Tableau Desktop.

Extract Your Data

In this article

- What’s new to extracts on the next page
- Changes to values and marks in the view on page 775
- Extract Your Data above
- About the Compute Calculations Now option for extracts on page 781
Extracts are saved subsets of data that you can use to improve performance or to take advantage of Tableau functionality not available or supported in your original data. When you create an extract of your data, you can reduce the total amount of data by using filters and configuring other limits. After you create an extract, you can refresh it with data from the original data. When refreshing the data, you have the option to either do a full refresh, which replaces all of the contents in the extract, or you can do an incremental refresh, which only adds rows that are new since the previous refresh.

Extracts are advantageous for several reasons:

- **Supports large data sets**: You can create extracts that contain billions of rows of data.
- **Fast to create**: If you’re working with large data sets, creating and working with extracts can be faster than working with the original data.
- **Help improve performance**: When you interact with views that use extract data sources, you generally experience better performance than when interacting with views based on connections to the original data.
- **Support additional functionality**: Extracts allow you to take advantage of Tableau functionality that's not available or supported by the original data, such as the ability to compute Count Distinct.
- **Provide offline access to your data**: Extracts allow you to save and work with the data locally when the original data is not available. For example, when you are traveling.

**What's new to extracts**

Beginning in version 10.5, when you create a new extract it uses the .hyper format. Extracts in the .hyper format take advantage of the improved data engine, which supports faster analytical and query performance for larger data sets.

Similarly, when an extract-related task is performed on a .tde extract using version Tableau Desktop 2018.2, the extract is upgraded to a .hyper extract. After a .tde extract is upgraded to a .hyper extract, it can’t be reverted back to .tde extract. For more information, see Extract Upgrade to .hyper Format on page 788.
Changes to values and marks in the view

To improve extract efficiency and scalability, values in extracts can be computed differently in version 2018.2 compared to versions 10.4 and earlier. Changes to how the values are computed can affect the way marks in your view are populated. In some rare cases, the changes can cause your view to change shape or become blank. These changes can also apply to multi-connection data sources, data sources that use live connections to file-based data, data sources that connect to Google Sheets data, cloud-based data sources, extract-only data sources, and WDC data sources.

To get an idea of some of the differences you might see in your view using version 2018.2, see the sections below.

**Format of date and date time values**

In version 2018.2, extracts are subject to more consistent and stricter rules around how date strings are interpreted through the DATE, DATETIME, and DATEPARSE functions. This affects how dates are parsed, or the date formats and patterns that are allowed for these functions. More specifically, the rules introduced in 2018.2 can be generalized as the following:

1. Dates are evaluated and then parsed by column, not by row.
2. Dates are evaluated and then parsed based on the locale of where the workbook was created, not on locale of the computer where the workbook is opened.

These new rules allow extracts to be more efficient and to produce results that are consistent with commercial databases.

However, because of these rules, particularly in international scenarios where the workbook is created in a locale different from the locale that the workbook is opened in or the server that the workbook is published to, you might see that 1.) date and datetime values change to different date and datetime values or 2.) date and datetime values change to Null. When your date and datetime values change to different date and datetime values or become Null, it's often an indication that there are issues with the underlying data.

Here are some common reasons why you might see changes to your date and datetime values in your extract data source using version 10.5 and later.

<table>
<thead>
<tr>
<th>Common causes of changes to date/datetime values</th>
<th>Common causes of null values</th>
</tr>
</thead>
<tbody>
<tr>
<td>• When a function has to parse multiple date formats in a single</td>
<td>• When a function has to parse multiple date</td>
</tr>
<tr>
<td>Common causes of changes to date/datetime values</td>
<td>Common causes of null values</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Column. Where the date is ambiguous and can be interpreted in several different ways, the date will be interpreted based on the format Tableau has determined for that column. For some examples, see <strong>Date scenario 1</strong> on the next page and <strong>Date scenario 2</strong> on page 778 below.</td>
<td></td>
</tr>
<tr>
<td>When a function has to parse a YYYY-MM-DD (ISO) format. Values that exceed what is allowed for &quot;YYYY,&quot; &quot;MM,&quot; or &quot;DD&quot; cause null values. For an example, see <strong>Date scenario 3</strong> on page 779.</td>
<td></td>
</tr>
<tr>
<td>When a function doesn't have enough information to derive the time, it can interpret a value as &quot;00:00:00.0&quot;, using &quot;0&quot; for hour, minute, second, and millisecond.</td>
<td></td>
</tr>
<tr>
<td>When a function has to parse a YYYY-MM-DD (ISO) format. For an example, see <strong>Date scenario 1</strong> on the next page and <strong>Date scenario 2</strong> on page 778 below.</td>
<td></td>
</tr>
<tr>
<td>When a function doesn't have enough information to derive the day, it can interpret a value as &quot;1&quot; or &quot;January&quot; for month.</td>
<td></td>
</tr>
<tr>
<td>When a function parses years, it is interpreted as the following:</td>
<td></td>
</tr>
<tr>
<td>Year &quot;07&quot; is interpreted as &quot;2007&quot;</td>
<td></td>
</tr>
<tr>
<td>Year &quot;17&quot; is interpreted as &quot;2017.&quot;</td>
<td></td>
</tr>
<tr>
<td>Year &quot;30&quot; is interpreted as &quot;2030.&quot;</td>
<td></td>
</tr>
<tr>
<td>Year &quot;69&quot; interpreted as &quot;2069.&quot;</td>
<td></td>
</tr>
<tr>
<td>Year &quot;00&quot; interpreted as &quot;2000.&quot;</td>
<td></td>
</tr>
<tr>
<td>Year &quot;19&quot; is interpreted as &quot;2019.&quot;</td>
<td></td>
</tr>
<tr>
<td>Year &quot;99&quot; is interpreted as &quot;2099.&quot;</td>
<td></td>
</tr>
<tr>
<td>Year &quot;07&quot; interpreted as &quot;2007.&quot;</td>
<td></td>
</tr>
<tr>
<td>Year &quot;17&quot; interpreted as &quot;2017.&quot;</td>
<td></td>
</tr>
<tr>
<td>Year &quot;30&quot; interpreted as &quot;2030.&quot;</td>
<td></td>
</tr>
<tr>
<td>Year &quot;69&quot; interpreted as &quot;2069.&quot;</td>
<td></td>
</tr>
<tr>
<td>When a function has to parse date values that contain trailing characters. For example, time zone and daylight savings suffixes and keywords, such as &quot;midnight&quot; cause null values.</td>
<td></td>
</tr>
<tr>
<td>When a function has to parse an invalid date or time. For example, 32/3/2012 causes a null value. In another example, 25:01:61 causes null values.</td>
<td></td>
</tr>
</tbody>
</table>
### Common causes of changes to date/datetime values

- Year "70" is interpreted as "1970."

### Common causes of null values

- a null value.
- When a function has to parse contradicting inputs. For example, suppose the pattern is ‘dd.MM (MMMM) y’ and the input string is ‘1.09 (August) 2017’, where both "9" and "August" are months. The result is a null value because the month values are not the same.
- When a function has to parse contradicting patterns. For example, a pattern that specifies a mix of Gregorian year (y) and ISO week (ww) causes null values.

### Date scenario 1

Suppose you have a workbook created in an English locale that uses .tde extract data source. The table below shows a column of string data contained in the extract data source.

<table>
<thead>
<tr>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/31/2018</td>
</tr>
<tr>
<td>31/10/2018</td>
</tr>
<tr>
<td>12/10/2018</td>
</tr>
</tbody>
</table>

Based on the particular English locale, the format of the date column was determined to follow the MDY (month, day, and year) format. The following tables show what Tableau displays based on this locale when the DATE function is used to convert string values into date values.

October 31,
If the extract is opened in a German locale, you see the following:

<table>
<thead>
<tr>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>31 October 2018</td>
</tr>
<tr>
<td>31 October 2018</td>
</tr>
<tr>
<td>12 October 2018</td>
</tr>
</tbody>
</table>

However, after the extract is opened in a German locale using version 2018.2, the DMY (day, month, and year) format of the German locale is strictly enforced and causes a `Null` value because one of the values doesn't follow DMY format.

<table>
<thead>
<tr>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null</td>
</tr>
<tr>
<td>October 31, 2018</td>
</tr>
<tr>
<td>October 12, 2018</td>
</tr>
</tbody>
</table>

Date scenario 2

Suppose you have another workbook created in an English locale that uses a .tde extract data source. The table below shows a column of numeric date data contained in the extract data source.

<table>
<thead>
<tr>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1112018</td>
</tr>
<tr>
<td>1212018</td>
</tr>
<tr>
<td>1312018</td>
</tr>
</tbody>
</table>
Based on the particular English locale, the format of the date column was determined to follow the MDY (month, day, and year) format. The following tables show what Tableau displays based on this locale when the DATE function is used to convert the numeric values into date values.

<table>
<thead>
<tr>
<th>Date</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/1/2018</td>
<td>11/1/2018</td>
</tr>
<tr>
<td>12/1/2018</td>
<td>12/1/2018</td>
</tr>
<tr>
<td>Null</td>
<td>Null</td>
</tr>
</tbody>
</table>

Date scenario 3

Suppose you have a workbook that uses a .tde extract data source. The table below shows a column of string data contained in the extract data source.

<table>
<thead>
<tr>
<th>Date</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-10-31</td>
<td>2018-10-31</td>
</tr>
<tr>
<td>2018-31-10</td>
<td>2018-10-12</td>
</tr>
<tr>
<td>2018-12-10</td>
<td>2018-12-10</td>
</tr>
</tbody>
</table>

Because the date uses the ISO format, the date column always follows the YYYY-MM-DD format. The following tables show what Tableau displays when the DATE function is used to convert string values into date values.

<table>
<thead>
<tr>
<th>Date</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 10, 2018</td>
<td>October 10, 2018</td>
</tr>
<tr>
<td>Null</td>
<td>Null</td>
</tr>
<tr>
<td>December 10, 2018</td>
<td>December 10, 2018</td>
</tr>
<tr>
<td>October 12, 2018</td>
<td>October 12, 2018</td>
</tr>
</tbody>
</table>
**Note:** In versions 10.4 (and earlier), ISO format and other date formats could have produced differing results depending on the locale of where the workbook was created. In an English locale for example, both 2018-12-10 and 2018/12/10 could produce December 12, 2018. However, in a German locale 2018-12-10 could produce December 12, 2018 and 2018/12/10 could produce October 12, 2018.

**Sort order and case sensitivity**

In version 2018.2, extracts have collation support and therefore can more appropriately sort string values that have accents or are cased differently.

For example, suppose you have a table of string values. In terms of sort order, this means that a string value like Égypte is now appropriately listed after Estonie and before Fidji.

*About Excel data:*

With regard to casing, this means that how Tableau stores values have changed between version 10.4 (and earlier) and version 10.5 (and later). However, the rules for sorting and comparing values haven't. In version 10.4 (and earlier), string values like "House," "HOUSE," and  "houSe" are treated the same and stored with one representative value. In version 10.5 (and later), the same string values are considered unique and therefore stored as individual values. For more information, see [Changes to the way values are computed](#) on page 324.

**Breaking ties in top N queries**

When a top N query in your extract produces duplicate values for a specific position in a rank, the position that breaks the tie can be different when using version 2018.2. For example, suppose you create a top 3 filter. Positions 3, 4, and 5 have the same values. When using version 10.4 and earlier, the top filter can return 1, 2, and 3 positions. However, when using version 2018.2, the top filter can return 1, 2, and 5 positions.

**Precision of floating-point values**

In version 2018.2, extracts are better at taking advantage of the available hardware resources on a computer and therefore able to perform mathematical operations in a highly parallel way. Because of this, real numbers can be aggregated by .hyper extracts in different order. When numbers are aggregated in different order, you might see different values in your view after the decimal point each time the aggregation is computed. This is because floating-point addition and multiplication is not necessarily associative. That is, \((a + b) + c\) is not necessarily the same as \(a + (b + c)\). Also, real numbers can be aggregated in different order because floating-point multiplication is not necessarily distributive. That is, \((a \times b) \times c\) is not necessarily the same as \(a \times

- 780 -
This type of floating-point rounding behavior in .hyper extracts resemble that of floating-point rounding behavior in commercial databases.

For example, suppose your workbook contains a slider filter on an aggregated field comprised of floating point values. Because the precision of floating-point values have changed, the filter might now exclude a mark that defines the upper or lower bound of the filter range. The absence of these numbers could cause a blank view. To resolve this issue, move the slider on the filter or remove and add the filter again.

**Accuracy of aggregations**

In version 2018.2, extracts optimize for large data sets by taking better advantage of the available hardware resources on a computer and therefore able to compute aggregations in a highly parallel way. Because of this, aggregations performed by .hyper extracts can resemble the results from commercial databases more than the results from software that specializes in statistical computations. If you’re working with a small data set or need a higher level of accuracy, consider performing aggregations through reference lines, summary card statistics, or table calculation functions like variance, standard deviation, correlation, or covariance.

**About the Compute Calculations Now option for extracts**

If the **Compute Calculations Now** option was used in a .tde extract using an earlier version of Tableau Desktop, certain calculated fields were materialized and therefore computed in advance and stored in the extract. If you upgrade the extract from a .tde extract to a .hyper extract, the previously materialized calculations in your extract are not included. You must use the **Compute Calculations Now** option again to ensure that materialized calculations are a part of the extract after the extract upgrade. For more information, see **Materialize Calculations in Your Extracts** on page 803.

**New Extract API**

You can use the Extract API 2.0 to create .hyper extracts. For tasks that you previously performed using the Tableau SDK, such as publishing extracts, you can use the Tableau Server REST API or the Tableau Server Client (Python) library. For refresh tasks, you can use the Tableau Server REST API as well. For more information, see **Tableau Extract API** on page 817.

**Create an extract**

Though there are number of places in your Tableau work flow where you can create an extract, the primary method is described below.
1. After you connect to your data and set up the data source on the Data Source page, in the upper-right corner, select **Extract**, and then click the **Edit** link to open the Extract Data dialog box.

2. (Optional) Do one or more of the following to define filters and limit the amount of data in your extract:
   - Click **Add** to define one or more filters to limit how much data gets extracted based on fields and their values.
Select **Aggregate data for visible dimensions** to aggregate the measures using their default aggregation. Aggregating the data consolidates rows, can minimize the size of the extract file, and increase performance.

When you choose to aggregate the data, you can also select **Roll up dates** to a specified date level such as Year, Month, etc. The examples below show how the data will be extracted for each aggregation option you can choose.
### Original data

<table>
<thead>
<tr>
<th></th>
<th>Date</th>
<th>Region</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/1/2009</td>
<td>South</td>
<td>$500</td>
</tr>
<tr>
<td>2</td>
<td>1/1/2009</td>
<td>West</td>
<td>$200</td>
</tr>
<tr>
<td>3</td>
<td>1/1/2009</td>
<td>West</td>
<td>$100</td>
</tr>
<tr>
<td>4</td>
<td>1/1/2009</td>
<td>East</td>
<td>$300</td>
</tr>
<tr>
<td>5</td>
<td>1/1/2009</td>
<td>South</td>
<td>$600</td>
</tr>
<tr>
<td>6</td>
<td>1/2/2009</td>
<td>South</td>
<td>$400</td>
</tr>
<tr>
<td>7</td>
<td>1/2/2009</td>
<td>East</td>
<td>$100</td>
</tr>
</tbody>
</table>

Each record is shown as a separate row. There are seven rows in your data.

### Aggregate data for visible dimensions (no roll up)

<table>
<thead>
<tr>
<th></th>
<th>Date</th>
<th>Region</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/1/2009</td>
<td>East</td>
<td>$300</td>
</tr>
<tr>
<td>2</td>
<td>1/1/2009</td>
<td>South</td>
<td>$500</td>
</tr>
<tr>
<td>3</td>
<td>1/1/2009</td>
<td>West</td>
<td>$300</td>
</tr>
<tr>
<td>4</td>
<td>1/2/2009</td>
<td>East</td>
<td>$100</td>
</tr>
<tr>
<td>5</td>
<td>1/2/2009</td>
<td>South</td>
<td>$1,000</td>
</tr>
</tbody>
</table>

Records with the same date and region have been aggregated into a single row. There are five rows in the extract.

### Aggregate data for visible dimensions (roll up dates to Month)

<table>
<thead>
<tr>
<th></th>
<th>Date</th>
<th>Region</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/1/2009</td>
<td>East</td>
<td>$400</td>
</tr>
<tr>
<td>2</td>
<td>1/1/2009</td>
<td>South</td>
<td>$1,500</td>
</tr>
<tr>
<td>3</td>
<td>1/1/2009</td>
<td>West</td>
<td>$300</td>
</tr>
</tbody>
</table>

Dates have been rolled up to the Month level and records with the same region have been aggregated into a single row. There are three rows in the extract.

- Select the number of rows you want to extract.

You can extract **All rows** or the **Top N** rows. Tableau first applies any filters and aggregation and then extracts the number of rows from the filtered and
aggregated results. The number of rows options depend on the type of data source you are extracting from.

Notes:

- Not all data sources support sampling. Therefore, you might not see the Sampling option in the Extract Data dialog box.
- Any fields that you hide first in the Data Source page or on the sheet tab will be excluded from the extract. Click the Hide All Unused Fields button to remove these hidden fields from the extract.

3. When finished, click OK.

4. Click the sheet tab. Clicking the sheet tab initiates the creating of the extract.

5. In the subsequent dialog box, select a location to save the extract, give the extract file a name, and then click Save.

   If the Save dialog box does not display, see the Troubleshoot extracts section, below.

**Tips for working with extracts**

**Save your workbook to preserve the connection to the extract**

After you create an extract, the workbook begins to use the extract version of your data. However, the connection to the extract version of your data is not preserved until you save the workbook. This means if you close the workbook without saving the workbook first, the workbook will connect to the original data source the next time you open it.

**Toggle between sampled data and entire extract**

When you’re working with a large extract, you might want to create an extract with a sample of the data so you can set up the view while avoiding long queries every time you place a field on a shelf on the sheet tab. You can then toggle between using the extract (with sample data) and using the entire data source by selecting a data source on the Data menu and then selecting Use Extract.

**Remove the extract from the workbook**

You can remove an extract at anytime by selecting the extract data source on the Data menu and then selecting Extract > Remove. When you remove an extract, you can choose to Remove the extract from the workbook only or Remove and delete the extract file. The latter option will delete the extract from your hard drive.
See extract history

You can see when the extract was last updated and other details by selecting a data source on the Data menu and then selecting Extract > History.

If you open a workbook that is saved with an extract and Tableau cannot locate the extract, select one of the following options in the Extract Not Found dialog box when prompted:

- **Locate the extract**: Select this option if the extract exists but not in the location where Tableau originally saved it. Click OK to open an Open File dialog box where you can specify the new location for the extract file.

- **Remove the extract**: Select this option if you have no further need for the extract. This is equivalent to closing the data source. All open worksheets that reference the data source are deleted.

- **Deactivate the extract**: Use the original data source from which the extract was created, instead of the extract.
• **Regenerate the extract**: Recreates the extract. All filters and other customizations you specified when you originally created the extract are automatically applied.

Troubleshoot extracts

• **Creating an extract takes a long time**: Depending on the size of your data set, creating an extract can take a long time. However, after you have extracted the data and saved it to your computer, performance will improve.

• **Extract is not created**: If your data set contains a really large number of columns (e.g., in the thousands), in some cases Tableau might not be able to create the extract. If you encounter problems, consider extracting fewer columns or restructuring the underlying data.

• **Save dialog does not display or extract is not created from a .twbx**: If you follow the above procedure to extract data from a packaged workbook, the Save dialog does not display. When an extract is created from a packaged workbook (.twbx), the extract file is automatically stored in the package of files associated with the packaged workbook. To access the extract file that you created from the packaged workbook, you must unpack the workbook. For more information, see Packaged Workbooks on page 2465.
Extract Upgrade to .hyper Format

In this article

- **New .hyper extract format** below
- What causes an extract upgrade? on the next page
  - Tasks that cause an extract upgrade on the next page
- Why keep an extract in .tde format? on page 796
- What to expect after an extract upgrade on page 796
  - Changes to opening extracts on page 796
  - Changes to sharing extracts on page 797
  - Potential differences in extract file size on page 797
  - Other issues on page 797
- Changes to your view after an extract upgrade on page 798
- Manually upgrade your extract on page 798

**New .hyper extract format**

Beginning in version 10.5, when you create a new extract, it uses the .hyper format instead of the .tde format. Extracts in the .hyper format take advantage of the improved data engine, which supports the same fast analytical and query performance as the data engine before it, but for even larger extracts.

Although there are many benefits of using .hyper extracts, the primary benefits include the following:

- **Create larger extracts**: You can create extracts with billions of rows of data. Because .hyper extracts can support more data, you can consolidate .tde extracts that you previously had to create separately into a single .hyper extract.

- **Create and refresh extracts faster**: While Tableau has always optimized performance for creating and refreshing extracts, version 2018.2 supports faster extract creation and refreshes for even larger data sets.

- **Experience better performance when interacting with views that use extract data sources**: Although smaller extracts continue to perform efficiently, larger extracts perform more efficiently.
What causes an extract upgrade?

Although you can continue to open and interact with .tde extracts in version 2018.2, when you perform an extract task on a .tde extract, the .tde extract is upgraded to a .hyper extract. After the extract is upgraded, you should be aware of the following backward limitations:

- You can't convert the upgraded extract back to a .tde extract.
- You can't open the upgraded extract in an earlier version of Tableau Desktop. This might affect how you and other users might interact with .tde extracts.
- You can't use the Export as Version in Tableau Desktop to downgrade a workbook that contains a .hyper extract.
- You can't publish the extract as an older version from Tableau Desktop to Tableau Server.
- You can't open a workbook in Tableau Desktop 10.4 and earlier that is downloaded from Tableau Server or Tableau Online using the Download Tableau Workbook option if the workbook contains a .hyper extract.

Tasks that cause an extract upgrade

There are three distinct ways a .tde extract can get upgraded to a .hyper extract: 1.) during an extract refresh (full or incremental), 2.) when appending data to an extract, and 3.) when an extract is upgraded manually. After an extract has been upgraded using Tableau Desktop, it's not automatically removed in case the .tde extract is being referenced by other workbooks.

In addition to the tasks that upgrade an extract, extract tasks performed outside of Tableau Desktop can cause an extract upgrade to the .tde extract you interact with. This includes the following tasks:

- Manual refresh (Tableau Server and Tableau Online)
- Incremental refresh on a published extract data source from Tableau Bridge (Tableau Online)
- A scheduled full or incremental extract refresh data source (Tableau Server and Tableau Online)
- A scheduled incremental extract refresh from Tableau Bridge (Tableau Online)
Impact of extract upgrade

When you're working in an environment where the version number of Tableau Desktop doesn't match the Tableau Desktop version that others are using, or doesn't match the version of Tableau Server or Tableau Online, you might encounter some extract compatibility issues depending on the task you want to perform.

Review a detailed explanation of the extract compatibility scenarios on the Tableau Support page or a summary explanation below to better understand when extract upgrades can occur and potential compatibility issues you or others might experience.

**Note:** In the tables below, "10.4" represents Tableau 10.4 and earlier and "10.5" represents Tableau 10.5 and later.

**Tasks using Tableau Desktop**

You should be aware of the following extract-related compatibility scenarios around tasks performed on different versions of Tableau Desktop.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>10.4 workbook</th>
<th>10.5 workbook</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.tde extract</td>
<td>.tde extract</td>
</tr>
<tr>
<td>Create new in 10.4</td>
<td>√</td>
<td>Not possible</td>
</tr>
<tr>
<td>in 10.5</td>
<td>Not possible</td>
<td>Not possible</td>
</tr>
<tr>
<td>Open in 10.4</td>
<td>√</td>
<td>Can't open workbook; you're asked to open the workbook in 10.5 or later</td>
</tr>
<tr>
<td>in 10.5</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Tasks</td>
<td>10.4 workbook</td>
<td>10.5 workbook</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td>.tde extract</td>
<td>.tde extract</td>
</tr>
<tr>
<td>Refresh or append in 10.4</td>
<td>√</td>
<td>Not possible</td>
</tr>
<tr>
<td>Workbook version remains unchanged, but extract upgrades to .hyper format in 10.5</td>
<td>Workbook version remains unchanged, but extract upgrades to .hyper format</td>
<td>√</td>
</tr>
<tr>
<td>Publish workbook from 10.5 to Tableau Server 10.4</td>
<td>√</td>
<td>Can't publish; in the Publish dialog, you see a &quot;server is running an older version than your copy of Tableau Desktop&quot; message, then you see a &quot;workbook cannot be downgraded&quot; error message</td>
</tr>
<tr>
<td>Publish workbook from 10.5 to Tableau Server 10.5 or Tableau</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Tasks</td>
<td>10.4 workbook</td>
<td>10.5 workbook</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td>.tde extract</td>
<td>.tde extract</td>
</tr>
<tr>
<td>Online</td>
<td></td>
<td></td>
</tr>
<tr>
<td>from 10.4 to Tableau Server</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>10.5 or Tableau Online</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publish extract data source</td>
<td></td>
<td></td>
</tr>
<tr>
<td>from 10.5 to Tableau Server</td>
<td>✓</td>
<td>Can't publish; you see a &quot;cannot publish, make sure you are connected to a compatible version&quot; error message</td>
</tr>
<tr>
<td>10.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>from 10.5 to Tableau Server</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>10.5 or Tableau Online</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connect to a published</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>from 10.4 to Tableau</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tasks</td>
<td>10.4 workbook</td>
<td>10.5 workbook</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td>.tde extract</td>
<td>.tde extract</td>
</tr>
<tr>
<td>data source</td>
<td>u Server 10.5 or Tableau Online</td>
<td></td>
</tr>
<tr>
<td>from 10.5 to Tableau Server 10.5 or Tableau Online</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Tasks on Tableau Server 10.5 or Tableau Online**

When working with extracts created in an earlier version of Tableau Desktop, you should be aware of the following extract-related compatibility scenarios around tasks performed on Tableau Server 2018.2 or Tableau Online.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>10.4 workbook</th>
<th>10.5 workbook</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.tde extract</td>
<td>.tde extract</td>
</tr>
<tr>
<td>Edit/save in web authoring on Tableau Server 10.4</td>
<td>✓</td>
<td>Not possible</td>
</tr>
<tr>
<td>Scenario</td>
<td>10.4 workbook</td>
<td>10.5 workbook</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td></td>
<td>.tde extract</td>
<td>.tde extract</td>
</tr>
<tr>
<td>on Tableau Server 10.5 or Tableau Online</td>
<td>Workbook changes to version 10.5, extract remains in .tde format</td>
<td>✓</td>
</tr>
<tr>
<td>Edit/save in web authoring then refresh or append on Tableau Server 10.4</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>on Tableau Server 10.5 or Tableau Online</td>
<td>Workbook changes to version 10.5, and extract upgrades to .hyper format</td>
<td>Workbook version remains unchanged, but extract upgrades to .hyper format</td>
</tr>
<tr>
<td>Download and open in Tableau Desktop 10.4</td>
<td>✓</td>
<td>Can't open workbook; you see a &quot;this workbook uses a .hyper extract and is not compatible with this version; open the workbook in version 10.5 or later&quot; error message, and then asked to locate the extract</td>
</tr>
</tbody>
</table>
### Scenario 10.4 workbook 10.5 workbook

<table>
<thead>
<tr>
<th>Scenario</th>
<th>.tde extract</th>
<th>.tde extract</th>
<th>.hyper extract</th>
</tr>
</thead>
<tbody>
<tr>
<td>in Tableau Desktop 10.5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Export As Version</td>
<td>✓</td>
<td>Option is greyed out</td>
<td>Option is greyed out</td>
</tr>
</tbody>
</table>

**Automating extract creation, refresh, and append tasks**

When working with extracts created in earlier versions of Tableau, you should be aware of the following extract-related compatibility scenarios when automating extract creation, refresh, and append tasks.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>.tde extract</th>
<th>.hyper extract</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create new extract using Tableau SDK</td>
<td>✓</td>
<td>Not possible</td>
</tr>
<tr>
<td>using Extract API 2.0</td>
<td>Not possible</td>
<td>✓</td>
</tr>
<tr>
<td>Refresh or append using 10.4 Tableau Command-Line utility</td>
<td>✓</td>
<td>Not possible</td>
</tr>
<tr>
<td>using 10.5 Tableau Command-Line utility</td>
<td>Workbook version remains unchanged, but extract upgrades to .hyper format</td>
<td>✓</td>
</tr>
</tbody>
</table>
Why keep an extract in .tde format?

If you cannot upgrade Tableau Desktop, you'll need to keep your extract in .tde format.

How to keep an extract in .tde format

To keep an extract in the .tde format, do not upgrade the extract. To avoid upgrading the extract, do not perform any of the tasks listed in the Tasks that cause an extract upgrade on page 789 section to a .tde extract using Tableau Desktop 2018.2. Then, consider the following suggestions if you need to maintain a .tde version of an extract:

- Perform extract refresh and append data tasks using Tableau Desktop 2018.2, but maintain an earlier version of Tableau Desktop and the ability to connect to the original data in case you need to recreate the .tde extract.
- Use an earlier version of Tableau Desktop to perform extract tasks, such as extract refresh or append data.
- If possible, disable existing extract refresh schedules on Tableau Server, Tableau Online, or Tableau Bridge until you can identify which extracts should and shouldn't be upgraded.

What to expect after an extract upgrade

After your extract has been upgraded, you can expect some additional changes when working in version 2018.2.

Changes to opening extracts

After a .tde extract is upgraded to a .hyper extract, you can't open an upgraded extract using an earlier version of Tableau Desktop. Consider some of the suggestions below to help resolve issues you have with opening extracts.

- **Upgrade Tableau Desktop to 2018.2:** If possible, consider upgrading Tableau Desktop to 2018.2 so that you can experience the benefits of .hyper extracts. For more information, see Upgrade Tableau Desktop in the Tableau Desktop Deployment Guide.

- **Recreate the extract using an earlier version of Tableau Desktop:** To recreate your .tde extract, you must be able to connect to the original data using an earlier version of Tableau Desktop. For more information, see Connect to and Prepare Data on page 319 and Downloading Previous Versions of Tableau in the Tableau Knowledge Base.
• **Retrieve earlier versions of a published extract data source or workbook:** If revision history is enabled on Tableau Server or Tableau Online, you can download an earlier version of a published extract data source or a published workbook that uses an extract data source. **Note:** Earlier workbooks that you retrieve that use an extract data source are saved in a .twb format and therefore will require you to recreate the extract after you open the workbook in an earlier version of Tableau Desktop. For more information, see [Work with Content Revisions](#) on page 2642.

Changes to sharing extracts

When you need to share the data in the extract data source with someone directly (for example, through email or a file share) be sure to save the extract as a .tdsx file. Alternatively, you can save the workbook with the extract data as a .twbx. If you only share the .hyper file instead of the .tdsx or .twbx, metadata about the .hyper extract like connection information and column name changes will be missing. For more information, see [Tableau File Types and Folders](#) on page 241.

Potential differences in extract file size

While .tde extract file sizes can vary depending on complexity of your data, after an extract upgrade, corresponding .hyper extract file sizes can also vary for the same reason. There may be some cases when some .hyper extracts become bigger after an extract upgrade and other cases when .hyper extracts become smaller after an extract upgrade.

Changes to the Export as Version option

When your workbook contains a .hyper extract, the **Export as Version** option in the File menu is not available. If you need to open a workbook that contains a .hyper extract using an earlier version of Tableau, you must first remove the extract from the workbook before you can downgrade it using the **Export as Version** option. After you've downgraded the workbook, you must recreate the .tde extract.

Other issues

For errors that you see after an extract upgrade, refer to the following articles in the Tableau Knowledge Base:

- Error when opening .hyper extract in Tableau Desktop 10.4 or earlier
- "Export as Version" grayed out for workbooks with .hyper extract
Changes to your view after an extract upgrade

After an extract has been upgraded, you might notice some changes to a view that uses the extract data source. This is because values in your extract can be computed differently in version 2018.2. Changes can include fewer marks in the view, more null values when you inspect the summary data, or the view itself has a different shape. In some rare cases, the view might even be blank. For more information about the type of changes you might see when interacting with extracts with this version of Tableau, see Changes to values and marks in the view on page 775.

Manually upgrade your extract

If you manage extracts locally, you can manually upgrade your .tde extract to a .hyper extract using Tableau Desktop.

1. In Tableau Desktop, open a workbook that uses a .tde extract.

2. Select the extract data source from the Data menu and then select Upgrade Extract.

Refresh Extracts

In this article

Before you refresh extracts on the next page
Configure a full extract refresh on the next page
Configure an incremental extract refresh on page 800
See extract history on page 801

When the original data changes, you can refresh the extract using Desktop by selecting a data source on the Data menu and then selecting Extract > Refresh. Extracts can be configured to be fully refreshed, replacing all of the data with what’s in the original data source, or incrementally refreshed, adding just the new rows since the previous refresh.

**Note:** You can also automate extract refreshes using the Tableau Data Extract Command Line Utility. For more information, see Tableau Data Extract Command-Line Utility on page 806.
Before you refresh extracts

If you want to refresh an extract, make note of the file format of the extract before you perform an extract refresh. If you perform a refresh on an .tde extract using version 2018.2, the extract is upgraded to .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, you will be unable to open the extract with previous versions of Tableau Desktop. For more information, see Extract Upgrade to .hyper Format on page 788.

Configure a full extract refresh

By default, extracts are configured to fully refresh. This means that every time you refresh the extract, all of the rows are replaced with the data in the original data source. While this kind of refresh ensures that you have an exact copy of what is in the original data, depending on the size of the extract, a full refresh can sometimes take a long time and be expensive on the database. If an extract is not configured for an incremental refresh, selecting to refresh the extract will perform a full refresh of the extract.

Publish to Tableau Server

If you plan to publish the extract as a data source to Tableau Server, you can specify a schedule for the extract refresh during publishing.
Similarly, if you are publishing the extract in a workbook to Tableau Server, you can also specify a schedule for the extract refresh during publishing. For more information, see Schedule Extract Refreshes as You Publish a Workbook on page 2540.

Publish to Tableau Online

If you plan to publish the extract as a data source to Tableau Online, your options for refreshing your data depend on the characteristics of your data sources. For more information on refreshing data from specific data sources, see Extract refresh options by data source.

Configure an incremental extract refresh

Rather than refreshing the entire extract, you can configure a refresh to add only the rows that are new since the previous time you extracted the data. For example, you may have a data source that is updated daily with new sales transactions. Rather than rebuild the entire extract each day, you can just add the new transactions that occurred that day. Then once a week you might want to do a full refresh just to be sure you have the most up to date data.

**Note:** Most data sources support an incremental refresh.

You can follow the steps below to set up an extract to be refreshed incrementally.

1. Select a data source on the **Data** menu and then select **Extract Data**.
2. In the Extract Data dialog box, select **All rows** as the number of Rows to extract. Incremental refresh can only be defined when you are extracting all rows in the database. You cannot increment a sample extract.
3. Select **Incremental refresh** and then specify a column in the database that will be used to identify new rows. For example, if you select a Date field, refreshing will add all rows whose date is after that last time you refreshed. Alternatively, you can use an ID column that increases as rows are added to the database.
**Note:** The data engine, which is the underlying mechanism that Tableau uses to create extracts, stores time values with a precision of up to 3 decimal places. If you specify a datetime or timestamp column for **Identify new rows using column**, and your database uses a higher precision than Tableau, you can end up with duplicate rows after an incremental refresh. For example, if the database has two rows, one with a datetime value of 2015-03-13 17:30:56.502352 and one with a datetime value of 2015-03-13 17:30:56.502852, Tableau will store both rows using a datetime value of 2015-03-13 17:30:56.502 thereby creating duplicate rows.

4. When finished, click **Extract**.

The steps above can be used to define a new extract or edit an existing extract for an incremental refresh. If you are editing an existing extract, the last refresh is shown so you can be sure you are updating the extract with the correct data.

**See extract history**

You can see a history of when the extract was refreshed by selecting a data source on the **Data** menu and then select **Extract > History**.

The Extract History dialog box shows the date and time for each refresh, whether it was full or incremental, and the number of rows that were added. If the refresh was from a file, it also shows the source file name.
Add Data to Extracts

In this article

Before you add data to extracts below
Add data from a file below
Add data from a data source on the next page

There are two ways you can add new data to an extract: from a file or from a data source. However, to add new data, you must connect first connect to data and create a new data source, and the columns in the file or data source must match the columns in the extract.

Before you add data to extracts

If you want to add data to an extract, make note of the file format of the extract before you perform the task. If you add data to a .tde extract using version 2018.2, the extract is upgraded to .hyper extract. While there are many benefits of upgrading to a .hyper extract, you will be unable to open the extract using previous versions of Tableau Desktop. For more information, see Extract Upgrade to .hyper Format on page 788.

Add data from a file

You can add new data to an extract from a file-based data source. Use this option when the file type of the extract is the same as the file type of the data that you want to add. Alternatively, you can add data from an extract (.tde or .hyper) file. For example, you may have text files that are generated for a task that is performed every day. To add each day’s worth of information to your extract whose original data source is also a text file, use the Append Data from File command.

To add data from a file

1. On the Data menu, select a data source, and then select Extract > Append Data from File.
2. Browse to and select the file that has the new data.

Note: By default, the file format of the extract's original data source is used. To add data from a Tableau data extract, click the file format drop-down list, and then
select Tableau Data Extract (*.tde) or Hyper Extracts (*.hyper).

3. When finished, click OK.

Add data from a data source

You can also add new data to an extract from another data source that's already open in the workbook. Use this option when the data from the data source that you want to add matches the extract data source before it was extracted. For example, you created an extract from a SQL Server database that has the past ten years worth of data. However, the maintained version of that data is stored on a different SQL Server database. You can add new data to the extract by using the Append Data from Data Source command.

**Note:** Joins or custom SQL should be specified in the data source before adding data to the extract.

To add data from a data source

1. On the Data menu, select a data source, and then select Extract > Append Data from Data Source.
2. In the dialog box, select the data source that you want to append.
3. When finished, click OK.

Using either option will add new rows to the extract. To see a summary of the number of rows that were added, select a data source on the Data menu and then select Extract > History.

**Note:** When you refresh this extract, the data will be replaced with the data from the original data.

Materialize Calculations in Your Extracts

You can use the Compute Calculations Now option to materialize calculations in your extract. When calculations are materialized, certain calculations are computed in advance and its values stored in the extract. Depending on the complexity of the calculations used in your extract, this can potentially speed up future queries by allowing Tableau to compute certain calculations in advance. Tableau can then use the precomputed value rather than computing the value each time a query against a calculation is made.
If the formula for a calculation that Tableau has already materialized changes or the calculation is deleted from the data source, the materialized calculation is removed from the extract until you use the **Compute Calculations Now** option again.

**When to materialize calculations**

You should not use the **Compute Calculations Now** option for all your extracts. Instead, try this option only when the query performance of your extract is slow as a result of complex calculations, such as string manipulations and regular expressions.

**Materialize calculations**

To materialize calculations in your extract, do the following:

1. In Tableau Desktop, select the extract data source from the **Data** menu.
2. Select **Extract > Compute Calculations Now**.

**Materialize calculations in extracts already published to Tableau Online or Tableau Server**

If you need to retroactively materialize calculations in extracts that are already published to Tableau Online or Tableau Server, you have the following options:

- For Tableau Online or Tableau Server, you can download the extract, open the extract in Tableau Desktop, use the **Compute Calculations Now** option (as described above), and then republish the extract.
- For Tableau Online only, as a site admin or data source owner, you can use `tabcmd` to refresh the published extract with the `-addcalculations` option to materialize the calculations. For more information, see [refreshextracts](https://help.tableau.com/current/doc/en-us/rn/rn_tableau_platform_api.htm#refreshextract) in the Tableau Online Help.

**Note:** Materializing calculations can increase the size of the extract.

**Calculations that cannot be materialized**

The following calculations cannot be materialized:

- Calculations that use unstable functions such as `NOW()` and `TODAY()`
- Calculations that use external functions such as `RAWSQL` and `R`
- Table calculations
- Level of detail (LOD) calculations
Update Server Data Sources That Are Using Extracts

You have the following options for updating extract data sources that are published to Tableau Server or Tableau Online:

- **Option 1:** You can add the data source or a workbook that connects to it to a refresh schedule in Tableau Server or Tableau Online (cloud-based data sources only).

- **Option 2:** You can update the data source in Tableau Desktop and then republish it.

- **Option 3:** You can add to or refresh the data source on Tableau Server or Tableau Online, from Tableau Desktop, without first adding to or refreshing the extract in Tableau Desktop.

The remainder of this topic describes option 3.

Update Server or Online extracts from Tableau Desktop

Before you attempt to update an extract data source on Tableau Server or Tableau Online, verify first that Tableau Desktop is connected to a published data source. You can verify that Tableau Desktop is connected to a published data source by the Tableau Server icon next to the data source name in the Data pane:

Note: If you want to refresh an extract data source, make note of the file format of the extract before you perform an extract refresh. If you perform a refresh on an .tde extract using version 2018.2, the extract is upgraded to .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, you will be unable to open the extract with previous versions of Tableau Desktop. For more information, see Extract Upgrade to .hyper Format on page 788 What's new to extracts on page 774.

To update the published data source, do the following:

1. In Tableau Desktop, right-click (control-click on a Mac) the data source in the Data pane
2. Select Tableau Data Server, and choose one of the following options:
• **Edit Server and Site Path**
  
  If you have changed the location of the data source in Tableau Server or Tableau Online, choose this option to point Tableau Desktop to the new, correct location.

• **Refresh from Source**

  Refreshes the extract (full or incremental) using the original data.
  
  This command is available only for extracts that include a connection to the original data. If you connected directly to an extract file (.hyper or .tde) and then published it, the connection to the original data is not included and this option is therefore unavailable.

• **Append from Data Source**

  Updates the extract from another open (and compatible) data source. If you connected directly to an extract file (.hyper or .tde) and then published it, the connection to the original data is not included and this option is therefore unavailable.

**Note:** If you see the Tableau Data Server option, but the commands in the submenu are unavailable, the data source exists on the server, but is not an extract.

It is also possible to update an extract on Tableau Server using a command-line utility. For more information, see Tableau Data Extract Command-Line Utility below.

**Tableau Data Extract Command-Line Utility**

You can automate extract refresh tasks using the Tableau Data Extract Command-Line Utility. This is a command-line utility that comes with Tableau Desktop, through which you can refresh published extract data sources or append data to them from a file.

Requirements for using the Tableau Data Extract Command-Line Utility include the following:

- It is available with Tableau Desktop on Windows and can run only on a Windows system.
- You can use it for extract data sources that don't use OAuth.
- You can use it to refresh single-connection data sources only. It does not work for multi-connection data sources.

**Before you run the utility**

When an extract refresh or append data is performed on extracts created in Tableau 10.4 and earlier (that is, a .tde extract), the extract is upgraded to .hyper extract automatically. While
there are many benefits of upgrading to a .hyper extract, you will be unable to open the extract with previous versions of Tableau Desktop.

For more information, see **Extract Upgrade to .hyper Format** on page 788.

**Run the utility**

1. Open the Command Prompt as an administrator and change to the Tableau Desktop bin directory. For example:

   ```
   cd C:\Program Files\Tableau\Tableau 2018.2\bin
   ```

2. Use either of the following commands, adding parameters described in the tables below.

   - `tableau refreshextract`
   - `tableau addfiletoextract`

**Note:** When using the utility, always specify `tableau` on the command line or in scripts, never `tableau.exe`.

**Syntax and parameters for the tableau refreshextract command**

Use `tableau refreshextract` to refresh an extract on Tableau Server or Tableau Online. Refreshing an extract updates an existing extract with any modifications that have been made to the data source since the last refresh.

To see help for this command, at the Windows command prompt, type the following command:

```
tableau refreshextract --help
```

**Using parameters**

- All options have a full form that you use with a double hyphen (for example, `--server`).
- Some options also have a short form that you use with a single hyphen (for example, `-s`).
- If the value for an option contains spaces, enclose it in quotation marks.
- The source being refreshed must be the original data source, not a .tde.
## Tableau refreshextract command options

<table>
<thead>
<tr>
<th>Short Form</th>
<th>Full Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--source-username &lt;user name&gt;</code></td>
<td></td>
<td>A valid user name for the data source connection. Use this option with <code>--source-password</code>, or use <code>--original-file</code> instead of the user name and password options. Note: You must provide the user name and password when refreshing a published extract, even if the data source was originally published with embedded credentials.</td>
</tr>
<tr>
<td><code>--source-password &quot;&lt;password&gt;&quot;</code></td>
<td></td>
<td>The password for the data source user.</td>
</tr>
<tr>
<td><code>--original-file &lt;path and file name&gt;</code></td>
<td></td>
<td>Path and file name for the data source to be refreshed on the server. For example: <code>--original-file c:\folder\file.csv</code> To refresh a multi-file data source, pass the path to a folder that contains the data files. For example: <code>--original-file c:\folder</code>. If the file is on a network share, use the UNC format for the path: <code>\server\path\filename.csv</code></td>
</tr>
<tr>
<td><code>--force-full-refresh</code></td>
<td></td>
<td>If the data source is set up for incremental refreshes, use this option to force a full extract refresh. If this option is not included, an</td>
</tr>
</tbody>
</table>
incremental refresh is performed. Not all data sources support incremental refresh.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-s</code> <code>&lt;server http address&gt;</code></td>
<td>The URL for the Tableau server on which the data is published. For Tableau Online, specify <a href="https://online.tableau.com">https://online.tableau.com</a>.</td>
</tr>
<tr>
<td><code>-t</code> <code>&lt;site id&gt;</code></td>
<td>In a multiple-site environment, specifies the site to which the command applies. For Tableau Online, use this argument if your user name is associated with more than one site. For Tableau Server, if you do not specify a site, the default site is assumed. The site id is independent of the site name, and it is indicated in the URL when you view the site in a browser. For example, if the URL for the page you see after signing in to Tableau Online is <a href="https://online.tableau.com/t/vernazza/views">https://online.tableau.com/t/vernazza/views</a> the site id is vernazza.</td>
</tr>
<tr>
<td><code>--data-source </code>&lt;data-source&gt;`</td>
<td>The name of the data source, as published to Tableau Server or Tableau Online.</td>
</tr>
<tr>
<td><code>--project </code>&lt;projectname&gt;`</td>
<td>The project to which the data source belongs. If this option is not included, the default project is assumed. If the project you want to specify is a child project nested within a project hierarchy, you must use this parameter along with the <code>--parent-project-path</code> parameter.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--parent-project-path</td>
<td>If a project to which the data source is published is not at the top level of a project hierarchy, use this parameter along with the --project parameter to specify the path to a nested project. Use the forward slash (/) to delimit project levels in the hierarchy. Use the backward slash (), to escape instances of forward or backward slash characters in project names. For example, for a project named Sandbox, in project Social, under top-level Marketing: --project Sandbox --parent-project-path Marketing/Social</td>
</tr>
<tr>
<td>-u &lt;username&gt;</td>
<td>Valid Tableau Server or Tableau Online user.</td>
</tr>
<tr>
<td>--username &lt;username&gt;</td>
<td></td>
</tr>
<tr>
<td>-p &quot;&lt;password&gt;&quot;</td>
<td>The password for the specified Tableau Server or Tableau Online user.</td>
</tr>
<tr>
<td>--password &quot;&lt;password&gt;&quot;</td>
<td></td>
</tr>
<tr>
<td>--proxy-username &lt;username&gt;</td>
<td>The user name for a proxy server.</td>
</tr>
<tr>
<td>--proxy-password &quot;&lt;password&gt;&quot;</td>
<td>The password for a proxy server.</td>
</tr>
<tr>
<td>-c &quot;&lt;path and file name&gt;&quot;</td>
<td>Path and file name information for a file containing configuration options for the command. Always enclose the path in double quotation marks. For more information, see Using a config file on page 815 below.</td>
</tr>
</tbody>
</table>
Sample tableau refreshextract command

The following command refreshes an extract named CurrentYrOverYrStats that has been published to Tableau Server on-premises. This command specifies the following:

- The name of your Tableau Server.
- Server user name and password.
- Project name.
- The name of the data source to refresh, along with the data source username and password.

```
C:\Program Files\Tableau\Tableau 2018.2\bin>tableau
refreshextract --server https://our_server_name --username
OurServerSignIn --password "OurServerPwd" --project "New
Animations" --datasource "CurrentYrOverYrStats" --source-
username OurDatabaseSignIn --source-password
"OurDatabasePassword"
```

The following command refreshes an extract named CurrentYrOverYrStats that has been published to Tableau Online. This command specifies the following:

- Tableau Online user and password.
- Tableau Online site and project names.
- The data source, which in this case is hosted by a cloud-based data source provider (for example, Salesforce.com), and the username and password to sign in to the hosted data source.

```
C:\Program Files\Tableau\Tableau 2018.2\bin>tableau
refreshextract --server https://online.tableau.com --
username email@domain.com --password "OurServerPwd" --site
vernazza --project "New Animations" --datasource
"CurrentYrOverYrStats" --source-username database_
user@hosted_datasource_provider.com --source-password "db_
password"
```

To refresh an extract of file-based data source, provide the path to the original file from which you created the extract. If the file is on a network share, use the UNC format instead of a mapped drive.
C:\Program Files\Tableau\Tableau 2018.2\bin>tableau
refreshextract --server https://online.tableau.com --
username email@domain.com --password "OurServerPwd" --site
vernazza --project "New Animations" --datasource
"CurrentYrOverYrStats" --original-file
"\server\path\filename.csv"

Syntax for tableau addfiletoextract

Use tableau addfiletoextract to append file content to an extract that has been
published to Tableau Server or Tableau Online. This command combines the two files.

If you want simply to update an existing extract with the latest changes, use the
refreshextract command instead. Using addfiletoextract to update an existing
extract will duplicate data instead.

To see help for this command, at the Windows command prompt, type the following command:

tableau addfiletoextract --help

All options have a full form that you use with a double hyphen (for example, --server). Some
options also have a short form that you use with a single hyphen (for example, -s). If the value
for an option contains spaces, enclose it in quotation marks.

tableau addfiletoextract command options

<table>
<thead>
<tr>
<th>Short Form</th>
<th>Full Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--file &lt;path and file name&gt;</td>
<td>Path and file name information for the data file containing data to append. The file can be from Excel, Access, a Tableau data extract, or a delimited text file. It cannot be password protected. Use UNC format if the file is on a network share. For example, \server\path\filename.csv</td>
</tr>
<tr>
<td>-s &lt;server http address&gt;</td>
<td>--server &lt;URL&gt;</td>
<td>The URL for the Tableau server on which the data is published.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><code>-t &lt;site id&gt;</code></td>
<td>In a multiple-site environment, specifies the site to which the command applies. For Tableau Online, you must include this argument if your user name is associated with more than one site. For Tableau Server, if you do not specify a site, the default site is assumed.</td>
<td></td>
</tr>
<tr>
<td><code>--site &lt;site id&gt;</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>--datasource &lt;datasource&gt;</code></td>
<td>The name of the data source, as published to Tableau Server or Tableau Online.</td>
<td></td>
</tr>
<tr>
<td><code>--project &lt;projectname&gt;</code></td>
<td>The project to which the data source belongs. If this option is not included, the default project is assumed. If the project you want to specify is a child project nested within a project hierarchy, you must use this parameter along with the <code>--parent-project-path</code> parameter.</td>
<td></td>
</tr>
<tr>
<td><code>--parent-project-path path/to/project</code></td>
<td>If a project to which the data source is published is not at the top level of a project hierarchy, use this parameter along with the <code>--project</code> parameter to specify the path to a nested project. Use the forward slash character (<code>/</code>) to delimit project levels in the hierarchy. Use the backward slash</td>
<td></td>
</tr>
</tbody>
</table>

For Tableau Online, specify [https://online.tableau.com](https://online.tableau.com).
(\), to escape instances of forward or backward slash characters in project names.

For example, for a project named Sandbox, in project Social, under top-level Marketing:

```
--project Sandbox --parent--project--path Marketing/Social
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-u &lt;username&gt;</code></td>
<td><code>--username &lt;username&gt;</code> Valid Tableau Server or Tableau Online user.</td>
</tr>
<tr>
<td><code>-p &quot;&lt;password&gt;&quot;</code></td>
<td><code>--password &quot;&lt;password&gt;&quot;</code> The password for the specified Tableau Server or Tableau Online user.</td>
</tr>
<tr>
<td><code>-c &quot;&lt;path and filename&gt;&quot;</code></td>
<td><code>--config-file &quot;&lt;path and filename&gt;&quot;</code> Path and file name information for a file containing configuration options for the command. Always enclose the path in double quotation marks. For more information, see Using a config file on the next page below.</td>
</tr>
</tbody>
</table>

Sample tableau addfiletoextract command

```bash
C:\Program Files\Tableau\Tableau 2018.2\bin>tableau addfiletoextract --server https://our_server_name --username OurServerSignIn --password "OurServerPwd" --project "New Animations" --datasource "CurrentYrOverYrStats" --file "C:\Users\user1\Documents\DataUploadFiles\AprMay.csv"
```
C:\Program Files\Tableau\Tableau 2018.2\bin>tableau
addfiletoextract --server https://online.tableau.com --username email@domain.com --password "OurServerPwd" --site vernazza --project "New Animations" --datasource "CurrentYrOverYrStats" --file "C:\Users\user2\Documents\DataUploadFiles\AprMay.csv"

Using a config file

You can use a plain text editor, such as Notepad or Text Edit, to create a config (configuration) file that you can use with `tableau refreshextract` or `tableau addfiletoextract`. A config file can be useful if you expect to update the same data source regularly over time. Instead of having to type the same options each time you run a command, you specify the config file. A config file also has the advantage of not exposing user names and passwords on the command line.

Create the config file

For example, say you created a file called `config.txt` and saved it to your Documents folder. And in the file, you included the parameter information shown below.

For an extract published to Tableau Server:

```plaintext
server=https://our_server_name
username=OurServerSignIn
password=OurServerPwd
project=New Animations
datasource=CurrentYrOverYrStats
```

For an extract from a hosted data source, published to Tableau Online, where `server` is `https://online.tableau.com`:

```plaintext
server=https://online.tableau.com
username=email@domain.com
password=OurPassword
project=New Animations
datasource=CurrentYrOverYrStats
source=username=database_user@hosted_datasource_provider.com
```
Reference the Config File from the Command Line

After you create the config file, you run the `tableau refreshextract` or `tableau addfiletoextract` command, pointing to the config file as the only option you use on the command line, and enclosing the config file’s path in double quotation marks. The syntax is as follows:

```
tableau refreshextract --config-file "<path>"
```

For example, to refresh the extract specified in the sample in the Create the config file on the previous page section, you would run the following command (making sure that you are working in the bin directory for your version of Tableau Desktop):

```
C:\Program Files\Tableau\Tableau 2018.2\bin>tableau
refreshextract --config-file "C:\Users\user1\Documents\config.txt"
```

Syntax Differences for Config Files

The syntax for specifying options inside a config file differs from the syntax you use on the command line in the following ways:

- Option names do not begin with dashes or hyphens.
- You use an equals sign (with no spaces) to separate option names from option values.
- Quotation marks are not necessary (or allowed) around values, even when they include spaces (as for the project option in the example shown earlier).

Use Windows Task Scheduler to Refresh Extracts

You can use Windows Task Scheduler, in combination with the Tableau Data Extract Command-Line Utility, to automate regular updates to Tableau Desktop data sources from within your corporate firewall. You can configure a task to occur once per day, week, or month, or after a specific system event. For example, run the task when the computer starts.

To learn more, see the Task Scheduler How To... page in the Microsoft TechNet library.
Tableau Extract API

Starting in Tableau 10.5, you can use the Tableau Extract API 2.0 to create .hyper extracts.

For other tasks that you previously performed using the Tableau SDK, such as publishing extracts, you can use the Tableau Server REST API or the Tableau Server Client (Python) library instead. You can also use the Tableau Server REST API to refresh extracts.

- For more information about the Tableau Extract API, see the Extract API 2.0 documentation.
- For more information about the .hyper format, see Extract Upgrade to .hyper Format on page 788.
- For more information about the Tableau Server REST API or the Tableau Server Client (Python) library, see the Tableau Server REST API documentation or the Tableau Server Client (Python) library documentation.

If you’re using Tableau 10.4 and earlier, you can continue to use the Tableau SDK to create .tde extracts and publish those extracts to Tableau Server. For more information, see the Tableau SDK documentation.

Both the SDK and the Extract API 2.0 support Windows, Linux, and the Mac, using C, C++, Java, and Python.

Manage Data Sources

The topics in this section describe the tasks you can perform after you have set up data sources.

Edit Data Sources

In this article

- How to edit the data source on the next page
- Navigate the Data Grid on page 819
- Manage Metadata on page 822
- Change the Location of the Data Source on page 824
- Replace Field References on page 825
- Replace Field References on page 825
At anytime during your analysis you can edit the data source used in the workbook.

How to edit the data source

1. On the **Data** menu, select a data source, and then select **Edit Data Source**.
2. On the data source page, make the changes to the data source.

You might want to edit the data source to:

- **Join Your Data** on page 657
- **Connect to a Custom SQL Query** on page 723 (Tableau Desktop)
- **Use a Stored Procedure** on page 733 (Tableau Desktop)
Navigate the Data Grid

The following enhancements made to the grid on the Data Source page help you better see the data in your data source and prepare it for analysis.

Sort columns and rows

Sort columns: Sort columns in the grid and metadata grid by selecting a sort option from the Sort fields drop-down list. Sort the columns by table or data source order.

Sort rows: Sort rows by clicking the sort button. Click the sort button once to sort rows in ascending order, click the sort button again to sort rows in descending order, and then click the sort button a final time to clear the sort.

A Modified sort state can occur when some tasks cause new columns to be added to the grid.
Change or reset field names

**Rename column:** Double-click the name of the column to rename the field.

<table>
<thead>
<tr>
<th>Orders</th>
<th>Orders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claire Gutte</td>
<td>Claire Gutte</td>
</tr>
<tr>
<td>Darrin Van Huff</td>
<td>Sean O’Donnell</td>
</tr>
<tr>
<td>Sean O’Donnell</td>
<td>Sean O’Donnell</td>
</tr>
<tr>
<td>Brosina Hoffman</td>
<td></td>
</tr>
</tbody>
</table>

**Reset name:** If you've renamed a field, click the column drop-down menu, and then select Reset Name to revert back to the original name of the field. You can also select multiple columns and perform the same action.

Original name indicates the name specified in the underlying data. You can use the Revert command to reset field names that have changed as a result of naming improvements Tableau has automatically made to the data source. For more information, see *Understand Field Type Detection and Naming Improvements* on page 768.
Create new columns and see extract data

**Create new calculations:** Create new calculations, groups, or bins based on existing fields in the data source. Click the column drop-down arrow and select one of the respective options.

If you create an aggregate calculation, the values displayed in the grid are not meaningful until the calculation is used in context of the view’s level of detail.

**See extract data:** For the Web Data Connector, file- and relational-based data sources in Extract mode, you can see extract data in the grid, including extract only calculations like MEDIAN.
When in Extract mode, the row order of values in the data source might be different than the row order of values in Live mode.

Copy values

**Copy values:** Copy values in the grid by selecting the values and then pressing Ctrl+C (Command-C on a Mac). Alternatively, to copy values in the metadata grid, select the values, right-click, and then select Copy.

Manage Metadata

Use the metadata area to quickly examine and change the structure of the Tableau data source. You can use the metadata area to review the columns in your data and their data types, do routine management tasks such as renaming a column, hiding a column, changing the data type of a column, and changing the geographical role of the column.

**Note:** Metadata management tasks vary when working with a multi-dimensional data source.

Go to the metadata area

After you have set up your data source, click the metadata area button. If you are connected to a cube (multidimensional) data source, the metadata area is shown by default. For other data source types, such as Salesforce or Tableau data extracts (TDE), the default view will vary.
Review metadata

The columns of the data source are represented as rows in the metadata area. Each row shows the data type of the column, the column name in the Tableau data source, the name of the table in the underlying data from which the column comes, and the name of the column in the underlying data.
Manage metadata

Depending on the type of data you're connected to, you can perform routine management tasks like hiding multiple columns at once or quickly renaming columns. For example, to hide multiple columns at once, press the CTRL or Command key, select the columns you want to hide, click the drop-down arrow next to a column name, and then select **Hide**. Changes that you make in the metadata area of the data source do not modify your underlying data.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Table</th>
<th>Remote Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Date]</td>
<td>REI</td>
<td>Date</td>
</tr>
<tr>
<td>![Category1]</td>
<td>REI</td>
<td>Category1</td>
</tr>
<tr>
<td>![Category2]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Category3]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Category4]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Category5]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Trans Description]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Total Units]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Total Sale Revenue]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Rename]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Copy Values]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Hide]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Aliases...]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Create Calculated Field...]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Create Group...]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Split]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Custom Split...]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>![Describe...]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Change the Location of the Data Source

When you edit a data source, you have the option of changing the location of the data source. For example, suppose the name or location of a data source you were using has changed and is no longer available using the previous connection information. In this case, you can direct the workbook to the correct location without losing your work.

You can use this same method to apply analyses performed on one data source to a similar data source. Suppose you create a workbook containing several views involving markets, products, sales, and profits and you want to apply the analyses to a new data source. Instead of recreating each view, you can edit the original data connection and specify a new data source.

To change the location of a data source

1. Right-click (control-click on a Mac) the data source in the Data pane and then select **Edit Connection**.
2. In the Edit Connection dialog box, navigate to the location of the data source or select a new data source.

Replace Field References

When you successfully connect to a new data source, all worksheets in the workbook that previously referred to the original data source now refer to the new data source. If the new data source does not have the same field names as the original workbook, the fields become invalid and are marked with an exclamation point 🔄. You can quickly resolve the problem by replacing the field’s references.

For example, say you have a workbook connected to a data source that contains a Customer Name field. Then you edit the data source to point to a new data source that has all the same data but instead of Customer Name, the field name has been changed to Name. The Customer Name field remains in the Data pane but is marked as invalid. To make the field valid, you can replace the references, which means you can map the invalid field to a valid field in the new data source (for example, Customer Name corresponds to Name).
To replace field references

1. Right-click (control-click on a Mac) the invalid field in the Data pane and then select Replace References.

2. In the Replace References dialog box, select a field from the new data source that corresponds to the invalid field.
Rename the Data Source

When you connect to a data source, you are given the option to give it a name for use in Tableau Desktop.

To rename the data source

- On the Data menu, select Rename.

Naming a connection is useful when you have a single workbook connected to many data sources. The name you assign can help you keep track of the specifics of the connection. You can also review the connection properties by selecting a data source on the Data menu, and then selecting Properties.
Duplicate the Data Source (Tableau Desktop)

Sometimes you’ll want to make changes to a data source such as add more tables, hide and show fields, set field defaults, and so on. When you make these changes it affects all sheets that use the data source. You can duplicate the data source so that you can make the changes without affecting the existing sheets.

To duplicate a data source

- On the Data menu, select a data source, and then select Duplicate.

When you duplicate a data source, the duplicate source name has “(copy)” appended to the end.
Refresh Data Sources

If the underlying data changes—for example, if new fields or rows are added, data values or field names are changed, or data is deleted, Tableau will reflect those changes the next time you connect to the data source. However, because Tableau Desktop queries the data and does not import the data, you can immediately update Tableau to reflect the data modifications without disconnecting, provided the changes have been saved in the underlying data first.

If you are connected to a data source that has been modified, you can immediately update Tableau Desktop with the changes by selecting a data source on the Data menu and then selecting Refresh.

If a field that is used in a Tableau worksheet is removed from the underlying data of the data source and then the data source is refreshed, a warning message displays indicating that the
field will be removed from the view and the worksheet will not display correctly because of the missing field.

**Note about web authoring (Tableau Online and Tableau Server):** If you connect to a published flat file (Excel or text) directly in Tableau Online or Tableau Server, that data will not be refreshed even if it’s modified. If your data is in an on-premises server and is published to the web through Tableau Desktop, it will be rendered as an extract and won’t be refreshed. If you need to keep data published through Tableau Desktop fresh on the web, you can use Tableau Bridge. To learn more about these requirements, see Keep Data Fresh and Publishers: Use Tableau Bridge to Keep Tableau Online Data Fresh on page 2535

Replace Data Sources

There are times when you may want to update a workbook or sheet to use a different data source. Rather than rebuild your workbook using a new data source, you can replace the data source.

**Note:** When you publish a data source to Tableau Server or Tableau Online, the current workbook in Tableau Desktop will automatically connect to the published data source, and automatically close the local data source. So it is not necessary to replace the data source when you publish a data source.

You can only replace one data source with another data source when both data sources are relational data sources, or when both data sources are cube (multidimensional) data sources. In Tableau, cube data sources are supported for Windows only.

Replacing a data source does not merge or edit the data sources. Rather, replacing a data source simply redirects fields used in the worksheet to map to corresponding fields in the new data source. To successfully replace a data source, any fields in the original data source (for example, calculated fields, groups, sets, parameters, etc.), should also exist in the new data source. If not, then you may want to manually copy and paste these fields to the new data source before replacing the old one.

The two data sources do not have to be identical, however, any differences between the workbooks will affect the sheets in the workbook and the fields in the view. More specifically, any fields, groups, sets, and calculated fields that do not exist in the new data source (or have a different name) are removed from the Data pane. If any fields were used in a view, they will
remain in the Data pane but marked as invalid. Additionally, you might see changes to custom sets, groups, and calculated fields that depend on the missing fields. For more information about how to replace field references to fix invalid fields, see Edit Data Sources on page 817.

1. Open a workbook that connects to the original data source.
2. Select Data > New Data Source and then connect to the new data source.
3. On the Data Source page, drag a table to the canvas to set up the data source (if this is not automatically done for you).
4. Go to the sheet tab and select Data > Replace Data Source.

![Image of Tableau interface showing Data Source and Replace Data Source options]

Note: You must have at least one field in the view to make the Replace Data Source option available.

5. In the Replace Data Source dialog box, select the Current data source and the Replacement data source.
6. When finished, click **OK**.

All worksheets, dashboards, and stories that used the *original* data source are updated to use the *new* data source. You can click **Undo** on the toolbar to revert the change and return to the original data source.

**Save Data Sources**

If you’ve created a data connection that you might want to use with other workbooks or share with colleagues, you can export (save) the data source to a file. You might want to do this also if you’ve added joined tables, default properties, or custom fields—such as groups, sets, calculated fields, and binned fields—to the Data pane.

**Note:** This topic describes how to save a data source that you reuse locally, if you do not publish to Tableau Server or Tableau Online. For information about sharing data sources on a server, see [Publish Data Sources and Workbooks](#).

**Options for saving a local data source**

You can save a data source to either of the following formats:

- **Data Source (.tds)** - contains only the information you need to connect to the data source, including the following:
  - Data source type
  - Connection information specified on the data source page; for example, database server address, port, location of local files, tables
- Groups, sets, calculated fields, bins
- Default field properties; for example, number formats, aggregation, and sort order

Use this format if everyone who will use the data source has access to the underlying file or database defined in the connection information. For example, the underlying data is a CSV file on your computer, and you are the only person who will use it; or the data is hosted on a cloud platform, and your colleagues all have the same access you do.

Packaged Data Source (.tdsx) - contains all information in the data source (.tds) file, as well as a copy of any local file-based data or extracts.

A packaged data source is a single zipped file. Use this format if you want to share your data source with people who do not have access to the underlying data that is defined in the connection information.

Save a data source

1. In Tableau Desktop, open the workbook that has the connection to the data you want to save as a file.
2. At the top of the Data pane, right-click (Control-click on Mac) the name of the data source, and then select Add to Saved Data Sources.
3. Enter a file name, select the file type (.tds or .tdsx), and then click Save.

Connect to your data source

By default, Tableau saves .tds and .tdsx files to the Datasources folder under your Tableau repository. When you use the default location, you can connect to the data source on the Connect pane.
If you specified a different location, you can connect to the data source by selecting **File > Open** and navigating to it.

**Updating a data source after the underlying data location changes**

If you move a local data file that a .tds file contains a reference to, when you try to open the data source in Tableau, you will be prompted to locate or replace the original data source. If you replace the original data source, the replacement data must be of the same type (for example, Excel or MySQL). To avoid saving a specific file path, save the data source as a .tdsx file.

**Upgrade Data Sources**

If you have workbooks that were created before Tableau Desktop 8.2 that use Microsoft Excel or text file data sources, or you are using the Excel or text file legacy connection option, you have the option to upgrade the data sources in your workbook. By upgrading your Excel and text file data sources you can take advantage of 1.) better data interpretation, and 2.) compatibility on the Mac. If you are considering upgrading an existing workbook to use the new connection, review the tables below to compare how your data is processed and displayed between the legacy and default connection types.

**Note:** If there are certain capabilities of the Jet driver that your data relies on, you might need to use the legacy connection to get the expected results. Workbooks that were created before Tableau Desktop 8.2 that use Excel and text file data sources automatically use the legacy connection. To create a new workbook that uses the legacy connection, in Tableau Desktop, navigate to your Excel or text file data, click the drop-down arrow on **Open**, and then select **Open with Legacy Connection**.
File formats and attributes

The following tables show some examples of the types of Excel and text files, and tables that you can connect to in Tableau when using the legacy and default connections types.

Excel

<table>
<thead>
<tr>
<th>Your Data</th>
<th>Legacy Connection</th>
<th>Default Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>.xlsx file format</td>
<td>Allows connections to Excel data in .xlsx file format.</td>
<td>Does not allow connections to Excel data in .xlsx format. Connections to Excel data in .xls or .xlsx file formats can be used instead.</td>
</tr>
<tr>
<td>Excel built-in named ranges</td>
<td>Allows connections to built-in named ranges.</td>
<td>Built-in named ranges are hidden.</td>
</tr>
<tr>
<td>Hidden sheet</td>
<td>Allows connections to a table in a hidden sheet.</td>
<td>Hides a table in a hidden sheet.</td>
</tr>
<tr>
<td>Very hidden sheet</td>
<td>Allows connections to a table in a very hidden sheet.</td>
<td>Hides a table in a very hidden sheet.</td>
</tr>
<tr>
<td>Table that contain charts</td>
<td>Allows connections to tables that contain Excel charts. However, the table does not contain any values.</td>
<td>Hides tables that contain Excel charts.</td>
</tr>
<tr>
<td>Empty table</td>
<td>Allows connections to an empty table.</td>
<td>Hides an empty table.</td>
</tr>
<tr>
<td>Table headers</td>
<td>Treats the first row of a table as a header.</td>
<td>Automatically detects whether the first row of a table is a header.</td>
</tr>
<tr>
<td>Table width</td>
<td>Limits the table width to 255 columns.</td>
<td>No table width constraints.</td>
</tr>
<tr>
<td>Empty columns</td>
<td>Empty columns and rows are visible.</td>
<td>Empty columns and rows are hidden because they do not contain any values.</td>
</tr>
<tr>
<td>Your Data</td>
<td>Legacy Connection</td>
<td>Default Connection</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>and rows</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Text**

<table>
<thead>
<tr>
<th>Your Data</th>
<th>Legacy Connection</th>
<th>Default Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>File name length</td>
<td>Does not allow connections to a file whose file name is more than 64 characters long.</td>
<td>No length constraints for file names.</td>
</tr>
<tr>
<td>Multiple period in the file name</td>
<td>Does not allow connections to a file whose file name contains multiple periods.</td>
<td>No period constraints for file names.</td>
</tr>
<tr>
<td>Number of columns in the file</td>
<td>Uses the number of fields present in the first row to determine the number of columns in the file.</td>
<td>Scans the entire file and uses the most common number of fields in a row to determine the number of columns in the file.</td>
</tr>
<tr>
<td>Headers rows</td>
<td>Does not automatically detect whether the first row in the file is a header.</td>
<td>Automatically detects whether the first row in the file is a header.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can manually specify that the first row in the file is a header.</td>
<td><strong>Note:</strong> You can manually override the detection.</td>
</tr>
<tr>
<td>Empty columns</td>
<td>Empty columns are visible.</td>
<td>Empty columns are hidden because they do not contain any values.</td>
</tr>
</tbody>
</table>

**Characters and formatting**

The following tables show some examples of how your Excel and text file data might be displayed in Tableau when using the legacy and default connection types.
## Excel

<table>
<thead>
<tr>
<th>Your Data</th>
<th>Legacy Connection</th>
<th>Default Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of characters in the field name</td>
<td>Field names are truncated to 64 characters.</td>
<td>No constraint on the number of characters in a field name.</td>
</tr>
<tr>
<td>Special characters in field names</td>
<td>Special characters are not allowed in field names. For example, special characters like quotation marks and periods are converted to number signs. Brackets are converted to parentheses.</td>
<td>Special characters are allowed in field names.</td>
</tr>
<tr>
<td>Leading and trailing spaces in field names</td>
<td>Leading and trailing spaces in field names are allowed.</td>
<td>Leading and trailing spaces in field names are automatically removed from column names.</td>
</tr>
<tr>
<td>Duplicate field names</td>
<td>For duplicate field names, an index number is appended onto the field name. For example, Test Scores1.</td>
<td>For duplicate field names, a space and an index number are appended to the field name. For example, Test Scores 1.</td>
</tr>
<tr>
<td>Excel cell formatting</td>
<td>Supports cell formatting configured using the Format Cells option in Excel.</td>
<td>Does not support cell formatting configured using the Format Cells option in Excel.</td>
</tr>
<tr>
<td>Precision with currency values</td>
<td>Fields that contain currency values are represented with a maximum precision of four digits</td>
<td>Full precision of currency values is represented.</td>
</tr>
</tbody>
</table>
Text

<table>
<thead>
<tr>
<th>Your Data</th>
<th>Legacy Connection</th>
<th>Default Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special characters in file names</td>
<td>Special characters are not allowed in file names.</td>
<td>Special characters are allowed.</td>
</tr>
<tr>
<td></td>
<td>For example, special characters like periods are converted to number signs.</td>
<td></td>
</tr>
<tr>
<td>Characters in header field names</td>
<td>Automatically removes leading spaces from header field names.</td>
<td>Automatically removes leading and trailing spaces from header field names.</td>
</tr>
<tr>
<td>Field separators</td>
<td>Recognizes only commas as field separators.</td>
<td>Automatically detects the field separator.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can manually override the detection.</td>
<td><strong>Note:</strong> You can manually override the detection.</td>
</tr>
<tr>
<td>Text enclosed with quotation marks</td>
<td>Quotation marks used to enclose text values remain visible.</td>
<td>Quotation marks used to enclose text values are automatically hidden.</td>
</tr>
<tr>
<td>ANSI and OEM code pages</td>
<td>ANSI and OEM code pages are supported.</td>
<td>ANSI an OEM code pages are automatically converted to be cross-platform compatible.</td>
</tr>
<tr>
<td>BOM code pages</td>
<td>Inconsistently supports BOM code pages.</td>
<td>Supports BOM code pages.</td>
</tr>
</tbody>
</table>

Data type detection

The following tables show some examples of how data type is detected and how certain values are displayed in Tableau when using the legacy and default connection types.

Excel

<table>
<thead>
<tr>
<th>Data Type Detection</th>
<th>Legacy Connection</th>
<th>Default Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>Data type of a column is</td>
<td>Data type of a column is</td>
</tr>
<tr>
<td>Data Type Detection</td>
<td>Legacy Connection</td>
<td>Default Connection</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>determined by the first 8 rows.</td>
<td>determined by the 95% of the first 10000 rows.</td>
<td>Note: You can manually override the detection.</td>
</tr>
<tr>
<td>Note: After the data type of a column in the table has been determined, it cannot be changed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date values without time</td>
<td>Date values are assigned a datetime data type.</td>
<td>Date values without time are assigned a date data type. Date values with time are assigned a datetime data type.</td>
</tr>
<tr>
<td>Number values</td>
<td>All number values are represented as real numbers.</td>
<td>Number values without decimal points are represented as integers.</td>
</tr>
<tr>
<td>Null values</td>
<td>If a column contains a null cell, the data type for the column is automatically designated as a string data type.</td>
<td>Null cells do not affect data type detection.</td>
</tr>
<tr>
<td>Reference errors or empty cells</td>
<td>If a column contains cells with reference errors or empty cells, the entire column is interpreted as a string data type.</td>
<td>Reference errors or cells with no values do not affect the data type detection.</td>
</tr>
<tr>
<td>Overriding data type detection</td>
<td>After the data type of a column in the table has been determined, it cannot be changed.</td>
<td>Data type of a column can be changed after the automatic detection.</td>
</tr>
<tr>
<td>Time value precision</td>
<td>The smallest measurement of time values is whole seconds.</td>
<td>The smallest measurement of time values is fractional seconds.</td>
</tr>
<tr>
<td>Data Type Detection</td>
<td>Legacy Connection</td>
<td>Default Connection</td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DecimalSymbol and CurrencyDecimalSymbol schema.ini values</td>
<td>Both DecimalSymbol and CurrencyDecimalSymbol schema.ini values are recognized.</td>
<td>DecimalSymbol and CurrencyDecimalSymbol schema.ini values are recognized. However if both values are used, DecimalSymbol takes precedence.</td>
</tr>
<tr>
<td>Cells formatted as text</td>
<td>The data type of a column is detected as a string data type when cells are formatted as text using the Format Cells option in Excel.</td>
<td>Does not support cell formatting configured using the Format Cells option in Excel.</td>
</tr>
</tbody>
</table>

Text

<table>
<thead>
<tr>
<th>Data Type Detection</th>
<th>Legacy Connection</th>
<th>Default Connection (8.2 and later)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>Data type of a column is determined by the first 25 rows.</td>
<td>Data type of a column is determined by the first 1024 rows.</td>
</tr>
<tr>
<td>Boolean (True/False) values</td>
<td>Boolean values are assigned the string data type.</td>
<td>Boolean values are assigned the boolean data type.</td>
</tr>
<tr>
<td>Values that become null</td>
<td>Spaces in a cell, whether enclosed in quotation marks or not, are treated as null values. Columns with null values are detected as the string data type.</td>
<td>Two field separators in a row are treated as a null value. Null values are ignored during data type detection.</td>
</tr>
</tbody>
</table>

Data source connection properties

The following table shows examples of the differences in data source connection properties in Tableau when using the legacy and default connection types.
<table>
<thead>
<tr>
<th>Property</th>
<th>Legacy Connection</th>
<th>Default Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom SQL</td>
<td>Custom SQL is allowed.</td>
<td>Does not allow the use of Custom SQL.</td>
</tr>
<tr>
<td>Join type</td>
<td>Allows left, right, and inner join types.</td>
<td>Allows left, right, inner, and full outer join types.</td>
</tr>
<tr>
<td>Join operators</td>
<td>Allows equal to (=), greater than (&gt;), greater than or equal</td>
<td>Allows equal to (=) join operators.</td>
</tr>
<tr>
<td></td>
<td>to (&gt;=), less than (&lt;), less than or equal to (&lt;=), and not</td>
<td></td>
</tr>
<tr>
<td></td>
<td>equal to (&lt;&gt;)) join operators.</td>
<td></td>
</tr>
</tbody>
</table>

**Close Data Sources**

You can close a data source at any time. Doing so does not modify the data source. Instead, it disconnects Tableau from the data so that you can no longer query it. Additionally, the data source is cleared from the Data pane and all open worksheets associated with the data source are cleared. If you accidentally close a data source, use the Undo button to reopen it. Close a data source by doing one of the following:

- Right-click (control-click on a Mac) the data source at the top of the Data pane and select **Close**.
Select a data source on the **Data** menu and then select **Close**.
Build Charts and Analyze Data

Tableau Desktop is a kind of laboratory in which you can discover the meaning that lies hidden in your data.

In this section you can discover the various features at your disposal as you build views, and learn the basic skills you need to create elegant, insightful views, dashboards, and stories.
Build Common Chart Types in Data Views

This section includes detailed exercises that guide you through the steps involved in building some common chart types in data views. All exercises use the Sample - Superstore data source, which is included with Tableau Desktop. This collection of topics is just a sample of the many types of data views that you can create in Tableau. For details on options for building views from scratch, see Build and Explore Data Views on page 961 and Build a Basic View to Explore Your Data on page 118.

For a 6-minute walkthrough on building data views, see the Getting Started with Visual Analytics free training video. Use your tableau.com account to sign in.

For an in-depth, 25-minute walkthrough of the Tableau environment, see the Getting Started free training video. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Create an Area Chart

An area chart is a line chart where the area between the line and the axis are shaded with a color. These charts are typically used to represent accumulated totals over time and are the conventional way to display stacked lines. Follow the steps below to create an area chart.

The basic building blocks for an area chart are as follows:

<table>
<thead>
<tr>
<th>Mark type:</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns shelf:</td>
<td>Dimension</td>
</tr>
<tr>
<td>Rows shelf:</td>
<td>Measure</td>
</tr>
<tr>
<td>Color:</td>
<td>Dimension</td>
</tr>
</tbody>
</table>

To create an area chart, follow the steps below:

1. Open Tableau Desktop and connect to the Sample - Superstore data source.
2. Navigate to a new worksheet.
3. From the **Data** pane, under Dimensions, drag **Order Date** to the **Columns** shelf.

4. On the Columns shelf, right-click **YEAR(Order Date)** and select **Month**.

5. From the **Data** pane, under Measures, drag **Quantity** to the **Rows** shelf.

6. From the **Date** pane, under Dimensions, drag **Ship Mode** to **Color** on the Marks card.

7. On the Marks card, click the Mark Type drop-down and select **Area**.
The visualization updates to the following:

You can add formatting to an area chart. For example, you can edit the color legend and turn on mark labels and borders. For more information, see *Format at the Worksheet Level* on page 2361.

You can also use highlight actions with area charts. For example, selecting a color in the legend or turning on the highlighter will highlight the entire area instead of just the line. For more information, see *Highlight Actions* on page 1794.
Build a Bar Chart

Use bar charts to compare data across categories. You create a bar chart by placing a dimension on the **Rows** shelf and a measure on the **Columns** shelf, or vice versa.

A bar chart uses the **Bar** mark type. Tableau selects this mark type when the data view matches one of the two field arrangements shown below. You can add additional fields to these shelves.

For more information about the **Bar** mark type, see [Bar mark on page 1106](#).

**Note:** At the end of the procedure is an extra step you can take to display totals at the tops of the bars.

<table>
<thead>
<tr>
<th>Creates Vertical Bars</th>
<th>Creates Horizontal Bars</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Vertical Bars" /></td>
<td><img src="image2" alt="Horizontal Bars" /></td>
</tr>
</tbody>
</table>

To create a bar chart that displays total sales over a four-year period, follow these steps:

1. Connect to the **Sample - Superstore** data source.

2. Drag the **Order Date** dimension to **Columns**.
   
   The data is aggregated by year and column headers appear.

3. Drag the **Sales** measure to **Rows**.
   
   The measure is aggregated as a sum and an axis is created. The column headers move to the bottom of the view.

   Tableau uses **Line** as the mark type because you added the date dimension.
4. On the Marks card, select Bar from the drop-down list.

The view changes to a bar chart.
The marks (which are bars in this case) are vertical because the axis is vertical. The length of each mark represents the sum of the sales for that year. The actual numbers you see here might not match the numbers you see—the sample data changes from time to time.

7. Drag the **Ship Mode** dimension to **Color** on the **Marks** card.

The view shows how different shipping modes have contributed to total sales over time. The ratios look consistent from year to year.
8. Drag the **Region** dimension to **Rows**, and drop it to the left of **Sales** to produce multiple axes for sales by region.
9. To view data in the West region only, you can filter out the other regions. To do this, drag the Region dimension again, this time from the Data pane to the Filters shelf.

10. In the Filter [Region] dialog box, clear the Central, East, and South check boxes, and then click OK.
This view gives you insight into your data—for example, how the ship mode changed in the West over the four-year period.
Check your work! Watch steps 1-10 below.

One Step Further: Add Totals To Stacked Bars

Adding totals to the tops of bars in a chart is sometimes as simple as clicking the Show Mark Labels icon in the toolbar. But when the bars are broken down by color or size, each individual segment would labeled, rather than the total for the bar. With a few steps, you can add a total label at the top of every bar even when the bars are subdivided as in the view you just created. In the following procedure you will technically be adding a reference line. But by configuring that "line" in a certain way, you end up with the labels you want.

1. From the Analytics pane, drag a Reference Line into the view and drop it on Cell.
2. In the Edit Line, Band, or Box dialog box, set the aggregation for \texttt{SUM(Sales)} to \texttt{Sum}, set \texttt{Label} to \texttt{Value}, and set \texttt{Line} under \texttt{Formatting} to \texttt{None}:
Then click **OK** to close the Edit Reference Line, Band, or Box dialog box.

Your view now has currency totals at the top of each bar:
You may need to adjust the view to make it look just right. If the bars are too narrow, the numbers are truncated; to fix this, press Ctrl + Right on the keyboard to make the bars wider. Or if you want to center the totals over the bars—by default, they are left-aligned. Do the following:

1. Right-click any of the totals on the bar chart and select **Format**.
2. In the Format window, in the **Reference Line Label** area, open the **Alignment**
control and select the Center option for Horizontal alignment:

See also:

*Creation of a Grouped Bar Chart* (Knowledge base article with embedded videos that describes how to create grouped bar charts, also known as side-by-side bar charts.)

**Build a Box Plot**

Use box plots, also known as box-and-whisker plots, to show the distribution of values along an axis.

Boxes indicate the middle 50 percent of the data (that is, the middle two quartiles of the data's distribution).
You can configure lines, called *whiskers*, to display all points within 1.5 times the interquartile range (in other words, all points within 1.5 times the width of the adjoining box), or all points at the maximum extent of the data, as shown in the following image:

![Box and Whisker Plot](image)

The basic building blocks for a box and whisker plot are as follows:

<table>
<thead>
<tr>
<th>Mark type:</th>
<th>Circle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns shelf:</td>
<td>Dimension</td>
</tr>
<tr>
<td>Rows shelf:</td>
<td>Measure</td>
</tr>
<tr>
<td>Detail:</td>
<td>Dimension</td>
</tr>
<tr>
<td>Reference Line:</td>
<td>Box Plot</td>
</tr>
</tbody>
</table>

For information on how to add a reference line, see *Reference Lines, Bands, Distributions, and Boxes* on page 1634.
To create a box plot that shows discounts by region and customer segment, follow these steps:

1. Connect to the **Sample - Superstore** data source.
2. Drag the **Segment** dimension to **Columns**.
3. Drag the **Discount** measure to **Rows**.

   Tableau creates a vertical axis and displays a bar chart—the default chart type when there is a dimension on the **Columns** shelf and a measure on the **Rows** shelf.

4. Drag the **Region** dimension to **Columns**, and drop it to the right of **Segment**.

   Now you have a two-level hierarchy of dimensions from left to right in the view, with regions (listed along the bottom) nested within segments (listed across the top).

5. Click **Show Me** in the toolbar, then select the box-and-whisker plot chart type.

Tableau displays the a box plot:
Notice that there are only a few marks in each box plot. Also, Tableau reassigned Region from the Columns shelf to the Marks card. When you changed the chart type to a box plot, Tableau determined what the individual marks in the plot should represent. It determined that the marks should represent regions. We'll change that.

6. Drag Region from the Marks card back to Columns, to the right of Segment.
The horizontal lines are flattened box plots, which is what happens when box plots are based on a single mark.

Box plots are intended to show a distribution of data, and that can be difficult when data is aggregated, as in the current view.

7. To disaggregate data, select **Analysis > Aggregate Measures**.

This command turns aggregation on or off, and because data is aggregated by default in Tableau, the first time you select this command, it disaggregates the data.

For more information, see **How to Disaggregate Data** on page 289.

Now, instead of a single mark for each column in the view, you see a range of marks, one for each row in your data source.
The view now shows the information we want to see. The remaining steps make the view more readable and appealing.

8. Click the Swap button to swap the axes:

The box plots now flow from left-to-right:

9. Right-click (control-click on Mac) the bottom axis and select Edit Reference Line.

10. In Edit Reference Line, Band, or Box dialog box, in the Fill drop-down list, select an interesting color scheme.
For more on these options, see Add a Box Plot on page 1656 in the Reference Lines, Bands, Distributions, and Boxes article.

Now your view is complete:

You can see that the discount was the same for all segments in the West. You can also see that the interquartile range (from the 25th percentile to the 75th percentile) for discount was greatest in the Central region for the consumer and corporate segments.

For more information about box plots, see Reference Lines, Bands, Distributions, and Boxes on page 1634.
Check your work! Watch steps 1-10 below:

**Build a Bullet Graph**

A bullet graph is a variation of a bar graph developed to replace dashboard gauges and meters. A bullet graph is useful for comparing the performance of a primary measure to one or more other measures. Below is a single bullet graph showing how actual sales compared to estimated sales.

Follow the steps below to learn how to create a bullet graph.

1. Open Tableau Desktop and connect to the **World Indicators** data source.
2. Navigate to a new worksheet.
3. Hold down Shift on your keyboard and then, on the **Data** pane, under Measures > Development, select **Tourism Inbound** and **Tourism Outbound**.
4. In the upper-right corner of the application, click **Show Me**.
5. In Show Me, select the **Bullet Graph** image.
6. Click **Show Me** again to close it.
7. From the **Data** pane, under Dimensions, drag **Region** to the **Rows** shelf.

The graph updates to look like the following:

![Graph with Region in Rows](image)

Check your work! Watch steps 3 - 7 below:

![Data pane and graph](image)

**Swap reference line fields**

Sometimes you might want to swap the reference lines fields. For example, the actual sales is shown as a reference distribution instead of a bar.

To swap the two measures, right-click (control-click on the Mac) the axis and select **Swap Reference Line Fields**.
Edit the distribution

Right-click (control-click on the Mac) the axis in the view and select **Edit Reference Line**, and then select one of the reference lines to modify.
Build a Gantt Chart

Use Gantt charts to show the duration of events or activities.

In a Gantt chart, each separate mark (usually a bar) shows a duration. For example, you might use a Gantt chart to display average delivery time for a range of products.

The basic building blocks for a gantt chart are as follows:

<table>
<thead>
<tr>
<th>Mark type:</th>
<th>Automatic or Gantt Bar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns shelf:</td>
<td>Date or Time field (continuous measure)</td>
</tr>
<tr>
<td>Rows shelf:</td>
<td>Dimension(s)</td>
</tr>
<tr>
<td>Size:</td>
<td>Continuous measure</td>
</tr>
</tbody>
</table>

For more information about the Gantt bar mark type, see Gantt Bar mark on page 1120.
To create a Gantt chart that shows how many days elapse on average between order date and ship date, follow these steps:

1. Connect to the **Sample - Superstore** data source.

2. Drag the **Order Date** dimension to **Columns**.

   Tableau aggregates the dates by year and creates column headers with labels for the years.

3. On the **Columns** shelf, click the **Year (Order Date)** drop-down arrow, and then select **Week Number**.

   ![Column Headers](image)

   The column headers change. Individual weeks are indicated by tick marks because there are 208 weeks in a four-year span—too many to show as labels in the view.
4. Drag the **Sub-Category** and **Ship Mode** dimensions to the **Rows** shelf. Drop **Ship Mode** to the right of **Sub-Category**.

This builds a two-level nested hierarchy of dimensions along the left axis.

Next, we'll size the marks according to the length of the interval between the order date and the ship date. To do this, create a calculated field to capture that interval.

5. In the toolbar menu, click **Analysis > Create Calculated Field**. You can also right-click (Control-click on Mac) any field in the **Data** pane and select **Create > Calculated Field**.
6. In the calculation dialog box, name your calculated field `OrderUntilShip`.

7. Clear any content that's in the **Formula** box by default.

8. In the **Formula** box, enter the following formula and then click **OK**:

   \[ \text{DATEDIFF('day', \{Order Date\}, \{Ship Date\})} \]

   The formula creates a custom measure that captures the difference between the **Order Date** and **Ship Date** values, in days.

9. Drag the **OrderUntilShip** measure to **Size** on the **Marks** card.

   The default aggregation for **OrderUntilShip** is **Sum**, but in this case it makes more sense to average the values.

10. Right-click (Control-click on Mac) the **SUM(OrderUntilShip)** field on the **Marks** card, and then select **Measure (Sum) > Average**.
The view is coming along. But there are too many marks squeezed into the view.
We can make our data more readable by filtering down to a smaller time window.

11. Hold down the Ctrl key (Option key on the Mac) and drag the Week(Order Date) field from the Columns shelf to the Filter shelf.

12. In the Filter Field dialog box, select Range of Dates and then click Next.

13. Set the range to a three-month time interval, such as 1/1/2013 to 3/31/2013, and then click OK.
It can be difficult to get the exact date using the sliders—it's easier just to enter the numbers you want directly into the date boxes or use the calendar to select the dates.

14. Drag the **Ship Mode** dimension to **Color** on the **Marks** card.

Now your view shows you all sorts of information about the lag between order times and ship times.

For example, you can see which ship modes are more prone to longer lag times, whether lag times vary by category, and whether lag times are consistent over time.

**Note:** If you publish this view to Tableau Server, you can include filters that let users interact with the view by varying the time window, or filtering out various sub-categories or ship modes. For more information, see **Publish Data Sources and Workbooks** on page 2495.
Check your work! Watch steps 1-14 below:

Build a Highlight Table

Use highlight tables to compare categorical data using color.

In Tableau, you create a highlight table by placing one or more dimensions on the **Columns** shelf and one or more dimensions on the **Rows** shelf. You then select **Square** as the mark type and place a measure of interest on the **Color** shelf.

You can enhance this basic highlight table by setting the size and shape of the table cells.

To create a highlight table to explore how profit varies across regions, product sub-categories, and customer segments, follow these steps:

1. Connect to the **Sample - Superstore** data source.

2. Drag the **Segment** dimension to **Columns**.

   Tableau creates headers with labels derived from the dimension member names.

3. Drag the **Region** and **Sub-Category** dimensions to **Rows**, dropping **Sub-Category** to the right of **Region**.
Now you have a nested table of categorical data (that is, the **Sub-Category** dimension is nested within the **Region** dimension).

4. Drag the **Profit** measure to **Color** on the **Marks** card.

Tableau aggregates the measure as a sum. The color legend reflects the continuous data range.

![Nested Table Example](image)

In this view, you can see data for only the Central region. Scroll down to see data for other regions.

In the Central region, copiers are shown to be the most profitable sub-category, and binders and appliances the least profitable.

5. Click **Color** on the **Marks** card to display configuration options. In the **Border** drop-down list, select a medium gray color for cell borders, as in the following image:
Now it's easier to see the individual cells in the view:

6. The default color palette is Orange-Blue Diverging. A Red-Green Diverging palette might
be more appropriate for profit. To change the color palette and to make the colors more distinct, do the following:

- Hover over the `SUM(Profit)` color legend, then click the drop-down arrow that appears and select **Edit Colors**.
- In the **Edit Colors** dialog box, in the **Palette** field, select **Red-Green Diverging** from the drop-down list.
- Select the **Use Full Color Range** check box and click **Apply** and then click **OK**.

![Edit Colors Dialog Box](image)

When you select this option, Tableau assigns the starting number a full intensity and the ending number a full intensity. If the range is from -10 to 100, the color representing negative numbers changes in shade much more quickly than the color representing positive numbers.

When you do not select **Use Full Color Range**, Tableau assigns the color intensity as if the range was from -100 to 100, so that the change in shade is the same on both sides of zero. The effect is to make the color contrasts in your view much more distinct.

For more information about color options, see **Color Palettes and Effects** on
Drag the **Sales** measure to **Size** on the **Marks** card to control the size of the boxes by the Sales measure. You can compare absolute sales numbers (by size of the boxes) and profit (by color).

Initially, the marks look like this:
8. To enlarge the marks, click **Size** on the **Marks** card to display a size slider:

9. Drag the slider to the right until the boxes in the view are the optimal size. Now your view is complete:
Check your work! Watch steps 1-9 below:
Build a Histogram

A histogram is a chart that displays the shape of a distribution. A histogram looks like a bar chart but groups values for a continuous measure into ranges, or bins.

The basic building blocks for a histogram are as follows:

<table>
<thead>
<tr>
<th>Mark type:</th>
<th>Automatic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows shelf:</td>
<td>Continuous measure (aggregated by Count or Count Distinct)</td>
</tr>
<tr>
<td>Columns shelf:</td>
<td>Bin (continuous or discrete).</td>
</tr>
</tbody>
</table>

*Note: This bin should be created from the continuous measure on the Rows shelf. For more information on how to create a bin from a continuous measure, see Create Bins from a Continuous Measure on page 956.*

In Tableau you can create a histogram using **Show Me**.

1. Connect to the **Sample - Superstore** data source.
2. Drag **Quantity** to **Columns**.
3. Click **Show Me** on the toolbar, then select the histogram chart type.
The histogram chart type is available in Show Me when the view contains a single measure and no dimensions.

Three things happen after you click the histogram icon in Show Me:

- The view changes to show vertical bars, with a continuous x-axis (1 – 14) and a continuous y-axis (0 – 5,000).
- The Quantity measure you placed on the Columns shelf, which had been aggregated as SUM, is replaced by a continuous Quantity (bin) dimension. (The green color of the field on the Columns shelf indicates that the field is continuous.)

  To edit this bin: In the Data pane, under Dimensions, right-click the bin and select Edit.

- The Quantity measure moves to the Rows shelf and the aggregation changes from SUM to CNT (Count).
The **Quantity** measure captures the number of items in a particular order. The histogram shows that about 4,800 orders contained two items (the second bar), about 2,400 orders contained 4 items (the third bar), and so on.

Let's take this view one step further and add **Segment** to **Color** to see if we can detect a relationship between the customer segment (consumer, corporate, or home office) and the quantity of items per order.

4. Drag **Segment** to **Color**.
The colors don't show a clear trend. Let's show the percentage of each bar that belongs to each segment.

5. Hold down the Ctrl key and drag the CNT(Quantity) field from the Rows shelf to Label.
Holding down the Ctrl key copies the field to the new location without removing it from the original location.

6. Right-click (Control-click on a Mac) the CNT(Quantity) field on the Marks card and select Quick Table Calculation > Percent of Total.

Now each colored section of each bar shows its respective percentage of the total quantity:
But we want the percentages to be on a per-bar basis.

7. Right-click the **CNT(Quantity)** field on the **Marks** card again and select **Edit Table Calculation**.

8. In the Table Calculation dialog box, change the value of the **Compute Using** field to **Cell**.
Now we have the view that we want:
There is still no evidence that the percentages by customer segment show any trend as the number of items in an order increases.
Check your work! Watch steps 1-8 below:

Building Line Charts

Line charts connect individual data points in a view. They provide a simple way to visualize a sequence of values and are useful when you want to see trends over time, or to forecast future values. For more information about the line mark type, see Line mark on page 1107.

**Note:** In views that use the line mark type, you can use the Path property in the Marks card to change the type of line mark (linear, step, or jump), or to encode data by connecting marks using a particular drawing order. For details, see Path properties in the Control the Appearance of Marks in the View on page 1123.

To create a view that displays the sum of sales and the sum of profit for all years, and then uses forecasting to determine a trend, follow these steps:

1. Connect to the Sample - Superstore data source.
2. Drag the Order Date dimension to Columns.
Tableau aggregates the date by year, and creates column headers.

3. Drag the **Sales** measure to **Rows**.

   Tableau aggregates **Sales** as SUM and displays a simple line chart.

4. Drag the **Profit** measure to **Rows** and drop it to the right of the **Sales** measure.

   Tableau creates separate axes along the left margin for **Sales** and **Profit**.
Notice that the scale of the two axes is different—the Sales axis scales from $0 to $700,000, whereas the Profit axis scales from $0 to $100,000. This can make it hard to see that sales values are much greater than profit values.

When you are displaying multiple measures in a line chart, you can align or merge axes to make it easier for users to compare values.

For more information about aligning the axes, see Compare two measures using dual axes on page 1829.

For more information about enforcing a single axis across multiple measures, see Blend axes for multiple measures into a single axis on page 1826.

With either of these options, you can create a combination chart to change the mark type for one of your measures.

For more information, see Create a combo chart (assign different mark types to measures) on page 1834.

5. Drag the SUM(Profit) field from Rows to the Sales axis to create a blended axis. The two pale green parallel bars indicate that Profit and Sales will use a blended axis when you release the mouse button.
The view updates to look like this:
The view is rather sparse because we are looking at a summation of values on a per-year basis.

6. Click the drop-down arrow in the **Year(Order Date)** field on the **Columns** shelf and select **Month** in the lower part of the context menu to see a continuous range of values over the four-year period.
The resulting view is a lot more detailed than the original view:
Notice that the values seem to go much higher just before the end of each year. A pattern like that is known as *seasonality*. If we turn on the forecasting feature in the view, we can see whether we should expect that the apparent seasonal trend will continue in the future.

7. To add a forecast, in the **Analytics** pane, drag the **Forecast** model to the view, and then drop it on **Forecast**.
We then see that, according to Tableau forecasting, the seasonal trend does continue into the future:
Check your work! Watch steps 1-7 below:

Build a Packed Bubble Chart

Use packed bubble charts to display data in a cluster of circles. Dimensions define the individual bubbles, and measures define the size and color of the individual circles.

The basic building blocks for a packed bubble chart are as follows:

<table>
<thead>
<tr>
<th>Mark type:</th>
<th>Circle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail:</td>
<td>Dimension</td>
</tr>
<tr>
<td>Size:</td>
<td>Measure</td>
</tr>
<tr>
<td>Color:</td>
<td>Dimension or Measure</td>
</tr>
<tr>
<td>Label (optional):</td>
<td>Dimension or Measure</td>
</tr>
</tbody>
</table>

To create a basic packed bubble chart that shows sales and profit information for different product categories, follow these steps:
1. Connect to the Sample - Superstore data source.

2. Drag the Category dimension to Columns.
   A horizontal axis displays product categories.

3. Drag the Sales measure to Rows.
   The measure is aggregated as a sum and a vertical axis appears.
   Tableau displays a bar chart—the default chart type when there is a dimension on the Columns shelf and a measure on the Rows shelf.

4. Click Show Me on the toolbar, then select the packed bubbles chart type.

Tableau displays the following packed bubble chart:
5. Drag **Region** to **Detail** on the **Marks** card to include more bubbles in the view.
Next we’ll add another layer of information to the view.

6. Drag **Profit** to **Color** on the **Marks** card:
7. Drag **Region** to **Label** on the **Marks** card to clarify what each bubble represents.
The size of the bubbles shows the sales for different combinations of region and category. The color of the bubbles shows the profit (the darker the green, the greater the profit).

For information about formatting mark labels, see Show and Hide Mark Labels on page 2180.

To further develop this view, you might edit the colors for Profit to show negative profit in a different color, or create a calculated field to shows profit divided by sales (that is, profit margin) and then drop that on Color instead of absolute profit. For more information, see Color Palettes and Effects on page 1145.
Check your work! Watch steps 1-7 below:

Build a Pie Chart

Use pie charts to show proportions of a whole.

The basic building blocks for a pie chart are as follows:

<table>
<thead>
<tr>
<th>Mark type</th>
<th>Color</th>
<th>Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pie</td>
<td>Dimension</td>
<td>Measure</td>
</tr>
</tbody>
</table>

To create a pie chart view that shows how different product categories contribute to total sales, follow these steps:

1. Connect to the Sample - Superstore data source.

2. Drag the Sales measure to Columns.
   Tableau aggregates the measure as a sum.
3. Drag the **Sub-Category** dimension to **Rows**.

   The default chart type is a bar chart.

4. Click **Show Me** on the toolbar, then select the pie chart type.
The result is a rather small pie:

5. To make the chart bigger, hold down Ctrl + Shift (hold down ⌘ + ⌘ on a Mac) and press B several times.

6. To add labels, drag the Sub-Category dimension from the Data pane to Label on the
7. If you don’t see labels, press Ctrl + Shift + B (press ⌘ + ⌥ + B on a Mac) to make sure most of the individual labels are visible.

You can make a pie chart interactive in a dashboard. For more information, see Actions and Dashboards on page 1814.

Note: Pie charts can also be used as a mark type in a visualization. For more information, see Pie mark on page 1119.
Check your work! Watch steps 1-7 below:

Build a Scatter Plot

Use scatter plots to visualize relationships between numerical variables.

In Tableau, you create a scatter plot by placing at least one measure on the **Columns** shelf and at least one measure on the **Rows** shelf. If these shelves contain both dimensions and measures, Tableau places the measures as the innermost fields, which means that measures are always to the right of any dimensions that you have also placed on these shelves. The word "innermost" in this case refers to the table structure.

<table>
<thead>
<tr>
<th>Creates Simple Scatter Plot</th>
<th>Creates Matrix of Scatter Plots</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Tableau Scatter Plot" /></td>
<td><img src="image" alt="Tableau Scatter Plot Matrix" /></td>
</tr>
</tbody>
</table>

- 907 -
A scatter plot can use several mark types. By default, Tableau uses the shape mark type. Depending on your data, you might want to use another mark type, such as a circle or a square. For more information, see Change the Type of Marks in the View on page 1101.

To use scatter plots and trend lines to compare sales to profit, follow these steps:

1. Open the **Sample - Superstore** data source.
2. Drag the **Profit** measure to **Columns**.
   
   Tableau aggregates the measure as a sum and creates a horizontal axis.
3. Drag the **Sales** measure to **Rows**.
   
   Tableau aggregates the measure as a sum and creates a vertical axis.

   Measures contain continuous numerical data. When you plot one number against another, you are comparing two numbers; the resulting chart is analogous to a Cartesian chart, with x and y coordinates.

   Now you have a one-mark scatter plot:

4. Drag the **Category** dimension to **Color** on the Marks card.
This separates the data into three marks—one for each dimension member—and encodes the marks using color.

5. Drag the **Region** dimension to **Detail** on the **Marks** card.

Now there are many more marks in the view. The number of marks is equal to the number of distinct regions in the data source multiplied by the number of departments. (If you're curious, use the **Undo** button on the toolbar to see what would have happened if you'd dropped the **Region** dimension on **Shape** instead of **Detail**.)
6. To add trend lines, from the Analytics pane, drag the Trend Line model to the view, and then drop it on the model type.

A trend line can provide a statistical definition of the relationship between two numerical values. To add trend lines to a view, both axes must contain a field that can be interpreted as a number—by definition, that is always the case with a scatter plot.

Tableau adds three linear trend lines—one for each color that you are using to distinguish the three categories.
7. Hover the cursor over the trend lines to see statistical information about the model that was used to create the line:

For more information, see Assess Trend Line Significance on page 1674. You can also customize the trend line to use a different model type or to include confidence bands. For more information, see Add Trend Lines to a Visualization on page 1661.
Check your work! See steps 1-7 below:

Build a Text Table

In Tableau, you typically create text tables (also called cross-tabs or pivot tables) by placing one dimension on the **Rows** shelf and another dimension on the **Columns** shelf. You then complete the view by dragging one or more measures to **Text** on the **Marks** card.

A text table uses the text mark type. Tableau uses this mark type automatically if the view is constructed using only dimensions (assuming the mark type is set to **Automatic**). For more information about the text mark type, see **Text mark on page 1114**.

To create a text table that shows sales totals by year and category, follow these steps:

1. Connect to the **Sample - Superstore** data source.
2. Drag the **Order Date** dimension to **Columns**.

See Also

**Example: Scatter Plots, Aggregation, and Granularity** on page 290
Tableau aggregates the date by year and creates column headers.

3. Drag the Sub-Category dimension to Rows.

Tableau creates row headers. Columns with headers plus rows with headers means that a valid table structure now exists. Now you can add a measure to the view to see actual data.

4. Drag the Sales measure to Text on the Marks card.

Tableau aggregates the measure as a sum.

Tableau uses text as the mark type. Each cell in the table displays the sum of sales for a particular year and sub-category.

We can see that the chairs and phones sub-categories had the highest sales in every year.

5. Drag the Region dimension to Rows and drop it to the left of Sub-Category. A small triangle will appear to indicate that the new field will be inserted to the left of the existing field.
The view now breaks out sales by region, in addition to year and sub-category.

Regions are listed alphabetically. You can drag Region to the right of Sub-Category to organize the view first by sub-category, and then by region.
You can use a table calculation to show percentages of total instead of raw dollar values. First, you must determine how to frame the calculation.

In this case, there are three dimensions in the view: Order Date, Sub-Category, and Region.

You could show percentages of total for a single dimension, but that can be unwieldy. For example, if you show percentages just by region, the percentages would be calculated across the two remaining dimensions: Sub-Category (there are 17 sub-categories) and Year(Order Date) (there are 4 years). So you would be dividing the total $17 \times 4 = 68$ ways. That would make for some tiny percentages.

Instead, show percentages using two dimensions: Year(Order Date) and Region. Then the percentages are calculated on the remaining dimension, Sub-Category, that is, you calculate percent of total within each highlighted area shown below.
The dimensions that you use to frame your calculation are called the **addressing fields**, and the fields in which you run your calculation are the **partition fields**.

For more information about these concepts, see [The basics: addressing and partitioning](#) on page 1525.

6. To create a table calculation to show percentages, right-click (control-click on Mac) the **SUM(Sales)** field on the **Marks** card, and then select **Add Table Calculation**.
7. In the Table Calculation dialog box, set **Calculation Type** to **Percent of Total**.

The options in the dialog box change depending on the type of calculation you choose.
For more information about using table calculations, see Transform Values with Table Calculations on page 1524.

8. For the Calculation definition, select Pane (Down), and then close the Table Calculation dialog box.

Now we see percentages calculated within each sub-category, duplicated for each year within each region. The numbers within each highlighted area add up to 100%.
**Pane (Down)** is the appropriate choice because it specifies that the calculation should be performed from top to bottom within each pane of the table. The table has two vertical dimensions, so **Table (Down)** would have calculated the percent of total from top to bottom for the entire table, ignoring the **Region** dimension.

The pane is always the finest level of detail for the relevant direction (across or down). If you had three dimensions on the vertical axis, you might have had to use field names to define the calculation, because only the dimension furthest to the left on the **Rows** shelf (defined as Table) and the dimension furthest to the right (defined as Pane) could be captured with the structural options.
Check your work! Watch steps 1-8 below:

**Build a Treemap**

Use treemaps to display data in nested rectangles. You use dimensions to define the structure of the treemap, and measures to define the size or color of the individual rectangles. Treemaps are a relatively simple data visualization that can provide insight in a visually attractive format.

The basic building blocks for a treemap are as follows:

<table>
<thead>
<tr>
<th><strong>Mark type:</strong></th>
<th>Automatic or Square</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Color:</strong></td>
<td>Dimension or Measure</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>Measure</td>
</tr>
<tr>
<td><strong>Label or Detail:</strong></td>
<td>Dimension(s)</td>
</tr>
</tbody>
</table>

To create a treemap that shows aggregated sales totals across a range of product categories, follow the steps below.
1. Connect to the **Sample - Superstore** data source.

2. Drag the **Sub-Category** dimension to **Columns**.

   A horizontal axis appears, which shows product categories.

3. Drag the **Sales** measure to **Rows**.

   Tableau aggregates the measure as a sum and creates a vertical axis.

   Tableau displays a bar chart—the default chart type when there is a dimension on the **Columns** shelf and a measure on the **Rows** shelf.

4. Click **Show Me** on the toolbar, then select the treemap chart type.

   Tableau displays the following treemap:
In this treemap, both the size of the rectangles and their color are determined by the value of Sales—the greater the sum of sales for each category, the darker and larger its box.

5. Drag the Ship Mode dimension to Color on the Marks card. In the resulting view, Ship Mode determines the color of the rectangles—and sorts them into four separate areas accordingly. Sales determines the size of the rectangles:

6. Try another option to modify the treemap: click the Undo button to remove Ship Mode
from view.

7. Drag the **Profit** measure to **Color** on the **Marks** card. Now **Profit** determines the color of the rectangles, and **Sales** determines their size:

With treemaps, **Size** and **Color** are the crucial elements. You can place measures on **Size** and **Color**, but placing a measure anywhere else has no effect. Treemaps can accommodate any number of dimensions, including one or even two on **Color**. But beyond that, adding dimensions only breaks the map into an ever greater number of smaller rectangles. For information on how to place two dimensions on **Color** in a treemap, see **Example – Multiple Fields on Color** on page 1156.
Check your work! Watch steps 1-7 below:

Build a Combination Chart

Combination charts are views that use multiple mark types in the same visualization. For example, you may show sum of profit as bars with a line across the bars showing sum of sales. You can also use combination charts to show multiple levels of detail in the same view. For example, you can have a line chart with individual lines showing average sales over time for each customer segment, then you can have another line that shows the combined average across all customer segments.

To create a combination chart, follow the steps below:

1. Open Tableau Desktop and connect to the Sample - Superstore data source.
2. Navigate to a new worksheet.
3. From the Data pane, under Dimensions, drag Order Date to the Columns shelf.
4. On the Columns shelf, right-click YEAR(Order Date) and select Month.
5. From the **Data** pane, under Measures, drag **Sales** to the **Rows** shelf.

6. From the **Data** pane, under Measures, drag **Profit** to the **Rows** shelf and place it to the right of SUM(Sales).

7. On the **Rows** shelf, right-click **SUM(Profit)** and select **Dual-Axis**.
The view updates to look like the following. Measure Names is added to Color on the Marks card to differentiate the lines.
**Note**: Some marks can be hidden behind others. To move the marks forward or backward, right-click one of the axes in the visualization and select Move Marks to Back or Move Marks to Front.

8. On the SUM(Profit) Marks card, click the Mark Type drop-down and select **Bar**.

9. In the visualization, right-click the **Profit** axis and select **Synchronize Axis**.
The visualization updates to look like the following.
Build Advanced Chart Types

The topics in this section illustrate how to create views that include industry-standard analytical benchmarks and visualization types.

**Note:** Most of these examples can only be built in Tableau Desktop—not as you edit views on the web. Before attempting to follow the steps in one of these topics, check the Applies To line at the top of the topic to determine whether the procedure is valid for editing on the web. If Tableau Server and Tableau Online are not listed, then the procedure will not work on the web.

Add a Calculated Column to a View

Sometimes, columns in your text table do not allow you to display the results of certain calculations as you might expect. In cases like this, you can create a calculated column that uses a single formula that automatically adjusts the value for each row in the table. For example, suppose you want to create a view that displays the sales for each year in several columns and the year-over-year (YOY) percentage change in the final column. There are different ways to get the result you want, but a fairly straightforward approach is to create custom calculations use them in the view.

**Objective**

To create a view like the following, that shows sales results for two years in the first two columns, and then the year-over-year change, as a percentage, in the third column.
How to Build It

The scenario uses the **Sample - Superstore** data source provided with Tableau Desktop to show how to build the view shown above.

Create the necessary calculated fields

1. Connect to the **Sample - Superstore** data source.

2. Choose **Analysis > Create Calculated Field** to open the calculation editor. Name the calculation **2013** and type or paste the following in the formula area:

   \[
   \text{IF \ YEAR([Order Date]) = 2013 \ THEN \ [Sales] \ ELSE \ 0 \ END}
   \]

3. Create a second calculated field and name it **2014**; the formula is the same except with
2014 instead of 2013:

$$\text{IF \ YEAR([\text{Order Date}]) = 2014 \ \text{THEN} \ [\text{Sales}] \ \text{ELSE} \ 0 \ \text{END}}$$

4. Create a third calculated field, **YOY Pct. Change**, to calculate the change from 2013 to 2014:

$$\frac{\text{SUM([2014])} - \text{SUM([2013])}}{\text{SUM([2013])}}$$

Format the calculated fields

1. Click **2013** in the Measures area of the Data pane and choose Default Properties > Number Format.

2. In the Default Number Format dialog box, set the format to **Currency (Custom)** and the **Decimal places** to 0.

3. Format the **2014** measure the exact same way.

4. Format the **YOY Pct. Change** field as **Percentage**, with 2 decimal places.

Build the view

1. Drag **2013** to Text on the Marks card.

2. Double-click **2014** and then **YOY Pct. Change**.
3. Drag **Measure Names** from **Rows** to **Columns**.

4. Drag **Sub-Category** to **Rows**.

   Your view should now look like the one at the beginning of this topic.

**Calculate Z-scores**

In statistics, the z-score (or standard score) of an observation is the number of standard deviations that it is above or below the population mean.

To calculate a z-score you must know the population mean and the population standard deviation. In cases where it is impossible to measure every observation of a population, you can estimate the standard deviation using a random sample.

Create a z-score visualization to answer questions like the following:

- What percentage of values fall below a specific value?
- What values can be considered exceptional? For example, in an IQ test, what scores represent the top five percent?
- What is the relative score of one distribution versus another? For example, Michael is taller than the average male and Emily is taller than the average female, but who is relatively taller within their gender?

As a general rule, z-scores lower than -1.96 or higher than 1.96 are considered unusual and interesting. That is, they are statistically significant outliers.

This article demonstrates how to calculate a z-score in Tableau.

1. Connect to the **Sample - Superstore** data source provided with Tableau Desktop.

2. Create a calculated field to calculate average sales.

   Choose **Analysis > Create Calculated Field** to open the calculation editor. Name the calculation **Average Sales** and type or paste the following in the formula area:

   ```
   WINDOW_AVG(SUM([Sales]))
   ```

3. Create another calculated field to calculate the standard deviation. Name the calculation **STDEVP Sales** and type or paste the following in the formula area:

   ```
   WINDOW_STDEVP(SUM([Sales]))
   ```

4. Create one more calculated field, this one to calculate the z-score. Name the calculation
**Z-score** and type or paste the following in the formula area:

\[
\frac{\text{SUM}([\text{Sales}]) - [\text{Average Sales}]}{\text{STDEVP Sales}}
\]

5. Drag **Z-Score** from the **Data** pane to **Columns** and **State** to **Rows**.

   Notice that the **Z-score** field on Columns has a table calculation icon on the right side (that is, a small triangle):

   ![Z-score icon](image)

   The **STDEVP Sales** function is based on the **WINDOW_STDEVP** function, which is a table calculation function. The **Z-Score** function, in turn, is a table calculation function because it includes **STDEVP Sales** in its definition. When you use a calculated field that includes a table calculation function in a view, it’s the same as adding a table calculation to a field manually. You can edit the field as a table calculation. In fact, that’s what you do next.

6. Click the **Z-score** field on Columns and choose **Compute Using > State**.

   This causes the z-scores to be computed on a per-state basis.

7. Click the **Sort Descending** icon on the toolbar:

   ![Sort Descending icon](image)

8. Hold down the Ctrl key and drag the **Z-score** field from **Columns** to **Color**.

   Ctrl + Drag copies a field as currently configured to an additional location.

9. Ctrl + Drag **Z-score** from Columns once again. This time drop it on **Label**.

   You now have a distribution of z-scores broken out by state. California and New York both have z-scores greater than 1.96. You could conclude from this that California and New York have significantly higher average sales than other states.
Visualize Key Progress Indicators

This article shows how to create a view that shows Key Progress Indicators (KPIs). A Key Performance Indicator is a measurable value that shows how effectively a company is achieving key business objectives. At a high level, the procedure requires you to do the following:

1. Create a view that includes the field or fields (measures) you want to assess.
2. Create a calculated field that establishes the threshold that demarcates success from failure.
3. Update the view to use KPI-specific shape marks to show which values are above the threshold and which are below.

Note: When connected to Microsoft Analysis Services, any KPI calculations defined in the cube are not available in Tableau. But as the procedure below demonstrates, you
can write your own KPI calculations directly inside Tableau, and then use Tableau parameters to create highly flexible what-if KPI analysis. For more on special considerations for cube data sources, see Cube Data Sources on page 2216.

The scenario uses the Sample - Superstore data source provided with Tableau Desktop to show how to build a KPI view that displays a green check mark for any sales figure over $25,000, and a red X for any sales figure under $25,000.

Create a view that includes the field you want to assess

In this case, that field is Sales.

1. Connect to the Sample - Superstore data source.
2. From the Dimensions area of the Data pane, drag Sub-Category to Rows and Region to Columns.
3. From the Measures area of the Data pane, drag Sales to Text on the Marks card.

Create a calculated field that establishes the threshold that demarcates success from failure

1. Choose Analysis > Create Calculated Field to open the calculation editor. Name the calculation KPI and type or paste the following in the formula area

   IF SUM ([Sales]) > 25000 THEN "Above Benchmark" ELSE "Below Benchmark" END

2. Click OK.
Update the view to use KPI-specific shape marks

1. On the Marks card, select Shape from the drop-down list of views:

2. Drag the \textbf{KPI} field from the \textbf{Measures} area of the \textbf{Data} pane to \textbf{Shape} on the Marks card.

3. Click \textbf{Shape} on the Marks card to open the \textbf{Edit Shape} dialog box.

4. From the \textbf{Select Shape Palette} drop down list, choose \textbf{KPI}.
   
   Now you are ready to associate specific values for the KPI field with specific shapes.

5. Click \textbf{Above Benchmark} under \textbf{Select Data Item} and then click the green check mark in the palette.

6. Click \textbf{Below Benchmark} under \textbf{Select Data Item} and then click the red X in the palette.
   
   The Edit Shape dialog box should now look like this:
7. Click OK to close the Edit Shape dialog box.

The shapes in the view show the correct indicators. Now you just need to hide the sales numbers.

8. Drag **SUM(Sales)** on the Marks card to **Detail**.
You now have a completed view that show how individual products (sub-categories) are performing across all four regions:

![Image](image)

The view may not be terribly exciting on its own, but it would make a nice addition to a dashboard that showed other performance metrics.

**Create a Pareto Chart**

A Pareto chart is a type of chart that contains both bars and a line graph, where individual values are represented in descending order by bars, and the ascending cumulative total is represented by the line. It is named for Vilfredo Pareto, an Italian engineer, sociologist, economist, political scientist, and philosopher, who formulated what has become known as the Pareto principal. Pareto made the observation that 80% of land was typically owned by 20% of the population. Pareto extended his principle by observing that 20% of the peapods in his
garden contained 80% of the peas. Eventually, the principal was further extrapolated by others to propose that for many events, roughly 80% of the effects come from 20% of the causes. In business, for example, 80% of profits not infrequently derive from 20% of the available products.

In Tableau, you can apply a table calculation to sales data to create a chart that shows the percentage of total sales that come from the top products, and thus identify the key segments of your customer base that are most important for your business’s success.

The procedure uses the Sample - Superstore data source provided with Tableau Desktop.

Preparing for the analysis

Before starting your analysis, decide what questions you want answered. These questions determine the category (dimension) and number (measure) on which to base the analysis. In the example to follow, the question is which products (captured by the Sub-Category dimension) account for the most total sales.

At a high level, the process requires you to do the following:

1. Create a bar chart that shows Sales by Sub-Category, in descending order.
2. Add a line chart that also shows Sales by Sub-Category.
3. Add a table calculation to the line chart to show sales by Sub-Category as a Running Total, and as a Percent of Total.

The scenario uses the Sample - Superstore data source provided with Tableau Desktop

Create a bar chart that shows Sales by Sub-Category in descending order

1. Connect to the Sample - Superstore data source.
2. From the Dimensions area of the Data pane, drag Sub-Category to Columns.
3. From the Measures area of the Data pane, drag Sales to Rows.
4. Click Sub-Category on Columns and choose Sort.

In the Sort dialog box, do the following:

a. Under Sort order, choose Descending.

b. Under Sort by, choose Field.

c. Leave all other values unchanged, including Sales as the selected field and Sum.
as the selected aggregation.

d. Click **OK** to exit the Sort dialog box.

Products are now sorted from highest sales to lowest.

**Add a line chart that also shows Sales by Sub-Category**

1. From the Measures area of the Data pane, drag **Sales** to the far right of the view, until a dotted line appears:

2. Drop Sales, to create a dual-axis view. It's a bit hard to see that there are two instances of the Sales bars at this point, because they are configured identically.

3. Select **SUM(Sales) (2)** on the Marks card, and change the mark type to **Line**.
This is what the view should look like at this point:

![Image of line chart with sub-category and sales data](image)

Add a table calculation to the line chart to show sales by Sub-Category as a running total, and as a percent of total

1. Click the second copy of SUM(Sales) on Rows and choose Add Table Calculation.

2. Add a primary table calculation to SUM(Sales) to present sales as a running total. Choose Running Total as the Calculation Type.
   
   Do not close the Table Calculation dialog box.

3. Add a secondary table calculation to present the data as a percent of total.
Click **Add Secondary Calculation** and choose **Percent of Total** as the **Secondary Calculation Type**.

This is what the Table Calculation dialog box should look like at this point:

![Table Calculation Dialog Box]

4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

5. Click Color in the Marks card to change the color of the line.
The result is now a Pareto chart:
Additional information

For additional tips on how you would compare the percentage of sales with the percentage of products, or draw reference lines that help make the 80-20 principles more apparent, see the Tableau On-Demand Training video titled Pareto.

Create a Population Pyramid

A population pyramid, also known as an age structure diagram, shows the distribution of various age groups in a population.

A common distribution often used with this type of visualization is female and male populations by age. To create a population pyramid using Tableau, first separate the population (measure)
into two groups, females and males, and then create "bins" for the age cohorts you want to represent in the population pyramid.

For example, suppose you are working with a table from the United States Census Bureau that contains population, gender, and age data.

To create a bin and divide a measure into two groups.

1. Download and open the following workbook from Tableau Public.
   2. Click **Download Workbook** in the upper-right corner and then open the workbook.

2. Select **Worksheet > Clear > Sheet**.

3. In the Data pane, right-click the **Age** field and select **Create > Bins**.

4. In the Create Bins dialog box, enter a bin size based on the age groups you're interested in focusing on, and then click **OK**.

   In this example, the bin size is 10. This means that the age cohorts are defined by increments of 10 years.

5. Drag the bin you just created to the Rows shelf.
6. Select **Analysis > Create a Calculated Field**, and then do the following:

   - Enter a name for the calculation. For this example, enter **Male Population**.
   - Enter a formula similar to the following to isolate the male constituent from population:
     
     \[
     \text{IF } \text{[Gender]} = 1 \text{ THEN } [\text{ESTBASE2010}] \text{ END}
     \]

     In this case, the census data has defined the Gender value for male as "1." The field "ESTBASE2010" contains estimated population values.

7. Similar to the step 5, select **Analysis > Create a Calculated Field**, and then do the following:

   - Enter a name for the calculation. For this example, enter **Female Population**.
   - Enter a formula similar to the following to isolate the female constituent from population:
     
     \[
     \text{IF } \text{[Gender]} = 2 \text{ THEN } [\text{ESTBASE2010}] \text{ END}
     \]

     In this case, the census data has defined the Gender value for female as "2." The field "ESTBASE2010" contains estimated population values.

8. Drag the calculated fields you created to the Columns shelf and the **Gender** field to the Color shelf.
**Note:** To change the color assignments, click the Color shelf and then click **Edit Colors**.

9. Right-click the axis for Male Population, select **Edit Axis**, and then select the check box for **Reversed** to reverse the order the values display on the axis, and then click **OK**.

After changing the sort order of the bin, the population pyramid looks like the following:
Create a Co-Occurrence Visualization Using a Parameter and a Set

You can create co-occurrence visualizations in Tableau. With a co-occurrence visualization, users can pick one field value and then see which other fields values it co-occurs with, and how often. One practical application of such analysis is a market basket analysis, which you can use to discover and understand customer purchasing behavior. You can use a market basket analysis to answer questions like the following:

- How many people bought both Product A and Product B?
- What other products do people who purchased Product A generally buy?
- Which other courses do students who enrolled in Course A frequently enroll in?

Follow the steps in this article to create a co-occurrence visualization (in this case, a market basket analysis view) using a parameter, calculated fields, and a set. At a high level, the steps involved are:

1. Create a parameter that you will use to dynamically modify the view based on the item you select.
2. Create calculated fields that you will use to return which items are also ordered when a
particular item is ordered.

3. Create a set to determine whether an order has the item that was selected in the parameter control.

4. Build the view to display which items are also contained in an order with the selected item.

The scenario uses the Sample - Superstore data source provided with Tableau Desktop. The details are covered in the following sections.

Create a Parameter

Follow these steps to create a parameter and then display a parameter control that users can use to pick a Sub-Category value.

1. Connect to the Sample - Superstore data source.

2. Right-click (control-click on a Mac) in the Data pane and select Create > Parameter.

3. In the Create Parameter dialog box, do the following:
   a. Name the parameter Order Contains.
   b. For the Data Type, choose String.
   c. For Allowable Values, choose List.
   d. In the List of values section click Add from Field > Sub-Category.
   e. Click OK.

4. Right-click the Order Contains parameter in the Data pane and choose Show Parameter Control.

Step 2: Create Calculated Fields

Follow these steps to create the calculated fields that you will use to show which items are also ordered when the item selected by the user (via the parameter control) is ordered.

1. Create a calculated field to identify products that the order also contains (in addition to the one that the user selects).

   Choose Analysis > Create Calculated Field to open the calculation editor. Name the calculated field Then Order Also Contains and type or paste the following in the formula area:
IF [Sub-Category] <> [Order Contains] THEN [Sub-Category] END

You may have to replace &lt; &gt; with <> after you paste.

2. Create another calculated field to identify matching products.

   Name the field **Product Matches** and type or paste the following in the formula area:

   IF [Sub-Category] = [Order Contains] THEN 1 END

Create a Set

1. Now create a set to determine whether an order has the item that was selected in the parameter control.

2. Hover the cursor over the **Order ID** dimension in the Data pane, click the down arrow at the right end of the field, and choose **Create > Set.**
3. In the Create Set dialog box, type **Order Has Selected Product** in the **Name** text box.

4. Go to the Condition tab, select **By field**, and in the drop-down lists make the following selections and entries to build the condition:
   a. In the first drop-down list, select **Product Matches**.
   b. In the second drop-down list, select **Sum**.
   c. In the next drop-down list, select **>=**.
   d. In the last text box, type **1**.
   e. Click **OK**.

5. Click **OK**.
Build the View

Finally, build the view to display which items are also contained in an order with the selected item.

1. Drag Then Order Also Contains to Columns.
2. Drag Order ID to Rows.

   In the warning dialog box, click Add all members.
3. Click the Order ID field on Rows and choose Measure > Count (Distinct) to change the aggregation.
4. Right-click (control-click on a Mac) the Null bar on the x-axis and choose Exclude.
5. Drag the Order Has Selected Product set to the Filters shelf.
6. Press Ctrl+W to swap the fields on Rows and Columns.

You (or your users) can now use the Order Contains parameter control to select an item in an order, and then see a bar chart showing which other items are also included in orders with the selected item.

Visualize Benford's Law

Benford’s Law is a mathematical law that states that the leading, or left-most, digit in many real-life data sources is distributed in a very specific manner. Specifically, the number 1 occurs as the leading digit about 30% of the time, and as numbers get larger they occur less frequently, with the number 9 occurring less than 5% of the time. When fraudsters are fabricating data,
they may not know to create fake data that conforms to Benford's law, and in some cases this makes it possible to detect fake data or at least to create doubt about its veracity.

This article describes how to apply Benford’s Law to Sales data, using the Sample - Superstore data source provided with Tableau Desktop.

The process requires you to do the following:

1. Create calculated fields to use in your view
2. Set up the view

The following sections break these procedures down into specific instructions.

Create calculated fields to use in your view

Follow these steps

1. Choose Analysis > Create Calculated Field to open the calculation editor. Name the calculation Leftmost Integer and type or paste the following in the formula area:

\[ \text{LEFT}(	ext{STR}([\text{Sales}]),1) \]

2. Create a second calculated field and name it Benfords Law. Type or paste the following in the formula area:

\[ \text{LOG}(	ext{INT}([\text{Leftmost Integer}])+1) - \text{LOG}(	ext{INT}([\text{Leftmost Integer}])) \]

Set up the view

Follow these steps.

1. Drag Leftmost Integer from the Dimensions area of the Data pane to Columns, and then drag Number of Records from the Measures area of the Data pane to Rows.
2. Click \text{SUM(Number of Records)} on Rows and choose Quick Table Calculation > Percent of Total.

Your view now shows the distribution of first digits, and the size of the bars (decreasing from left to right) suggests that the data in this case conforms to Benford's law. But we can do more to frame the data by adding reference distributions.

3. Drag Benfords Law from the Measures area of the Data pane to Detail on the Marks card, and then click Benfords Law on the Marks card and choose Measure > Minimum.
4. Switch from the **Data** pane to the **Analytics** pane and drag **Distribution Band** into the view. Drop it on **Cell**.

![Distribution Band in Tableau](image)

**Note**: Distribution Bands are supported on web platforms starting with Tableau 10.2.

5. In the Edit Reference Line, Band, or Box dialog box, do the following:
a. Click in the Value field to view an additional set of options:

![Image of Edit Reference Line, Band, or Box dialog box]

b. In the Percentages area, type 80, 100, 120.

   This specifies that you want bands spanning from 80 to 100 percent, and from 100 to 120 percent. Next you will specify what value the percentages are referencing.

c. In the Percent of field, choose MIN(Benfords Law).

   The Value field should now read 80%, 100%, 120% of Average Min. Benfords Law.

   The remaining steps configure the appearance of the reference bands.

d. Set Label to None.

e. Set Line to the thinnest available line.

f. Choose Fill Below.

g. From Fill, select Stoplight.

h. Click OK to exit the Edit Reference Line, Band, or Box dialog box.

6. Click the toolbar button to display mark labels:
Even though Superstore is demo data, it's realistic as far as conforming to Benford's law. The blue bars that indicate actual percentages of initial digits align very well with the 100% value (that is, the line that separates the green zone from the yellow zone in the distribution bands) that shows expected Benford values in the view.

Create Bins from a Continuous Measure

Sometimes it's useful to convert a continuous measure (or a numeric dimension) into bins.
Any discrete field in Tableau can be considered as a set of bins. For example, suppose you create a view with Profit on Rows and State on Columns. You could consider the State field as a set of bins—each profit value is sorted into a bin corresponding to the state from which the value was recorded. But if you want to see values for Profit assigned to bins without reference to a dimension, you can create a numeric bin, with each individual bin corresponding to a range of values.

**Note:** You can bin data only for relational data sources and binned fields cannot be used in calculations. However, it is possible to create a calculated field that will replicate a bin of a specific size. For example: (FLOOR([Sales]/1000)*1000) will create bins with a size of 1000. By dragging this calculation to the dimensions pane, you can use these bins with cube data sources and calculated fields.

When you create bins from a measure you create a new dimension. That's because you are creating a field with a limited and discrete set of possible values out of a field with an unlimited, continuous range of values. However, once the dimension is created, you can convert it to a continuous dimension. This can be useful, for example, if you want to create a histogram. See **Create a Histogram from a Binned Dimension** on page 960.

**Create a Binned Dimension:**
1. In the **Data** pane, right-click (control-click on Mac) a measure and select **Create > Bins**.

![Data pane screenshot showing a right-click menu with 'Create > Bins' option highlighted]

2. In the Create Bins dialog box, accept the proposed New field name or specify a different name for the new field.

![Create Bins dialog box with 'Sales' field and '4+7' bin size selected]

On the web, the dialog box is named Edit Bins and has a slightly different appearance, but the options are the same.
3. Either enter a value in the **Size of bins** field or have Tableau calculate a value for you.

If Tableau can perform the optimizing calculation quickly enough, the value you see initially in **Size of bins** is Tableau’s estimate of the optimal bin size.

If Tableau cannot perform the optimizing calculation quickly, the **Size of bins** field defaults to 10. In this case you can click **Suggest Bin Size** to have Tableau perform the optimizing calculation.

The formula that Tableau uses to calculate an optimal bin size is

\[
\text{Number of Bins} = 3 + \log_2(n) \times \log(n)
\]

In the formula, \( n \) is the number of distinct rows in the table. The size of each bin is determined by dividing the difference between the smallest and the largest values by the number of bins.

The four read-only fields in the lower part of the Create Bins dialog box show you the data that Tableau uses to suggest a bin size. You can also consider these values if you want to set a bin size manually. The values are:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>The field's minimum value.</td>
</tr>
<tr>
<td>Max</td>
<td>The field's maximum value.</td>
</tr>
<tr>
<td>Diff</td>
<td>The difference between the field's minimum and maximum values.</td>
</tr>
<tr>
<td>CntD</td>
<td>The number of distinct values (rows) in the data.</td>
</tr>
</tbody>
</table>

After you click **OK** to dismiss the Create Bins dialog box, a new binned field appears in the **Dimensions** area of the **Data** pane.
When you add a binned dimension to the view, each bin acts as an equal-sized container that summarizes data for a specific range of values. Column or row headers are created, where each bin label designates the lower limit of the range of numbers that is assigned to the bin. Note that the lower limit is inclusive.

Create a Histogram from a Binned Dimension

If you create a binned dimension, you can use it as the starting point for creating a histogram. Using the **Sales (bin)** dimension created according to the instructions above, use the following steps to create a histogram.

**Note:** A quicker way to create a histogram is using Show Me. See Build a Histogram on page 881 in the Build-It-Yourself Exercises section for information on creating a histogram using Show Me.

1. Click the **Sales (bin)** dimension in the Data pane and choose **Convert to continuous**.
2. Drag the **Sales (bin)** dimension from the Data pane and drop it on the **Columns** shelf.
3. Drag the original **Sales** field from the Measures area of the Data pane and drop it on the
4. Click **SUM(Sales)** on **Rows** and change the aggregation from Sum to Count.

The result is a histogram:

Build and Explore Data Views

This section describes the various features at your disposal and tasks related to building visualizations in Tableau.

For a 6-minute walkthrough on building data views, see the [Getting Started with Visual Analytics](#) free training video. Use your [tableau.com](http://tableau.com) account to sign in.

For an in-depth, 25-minute walkthrough of the Tableau environment, see the [Getting](#)
Organize and Customize Fields in the Data Pane

The Data pane has many features and functions to help you organize and customize your data fields, find fields, and hide fields.

In this article

- Organize fields in folders
- Sort fields
- Find fields
- Rename fields
- Hide or Unhide fields
- Add calculated fields to the Data pane

Organize the Data Pane

You can reorganize the items in the Data pane from its default layout using folders or through sorting.

Group fields in folders

To make data sources with many fields easier to work with, you can organize the Data pane items into folders. Items like fields, parameters, and sets can be grouped into folders.

Click the **Group by Folder** option in the Data pane menu, or in a field's context menu.
Notes on grouping options

- When you connect to a single table in your data source, grouping by folder is enabled by
default.

- When you connect to a data source with multiple tables, grouping by table is enabled.

- When the **Group by Data Source Table** option is selected, the dimensions and measures are grouped according to the database table they belong to. This is especially useful when you have several joined tables.

- "Group by" options are only available for relational data sources—not for multidimensional (cube) data sources.

**To group fields into folders**

1. In the Data pane, select the fields you want to group together or right-click (control-click on Mac) an empty area in the Data pane.

2. Click **Folders > Create Folder**.

3. When prompted, name the new folder.
After you create a folder structure, you can drag fields from one folder to another or duplicate a field you want to have available in more than one folder.

To add a field to a folder

- Drag a field on top of the folder name to add the field to the folder. If the folder is expanded, you can drag the field into the general area of the folder.
Sort fields in the Data pane

When organizing the Data pane with or without folders, you can have Tableau sort the items. These Sort by options are also located in the Data pane menu.
Sort options are only available for relational data sources—not for multidimensional (cube) data sources. You can sort by one of the following options:

- **Sort by Name** – lists the dimensions and measures in alphabetical order according to their field aliases.
- **Sort by Data Source Order** – lists the dimensions and measures in the order they are listed in the underlying data source.

---

**Find fields**

You can search for fields, folders and hierarchies in the Data pane. When there are many fields in your data source it can be difficult to find a specific one like “Date” or “Customer” or “Profit,” or to find all fields that end in "xyz." To find an item, do the following:

Click the **Find Field** icon at the top of the Data pane (Ctrl + F in Windows, Command-F on a Mac) and enter the name of the item you want to search for. You can also enter a string of characters, to search for all item names that contain that string.

As you type in the search box, search filters the contents of the Data pane to show all fields, folders or hierarchies that contain the typed string.
Search remains open until you click the **Find Field** icon or press **Ctrl + F** again.

---

## Rename Fields

You can rename fields in the Data pane. For example, you could rename a field named **Customer Segment** in the data source to be **Business Segment** in Tableau. You can also rename user-created fields. Renaming a field does not change the name of the field in the underlying data source, rather it is given a special name that appears only in Tableau workbooks. The changed field name is saved with the workbook as well as when you export the data source. You can rename any type of field: dimensions, measures, sets, or parameters.

### Rename a field

1. Click field name in the Data pane and hold the mouse button down until the field name is shown in an edit box:
You can also press F2 or Ctrl + Enter to make the field name editable.

2. Type the new name for the field and press Enter.

The field displays with the new name in the Data pane.

Revert to the Default Field Name

If the field you renamed was from the original data source you can click the field name in the Data pane and hold the mouse button down until the field name is shown in the box. At the right of the edit field is a small circular arrow that you can click to restore the original data source field name:

To revert the names of multiple fields that were in the original data source, select them all, right-click, and then choose **Reset Names**.

---

**Combine Fields**

Combine fields to create a cross product of members from different dimensions. You can combine dimensions if you want to encode a data view using multiple dimensions.

To combine the fields, select multiple dimensions in the Data pane and then right-click (control-click on a Mac) the fields and select **Create > Combined Field**.
**Note:** For cube (multidimensional) data sources, you must select levels from different hierarchies. In Tableau, cube data sources are supported only in Windows.

For example, the selections shown below will produce a new field that consists of the Category and Sub-Category dimensions.

The two dimensions are combined into a new dimension. The name of the field is automatically created from the names of the original fields. Right-click (control-click on a Mac) the new field and select **Rename** to change the name.

When you use the new field in a view, a header is created for each combination of the two original dimensions. For example, the view below shows the members of the combined Category and Sub-Category fields.
### Hide or Unhide Fields

You can selectively hide or show fields in the Data pane. To hide a field, right-click (control-click on a Mac) the field and select **Hide**.

When you want to change your fields from hidden to visible, select **Show Hidden Fields** on the Data pane menu.

---

**Note:** For cube data sources, to choose to display the fully qualified name, right-click (control-click on a Mac) the combined field in the Data pane and select **Qualify Member Names**.
The hidden fields are then shown in gray in the Data pane. You can then select one or more hidden fields, right-click (control-click on a Mac) and select **Unhide**.

Select **Hide All Unused Fields** on the Data pane menu to quickly hide all of the fields that are not being used in the workbook.

**Note:** Hiding fields can be a good way to decrease the size of a data extract file because hidden fields are automatically excluded from the extract.
Add Calculated Fields to the Data Pane

You can create calculated fields that appear in the Data pane. These new computed fields can be used like any other field. Select Create Calculated Field on the Data pane menu. Alternatively, select Analysis > Create Calculated Field.

Edit Default Settings for Fields

When you drag fields to shelves, the data is represented as marks in the view. The fields and their marks are displayed initially based on their default settings. You can control these default settings by clicking the drop-down arrow on a field.

The Default Properties menu includes default settings for aggregation, comments, number formatting, color, shape, and totals (based on the type of field).

In this article

- Set the default aggregation for a measure
- Add comments for specific fields
- Set the default number format
- Set the default color
- Set the default shape
- Set the default sort order for the values within a categorical field

Set the default aggregation for a measure

You can specify a default aggregation for any measure. The default aggregation will be used automatically when the measure is first totaled in the view.

1. Right-click (control-click on a Mac) any measure in the Data pane and select Default Properties > Aggregation.
2. On the Aggregation list, select an aggregation.
Whether you are specifying the aggregation for a field on a shelf or the default aggregation in the Data pane, you can select from several aggregations. See Data Aggregation in Tableau on page 279 to learn about each type of aggregation.

Add default comments for specific fields

Fields can have comments that describe them. The comments display in a tooltip in the Data pane and in the Calculated fields dialog box. Field comments are a good way to give more context to the data in your data source. Comments are especially useful when you are building a workbook for others to use.

To add a default comment for a field

1. Right-click (control-click on a Mac) a field in the Data pane and select Default Properties > Comment.
2. Write a comment in the subsequent dialog box. Comments support rich text formatting that will be represented in the tooltip.

3. When finished, click OK.

Now when you hover the cursor over the field in the Data pane, you see the comment.

Set the default number format

You can set the default number format for date and number fields. For example, you may want to always show the Sales values as currency using the U.S. dollar sign and two decimal places. Or you may want to always show Discount as a percentage.

To set the default formats, right-click (control-click on Mac) a date or number field and select either Date Format or Number Format on the Default Properties menu. A dialog box opens where you can specify a default format.
Set the default color

When you use a dimension to color encode the view, default colors are assigned to the field’s values. Color encodings are shared across multiple worksheets that use the same data source to help you create consistent displays of your data. For example, if you define the Western region to be green, it will automatically be green in all other views in the workbook. To set the default color encodings for a field, right-click (control-click on Mac) the field in the Data pane and select Default Properties > Color.

For information about color properties, and how to configure and customize colors in Tableau, see Color Palettes and Effects on page 1145,

Set the default shape

When you use a dimension to shape encode the view, default shapes are assigned to the field’s values. Shape encodings are shared across multiple worksheets that use the same data source to help you create consistent displays of your data. For example, if you define that Furniture products are represented with a square mark, it will automatically be changed to a square mark in all other views in the workbook.

To set the default shape encodings for a field, right-click (control-click on Mac) the field in the Data pane and select Default Properties > Shape.
Set the default sort order for the values within a categorical field

You can set a default sort order for the values within a categorical field so that every time you use the field in the view, the values will be sorted correctly. For example, let’s say you have an Order Priority field that contains the values High, Medium, and Low. When you place these in the view, by default they will be listed as High, Low, Medium because they are shown in alphabetical order. You can set a default sort so that these values are always listed correctly.

To set the default sort order, right-click (control-click on a Mac) a dimension and select Default Properties > Sort. Then use the sort dialog box to specify a sort order.

**Note:** The default sort order also controls how the field values are listed in a filter in the view.

Create Aliases to Rename Members in the View

You can create aliases (alternate names) for members in a dimension so that their labels appear differently in the view.

Aliases can be created for the members of discrete dimensions only. They cannot be created for continuous dimensions, dates, or measures.

**Note:** When using a published data source, you cannot create or edit aliases.

To create an alias:
1. In the Data pane, right-click a dimension and select **Aliases**.

2. In the Edit Aliases dialog box, under **Value (Alias)**, select a member and enter a new name.
Tip: To reset the member names back to their original names, click **Clear Aliases**.

3. To submit your changes:
   - In Tableau Desktop, click **OK**.
   - On Tableau Server or Tableau Online, click the X icon in the top-right corner of the dialog box.

When you add the field to the view, the alias names appear as labels in the view. For example,
Convert Fields between Discrete and Continuous

You can convert measures from discrete to continuous or from continuous to discrete. And you can convert date dimensions and other numeric dimensions to be either discrete or continuous.

Convert measures

You can convert measures from discrete to continuous or from continuous to discrete. Click the field and choose Discrete or Continuous. The field is green when it is continuous, and blue when it is discrete.

For measures in the Data pane, right-click the field and choose Convert to Discrete or Convert to Continuous. The color of the field changes accordingly.

Convert date fields

You can convert Date fields between discrete and continuous. Click any Date field in the view and choose one of the options on the context menu to change it from discrete to continuous or from continuous to discrete:
Click any of the options in the blue areas to configure the field as a discrete date. Choosing one of these options creates what is known as a "date part."

Click any of the option in the green areas to configure the field as a continuous date. Choosing one of these options creates what is known as a "truncated date."

To convert a Date field in the Data pane (and thus to determine the default result when you drag it into a view), right-click the field and choose Convert to Discrete or Convert to Continuous.

Convert a Measure to a Dimension

You can convert a field from a measure to a dimension in the current view. Or, if you want the change to affect all future uses of the field in the workbook, you can convert a field in the Data
pane from a measure to a dimension.

Convert a measure in the view into a discrete dimension

You can drag a field from the **Measures** area in the **Data** pane but then use it as a dimension in the view. For example, suppose you want to know the aggregated sales totals for each possible discount rate. The view you are aiming for looks like this:

The **Discount** field contains numeric data, so when you connect to the data source, Tableau assigns it to the **Measures** area in the **Data** pane. In the **Sample - Superstore** data source, which is included with Tableau Desktop, the values for **Discount** range from 0% to 80%.

Here are the steps to create the view shown above:

1. Drag **Sales** to **Rows** and **Discount** to **Columns**. Tableau shows you a scatter plot—this is the default chart type when you put one measure on **Rows** and another on **Columns**.
Tableau aggregates **Discount** as AVG, and **Sales** as SUM. The fields are both continuous, so along the bottom and left side of the view Tableau displays axes (and not column or row headers).

2. To treat **Discount** as a dimension, click the drop-down arrow on the field (on the **Columns** shelf) and select **Dimension** from the context menu. Tableau no longer aggregates the values for **Discount**, so what you see now is a line. But the values for **Discount** are still continuous, so Tableau still shows continuous axes for both fields:
3. To complete the process, click the drop-down arrow on the **Discount** again and select **Discrete** from the context menu. The transformation of **Discount** is now complete. You now see the bar chart as in the initial image at the beginning of this topic. Across the bottom, you now see column headers (0%, 10%, 20%, etc.) instead of an axis.

Let’s review how we got to this point:

<table>
<thead>
<tr>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convert <strong>Discount</strong> from a measure to a dimension...</td>
<td>Sales values are no longer aggregated according to discount rate, resulting in a line chart instead of a scatter plot.</td>
</tr>
<tr>
<td>Convert <strong>Discount</strong> from continuous to discrete...</td>
<td>Tableau shows headers at the bottom of the view, instead of a continuous axis.</td>
</tr>
</tbody>
</table>

The only thing left to do is to drag **Sales** to Label and then format the labels for readability.

The resulting chart is somewhat useful because there are only 12 unique values for **Discount** in the data source. Had there been a unique value for each row, which would not have been unusual for a numeric field, the number of individual bars in the resulting view would have been equal to the number of rows in the data source, which would probably not result in a useful visualization.
Converting a measure in the Data pane into a dimension

When you first connect to a data source, Tableau assigns most fields that contain quantitative, numerical information (that is, fields where the values are numbers) to the Measures area in the Data pane. The exception is for fields where the name suggests the data type, such as Year or Month (which Tableau would identify as Date dimensions) or fields containing words like “ID” and “Key,” which Tableau would categorize as dimensions, even when they are numeric.

However, you might decide that some of these fields that Tableau has categorized as measures should actually be dimensions. Postal codes are a classic example—they often consist entirely of numbers, but the information is categorical and not continuous—you would never want to aggregate postal codes by adding or averaging them. Similarly, a field containing individuals’ ages may be categorized as a measure by default in Tableau because it contains numeric data. In some cases you may want to add or average ages, but you might also want to look at each individual age as a bin or category, in which case you want Tableau to create headers for this field rather than an axis. If this is how you want to use age in your view, you can convert the field to a dimension.

To convert a measure to a dimension in the Data pane, do either of the following.

- Click and drag the field from the measures area of the Data pane and drop it into the dimensions area.
Right-click (control-click on a Mac) the measure in the Data pane and select **Convert to Dimension**.

If you place a field that you converted from a measure to a dimension on a shelf, it now produces headers instead of an axis.
Create Hierarchies

When you connect to a data source, Tableau automatically separates date fields into hierarchies so you can easily break down the viz. You can also create your own custom hierarchies. For example, if you have a set of fields named Region, State, and County, you can create a hierarchy from these fields so that you can quickly drill down between levels in the viz.

Watch a video: To see related concepts demonstrated in Tableau, watch Drill Down and Hierarchies, a 5-minute free training video. Use your tableau.com account to sign in.

In this article

Create a hierarchy below
Drill up or down in a hierarchy on page 989
Remove a hierarchy on page 990

Create a hierarchy

To create a hierarchy:
1. In the **Data** pane, drag a field and drop it directly on top of another field.

![Data pane with fields]

**Note:** When you want to create a hierarchy from a field inside a folder, right-click (control-click on a Mac) the field and then select **Create Hierarchy**.

2. When prompted, enter a name for the hierarchy and click **OK**.

![Create Hierarchy dialog box]

3. Drag any additional fields into the hierarchy. You can also re-order fields in the hierarchy by dragging them to a new position.
Drill up or down in a hierarchy

When you add a field from a hierarchy to the visualization, you can quickly drill up or down in the hierarchy to add or subtract more levels of detail.

To drill up or down in a hierarchy:
In the visualization, click the + or - icon on the hierarchy field.

Remove a hierarchy

To remove a hierarchy:

- In the Data pane, right-click (control-click on a Mac) the hierarchy and select Remove Hierarchy.

The fields in the hierarchy are removed from the hierarchy and the hierarchy disappears from the Data pane.

Group Your Data

You can create a group to combine related members in a field. For example, if you are working with a view that shows average test scores by major, you might want to group certain majors together to create major categories. English and History might be combined into a group called Liberal Arts Majors, while Biology and Physics might be grouped as Science Majors.

Groups are useful for both correcting data errors (e.g., combining CA, Calif., and California into one data point) as well as answering "what if" type questions (e.g., "What if we combined the East and West regions?").
Create a group

There are multiple ways to create a group. You can create a group from a field in the Data pane, or by selecting data in the view and then clicking the group icon.

Create a group by selecting data in the view

1. In the view, select one or more data points and then, on the tooltip that appears, click the group icon

   Note: You can also select the group icon on the toolbar at the top of the workspace.

If there are multiple levels of detail in the view, you must select a level to group the members. You can select to group all dimensions, or just one.
Create a group from a field in the Data pane

1. In the Data pane, right-click a field and select Create > Group.

2. In the Create Group dialog box, select several members that you want to group, and then
click **Group**.

The selected members are combined into a single group. A default name is created using the combined member names.

To rename the group, select it in the list and click **Rename**.

**Tip**: You can search for members using the **Find** option near the bottom-right of the dialog box. (Tableau Desktop only)
Include an Other Group

When you create groups in Tableau, you have the option to group all remaining, or non-grouped members in an Other group.

The Include Other option is useful for highlighting certain groups or comparing specific groups against everything else. For example, if have a view that shows sales versus profit product category, you might want to highlight the high and low performing categories in the view, and group all the other categories into an "Other" group.

<table>
<thead>
<tr>
<th>Includes Other</th>
<th>Does not include Other</th>
</tr>
</thead>
</table>

- 994 -
To include an Other group:
1. In the Data pane, right-click the group field and select **Edit Group**.

2. In the Edit Group dialog box, select **Include 'Other'**.

---

**Edit a Group**

After you have created a grouped field, you can add and remove members from the groups, create new groups, change the default group names, and change the name of the grouped field. You can make some changes directly in the view, and others through the Edit Group dialog box.

**To add members to an existing group:**
- In the **Data** pane, right-click the group field, and then click **Edit Group**.

- In the **Data** pane, right-click the group field, and then click **Edit Group**.

  ![Diagram of a data pane with dimensions and a group field expanded]

- In the **Edit Group** dialog box, select one or more members and drag them into the group you want.

- Click **OK**.

**To remove members from an existing group:**

- In the **Data** pane, right-click the group field, and then click **Edit Group**.

- In the **Edit Group** dialog box, select one or more members, and then click **Ungroup**.

  The members are removed from the current group. If you have an Other group, the
members are added to it.

- Click **OK**.

**To create a new group in a group field:**

- In the **Data** pane, right-click the group field, and then click **Edit Group**.
- In the Edit Group dialog box, select one or more members, and then click **Group**.
- Click **OK**.

**Note:** To rename a group, select the group in the Edit Group dialog box, and then click **Rename**.

**See also**

- **Color a View Using Groups** below
- **Correct Data Errors or Combine Dimension Members by Grouping Your Data** on page 1002

**Color a View Using Groups**

In addition to correcting data errors and combining dimensions, you can visually identify groups of related marks by selecting marks directly. This technique is especially useful when working with a scatter plot or a view that doesn’t have headers that you can select when defining the group.

1. Press and hold the **CTRL** or **Shift** key on the keyboard to select one or more marks in the view. On a Mac, use the **Shift** key.

2. To create a group do one of the following:

   - For Tableau Desktop, click the **Group** button on the toolbar or right-click and select **Group**.
For Tableau Server or Tableau Online, click the Group icon on the tooltip.

The selected marks are grouped and all other members are combined into an "Other" category. The new group field is automatically added to Color. For more information about grouping fields using Other, see Include an Other Group on page 994.
Note: If there were already fields on Color, they are moved to Detail and replaced with the group field.

When you create groups by selecting marks, it’s possible that the marks will represent multiple dimensions. For example, you may have a scatter plot that shows the Sales vs. Profit by Region and Sub-Category. In this case, a selection of marks will represent members of both the Region and Sub-Category dimensions. When the selection represents multiple dimensions, the Group menu lets you choose to group on All Dimensions or on a particular dimension.
In the example above, the 5 selected marks represent a region and the following sub-category combinations.

- West, Copiers
- West, Binders
- West, Accessories
- East, Copiers
- Central, Copiers
- South, Art
- South, Envelopes
- South, Labels
- West, Envelopes
- West, Labels

The views below show the results of grouping these measures on All Dimensions, Sub-Category, and Region.

The 5 marks are combined and the rest of the marks are added to an "Other" category. All marks associated with any of the three sub-categories are combined and everything else is added.

All marks associated with the four regions are combined and everything else is added.
Correct Data Errors or Combine Dimension Members by Grouping Your Data

If you are using groups to correct data errors (for example, to combine "CA" and "California") or to combine dimensions members (for example, to combine "East" and "West" regions), the easiest option is to select headers in the view.

1. Press and hold the **CTRL** or **Shift** key on the keyboard to multi-select headers in the view. On a Mac, press and hold the Command key.

2. To create a group, do one of the following:

   - On Tableau Desktop, click the **Group** button on the toolbar, right-click and select **Group** or click the **Group** icon on the tooltip.
On Tableau Server or Tableau Online, click the **Group** icon on the tooltip.

The selected members are combined into a single member. In this example, the view now shows the SUM(Sales) across all binders, envelopes and fasteners as a single mark. A default is automatically constructed using the combined member names. The dimension on the Rows or Columns shelf is replaced with the new grouped field.
Create Sets

You can use sets to compare and ask questions about a subset of data. Sets are custom fields that define a subset of data based on some conditions.

In this article

- Create a dynamic set below
- Create a fixed set on page 1008
- Add or remove data points from sets on page 1010
- Use sets in the visualization on page 1010
- Show In/Out members in a set on page 1010
- Combine Sets on page 1012
- Examples of Sets on page 1013

Create a dynamic set

The members of a dynamic set change when the underlying data changes. Dynamic sets can only be based on a single dimension.

To create a dynamic set:
1. In the Data pane, under Dimensions, right-click a field and select **Create > Set**.

2. In the Create Set dialog box, configure your set. You can configure your set using the following tabs:

   - **General**: Use the General tab to select one or more values that will be considered when computing the set.

     You can alternatively select the **Use all** option to always consider all members even when new members are added or removed.

   - **Condition**: Use the Condition tab to define rules that determine what members to
include in the set.

For example, you might specify a condition that is based on total sales that only includes products with sales over $100,000.

![Create Set dialog box](image)

**Note**: Set conditions work the same as filter conditions. See *Filter Data from Your Views* on page 1162 to learn more.

- **Top**: Use the Top tab to define limits on what members to include in the set.
For example, you might specify a limit that is based on total sales that only includes the top 5 products based on their sales.

3. When finished, click **OK**.

   The new set is added to the bottom of the Data pane, under the Sets section. A set icon indicates the field is a set.

**Note:** Set limits work the same as Filter limits. See *Filter Data from Your Views* on page 1162 to learn more.
Create a fixed set

The members of a fixed set do not change. A fixed set can be based on a single dimension or multiple dimensions.

To create a fixed set:

1. In the visualization, select one or more marks (or headers) in the view.
2. Right-click the mark(s) and select Create Set.

3. In the Create Set dialog box, type a name for the set.

4. Optionally complete any of the following:
   
   - By default, the set includes the members listed in the dialog box. You can select the option to Exclude these members instead. When you exclude, the set will include all of the members you didn't select.

   - Remove any dimensions that you don't want to be considered by clicking the red "x" icon that appears when you hover over a column heading
- Remove any specific rows that you don't want to include in the set by clicking the red "x" icon that appears when you hover over the row X.

- If the marks you selected represent multiple dimensions, each member of the set will be a combination of those dimensions. You can specify the character that separates the dimension values. To do so, for Separate members by, enter a character of your choice.

- Select Add to Filters shelf to automatically move the set to the Filters shelf once it is created.

5. When finished, click OK.

The new set is added to the bottom of the Data pane, under the Sets section. A set icon 🗄️ indicates the field is a set.
Add or remove data points from sets

If you created a set using specific data points, you can add more data to or subtract data from the set.

**To add or remove data points from a set:**

1. In the visualization, select the data points you want to add or remove.
2. In the tooltip that appears, click the Sets drop-down icon, and then click **Add to [set name]** or **Remove from [set name]** to add or remove data from a particular set.

Use sets in the visualization

After you create a set, it displays at the bottom of the Data pane in the Sets section. You can drag it into the viz like any other field.

When you drag a set to the viz in Tableau Desktop, you can choose to show the members of the set or aggregate the members into In/Out categories.

In Tableau Server or Tableau Online you can only aggregate the members of the set into In/Out categories.

**Show In/Out members in a set**

In most cases, when you drag a set to the viz, Tableau displays the set using the In/Out mode. This mode separates the set into two categories:
- In - The members in the set.
- Out - Any members that are not part of the set.

For example, in a set defined for the top 25 customers, the top customers would be part of the In category and all other customers would be part of the Out category.

Using the In/Out mode makes it easy to compare the members in the set to everything else.

**To show In/Out members in the visualization:**

- In Tableau Desktop, right-click the set in the visualization workspace and select **Show In/Out of Set**.

When a set is in In/Out mode, the field on the shelf is prefaced by the text, "IN/OUT" followed by the set name.
Show Members in Set

As an alternative to showing the set using In/Out mode, you can list the members in the set. Showing the members in the set automatically adds a filter to the view that includes only the members of the set.

**To switch a set to list the individual members:**

- In the visualization workspace, right-click the set and select **Show Members in Set**.

**Note:** To display the fully qualified member names for cubes, right-click the set in the Data pane and select **Qualify Member Names**.

Combine Sets

You can combine two sets to compare the members. When you combine sets you create a new set containing either the combination of all members, just the members that exist in both, or members that exist in one set but not the other.

Combining sets allows you to answer complex questions and compare cohorts of your data. For example, to determine the percentage of customers who purchased both last year and this year, you can combine two sets containing the customers from each year and return only the customers that exist in both sets.

To combine two sets, they must be based on the same dimensions. That is, you can combine a set containing the top customers with another set containing the customers that purchased last year. However, you cannot combine the top customers set with a top products set.

**To combine sets:**

1. In the Data pane, under Sets, select the two sets you want to combine.
2. Right-click the sets and select **Create Combined Set**.
3. In the Create Set dialog box, do the following
   - Type a name for the new combined set.
   - Verify that the two sets you want to combine are selected in the two drop-down
menus.

- Select one of the following options for how to combine the sets:
  - **All Members in Both Sets** - the combined set will contain all of the members from both sets.
  - **Shared Members in Both Sets** - the combined set will only contain members that exist in both sets.
  - **Except Shared Members** - the combined set will contain all members from the specified set that don't exist in the second set. These options are equivalent to subtracting one set from another. For example, if the first set contains Apples, Oranges, and Pears and the second set contains Pears and Nuts; combining the first set except the shared members would contain just Apples and Oranges. Pears is removed because it exists in the second set.

- Optionally specify a character that will separate the members if the sets represent multiple dimensions.

4. When finished, click **OK**.

**Note:** This functionality is not available in workbooks created before version 8.2 that use Microsoft Excel or text file data sources, workbooks that use the legacy connection, or workbooks that use Microsoft Access data sources.

**Examples of Sets**

There are many ways you can use sets to answer complex questions and compare cohorts of data. Below are some examples of ways you can use sets to define and compare subsets of data.

**How do members of a set contribute to the total?**

You may have all kinds of questions surrounding how the members in a set contribute to the overall total. For example, what percent of total sales come from repeat customers? You can answer these types of questions using the IN/OUT mode for a set.

The example below uses sales data to create a set for customers who have purchased 5,000 USD or more in products.
Create the set

1. Right-click (control-click on Mac) the Customer Name dimension in the Data pane and select Create > Set.

2. In the Create Set dialog box, type a name for the set. In this example, we'll call the set, "Customers"

3. Select the Use all option so the condition always applies to all values even when new customers are added.

4. On the Condition tab, click By field, and then define a condition that only includes customers when Sum of Sales is greater than or equal to 5,000.
5. Click OK.

Create the visualization

1. Drag the new set from the Sets area at the bottom of the Data pane to the Rows shelf.

2. From Measures, drag Sales to the Columns shelf. The view now shows the total sales for customers who have purchased more than 5,000 USD of product and the total sales for all other customers.

3. Finally, click on the drop-down arrow on the Sum (Sales) field on the Column shelf and select Quick Table Calculation > Percent of Total on the context menu.

The view now shows that customers with sales greater than or equal to 5,000 make up about 39% of the overall sales.
How many members of a set exist in another set?

Another common use of sets is to compare subsets of data or cohorts. For example, you may wonder how many customers that purchased last year also purchased this year. Or if a customer purchased a specific product, what other products did they buy? You can answer these types of questions by creating multiple sets and combining them. The example below uses sales data to determine how many customers who purchased in 2012 also purchased in 2013.

Create a combined set

1. Drag the **Customer Name** field to the Rows shelf.
2. Drag the **Order Date** field to the Filters shelf.
3. In the Filter Field dialog box, select **Years** and click **Next**.
4. In the Filter dialog box, select **2012** and click **OK**.
5. Back in the view, press CTRL + A (Command-A on a Mac) on your keyboard to select all of the customers.

6. Right-click (control-click on Mac) the selection and select Create Set.

7. In the Create Set dialog box that opens, type a name for the set. In this example, we’ll call the set "Customers (2012)".

8. Click OK.

9. On the Filters shelf, right-click (control-click on Mac) Order Date and select Edit Filter.

10. In the Filter dialog box, change the filter to only include 2013 instead of 2012, and then click OK.
11. Again, press CTRL + A (Command-A on a Mac) on your keyboard to select all of the customers.

12. In the view, right-click (control-click on Mac) the selection and select Create Set.

13. In the Create set dialog box that opens, type a name for the set. This set will be called "Customers (2013)".

14. Click OK.

15. In the Data pane, select both the Customers 2012 and Customers 2013 by holding the Ctrl key (Command key on a Mac) on your keyboard as you select.

16. Right-click (control-click on Mac) the selection and select Create Combined Set.

17. In the Create Set dialog box, type a name for the new set. In this example, we'll call the set "Customers (2012 & 2013)".

18. Make sure the correct two sets are selected in the drop-down menus.
19. Select the option to include **Shared Members in Both Sets**.

![Create Set [Set 2]](image)

20. Click **OK**.

**Create the visualization**

1. At the bottom of the workbook, click the New Worksheet icon.

2. In the new worksheet, drag the **Customer Name** dimension to the **Rows** shelf.

3. Click the drop-down arrow on the Customer Name field on the Rows shelf and select **Measure > Count (Distinct)** from the context menu.

4. Finally, from the **Sets** area of the **Data** pane, drag the **Customers (2012 & 2013)** field to the **Filters** shelf. You can see that 437 customers purchased products in both 2012
Hierarchical sets and descendants

A hierarchical set filters data to the selected members and all of their descendants. They are unique to multidimensional (cube) data sources and are defined within the data source prior to connecting to Tableau Desktop.
When you create sets in Tableau from a cube data source, descendants and any hierarchical structures are automatically included with the selected members.

For example, a set named Dairy is created from the Product hierarchy. As shown below, it includes only the Dairy product department.

Consider the following view. The Product Department dimension is placed on the Rows shelf and the Store Sales measure is placed on the Columns shelf.
If you place the Dairy set on the Filters shelf, you can see that the view is filtered to include only the Dairy product categories.

As shown below, you can drill down into Product Department to reveal the Product Category, Product Subcategory, and Brand Name levels. As these descendants are revealed, row headers are added to the view. This is because a set filter allows you to view the levels of detail contained within the filtered members.
Create Sets for Top N and Others

If you collect large sets of data that you want to visualize, you might find that limiting the amount of information displayed to an important subset of records helps you work with and answer questions about the data more effectively.

This article describes how to create an interactive view that separates your customers into two dynamic groups:

- The top N customers
- All other customers

The view includes a control that your users can adjust to change the number of customers included in the top customers group. When they change the number, the view updates accordingly.

Sets and supported data sources

The method described in this article for creating a view of the top customers uses the In/Out functionality of sets.

Sets were introduced with Tableau Desktop version 8.0.
For live connections, the In/Out functionality requires a relational or multidimensional data source.

If you use a file-based data source, such as a Microsoft Excel workbook or text file, you can take an extract with which you can create sets.

**Step 1: Create the parameter**

1. In Tableau Desktop, open a new workbook and connect to the Sample-Superstore data source.
2. Open a new worksheet.
3. In the Data pane, click the drop-down to the right of Dimensions and select Create Parameter.
4. In the Create Parameter dialog box, do the following:
   - In the Name text box, type Top Customers 2.
   - For Data type, select Integer.
   - For Current value, type 5.
   - For Allowable values, click Range.
   - Under Range of values, do the following:
     - Click Minimum and type 5.
     - Click Maximum and type 20.
     - Click Step size and type 5.

This parameter will be used, in combination with the top N set you will create in the next step, to quickly adjust the top N value in the view.

**Step 2: Create the top N customers set**

1. In the Data pane, under Dimensions, right-click Customer Name, and select Create > Set.
2. In the Create Set dialog box that opens, do the following:
   - In the Name text box, type Top N Customers by Sales.
   - Click the Top tab.
- Select **By Field**.
- From the field drop-down list (Category), select **Sales**.
- From the aggregation drop-down list, select **Sum**.
- When finished, click **OK**.

**Step 3: Set up the view**

1. From **Sets**, drag **Top N Customers by Sales** to the **Rows** shelf.
2. From **Dimensions**, drag **Customer Name** to the **Rows** shelf, positioning it to the right of the set.
3. From **Measures**, drag **Sales** to the **Columns** shelf.
4. On the toolbar, click the Sort Descending button ![Sort Descending button](image) to make sure that the set is working.
5. In the **Data** pane, under **Sets**, right-click **Top N Customers by Sales**, and then click **Create Calculated Field**.

6. In the Calculated Field dialog box that opens, complete the following steps:

   - In the **Name** text box, type **Subset Labels**.
   - In the **Formula** text box, type the following formula to create dynamic labels for the customers in the set:

     ```
     IF [Top N Customers by Sales]
     THEN "Top " + str([Top Customers 2]) + " Customers"
     ELSE "Others"
     END
     ```

   - When finished, click **OK**.

7. From **Dimensions**, drag **Subset Labels** to the **Rows** shelf, placing it between the Top
N set and the **Customer Name** dimension.

8. On the **Rows** shelf, right-click the **IN/OUT(Top N Customers by Sales)** set, and then clear **Show Header**.

   This hides the In/Out labels while retaining the sort order so that your top N subset always appears at the top of the view.

9. From **Sets**, drag **Top N Customers by Sales** to **Color** on the **Marks** card.

---

Step 4: Combine the Top N set with a dynamic parameter

1. In the **Data** pane, right-click **Top N Customers by Sales**, and then select **Edit Set**.

2. In the Edit Set dialog box, do the following:
   - Select the **Top** tab.
   - Click the value drop-down menu, and select the **Top Customers 2** parameter.
   - Click **OK**.
This links the Top N Customers by Sales set to the Top Customers 2 dynamic parameter, instead of to a static list of 10.

This parameter will be used in combination with the Top N Customers by Sales set, to adjust the top N value in the view.

3. In the Data pane, under Parameters, right-click the Top Customers 2 parameter, and select Show Parameter Control.

You can control the top N value by using the Top Customers 2 parameter control that appears in the view.
Additional tips for improving the view's functionality

Here are some additional steps you can take to give your viewers more flexibility in displaying the customer subsets.

1. From the Data pane drop-down menu, select Create Parameter.

2. In the Create Parameter dialog box that opens, do the following:
   - For Name, type Expand or Collapse.
   - For Data type, select String.
   - For Allowable values, select List.
   - In the List of values, type the values Expand and Collapse.
• When finished, click OK.

3. Select Analysis > Create Calculated Field.

4. In the Create Calculated Field dialog box that opens, do the following to create a calculation that uses the parameter you just created. This calculation enables viewers to specify how to view customers in the Others subset:
   - For Name, enter Customer Names Calc.
   - In the formula box, type the following formula, and then click OK:
     
     ```
     IF [Expand or Collapse]="Collapse" THEN
     IF [Top N Customers by Sales]
     THEN [Customer Name]
     ELSE "Others" END
     ELSE [Customer Name] END
     ```

5. In the Data pane, under Parameters, right-click the Expand or Collapse parameter, and select Show Parameter Control.

6. From the Columns shelf, drag the SUM(Sales) measure to Label on the Marks card.

7. From Dimensions, drag Customer Names Calc directly on top of the Customer Name field on the Rows shelf, so that it replaces it.
Now you can use the **Expand or Collapse** parameter control to see the list of names in the top N customers and the remaining customers rolled up into a single **Others** entry.

Create Parameters

Parameters are dynamic values that can replace constant values in calculations, filters, and reference lines.
For example, you may create a calculated field that returns true if Sales is greater than $500,000 and otherwise return false. You can replace the constant value of “500000” in the formula with a parameter. Then using the parameter control you can dynamically change the threshold in your calculation. Alternatively, you may have a filter to show the top 10 products by profit. You can replace the fixed value “10” in the filter to by a dynamic parameter so you can quickly look at the top 15, 20, and 30 products.

**Note:** You cannot create parameters or display parameter controls when you edit views on the web. But you can use parameters in calculations, reference lines, and filters.

In this article

- Create a parameter below
- Edit a parameter on page 1035
- Use a parameter in a calculation on page 1036
- Use a parameter in a filter on page 1036
- Use a parameter in a reference line on page 1037
- Show a parameter control in the viz on page 1038

Create a parameter

Follow the instructions below to create a new parameter from the Data pane.

1. In the Data pane, click the drop-down arrow in the upper right corner and select **Create Parameter**.

![Create Parameter in Data pane](image)
2. In the Create Parameter dialog box, give the field a **Name** and optionally write a **Comment** to describe the parameter.

![Create Parameter dialog box](image)

3. Specify the data type for the values it will accept:

![Data type selection](image)

4. Specify a current value. This is the default value for the parameter.

![Current value](image)

5. Specify the display format to use in the parameter control.
6. Specify how the parameter will accept values. You can select from the following options:

- **All** - the parameter control is a simple type in field.
- **List** - the parameter control provides a list of possible values for you to select from.
- **Range** - the parameter control lets you select values within a specified range.

The availability of these options is determined by the data type. For example, a string parameter can only accept all values or a list. It does not support a range.

If you select List, you must specify the list of values. Click in the left column to type a value. Each value can also have a display alias. You can copy and paste a list of values by clicking **Paste from Clipboard**. Alternatively you can add the members of a field as the list of values by selecting **Add from Field**.
If you select **Range** you must specify a minimum, maximum and step size. For example, you can define a date range between January 1, 2010 and December 31, 2010, with the step size set to 1 month to create a parameter control that lets you select each month in 2010.

![Range of values](image)

7. When finished, click **OK**.

   The parameter is listed in the Parameters section at the bottom of the **Data** pane.

![Parameters](image)

It is also available everywhere else you can use a parameter—for example, on the Top tab in the Filter dialog box, or in the Reference Line dialog box. Parameters are global across the workbook and can be used in any worksheet.

**Edit a parameter**

You can edit parameters from the Data pane or the parameter control. Follow the instructions below to edit a parameter:
1. Do one of the following:
   - Right-click (Control-click on a Mac) the parameter in the Data pane and select Edit.
   - Select Edit Parameter on the parameter control card menu.

2. In the Edit Parameter dialog box, make the modifications as necessary.

3. When finished, click OK. The parameter is updated along with any calculations that use it.

To delete a parameter, right-click it in the Data pane and select Delete. Any calculated fields that use the deleted parameter will become invalid.

Use a parameter in a calculation

Parameters give you a way to dynamically modify values in a calculation. Rather than manually editing the calculation (and all dependent calculations), you can use a parameter. Then when you want to change the value, you open the parameter control, change the value, and all of the calculations that use that parameter are updated.

To use a parameter in a calculation, drag the parameter from the Data pane and drop it in the calculation editor, either at a new location in the formula or to replace a part of the current formula:

Use a parameter in a filter

Parameters give you a way to dynamically modify values in a TopN filters. Rather than manually setting the number of values you want to show in the filter, you can use a parameter. Then when you want to change the value, you open the parameter control and the filter updates. For example, when creating a filter to show the Top 10 products based on total profit, you may want
to use a parameter instead of the fixed “10” value. That way, you can quickly update the filter to show the top 10, 20, or 30 products.

A list of parameters is available in the drop-down lists on the Top tab of the Filter dialog box. Select the parameter you want to use in the filter.

To show the parameter control, right-click the parameter in the Data pane and select Show Parameter Control. Use the parameter control to modify the filter to show the top 10 products, 15 products, 20 products, and so on.

Use a parameter in a reference line

Parameters give you a way to dynamically modify a reference line, band, or box. For example instead of showing a reference line at a fixed location on the axis, you can reference a parameter. Then you can use the parameter control to move the reference line.
A list of parameters is available in the Value drop-down list in the Add Reference Line, Band, or Box dialog box. Select the parameter you want to use.

![Add Reference Line, Band, or Box dialog box]

The reference line is drawn at the Current Value specified by the parameter. To open the parameter control, right-click (Control-click on a Mac) the parameter in the Data pane and then select Show Parameter Control. Use the parameter control to change where the reference line is drawn.

**Show a parameter control in the viz**

The parameter control is a worksheet card that lets you modify the parameter value. Parameter controls are very similar to filter cards in that they contain controls that modify the view. You can open parameter controls on worksheets and dashboards and they are included when you save to the web or publish to Tableau Server.

To open the parameter control, right-click (Control-click) the parameter in the Data pane and select Show Parameter Control.
Like other cards, parameter controls have a menu that you can open using the drop-down arrow in the upper right corner of the card. Use this menu to customize the display of the control. For example, you can show a list of values as radio buttons, a compact list, a slider, or a type in field. The options available on this menu depend on the data type of the parameter as well as whether it accepts all, a list, or a range of values.

Use Parameters to Make Views More Interactive

Parameters are useful when you want to add interactivity and flexibility to a report, or to experiment with what-if scenarios. Suppose you are unsure which fields to include in your view or which layout would work best for your viewers. You can incorporate parameters into your view to let viewers choose how they want to look at the data.

When you work with parameters, consider the following two things that are important in making them useful:

- They need to be used in calculations.
- The parameter control needs to be displayed so that viewers can interact with it.
Before you begin, decide which fields you want to make interactive. For example, you could allow users to view the categories within a dimension by color, or to view sales data over a period of time that they choose, and so on. The example described here sets up a table for which users can select the dimensions to display in the columns and rows.

**Tip:** To learn how to create parameters for what-if and other common scenarios, see the on-demand training video Parameters on the Tableau website. Also see Use Parameters to Add Multiple Views to Your Viz and Using a parameter to change the view in a dashboard.

Create the parameters and calculated fields

These steps use the Superstore sample to create a new parameter while you build the calculated field that will take advantage of it.

**Step 1**

Select **Analysis > Create Calculated Field**.

**Step 2**

In the Calculated Field dialog box, for **Name**, type **Column 1 Category**.

**Step 3**

Next to **Parameters**, click **Create**, and in the **Create Parameter** dialog box, complete the following steps.

1. Name the parameter appropriately so that viewers can tell what changing it will do. This example uses **Select Column 1 Heading**.
2. For **Data type**, select **String**.
3. For **Allowable Values**, select **List**, type **None** as the first value in the list, and then press **Enter**.
4. Complete the list by typing the names of the additional dimension fields that you want to expose through the parameter.

**Note:** This example uses the customer name, customer segment, region, department, and category fields. These are all dimensions of the same data type (string). If you wanted to include a measure such as profit in this list, one option would be to convert the measure to a string value. You would do this when you build the calculated field, using the `STR()` function. This article covers only the single data type scenario.
The **Display As** aliases default to the field name, and for this exercise you can leave them as they are.

5. Click **OK** to return to the Calculated Field dialog box.

**Step 4**

Repeat the previous step to create the following additional parameters:

- Select Column 2 Heading
- Select Row 1 Heading
- Select Row 2 Heading

**Tip:** Instead of typing each value in the list, click **Add from Parameter** to add them from **Select Column 1 Heading**.

**Step 5**

In the Calculated Field dialog box, for **Formula**, build the following calculation:
CASE [Select Column 1 Heading]
WHEN 'Customer Name' THEN [Customer Name]
WHEN 'Customer Segment' THEN [Customer Segment]
WHEN 'Region' THEN [Region]
WHEN 'Department' THEN [Department]
WHEN 'Category' THEN [Category]
ELSE ''
END

Confirm that the status message indicates that the formula is valid, and then click OK.

Note: ELSE accounts for the None value that you included in the parameter, and it returns an empty string.

Step 6

Create three more calculated fields, one for each of the additional parameters you created:

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Calculated field name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Column 2 Heading</td>
<td>Column 2 Category</td>
</tr>
<tr>
<td>Select Row 1 Heading</td>
<td>Row 1 Category</td>
</tr>
<tr>
<td>Select Row 2 Heading</td>
<td>Row 2 Category</td>
</tr>
</tbody>
</table>
The basic formula for each calculated field is the same as in the previous step, except that you reference a different parameter in each `CASE` statement.

Give viewers a way to interact with views
Now you expose the parameter control so users can select the categories they want to display.

**Step 1**
For each parameter you created, do the following:
In the Parameters pane, right-click the parameter and select **Show Parameter Control**.

![Parameter Control Menu](image)

**Step 2**
From the Dimensions pane, drag the calculated fields you created to the Columns and Rows shelves.

**Step 3**
From the Measures pane, drag a measure to the view. In this example, **Sales** is placed on **Label** on the Marks card.

**Step 4**
Test your parameters by selecting fields in the parameter controls.

**Tips:**
• Sort the dynamic dimension fields alphabetically.
• Hide field labels for rows and columns.

![Image of Tableau interface]

Step 5
Reset all parameters to **None** and publish the workbook to Tableau Server.

![Image of Tableau Server menu]

Viewers can set up their own reports, save their parameter settings, and share views with others.

**Additional information**

For information about building views dynamically, see **Swap Measures Using Parameters**.

For information about the different areas in the Tableau interface in which you can create and incorporate parameters, see **Create Parameters on page 1031** and its related topics in the Tableau Help.
Example - Add a Parameter to a Map View

This example uses the World Indicators sample data source to demonstrate the following:

- How to build a map view that shows the birth rate for each country in the world.
- How to create a calculated field that distinguishes countries with a low birth rate from those with a high birth rate.
- How to create and display a parameter so that users can set the threshold for low vs. high birth rate.

Build a map view

1. In the Data pane, double-click Latitude and then Longitude.
   
   Tableau puts Longitude on Columns, Latitude on Rows, and displays a map of the world.

2. Drag the Year dimension to Filters.

3. In the Filter Field [Year] dialog box, choose Years and then click Next:
4. In the Filter [Year of Year] dialog box, select **2012** and then click **OK**.
5. Drag the **Country** dimension to **Detail**.

6. Set the Marks type to Map:
7. Drag the **Birth Rate** measure to **Label**.

You now have a map that shows birth rates for countries around the world:
You can zoom the map or hover to see a tooltip for any country.

Create a calculated field to set a threshold

Next, you'll distinguish a low birth rate from a high birth rate.

1. From the top menu, select **Analysis > Create Calculated Field**.

2. Name the field **High Birth Rate** and type or paste this calculation in the formula field:

   \[
   \text{IF ([Birth Rate]) } \geq 0.014 \text{ THEN "High" ELSE "Low" END}
   \]

   The value 0.014 is equivalent to 1.4%. The range of actual values that we can see on the map range from below 1% up to nearly 5%.

   When you click **OK** to apply and save this calculation, Tableau categorizes it as a dimension.

3. Drag **High Birth Rate** to **Color**.

   The map now shows low birth rate countries in one color and high birth rate countries in another:

   ![Map showing high and low birth rates](image)

But the definition of high birth rate as anything equal to or greater than 1.4% is arbitrary—that value was chosen because it divided the world's countries about evenly.
Instead, you can let users define that threshold, or give them a control that they can use to see how changing the threshold changes the map. To do this, you create a parameter.

**Create a parameter**

1. Right-click (control-click on a Mac) in the Data pane and select Create > Parameter.

   **Note:** If you clicked on a field on the Data pane, some fields might be filled. You can just change the fields as needed.

2. In the Create Parameter dialog box, name the new parameter *Set Birth Rate* and configure it as shown:

   ![Image of Create Parameter dialog box]

   For information on the fields in the Create Parameter dialog box, see Create Parameters on page 1031.

   Because the Data Type is Float, the parameter control, when you display it in the next procedure, will be in the form of a slider. This is because floating point values are continuous—there are an infinite number of possible values.
The **Current value** sets the default for the parameter: 0.019 is 1.9%. The **Range of values** section sets the minimum and the maximum values and the step size—that is, the least amount by which the value can change.

3. Click **OK**.

**Create and display the parameter control**

Now you must connect the parameter to the **High Birth Rate** field.

1. Right-click **High Birth Rate** in the **Data** pane and select **Edit**.

2. Replace the hard-coded 0.014 value in the field definition with the parameter name:

   ```
   IF ([Birth Rate]) >= [Set Birth Rate] THEN "High" ELSE "Low"
   END
   ```

   Then click **OK**.

3. Right-click the **Set Birth Rate** parameter in the **Data** pane and select **Show Parameter Control**.

By default, the parameter control is shown on the right. Now you and users of your view can raise or lower this value incrementally to see how changing the definition of "high birth rate" affects the map.
Example - Swap Measures Using Parameters

You may want to create a view that shows how the values of two measures compare against each other. But what if you also wanted to be able to choose which measures were being compared—or better yet, add a control to the view that would let any user select the measures to be compared. You can create such a view using parameters and calculated fields. The calculated fields replace the measures in the view and can be set interactively by the user with parameter controls.

Overview

The process requires you to do the following:

1. Create the parameters
2. Create calculated fields to change the measures in the view
3. Set up the view

This article includes an additional section that you can use as an alternative to step 2:

- Create calculated fields to change measures and specify aggregations

The following sections break these procedures down into specific instructions.

The scenario uses the Sample - Superstore data source provided with Tableau Desktop.

Create the parameters

Follow these steps.

1. Click the drop-down arrow to the right of the word Dimensions in the Data pane and select Create Parameter.
2. In the Create Parameter dialog box, do the following:

   a. Name the parameter **Placeholder 1 Selector**.

   b. Set the **Data type** to **String**.

   c. Skip down to the **Allowable values** field, and choose **List**.

   d. Type individual measure names in the **List of values** area: **Discount**, **Profit**, **Quantity**, and **Sales**.

      The Edit Parameter dialog box should now look like this:
e. Click **OK** to close the Edit Parameter dialog box.

3. Create a second parameter, **Placeholder 2 Selector**, with the exact same configuration.

There are different ways you can do this. The easiest way is to click **Placeholder 1 Selector** in the Data pane, choose **Duplicate**, and then change the name of the duplicated parameter to **Placeholder 2 Selector**.

Create calculated fields to change the measures in the view

Follow these steps.

1. Choose **Analysis > Create Calculated Field** to open the calculation editor. Name the calculation **Placeholder 1** and type or paste the following in the formula area:

   ```sql
   CASE [Placeholder 1 Selector]
   WHEN "Discount" THEN [Discount]
   WHEN "Profit" THEN [Profit]
   ```
WHEN "Quantity" THEN [Quantity]

WHEN "Sales" THEN [Sales]

END

Note: The calculated field must reference the Value entry for a given row, and not the Display As value.

2. Click OK to exit the calculation editor.

3. Create a second calculated field, Placeholder 2, with the same definition. Again, the easiest way to do this is to click Placeholder 1 in the Data pane, choose Duplicate, and then change the name of the duplicated field to Placeholder 2.

Set up the view

Follow these steps.

1. Drag Placeholder 2 to Columns and Placeholder 1 to Rows.

   Because you dragged measures to both shelves, the default view is a scatter plot. For more on why Tableau does this, see Example: Scatter Plots, Aggregation, and Granularity on page 290.

2. Drag Customer Name to Detail and Region to Color.

3. In the Parameters area of the Data pane, click each Placeholder 1 Selector and choose Show Parameter Control. Then do the same for Placeholder 2 Selector.

4. Tableau displays the parameter controls beyond the right side of the view by default. Drag them over to the left side to make them easier for your users to see.

Your view is now complete. The parameter controls let users select the measures to be used on the X and Y axes. For example, the view below on the left shows Quantity vs. Discount, while in the view the right the parameter controls have been used to show Profit vs. Sales.
Create calculated fields to change measures and specify aggregations

As an alternative to the Create calculated fields to change the measures in the view section, above, consider creating calculated fields that specify aggregations for individual measures. As written above, the calculated fields do not specify aggregations. Notice in the image above that Tableau automatically assigns an aggregation (SUM) to the Placeholder 1 and Placeholder 2 fields. But you know your data, and you may want to dictate which aggregation Tableau uses for your measures. So instead of the calculated field definition above, consider a definition like the following:

CASE [Placeholder 1 Selector]
  WHEN "Discount" THEN SUM([Discount])
  WHEN "Profit" THEN AVG([Profit])
  WHEN "Quantity" THEN SUM([Quantity])
  WHEN "Sales" THEN AVG([Sales])
END

It's up to you to decide whether to explicitly aggregate measures in your field definitions. The only thing you cannot do is mix-and-match: that is, you cannot define aggregations for some measures, but not for others.
Here’s how the **Profit vs. Sales** scatter plot changes when you specify AVG as the aggregation for these fields, as opposed to not specifying an aggregation and letting Tableau default to SUM:

**Profit vs. Sales** with default aggregation  
**Profit vs. Sales** with explicit aggregation

Similar, but different.
Dates and Times

How you work with dates in Tableau depends on whether you are using a relational or cube (multidimensional) data source. This section discusses the differences.

Dates in Cube (Multidimensional) Data Sources

In Tableau Desktop, cube (multidimensional) data sources are supported only in Windows.

For cube data sources, dates dimensions are usually organized into hierarchies that contain levels such as year, quarter, and month. In addition, some multidimensional data sources have time intelligence enabled, which makes it possible to look at data levels different ways, such as Months by Year, Months by Quarter, Weekends, etc. These levels are represented as attributes of the hierarchy. Hierarchies and attributes are defined when the cube is created and you cannot modify them in Tableau. For example, the Year dimension from an Oracle Essbase data source is shown below.

When you place a multidimensional date on a shelf, the field is treated like any other dimension. For example, you can drill down, drill up, and so on.

Dates in Relational Data Sources

For relational data sources, dates and times are automatically placed in the Dimensions area of the Data pane and are identified by the date or date-time icon. For example, the Order Date and Ship Date dimensions from an Excel data source are shown below.
When you place a relational date on a shelf, the field name is automatically modified to reflect the default date level. Tableau defines the default date level to be the level at which there are multiple instances. For example, if the date field includes multiple years, the default level is year. However, if the date field contains data for just one year but includes multiple months, then the default level is month.

If you don’t want Tableau to automatically select a date level and would rather have a date dimension be a continuous field, you can right-click (control-click on Mac) the field in the Data pane and select **Convert to Continuous**. The dimension then turns green in the Data pane; now when you use that dimension in a view, it will be continuous. You can easily revert back by selecting **Convert to Discrete** from the field’s context menu in the Data pane. You can also convert a field in the view to continuous while it is on a shelf by selecting **Continuous** on its context menu (which you can see when you right-click (control-click on Mac) the field). The field on the shelf turns green but the field in the Data pane is still discrete.

**Date Properties for a Data Source**

You can set date properties for a data source. To do so, right-click (control-click on Mac) a data source in the Data pane and choose Date Properties:
The options in the Date Properties dialog box are:

- **Week start** - Specifies which day is considered the first day of the week. You can override the **Week start** value you set at the data source level by including a `start_of_week` value with certain date functions (DATEDIFF, DATENAME, DATEPART, and DATETRUNC). See Date Functions on page 1298.

The initial value for this setting is in some cases determined by your data source. If the data source does not initialize the start-of-week day, then Tableau consults the system's locale settings. The day which is considered the first day of the week varies from region to region. For example, Sunday is the first day of the week in the US, while Monday is the first day in the EU.

This setting is disabled for multidimensional data sources because this information is defined when cube designer creates the date/time dimension.

After a data source is created, the **Week start** value does not automatically change if for any reason the default start-of-week setting changes in the host operating system (for example, if the data source is uploaded to Tableau Server in a different country).

Calendar controls reflect the workbook locale rather than the data source's **Week start** setting. The exception is that a week trunc start of week that shows the week number in it will use the data source's **Week start** setting in order to provide a consistent week number value in the calendar.

- **Fiscal year start** - Specifies which month is considered the first month of the fiscal year. To specify whether a date dimension uses the standard calendar (Jan. 1 - Dec. 31) or the fiscal calendar, right-click (control-click on Mac) the field in the Data pane and select **Default Properties > Calendar Type** and then select **Standard Calendar** or **Fiscal**.
Calendar. For any date field in the view, you can then specify whether to use the standard calendar or the fiscal calendar. For details, see Fiscal Dates on page 1068.

- **Date format** - Specifies the default format for data dimensions, as displayed, for example, in tooltips. To override the default date format for a date dimension, right-click (control-click on Mac) the field in the Data pane and select **Default Properties > Date Format** and then select one of the available formats.

Day of the week sorting

If you are working in a language for which Tableau does not provide a local version, set your workbook locale to assure that Tableau can sort the days of the week in the correct chronological order. Otherwise, Tableau will sort the names of the days alphabetically. For information on setting the locale, see Language and Locale on page 242.

If none of the supported locales is appropriate, you can sort the days of the week manually. See **Sort Data in a Visualization** on page 1202.

Supported date formats

When working with dates, Tableau retrieves date formats automatically from the data source.

For a list of supported date formats, see the table of supported date format symbols in Custom Date Formats on page 1071.

Changing Date Levels

For both relational and multidimensional data sources, you can change the date level using the field’s context menu after dragging it to a shelf. For cube (multidimensional) dates, the levels available in the context menu are given by the levels defined in the date hierarchy. For relational dates, you can select between the discrete date levels at the top of the menu or the continuous date at the bottom. A preview of each date level is shown. Below is the field menu for a relational date with the discrete Year level selected.

In Tableau Desktop, cube (multidimensional) data sources are supported only in Windows.
Click any of the options in the blue areas to configure the field as a discrete date. 
Selecting one of these options creates what is known as a "date part."

Click any of the options in the green areas to configure the field as a continuous date. 
Selecting one of these options creates what is known as a "truncated date."

When you select a particular level, Tableau asks the data source to perform a computation on the date field. For example, suppose a particular row in your data source has a date entry of 01/23/16. The year is 2016, the quarter is 1 because January falls in the first quarter, and the week number is 4 because January 23rd falls in the fourth week. How the date level is computed depends on the configured Date Properties for the data source. See Date Properties for a Data Source on page 1059.
**Note:** Some date levels might not make sense for your relational data source. For example, if the date format does not include time information such as hour, minute, or second, then selecting one of these options will not add any data to your view.

You can work with dates at varying levels of detail simultaneously. You do this by clicking the control on a date field in the view. This is known as drilling into a date. For example, if you drill into a discrete (blue) date field \textit{YEAR(Order Date)}, Tableau adds a second field to the right of the initial date, \textit{QUARTER(Order Date)}, as in the image below. You can continue drilling down until you reach the finest available granularity for the date field.

You can also drag date fields to the \textbf{Rows} or \textbf{Columns} shelf multiple times in order to nest them and to drill down into them at varying levels of detail.

For example, the view shown below drills down from the year level to display the quarter level as well.

![Diagram showing drilling into dates](image)

By removing one of the date fields from the Columns shelf, and then selecting \textbf{Month} from the remaining date field’s context menu, you can display the data for each month across all years.
To display even finer granularity, select Month from the lower, continuous section of the field's context menu.
Tableau then displays the dates across the range of available years, at a month-by-month level of granularity.
Custom Dates

You might want to create a custom date if you always use a date at a specific level, or if you want to create calculations that rely on a binned or truncated date.

You can create a custom date by working in the Data pane or by using the DATEPART and DATETRUNC functions in a calculated field.

The custom date you create becomes a new field in the Data pane.

1. Right-click (Control-click on Mac) a date field in the Data pane and select Create > Create Custom Date.

2. In the Create Custom Date dialog box, type a name for the custom date, such as Date (Quarter, Year).

3. From the Detail list, select the level you want to show the date at.

4. Specify whether you want the custom date to be discrete (select Date Parts) or continuous (select Date Value).
When you are finished, click **OK**.

**Example - format column headers as Month, Day, Year**

To format column headers as Month, Day, Year (for example, January 1, 2011), follow the steps below. This example uses the Sample - Superstore data source provided with Tableau Desktop.

1. Drag **[Order Date]** to **Columns**.
2. Drag **Sales** to **Rows**.
3. Right-click (Control-click on a mac) **[Order Date]** on the **Columns** shelf and choose **More > Custom**.
4. In the Custom Date dialog box, choose **Month / Day /Year** from the **Detail** drop-down list.

The view now has the column headers in the correct format:
Fiscal Dates

In some situations, a date field needs to be expressed in terms of an organization's fiscal year. Calendar years run from January 1st until December 31st, but an organization’s fiscal year might start in a different month. For example, a company’s fiscal year might run from June 1st through May 31st. In such cases, it’s appropriate to display some date values in a view using fiscal equivalents (Fiscal Year, Fiscal Quarter, and Fiscal Week Number) rather than their calendar equivalents.

To set the fiscal year start month for a data source, follow these steps:

1. Right-click (Control-click on Mac) the data source in the Data pane to open the Date Properties dialog box.

2. Set the Fiscal year start field to the appropriate month.

For each date dimension, you set the fiscal year start-month separately. In the Data pane, right-click a date dimension (Ctrl-click on Mac), and select Default Properties > Fiscal Year Start.
Note: Date functions do not take account of the configured fiscal year start. See Date Functions on page 1298.

Whether a given level of a date dimension is affected by the use of the Fiscal Calendar depends on the specific case.

<table>
<thead>
<tr>
<th>Date Level</th>
<th>When Converted to Fiscal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>Reflects the fiscal year. For example, if Fiscal year start is set to April, the year for the date June 1, 2004 would be shown as FY 2005.</td>
</tr>
<tr>
<td>Quarter</td>
<td>The Quarter reflects the fiscal quarter. For example, if Fiscal year start is set to April, the quarter for the date June 1, 2004 would be Q1.</td>
</tr>
<tr>
<td>Month</td>
<td>No change in behavior. The calendar month is the same as the fiscal month.</td>
</tr>
<tr>
<td>Day</td>
<td>No change in behavior. The calendar day is the same as the fiscal day.</td>
</tr>
<tr>
<td>Hour</td>
<td>No change in behavior. The calendar hour is the same as the fiscal hour.</td>
</tr>
<tr>
<td>Minute</td>
<td>No change in behavior. The calendar minute is the same as the fiscal minute.</td>
</tr>
<tr>
<td>Second</td>
<td>No change in behavior. The calendar second is the same as the fiscal second.</td>
</tr>
<tr>
<td>Week Number</td>
<td>The Week Number reflects the fiscal week number. For example, if Fiscal year start is set to April, the week number for the date April 1, 2004 would be 1.</td>
</tr>
<tr>
<td>Weekday</td>
<td>No change in behavior. The calendar weekday is the same as the fiscal weekday.</td>
</tr>
<tr>
<td>MM/YYYY</td>
<td>No change in behavior. This date format always displays calendar dates, even when a fiscal year has been assigned.</td>
</tr>
<tr>
<td>M/D/Y</td>
<td>This date format always displays Calendar dates, even when a fiscal year has been assigned.</td>
</tr>
</tbody>
</table>

The only date levels that explicitly indicate that the fiscal calendar is in use are the Year and Quarter level. Specifically, fiscal years and quarters are shown with the FY prefix. This is not
true for fiscal quarters or week numbers, however, which are not shown with any special fiscal markings.

Fiscal year designations for any given date dimension are applied to all instances of the field in the Tableau workbook. Fiscal dates can only be applied to dimensions in a relational data source.

Fiscal year formatting is applied to all date formats that include a year, or a year and a quarter. In particular, if you apply a custom date format, and only use the “y” and “q” placeholders, then FY will be prepended to each year.

Date Truncations

When a date dimension is using a fiscal calendar, only the following date parts and truncations will reflect the fiscal calendar:

<table>
<thead>
<tr>
<th>Date part or truncation</th>
<th>When using a fiscal calendar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year date part</td>
<td>Reflects the fiscal year. For example, if Fiscal year start is set to April, the year for the date June 1, 2004 would be shown as FY 2005.</td>
</tr>
<tr>
<td>Quarter date part</td>
<td>Reflects the fiscal quarter. For example, if Fiscal year start is set to April, the quarter for the date June 1, 2004 would be Q1.</td>
</tr>
<tr>
<td>Week Number date part</td>
<td>Reflects the fiscal week number. For example, if Fiscal year start is set to April, the week number for the date April 1, 2004 would be 1.</td>
</tr>
<tr>
<td>Year date truncation</td>
<td>Truncates to the start of the fiscal year. For example, if Fiscal year start is set to May, the date June 1, 2004 would become May 1, 2004.</td>
</tr>
<tr>
<td>Quarter date truncation</td>
<td>Truncates to the start of the fiscal quarter. For example, if Fiscal year start is set to July, the date June 1, 2004 would become April 1, 2004.</td>
</tr>
</tbody>
</table>

Perfect Pivoting with Dates

You can perfect pivot dates by placing different date levels on different worksheet shelves simultaneously. Place the date field on a variety of shelves and then select the desired date level from the fields’ context menus.

For example, the following line chart displays years as column headers and then color-encodes the marks by quarter.
You can separate the marks by month and by quarter as shown below.

Custom Date Formats

This article discusses using the custom date format field to format dates in a view. For an overview of how Tableau works with dates, see Dates and Times on page 1058, or Changing
How to find the custom date format field

Format a date field in a view

To format a date field in the view, right-click (Control-click on a Mac) the field and choose Format.

This will open the Format panel to the left of your view. Select the Dates field.
When you format dates, Tableau presents a list of available formats. In most cases, the last item in the list is Custom. You specify a custom date using format symbols listed in the Supported date format symbols on the next page table, either alone or in combination.

Format a date field in the Data pane

To format a date field in the Data pane, right-click the field and choose Default Properties > Date Format.
The date formats in the table are supported when your workbook is connected to a Tableau data extract or has a live connection to a data source that also supports the date format. (Refer to your data source’s documentation to verify that the date format you want is supported.)

Tableau retrieves date formats from the data source. Tableau Server can also retrieve date formats from the Run As user account on the server that is running Tableau Server.

**Note:** The following date formats might not be the same as those used with the `DATEPARSE` function. See [Convert a Field to a Date Field](#) on page 740 for more information.

In this article:

- Supported date format symbols below
- Custom date format examples on page 1077
- Support for Japanese era-based date formats on page 1078
- Using literal text in a date format on page 1079
- Format syntax in `DATEPARSE` function for extract data sources on page 1079

Supported date format symbols

Use the following symbols to construct a custom date format.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(:)</td>
<td>Time separator. In some locales, a different character is used to represent the time separator. The time separator separates hours, minutes, and seconds when time values are formatted. The actual character used as the time separator in formatted output is determined by your system settings.</td>
</tr>
<tr>
<td>(/)</td>
<td>Date separator. In some locales, a different character is used to represent the date separator. The date separator separates the day, month, and year when date values are formatted. The actual character used as the date separator in formatted output is determined by your system settings.</td>
</tr>
<tr>
<td>c</td>
<td>Display the date as ddddd and display the time as tttttt, in that order. Display only date information if there is no fractional part to the date serial number; display only time information if there is no integer portion.</td>
</tr>
<tr>
<td>d</td>
<td>Display the day as a number without a leading zero (1 31).</td>
</tr>
<tr>
<td>dd</td>
<td>Display the day as a number with a leading zero (01 31).</td>
</tr>
<tr>
<td>ddd</td>
<td>Display the day as an abbreviation (Sun Sat).</td>
</tr>
<tr>
<td>dddd</td>
<td>Display the day as a full name (Sunday Saturday).</td>
</tr>
<tr>
<td>ddddd</td>
<td>Display the date as a complete date (including day, month, and year), formatted according to your system’s short date format setting. The default short date format is m/d/yy.</td>
</tr>
<tr>
<td>dddddd</td>
<td>Display a date serial number as a complete date (including day, month, and year) formatted according to the long date setting recognized by your system. The default long date format is mmmm dd, yyyy.</td>
</tr>
<tr>
<td>aaaa</td>
<td>The same as dddd, only it’s the localized version of the string.</td>
</tr>
<tr>
<td>w</td>
<td>Display the day of the week as a number (1 for Sunday through 7 for Saturday).</td>
</tr>
<tr>
<td>ww</td>
<td>Display the week of the year as a number (1 54).</td>
</tr>
<tr>
<td>m</td>
<td>Display the month as a number without a leading zero (1 12). If m immediately follows h or hh, the minute rather than the month is displayed.</td>
</tr>
</tbody>
</table>
| mm     | Display the month as a number with a leading zero (01 12). If m immediately fol-
<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lows h or hh</td>
<td>the minute rather than the month is displayed.</td>
</tr>
<tr>
<td>mmm</td>
<td>Display the month as an abbreviation (Jan Dec).</td>
</tr>
<tr>
<td>mmmm</td>
<td>Display the month as a full month name (January December).</td>
</tr>
<tr>
<td>oooo</td>
<td>The same as mmmm, but localized.</td>
</tr>
<tr>
<td>q</td>
<td>Display the quarter of the year as a number (1 4).</td>
</tr>
<tr>
<td>y</td>
<td>Display the day of the year as a number (1 366).</td>
</tr>
<tr>
<td>yy</td>
<td>Display the year as a 2-digit number (00 99).</td>
</tr>
<tr>
<td>yyyy</td>
<td>Display the year as a 4-digit number (100 9999).</td>
</tr>
<tr>
<td>h</td>
<td>Display the hour as a number without leading zeros (0 23).</td>
</tr>
<tr>
<td>Hh</td>
<td>Display the hour as a number with leading zeros (00 23).</td>
</tr>
<tr>
<td>N</td>
<td>Display the minute as a number without leading zeros (0 59).</td>
</tr>
<tr>
<td>Nn</td>
<td>Display the minute as a number with leading zeros (00 59).</td>
</tr>
<tr>
<td>S</td>
<td>Display the second as a number without leading zeros (0 59).</td>
</tr>
<tr>
<td>Ss</td>
<td>Display the second as a number with leading zeros (00 59).</td>
</tr>
<tr>
<td>000</td>
<td>Display milliseconds. Use a period character as a separator before specifying milliseconds.</td>
</tr>
<tr>
<td>tttt</td>
<td>Display a time as a complete time (including hour, minute, and second), formatted using the time separator defined by the time format recognized by your system. A leading zero is displayed if the leading zero option is selected and the time is before 10:00 A.M. or P.M. The default time format is h:mm:ss.</td>
</tr>
<tr>
<td>AM/PM</td>
<td>Use the 12-hour clock and display an uppercase AM with any hour before noon; display an uppercase PM with any hour between noon and 11:59 P.M.</td>
</tr>
<tr>
<td>am/pm</td>
<td>Use the 12-hour clock and display a lowercase AM with any hour before noon; display a lowercase PM with any hour between noon and 11:59 P.M.</td>
</tr>
<tr>
<td>A/P</td>
<td>Use the 12-hour clock and display an uppercase A with any hour before noon; display an uppercase P with any hour between noon and 11:59 P.M.</td>
</tr>
</tbody>
</table>
Use the 12-hour clock and display a lowercase A with any hour before noon; display a lowercase P with any hour between noon and 11:59 P.M.

Use the 12-hour clock and display the AM string literal as defined by your system with any hour before noon; display the PM string literal as defined by your system with any hour between noon and 11:59 P.M. AMPM can be either uppercase or lowercase, but the case of the string displayed matches the string as defined by your system settings. The default format is AM/PM.

Custom date format examples

Any of the date formats symbols in the table above can be used alone or in combination.

Specifying a custom format yyyy-MM-dd HH:mm:ss.000 would produce dates in the format 2015-05-10 11:22:16.543. Such a format might be appropriate for scientific data.

Specifying a custom format DDDD DD would produce dates that show the Weekday and the Day, as shown below.

Specifying a custom format yy-mm-dd (dddd) would produce dates in the format 18-01-04 (Thursday).
Specifying a custom format "Q"1 YYYY would produce dates that show Q1 2018.

Support for Japanese era-based date formats

Starting with version 9.3, Tableau supports Japanese emperor-era-based date (Wareki) formats. Here’s how to apply an era-based date format to a field in your view:

1. Set your workbook locale to Japanese.
2. Right-click the field in the view for which you want to set a date format.
3. Choose Format.
4. In the Format pane, from the Dates drop-down list, select a format.

If the format that you want isn’t listed, you can construct your own date format. To do this, choose Custom format in the Dates box, then type your format using the Tableau date placeholders. The following era-based year placeholders are available:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
<td>Short era name (such as H for the Heisei era).</td>
</tr>
<tr>
<td>gg</td>
<td>Era name (such as 平成).</td>
</tr>
<tr>
<td>ggg</td>
<td>Long era name (for Japanese, this is currently the same as the regular era name).</td>
</tr>
<tr>
<td>e</td>
<td>Era-based year, such as 1 for the first year of an era.</td>
</tr>
<tr>
<td>ee</td>
<td>Era-based year, such as 01 for the first year of an era. If there is only one digit, then the era-based year will have a zero added to the front.</td>
</tr>
</tbody>
</table>

If your workbook locale is not Japanese, you can create a custom date format, then insert the language code !ja_JP! in front of your format, so that it looks like this:

!ja_JP! gg ee"年"mm"月"dd"日"

The language code forces the date to be treated as if it is a Japanese date.

Era-based dates are not fully supported by the Tableau Server browser view. In particular, if you publish a workbook that contains a quick filter, the e and g placeholders will not be filled in:
To avoid this issue, do not show era-based dates in quick filters if your workbook will be viewed in a browser.

Using literal text in a date format

You may want your date format to include some words or phrases, such as Fiscal Quarter q of yyyy. However, if you type that text directly into the Tableau format box, it may treat the letters like date parts:

<table>
<thead>
<tr>
<th>Quarter of Order Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fi01/1/2010 1st quarter of 2010</td>
</tr>
<tr>
<td>Fi04/1/2010 2nd quarter of 2010</td>
</tr>
<tr>
<td>Fi07/1/2010 3rd quarter of 2010</td>
</tr>
<tr>
<td>Fi10/1/2010 4th quarter of 2010</td>
</tr>
</tbody>
</table>

To prevent Tableau from doing this, put double quotes around the letters and words that should not be treated as date parts: “Fiscal Quarter” q “of” yyyy.

If you want a literal quote inside of a quoted section, insert this code: “\”\”. For example, the format “Fiscal \”\” Quarter” would be formatted as Fiscal “ Quarter.

Format syntax in DATEPARSE function for extract data sources

If you’re using the DATEPARSE function in an extract, use the syntax defined by the Unicode Consortium.

The following table lists the field types that can be represented in the format parameter of the DATEPARSE function. Click the field type to get information about the symbols, field patterns, examples, and descriptions from the Unicode Consortium website.

<table>
<thead>
<tr>
<th>Unit of time</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Era</td>
<td>n/a</td>
</tr>
</tbody>
</table>
| Year         | All symbols are supported in .hyper extracts with the exception of "U."

Notes:
- Negative values denote a year before Christ (BC). For example, DATEPARSE('y','-'-10') returns the first January of 11BC and DATEPARSE('y','-'-0') returns the
<table>
<thead>
<tr>
<th>Unit of time</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>first January of 1BC.</td>
</tr>
<tr>
<td></td>
<td><strong>Notes:</strong></td>
</tr>
<tr>
<td></td>
<td>- When working with calendar year &quot;y,&quot; the pattern &quot;yy&quot; requests the two low-order digits of the year. For numbers &lt; 70, the DATEPARSE function returns the year 2000+x. For numbers &gt;=70, the DATEPARSE function returns the year 1900+x.</td>
</tr>
<tr>
<td></td>
<td>- When working with &quot;Y&quot; in &quot;ISO week date&quot; based calendars, the year transition occurs on a week boundary and may differ from calendar year transition. The &quot;Y&quot; designation is used in conjunction with pattern character &quot;w&quot; in ISO year-week calendar. The ISO week date system is effectively a leap week calendar system that is part of the ISO 8601 date and time standard. Similar to &quot;y,&quot; negative values for &quot;Y&quot; denote a year before Christ (BC).</td>
</tr>
<tr>
<td>Month</td>
<td>All symbols are supported in .hyper extracts with the exception of &quot;I.&quot;</td>
</tr>
<tr>
<td>Notes:</td>
<td><strong>Notes:</strong></td>
</tr>
<tr>
<td></td>
<td>- The month designations is used in conjunction with &quot;d&quot; for day number.</td>
</tr>
<tr>
<td></td>
<td>- In contrast to ICU, .hyper extracts allow values 1-12. Other values will cause an error.</td>
</tr>
<tr>
<td>Week</td>
<td>All symbols are supported in .hyper extracts with the exception of &quot;W.&quot;</td>
</tr>
<tr>
<td>Notes:</td>
<td><strong>Notes:</strong></td>
</tr>
</tbody>
</table>
|              | - When working with "w," in contrast to ICU, .hyper extracts allow only valid weeks. A year has 52 or 53 weeks (ISO 8601). The DATEPARSE function validates the input. For example, an error occurs for the 53rd
<table>
<thead>
<tr>
<th>Unit of time</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week of 2016</td>
<td>week of 2016 because the 53rd week does not exist for 2016.</td>
</tr>
<tr>
<td></td>
<td>• When working with &quot;W,&quot; ICU does not support this designation, but it's useful for dates like 1st Monday of September.</td>
</tr>
<tr>
<td>Day Notes:</td>
<td>• When working with &quot;d,&quot; in contrast to ICU, .hyper extracts only allow valid day numbers. For example, an error occurs for the 31st of February.</td>
</tr>
<tr>
<td></td>
<td>• When working with &quot;D,&quot; in contrast to ICU, .hyper extracts only allow valid day numbers. For example, an error occurs for the 366th day of 2017.</td>
</tr>
<tr>
<td>Hour Notes:</td>
<td>Only &quot;h&quot; and &quot;H&quot; symbols are supported in .hyper extracts.</td>
</tr>
<tr>
<td></td>
<td>• When working with &quot;h,&quot; .hyper extract don't allow negative values for this field. Negative values cause an error.</td>
</tr>
<tr>
<td></td>
<td>• When working with &quot;H,&quot; .hyper extracts don't allow negative values for this field. Negative values will cause an error.</td>
</tr>
<tr>
<td>Minute Note:</td>
<td>In contrast to ICU, .hyper extracts don't allow negative values for this field. Negative values will cause an error.</td>
</tr>
<tr>
<td>Second Notes:</td>
<td>In contrast to ICU, .hyper extracts don't allow negative values for this field. Negative values will cause an error.</td>
</tr>
<tr>
<td></td>
<td>• When working with &quot;S,&quot; DATEPARSE('ss.SSSS', '12.3456') returns 1990-01-01 00:00:12:3456 AD.</td>
</tr>
<tr>
<td>Quarter Note:</td>
<td>In contrast to ICU, .hyper extracts only allow values 1-4.</td>
</tr>
<tr>
<td>Unit of time</td>
<td>Notes</td>
</tr>
<tr>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>All other values will cause an error.</td>
</tr>
</tbody>
</table>

### Weekday Notes:
- When working with "e and "ee," in contrast to ICU, .hyper extracts only allow values 1-7. All other values will cause an error.
- When working with "c..cc," in contract to ICU, .hyper extracts only allow values 1-7. All other values will cause an error.

| Period | n/a |

### Format Dates Using ISO-8601 Weeks and Years

For many European locales (listed below), Tableau allows you to format dates using ISO-8601 weeks and years.

### Expand to see European locales that support ISO-8601 in Tableau

da-DK, Danish (Denmark)
se-SE, Sami, Northern (Sweden)
rm-CH, Romansh (Switzerland)
 sms-FI, Sami, Skolt (Finland)
is-IS, Icelandic (Iceland)
de-DE, German (Germany)
km-KH, Khmer (Cambodia)
nl-NL, Dutch (Netherlands)
smj-SE, Sami, Lule (Sweden)
fi-FI, Finnish (Finland)
nn-NO, Norwegian Nynorsk (Norway)
it-IT, Italian (Italy)
it-CH, Italian (Switzerland)
nb-NO, Norwegian Bokmål (Norway)
pl-PL, Polish (Poland)
sv-SE, Swedish (Sweden)
et-EE, Estonian (Estonia)
lv-LV, Latvian (Latvia)
hsb-DE, Upper Sorbian (Germany)
kl-GL, Greenlandic (Greenland)
mk-MK, Macedonian (Macedonia, FYRO)
fo-FO, Faroese (Faroe Islands)
se-NO, Sami, Northern (Norway)
as-IN, Assamese (India)
fy-NL, Frisian (Netherlands)
se-FI, Sami, Northern (Finland)
ps-AF, Pashto (Afghanistan)
prs-AF, Dari (Afghanistan)
de-CH, German (Switzerland)
sv-FI, Swedish (Finland)
dsb-DE, Lower Sorbian (Germany)
de-LU, German (Luxembourg)
fr-CH, French (Switzerland)
smj-NO, Sami, Lule (Norway)
de-LI, German (Liechtenstein)
sma-NO, Sami, Southern (Norway)
sma-SE, Sami, Southern (Sweden)
The week and year numbering in an ISO-8601 calendar is different from a standard Gregorian calendar. Here's how January 2nd, 2011 would be represented:

<table>
<thead>
<tr>
<th>Calendar System</th>
<th>Week Number</th>
<th>Year Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Gregorian</td>
<td>1</td>
<td>2011</td>
</tr>
<tr>
<td>ISO-8601</td>
<td>52</td>
<td>2010</td>
</tr>
</tbody>
</table>

When formatting dates, it’s important to ensure that your month, week and year numbers all come from the same calendar system. Otherwise, the date might not make sense. There are two ways to do this:

- Let Tableau guess which calendar system you want to use for each placeholder. For example, if you type `mm yyyy` (a month followed by a year), Tableau uses the standard Gregorian year. In this case, it makes sense to use that year with a month number.
- In your format string, place an annotation after each week (ww) or year (yyyy) placeholder, to indicate what kind of placeholder it should be:

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Calendar to Use</th>
<th>Example Format String</th>
</tr>
</thead>
<tbody>
<tr>
<td>[y]</td>
<td>Standard Gregorian</td>
<td>ww[y] yyyy[y]</td>
</tr>
</tbody>
</table>

**Important:** To get correct ISO-8601 week numbering, your computer's location must be set to one of the above locales, and you must set your data source start of week to Monday. (For information on how to do this, see Date Properties for a Data Source on page 1059.) Otherwise, Tableau will number weeks using your data source start of week setting, which may result in partial weeks at the beginning and end of years being created.

**Note:** ISO week numbering will not occur if you are using a fiscal year that doesn’t start in January, regardless of the locale you choose.
Sample format strings for the date December 31, 2013

Here are sample format strings for the date December 31, 2013, where m stands for a month or day-of-month placeholder. Time placeholders, such as hours, minutes, and seconds, don't influence whether or not a year is week-based or year-based.

<table>
<thead>
<tr>
<th>Format string</th>
<th>Formatted output</th>
</tr>
</thead>
<tbody>
<tr>
<td>yyyy</td>
<td>2013</td>
</tr>
<tr>
<td>yyyy[Y]</td>
<td>2014</td>
</tr>
<tr>
<td>yyyy ww</td>
<td>2014 1</td>
</tr>
<tr>
<td>yyyy mm</td>
<td>2013 12</td>
</tr>
<tr>
<td>ww yyyy</td>
<td>1 2014</td>
</tr>
<tr>
<td>mm yyyy</td>
<td>12 2013</td>
</tr>
<tr>
<td>ww yyyy[y]</td>
<td>1 2013</td>
</tr>
<tr>
<td>mm yyyy[Y]</td>
<td>12 2014</td>
</tr>
<tr>
<td>yyyy mm ww</td>
<td>2013 12 1</td>
</tr>
<tr>
<td>mm yyyy ww</td>
<td>12 2013 1</td>
</tr>
<tr>
<td>ww mm yyyy</td>
<td>1 12 2013</td>
</tr>
<tr>
<td>ww mm yyyy[Y]</td>
<td>1 12 2014</td>
</tr>
<tr>
<td>ww yyyy mm</td>
<td>1 2014 12 (format is ambiguous)</td>
</tr>
<tr>
<td>ww yyyy[y] mm</td>
<td>1 2013 12</td>
</tr>
<tr>
<td>ww yyyy mm yyyy</td>
<td>1 2014 12 2013</td>
</tr>
<tr>
<td>ww yyyy yyyy mm</td>
<td>1 2014 2013 12</td>
</tr>
<tr>
<td>yyyy mm ww yyyy</td>
<td>2013 12 1 2014</td>
</tr>
</tbody>
</table>
Continuous Dates

You can treat a date as a continuous quantity after placing the field on a shelf. You do this by selecting one of the Continuous date options on the field’s context menu (lower list of date levels). Continuous dates draw a quantitative axis for the date values.

For example, the view below displays the sales as a function of a continuous order date and is color-encoded by category. As you can see, the color of the Order Date field changes from blue to green after it is converted to a continuous quantity.

![Graph](image)

Treating dates as a continuous quantity is particularly useful when you use Gantt bars or want to see trends using line charts as shown above.

By default, date dimensions are discrete fields for which Tableau automatically selects a date level when it is placed on a shelf. To make a date dimension continuous by default, right-click (control-click on Mac) the field in the Data pane and select Convert to Continuous. The field turns green and is automatically converted to a continuous field when you drag it to a shelf. To revert to discrete again, right-click (control-click on Mac) the field in the Data pane and select Convert to Discrete.
Start Building a Visualization by Dragging Fields to the View

You build visualizations by adding fields from the Data pane to the view. As you build a visualization, you can add as many fields as necessary to different areas of the view, and then move those fields around as you explore your data.

Watch a video: To see related concepts demonstrated in Tableau, watch Getting Started with Visual Analytics, a 6-minute free training video. Use your tableau.com account to sign in.

For an in-depth, 25-minute walkthrough of the Tableau environment, see the Getting Started free training video. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

In this article

Different ways to start building a view below
Dragging basics on the next page
Add headers on page 1090
Add axes on page 1092
Rearrange rows and columns on page 1096
Automatically add fields to the view with double-click on page 1097

Different ways to start building a view

When you build a view, you add fields from the Data pane. You can do this in different ways. For example:

- Drag fields from the Data pane and drop them onto the cards and shelves that are part of every Tableau worksheet.
- Double-click one or more fields in the Data pane.
- Select one or more fields in the Data pane and then choose a chart type from Show Me, which identifies the chart types that are appropriate for the fields you selected. For details, see Use Show Me to Start a View on page 2170.
- Drop a field on the Drop field here grid, to start creating a view from a tabular
As you start exploring data in Tableau, you will find there are many ways to build a view.
Tableau is extremely flexible, and also very forgiving. As you build a view, if you ever take
a path that isn’t answering your question, you can always undo to a previous point in your
exploration.

- To undo or redo, click undo ← or redo → on the toolbar.

You can undo all the way back to the last time you opened the workbook. You can undo
or redo an unlimited number of times.

Dragging basics

When you begin creating a new data view on a blank worksheet, you can drag a field from the
Data pane and drop it directly into the view, rather than to a shelf.
In Tableau Desktop, while dragging fields, you can hover over the different areas in the view to see how the field will be incorporated into the structure of the view. For example, dimensions typically add a row and column headers to the view, while measures add continuous axes. Below are some examples of how fields can be added to the view.

- **Dimensions add headers**
- **Measures add axes**
- **Show Me automatically adds the field according to best practices**

For a more advanced discussion of dimensions and measures, see *Dimensions and Measures, Blue and Green* on page 250.

When you drop a field on one of the active areas in the view, the field is added to the view and also to one of the shelves or cards. For example, in the view below the **Segment** dimension
was added to the **Rows** shelf and the **Profit** measure was added to the **Columns** shelf—and automatically aggregated so that the values in the data are summed.

You can, of course, also drag fields directly to the shelves instead of to the active areas in the view. You can also drag fields from one shelf to another shelf. To remove a field from a shelf, drag it off the worksheet or select **Remove** from the field's menu (available when you right-click (control-click on Mac) a field in the view. In Tableau Desktop, you can also quickly remove multiple fields from a shelf by right-clicking (control-click on Mac) the shelf and selecting **Clear Shelf**.

---

**Add headers**

You can add headers to a view by dragging a discrete (blue) dimension and dropping it to either side of an existing header, or to the left of an axis. For example, in the view below you can add the **Region** dimension by dragging it and dropping it to the right of the Sub-Category names.
As you hover over the view, a dotted black line indicates active areas where you can add headers.

If you drop **Region** there, it adds a column for Region along the top of the table, and slices the data so that you see results for each region within each sub-category.
Add axes

You can add axes by dragging a continuous (green) measure and dropping it on an active area in the view. If an axis already exists in the view you can replace the existing axis, blend the new measure with the existing axis, or add a secondary axis.

Replace the Existing Axis

Drag the new measure to the top left portion of the axis in the view. A small square drop zone appears and a single axis icon displays to indicate that a single axis will be left when you drop the measure. In this case you are replacing one measure (Sales) with another (Profit).
Blend the Measures on Single Axis

You can show multiple measures on a single axis by dragging the new measure directly on top of the existing axis. Blending measures uses the Measure Names and Measure Values fields. For more information, see Measure Values and Measure Names on page 2196.

In the image below you are adding a second measure (Profit) to the existing measure (Sales) on the axis.
The result is a view with both measures plotted along a single axis. In this case, Sales is shown in orange and Profit is shown in blue.
Add a Dual Axis

Drag the field to the right side of the view to add the measure as a dual axis. Dual Axes are useful when you want to compare two fields that have different scales. In this case, blending the two axes would distort the view. Instead you can add a dual axis.

The result is a dual axis view where the Profit axis corresponds to the blue line and the Sales axis corresponds to the orange line.
You can add up to four axes to the view: two on the Columns shelf and two on the Rows shelf. To turn a measure into a dual axis, right-click (control-click on Mac) a field on the **Rows** or **Columns** shelf and select **Dual Axis**.

---

**Rearrange rows and columns**

You can rearrange the rows and columns in the view by changing the order of the fields in the Rows and Columns shelves, or by dragging the selection border for headers or an axis.
Automatically add fields to the view with double-click

Tableau supports an additional method for automatically generating views of data called Automatic Double-Click. To use this method, double-click fields in the Data pane you are interested in. Tableau automatically adds each field to the view. That is, each double-click results in an additional field added to a shelf in an intelligent way. Like Show Me, this function leverages Tableau's ability to make an intelligent “best guess” of how the data should be displayed. Double-clicking sometimes creates a view that you can use without further modification. More often, it can provide a starting point, which you can then modify to get the ideal result.

1. Double-clicking the Profit measure in the Data pane automatically adds that field to the view in an intelligent way.
2. Double-clicking the **Category** dimension in the **Data** pane automatically adds that field to the view based on the fact that Profit is already on the Rows shelf.
3. Double-clicking the **Order Date** dimension in the **Data** pane automatically adds that field to the **Columns** shelf based on the placement of other fields in the view. As you double-click fields they are successively added to the view. The order in which you click fields determines the type of view created.
The following table describes some of the rules used in creating automatic views by double-clicking fields in the **Data** pane.

<table>
<thead>
<tr>
<th>Text Table</th>
<th>Adding a dimension first produces a text table (or cross-tab). All subsequent clicks on fields result in refinement of the text table.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bars</td>
<td>Adding a measure first and then a dimension produces a bar view. All subsequent clicks result in refinement of the bar view, unless a date dimension is added, at which time the view is changed to a line.</td>
</tr>
<tr>
<td>Line</td>
<td>Adding a measure and then a date dimension produces a line view. All subsequent clicks result in refinement of the line view.</td>
</tr>
<tr>
<td>Continuous Line</td>
<td>Adding a continuous dimension and then a measure produces a continuous line view. Subsequent dimensions result in refinement of the continuous line view. Subsequent measures add quantitative axes to the view.</td>
</tr>
<tr>
<td>Scatter</td>
<td>Adding a measure and then another measure produces a scatter view. Subsequent dimensions result in refinement to the scatter view. Subsequent measures will create a scatter matrix.</td>
</tr>
</tbody>
</table>
Adding a geographic field produces a map view with latitude and longitude as axes and the geographic field on the Level of Detail shelf. Subsequent dimensions add rows to the view while subsequent measures further refine the map by adding size and color encoding.

See Also

- Control the Appearance of Marks in the View on page 1123
- Change the Type of Marks in the View below
- Build a Basic View to Explore Your Data on page 118
- Build Common Chart Types in Data Views on page 844

Change the Type of Marks in the View

You can change the type of marks displayed in the view to fit your analysis. For example, you can change the marks from bars to lines. This article describes how to change the type of marks used in the view, and explains the different types of marks available.

Change the mark type

To change the mark type:
On the Marks card, click the Mark Type drop-down and select an option from the list.

For descriptions of each mark type, see the following sections:

- Automatic mark
- Bar mark
- Line mark
- Area mark
- Square mark
- Circle mark
- Shape mark
- Text mark
- Map mark
- Pie mark
- Gantt Bar mark
- Polygon mark

**Automatic mark**

When the Marks card drop-down menu is set to Automatic, Tableau automatically selects the best mark type for your data view. The mark type that is automatically selected is determined by the inner fields on the Rows and Columns shelves. The icon in the Marks card drop-down
menu indicates which type of mark was automatically selected. The mark types below are automatically selected for the following scenarios.

Text
The Text mark type is automatically selected when there are dimensions as inner fields on both the Rows and Columns shelves.

Shape
The Shape mark type is selected when there are measures as inner fields on both the Rows and Columns shelves.
Bar

The Bar mark type is selected when there is a dimension and a measure as inner fields on **Rows** and **Columns** shelves.
Line

The Line mark type is selected when there is a date field and a measure as the inner fields on the **Rows** and **Columns** shelves.
**Note:** You can override the default selection and use any mark type that provides insight into your data. However, you should exercise some caution when manually selecting a mark type because the resulting view might hide important information about your data.

---

**Bar mark**

The Bar mark type is useful when you want to compare measures across categories, or when you want to break data down into stacked bars. Tableau displays your data using bars when:

- The Marks card drop-down menu is set to **Automatic**, and you place a dimension and a measure as the inner fields on the **Rows** and **Columns** shelves. If the dimension is a date dimension, the Line mark is used instead.

- You select **Bar** from the Marks card drop-down menu.

Marks are automatically stacked.
**Note**: For views where the mark type is Bar and there are continuous (green) fields on both **Rows** and **Columns**, Tableau supports additional options and defaults for sizing the bar marks on the axis where the bars are anchored. See the Continuous axis mark sizing section in *Change the size of marks* on page 1125 for details.

The data view shown below displays a dimension as the columns of the table, and a measure as the rows of the table. It is also color-encoded by a dimension (**Category**). Because the Marks card drop-down menu is set to **Automatic**, data is displayed using bars.

To try some hands-on exercises for building bar charts, see *Build a Bar Chart* on page 847.

---

**Line mark**

The line mark type is useful when you want to see trends in data over time, your data are ordered, or interpolation makes sense. Tableau displays data using lines when:
The Marks card drop-down menu is set to **Automatic**, and you place one or more measures on either the **Columns** shelf or the **Rows** shelf, and then plot the measures against a date dimension or a continuous dimension.

- You select **Line** from the Marks card drop-down menu.

**Note:** In views that use the line mark type, you can use the **Path** property in the Marks card to change the type of line mark (linear, step, or jump), or to encode data by connecting marks using a particular drawing order. For details, see **Path properties** in the *Control the Appearance of Marks in the View* on page 1123.

The data view shown below displays a date dimension as the columns of the table and a measure as the rows of the table.

As the density of data increases, trends are often easier to see when using lines. This view shows 93 data points.
To try some hands-on exercises for building line charts, see Building Line Charts on page 889.

Area mark

The Area mark type is useful when you have a view where the marks are stacked and do not overlap. For more information about stacked marks, see Stack Marks on page 2194. In an area chart, the space between each line and the next line is filled with a color. This type of chart is not the best way to show specific values along the line but it can clearly show the total values so you can get an idea of how a dimension is contributing to an overall trend. Tableau displays your data using areas when:

- The Marks card drop-down menu is set to Automatic and one or more measures is plotted against a date dimension or continuous dimension. The Analysis > Stack Marks option must also be selected.

The view below shows a date dimension on the Columns shelf and a measure on the Rows shelf. Each line represents the sales for a product category. The lines have been stacked so that they do not overlap.
**Note:** You can add labels, edit the color legend, and highlight areas. In addition, you can turn on color effects, such as mark borders. For more information, see Control the Appearance of Marks in the View on page 1123.

---

**Square mark**

The square mark type is useful when you want to clearly see individual data points. When you select Square from the Marks card drop-down menu, Tableau displays your data using squares.

The view shown below displays several dimensions on both the rows and columns shelves. By default, the data is displayed using text. However, when you place a measure on Color, Tableau converts the marks to squares and creates a heat map. To try some hands-on exercises for building heat maps, see Build a Highlight Table on page 874.
When you add additional levels of detail by placing a dimension on **Detail, Color, Shape, Size,** or **Label** on the **Marks** card, the squares are displayed side-by-side and wrap to fill the cell. If the window is too small to show all of the squares, an ellipses indicates that there are more values than you can see.

You can also use the square mark to create treemaps. A treemap displays hierarchical data as a set of nested rectangles. To try a hands-on exercise for building a treemap, see **Build a Treemap** on page 920.

**Circle mark**

When you select **Circle** from the Marks card drop-down menu, Tableau displays your data using filled circles.

If the mark type was set to **Automatic**, Tableau would display the data using a shape (that is, an open circle).
Shape mark

The shape mark type is useful when you want to clearly see individual data points while also viewing categories associated with those points. Tableau displays your data using a shape when:

- The Marks card drop-down menu is set to **Automatic**, and you place one or more measures on both the **Rows** and the **Columns** shelves.
- You select **Shape** from the Marks card drop-down menu.

The view shown below displays the data from two measures. Because the Marks card drop-down menu is set to **Automatic**, the data are displayed using a shape.
By default, the shape used is an open circle. To select a different shape, click **Shape** on the Marks card. Twenty unique shapes are available:

To add more data to the view, you can place a field on **Shape** on the Marks card. Tableau separates the marks according to the values in the field. If the field is a dimension, each member is assigned a unique shape. If the field is a measure, the measure is automatically
binned into distinct buckets and each bucket is assigned a unique shape. The shape legend displays how shapes are distributed.

As shown below, the **Ship Mode** dimension is placed on **Shape** on the Marks card to encode each mark with information about how the order was shipped.

For details on how to build a scatterplot, see *Build a Scatter Plot* on page 907.

---

**Text mark**

The Text mark type is useful when you want to display the numbers associated with one or more dimension members. This type of view is often called a text table, a cross-tab, or a pivot table. Tableau displays your data using text when:

- The Marks card drop-down menu is set to **Automatic**, and you place one or more dimensions as the inner fields on both the **Rows** and the **Columns** shelves.
- You select **Text** from the Marks card drop-down menu.

Initially, the data is displayed as **Abc**.
To complete the view, you must place a field (typically a measure) on Text on the Marks card. In the view below, the Sales measure, which is aggregated as a sum, is used to complete the table.
Note: To create a cross-tab of any data view, select **Worksheet > Duplicate as Crosstab**.

When you add additional levels of detail by placing a dimension on **Detail, Color, Shape, Size**, or **Text** on the Marks card, the values are shown side-by-side and wrap to fill the cell. Allowing the text to stack up makes it possible to create word cloud visualizations like the one shown below.
If the application window is too small for all of the text values, some of the displayed text values will be truncated.

If the application window is too small for all of the text values, the cell displays an ellipses to indicate that there are more values than can be displayed. Number values that can't be displayed are indicated with pound signs (#).

For details on how to build a text table, see Build a Text Table on page 912.
Map mark

The Map mark type uses geocoding to fill a polygon or line with a color based on data. The primary use of the map mark type is for creating polygon or line maps. The area to be filled is defined by the geographic fields used in the view. Tableau displays data using a filled map when:

- A geographic dimension is on **Detail** on the Marks card, along with **Latitude** and **Longitude** on the **Rows** and **Columns** shelves. A measure or continuous dimension is added to **Color** on the Marks card.
- You select **Map** from the Marks card drop-down menu.

The data view below shows a map with the **State** geographic dimension on **Detail** on the Marks card. The **Profit** measure is on **Color** on the Marks card.

The polygon for each state is filled with a color based on the profit for that state. The profit in Texas is low (in fact, negative) while the profit in California is high.
You can also use the Map mark type when you are plotting two measures against each other. By default, placing measures on both the **Rows** and **Columns** shelves creates a scatter plot. However, adding a geographic dimension allows you to change the mark type to Map. Each mark becomes the area defined by the geographic field. For example, the view below shows **Sales versus Profit** for a selection of states. Rather than showing a round mark with a label for each state, the view uses the Map mark type to draw the outline of each state. While this works well for recognizable areas, it is not the best choice for areas that are similar in shape or difficult to recognize.

For more details on building maps, see [Get Started Mapping with Tableau on page 1862](#) and [Geographic Data Analysis in Tableau on page 1862](#).

---

**Pie mark**

You can use the Pie mark to show proportions. Although generally this type of information can be better shown using stacked bar charts, there are cases where pie marks can be very effective—for example, when trying to convey the percentage allocation of marketing expenses by state where the spending of geographically close states are very relevant.
Tableau will never use the Pie mark as an automatic mark type, but you can select **Pie** from the Marks card drop-down menu.

When you select the Pie mark type, an additional target named Angle is displayed on the Marks card. The Angle target determines the angular measure of the pie wedges. For example, if you place a measure such as **Sales** on **Angle** on the Marks card, the 360 degrees of the pie corresponds to the total sum of sales, and each wedge is divided by the values of the field on **Color** on the Marks card.

The view below shows the time it took to ship products by various ship modes. The data overlays a map and shows data by state. We can see that standard class shipping takes the longest in most areas.

For details on how to build and use pie charts, see **Build a Pie Chart** on page 903 and **Get Started Mapping with Tableau** on page 1862.
• The Marks card drop-down menu is set to **Automatic** and you place one or more dimensions on either the **Columns** shelf or the **Rows** shelf, and then plot the dimensions against a continuous quantity.

• You select **Gantt Bar** from the Marks card drop-down menu.

The distinguishing characteristic of Gantt Bars is that the length of every mark is proportional to the measure placed on **Size** on the Marks card.

The view below displays a dimension as a function of a continuous date. If the Marks card drop-down menu is set to **Automatic**, the data would be displayed using bars. Selecting **Gantt Bar** and adding additional fields will display a view like the following.

In particular, placing the **Days to Ship** measure on **Size** on the Marks card causes every bar in the view to be drawn with a length that indicates the delivery time of an order. Additionally, placing the **Ship Mode** dimension on **Color** on the Marks card causes each bar to be colored according to the ship mode.

For details on how to build Gantt Bar chart, see **Build a Gantt Chart** on page 867.
Polygon mark

Polygons are points connected by lines enclosing an area. The polygon mark type is useful when you want to connect points to create data areas. Tableau displays data using polygons when you select Polygon from the Marks menu.

The polygon mark is not commonly used and often requires a specially constructed data source.

The view shown below comes from a specially constructed data source that holds geographic and election data. It displays the 48 contiguous US states as a function of latitude and longitude and color-encodes each state by the 2000 presidential election results.

If Marks is set to Automatic, the data will be displayed using a shape. Manually selecting Polygon and adding additional fields to the view causes a different view to be created.

![Map of the United States with states color-coded by election results.](image)

Every state is considered to be a polygon in the data source. The PolygonID field on the Detail target is distinct for each US state. You can remove states from the view by filtering this field.

Additionally, you can specify the drawing order of the lines that constitute each polygon by placing a field on the Path target. In this example, the PointOrder measure is used to draw each state.
Control the Appearance of Marks in the View

Control color, size, shape, detail, text, and tooltips for marks in the view using the Marks card. Drag fields to buttons on the Marks card to encode the mark data. Click the buttons on the Marks card to open Mark properties.

In this article

- Assign colors to marks below
- Change the size of marks on page 1125
- Add labels or text for marks on page 1131
- Separate marks in the view by dimension members on page 1132
- Add tooltips to marks on page 1133
- Change the shape of marks on page 1137
- Draw paths between marks on page 1143

Assign colors to marks

To assign a color to marks in the view, do one of the following:

- On the Marks card, click Color, and then select a color from the menu.
  
  This updates all marks in the view to the color you choose. All marks have a default color, even when there are no fields on Color on the Marks card. For most marks, blue is the default color; for text, black is the default color.

- From the Data pane, drag a field to Color on the Marks card.
  
  Tableau applies different colors to marks based on the field's values and members. For example, if you drop a discrete field (a blue field), such as Category, on Color, the marks in the view are broken out by category, and each category is assigned a color.
If you drop a continuous field, such as SUM(sales), on Color, each mark in the view is colored based on its sales value.
Edit colors

To change the color palette or customize how color is applied to your marks:

- On the Marks card, click **Color > Edit Colors**.

For more information, see **Color Palettes and Effects** on page 1145.

Change the size of marks

To change the size of marks in the view, do one of the following:

- On the Marks card, click **Size**, and then move the slider to the left or right.

![Marks card](image)

The Size slider affects different marks in different ways, as described in the following table.

<table>
<thead>
<tr>
<th>Mark Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circle, Square, Shape, Text</td>
<td>Makes the mark bigger or smaller.</td>
</tr>
<tr>
<td>Bar, Gantt Bar</td>
<td>Makes bars wider or narrower.</td>
</tr>
<tr>
<td>Line</td>
<td>Makes lines thicker or thinner.</td>
</tr>
<tr>
<td>Polygon</td>
<td>You cannot change the size of a polygon.</td>
</tr>
<tr>
<td>Pie</td>
<td>Makes the overall size of the pie bigger and smaller.</td>
</tr>
</tbody>
</table>
The size of your data view is not modified when you change marks using the **Size** slider. However, if you change the view size, the mark size might change to accommodate the new formatting. For example, if you make the table bigger, the marks might become bigger as well.

- **From the Data pane, drag a field to Size on the Marks card.**

![Tableau interface showing Size slider and marks with sizes](image)

When you place a discrete field on **Size** on the **Marks** card, Tableau separates the marks according to the members in the dimension, and assigns a unique size to each member. Because size has an inherent order (small to big), categorical sizes work best for ordered data like years or quarters.

Size-encoding data with a discrete field separates the marks in the same way as the **Detail** property does, and then provides additional information (a size) for each mark. When you add categorical size encoding to a view, Tableau displays a legend showing the sizes assigned to each member in the field on the **Size** target. You can modify how these sizes are distributed using the **Edit Sizes** dialog box.

When you place a continuous field on **Size** on the **Marks** card, Tableau draws each mark with a different size using a continuous range. The smallest value is assigned the smallest sized mark and the largest value is represented by the largest mark.
When you add quantitative size encoding to the view, Tableau displays a legend showing the range of values over which sizes are assigned. You can modify how these sizes are distributed using the Edit Sizes dialog box.

**Edit marks sizes**

To edit the size of marks, or change how size is being applied to marks in the view:

1. On the Size legend card (which appears when you add a field to Size on the Marks card), click the drop-down arrow in the right-hand corner and select **Edit Sizes**.

2. In the Edit Sizes dialog box that appears, make your changes and then click **OK**.

   The options available depend on whether the field being applied to Size is a continuous or discrete field.

   **For continuous fields, you can do the following:**

   - **For Sizes vary**, click the drop-down box and select one of the following:
     - **Automatically** - Selects the mapping that best fits your data. If the data is numeric and does not cross zero (all positive or all negative), the From zero mapping is used. Otherwise, the By range mapping is used.
     - **By range** - Uses the minimum and maximum values in the data to
determine the distribution of sizes. For example, if a field has values from 14 to 25, the sizes are distributed across this range.

- **From zero** - Sizes are interpolated from zero, assigning the maximum mark size to the absolute value of the data value that is farthest from zero.

- Use the range slider to adjust the distribution of sizes. When the From zero mapping is selected from the Sizes vary drop-down menu, the lower slider is disabled because it is always set to zero.

- Select **Reversed** to assign the largest mark to the smallest value and the smallest mark to the largest value. This option is not available if you are mapping sizes from zero because the smallest mark is always assigned to zero.

- To modify the distribution of sizes, select the **Start value in legend** and **End value for range** check boxes and type beginning and end values for the range.

For discrete fields, you can do the following:

- Use the range slider to adjust the distribution of sizes.

- Select **Reversed** to assign the largest mark to the smallest value and the
smallest mark to the largest value.

Continuous axis mark sizing

For views where the mark type is **Bar** and there are continuous (green) fields on both **Rows** and **Columns**, Tableau supports additional options and defaults for sizing the bar marks on the axis where the bars are anchored.

- The bar marks in histograms are continuous by default (with no spaces between the marks), and are sized to match the size of the bins. See **Build a Histogram** on page 881 for an example.

- When there is a field on **Size**, you can determine the width of the bar marks on the axis where the bars are anchored by using the field on **Size**. To do this, click the **Size** card and select **Fixed**.
When there is no field on **Size**, you can specify the width of the bar marks on the axis where the bars are anchored in axis units. To do this, click the **Size** card, choose **Fixed**, and then type a number in the **Width in axis units** field.
When there is a continuous date field on the axis where the bars are anchored, the width of the marks is set to match the level of the date field. For example, if the level of the continuous date field is MONTH, the bars are exactly one month wide—that is, slightly wider for 31-day months than for 30-day months. You can configure the width of the bars by clicking the **Size** card, choosing **Fixed**, and then typing a number in the **Width in days** field, but the resulting bar widths don't take into account the varying lengths of time units such as months and years.

---

**Add labels or text for marks**

To add mark labels or text to the visualization:

- From the **Data** pane, drag a field to **Label** or **Text** on the Marks card.

When working with a text table, the Label shelf is replaced with Text, which allows you to view the numbers associated with a data view. The effect of text-encoding your data view depends on whether you use a dimension or a measure.
• Dimension – When you place a dimension on Label or Text on the Marks card, Tableau separates the marks according to the members in the dimension. The text labels are driven by the dimension member names.

• Measure – When you place a measure on Label or Text on the Marks card, the text labels are driven by the measure values. The measure can be either aggregated or disaggregated. However, dis-aggregating the measure is generally not useful because it often results in overlapping text.

Text is the default mark type for a text table, which is also referred to as a cross-tab or a PivotTable.

Separate marks in the view by dimension members

To separate marks in the view (or add more granularity):

• From the Data pane, drag a dimension to Detail on the Marks card.
When you drop a dimension on **Detail** on the Marks card, the marks in a data view are separated according to the members of that dimension. Unlike dropping a dimension on the **Rows** or **Columns** shelf, dropping it on **Detail** on the Marks card is a way to show more data without changing the table structure.

---

### Add tooltips to marks

Tooltips are details that appear when you hover over one or more marks in the view. Tooltips are also convenient for quickly filtering or removing a selection, or viewing underlying data. You can edit a tooltip to include both static and dynamic text. You can also modify which fields are included in a tooltip and whether you want to be able to use those fields to select marks in the view.

For details on showing a visualization from a worksheet in a tooltip (Viz inTooltip), **Create Views in Tooltips (Viz in Tooltip)** on page 1773.

#### Add a tooltip

1. Drag a field to **Tooltip** on the **Marks** card.

2. Click **Tooltip** on the **Marks** card to open the Edit Tooltip dialog box, where you can add text, rearrange the tooltip contents, and insert more fields.
Note: Dimensions are added to the tooltip using the ATTR aggregation, which means the tooltip may sometimes display as an asterisk. The asterisk indicates that there are multiple dimension members that apply to the mark you are pointing at. For example, a mark may represent the aggregated sales for all regions. Adding the Region field to the tooltip results in an asterisk because the mark represents more than one region. To avoid showing an asterisk, add the dimension to Detail on the Marks card or use it elsewhere in the view to ensure the marks are at the same level of detail.

Tooltip options

After you open the Edit Tooltip dialog box, there are several options that you can choose from to format the tooltips in your view and configure their behavior. You can choose from the options below.
• **Format tooltips:** Tooltips are specified on a per-sheet basis and can be formatted using the tools on the top of the Edit Tooltip dialog box.

• **Adding dynamic text:** Use the *Insert* menu at the top of the dialog box to add dynamic text such as field values, sheet properties, and more. The *All Fields* command on the *Insert* menu adds all field names and values that are used in the view to the tooltip for any mark in the view.

• **Show tooltips:** Tooltips are shown by default. If you prefer to hide automatic tooltips, clear the *Show Tooltips* check box.

• **Configure tooltip behavior:** Configure how tooltips behave by selecting one of the following options in the drop-down list next to the *Show Tooltips* check box:
  
  • **Responsive - Show tooltips instantly:** Select this option to show tooltips instantly as you move the cursor over the marks in the view. This is the default for all views.

    With this option, tooltips appear without command buttons. You must first click a mark in the view to see the command buttons.

  • **On Hover - Show tooltips on hover:** Select this option to show tooltips only after you rest the cursor on a mark. However, with this option, command buttons appear on the tooltip without any further action from you (unless you specify otherwise in the Edit Tooltips dialog box).

  • **Show commands:** Select the *Include command buttons* check box to add *Keep*
Only, Exclude, Group Members, Create Set and View Data buttons at the top of the tooltip. These buttons show both in Tableau Desktop and when the view is published to the web or viewed on a mobile device.

For more information about grouping members, see Correct Data Errors or Combine Dimension Members by Grouping Your Data on page 1002. For more information about creating sets, see Create Sets on page 1004.

- **Select marks in a view from the tooltip.** Select the Allow selection by category check box to be able to select marks in a view that have the same value by clicking on a discrete field in a tooltip. If this check box is selected, you can use this feature to select marks in a view in Tableau Desktop, when the view is published to the web or viewed on a mobile device.

- **Add or remove fields:** To add and remove fields when using the automatic tooltip, right-click (control-click on Mac) the field on one of the shelves in the view and select Include in Tooltip.

![Tableau View with Automatic Tooltip Settings](image)

**Note:** The Include in Tooltip option is only available if you have not customized the tooltip. If you have customized the tooltip, you can return to the automatic tooltip by clicking Tooltip on the Marks card and then clicking Reset in the Edit Tooltip dialog box.

Back to top
Change the shape of marks

To change the shapes of marks:

- From the **Data** pane, drag a field to **Shape** on the Marks card.

When you place a dimension on **Shape** on the **Marks** card, Tableau separates the marks according to the members in the dimension, and assigns a unique shape to each member. Tableau also displays a shape legend, which shows each member name and its associated shape. When you place a measure on **Shape** on the **Marks** card, the measure is converted to a discrete measure.

Shape-encoding data separates the marks in the same way as the Detail property does, and then provides additional information (a shape) for each mark. Shape is the default mark type when measures are the inner most fields for both the **Rows** shelf and the **Columns** shelf.

In the view below, the marks are separated into different shapes according to the members of the **Customer Segment** dimension. Each shape reflects the customer segment’s contribution to profit and sales.
Edit shapes

By default, ten unique shapes are used to encode dimensions. If you have more than 10 members, the shapes repeat. In addition to the default palette, you can choose from a variety of shape palettes, including filled shapes, arrows, and even weather symbols.

1. Click **Shape** on the **Marks** card, or select **Edit Shape** on the legend’s card menu.
2. In the Edit Shape dialog box, select a member on the left and then select the new shape in the palette on the right. You can also click **Assign Palette** to quickly assign the shapes to the members of the field.

Select a different shape palette using the drop-down menu in the upper right.

**Note:** Shape encodings are shared across multiple worksheets that use the same data source. For example, if you define Furniture products to be represented by a square, they will automatically be squares in all other views in the workbook. To set the default shape encodings for a field, right-click (control-click on Mac) the field in the **Data** pane and select **Default Properties > Shape**.
Use Custom shapes

You can add custom shapes to a workbook by copying shape image files to the Shapes folder in your Tableau Repository, which is located in your Documents folder. When you use custom shapes, they are saved with the workbook. That way the workbook can be shared with others.

1. Create your shape image files. Each shape should be saved as its own file and can be in any of several image formats including bitmap (.bmp), portable network graphic (.png), .jpg, and graphics interchange format (.gif).

2. Copy the shape files to a new folder in the My Tableau Repository\Shapes folder in your Documents folder. The name of the folder will be used as the name of the palette in Tableau. In the example below, two new palettes are created: Maps and My Custom Shapes.

3. In Tableau, click the drop-down arrow on the shape legend, and select Edit Shape.
4. Select the new custom palette in the drop-down list. If you modified the shapes while Tableau was running, you may need to click **Reload Shapes**.

5. You can either assign members shapes one at a time, or click **Assign Palette** to automatically assign the shapes to the members.
Note: You can return to the default palette by clicking the Reset button. If you open a workbook that uses custom shapes that you don’t have, the workbook will show the custom shapes because the shapes are saved as part of the workbook. However, you can click Reload Shapes in the Edit Shapes dialog box to use the ones in your repository instead.

Below are some examples of views that use both the default and custom shape palettes.
Tips for creating custom shapes

When you create custom shapes there are a few things that you can do to improve how your shapes look and function in the view. If you are creating your own shapes, we recommend following general guidelines for making icons or clip art.

- **Suggested size** - Unless you plan on using Size to make the shapes really large, you should try to make your original shape size close to 32 pixels by 32 pixels. However, the original size depends on the range of sizes you want available in Tableau. You can resize the shapes in Tableau by clicking Size on the Marks card, or by using the cell size options on the Format menu.

- **Adding color encoding** - If you plan to also use Color to encode shapes, you should use a transparent background. Otherwise, the entire square of the image will be colored rather than just the symbol. GIF and PNG file formats both support transparency. GIF files support transparency for a single color that is 100% transparent, while PNG files support alpha channels with a range of transparency levels available on every pixel in the image. When Tableau color encodes a symbol, the amount of transparency for each pixel won’t be modified, so you can maintain smooth edges.
**Note**: Avoid including too much transparency around an image. Make the size of the custom shape as close to the size of the image as possible. Extra transparent pixels around the edges of the image can negatively affect the hover or click behavior near the image, especially when custom shapes overlap each other. When the actual shape area is bigger than what is visible, it can make hovering and clicking the shape more difficult and less predictable for users.

- **File formats** - Tableau doesn’t support symbols that are in the Enhanced Meta File format (.emf). The shape image files can be in one of the following formats: .png, .gif, .jpg, .bmp, and .tiff.

**Draw paths between marks**

The Path property is available when the **Line** or **Polygon** mark type is selected in the **Marks** card drop-down menu. For more information about how to change the type of mark displayed in your viz, see *Change the Type of Marks in the View* on page 1101.

You can use the Path property in the Marks card to change the type of line mark (linear, step, or jump), or to encode data by connecting marks using a particular drawing order. You can path-encode your data using either a dimension or a measure.

- **Dimension** – When you place a dimension on **Path** on the Marks card, Tableau connects the marks according to the members in the dimension. If the dimension is a date, the drawing order is driven by the date order. If the dimension includes words such as customer names or product types, the drawing order is driven by the order of the members in the data source. You can change the order that data points are connected by changing the sort order of the members. For more information, see *Sort Data in a Visualization* on page 1202.

- **Measure** – When you place a measure on **Path** on the Marks card, Tableau connects the marks according to the values of the measure. The measure can be aggregated or disaggregated.

**Change the line type (linear, step, jump)**

When the mark type is set to be a line (Automatic or Line), you can click the **Path** property in the Marks card to change the line type.
Use these line types for numeric data that remains constant for periods of time, with noticeable changes or deltas – such as account balances, inventory levels, or interest rates. Step lines work well for emphasizing the magnitude of change. Jump lines help to emphasize the duration of change between data points.

1. Click **Path** in the Marks card.

2. Select **Linear**, **Step**, or **Jump** to change the line type.

Create a path-encoded view

To create a useful path-encoded view, your data table should contain at least one measure. You can't create a path that connects only categorical data (dimensions).

The view below was created using storm data from the Atlantic basin in 2010. The view uses line marks with the path determined by the date of the storm. This lets you see the path of the storm. By placing the continuous date on **Path** on the **Marks** card, this tells Tableau to draw the lines in chronological order. For more information, see Create Maps that Show a Path Over
Color Palettes and Effects

All marks have a default color, even when there are no fields on Color on the Marks card. For most marks, blue is the default color; for text, black is the default color. Also see Assign colors to marks on page 1123 and Example – Multiple Fields on Color on page 1156.

In this article

- Categorical palettes
- Quantitative palettes
- Configure color effects
Categorical Palettes

When you drop a field with discrete values (typically a dimension) on Color on the Marks card, Tableau uses a categorical palette and assigns a color to each value of the field. Categorical palettes contain distinct colors that are appropriate for fields with values that have no inherent order, such as departments or shipping methods.

To change colors for values of a field, click in the upper-right corner of the color legend. In Tableau Desktop, select **Edit Colors** from the context menu. In Tableau Server or Tableau Online, the Edit Colors dialog opens automatically.

**Tableau Desktop version**

**Web version**

![Edit Colors Dialog](image)

Change the color for a value

1. Click on an item on the left, under **Select Data Item**.
2. Click a new color in the palette on the right. In Tableau Desktop you can hover over a swatch to identify the color.
3. Repeat for as many values that you want to change.
4. In Tableau Desktop, click **OK** to exit the Edit Colors dialog box. In Tableau Server or Tableau Online, simply close the dialog box.
Select a different palette

The Select Color Palette drop-down list in the Edit Colors dialog box provides color palettes that you can use for discrete fields. The list contains both categorical and ordinal palettes.

At the top of the list are categorical palettes, such as Tableau 10. As noted above, categorical palettes are appropriate for discrete fields with no inherent order.

At the bottom of the list are ordinal palettes such as Orange. Ordinal palettes contain a range of related colors and are appropriate for fields that have an associated order, such as dates or numbers.

After you select a palette, click Assign Palette to automatically assign the new palette colors to the members in the field.

To return to the Automatic palette and the default color assignments, click Reset in the Edit Colors dialog box.

Quantitative Palettes

When you drop a field with continuous values on the Marks card (typically a measure), Tableau displays a quantitative legend with a continuous range of colors.

You can change the colors used in the range, the distribution of color, and other properties. To edit colors, click in the upper right of the color legend. In Tableau Desktop, select Edit Colors from the context menu. In Tableau Server or Tableau Online, the Edit Colors dialog opens automatically.

When there are both negative and positive values for the field, the default range of values will use two color ranges and the Edit Colors dialog box for the field has a square color box on either end of the range. This is known as a diverging palette.
When all values are either positive or negative, the default range of values will use a single color range and the Edit Colors dialog box for the field has a square color box only at the right end of the range. This is known as a sequential palette.

You can specify whether Tableau uses a diverging or a sequential palette for a continuous field on Color, and also configure the range of colors for the field’s values.

The Palette drop-down list provides a range of color palettes from which you can choose. There are two types of quantitative palettes available for continuous fields:
• All palettes with *Diverging* in the name are diverging quantitative palettes—for example, *Orange-Blue Diverging*. You can choose a diverging palette for any continuous field—it isn’t necessary for the range of values to contain both positive and negative numbers.

To change the colors for a diverging palette, click one of the square color boxes at either end of the palette spectrum. Depending on whether you are authoring in Tableau Desktop or on the web, do one of the following:

- In Tableau Desktop, in the color configuration dialog box (which is part of your computer’s operating system), select a color from the color picker or enter custom values.
- In Tableau Server or Tableau Online, enter a custom Hex value in the **Custom Color** field. If the value isn’t valid, no changes are made.

• All other palettes are sequential quantitative palettes. To change the colors for a sequential palette, click the square color box at the right end of the palette spectrum to either open the color configuration dialog box (Tableau Desktop), or enter a custom Hex value in the **Custom Color** field (Tableau Server or Tableau Online).

---

**Options for quantitative palettes**

The following options are available in the Edit Colors dialog box for a continuous field.

**Note:** Options differ where noted for Tableau Server and Tableau Online.

**Stepped Color**

Select **Stepped Color** to group values into uniform bins, where each bin is associated with a color. Use the spin control to specify how many steps (bins) to create. For example, for a range of values from 0 to 100 you could specify five steps to sort values into five bins (0-20, 20-40, etc.).

| Tableau Desktop version | Web version |
If a diverging color palette is selected, the point where the palette transitions between colors is shown on the color ramp with a small black tick mark. When the number of steps is odd, the mark is placed in the middle of the transitional step. When the number of steps is even, the mark is placed at the boundary between the steps where the color changes.

Reversed

Select **Reversed** to invert the order of colors in the range. For example, if you want lower values to have a darker intensity in a sequential palette, reverse the palette. For a diverging palette, reversing the color palette means swapping the two colors in the palette, in addition to inverting the shades within each color range.

Use Full Color Range

With a diverging (two-color) palette, you can select to **Use Full Color Range**. Tableau assigns both the starting number and the ending number a full intensity for both color ranges. So if the range is from -10 to 100, the color representing negative numbers will be adjusted to change in shade much more quickly than the color representing positive numbers. If you don’t select **Use Full Color Range**, Tableau assigns the color intensity as if the range of values was from -100 to 100, so that the change in shade is the same on both sides of zero. This means there will be much less change on the negative side, where actual values only range from -10 to 0, than on the positive side, where values range from 0 to 100.

The image on the left below shows a red-green diverging color palette for values from -858 to 72,986. Without using the full color range, -858 (associated with the small box at the lower right of the chart) shows as gray, because -858 is only about 1% as far to the negative side as
72,986 is to the positive side. When the full color range is used, as in the image on the right, -858 shows as a dark red, equal in intensity to the maximum positive value.

Include totals

Select Include Totals to include totals, sub-totals and grand totals in the color encoding. This option only applies when total values are included in the view.

Limit the color range

In Tableau Desktop, when you click Advanced in the Edit Colors dialog box, you can choose to specify the start, end, and center values for the range by selecting the check box and typing a new value into the field and the color ramp is adjusted accordingly.

The Start value is the lower limit in the range, the End value is the upper limit, and the Center value is where the neutral color is located on a diverging color palette.

**Note:** This option is not currently available in Tableau Server or Tableau Online.

Reset the color range

To return to the Automatic palette and the default color assignments, click Reset in the Edit Colors dialog box.

**Note:** If you are in web authoring mode and click Reset, the color palette will return to the default settings. Any options that were set in the Advanced option will also be reset. To undo this action, you can click Undo in the top menu. If your changes have already been saved, you must change the Advanced options in Tableau Desktop and republish the view.

Back to top
Configure Color Effects

Click the **Color** drop down on the **Marks** card to configure additional **Color** settings not related to the actual colors shown.

### Tableau Desktop version

![Tableau Desktop version](image)

### Web version

![Web version](image)

#### Opacity

Modify the opacity of marks by moving the slider.

Adjusting opacity is especially useful in dense scatter plots or when you are looking at data overlaying a map or background image. As you slide the slider toward the left, marks become more transparent.

#### Mark borders

By default, Tableau displays all marks without a border. You can turn on mark borders for all mark types except text, line, and shape. On the **Color** drop-down control, select a mark border color.

### Tableau Desktop version

![Tableau Desktop version](image)

### Web version

![Web version](image)
Borders can be useful for visually separating closely spaced marks. For example, the views below show a scatterplot with mark borders turned on (left) and turned off (right). When borders are turned on, marks are easier to distinguish in areas where they are tightly clustered.

**Note:** You can also use the opacity setting to show the density of marks.
When you are viewing a large number of color-encoded small marks, it is usually better to leave mark borders off. Otherwise borders can dominate the view, making it difficult to see the color encoding.

For example, the views below show bars that are segmented by a large number of color-encoded dimension members. With mark borders turned on (right), some of the narrower marks are difficult to identify by color. With borders turned off (left), the marks are easy to distinguish.

Mark halos

To make marks more visible against a background image or map, surround each mark with a solid contrasting color called a halo. Mark halos are available when you have a background image or a background map. On the Color drop-down control, select a mark halo color.
Markers

In Tableau Desktop, when you are using the Line mark type, you can add a marker effect to show or hide points along the line. You can show selected points, all points, or no points. On the Color drop-down control, select a marker in the Effects section.

Note: This option is not currently available in Tableau Server or Tableau Online.
Example - Multiple Fields on Color

If you drop one field on Color and then drop a different field on Color, the second field replaces the first field. However, depending on the chart type, for example treemaps and bullet graphs, you can put multiple fields on color. You can use one field to set the hue, and the other to show gradations within that hue.

Follow these steps, using the Sample - Superstore data source, to build a treemap with two fields on Color.

1. Drag **Category** and **Sub-Category** to **Columns**.
2. Drag **Sales** to **Size** on the **Marks** card.
3. Click **Show Me** in the toolbar, then select the Treemap chart type.
Tableau moves all fields to the **Marks** card, putting **SUM(Sales)** on both **Size** and **Color**, and **Category** and **Sub-Category** on **Label**.
4. Click the label icon to the left of **Category** on the **Marks** card and select **Color**:

**Category** replaces **SUM(Sales)** on **Color**. The marks are still sized by the sum of **Sales**, but now they are colored by **Category**:
5. Click the label icon to the left of **Sub-Category** on the Marks card and select **Color**. Tableau uses distinct, categorical colors for the first field, **Category**, and a range of sequential shades to distinguish values for the second field, **Sub-Category**.
The size of the individual rectangles is still determined by **Sales, per Category and Sub-Category**.

The two fields on **Color (Category and Sub-Category)** are related within a hierarchy, so if you swap their positions on the **Marks** card, moving **Sub-Category** to be above **Category**, the effect is the same as if you had removed **Category** from the view altogether. The treemap changes to show a uniquely colored rectangle for each **Sub-Category**:

When the two fields on Color are not related within a hierarchy, you can switch the order of the fields on the Marks card so that the field that was used for categorical colors was used for sequential shades, and vice versa.

If you aren’t satisfied with the colors that Tableau used, you can change them. To open the Edit Colors dialog, do one of the following:

- In Tableau Desktop, double-click the color legend.
- In Tableau Server or Tableau Online, click the drop-down arrow in the top right-hand corner of the legend.
6. Make the view more readable by adding **Category**, **Sub-Category**, and **Sales** to **Label**. Users can hover to see tooltips for any rectangle that is too small to show text by default.

---

**Filter and Sort Data in the View**

This section describes the various ways you can filter and sort data in Tableau visualizations. Read the following articles for information on how to filter data from your visualizations and across multiple worksheets and data sources, sort data in the visualization, create nested sorts, and more.

**Watch a video:** For a 3-minute introduction to filtering in Tableau, watch the Ways to **Filter** free training video. Use your **tableau.com** account to sign in.

For a 4-minute walk through of the various ways to sort in Tableau, watch the **Sorting**
free training video. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Filter Data from Your Views

Filtering is an essential part of analyzing data. This article describes the many ways you can filter data from your view. It also describes how you can display interactive filters in the view, and format filters in the view.

Watch a Video: To see related concepts demonstrated in Tableau, watch these free training videos: Ways to Filter (2 minutes), Where Tableau Filters (4 minutes), Using the Filter Shelf (7 minutes), Interactive Filters (4 minutes), Additional filtering topics (7 minutes), and Filtering for Top Across Panes (6 minutes). Use your tableau.com account to sign in.

In this article

Filtering Order of Operations below
Select to keep or exclude data points in your view on the next page
Select headers to filter data on page 1164
Drag dimensions, measures, and date fields to the Filters shelf on page 1165: Categorical data | Quantitative data | Dates | Table Calculations
Display interactive filters in the view on page 1176
Set options for filter card interaction and appearance on page 1178: Filter card modes | Customize filter cards

Filtering Order of Operations

Before you begin filtering data in Tableau, it's important to understand the order in which Tableau executes filters in your workbook.

Tableau performs actions on your view in a very specific order; this is called the Order of Operations. Filters are executed in the following order:
1. Extract filters
2. Data source filters
3. Context filters
4. Filters on dimensions (whether on the Filters shelf or in filter cards in the view)
5. Filters on measures (whether on the Filters shelf or in filter cards in the view)

To learn more, see Tableau's Order of Operations on page 267.

Note: When you drag a discrete dimension to the Filters shelf, the Filter dialog box offers four tabs for filtering: General, Wildcard, Condition, and Top. The settings on each of these tabs are additive starting with the General tab; what you set on each tab will affect the filter results on each tab to the right. For details on the Filter dialog box, see Drag dimensions, measures, and date fields to the Filters shelf on page 1165 and Filter categorical data (dimensions) on page 1166.

Select to keep or exclude data points in your view

You can filter individual data points (marks), or a selection of data points from your view. For example, if you have a scatter plot with outliers, you can exclude them from the view so you can better focus on the rest of the data.

To filter marks from the view, select a single mark (data point) or click and drag in the view to select several marks. On the tooltip that appears, you can:
- Select **Keep Only** to keep only the selected marks in the view.

- Select **Exclude** to remove the selected marks from the view.

**Note:** These filtering options are not available if a Wildcard Match filter is already specified for the same field. See Drag dimensions, measures, and date fields to the Filters shelf on the next page to learn more about Wildcard Match filters.

Select headers to filter data

You can also select headers to filter them from your view.

To filter entire rows or columns of data from your view, select the header in the view. On the tooltip that appears, select to **Exclude** or **Keep Only** the selected data.
When you select a table header that is part of a hierarchy, all of the next level headers are also selected. For example, the view shown below consists of two unrelated dimensions placed on the **Columns** shelf, and two levels of the same hierarchy placed on the **Rows** shelf.

The selected row headers include the Furniture member of the **Category** dimension, and the Binders and Labels members of the **Sub-category** dimension. When Furniture is selected, all members from the next (inner) level in the hierarchy are automatically selected. In this case, that means the Bookcases, Chairs, Furnishings, and Tables members.

Drag dimensions, measures, and date fields to the Filters shelf

Another way to create a filter is to drag a field directly from the Data pane to the Filters shelf.

In Tableau Desktop, when you add a field to the Filters shelf, the Filter dialog box opens so you can define the filter. The Filter dialog box differs depending on whether you are filtering categorical data (dimensions), quantitative data (measures), or date fields.

In web authoring, when you add a field to the Filters shelf, an interactive filter appears in the view.
Filter categorical data (dimensions)

Dimensions contain discrete categorical data, so filtering this type of field generally involves selecting the values to include or exclude.

**In Tableau Desktop**

When you drag a dimension from the Data pane to the Filters shelf in Tableau Desktop, the following Filter dialog box appears:
There are four tabs in the dialog box:

- **General**: Use the General tab to select the values you want to include or exclude.

- **Wildcard**: Use the Wildcard tab to define a pattern to filter on. For example, when filtering on email addresses you might want to only include emails from a specific domain. You can define a wildcard filter that ends with "@gmail.com" to only include Google email addresses.

- **Condition**: Use the Condition tab in the Filter dialog box to define rules to filter by. For example, in a view showing the average Unit Price for a collection of products, you may want to only show the Products that have an average unit price that is greater than or equal to $25. You can use the built-in controls to write a condition or you can write a custom formula.

- **Top**: Use the Top tab in the Filter dialog box to define a formula that computes the data that will be included in the view. For example, in a view that shows the average Time to
Ship for a collection of products, you can decide to only show the top 15 products by Sales. Rather than having to define a specific range for Sales (e.g., greater than $100,000), you can define a limit (top 15) that is relative to the other members in the field (products).

**Important Note:** Each tab adds additional definitions to your filter. For example, you can select to exclude values under the General tab, and also add limits under the Top tab. Selections and configurations from both tabs are applied to your filter.

At any time, you can see the definitions of your filter under Summary on the General tab.

**In web authoring**

When you drag a dimension from the Data pane to the Filters shelf in web authoring, an interactive filter appears in the view.

You can then select the items you want to include or exclude in the view.

<table>
<thead>
<tr>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔ (All)</td>
</tr>
<tr>
<td>✔ Furniture</td>
</tr>
<tr>
<td>✔ Office Supplies</td>
</tr>
<tr>
<td>✔ Technology</td>
</tr>
</tbody>
</table>

**Filter quantitative data (measures)**

Measures contain quantitative data, so filtering this type of field generally involves selecting a range of values that you want to include.

**In Tableau Desktop**

When you drag a measure from the Data pane to the Filters shelf in Tableau Desktop, the following dialog box appears:
Select how you want to aggregate the field, and then click **Next**.

In the subsequent dialog box, you’re given the option to create four types of quantitative filters:

**Range of Values**: Select the Range of Values option to specify the minimum and maximum values of the range to include in the view. The values you specify are included in the range.

**At Least**: Select the At Least option to include all values that are greater than or equal to a specified minimum value. This type of filter is useful when the data changes often so specifying an upper limit may not be possible.

**At Most**: Select the At Most option to include all values that are less than or equal to a specified maximum value. This type of filter is useful when the data changes often so specifying a lower limit may not be possible.

**Special** (Tableau Desktop only): Select the Special option to filter on Null values. Include only Null values, Non-null values, or All Values.

**Note**: If you have a large data source, filtering measures can lead to a significant degradation in performance. It is sometimes much more efficient to filter by creating a set containing the measure and then apply a filter to the set. For more information about creating sets, see **Create Sets** on page 1004.
In web authoring

When you drag a measure from the Data pane, to the Filters shelf in web authoring, the field is automatically aggregated as a SUM and an interactive filter appears in the view.

To change the aggregation of the filter:

- On the Filters shelf, right-click the field, select Measure, and then select an aggregation.

To change the type of quantitative filter in the view:

- Click the filter card drop-down and select from the following:

  **Range of Values:** Select the Range of Values option to specify the minimum and maximum values of the range to include in the view. The values you specify are included in the range.

    ![Sales](image1)

    **At Least:** Select the At Least option to include all values that are greater than or equal to a specified minimum value. This type of filter is useful when the data changes often so specifying an upper limit may not be possible.

    ![Sales](image2)

    **At Most:** Select the At Most option to include all values that are less than or equal to a specified maximum value. This type of filter is useful when the data changes often so specifying a lower limit may not be possible.

    ![Sales](image3)
Filter dates

In Tableau Desktop

When you drag a date field from the Data pane to the Filters shelf in Tableau Desktop, the following Filter Field dialog box appears:

You can select whether you want to filter on a relative date; filter between a range of dates; or select discrete dates or individual dates to filter from the view.

- **Filter relative dates**: Click Relative dates to define a range of dates that updates based on the date and time you open the view. For example, you may want to see Year to Date sales, all records from the past 30 days, or bugs closed last week. Relative date filters can also be relative to a specific anchor date rather than today.

- **Filter a range of dates**: Select Range of dates to define a fixed range of dates to filter. For example, you may want to see all orders placed between March 1, 2009 and June
Filter discrete dates: Select a discrete date value in the dialog box if you want to include entire date levels. For example, if you select Quarters, you can choose to filter specific quarters (e.g. Q1, Q2, Q3, Q4) from your view, regardless of the year.

Latest date preset: If you want to ensure that only the most recent date in a data source is selected in the filter when the workbook is shared or opened, select a discrete date such as Month/Day/Year or Individual Dates and then, on the General tab, select Filter to latest date value when workbook is opened.
Notes: When you filter to the latest date value, this setting applies only to data source filters in a workbook.

In the order of operations, the latest date filter is global to the workbook, while context filters apply per worksheet. The latest date is determined just after the workbook opens for first use, after data source filters, but before context filters. At that point the date is set, and the latest date preset is used as a dimension filter.

If you are using additional filters in views (including relative date filters and context filters), the latest date value setting may result in an empty view with no data when those additional filters do not select data from the latest date in the database.

On Tableau Server and Tableau Online, presets are applied when the view first loads in the browser, but not when the browser or data is refreshed.

- **Filter individual dates**: Select Individual dates to filter specific dates from your view.

- **Additional date filter options**: When you select Relative dates or Range of dates, the Filter dialog box opens. In that dialog box, you can define a Starting date or Ending date. You can also select Special to include null dates, non-null dates, or all dates.

In web authoring

When you drag a date field from the Data pane to the Filters shelf in web authoring, a date range filter appears in the view.

To change the type of filter, click the filter card drop-down and select from the following:

- **Relative Date**: Click Relative dates to define a range of dates that updates based on the date and time you open the view.
- Range of Dates: Select Range of dates to define a fixed range of dates to filter

- Start Date: Select Start date to define a fixed start date to filter on.

- End Date: Select End date to define a fixed end date to filter on.

- Browse Periods: Select browse periods to select a period of time to filter by, such as one day, one week, one month, one year, five years, etc.
Filter table calculations

To create a table calculation filter, create a calculated field, and then place that field on the Filters shelf.

Filters based on table calculations do not filter out underlying data in the data set, because table calculation filters are applied last in the order of operations. This means Tableau evaluates any table calculations in the view first, and then applies table calculation filters on the results in the current view.

Apply table calculation filter to totals

When you show totals in a view and you want a table calculation filter to apply to the totals, you can select **Apply to totals** in the drop-down menu for that filter (on the Filters shelf). This option lets you decide when a table calculation filter should be applied to totals.
This options is available when you show totals in the view, and you add a table calculation filter to the view. Select **Apply to totals** to apply the table calculation filter to all of the results in the table, including the totals.

Display interactive filters in the view

When an interactive filter is shown, you can quickly include or exclude data in the view.

**Note:** In web authoring, interactive filters are automatically added to the view when you drag a field to the Filters shelf.

**To show a filter in the view:**
1. In the view, click the field drop-down menu and select **Show Filter**.

The field is automatically added to the Filters shelf (if it is not already being filtered), and a filter card appears in the view. Interact with the card to filter your data.

**Note:** In Tableau Desktop, you can add an interactive filter to the view for a field that is not currently used in the view. To do so, In the Data pane, click the field drop-down menu, and then select **Show Filter**.
Set options for filter card interaction and appearance

After you show a filter, there are many different options that let you control how the filter works and appears. You can access these options by clicking the drop-down menu in the upper right corner of the filter card in the view.

Some options are available for all types of filters, and others depend on whether you’re filtering a categorical field (dimension) or a quantitative field (measure).

You can customize how filters appear in the view, in dashboards, or when published to Tableau Server or Tableau Online.

Here are some of the general filter card options:

- **Edit Filter** (Tableau Desktop only) - This option opens the main Filter dialog box so you can further refine the filter by adding conditions and limits.

- **Remove Filter** (Tableau Desktop only) - Removes the filter from the Filters shelf and removes the filter card from the view.

- **Apply to worksheets** - Allows you to specify whether the filter should apply to only the current worksheet or be shared across multiple worksheets. For more information, see [Apply Filters to Multiple Worksheets](#) on page 1199.

- **Format Filters** (Tableau Desktop only) - Customize the font and colors of all your filter cards in the view.

- **Only relevant values** - Specifies which values to show in the filter. When you select this option other filters are considered and only values that pass these filters are shown. For example, a filter on State will only show the Eastern states when a filter on Region is set. You can use the toggle at the top of the filter card to switch between this option and the All Values in Database option.

- **All values in hierarchy** - Specifies which values to show in the filter. When you create a filter from a hierarchical field, this option is selected by default. Filter values are displayed
based on relevance of the parent/child relationships in the hierarchy.

- **All values in database** - Specifies which values to show in the filter. When you select this option all values in the database are shown regardless of the other filters on the view.

- **All values in context** (Tableau Desktop only) - When one of the filters in the view is a context filter, select this option on a different filter to only display values that pass through the context filter. For more information, see [Improve View Performance with Context Filters](#) on page 1194.

- **Include values** - When this option is selected, the selections in the filter card are included in the view.

- **Exclude values** - When this option is selected, the selections in the filter card are excluded from the view.

- **Hide Card** (Tableau Desktop only) - Hides the filter card but does not remove the filter from the Filters shelf.

---

**Filter card modes**

You can control the appearance and interaction of your filter card in the view by selecting a filter card mode.

To select a filter card mode, in the view, click the drop-down menu on the filter card and then select a mode from the list.
The types of filter card modes you see in the list of options depend on whether your filter is on a dimension or a measure. Below you can find brief descriptions of the types of filter card modes available for dimensions and measures.

For dimensions, you can choose from the following filter modes:

- **Single Value (List)**: Displays the values of the filter as a list of radio buttons where only a single value can be selected at a time.

- **Single Value (Dropdown)**: Displays the values of the filter in a drop-down list where only a single value can be selected at a time.

- **Single Value (Slider)**: Displays the values of the filter along the range of a slider. Only a single value can be selected at a time. This option is useful for dimensions that have an
implicit order such as dates.

- **Multiple Values (List):** Displays the values in the filter as a list of check boxes where multiple values can be selected.

- **Multiple Values (Dropdown):** Displays the values of the filter in a drop-down list where multiple values can be selected.

- **Multiple Values (Custom List):** Displays a text box where you can type a few characters and search for the value. Alternatively, you can type or paste a list of values into the text box to create a custom list of values to include.

- **Wildcard Match:** Displays a text box where you can type a few characters. All values that match those characters are automatically selected. You can use the asterisk character as a wildcard character. For example, you can type “tab*” to select all values that begin with the letters “tab”. Pattern Match is not case sensitive. If you are using a multidimensional data source, this option is only available when filtering single level hierarchies and attributes.

For measures, you can choose from the following filter modes:

- **Range of Values/Dates:** shows the filtered values as a pair of sliders that you can adjust to include or exclude more values. Click on the upper and lower limit readouts to enter the values manually.

  The darker area inside the slider range is called the data bar. It indicates the range in which data points actually lie in the view. Use this indicator to determine a filter that makes sense for the data in your data source. For example, you may filter the Sales field to only include values between $200,000 and $500,000 but your view only contains values between $250,000 and $320,000. The range of data you can see in the view is indicated by the data bar while the sliders show you the range of the filter.

  **Note:** Data bars only show in filters where the filtered field is also used in the view (e.g., on Columns, Rows, or on the Marks card, and so on) and are at the same aggregation level as the field on the Filters shelf. For example, a filter on SUM (Sales) will only display data bars if the SUM(Sales) field is used in the view. It won't show if AVG(Sales) is used in the view. Even though in both scenarios, the filtered field, Sales is used in the view; in the latter case the aggregation is different.
than the aggregation of the filter.

- **At Least/Starting Date**: shows a single slider with a fixed minimum value. Use this option to create a filter using an open ended range.

- **At Most/Ending Date**: shows a slider with a fixed maximum value. Use this option to create a filter using an open ended range.

- **Relative to Now**: this option shows a control where you can define a dynamic date range that updates based on when you open the view. The option is only available for filters on continuous date fields.

- **Browse periods**: shows common date ranges such as past day, week, month, three months, one year, and five years. This option is only available for filters on continuous date fields.

**Note**: When you expose a filter for Measure Values or Measure Names as a single value list, selecting All will automatically convert the filter to a multiple values list. For information on Measure Values and Measure Names, see [Measure Values and Measure Names](#) on page 2196.

---

**Customize filter cards**

In addition to the general filter options and the filter modes, you control how your filter appears in the worksheet, on dashboards, or when published to the web even further in Tableau Desktop.

To customize filters, click the filter card drop-down menu and select **Customize**.
You can select from the following options:

- **Show “All” Value** - toggles whether to show the “All” option that displays by default in multiple values and single value lists.

- **Show Search Button** - toggles whether to show the search button at the top of the filter.

- **Show Include/Exclude** - toggles whether to show the Include Values and Exclude Values commands on the filter card menu. When shown, users can switch the filter between include and exclude modes.

- **Show Filter Types** - toggles whether to let users change the type of quick filter shown. For example, when shown, a user can change a multiple values list to a compact list.

- **Show More/Fewer button** - toggles whether to show the More/Fewer button at the top of the filter.
• **Show All Values button** - toggles whether to display the Show All Values button on the filter card.

Whenever data is excluded in the filter, the small red "x" appears on the Show All Values button. When all values are showing, the red "x" disappears.

• **Show Apply Button** - toggles whether to show the Apply button at the bottom of the filter. When shown, changes to the filter are only applied after you click the button. Pending changes are indicated with a green color. This option is only available in multiple values lists and dropdowns. This option is available in web authoring.

• **Show Readouts** - controls whether the minimum and maximum values are displayed as text above a range of values. The readouts can be used to manually type a new value instead of using the sliders.

• **Show Slider** - controls whether the slider displays. When this option is cleared, the filter only displays the readouts.

• **Show Null Controls** - shows a drop-down list that lets you control how the filter handles null values. You can select from values in a range; values in a range and null values; null values only, non-null values, or all values.

**Note:** Not all of the above options are available for views published to Tableau Server or Tableau Online.
Filter Data Across Multiple Data Sources

While working with multiple data sources in a workbook, you might want to compare the data between them using a field they have in common. To do so, you can apply a filter across multiple primary data sources.

For example, the following dashboard shows the order quantity, average sales, and average profit for customers. It has three views. Each of the views uses a different data source as its primary data source, and those data sources all have a field in common: Customer Name. There is also a filter card in the view for Customer Name.
This is an interesting dashboard with a lot of great information, but you might want to update all of the views in the dashboard at the same time by the customer you're analyzing. For example, maybe you want to see the average sales, profit, and number of orders you've received from one of your customers, Aaron Riggs.

To do so, you can filter all three data sources on the Customer Name field.

Follow the steps below to learn how to filter data across multiple data sources.

**Before you start, you might want to know the following:**

In 10.0, you can only filter data across multiple primary data sources. You cannot filter data across secondary data sources. Filtering data across a worksheet's secondary data source is not currently supported in Tableau Desktop.

If you want to filter data across secondary data sources, consider the following alternatives:

- Join tables that are in the same data source, instead of blending them. For more information, see [Join Your Data on page 657](#).
- Create a cross-database join if your tables are in different data sources. For more information, see [Join Your Data on page 657](#).
- Make the primary data source the secondary data source, and the secondary data source the primary. **Note:** This might not always be feasible given the level of detail you want in the final view.

**Note:** To ensure the data strings with mixed capitalization are treated as case-insensitive in the filter, create a calculated field using the `UPPER()` string function, and then create the filter relationship using that calculated field.

For more FAQs about cross data-source filters, see the [Cross data-source filtering FAQs](#) forum post in the Tableau Community.

**Step 1: Define relationships between your data sources**

Before you can create relationships between data sources, you must ensure that there is a common field between the data sources you're mapping. The fields do not need to be named the same in each data source, but they should have some data in common.
After you've identified the common fields, you must create relationships between them, or *map the fields* to one another.

If the fields happen to have the same name, Tableau might recognize them as related automatically. You can modify an existing relationship that was created automatically by Tableau, or create a new relationship between two fields in different data sources, by following the procedure below.

To define a relationship between your two data sources:

1. Select **Data > Edit Relationships**.
2. In the Edit Relationships dialog box, do the following:
   - Select a data source for **Primary data source**, and select a data source for **Secondary data source**.
   - Select **Custom**.
   - To define a relationship between fields that have different names, click **Add**.
     To edit an existing relationship, select the fields on the right, and then click **Edit**.
3. In the Add/Edit Field Mapping dialog box, do the following, and then click **OK**:
   - Under **Primary data source field**, select a field.
   - Under **Secondary data source field**, select a field with similar data to the field you selected for the primary data source.

   For more information about editing relationships, see **Step 4: (Optional) Define or edit relationships** on page 688.

   **Note:** After you define your relationships, you do not need to enable blends (that is, you do not have to click the link icon in the **Data** pane) to filter across your data sources.

   For more information about the difference between primary and secondary data sources, see **Primary and secondary data sources** on page 683.

**Step 2: Add a filter to the view**

After you have defined relationships between your data sources, go to one of your worksheets and drag a dimension to the **Filters** shelf. Then select to include or exclude data from the view.
For more information about filtering your data, see Filter Data from Your Views on page 1162.

Optional Step: Display a filter card in the view. Right-click the field on the Filters shelf and select Show filter. A filter card for that field appears in the view.

For more information about filter cards (previously known as quick filters), see Display interactive filters in the view on page 1176.

Step 3: Apply the filter to your worksheets

After you set up your filter, you can apply that filter to all the worksheets in your workbook that use related data sources, or you can apply it to selected worksheets.

- To apply the filter to all worksheets that use related data sources, right-click the field on the Filters shelf and select Apply to worksheets > All using related data sources.

The Customer Name field is added to the Filters shelf on every worksheet that uses a related data source. An icon is added next to the field on the Filters shelf, indicating that the filter is being applied to multiple data sources.

If you hover over the field on the Filters shelf in any worksheet, you can find details about the type of filter, the source field for the filter, and the other sheets the filter applies to.

- To apply the filter to selected worksheets, right-click the field on the Filters shelf and select Apply to worksheets > Selected worksheets.

The Customer Name field is added to the Filters shelf on the worksheets you selected. An icon is added next to the field on the Filters shelf to indicate that the filter is being
applied to select worksheets.

Now, on the dashboard, when you filter the view down to Aaron Riggs, all three views update and you can see that Aaron made orders in 2010, 2011, and 2013, and spent an average of 3,700 USD. The average profit for these orders was approximately 1,000 USD.

Source fields and target fields

When you apply a filter to multiple data sources, you create a source field and one or more target fields. Both source fields and target fields appear on the Filter shelf in their respective worksheets.

The source field is the field you're filtering with.

The target field on any given worksheet is a field from another data source that is related to the source field. This is the field that is actually being filtered on that worksheet.
The source field determines the data that is included or excluded from the target fields. For example, let's say you have three worksheets that use three separate data sources (A, B, and C) as their primary data source. Each of the data sources has a field in common (Fruit), and the data is as follows:

<table>
<thead>
<tr>
<th>Fruit from data source A</th>
<th>Fruit from data source B</th>
<th>Fruit from data source C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apples</td>
<td>Apples</td>
<td>Apples</td>
</tr>
<tr>
<td>Oranges</td>
<td>Grapefruit</td>
<td>Grapefruit</td>
</tr>
<tr>
<td>Bananas</td>
<td>Oranges</td>
<td>Oranges</td>
</tr>
<tr>
<td></td>
<td>Pears</td>
<td>Tomatoes</td>
</tr>
</tbody>
</table>

If the Fruit field from data source A is the source field for the cross data source filter, then the data that appears for the target fields is as follows:

<table>
<thead>
<tr>
<th>Fruit from data source A - Source field</th>
<th>Fruit from data source B - Target field</th>
<th>Fruit from data source C - Target field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apples</td>
<td>Apples</td>
<td>Apples</td>
</tr>
<tr>
<td>Oranges</td>
<td>Oranges</td>
<td>Oranges</td>
</tr>
<tr>
<td>Bananas</td>
<td></td>
<td>Bananas</td>
</tr>
</tbody>
</table>

Any data that does not match the data in the source field is excluded from the target fields, and will not appear in your worksheets or in your filter cards.

On the Filters shelf, source fields are indicated with either a 🍐 icon when the source field applies to all worksheets with a related data source, or a 🍏 icon when it applies to select
worksheets. Target fields are indicated with a icon on the field on the Filters shelf. They are also indicated with a or icon next to the field on the Filters shelf.

Create Relative Date Filters

You can create filters to show a date period that is relative to when you open the view. For example, create dynamic filters to only show the current week, the year to date, or the past 10 days. Relative date filters make it easier to create lasting views that you can publish and share.

Step 1: Drag a Date Field to the Filter Shelf

Right-click (control-click on a Mac) and drag a date field from the Data window to the Filters shelf, and then click Relative Date in the Filter Field dialog box.

Then click Next.
Step 2: Select a Time Unit

In the Filter dialog box, click **Relative dates** and then select the unit of time you want to base your filter on. For example, to filter to only show the three most recent weeks, select **Weeks**.

![Filter dialog box showing Relative dates and time units]

Step 3: Define the Date Period

Use the options in the lower part of the Filter dialog box to specify which weeks to include in the view. To show the last three weeks click **Last** and then select the number **3**.

![Options in the Filter dialog box]

Date periods include the current unit of time. For example, if you select the last three weeks Tableau will include the current week and the two previous weeks. The range of time that you have selected is displayed in the upper right of the Filter dialog box.
Tableau will include all available dates for the time unit selected. For example, if you select Month as the time unit, and the current date is January 7th, Tableau will display dates for January 1st through January 31st.

**Step 4: Watch the View Update**

The filter will now update to always show the three weeks preceding the current date.

Relative date filters, once created, can be displayed in the view as filter cards.
Improve View Performance with Context Filters

By default, all filters that you set in Tableau are computed independently. That is, each filter accesses all rows in your data source without regard to other filters. However, you can set one or more categorical filters as context filters for the view. You can think of a context filter as being an independent filter. Any other filters that you set are defined as dependent filters because they process only the data that passes through the context filter.

You may create a context filter to:

- Improve performance – If you set a lot of filters or have a large data source, the queries can be slow. You can set one or more context filters to improve performance.
- Create a dependent numerical or top N filter – You can set a context filter to include only the data of interest, and then set a numerical or a top N filter.

For example, suppose you're in charge of breakfast products for a large grocery chain. Your task is to find the top 10 breakfast products by profitability for all stores. If the data source is very large, you can set a context filter to include only breakfast products. Then you can create a top 10 filter by profit as a dependent filter, which would process only the data that passes through the context filter.

**Note:** As of Tableau 9.0, context filters no longer create temporary tables, except for generic ODBC data sources and customized data sources.

Create Context Filters

To create a context filter, select **Add to Context** from the context menu of an existing categorical filter. The context is computed once to generate the view. All other filters are then computed relative to the context. Context filters:

- Appear at the top of the Filters shelf.
- Are identified by a gray color on the Filters shelf.
- Cannot be rearranged on the shelf.

As shown below, the **Ship Mode** dimension is set to be the context for a view. The **Region** filter is computed using only the data that passes through **Ship Mode**.
You can modify a context filter by:

- Removing the field from the Filters shelf – If other context filters remain on the shelf, a new context is computed.

- Editing the filter – A new context is computed each time you edit a context filter.

- Selecting **Remove from Context** – The filter remains on the shelf as a standard filter. If other context filters remain on the shelf, a new context is computed.

**Speed up Context Filters**

To improve performance of context filters, especially on large data sources, follow these general rules.

- Using a single context filter that significantly reduces the size of the data set is much better than applying many context filters. In fact, if a filter does not reduce the size of the data set by one-tenth or more, it is actually worse to add it to the context because of the performance cost of computing the context.

- Complete all of your data modeling before creating a context. Changes in the data
model, such as converting dimensions to measures, require recomputing the context.

- Set the necessary filters for the context and create the context before adding fields to other shelves. Doing this work first makes the queries that are run when you drop fields on other shelves much faster.

- If you want to set a context filter on a date you can use a continuous date. However, using date bins like YEAR(date) or context filters on discrete dates are very effective.

If your data set is heavily indexed, context filters may not provide performance improvement and may actually cause slower query performance.

Context filters can adversely affect any query performance improvements when using the **Include joined tables only when referenced** option in the Tables dialog box. See the note at the bottom of **Join Your Data** on page 657.

**Example - Create Context Filters**

This example walks you through how to create a context filter. First you’ll filter a view to show the top 10 products by sales. Then you’ll create a context filter on product category so you can see the top 10 furniture products.

1. Use the **Sample - Superstore** data source to create the initial view shown below. The view shows the sales for all sub-categories, sorted with the highest sale at the top.
2. Now create a Top 10 filter to just show the top selling products. You can create this filter by dragging the **Sub-Category** field to the Filters shelf. In the Filter dialog box, switch to the **Top** tab and define a filter that is Top 10 by Sum of Sales. See [Filter Data from Your Views on page 1162](#) to learn more about defining a Top N filter.

3. When you click **OK**, you’ll see that the view is filtered to show the top 10 product sub-categories in terms of sales.
4. Now, let's add another filter to show only furniture products. Drag the **Category** field to the Filters shelf and select only **Furniture**. When finished, click **OK**.

The view is filtered but instead of 10 products, it now shows 3. This is because by default all filters are evaluated separately and the view shows the intersection of the results. So this view shows that three of the top 10 overall products are furniture products.

5. To find out what the top 10 furniture products are we need to make the Category filter a context filter. Right-click the field on the Filters shelf and select **Add to Context**.

6. The filter is marked as a context filter and the view updates to show the top four furniture products. Why not 10? Because only four of the sub-categories contain furniture. But we now know that the Top 10 filter is being evaluated on the results of that context.
Apply Filters to Multiple Worksheets

When you add a filter to a worksheet, by default it applies to the current worksheet. Sometimes, however, you might want to apply the filter to other worksheets in the workbook.

You can select specific worksheets to apply the filter to or apply it globally to all worksheets that use the same data source or related data sources. For example, you might have a filter that only includes a specific region or product of interest. Rather than adding this filter every time you create a new worksheet, you can simply create the filter once and then apply it to multiple worksheets.

Apply filters to all worksheets that use a related primary data source

This option applies the filter to all worksheets that use related data sources as their primary data source.

**Note:** To apply a filter to worksheets that use a related primary data source in web authoring, relationships between the data sources must be set up in Tableau Desktop, and then published to the web.

For more information, see [Filter Data Across Multiple Data Sources](#) on page 1185.

To apply a filter to all worksheets using a related primary data source:

- On the Filters shelf, right-click the field and select **Apply to Worksheets > All Using Related Data Sources**.

Filters that use this option are global across the workbook.

Filters that apply to all related data sources are marked with an icon. The filter is automatically created on any existing worksheets, and on any new worksheets you create that use a related data source.
Any changes you make to the filter affects all of those worksheets.

Apply filters to all worksheets that use the current primary data source

This option applies the filter to all worksheets that use the current worksheet's primary data source as their primary data source.

To apply a filter to all worksheets using the current primary data source:

- On the Filters shelf, right-click the field and select **Apply to Worksheets > All Using This Data Source**.

Filters that use this option are global across the workbook.

Filters that apply to all worksheets are marked with a data source icon ☐. The filter is automatically created on any new worksheets you create after you drag a field to the view.

Any changes you make to the filter affects all of those worksheets.

**Note:** If you are blending multiple data sources in a view, **All Using This Data Source** adds the filter to all sheets that use the same primary data source in the current sheet. The sheets that the filter gets applied to is not based on the filter field’s data source.

Apply filters to select worksheets (Tableau Desktop only)

This option opens a dialog box where you can select from a list of worksheets that use the same data source or related data sources.

To apply a filter to select worksheets:

- On the Filters shelf, right-click the field and select **Apply to Worksheets > Selected Worksheets**.
- In the Apply Filter to Worksheets dialog box, select the worksheets that you want to apply the filter. If any of the sheets already contain a filter on the same field, the dialog box will provide details about the filter.
If you select the sheet, the current filter will override any existing filter selections.

Filters that apply to a selection of worksheets are marked with the worksheet icon ![Worksheet Icon]. Any changes you make to the filter affect all of the selected worksheets.

**Apply filters to the current worksheet only**

This option only applies to the current worksheet. This option is selected by default when you create new filters. Filters that are local to the current worksheet are shown without any additional icons.

To apply a filter to the current worksheet only:

- On the Filters shelf, right-click the field and select **Apply to Worksheets > Only This Worksheet**.

If you apply a filter to all worksheets or selected worksheets and then change the setting to apply the filter to just the current worksheet, the filters are not removed from all other
worksheets. Rather, the filters are disconnected and they are all made local to their respective worksheets. You can go to each worksheet and remove the filter or modify selections.

Filter all worksheets on a dashboard (Tableau Desktop only)

This option applies the filter to all worksheets in the dashboard that use the same data source or related data sources as their primary data source.

To filter all worksheets on a dashboard:

- In a dashboard, click the drop-down menu on a filter card and select **Apply to Worksheets > Selected worksheets**.
- In the Apply Filter to Worksheets dialog box, click **All on dashboard**, and then click **OK**.

**Note:** In previous versions of Tableau Desktop, the **All Using This Data Source** option was called Make Global and the **Only This Worksheet** option was Make Local.

Sort Data in a Visualization

There are many ways to sort data in a visualization in Tableau. This article demonstrates all the ways to sort data in a visualization.

In this article

- **Sort data on an axis** below
- **Sort specific fields in the viz (Tableau Desktop only)** on the next page
- **Sort data using the toolbar** on page 1205
- **Sort data using headers or legends** on page 1206
- **Create a nested sort** on page 1208

Sort data on an axis

A quick way to sort data is to use the sort buttons on an axis. To do so:

1. In a worksheet, hover over a viz axis.
2. Click the sort icon that appears.
One click sorts the data in ascending order. Two clicks sorts it in descending order. Three clicks clears the sort.

The sort will update correctly if the underlying data changes.

Sort specific fields in the viz (Tableau Desktop only)

To sort a specific field in the visualization:

1. In a worksheet, right-click (control-click on Mac) the field you want to sort and select **Sort**.
   
   **Note**: You can only sort dimensions.

2. In the **Sort** dialog box that opens, do the following:
   
   - For **Sort order**, select one of the following options:
     
     - **Ascending** - sorts the data in ascending order from lowest to highest or least to most.
     
     - **Descending** - sorts the data in descending order from highest to lowest or most to least.
For **Sort by**, select one of the following options:

- **Data source order** - the order that the data source naturally orders the data. Generally for relational data sources, this tends to be in alphabetical order—more specifically, in natural sort order, which is identical to alphabetical order, except that multi-digit numbers are ordered as a single character. So, for example, in alphabetical sorting, "z11" comes before "z2" because "1" is evaluated as being smaller than "2", whereas in natural sorting "z2" comes before "z11" because "2" is evaluated as being smaller than "11".

If you are using a cube, this order is the defined hierarchal order of the members within a dimension.

- **Alphabetic** - sorts the data alphabetically.

  **Note:** This sort is case sensitive. It will sort [A-Z] before [a-z].

  To create a non-case sensitive sort, create a calculated field using the `UPPER()` or `LOWER()` functions to transform your strings into a consistent case. For more information, see *String Functions* on page 1288.

- **Field** - sorts the data based on values in another field. For example, you can select to sort by the Sales field, using the average aggregation.

- **Manual** - allows you to manually sort the data in the order you prefer. For example, if you want to sort high school years in order: Freshman, Sophomore, Junior, Senior.

3. When finished, click **OK**.

   The visualization updates with the new sorting.
You should keep the following rules in mind when interpreting the sort results:

- Tableau computes the sort across the entire table using the specified criteria. See **Sort Data in a Visualization** on page 1202 for more information.

- Sorts do not break the dimension hierarchy. Sorted fields are always displayed within the ordered context already set forth by the fields on the Rows and Columns shelves. This means that Tableau will not rearrange any of the headers of the fields that appear before (to the left of) the sorted field.

If you want to break the dimension hierarchy when sorting a multidimensional data source, place only the hierarchy level that you want sorted on the Rows or Columns shelf.

---

**Sort data using the toolbar**

You can quickly sort data in ascending or descending order using the toolbar. To do so:
- In a worksheet, on the toolbar located just above the Columns shelf, click the Descending sort icon to sort data from highest/most to lowest/least, or the Ascending icon to sort data from lowest/least to highest/most.

<table>
<thead>
<tr>
<th>Sub-Category</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories</td>
<td>$0</td>
</tr>
<tr>
<td>Appliances</td>
<td>$50,000</td>
</tr>
<tr>
<td>Art</td>
<td>$100,000</td>
</tr>
<tr>
<td>Binders</td>
<td>$150,000</td>
</tr>
<tr>
<td>Bookcases</td>
<td>$200,000</td>
</tr>
<tr>
<td>Chairs</td>
<td>$250,000</td>
</tr>
<tr>
<td>Copiers</td>
<td>$300,000</td>
</tr>
<tr>
<td>Envelopes</td>
<td>$0</td>
</tr>
<tr>
<td>Fasteners</td>
<td>$50,000</td>
</tr>
<tr>
<td>Furnishings</td>
<td>$100,000</td>
</tr>
<tr>
<td>Labels</td>
<td>$150,000</td>
</tr>
<tr>
<td>Machines</td>
<td>$200,000</td>
</tr>
<tr>
<td>Paper</td>
<td>$250,000</td>
</tr>
<tr>
<td>Phones</td>
<td>$300,000</td>
</tr>
<tr>
<td>Storage</td>
<td>$0</td>
</tr>
<tr>
<td>Supplies</td>
<td>$50,000</td>
</tr>
<tr>
<td>Tables</td>
<td>$100,000</td>
</tr>
</tbody>
</table>

Sort data using headers or legends

You can manually sort data in your visualization by dragging and dropping members in a header or legend, such as a color or shape legend.

To manually sort data using a header:

- In a worksheet, select a member in a header and drag it up or down.
  
  The viz updates with the new sort.
To manually sort data using a legend:

- In a worksheet, select a member in a legend and drag it up or down.

The viz updates with the new sort. Notice that the order the data points in the view appear also updated.
Create a nested sort

With the release of Tableau Desktop 2018.2, you can now create a nested sort without any additional calculated fields or combined dimensions.

Suppose you are looking at sales sub-categories for all regions in your country. In the view below, you can see that the Central region sells more products in the Chairs subcategory than any other subcategory. However it’s harder to compare how Chairs ranked in other regions.

To see which subcategories ranked the highest in each region, you might want to sort the subcategories within each region instead. You can do this with nested sorting.

Set up the view

1. In Tableau Desktop, open a new workbook and connect to the Sample Superstore data source that comes with Tableau.

2. Navigate to a new worksheet.

3. From Measures, drag Sales to the Columns shelf.

4. From Dimensions, drag Region to the Rows shelf.

5. From Dimensions, drag Sub-Category to the Rows shelf, to the right of the Region field.

Create a nested sort

In a worksheet, hover over a viz axis.
Click the sort icon that appears at the top of the view.

**One** click sorts the data in ascending order. **Two** clicks sorts it in descending order. **Three** clicks clears the sort.
Now you have a nested sort where the subcategories are listed in descending order of sum of sales within each region. You can see that Chairs are the highest selling subcategory in the Central region, but in the East region, Phones are the best selling product; Chairs are the second-best selling product.
Note: You might also want to change the formatting so that the row dividers are at the row level (sub-category) instead of on the pane level (region). To do so:

1. Select Format > Borders.
2. In the Format Borders pane, under Row Dividers, move the Level slider to the right.

Analyze Data

This section describes the various ways you can perform more advanced analysis in Tableau. Read the following articles for information on how to create calculated fields, find clusters in data, calculate percentages, and use various tools to explore and inspect data.

Watch a video: For a 3-minute introduction to calculations, watch the Getting Started with Calculations free training video. Use your tableau.com account to sign in.

For a 7-minute walk through on how to create reference lines in Tableau, watch the Reference Lines free training video. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Calculate Percentages in Tableau

Any analysis in Tableau can be expressed in terms of percentages. For example, rather than viewing sales for every product, you might want to view each product’s sales as a percentage of the total sales for all products.

In this article

About percentages below
How to calculate percentages on page 1214
Percentage options on page 1214

About percentages

There are two factors that contribute to the percentage calculation:
1. The data to which you compare all percentage calculations

Percentages are a ratio of numbers. The numerator is the value of a given mark. The denominator depends on the type of percentage you want, and is the number to which you compare all your calculations. The comparison can be based on the entire table, a row, a pane, and so on. By default, Tableau uses the entire table. Other percentage calculations are available via the Percentage of menu item. See Percentage options on page 1214.

The figure below is an example of a text table with percentages. The percentages are calculated with the Sales measure aggregated as a summation, and are based on the entire table.

![Image of a text table with percentages]

2. The aggregation

Percentages are computed on the basis of the aggregation for each measure. Standard aggregations include summation, average, and several others. See Data Aggregation in Tableau on page 279 for more information.

For example, if the aggregation applied to the Sales measure is a summation, then the default percentage calculation (percent of table) means that each number displayed is the SUM(Sales) for that mark divided by the SUM(Sales) for the entire table.

In addition to using predefined aggregations, you can use custom aggregations when calculating percentages. You define your own aggregations by creating a calculated field. Once
the new field is created, you can use percentages on the field as you would any other field. See Aggregate Functions in Tableau on page 1320 for more information.

Percent calculations can also be applied to disaggregated data. In this case, all values are expressed as the percentage of a summation. You cannot choose any other aggregation.

Example

The view below shows a nested bar chart created using two dimensions and a measure that is aggregated as a maximum. Additionally, the data are color-encoded by a dimension and the default percentage calculation has been applied. Notice that the axis labels are modified to reflect the percent calculation.

The tooltip reveals that the maximum sales for furniture in the east in 2011 is 17.70% of the maximum for the entire table. What is the maximum for the table? If you recreate the view you'll see that the maximum occurs in the South, in the Technology category, in the year 2011. The tooltip for this bar segment would reveal a maximum sales of 100%.

The next view displays two disaggregated measures as a scatter plot. Again, the default percentage calculation has been applied as reflected by the modified axis labels.

The tooltip shows that the selected data point constitutes -0.475 percent of total profit and a 0.3552 percent of total sales. Percentage calculations are based on the entire data source.
How to calculate percentages

To calculate percentages in your visualization:

- Select Analysis > Percentages Of, and then select a percentage option.

Percentage options

Computing a percentage involves specifying a total on which the percentage is based. The default percentage calculation is based on the entire table. You can also choose the following options:

- Percent of Table on the next page
- Percent of Column on page 1216
- Percent of Row on page 1216
- Percent of Pane on page 1217
- Percent of Row in Pane on page 1218
- Percent of Column in Pane on page 1219
- Percent of Cell on page 1219

The option you choose is applied uniformly to all measures that appear on a worksheet. You cannot choose Percent of Column for one measure and Percent of Row for another.
The percentage options on the Analysis menu correspond to the percentage table calculations. When you select a percentage option, you are actually adding a Percent of Total table calculation. See Transform Values with Table Calculations on page 1524 for more information.

If you are unsure what the current percentage calculation means, display the grand totals. This provides more information about each row and column. For example, if you select Percent of Row while displaying grand totals, you will see that the total for each row is exactly 100%. See Show Totals in a Visualization on page 1754 for more information on grand totals.

The percent calculation options are described in the following sections. In each case, the grand totals are displayed as well.

Percent of Table

When you select Percentage Of > Table from the Analysis menu, each measure on the worksheet is expressed as a percentage of the total for the entire worksheet (table). For example, Technology in the East region accounts for 3.79% of total sales in 2014. The grand totals for rows show that 2014 accounts for 31.95% of the total sales. Summing the grand totals for rows or for columns yields 100% of the total.
Percent of Column

When you select **Percentage of > Column** from the **Analysis** menu, each measure on the worksheet is expressed as a percentage of the total for the column. The values within the red box add up to 100%.

![Worksheet Example]

Percent of Row

When you select **Percentage of Row**, each measure on the worksheet is expressed as a percentage of the total for the row. The values within the red box add up to 100%.
Percent of Pane

When you select **Percentage of Pane** from the **Analysis** menu, each measure on the worksheet is expressed as a percentage of the total for the panes in the view. This option is equivalent to **Percent of Table** when the table consists of only a single pane.

In the following view, the red box constitutes a single pane; the values within the red box add up to 100%.
Percent of Row in Pane

When you select **Percentage of > Row in Pane** from the **Analysis** menu, each measure on the worksheet is expressed as a percentage of the total for a row within a pane. This option is equivalent to as **Percent of Row** when the table is only a single pane wide.

In the following view, the red box constitutes a row within a pane; the values within the red box add up to 100%.

**Note:** If you place **Measure Names** as the inner dimension on the **Columns** shelf (that is, the dimension farthest to the right), Tableau will return 100% for each mark because
you cannot total up the values for multiple measure names. For example, you can’t total up the values for SUM(Sales) and SUM(Profit).

Percent of Column in Pane

When you select **Percentage of > Column in Pane** from the **Analysis** menu, each measure in the worksheet is expressed as a percentage of the total for a column within a pane. This option is equivalent to as **Percent of Column** when the table is only a single pane high.

In the following view, the red box constitutes a column within a pane; the values within the red box add up to 100%.

If you place **Measure Names** as the inner dimension on the **Rows** shelf (that is, the dimension farthest to the right on the shelf), Tableau will return 100% for each mark because you cannot total up the values for multiple measure names. For example, you can’t total up the values for SUM(Sales) and SUM(Profit).

Percent of Cell

When you select **Percentage Of > Cell** from the Analysis menu, each measure on the worksheet is expressed as a percentage of the total for each individual cell in the view. Most views show only one value per cell, in which case all cells show a percentage of 100%. But in
some cases, as, for example, when you disaggregate data, a single cell can contain multiple values:

See Also

Transform Values with Table Calculations on page 1524

Create Custom Fields with Calculations

If your underlying data doesn't include all of the fields you need to answer your questions, you can create new fields in Tableau using calculations and then save them as part of your data source. These fields are called calculated fields.

This section provides all the documentation you need to get started with calculations in Tableau.

Start here

Get Started with Calculations in Tableau on the next page
Create a Simple Calculated Field on page 1227

Transform Values with Table Calculations on page 1524
Create Level of Detail Expressions in Tableau on page 1567
Get Started with Calculations in Tableau

This article describes how to create and use calculated fields in Tableau using an example. You'll learn Tableau calculation concepts, as well as how to create and edit a calculated field. You will also learn how to work with the calculation editor, and use a calculated field in the view.

If you're new to Tableau calculations or to creating calculated fields in Tableau, this is a good place to start.

In this article

Why Use Calculated Fields below
Types of calculations  below
Create a calculated field on the next page
Use a calculated field in the view on page 1224
Edit a Calculated Field on page 1226

Why Use Calculated Fields

Calculated fields allow you to create new data from data that already exists in your data source. When you create a calculated field, you are essentially creating a new field (or column) in your data source, the values or members of which are determined by a calculation that you control. This new calculated field is saved to your data source in Tableau, and can be used to create more robust visualizations. But don't worry: your original data remains untouched.

You can use calculated fields for many, many reasons. Some examples might include:

- To segment data
- To convert the data type of a field, such as converting a string to a date.
- To aggregate data
- To filter results
- To calculate ratios

Types of calculations

You create calculated fields using calculations. There are three main types of calculations you can use to create calculated fields in Tableau:
• **Basic calculations** - Basic calculations allow you to transform values or members at the data source level of detail (a row-level calculation) or at the visualization level of detail (an aggregate calculation).

• **Level of Detail (LOD) expressions** - Just like basic calculations, LOD calculations allow you to compute values at the data source level and the visualization level. However, LOD calculations give you even more control on the level of granularity you want to compute. They can be performed at a more granular level (INCLUDE), a less granular level (EXCLUDE), or an entirely independent level (FIXED) with respect to the granularity of the visualization.

For more information, see Create Level of Detail Expressions in Tableau on page 1567.

• **Table calculations** - Table calculations allow you to transform values at the level of detail of the visualization only. For more information, see Transform Values with Table Calculations on page 1524.

The type of calculation you choose depends on the needs of your analysis and the question you want to answer.

Create a calculated field

Once you have determined the type of calculation you want to use, it's time to create a calculated field. This example uses a basic calculation.

**Note:** The example in this article uses the Sample-Superstore data source that comes with Tableau Desktop. To follow along with the steps in this article, connect to the Sample-Superstore saved data source and navigate to Sheet 1.

1. In Tableau, select Analysis > Create Calculated Field.
2. In the Calculation Editor that opens, do the following:
   - Enter a name for the calculated field. In this example, the field is called, Discount Ratio.
   - Enter a formula. This example uses the following formula:

   ```
   IIF([Sales] !=0, [Discount]/[Sales],0)
   ```
This formula checks if sales is not equal to zero. If true, it returns the discount ratio (Discount/Sales); if false, it returns zero.

**Tip:** To see a list of available functions, click the triangle icon on the right-side of the Calculation Editor.

![Calculation Editor](image)

Each function includes syntax, a description, and an example for your reference.

Double-click a function in the list to add it to the formula.

For more tips, see **Tips for Working with Calculated Fields in Tableau** on page 1614.

3. When finished, click **OK**.

The new calculated field is added to Measures in the Data pane because it returns a number. An equal sign (=) appears next to the data type icon. All calculated fields have equal signs (=) next to them in the **Data** pane.

```
# Discount
# Discount Ratio
# Profit
# Profit Ratio
# Quantity
# Sales
# Latitude (generated)
# Longitude (generated)
```
Use a calculated field in the view

**Step 1: Build the view**

1. From Dimensions, drag **Region** to the **Columns** shelf.
2. From Dimensions, drag **Category** to the **Rows** shelf.
3. On the **Rows** shelf, click the plus icon (+) on the **Category** field to drill-down to Subcategory.

The view updates to look like this:

![View Update](image-url)
Step 2: Add the calculated field to the view

1. From Measures, drag **Discount Ratio** to **Color** on the Marks card.
   
The view updates to highlight table.

   ![Table view with Binders heavily discounted in the Central region.]

   You can see that Binders are heavily discounted in the Central region. Notice that Discount Ratio is automatically aggregated as a sum.

2. On the Rows shelf, right-click **SUM(Discount Ratio)** and select **Measure (Sum) > Average**.
The view updates with the average of discount ratio shown.

Edit a Calculated Field

If at any time you need to change a calculation, you can edit the calculated field and it will update across your entire workbook.

To edit a calculated field:

1. In the Data pane, right-click the calculated field and select Edit.
2. In the Calculation Editor that opens, you can do the following:
   - Edit the name of the calculated field.
   - Update the formula.

   For this example, the formula is changed to return a discount ratio for orders over 2000 USD in sales:

   \[ \text{IIF}([\text{Sales}] > 2000, \frac{[\text{Discount}]}{[\text{Sales}]}, 0) \]

3. Click OK.

   The view updates to reflect the changes automatically. You do not need to re-add the
updated calculated field to the view.

See Also:

Understanding Calculations in Tableau on page 1229
Formatting Calculations in Tableau on page 1263
Functions in Tableau on page 1275
Create Level of Detail Expressions in Tableau on page 1567
Transform Values with Table Calculations on page 1524

Create a Simple Calculated Field

Sometimes your data source does not contain a field (or column) that you need for your analysis. For example, your data source might contain fields with values for Sales and Profit, but not for Profit Ratio. If this is the case, you can create a calculated field for Profit Ratio using data from the Sales and Profit fields.

This topic demonstrates how to create a simple calculated field using an example.
Step 1: Create the calculated field

1. In a worksheet in Tableau, select Analysis > Create Calculated Field.
2. In the Calculation Editor that opens, give the calculated field a name.
   In this example, the calculated field is called Profit Ratio.

Step 2: Enter a formula

1. In the Calculation Editor, enter a formula.
   This example uses the following formula:

   \[
   \frac{\text{SUM}([\text{Profit}])}{\text{SUM}([\text{Sales}])}
   \]

   Formulas use a combination of functions, fields, and operators. To learn more about creating formulas in Tableau, see Formatting Calculations in Tableau on page 1263 and Functions in Tableau on page 1275.

2. When finished, click OK.
   The new calculated field is added to the Data pane. If the new field computes quantitative data, it is added to Measures. If it computes qualitative data, it is added to Dimensions.
   You are now ready to use the calculated field in the view.
Check your work! Watch how to create a simple calculated field in action:

See Also

Get Started with Calculations in Tableau on page 1221

Formatting Calculations in Tableau on page 1263

Functions in Tableau on page 1275

Create Level of Detail Expressions in Tableau on page 1567

Transform Values with Table Calculations on page 1524

Understanding Calculations in Tableau

This series introduces the basics of understanding calculations in Tableau. In this topic, you'll learn why and when to use calculations.

In this series

Understanding Calculations in Tableau above (You are here)

Types of calculations
Choosing the right calculation
Tips for learning how to create calculations

Why use calculations

Calculations allow you to create new data from data that already exists in your data source, as well as perform computations on your data. This allows you to perform complex analyzes and add fields to your data source on your own and on the fly.

When to use calculations

The first hurdle to learning calculations in Tableau is to recognize when you actually need to use one. You can use calculations for many, many reasons. Some examples might include:

- To segment data
- To convert the data type of a field, such as converting a string to a date.
- To aggregate data
- To filter results
- To calculate ratios

Some common scenarios might include:

- **The data you need for your analysis is missing from your data source.**
  
  For example, if you have a Sales and Profit field in your data source, but you want to calculate cost, you can *create* a Cost field using a formula similar to the following.

  \[
  \text{[Sales]} - \text{[Profit]}
  \]
• You want to transform values in your visualization.

For example, you might want to calculate the difference in profit from one year to the other. You can use a quick table calculation to show the difference in profit in the visualization.

• You want to quickly categorize data.
For example, you might want to quickly color the data in your visualization as profitable or nonprofitable. You can create a calculated field using a calculation similar to the following and then add it to Color on the Marks card.

```
IF SUM([Profit]) > 0
THEN "Profitable"
ELSE "Nonprofitable"
END
```

---

**Continue to:** Types of Calculations in Tableau below

### Types of Calculations in Tableau

This article explains the types of calculations you can use in Tableau. You’ll learn the difference between each calculation and how they are computed.

There are three main types of calculations you can use to create calculated fields in Tableau:

- Basic expressions
- Level of Detail (LOD) expressions
- Table calculations on page 1241
Basic expressions

Basic expressions allow you to transform values or members at the data source level of detail (a row-level calculation) or at the visualization level of detail (an aggregate calculation).

For example, consider the following sample table, which contains data on two popular fantasy authors and their books. Perhaps you want to create a column with only the author's last name and a column that displays how many books are in each series.

<table>
<thead>
<tr>
<th>Book ID</th>
<th>Book Name</th>
<th>Series</th>
<th>Year Released</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Lion, the Witch and the Wardrobe</td>
<td>The Chronicles of Narnia</td>
<td>1950</td>
<td>C. S. Lewis</td>
</tr>
<tr>
<td>2</td>
<td>Prince Caspian: The Return to Narnia</td>
<td>The Chronicles of Narnia</td>
<td>1951</td>
<td>C. S. Lewis</td>
</tr>
<tr>
<td>3</td>
<td>The Voyage of the Dawn Treader</td>
<td>The Chronicles of Narnia</td>
<td>1952</td>
<td>C. S. Lewis</td>
</tr>
<tr>
<td>4</td>
<td>The Silver Chair</td>
<td>The Chronicles of Narnia</td>
<td>1953</td>
<td>C. S. Lewis</td>
</tr>
<tr>
<td>5</td>
<td>The Horse and His Boy</td>
<td>The Chronicles of Narnia</td>
<td>1954</td>
<td>C. S. Lewis</td>
</tr>
<tr>
<td>6</td>
<td>The Magician's Nephew</td>
<td>The Chronicles of Narnia</td>
<td>1955</td>
<td>C. S. Lewis</td>
</tr>
<tr>
<td>7</td>
<td>The Last Battle</td>
<td>The Chronicles of Narnia</td>
<td>1956</td>
<td>C. S. Lewis</td>
</tr>
<tr>
<td>8</td>
<td>Harry Potter and the Philosopher's Stone</td>
<td>Harry Potter</td>
<td>1997</td>
<td>J. K. Rowling</td>
</tr>
<tr>
<td>9</td>
<td>Harry Potter and Chamber of Secrets</td>
<td>Harry Potter</td>
<td>1998</td>
<td>J. K. Rowling</td>
</tr>
<tr>
<td>10</td>
<td>Harry Potter and the Prisoner of Azkaban</td>
<td>Harry Potter</td>
<td>1999</td>
<td>J. K. Rowling</td>
</tr>
<tr>
<td>11</td>
<td>Harry Potter and the Goblet of</td>
<td>Harry Potter</td>
<td>2000</td>
<td>J. K. Rowling</td>
</tr>
</tbody>
</table>
Row-level calculations

To create a column that displays the author's last name for every row in the data source, you can use the following row-level calculation:

```
SPLIT([Author], '.', 3)
```

The result can be seen below. The new column, titled **Author Last Name** is shown on the far right. The colors demonstrate the level of detail the calculation is performed at. In this case, the calculation is performed at the row-level of the data source, so each row is colored separately.
| 6  | The Magician's Nephew | The Chronicles of Narnia | 1955 | C. S. Lewis | Lewis |
| 7  | The Last Battle       | The Chronicles of Narnia | 1956 | C. S. Lewis | Lewis |
| 8  | Harry Potter and the Philosopher's Stone | Harry Potter | 1997 | J. K. Rowling | Rowling |
| 9  | Harry Potter and Chamber of Secrets | Harry Potter | 1998 | J. K. Rowling | Rowling |
| 10 | Harry Potter and the Prisoner of Azkaban | Harry Potter | 1999 | J. K. Rowling | Rowling |
| 11 | Harry Potter and the Goblet of Fire | Harry Potter | 2000 | J. K. Rowling | Rowling |
| 12 | Harry Potter and the Order of the Phoenix | Harry Potter | 2003 | J. K. Rowling | Rowling |
| 14 | Harry Potter and the Deathly Hallows | Harry Potter | 2007 | J. K. Rowling | Rowling |

**Aggregate calculations**

To create a column that displays how many books are in each series, you can use the following aggregate calculation:

```
COUNT([Series])
```

The result can be seen below. The new column, titled **Number of Books in Series - at Series level of detail** shows how that calculation would be performed at the Series level of detail in the view. The colors help demonstrate the level of detail in which the calculation is being performed.
<table>
<thead>
<tr>
<th>Series</th>
<th>Number of Books in Series - at Series level of detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Chronicles of Narnia</td>
<td>7</td>
</tr>
<tr>
<td>The Chronicles of Narnia</td>
<td></td>
</tr>
<tr>
<td>The Chronicles of Narnia</td>
<td></td>
</tr>
<tr>
<td>The Chronicles of Narnia</td>
<td></td>
</tr>
<tr>
<td>The Chronicles of Narnia</td>
<td></td>
</tr>
<tr>
<td>The Chronicles of Narnia</td>
<td></td>
</tr>
<tr>
<td>Harry Potter</td>
<td>7</td>
</tr>
<tr>
<td>Harry Potter</td>
<td></td>
</tr>
<tr>
<td>Harry Potter</td>
<td></td>
</tr>
<tr>
<td>Harry Potter</td>
<td></td>
</tr>
<tr>
<td>Harry Potter</td>
<td></td>
</tr>
<tr>
<td>Harry Potter</td>
<td></td>
</tr>
<tr>
<td>Harry Potter</td>
<td></td>
</tr>
</tbody>
</table>

In Tableau, the data looks like this:
But if you drag in **Book Id**, (which is a more granular field), the calculation updates based on that new granularity since aggregate calculations are performed at the visualization level of detail.
Level of Detail (LOD) expressions

Just like basic expressions, LOD expressions allow you to compute values at the data source level and the visualization level. However, LOD expressions give you even more control on the level of granularity you want to compute. They can be performed at a more granular level (INCLUDE), a less granular level (EXCLUDE), or an entirely independent level (FIXED).

For more information, see Create Level of Detail Expressions in Tableau on page 1567.

For example, consider the same sample table as above. If you wanted to compute when a book series was launched, you might use the following LOD expression:

\{ FIXED [Series]: (MIN([Year Released])) \}

The result can be seen below. The new column, titled Series Launched, displays the minimum year for each series. The colors help demonstrate the level of detail in which the calculation is being applied.

<table>
<thead>
<tr>
<th>Book ID</th>
<th>Book Name</th>
<th>Series</th>
<th>Year Released</th>
<th>Author</th>
<th>Series Launched</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Lion, the Witch and the Wardrobe</td>
<td>The Chronicles of Narnia</td>
<td>1950</td>
<td>C. S. Lewis</td>
<td>1950</td>
</tr>
<tr>
<td>2</td>
<td>Prince Caspian: The Return to Narnia</td>
<td>The Chronicles of Narnia</td>
<td>1951</td>
<td>C. S. Lewis</td>
<td>1950</td>
</tr>
<tr>
<td>3</td>
<td>The Voyage of the Dawn Treader</td>
<td>The Chronicles of Narnia</td>
<td>1952</td>
<td>C. S. Lewis</td>
<td>1950</td>
</tr>
<tr>
<td>4</td>
<td>The Silver Chair</td>
<td>The Chronicles of Narnia</td>
<td>1953</td>
<td>C. S. Lewis</td>
<td>1950</td>
</tr>
<tr>
<td>5</td>
<td>The Horse and His Boy</td>
<td>The Chronicles of Narnia</td>
<td>1954</td>
<td>C. S. Lewis</td>
<td>1950</td>
</tr>
<tr>
<td>6</td>
<td>The Magician's Nephew</td>
<td>The Chronicles of Narnia</td>
<td>1955</td>
<td>C. S. Lewis</td>
<td>1950</td>
</tr>
<tr>
<td>7</td>
<td>The Last Battle</td>
<td>The Chronicles of Narnia</td>
<td>1956</td>
<td>C. S. Lewis</td>
<td>1950</td>
</tr>
<tr>
<td>#</td>
<td>Title</td>
<td>Author</td>
<td>Year</td>
<td>Author</td>
<td>Year</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------------------</td>
<td>------------</td>
<td>------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>8</td>
<td>Harry Potter and the Philosopher's Stone</td>
<td>Harry Potter</td>
<td>1997</td>
<td>J. K. Rowling</td>
<td>1997</td>
</tr>
</tbody>
</table>

In Tableau, the calculation remains at the Series level of detail since it uses the FIXED function.
If you add another field to the view (which adds more granularity) the values for the calculation are not affected, unlike an aggregate calculation.
Table calculations

Table calculations allow you to transform values at the level of detail of the visualization only.

For more information, see Transform Values with Table Calculations on page 1524.

For example, consider the same sample table as above. If you wanted to compute the number of years since the author released their last book, you might use the following table calculation:

\[
\text{ATTR}([\text{Year Released}]) - \text{LOOKUP} (\text{ATTR}([\text{Year Released}]), -1)
\]

The result is shown below. The new column, titled Years Since Previous Book, displays the number of years between the book released in that row and the book released in the previous row (on the far right-side of the column) and demonstrates how the table calculation is being computed (on the left-side of the column).

The colors help demonstrate how the table calculation is being computed. In this case, the table calculation is being computed down each pane.
Note: Depending on the table calculation and how it is being computed across the table, the results may vary. For more information, see Transform Values with Table Calculations on page 1524.

<table>
<thead>
<tr>
<th>Book ID</th>
<th>Book Name</th>
<th>Series</th>
<th>Year Released</th>
<th>Author</th>
<th>Years Since Previous Book</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Lion, the Witch and the Wardrobe</td>
<td>The Chronicles of Narnia</td>
<td>1950</td>
<td>C. S. Lewis</td>
<td>NULL</td>
</tr>
<tr>
<td>2</td>
<td>Prince Caspian: The Return to Narnia</td>
<td>The Chronicles of Narnia</td>
<td>1951</td>
<td>C. S. Lewis</td>
<td>1951-1950, 1950</td>
</tr>
<tr>
<td>3</td>
<td>The Voyage of the Dawn Treader</td>
<td>The Chronicles of Narnia</td>
<td>1952</td>
<td>C. S. Lewis</td>
<td>1952-1951, 1951</td>
</tr>
<tr>
<td>4</td>
<td>The Silver Chair</td>
<td>The Chronicles of Narnia</td>
<td>1953</td>
<td>C. S. Lewis</td>
<td>1953-1952, 1952</td>
</tr>
<tr>
<td>5</td>
<td>The Horse and His Boy</td>
<td>The Chronicles of Narnia</td>
<td>1954</td>
<td>C. S. Lewis</td>
<td>1954-1953, 1953</td>
</tr>
<tr>
<td>7</td>
<td>The Last Battle</td>
<td>The Chronicles of Narnia</td>
<td>1956</td>
<td>C. S. Lewis</td>
<td>1956-1955, 1955</td>
</tr>
<tr>
<td>8</td>
<td>Harry Potter and the Philosopher's Stone</td>
<td>Harry Potter</td>
<td>1997</td>
<td>J. K. Rowling</td>
<td>NULL</td>
</tr>
<tr>
<td>Order of the Phoenix</td>
<td>Rowling</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Harry Potter and the Half-Blood Prince</td>
<td>J. K. Rowling</td>
<td>2005</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Harry Potter and the Deathly Hallows</td>
<td>J. K. Rowling</td>
<td>2007</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In Tableau, the data looks like this:

![Tableau visualization](image)

However, if you change the visualization in a way that affects the layout, such as removing a dimension from the view, the calculation values change.

For example, in the image below, **Author** is removed from the viz. Since the table calculation is computed by pane, removing Author changes the granularity and layout of the viz (instead of two panes there is now only one). The table calculation therefore calculates the time between 1956 and 1997.
Continue to **Choosing the Right Calculation Type** below

Also in this series:

*Understanding Calculations in Tableau*

*Tips for Learning How to Create Calculations*

**Choosing the Right Calculation Type**

The type of calculation you choose depends on the needs of your analysis, the question you want to answer, and the layout of your visualization.
Which calculation is right for your analysis?

Choosing the type of calculation to use for your analysis is not always easy. When trying to decide, consider the following questions and examples.

Note: This content was originally published on the Tableau Blog. See A Handy Guide to Choosing the Right Calculation for Your Question to read it.

Basic expression or table calculation?

**Question 1:** Do you already have all the data values you need on the visualization?

If the answer is **yes**: You can use a table calculation.

If the answer is **no**: Use a basic calculation.

**Example:**

Consider the following two visualizations. The visualization on the left is a bar chart that shows the total sales per country/region. The visualization on the right also shows sales per country/region, but sales has been disaggregated.

How could you calculate the 90th percentile of sales for each of these visualizations?
The bar chart on the left is aggregated by SUM. Therefore, there is not enough detail in this view to use a table calculation. You can use a basic aggregate expression to calculate the 90th percentile of sales for each country in this example using the following formula:

$$\text{PERCENTILE}([\text{Sales}], .90)$$

This results in a value for 90th percentile per country as a label for each bar.

However, the chart on the left includes a data value for every sales order. A larger distribution and outliers are shown. There is enough detail in the view to use a table calculation.

You can calculate the 90th percentile of sales for each country by using a distribution band (equivalent to a table calculation). There is more context in this visualization.
Both calculations achieve the same values, but the insights you gather from each differ based on the level of detail (the amount of data) in the visualization.

Basic expression or Level of Detail (LOD) expression?

If you don’t have all the data you need on the visualization, you need your calculation to be passed through to the data source. This means you must use a basic calculation or an LOD Expression.

If you answered no to question 1 ask yourself this:

**Question 2:** Does the granularity of your question match either the granularity of the visualization or the granularity of the data source?

If the answer is yes: Use a basic expression.

If the answer is no: Use a Level of Detail (LOD) expression.
Example

Consider the following visualization. It shows the 90th percentile of sales for all orders in each country.

This example uses the Sample-Superstore data source that comes with Tableau. If you are familiar with the Sample-Superstore data source, you might know that there is one row of data per Order ID. Therefore, the granularity of the data source is Order ID. The granularity of the visualization, however, is Country.

If you want to know what the 90th percentile value of sales is for orders in each country at the order ID level of granularity, you can use the following LOD expression:

```
{INCLUDE [Order ID] : SUM([Sales])}
```
You can then change the field to aggregate at the 90th percentile in the view.

To do so, click the field drop-down and select **Measure > Percentile > 90**.

The following diagram demonstrates how the LOD Expression works in this case:

1. The data starts completely aggregated at SUM(Sales) and then moves down to the Country level of detail: SUM(Sales) at Country.

2. The LOD calculation is applied and the data gains more granularity: SUM(Sales) at Country + Order ID.

3. The LOD calculation is aggregated to the 90th percentile: PCT90(SUM(Sales) at Country + Order ID)

The result is as follows:
Table calculation or Level of Detail (LOD) expression?

When choosing between a table calculation or an LOD calculation, the process is very similar to choosing between a table calculation and a basic expression. Ask yourself the following questions:

*Do you already have all the data values you need on the visualization?*

- If the answer is **Yes**, then use a table calculation.
- If the answer is **No**, then ask yourself: *Does the granularity of the question match either the granularity of the visualization or the granularity of the data source?* If the answer is **No**, then use an LOD calculation.
Table calculations only

There are some scenarios where only a table calculation will do. These include:

- Ranking
- Recursion (e.g. cumulative totals)
- Moving calculations (e.g. rolling averages)
- Inter-row calculations (e.g. period vs. period calculations)

If your analysis requires any of these scenarios, use a table calculation.

Example

Consider the following visualization. It shows the average closing price for several stocks between September 2014 and September 2015.
If you want to see the number of times the closing price exceeded its record close value to date, you must use a table calculation, specifically a recursive calculation.

Why? Because table calculations can output multiple values for each partition of data (cell, pane, table), while basic and LOD expressions can only output a single value for each partition or grouping of data.

To calculate the number of times the closing price exceeded its record closing price for each stock, there are a few steps you need to take.

1. You need to consider all the previous values before to tell if you have reached a new maximum close value. You can do this with a `RUNNING_MAX` function. For example, consider the following calculation computed using Day (across the table), titled **Record to Date**:

   ```plaintext
   RUNNING_MAX(AVG([Close]))
   ```

2. Next, you can flag the days when the record was broken using the following calculation computed using Day (across the table), titled **Count Days Record Broken**:

   ```plaintext
   IF AVG([Close]) = [Record to Date] THEN 1
   ```
3. Finally, you can count these days using the following calculation computed using Day (across the table):

```
ELSE 0
END
```

```
RUNNING_SUM([Count Days Record Broken])
```

When you add the final calculated field to the view in place of Avg(Close), you get something like this:

---

*Continue to Tips for Learning How to Create Calculations on the next page*

Also in this series:

- Understanding Calculations in Tableau
- Types of Calculations in Tableau
Tips for Learning How to Create Calculations

Now that you know the difference between the types of calculations and when to use them in your analysis, how do you actually know how to create the formula for them? This is where things can get tricky.

There is no easy way to know exactly how to create the perfect formula; it takes practice and research. However, here are some ways you can start to learn:

- **Know your question or purpose**
  
  If you know the type of data you need, this can help you choose the correct function, as well as format your formula properly. For example, if you need to show profit for an order, you might create a formula similar to the following: SUM (Sales) - SUM (Cost).

- **Learn Tableau functions**
  
  There are many different functions available in Tableau. Each type serves a different purpose. To learn more, see Functions in Tableau on page 1275.

- **Learn how to format calculations**
  
  Once you are familiar with the different types of Tableau functions and their purpose, make sure to learn how to format calculations using the proper syntax. To learn more, see Formatting Calculations in Tableau on page 1263.

- **Learn from other examples**
  
  Learning by example is very effective. Try to collect a list of common calculations to use in your everyday analysis. There are several resources available to you:

  - Tableau Community Resource: Calculation Reference Library
  - Tableau Blog: Top 10 Tableau table calculations
  - Tableau Blog: Top 15 LOD Expressions
  - Tableau Blog: Save time and stay consistent with a template of calculated fields

Still having trouble? Ask for help on the Tableau Community.

Also in this series:

Understanding Calculations in Tableau

Types of Calculations in Tableau

Choosing the Right Calculation Type on page 1244
More calculations content:

Get Started with Calculations in Tableau on page 1221

Create a Simple Calculated Field on page 1227

Best Practices for Creating Calculations in Tableau

This article describes several tips and guidelines for creating efficient calculations in Tableau. These guidelines are meant to help you optimize your workbook performance. For more information about all the ways you can improve workbook performance, see the Optimize Workbook Performance on page 2438 series.

General Rule: Avoid using a calculated field multiple times in another calculation

Referencing the same calculated field multiple times within another calculation will result in performance issues. If you use a calculated field within a calculation (also known as creating a Nested Calculation), try to reference it only once in the calculation.

Note that referencing a field (terminal field) multiple times in a calculation shouldn’t degrade performance.

Example

Let’s say you create a calculated field that uses a complicated multiple line calculation to find mentions, or Twitter handles, in tweets. The calculated field is titled, Twitter Handle. Each handle that is returned starts with the '@' sign (for example: @user).

For your analysis, you want to remove the '@' symbol.

To do so, you can use the following calculation to remove the first character from the string:

```
RIGHT([Twitter Handle], LEN([Twitter Handle]) -1)
```

This calculation is quite simple. However, since it references the Twitter Handle calculation twice, it performs that calculation twice for each record in your data source: once for the RIGHT function and again for the LEN function.

In order to avoid calculating the same calculation more than once, you can rewrite the calculation to one that uses the Twitter Handle calculation only once. In this example, you can use MID to accomplish the same goal:
Tip 1: Convert multiple equality comparisons to a CASE expression or a group

Let's say you have the following calculation, which uses the calculated field, Person (calc), multiple times and employs a series of OR functions. This calculation, though a simple logical expression, will cause query performance issues because it performs the Person (calc) calculation at least ten times.

```sql
IF [Person (calc)] = 'Henry Wilson'
OR [Person (calc)] = 'Jane Johnson'
OR [Person (calc)] = 'Michelle Kim'
OR [Person (calc)] = 'Fred Suzuki'
OR [Person (calc)] = 'Alan Wang'
THEN 'Lead'
ELSEIF [Person (calc)] = 'Susan Nguyen'
OR [Person (calc)] = 'Laura Rodriguez'
OR [Person (calc)] = 'Ashley Garcia'
OR [Person (calc)] = 'Andrew Smith'
OR [Person (calc)] = 'Adam Davis'
THEN 'IC'
END
```

Instead of using an equality comparison, try the following solutions.

Solution 1

Use a CASE expression. For example:

```sql
CASE [Person (calc)]
WHEN 'Henry Wilson' THEN 'Lead'
WHEN 'Jane Johnson' THEN 'Lead'
WHEN 'Michelle Kim' THEN 'Lead'
WHEN 'Fred Suzuki' THEN 'Lead'
WHEN 'Alan Wang' THEN 'Lead'
```
WHEN 'Susan Nguyen' THEN 'IC'
WHEN 'Laura Rodriguez' THEN 'IC'
WHEN 'Ashley Garcia' THEN 'IC'
WHEN 'Andrew Smith' THEN 'IC'
WHEN 'Adam Davis' THEN 'IC'
END

In this example, the calculated field, Person (calc), is only referenced once. Therefore, it is only performed once. CASE expressions are also further optimized in the query pipeline, so you gain an additional performance benefit.

Solution 2

Create a group instead of a calculated field. For more information, see Group Your Data on page 990.

Tip 2: Convert multiple string calculations into a single REGEXP expression

Note: REGEXP calculations are available only when using Tableau data extracts or when connected to Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), and Oracle data sources. For more information, see Additional Functions on page 1373.

Example 1: CONTAINS

Let’s say you have the following calculation, which uses the calculated field, Category (calc), multiple times. This calculation, though also a simple logical expression, will cause query performance issues because it performs the Category (calc) calculation multiple times.

IF CONTAINS([Segment (calc)],'UNKNOWN')
OR CONTAINS([Segment (calc)],'LEADER')
OR CONTAINS([Segment (calc)],'ADVERTISING')
OR CONTAINS([Segment (calc)],'CLOSED')
OR CONTAINS([Segment (calc)],'COMPETITOR')
OR CONTAINS([Segment (calc)], 'REPEAT')
THEN 'UNKNOWN'
ELSE [Segment (calc)] END

You can use a REGEXP expression to get the same results without as much repetition.

Solution

IF REGEXP_MATCH([Segment (calc)], '
UNKNOWN|LEADER|ADVERTISING|CLOSED|COMPETITOR|REPEAT') THEN
'UNKNOWN'
ELSE [Segment (calc)] END

With string calculations that use a similar pattern, you can use the same REGEXP expression.

Example 2: STARTSWITH

IF STARTSWITH([Segment (calc)], 'UNKNOWN')
OR STARTSWITH([Segment (calc)], 'LEADER')
OR STARTSWITH([Segment (calc)], 'ADVERTISING')
OR STARTSWITH([Segment (calc)], 'CLOSED')
OR STARTSWITH([Segment (calc)], 'COMPETITOR')
OR STARTSWITH([Segment (calc)], 'REPEAT')
THEN 'UNKNOWN'

Solution

IF REGEXP_MATCH([Segment (calc)], '
UNKNOWN|LEADER|ADVERTISING|CLOSED|COMPETITOR|REPEAT') THEN
'UNKNOWN'
ELSE [Segment (calc)] END

Note that the '^' symbol is used in this solution.
Example 3: ENDSWITH

IF ENDSWITH([Segment (calc)], 'UNKNOWN')
OR ENDSWITH([Segment (calc)], 'LEADER')
OR ENDSWITH([Segment (calc)], 'ADVERTISING')
OR ENDSWITH([Segment (calc)], 'CLOSED')
OR ENDSWITH([Segment (calc)], 'COMPETITOR')
OR ENDSWITH([Segment (calc)], 'REPEAT')
THEN 'UNKNOWN'
ELSE [Segment (calc)] END

Solution

IF REGEXP_MATCH([Segment (calc)], '
(UNKNOWN|LEADER|ADVERTISING|CLOSED|COMPETITOR|REPEAT)\$')
THEN 'UNKNOWN'
ELSE [Segment (calc)] END

Note that the '$' symbol is used in this solution.

Tip 3: Manipulate strings with REGEXP instead of LEFT, MID, RIGHT, FIND, LEN

Regular expressions can be a very powerful tool. When doing complex string manipulation, consider using regular expressions. In a lot of cases, using a regular expression will result in a shorter and more efficient calculation. For more information about regular expressions, see the Become a regex regular and wrangle imperfect data post on the Tableau blog.

Example 1

Let’s say you have the following calculation, which removes protocols from URLs. For example: "https://www.tableau.com" becomes "www.tableau.com".

IF (STARTSWITH([Server], "http://")) THEN
MID([Server], Len("http://") + 1)
ELSEIF(STARTSWITH([Server], "https://")) THEN
MID([Server], Len("https://") + 1)

ELSEIF (STARTSWITH([Server], "tcp:")) THEN
    MID([Server], Len("tcp:"), 1)
ELSEIF (STARTSWITH([Server], "\\")) THEN
    MID([Server], Len("\\"), 1)
ELSE
    [Server]
END

Solution

You can simplify the calculation and improve performance by using a REGEXP_REPLACE function.

REGEXP_REPLACE([Server], "^\(http://|https://|tcp://|\\\)\", "")

Example 2

Let's say you have the following calculation, which returns the second part of an IPv4 address. For example: "172.16.0.1" becomes "16".

IF (FINDNTH([Server], ".", 2) > 0) THEN
    MID([Server],
    FIND([Server], ".") + 1,
    FINDNTH([Server], ".", 2) - FINDNTH([Server], ".", 1) - 1
)
END

Solution

You can simplify the calculation and improve performance by using a REGEXP_EXTRACT function.

REGEXP_EXTRACT([Server], ".\([^\\.]\)*\)\")
Tip 4: Do not use sets in calculations

If you are using sets in a calculation, consider replacing them with an alternative, but equivalent calculation.

Example

Let's say you have the following calculation, which uses the set, Top Customers (set).

```
IF ISNULL([Customer Name]) OR [Top customers (set)] THEN [Segment] ELSE [Customer Name] END
```

Solution 1

If the set is simple, you can create a calculated field that returns the same result as the set. For example:

```
CASE [Customer Name]
WHEN 'Henry Wilson' THEN True
WHEN 'Jane Johnson' THEN True
WHEN 'Michelle Kim' THEN True
WHEN 'Fred Suzuki' THEN True
WHEN 'Alan Wang' THEN True
ELSE False
END
```

Note: Using the pattern WHEN TRUE ... ELSE is recommended in this situation to avoid performance issues due to the use of sets. It is not a recommended pattern in most scenarios.

Solution 2

If the set is more complex, consider creating a group that maps all the elements in the set to a given value or attribute, such as 'IN', and then modify the calculation to check for that value/attribute. For example:
IF ISNULL([Customer Name]) OR [Top Customers(group)]='IN' THEN [Segment] ELSE [Customer Name] END

For more information, see Group Your Data on page 990 and Create Sets on page 1004.

Tip 5: Do not use sets to group your data

Sets are meant to make comparisons on subsets of data. Groups are meant to combine related members in a field. Converting sets to groups, such as with the following example, is not recommended:

IF [Americas Set] THEN "Americas"
ELSEIF [Africa Set] THEN "Africa"
ELSEIF [Asia Set] THEN "Asia"
ELSEIF [Europe Set] THEN "Europe"
ELSEIF [Oceania Set] THEN "Oceania"
ELSE "Unknown"
END

This is not recommended for the following reasons:

- **Sets are not always exclusive.** Some members can appear in multiple sets. For example, Russia could be placed both in the Europe set and the Asia set.

- **Sets cannot always be translated to groups.** If the sets are defined by exclusion, conditions, or limits, it might be difficult or even impossible to create an equivalent group.

**Solution**

Group your data using the Group feature. For more information, see Group Your Data on page 990.

See Also

Create Efficient Calculations on page 2448
Optimize Workbook Performance on page 2438
Types of Calculations in Tableau on page 1232
Functions in Tableau on page 1275
Formatting Calculations in Tableau

This article describes how to create and format calculations in Tableau. It lists the basic components of calculations, and explains the proper syntax for each.

In this article

Calculation building blocks below
At a glance: calculation syntax on the next page
Calculation syntax in detail on page 1265
Understanding data types in calculations on page 1273

Calculation building blocks

There are four basic components to calculations in Tableau:

- **Functions** - Statements used to transform the values or members in a field.
- **Fields** - Dimensions or measures (columns) from your data source.
- **Operators** - Symbols that denote an operation.
- **Literal expressions** - Constant values that are represented “as is”, such as "Profitable" and "Unprofitable".

For example, consider the following calculation:

```plaintext
IF [Profit per Day] > 000 THEN "Highly Profitable"
ELSEIF [Profit per Day] <= 0 THEN "Unprofitable"
ELSE "Profitable"
END
```

The components of this calculation can be broken down into the following:

**Functions**: IF, THEN, ELSEIF, ELSE, and END

**Field(s)**: Profit per Day

**Operators**: > and <=

**Literal expressions**:
- String literals: "Highly Profitable", "Unprofitable", and "Profitable"
- Numeric literals: 2000 and 0
Note that not all calculations need to contain all four components. For example, a calculation might not contain a literal expression, such as the calculation \( \text{SUM}([\text{Sales}]) / \text{SUM}([\text{Cost}]), \) which only contains the function, \( \text{SUM} \); the division operator (\(/\)); and the fields, Sales and Cost.

Additionally, calculations can contain:

- **Parameters** - Placeholder variables that can be inserted into calculations to replace constant values.

- **Comments** - Notes about a calculation or its parts, not included in the computation of the calculation.

For more information about how to use and format each of these components in a calculation, see the sections below.

---

**At a glance: calculation syntax**

<table>
<thead>
<tr>
<th>Components</th>
<th>Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functions</strong></td>
<td>See <a href="#">Tableau Functions (Alphabetical) on page 1449</a> for examples of how to format all functions in Tableau.</td>
<td>( \text{SUM} ) (expression)</td>
</tr>
<tr>
<td><strong>Fields</strong></td>
<td>A field in a calculation is often surrounded by brackets [ ]. See <a href="#">Field syntax on page 1267</a> for more information.</td>
<td>([\text{Sales}])</td>
</tr>
<tr>
<td><strong>Operators</strong></td>
<td>(+, -, *, /, %, =, =, &gt;, &lt;, &gt;=, &lt;=, !=, &lt;&gt;, ^, \text{AND}, \text{OR}, \text{NOT}, ( )). See <a href="#">Operator syntax on page 1268</a> for information on the types of operators you can use in Tableau calculations, as well as the order they are performed in a formula.</td>
<td>([\text{Profit}] - [\text{Sales}])</td>
</tr>
<tr>
<td><strong>Literal expressions</strong></td>
<td>Numeric literals are written as numbers. String literals are written with quotation marks. Date literals are written with the # symbol. Boolean literals are written as either true or false.</td>
<td>1.3567 or 27 &quot;Unprofitable&quot; #August 22, 2005#</td>
</tr>
</tbody>
</table>
Null literals are written as null.
See **Literal expression syntax** on page 1271 for more information.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>A parameter in a calculation is surrounded by brackets [ ]. See <strong>Create Parameters</strong> on page 1031 for more information.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comments</td>
<td>To enter a comment in a calculation, type two forward slashes // . See <strong>Add comments to a calculation</strong> on page 1273 for more information.</td>
</tr>
</tbody>
</table>
|            | SUM ([Sales])/SUM ([Profit])  
//John's calculation  
//To be used for profit ratio  
//Do not edit |

Calculation syntax in detail

See the following sections to learn more about the different components of Tableau calculations and how to format them to work in Tableau.

**Function syntax**

In Tableau, functions are the main components of a calculation and can be used for a variety of different purposes.

Functions are colored blue in Tableau calculations, with the exception of logical functions, which are colored black.

Every function in Tableau requires a particular syntax. For instance, the SUM function requires the following syntax: `SUM(expression)` (i.e. `SUM([Sales])`). The PERCENTILE function, on the other hand, requires the following syntax: `PERCENTILE(expression, number)` (i.e. `PERCENTILE([Sales],0.90)`).

At any time in Tableau, you can look up how to use and format a particular function.

To open the list of functions in Tableau:
1. Select **Analysis > Create Calculated Field**.

2. In the Calculation Editor that opens, click the triangle icon (located on the right edge of the editor).

   A list of functions appears for you to browse. When you select a function from the list, the section on the far right updates with information about that function’s required syntax (1) and description (2), as well as with one or more examples (3).

   ![Function selection screenshot]

**Using multiple functions in a calculation**

You can use more than one function in a calculation. For example:

\[
ZN(SUM([\text{Order Quantity}])) - \text{WINDOW}\_AVG(SUM([\text{Order Quantity}]))
\]

There are three functions in the calculation: ZN, SUM, and WINDOW\_AVG.

The ZN function and the WINDOW\_AVG function are separated with the subtraction operator (-).

A function can also be part of another function (or, nested), as is the case with the \(ZN(SUM([\text{Order Quantity}]))\) portion of the example above. In this case, the SUM of Order Quantity is computed before the ZN function because it is inside parentheses. For more information on why, see **Parentheses on page 1271**.

---

**Back to top**
Field syntax

Fields can be inserted into your calculations. Often, a function’s syntax will indicate where a field should be inserted into the calculation. For example: \texttt{SUM(expression)}.

Field names should be encompassed by brackets [ ] in a calculation when the field name contains a space or is not unique. For example, [Sales Categories].

The type of function you use will determine the type of field you use. For example, with the \texttt{SUM} function, you can insert a numerical field, but you cannot insert a date field. For more information, see \textbf{Understanding data types in calculations} on page 1273.

The fields you choose to include in your calculations also depends on the purpose of calculation. For example, if you want to calculate profit ratio your calculation will use the Sales and Profit fields from your data source:

\[
\text{SUM([Sales])}/\text{SUM([Profit])}
\]

To add a field to a calculation, do one of the following:

- Drag it from the Data pane or the view and drop it in the calculation editor.
- In the Calculation Editor, type the field name. Note: The Calculation Editor attempts to auto-complete field names.

Fields are colored orange in Tableau calculations.
Operator syntax

To create calculations, you need to understand the operators supported by Tableau. This section discusses the basic operators that are available, as well as the order (precedence) they are performed.

Operators are colored black in Tableau calculations.

+ (addition)

The + operator means addition when applied to numbers and concatenation when applied to strings. When applied to dates, it can be used to add a number of days to a date. For example:

7 + 3
Profit + Sales
'abc' + 'def' = 'abcdef'
#April 15, 2004# + 15 = #April 30, 2004#

- (subtraction)

The - operator means subtraction when applied to numbers and negation if applied to an expression. When applied to dates, it can be used to subtract a number of days from a date. Hence it can also be used to calculate the difference in days between two dates. For example:

7 - 3
Profit - Sales
-(7+3) = -10
#April 16, 2004# - 15 = #April 1, 2004#
#April 15, 2004# - #April 8, 2004# = 7

* (multiplication)

The * operator means numeric multiplication. For example, 5 * 4 = 20.
/ (division)
The / operator means numeric division. For example, \(20 / 4 = 5\).

% (modulo)
The % operator returns the remainder of a division operation. For example, \(9 \% 2\) returns 1 because 2 goes into 9 four times with a remainder of 1. Modulo can only operate on integers.

==, =, >, <, >=, <=, !=, <> (comparisons)
These are the basic comparison operators that can be used in expressions. Their meanings are as follows: == or = (equal to), > (greater than), < (less than), >= (greater than or equal to), <= (less than or equal to), != and <> (not equal to).
Each operator compares two numbers, dates, or strings and returns a either TRUE, FALSE, or NULL.

^ (power)
This symbol is equivalent to the POWER function. It raises a number to the specified power.
For example:
\(6^3 = 216\)

AND
This is a logical operator. An expression or a boolean must appear on either side of it. For example,
\[ 	ext{IIF(Profit =100 AND Sales =1000, "High", "Low")} \]
If both expressions are TRUE (i.e., not FALSE and not NULL), then the result is TRUE. If either expression is NULL, then the result is NULL. In all other cases, the result is FALSE.
If you create a calculation in which the result of an AND comparison is displayed on a worksheet, Tableau displays TRUE and FALSE. If you would like to change this, use the Format area in the format dialog.
The AND operator employs "short circuit evaluation." This means that if the first expression is evaluated to be TRUE, then the second expression is not evaluated at all. This can be helpful if the second expression results in an error when the first expression is TRUE, because the second expression in this case is never evaluated.
OR

This is a logical operator. An expression or a boolean must appear on either side of it. For example,

IIF(Profit =100 OR Sales =1000, "High", "Low")

If either expression is TRUE, then the result is TRUE. If both expressions are FALSE, then the result is FALSE. If both expressions are NULL, then the result is NULL.

If you create a calculation in which the result of an OR comparison is displayed on a worksheet, Tableau displays TRUE and FALSE. If you would like to change this, use the Format area in the format dialog.

The OR operator employs "short circuit evaluation." This means that if the first expression is evaluated to be TRUE, then the second expression is not evaluated at all. This can be helpful if the second expression results in an error when the first expression is TRUE, because the second expression in this case is never evaluated.

NOT

This is a logical operator. It can be used to negate another boolean or an expression. For example,

IIF(NOT(Sales = Profit),"Not Equal","Equal")

Operator precedence

All operators in a calculation are evaluated in a specific order. For example, 2*1+2 is equal to 4 and not equal to 6, because multiplication is performed before addition (the * operator is always evaluated before the + operator).

If two operators have the same precedence (such as addition and subtraction (+ or -) they are evaluated from left to right in the calculation.

Parentheses can be used to change the order of precedence. See the Parentheses on the next page section for more information.

<table>
<thead>
<tr>
<th>Precedence</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-(negate)</td>
</tr>
<tr>
<td>2</td>
<td>^ (power)</td>
</tr>
<tr>
<td>Precedence</td>
<td>Operator</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>3</td>
<td>*, /, %</td>
</tr>
<tr>
<td>4</td>
<td>+, –</td>
</tr>
<tr>
<td>5</td>
<td>==, &gt;, &lt;, &gt;=, &lt;=, !=</td>
</tr>
<tr>
<td>6</td>
<td>NOT</td>
</tr>
<tr>
<td>7</td>
<td>AND</td>
</tr>
<tr>
<td>8</td>
<td>OR</td>
</tr>
</tbody>
</table>

Parentheses

Parentheses can be used as needed to force an order of precedence. Operators that appear within parentheses are evaluated before those outside of parentheses, starting from the innermost parentheses and moving outward.

For example, \((1 + (2*2+1)*(3*6/3)) = 31\) because the operators within the innermost parentheses are performed first. The calculation is calculated in the following order:

1. \((2*2+1) = 5\)
2. \((3*6/3) = 6\)
3. \((1+ 5*6) = 31\)

Literal expression syntax

This section describes the proper syntax for using literal expressions in Tableau calculations. A literal expression signifies a constant value that is represented “as is.” When you are using functions you will sometimes want to use literal expressions to represent numbers, strings, dates, and more.

For example, you may have a function where your input is a date. Rather than just type “May 1, 2005”, which would be interpreted a string, you would type #May 1, 2005#. This is equivalent to using a date function to convert the argument from a string to a date (refer to Date Functions on page 1298).

You can use numeric, string, date, boolean, and null literals in Tableau calculations. Each type, and how to format them, are described below.
Literal expressions are colored black and gray in Tableau calculations.

**Numeric Literals**

A numeric literal is written as a number. For example, to input the number one as a numeric literal, enter 1. If you want to input the number 0.25 as a numeric literal, enter 0.25.

**String Literals**

A string literal can be written either using single quote or double quote. If your string has a single or double quote within it, simply type the symbol twice. For example, to input the string “cat” as a string literal, enter ‘cat’ or “cat”.

Additionally, if you want to type the string “She’s my friend.” as a string literal, type `She’s my friend.` or “She’s my friend.”

**Date Literals**

Date literals are signified by the pound symbol (#). To input the date “August 22, 2005” as a literal date, enter the ISO formatted date, #2005-08-22#.

**Boolean Literals**

Boolean literals are written as either true or false. To input “true” as a boolean literal, enter true.

**Null Literals**

Null literals are written as Null. To input “Null” as a Null literal, enter Null.

---

**Add parameters to a calculation**

Parameters are placeholder variables that can be inserted into calculations to replace constant values. When you use a parameter in a calculation, you can then expose a parameter control in a view or dashboard to allow users to dynamically change the value.

For details, see [Use a parameter in a calculation](#) on page 1036.

Parameters are colored purple in Tableau calculations.
Add comments to a calculation

You can add comments to a calculation to make notes about it or its parts. Note that comments are not included in the computation of the calculation.

To add a comment to a calculation, type two forward slash (//) characters.

For example:

```
SUM([Sales])/SUM([Profit])  //John’s calculation
```

In this example, //John’s calculation is a comment.

A comment starts at the two forward slashes (//) and goes to the end of the line. To continue with your calculation, you must start a new line.

A multiline comment can be written by starting each line with two forward slashes (/()). For example:

```
SUM([Sales])/SUM([Profit])  //John's calculation
//To be used for profit ratio
//Do not edit
```

Comments are colored gray in Tableau calculations.

Understanding data types in calculations

Tableau supports string, date/datetime, number, and boolean data types. If you create calculated fields, you need to know how to use and combine the different data types in calculations. Many functions that are available to you when you define a calculation only work when they are applied to specific data types.

For example, the DATEPART() function can accept only a date/datetime data type as an argument. You can enter DATEPART('year', #2004-04-15#) and expect a valid result: 2004. You cannot enter DATEPART('year', "Tom Sawyer") and expect a valid result. In fact, this example returns an error because "Tom Sawyer" is a string, not a date/datetime.
Note: Although Tableau will attempt to fully validate all calculations, some data type errors cannot be found until the query is run against the database. These issues appear as error dialogs at the time of the query rather than in the calculation dialog box.

The data types supported by Tableau are described below. Refer to Type Conversion on page 1309 to learn about converting from one data type to another.

STRING

A sequence of zero or more characters. For example, "Wisconsin", "ID-44400", and "Tom Sawyer" are all strings. Strings are recognized by single or double quotes. The quote character itself can be included in a string by repeating it. For example, 'O''Hanrahan'.

DATE/DATETIME

A date or a datetime. For example "January 23, 1972" or "January 23, 1972 12:32:00 AM". If you would like a date written in long-hand style to be interpreted as a a date/datetime, place the # sign on either side of it. For instance, "January 23, 1972" is treated as a string data type but #January 23, 1972# is treated as a date/datetime data type.

NUMBER

Numerical values in Tableau can be either integers or floating-point numbers.

With floating-point numbers, results of some aggregations may not always be exactly as expected. For example, you may find that the SUM function returns a value such as -1.42e-14 for a column of numbers that you know should sum to exactly 0. This happens because the Institute of Electrical and Electronics Engineers (IEEE) 754 floating-point standard requires that numbers be stored in binary format, which means that numbers are sometimes rounded at extremely fine levels of precision. You can eliminate this potential distraction by formatting the number to show fewer decimal places. For more information see, Tableau Functions (by Category) on page 1380 and select Numbers.

Operations that test floating point values for equality can behave unpredictably for the same reason. Such comparisons can occur when using level of detail expressions as dimensions, in categorical filtering, creating ad-hoc groups, creating IN/OUT sets, and with data blending.
Note: The largest signed 64-bit integer is $9,223,372,036,854,775,807$. When connecting to a new data source, any column with data type set to Number (whole) can accommodate values up to this limit; for larger values, because Number (whole) does not use floating-points, Tableau displays "Null". When the data type is set to Number (decimal), larger values can be accommodated.

**BOOLEAN**

A field that contains the values TRUE or FALSE. An unknown value arises when the result of a comparison is unknown. For example, the expression $7 > \text{Null}$ yields unknown. Unknown booleans are automatically converted to Null.

See Also

- **Understanding Calculations in Tableau** on page 1229
- **Create a Simple Calculated Field** on page 1227
- **Get Started with Calculations in Tableau** on page 1221

Functions in Tableau

Tableau supports many functions for use in Tableau calculations. To learn more about the functions in Tableau, see the following articles and reference topics:

**Number Functions**

This article introduces number functions and their uses in Tableau. It also demonstrates how to create a number calculation using an example.

In this article

- **Why use number functions** on the next page
- **List of number functions available in Tableau**
- **Create a number calculation example**
Why use number functions

Number functions allow you to perform computations on the data values in your fields. Number functions can only be used with fields that contain numerical values. For more information, see Data Types on page 263.

For example, you might have a field that contains values for the variance in your budget, titled Budget Variance. One of those values might be -7. You can use the ABS function to return the absolute value of that number, and all the other numbers in that field.

The calculation might look something like this:

\[
\text{ABS}[\text{Budget Variance}]
\]

Therefore, \( \text{ABS}(-7) = 7 \).

Number functions available in Tableau

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ABS</strong></td>
<td><code>ABS(number)</code></td>
<td>Returns the absolute value of the given number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Examples:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \text{ABS}(-7) = 7 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \text{ABS}([\text{Budget Variance}]) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The second example returns the absolute value for all the numbers contained in the Budget Variance field.</td>
</tr>
<tr>
<td><strong>ACOS</strong></td>
<td><code>ACOS(number)</code></td>
<td>Returns the arc cosine of the given number. The result is in radians.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \text{ACOS}(-1) = 3.14159265358979 )</td>
</tr>
<tr>
<td>Function</td>
<td>Table</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td></td>
</tr>
</tbody>
</table>
| **ASIN** | ASIN(number) | Returns the arc sine of a given number. The result is in radians.  <br>Example:  
\[ \text{ASIN}(1) = 1.5707963267949 \] |
| **ATAN** | ATAN(number) | Returns the arc tangent of a given number. The result is in radians.  
Example:  
\[ \text{ATAN}(180) = 1.5652408283942 \] |
| **ATAN2** | ATAN2(y number, x number) | Returns the arc tangent of two given numbers (x and y). The result is in radians.  
Example:  
\[ \text{ATAN2}(2, 1) = 1.10714871779409 \] |
| **CEILING** | CEILING(number) | Rounds a number to the nearest integer of equal or greater value.  
Example:  
\[ \text{CEILING}(3.1415) = 4 \] |

**Availability by data source:**

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Access</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Supported</td>
</tr>
<tr>
<td>Text File</td>
<td>Supported</td>
</tr>
<tr>
<td>Statistical File</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>Status</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Tableau Server</td>
<td>Supported</td>
</tr>
<tr>
<td>Actian Vector</td>
<td>Not supported</td>
</tr>
<tr>
<td>Amazon Aurora</td>
<td>Not supported</td>
</tr>
<tr>
<td>Amazon EMR Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td>Not supported</td>
</tr>
<tr>
<td>Aster Database</td>
<td>Not supported</td>
</tr>
<tr>
<td>Cloudera Hadoop</td>
<td>Supported</td>
</tr>
<tr>
<td>DataStax Enterprise</td>
<td>Supported</td>
</tr>
<tr>
<td>EXASOL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Firebird</td>
<td>Not supported</td>
</tr>
<tr>
<td>Google Analytics</td>
<td>Supported</td>
</tr>
<tr>
<td>Google BigQuery</td>
<td>Supported</td>
</tr>
<tr>
<td>Google Cloud SQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Hortonworks Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>IBM BigInsights</td>
<td>Not supported</td>
</tr>
<tr>
<td>IBM DB2</td>
<td>Not supported</td>
</tr>
<tr>
<td>IBM PDA (Netezza)</td>
<td>Not supported</td>
</tr>
<tr>
<td>MapR Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>MarkLogic</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft Analysis Services</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft PowerPivot</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>Not supported</td>
</tr>
<tr>
<td>Database</td>
<td>Support</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>MySQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Oracle</td>
<td>Not supported</td>
</tr>
<tr>
<td>Oracle Essbase</td>
<td>Not supported</td>
</tr>
<tr>
<td>Actian Matrix (ParAccel)</td>
<td>Not supported</td>
</tr>
<tr>
<td>Pivotal Greenplum</td>
<td>Not supported</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Progress OpenEdge</td>
<td>Not supported</td>
</tr>
<tr>
<td>Salesforce</td>
<td>Supported</td>
</tr>
<tr>
<td>SAP HANA</td>
<td>Not supported</td>
</tr>
<tr>
<td>SAP Sybase ASE</td>
<td>Not supported</td>
</tr>
<tr>
<td>SAP Sybase IQ</td>
<td>Not supported</td>
</tr>
<tr>
<td>Spark SQL</td>
<td>Supported</td>
</tr>
<tr>
<td>Splunk</td>
<td>Not supported</td>
</tr>
<tr>
<td>Teradata</td>
<td>Not supported</td>
</tr>
<tr>
<td>Teradata OLAP Connector</td>
<td>Not supported</td>
</tr>
<tr>
<td>Vertica</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

### COS

**COS(number)**

Returns the cosine of an angle. Specify the angle in radians.

Example:

```
COS(PI( ) /4) = 0.707106781186548
```
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>COT(PI( )/4)</td>
<td>Returns the cotangent of an angle in radians.</td>
<td>COT(PI( )/4) = 1</td>
</tr>
<tr>
<td>DEGREES(number)</td>
<td>Converts a given number in radians to degrees.</td>
<td>DEGREES(PI( )/4) = 45.0</td>
</tr>
<tr>
<td>DIV(integer1, integer2)</td>
<td>Returns the integer part of a division operation, in which integer1 is divided by integer2.</td>
<td>DIV(11,2) = 5</td>
</tr>
<tr>
<td>EXP(number)</td>
<td>Returns e raised to the power of the given number.</td>
<td>EXP(2) = 7.389&lt;br&gt;EXP(-[Growth Rate]*[Time])</td>
</tr>
<tr>
<td>FLOOR(number)</td>
<td>Rounds a number to the nearest integer of equal or lesser value.</td>
<td>FLOOR(3.1415) = 3</td>
</tr>
</tbody>
</table>

**Availability by data source:**

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Access</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Text File</td>
<td>Supported</td>
</tr>
<tr>
<td>Statistical File</td>
<td>Supported</td>
</tr>
<tr>
<td>Tableau Server</td>
<td>Supported</td>
</tr>
<tr>
<td>Actian Vector</td>
<td>Not supported</td>
</tr>
<tr>
<td>Amazon Aurora</td>
<td>Not supported</td>
</tr>
<tr>
<td>Amazon EMR Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td>Not supported</td>
</tr>
<tr>
<td>Aster Database</td>
<td>Not supported</td>
</tr>
<tr>
<td>Cloudera Hadoop</td>
<td>Supported</td>
</tr>
<tr>
<td>DataStax Enterprise</td>
<td>Supported</td>
</tr>
<tr>
<td>EXASOL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Firebird</td>
<td>Not supported</td>
</tr>
<tr>
<td>Google Analytics</td>
<td>Supported</td>
</tr>
<tr>
<td>Google BigQuery</td>
<td>Supported</td>
</tr>
<tr>
<td>Google Cloud SQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Hortonworks Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>IBM BigInsights</td>
<td>Not supported</td>
</tr>
<tr>
<td>IBM DB2</td>
<td>Not supported</td>
</tr>
<tr>
<td>IBM Netezza</td>
<td>Not supported</td>
</tr>
<tr>
<td>MapR Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>MarkLogic</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft Analysis Services</td>
<td>Not supported</td>
</tr>
<tr>
<td>Database</td>
<td>Supported/Supported or Not supported</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>Microsoft PowerPivot</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>Not supported</td>
</tr>
<tr>
<td>MySQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Oracle</td>
<td>Not supported</td>
</tr>
<tr>
<td>Oracle Essbase</td>
<td>Not supported</td>
</tr>
<tr>
<td>ParAccel</td>
<td>Not supported</td>
</tr>
<tr>
<td>Pivotal Greenplum</td>
<td>Not supported</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Progress OpenEdge</td>
<td>Not supported</td>
</tr>
<tr>
<td>Salesforce</td>
<td>Supported</td>
</tr>
<tr>
<td>SAP HANA</td>
<td>Not supported</td>
</tr>
<tr>
<td>SAP Sybase ASE</td>
<td>Not supported</td>
</tr>
<tr>
<td>SAP Sybase IQ</td>
<td>Not supported</td>
</tr>
<tr>
<td>Spark SQL</td>
<td>Supported</td>
</tr>
<tr>
<td>Splunk</td>
<td>Not supported</td>
</tr>
<tr>
<td>Teradata</td>
<td>Not supported</td>
</tr>
<tr>
<td>Teradata OLAP Connector</td>
<td>Not supported</td>
</tr>
<tr>
<td>Vertica</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

**HEXBINX**  
HEXBINX(number, number)  
Maps an x, y coordinate to the x-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately.

HEXBINX and HEXBINY are binning and plotting functions for hexagonal bins. Hexagonal bins are an efficient and elegant option for visualizing data in an x/y plane such as
a map. Because the bins are hexagonal, each bin closely approximates a circle and minimizes variation in the distance from the data point to the center of the bin. This makes the clustering both more accurate and informative.

Example:

```
HEXBINX([Longitude], [Latitude])
```

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEXBINY</td>
<td>Maps an x, y coordinate to the y-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately.</td>
<td>(\text{HEXBINY}([\text{Longitude}], [\text{Latitude}]))</td>
</tr>
<tr>
<td>LN</td>
<td>Returns the natural logarithm of a number. Returns Null if number is less than or equal to 0.</td>
<td>(\text{LN}(\text{number}))</td>
</tr>
<tr>
<td>LOG</td>
<td>Returns the logarithm of a number for the given base. If the base value is omitted, base 10 is used.</td>
<td>(\text{LOG}(\text{number}, \text{base}))</td>
</tr>
</tbody>
</table>
| MAX | Returns the maximum of the two arguments, which must be of the same type. Returns Null if either argument is Null. MAX can also be applied to a single field in an aggregate calculation. | Examples:
  - \(\text{MAX}(4,7)\)
  - \(\text{MAX}(\text{Sales},\text{Profit})\)
  - \(\text{MAX}([\text{First Name}],[\text{Last Name}])\) |
<p>| MIN | Returns the minimum of the two arguments, which must be of the same type. Returns Null if either argument is Null. MIN can also be applied to a single field in an aggregate calculation. | Examples: |</p>
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI</td>
<td>Returns the numeric constant pi: 3.14159.</td>
<td></td>
</tr>
</tbody>
</table>
| POWER (number, power) | Raises the number to the specified power. | \[
\text{POWER}(5, 2) = 5^2 = 25 \\
\text{POWER}(\text{Temperature}, 2) \\
\text{You can also use the} ^ \text{ symbol:} \\
5^2 = \text{POWER}(5, 2) = 25
\] |
| RADIANS (number) | Converts the given number from degrees to radians. | \[
\text{RADIANS}(180) = 3.14159
\] |
| ROUND (number, [decimals]) | Rounds numbers to a specified number of digits. The decimals argument specifies how many decimal points of precision to include in the final result. If decimals is omitted, number is rounded to the nearest integer. | This example rounds every Sales value to an integer: \[
\text{ROUND}(\text{Sales})
\] Some databases, such as SQL Server, allow specification of a negative length, where -1 rounds number to 10’s, -2 rounds to 100’s, and so on. This is not true of all |
For example, it is not true of Excel or Access.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SIGN</strong></td>
<td>Returns the sign of a number: The possible return values are -1 if the number is negative, 0 if the number is zero, or 1 if the number is positive.</td>
<td>If the average of the profit field is negative, then $\text{SIGN(AVG(Profit))} = -1$</td>
</tr>
</tbody>
</table>
| **SIN** | Returns the sine of an angle. Specify the angle in radians. | $\text{SIN}(0) = 1.0$  
$\text{SIN}(\pi /4) = 0.707106781186548$ |
| **SQRT** | Returns the square root of a number. | $\text{SQRT}(25) = 5$ |
| **SQUARE** | Returns the square of a number. | $\text{SQUARE}(5) = 25$ |
| **TAN** | Returns the tangent of an angle. Specify the angle in radians. | $\text{TAN}(\pi /4) = 1.0$ |
| **ZN** | Returns the expression if it is not null, otherwise returns | |
| (expression) | zero. Use this function to use zero values instead of null values. Example: 

ZN([Profit]) = [Profit] |

### Create a number calculation

Follow along with the steps below to learn how to create a number calculation.

1. In Tableau Desktop, connect to the **Sample - Superstore** saved data source, which comes with Tableau.
2. Navigate to a worksheet and select **Analysis > Create Calculated Field**.
3. In the calculation editor that opens, do the following:
   - Name the calculated field **Minimum Sales transaction**
   - Enter the following formula:

```
MIN(Sales)
```

   - When finished, click **OK**.

The new number calculation appears under **Measures** in the **Data** pane. Just like your other fields, you can use it in one or more visualizations.

When Minimum Sales is placed on Text on the Marks card in the worksheet, its name is changed to **AGG(Minimum Sales)**, which indicates that it cannot be aggregated any further, since it is already aggregated down to the lowest level of detail (the smallest sales value for all records).

This example shows the minimum sales per category.
When subcategory is brought into the view, the minimum sales for each subcategory are shown.

<table>
<thead>
<tr>
<th>Category</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Furniture</td>
<td>1.892</td>
</tr>
<tr>
<td>Office Supplies</td>
<td>0.444</td>
</tr>
<tr>
<td>Technology</td>
<td>0.990</td>
</tr>
</tbody>
</table>
See Also

Functions in Tableau on page 1275
Tableau Functions (Alphabetical) on page 1449
Tableau Functions (by Category) on page 1380
Formatting Calculations in Tableau on page 1263

String Functions

This article introduces string functions and their uses in Tableau. It also demonstrates how to create a string calculation using an example.

In this article

Why use string functions on the next page
List of string functions in Tableau
Why use string functions

String functions allow you to manipulate string data (i.e. data made of text).

For example, you might have a field that contains all of your customers' first and last names. One member might be: Jane Johnson. You can pull the last names from all your customers into a new field using a string function.

The calculation might look something like this:

\[
\text{SPLIT([Customer Name], ', ', 2)}
\]

Therefore, \( \text{SPLIT('Jane Johnson', ', ', 2)} = 'Johnson' \).

String functions available in Tableau:

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>\text{ASCII(string)}</td>
<td>Returns the ASCII code for the first character of string.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\text{ASCII('A')} = 65</td>
</tr>
<tr>
<td>CHAR</td>
<td>\text{CHAR(number)}</td>
<td>Returns the character encoded by the ASCII code number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\text{CHAR(65)} = 'A'</td>
</tr>
<tr>
<td>CONTAINS</td>
<td>\text{CONTAINS(string,substring)}</td>
<td>Returns true if the given string contains the specified substring.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>CONTAINS</td>
<td>Substring containing the given string</td>
<td>CONTAINS(&quot;Calculation&quot;, &quot;alcu&quot;) = true</td>
</tr>
<tr>
<td>ENDSWITH</td>
<td>True if the given string ends with the specified substring</td>
<td>ENDSWITH(&quot;Tableau&quot;, &quot;leau&quot;) = true</td>
</tr>
<tr>
<td>FIND</td>
<td>The index position of substring in string, or 0 if not found</td>
<td>FIND(&quot;Calculation&quot;, &quot;alcu&quot;) = 2</td>
</tr>
<tr>
<td>FINDNTH</td>
<td>The position of the nth occurrence of substring within the specified string</td>
<td>FINDNTH(&quot;Calculation&quot;, &quot;alcu&quot;, 2) = 2</td>
</tr>
</tbody>
</table>

Note: FINDNTH is not available for all data sources.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LEFT</strong></td>
<td>Returns the left-most number of characters in the string.</td>
<td><code>LEFT(&quot;Matador&quot;, 4) = &quot;Mata&quot;</code></td>
</tr>
<tr>
<td><strong>LEN</strong></td>
<td>Returns the length of the string.</td>
<td><code>LEN(&quot;Matador&quot;) = 7</code></td>
</tr>
<tr>
<td><strong>LOWER</strong></td>
<td>Returns string, with all characters lowercase.</td>
<td><code>LOWER(&quot;ProductVersion&quot;) = &quot;productversion&quot;</code></td>
</tr>
<tr>
<td><strong>LTRIM</strong></td>
<td>Returns the string with any leading spaces removed.</td>
<td><code>LTRIM(&quot; Matador &quot;) = &quot;Matador &quot;</code></td>
</tr>
<tr>
<td><strong>MAX</strong></td>
<td>Returns the maximum of <code>a</code> and <code>b</code> (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, <code>MAX</code> finds the value that is highest in the sort sequence defined by the database for that column. It returns <code>Null</code> if either argument is <code>Null</code>.</td>
<td>Example:</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>MAX</strong></td>
<td>Returns the maximum of two arguments.</td>
<td></td>
</tr>
<tr>
<td><strong>MID</strong></td>
<td>Returns the string starting at index position <code>start</code>. The first character in the string is position 1. If the optional argument <code>length</code> is added, the returned string includes only that number of characters.</td>
<td></td>
</tr>
<tr>
<td><strong>MIN</strong></td>
<td>Returns the minimum of <code>a</code> and <code>b</code> (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, MIN finds the value that is lowest in the sort sequence. It returns <code>Null</code> if either argument is <code>Null</code>.</td>
<td></td>
</tr>
<tr>
<td><strong>REPLACE</strong></td>
<td>Searches string for <code>substring</code> and replaces it with <code>replacement</code>. If <code>substring</code> is not found, the string is not changed.</td>
<td></td>
</tr>
<tr>
<td><strong>RIGHT</strong></td>
<td>Returns the right-most number of characters in string.</td>
<td></td>
</tr>
</tbody>
</table>

**Examples:**

- `MAX("Apple","Banana") = "Banana"
- `MID("Calculation", 2) = "alculation"
- `MID("Calculation", 2, 5) = "alcul"
- `MIN("Apple","Banana") = "Apple"
- `REPLACE("Version8.5", "8.5", "9.0") = "Version9.0"`
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RTRIM</strong>&lt;br&gt;<strong>RTRIM</strong>(string)</td>
<td>Returns string with any trailing spaces removed.</td>
<td>RIGHT(&quot;Calculation&quot;, 4) = &quot;tion&quot;</td>
</tr>
<tr>
<td><strong>SPACE</strong>&lt;br&gt;<strong>SPACE</strong>(number)</td>
<td>Returns a string that is composed of the specified number of repeated spaces.</td>
<td>SPACE(1) = &quot; &quot;</td>
</tr>
<tr>
<td><strong>SPLIT</strong>&lt;br&gt;<strong>SPLIT</strong>(string, delimiter, token number)</td>
<td>Returns a substring from a string, using a delimiter character to divide the string into a sequence of tokens. The string is interpreted as an alternating sequence of delimiters and tokens. So for the string <code>abc-defgh-i-jkl</code>, where the delimiter character is '-', the tokens are <code>abc</code>, <code>defgh</code>, <code>i</code>, and <code>jkl</code>. Think of these as tokens 1 through 4. SPLIT returns the token corresponding to the token number. When the token number is positive, tokens are counted starting from the left end of the string; when the token number is negative, tokens are counted starting from the right.</td>
<td>Examples:&lt;br&gt;SPLIT (‘a-b-c-d’, ‘-’, 2) = ‘b’&lt;br&gt;SPLIT (‘a</td>
</tr>
</tbody>
</table>

**Note:** The split and custom split commands are
available for the following data sources types: Tableau data extracts, Microsoft Excel, Text File, PDF File, Salesforce, OData, Microsoft Azure Market Place, Google Analytics, Vertica, Oracle, MySQL, PostgreSQL, Teradata, Amazon Redshift, Aster Data, Google Big Query, Cloudera Hadoop Hive, Hortonworks Hive, and Microsoft SQL Server.

Some data sources impose limits on splitting string. The following table shows which data sources support negative token numbers (splitting from the right) and whether there is a limit on the number of splits allow per data source. A SPLIT function that specifies a negative token number and would be legal with other data sources will return this error with these data sources: “Splitting from right is not support by the data source.”

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Left/Right Constraints</th>
<th>Maximum Number of Splits</th>
<th>Version limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tableau Data Extract</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Text file</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Salesforce</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>OData</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Google Analytics</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Tableau</td>
<td>Both</td>
<td>Infinite</td>
<td>Supported</td>
</tr>
<tr>
<td>Data Server</td>
<td>Support</td>
<td>Version</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>Vertica</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>MySQL</td>
<td>Both</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Left only prior to version 9.0; both for version 9.0 and above</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Teradata</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Aster Database</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Google BigQuery</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Hortonworks Hadoop</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Cloudera Hadoop</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Microsoft</td>
<td>Both</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Impala supported starting in version 2.3.0.

Version 14 and later

2008 and
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTSWITH</td>
<td>Returns true if string starts with substring.</td>
<td>STARTSWITH(“Joker”, “Jo”) = true</td>
</tr>
<tr>
<td></td>
<td>Leading white spaces are ignored.</td>
<td></td>
</tr>
<tr>
<td>TRIM</td>
<td>Returns the string with leading and trailing spaces removed.</td>
<td>TRIM(&quot; Calculation &quot;) = &quot;Calculation&quot;</td>
</tr>
<tr>
<td>UPPER</td>
<td>Returns string, with all characters uppercase.</td>
<td>UPPER(&quot;Calculation&quot;) = &quot;CALCULATION&quot;</td>
</tr>
</tbody>
</table>

Create a string calculation

Follow along with the steps below to learn how to create a string calculation.

1. In Tableau Desktop, connect to the **Sample - Superstore** saved data source, which comes with Tableau.
2. Navigate to a worksheet.
3. From the **Data** pane, under Dimensions, drag **Order ID** to the **Rows** shelf.
Notice that every order ID contains values for country (CA and US, for example), year (2011), and order number (100006). For this example, you will create a calculation to pull only the order number from the field.

4. Select **Analysis > Create Calculated Field**.

5. In the calculation editor that opens, do the following:
   - Name the calculated field **Order ID Numbers**.
   - Enter the following formula:
     
     \[
     \text{RIGHT}([\text{Order ID}], \ 6)
     \]
     
     This formula takes the specified digits (6) from the right of the string and pulls them into a new field.

     Therefore, \( \text{RIGHT}('CA-2011-100006', \ 6) = '100006'. \)

   - When finished, click **OK**.

   The new calculated field appears under **Dimensions** in the **Data** pane. Just like your other fields, you can use it in one or more visualizations.

6. From the **Data** pane, drag **Order ID Numbers** to the **Rows** shelf. Place it to the right of **Order ID**.

   Notice how the fields differ now.
This article introduces date functions and their uses in Tableau. It also demonstrates how to create a date calculation using an example.

### In this article

- Why use date functions on the next page
- Gregorian Calendar vs. ISO 8601 Standard on the next page
- List of date functions in Tableau
- Create a date calculation example
- date_part values
Why use date functions

Date functions allow you to manipulate dates in your data source.

For example, you might have a date field with year, month, and day for each value (2004-04-15). From these existing values, you can create new date values with a date function, such as the DATETRUNC function. For example, you can find the date of the beginning of the quarter for any existing date value.

The date calculation might look something like this:

```
DATETRUNC('quarter', [Order Date])
```

So, if the original date is '3/27/2011', using the above calculation would return '1/1/2011' to indicate that Q1 started on January 1. If the original date is '5/3/2011', then the calculation would return '4/1/2011' to indicate that Q2 started on April 1, four months into the year.

See the Create a date calculation on page 1305 section below for an example.

Gregorian Calendar vs. ISO 8601 Standard

If you are using a .hyper extract, date functions can be calculated using the traditional Gregorian calendar or the ISO 8601 Standard. For more information on creating a .hyper extract, see Extract Upgrade to .hyper Format on page 788.

The ISO 8601 format is an international standard for calculating dates and times that differs from the Gregorian calendar due to how the starting week of a year (Week 1) is calculated. In the Gregorian Calendar, the user can define on which day a week begins. In the ISO 8601 Standard, the week always begins on a Monday.

In a Gregorian Calendar when a new year starts, Week 1 of the year is counted as starting on the 1st of January, regardless of where in the weekday the 1st of January occurs. If January 1st falls on a Saturday, then Week 1 will have one day in it and Week 2 will begin on the following Sunday.

In the ISO 8601 format, Week 1 of a new year begins on a Monday and has four or more days in January. For example, if January 1st falls on a Saturday, then Week 1 will not begin until the following Monday, January 3rd. Calculating dates this way makes sure that there are a consistent number of days in Week 1 of a new year.
## Date functions available in Tableau:

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
</table>
| **DATEADD** | `DATEADD(date_part, interval, date)` | Returns the specified date with the specified number interval added to the specified date part of that date. **Supports ISO 8601 dates.**

Example:

```
DATEADD('month', 3, #2004-04-15#) = 2004-07-15 12:00:00 AM
```

This expression adds three months to the date #2004-04-15#.

| **DATEDIFF** | `DATEDIFF(date_part, date1, date2, [start_of_week])` | Returns the difference between `date1` and `date2` expressed in units of `date_part`. **Supports ISO 8601 dates.**

The `start_of_week` parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If it is omitted, the start of week is determined by the data source. See [Date Properties for a Data Source](#) on page 1059.

Examples:

```
DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'monday') = 1
DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'sunday') = 0
```

The first expression returns 1 because when...
<table>
<thead>
<tr>
<th>DATENAME</th>
<th>DATENAME (date_part, date, [start_of_week])</th>
<th>Returns <code>date_part</code> of <code>date</code> as a string. The <code>start_of_week</code> parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If <code>start_of_week</code> is omitted, the start of week is determined by the data source. See Date Properties for a Data Source on page 1059. Supports ISO 8601 dates. Examples:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><code>DATENAME('year', #2004-04-15#) = &quot;2004&quot;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>DATENAME('month', #2004-04-15#) = &quot;April&quot;</code></td>
</tr>
<tr>
<td>DATEPART</td>
<td>DATEPART (date_part, date, [start_of_week])</td>
<td>Returns <code>date_part</code> of <code>date</code> as an integer. The <code>start_of_week</code> parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If <code>start_of_week</code> is omitted, the start of week is determined by the data source. See Date Properties for a Data Source on page 1059. <strong>Note:</strong> When the <code>date_part</code> is weekday, the <code>start_of_week</code> parameter is ignored.</td>
</tr>
</tbody>
</table>

`start_of_week` is 'monday', then 22 September (a Sunday) and 24 September (a Tuesday) are in different weeks. The second expression returns 0 because when `start_of_week` is 'sunday' then 22 September (a Sunday) and 24 September (a Tuesday) are in the same week.
This is because Tableau relies on a fixed weekday ordering to apply offsets.

**Supports ISO 8601 dates.**

Examples:

```
DATEPART('year', #2004-04-15#) = 2004
DATEPART('month', #2004-04-15#) = 4
```

| DATETRUNC | DATETRUNC(date_part, date, [start_of_week]) | Truncates the specified date to the accuracy specified by the date_part. This function returns a new date. For example, when you truncate a date that is in the middle of the month at the month level, this function returns the first day of the month. The start_of_week parameter, which you can use to specify which day is to be considered the first day of the week, is optional. Possible values are 'monday', 'tuesday', etc. If `start_of_week` is omitted, the start of week is determined by the data source. See [Date Properties for a Data Source](#) on page 1059.

**Supports ISO 8601 dates.**

Examples:

```
DATETRUNC('quarter', #2004-08-15#) = 2004-07-01 12:00:00 AM
DATETRUNC('month', #2004-04-15#) = 2004-04-01 12:00:00 AM
```

| DAY | DAY(date) | Returns the day of the given date as an integer.

Example:
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISDATE</td>
<td>Returns true if a given string is a valid date.</td>
<td>ISDATE(&quot;April 15, 2004&quot;) = true</td>
</tr>
<tr>
<td>MAKEDATE</td>
<td>Returns a date value constructed from the specified year, month, and date.</td>
<td>MAKEDATE(2004, 4, 15) = #April 15, 2004#</td>
</tr>
<tr>
<td>MAKEDATETIME</td>
<td>Returns a datetime that combines a date and a time. The date can be a date,</td>
<td>MAKEDATETIME(&quot;1899-12-30&quot;, #07:59:00#) = #12/30/1899 7:59:00 AM#</td>
</tr>
<tr>
<td></td>
<td>datetime, or a string type. The time must be a datetime. This function is</td>
<td>MAKEDATETIME([Date], [Time]) = #1/1/2001 6:00:00 AM#</td>
</tr>
<tr>
<td></td>
<td>available only for MySQL-compatible connections (which for Tableau are, in</td>
<td></td>
</tr>
<tr>
<td></td>
<td>addition to MySQL, Amazon Aurora and Amazon Aurora).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Examples:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKEDATETIME(&quot;1899-12-30&quot;, #07:59:00#) = #12/30/1899 7:59:00 AM#</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKEDATETIME([Date], [Time]) = #1/1/2001 6:00:00 AM#</td>
<td></td>
</tr>
<tr>
<td>MAKETIME</td>
<td>Returns a date value constructed from the specified hour, minute, and second.</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>MAKETIME</td>
<td>Converts minutes, seconds, and years into a date. Available for Tableau Data Extracts. Check for availability in other data sources. Example:</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>Usually applied to numbers but also works on dates. Returns the maximum of a and b (a and b must be of the same type). Returns Null if either argument is Null. Examples:</td>
<td>MAX(#2004-01-01#, #2004-03-01#) = 2004-03-01 12:00:00 AM MAX([ShipDate1], [ShipDate2])</td>
</tr>
<tr>
<td>MIN</td>
<td>Usually applied to numbers but also works on dates. Returns the minimum of a and b (a and b must be of the same type). Returns Null if either argument is Null. Examples:</td>
<td>MIN(#2004-01-01#, #2004-03-01#) = 2004-01-01 12:00:00 AM MIN([ShipDate1], [ShipDate2])</td>
</tr>
<tr>
<td>MONTH</td>
<td>Returns the month of the given date as an integer. Example:</td>
<td>MONTH(#2004-04-15#) = 4</td>
</tr>
<tr>
<td>NOW</td>
<td>Returns the current date and time.</td>
<td></td>
</tr>
</tbody>
</table>
The return varies depending on the nature of the connection:

- For a live, unpublished connection, NOW returns the data source server time.
- For a live, published connection, NOW returns the data source server time.
- For an unpublished extract, NOW returns the local system time.
- For a published extract, NOW returns the local time of the Tableau Server Data Engine. When there are multiple worker machines indifferent time zones, this can produce inconsistent results.

Example:

```
NOW( ) = 2004-04-15 1:08:21 PM
```

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>TODAY</td>
<td>Returns the current date.</td>
<td>TODAY( ) = 2004-04-15</td>
</tr>
<tr>
<td>YEAR</td>
<td>Returns the year of the given date as an integer.</td>
<td>YEAR(#2004-04-15#) = 2004</td>
</tr>
</tbody>
</table>

Create a date calculation

Follow along with the steps below to learn how to create a date calculation.
1. In Tableau Desktop, connect to the Sample-Superstore saved data source, which comes with Tableau.

2. Navigate to a worksheet.

3. From the Data pane, under Dimensions, drag Order Date to the Rows shelf.

4. On the Rows shelf, click the plus icon (+) on the YEAR(Order Date) field. QUARTER(Order Date) is added to the Rows shelf and the view updates.

5. On the Rows shelf, click the plus icon (+) on the QUARTER(Order Date) field to drill down to MONTH(Order Date).
6. Select **Analysis > Create Calculated Field**.

7. In the calculation editor that opens, do the following:
   
   - Name the calculated field, Quarter Date.
   
   - Enter the following formula:

     ```
     DATETRUNC('quarter', [Order Date])
     ```

   - When finished, click **OK**.

     The new date calculated field appears under **Dimensions** in the **Data** pane. Just like your other fields, you can use it in one or more visualizations.

8. From the **Data** pane, under Dimensions, drag **Quarter Date** to the **Rows** shelf and place it to the right of MONTH(Order Date).

     The visualization updates with year values. This is because Tableau rolls date data up to the highest level of detail.

9. On the Rows shelf, right-click **YEAR(Quarter Date)** and select **Exact Date**.

10. On the Rows shelf, right-click **YEAR(Quarter Date)** again and select **Discrete**.

    The visualization updates with the exact quarter date for each row in the table.
date_part values

Many date functions in Tableau use date_part, which is a constant string argument. The valid date_part values that you can use are:

<table>
<thead>
<tr>
<th>date_part</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>'year'</td>
<td>Four-digit year</td>
</tr>
<tr>
<td>'quarter'</td>
<td>1-4</td>
</tr>
<tr>
<td>'month'</td>
<td>1-12 or &quot;January&quot;, &quot;February&quot;, and so on</td>
</tr>
<tr>
<td>'dayofyear'</td>
<td>Day of the year; Jan 1 is 1, Feb 1 is 32, and so on</td>
</tr>
<tr>
<td>'day'</td>
<td>1-31</td>
</tr>
<tr>
<td>'weekday'</td>
<td>1-7 or &quot;Sunday&quot;, &quot;Monday&quot;, and so on</td>
</tr>
<tr>
<td>'week'</td>
<td>1-52</td>
</tr>
<tr>
<td>'hour'</td>
<td>0-23</td>
</tr>
<tr>
<td>'minute'</td>
<td>0-59</td>
</tr>
<tr>
<td>'second'</td>
<td>0-60</td>
</tr>
<tr>
<td>'iso-year'</td>
<td>Four-digit ISO 8601 year</td>
</tr>
<tr>
<td>date_part</td>
<td>Values</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>'iso-quarter'</td>
<td>1-4</td>
</tr>
<tr>
<td>'iso-week'</td>
<td>1-52, start of week is always Monday</td>
</tr>
<tr>
<td>'iso-weekday'</td>
<td>1-7, start of week is always Monday</td>
</tr>
</tbody>
</table>

For more information on formatting date functions, see the **Literal Expressions** section of the **At a glance: calculation syntax** on page 1264 table.

**Note:** Date functions do not take account of the configured fiscal year start. See **Fiscal Dates** on page 1068 for more information.

### See Also

- **Fiscal Dates** on page 1068
- **Dates and Times** on page 1058
- **Date Properties for a Data Source** on page 1059
- **Custom Dates** on page 1066
- **Custom Date Formats** on page 1071
- **Tableau Functions (Alphabetical)** on page 1449
- **Tableau Functions (by Category)** on page 1380
- **Formatting Calculations in Tableau** on page 1263
- **Functions in Tableau** on page 1275

### Type Conversion

This article introduces type conversion functions and their uses in Tableau. It also demonstrates how to create a type conversion calculation using an example.
In this article

Why use type conversion functions below
List of type conversion functions in Tableau
Create a type conversion calculation example

Why use type conversion functions

Type conversion functions allow you to convert fields from one data type to another. For example, you can convert numbers to strings, such as age values (numbers) to string values so that Tableau does not try to aggregate them.

The calculation for such a task might look similar to the following:

```
STR([Age])
```

Type conversion functions available in Tableau:

The result of any expression in a calculation can be converted to a specific data type. The conversion functions are `STR()`, `DATE()`, `DATETIME()`, `INT()`, and `FLOAT()`. For example, if you want to cast a floating point number like 3.14 as an integer, you could write `INT(3.14)`. The result would be 3, which is an integer. The casting functions are described below.

A boolean can be cast to an integer, float, or string. It cannot be cast to a date. True is 1, 1.0, or “1”, while False is 0, 0.0 or “0”. Unknown maps to Null.

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE(expression)</td>
<td>Returns a date given a number, string, or date expression. Examples:</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>DATE([Employee Start Date])</td>
<td>Returns a date given a string. Example:</td>
<td>DATE(&quot;April 15, 2004&quot;) = #April 15, 2004#</td>
</tr>
<tr>
<td>DATE(&quot;April 15, 2004&quot;)</td>
<td>Returns a date given a string. Example:</td>
<td>DATE(&quot;April 15, 2004&quot;) = #April 15, 2004#</td>
</tr>
</tbody>
</table>

Quotation marks are required in the second and third examples.

**DATETIME**

Returns a datetime given a number, string, or date expression.

Example:

| DATETIME("April 15, 2005 07:59:00") | = April 15, 2005 07:59:00 |

**DATEPARSE**

Converts a string to a datetime in the specified format. Support for some locale-specific formats is determined by the computer's system settings. Letters that appear in the data and do not need to be parsed should be surrounded by single quotes ('). For formats that do not have delimiters between values (for example, MMddyy), verify that they are parsed as expected. The format must be a constant string, not a field value. This function returns Null if the data does not match the format.

This function is available for several connectors. For more information, see Convert a Field to a Date Field on page 740.

Examples:

<p>| DATEPARSE (&quot;dd.MMMM.yyyy&quot;, &quot;15.April.2004&quot;) | = #April 15, 2004# |</p>
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATEPARSE</td>
<td>Parses a date string</td>
<td><code>DATEPARSE (&quot;h'h' m'm' s's'&quot;, &quot;10h 5m 3s&quot;) = #10:05:03#</code></td>
</tr>
<tr>
<td>FLOAT</td>
<td>Casts its argument as a floating point number.</td>
<td><code>FLOAT(3) = 3.000</code>&lt;br&gt;<code>FLOAT([Age])</code> converts every value in the Age field to a floating point number.</td>
</tr>
<tr>
<td>INT</td>
<td>Casts its argument as an integer. For expressions, this function truncates results to the closest integer toward zero.</td>
<td><code>INT(8.0/3.0) = 2</code>&lt;br&gt;<code>INT(4.0/1.5) = 2</code>&lt;br&gt;<code>INT(0.50/1.0) = 0</code>&lt;br&gt;<code>INT(-9.7) = -9</code>&lt;br&gt;When a string is converted to an integer it is first converted to a float and then rounded.</td>
</tr>
<tr>
<td>STR</td>
<td>Casts its argument as a string.</td>
<td><code>STR([Age])</code>&lt;br&gt;This expression takes all of the values in the measure called Age and converts them to strings.</td>
</tr>
</tbody>
</table>
Create a type conversion calculation

Follow along with the steps below to learn how to create a type conversion calculation.

1. In Tableau Desktop, connect to the Sample - Superstore saved data source, which comes with Tableau.
2. Navigate to a worksheet.
3. Select Analysis > Create Calculated Field.
4. In the calculation editor that opens, do the following:
   - Name the calculated field, Postal Code String.
   - Enter the following formula:
     
     ```
     STR([Postal Code])
     ```

     This calculation converts the Postal Code field from a number to a string.
   - When finished, click OK.

     The new calculated field appears under Dimensions in the Data pane. Just like your other fields, you can use it in one or more visualizations.

     Converting this field from a number to a string ensures that Tableau treats it as a string and not a number (i.e. Tableau does not aggregate it).

See Also

Data Types on page 263
Convert a Field to a Date Field on page 740
Formatting Calculations in Tableau on page 1263
Functions in Tableau on page 1275
Tableau Functions (Alphabetical) on page 1449
Tableau Functions (by Category) on page 1380
Logical Functions

This article introduces logical functions and their uses in Tableau. It also demonstrates how to create a logical calculation using an example.

In this article

Why use logical calculations below
List of logical functions in Tableau
Create a logical calculation example

Why use logical calculations

Logical calculations allow you to determine if a certain condition is true or false (boolean logic). For example, you might want to quickly see if sales for each country you distribute your merchandise to were above or below a certain threshold.

The logical calculation might look something like this:

```
SUM(Sales) > 1,000,000
```

Logical functions available in Tableau:

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
</table>
| AND      | IF <expr1> AND <expr2> THEN <then> END | Performs a logical conjunction on two expressions. Example:
```
IF (ATTR([Market]) = "Africa" AND SUM([Sales]) > [Emerging Threshold]) THEN "Well Performing"
```
| CASE     | CASE <expression> | Performs logical tests and returns appropriate values. |
The CASE function evaluates expression, compares it to a sequence of values, value1, value2, etc., and returns a result. When a value that matches expression is encountered, CASE returns the corresponding return value. If no match is found, the default return expression is used. If there is no default return and no values match, then Null is returned.

CASE is often easier to use than IIF or IF THEN ELSE.

Typically, you use an IF function to perform a sequence of arbitrary tests, and you use a CASE function to search for a match to an expression. But a CASE function can always be rewritten as an IF function, although the CASE function will generally be more concise.

Many times you can use a group to get the same results as a complicated case function.

Examples:

```
CASE [Region] WHEN 'West' THEN 1
    WHEN 'East' THEN 2 ELSE 3 END
```

```
CASE LEFT(DATENAME('weekday', [Order Date]),3) WHEN 'Sun' THEN 0
    WHEN 'Mon' THEN 1 WHEN 'Tue' THEN 2 WHEN 'Wed' THEN 3 WHEN 'Thu'
    THEN 4 WHEN 'Fri' THEN 5 WHEN 'Sat' THEN 6 END
```

ELSE
THEN <then>
ELSE <else>
END

Tests a series of expressions returning the <then> value for the first true <expr>.

Example:
<table>
<thead>
<tr>
<th>ELSEIF</th>
<th>IF &lt;expr&gt; THEN &lt;then&gt; [ELSEIF &lt;expr2&gt; THEN &lt;then2&gt;...] [ELSE &lt;else&gt;] END</th>
<th>Tests a series of expressions returning the &lt;then&gt; value for the first true &lt;expr&gt;. Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td>END</td>
<td>IF &lt;expr&gt; THEN &lt;then&gt; [ELSEIF &lt;expr2&gt; THEN &lt;then2&gt;...] [ELSE &lt;else&gt;] END</td>
<td>Tests a series of expressions returning the &lt;then&gt; value for the first true &lt;expr&gt;. Must be placed at the end of an expression. Example:</td>
</tr>
<tr>
<td>IF</td>
<td>IF &lt;expr&gt; THEN &lt;then&gt; [ELSEIF &lt;expr2&gt; THEN &lt;then2&gt;...] [ELSE &lt;else&gt;] END</td>
<td>Tests a series of expressions returning the &lt;then&gt; value for the first true &lt;expr&gt;. Example:</td>
</tr>
<tr>
<td>IFNULL</td>
<td>IFNULL(expr1, expr2)</td>
<td>Returns &lt;expr1&gt; if it is not null, otherwise returns &lt;expr2&gt;. Example:</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **IIF**  | Checks whether a condition is met, and returns one value if TRUE, another value if FALSE, and an optional third value or NULL if unknown.  
**Example:**  
IIF([Profit] > 0, 'Profit', 'Loss') | | |
| **ISDATE** | Returns true if a given string is a valid date.  
**Example:**  
ISDATE("2004-04-15") = True | | |
| **ISNULL** | Returns true if the expression does not contain valid data (Null).  
**Example:**  
ISNULL([Profit]) | | |
| **MAX**  | Returns the maximum of a single expression across all records or the maximum of two expressions for each record.  
**Example:**  
MAX([Sales]) | | |
| **MIN**  | Returns the minimum of an expression across all records or the minimum of two expressions for each record.  
**Example:**  | | |
| IFNULL([Profit], 0) | | |
| **NOT** | IF NOT <expr> THEN <then> END | Performs logical negation on an expression.  
Example:  
\[
\text{IF NOT } [\text{Profit}] > 0 \ \text{THEN} \ "\text{Unprofitable}" \ \text{END}
\] |
| **OR** | IF <expr1> OR <expr2> THEN <then> END | Performs a logical disjunction on two expressions.  
Example:  
\[
\text{IF } [\text{Profit}] < 0 \ \text{OR} \ [\text{Profit}] = 0 \ \text{THEN} \ "\text{Needs Improvement}" \ \text{END}
\] |
| **THEN** | IF <expr> THEN <then> [ELSEIF <expr2> THEN <then2>...] [ELSE <else>] END | Tests a series of expressions returning the <then> value for the first true <expr>.  
Example:  
\[
\text{IF } [\text{Profit}] > 0 \ \text{THEN} \ '\text{Profitable}' \ \text{ELSEIF} \ [\text{Profit}] = 0 \ \text{THEN} \ '\text{Break even}' \ \text{ELSE} \ '\text{unprofitable}' \ \text{END}
\] |
| **WHEN** | CASE <expr> WHEN <Value1> THEN <return1> ... [ELSE <else>] END | Finds the first <value> that matches <expr> and returns the corresponding <return>.  
Example:  
\[
\text{CASE} \ [\text{RomanNumbereral}] \ \text{WHEN} \ 'I' \ \text{THEN} \ 1 \ \text{WHEN} \ 'II' \ \text{THEN} \ 2 \ \text{ELSE} \ 3 \ \text{END}
\] |
| **ZN** | ZN(expression) | Returns <expression> if it is not null, otherwise returns zero.  
Example: |
Create a logical calculation

Follow along with the steps below to learn how to create a logical calculation.

1. In Tableau Desktop, connect to the **Sample - Superstore** saved data source, which comes with Tableau.

2. Navigate to a worksheet.

3. From the **Data** pane, drag **State** to the **Rows** shelf.

4. From the **Data** pane, drag **Category** to the **Rows** shelf and place it to the right of **State**.

5. From the **Data** pane, drag **Sales** to the **Columns** shelf.

6. Select **Analysis > Create Calculated Field**.

7. In the calculation editor that opens, do the following:
   - Name the calculated field, **KPI**.
   - Enter the following formula:
     
     ```
     SUM([Profit]) > 0
     ```

     This calculation quickly checks if a member is greater than zero. If so, it returns true; if not, it returns false.

    - When finished, click **OK**.
      
      The new calculated field appears under **Measures** in the Data pane. Just like your other fields, you can use it in one or more visualizations.

8. From the **Data** pane, drag **KPI** to **Color** on the Marks card.
You can now see which categories are losing money in each state.

See Also

**Visualize Key Progress Indicators** on page 934

Logic Calculations free video tutorial

**Functions in Tableau** on page 1275

**Tableau Functions (by Category)** on page 1380

**Tableau Functions (Alphabetical)** on page 1449

Aggregate Functions in Tableau

This article introduces aggregate functions and their uses in Tableau. It also demonstrates how to create an aggregate calculation using an example.
In this article

Why use aggregate functions below
Aggregate functions available in Tableau below
Create an aggregate calculation on page 1328
Rules for aggregate calculations on page 1330

Why use aggregate functions

Aggregate functions allow you to summarize or change the granularity of your data.

For example, you might want to know exactly how many orders your store had for a particular year. You can use the COUNTD function to summarize the exact number of orders your company had, and then break the visualization down by year.

The calculation might look something like this:

```
COUNTD(Order ID)
```

The visualization might look something like this:

![Visualization of COUNTD(Order ID)](image)

Aggregate functions available in Tableau

**Aggregations and floating-point arithmetic:** The results of some aggregations may not always be exactly as expected. For example, you may find that the Sum function returns a value such as -1.42e-14 for a column of numbers that you know should sum to exactly 0. This happens because the Institute of Electrical and Electronics Engineers (IEEE) 754 floating-point standard requires that numbers be stored in binary format, which means that numbers are sometimes rounded at extremely fine levels of precision. You can eliminate this potential
distraction by using the ROUND function (see **Number Functions** on page 1275) or by formatting the number to show fewer decimal places.

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTR</td>
<td><code>ATTR(expression)</code></td>
<td>Returns the value of the expression if it has a single value for all rows. Otherwise returns an asterisk. Null values are ignored.</td>
</tr>
<tr>
<td>AVG</td>
<td><code>AVG(expression)</code></td>
<td>Returns the average of all the values in the expression. AVG can be used with numeric fields only. Null values are ignored.</td>
</tr>
</tbody>
</table>
| COLLECT  | `COLLECT(spatial)` | An aggregate calculation that combines the values in the argument field. Null values are ignored.  
**Note:** The COLLECT function can only be used with spatial fields.  
Example:  
```
COLLECT([Geometry])
``` |
| CORR     | `CORR(expression1, expression2)` | Returns the Pearson correlation coefficient of two expressions.  
The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and -1 is an exact negative relationship.  
CORR is available with the following data sources:  
- Tableau data extracts (you can create an extract from any data source)  
- Cloudera Hive  
- EXASolution |
Firebird (version 3.0 and later)
Google BigQuery
Hortonworks Hadoop Hive
Oracle
PostgreSQL
Presto
SybaseIQ
Teradata
Vertica

For other data sources, consider either extracting the data or using WINDOW_CORR. See Table Calculation Functions on page 1347.

**Note:** The square of a CORR result is equivalent to the R-Squared value for a linear trend line model. See Trend Line Model Terms on page 1671.

Example:
You can use CORR to visualize correlation in a disaggregated scatter plot. The way to do this is to use a tableScoped level of detail expression. For example:

```
{CORR(Sales, Profit)}
```

With a level of detail expression, the correlation is run over all rows. If you used a formula like CORR(Sales, Profit) (without the surrounding brackets to make it a level of detail expression), the view would show the correlation of each individual point in the scatter plot with each other point, which is undefined.

See Table-Scoped on page 1580

<p>| COUNT | COUNT (expression) | Returns the number of items in a group. Null values are not counted. |</p>
<table>
<thead>
<tr>
<th>COUNTD</th>
<th>COUNTD (expression)</th>
<th>Returns the number of distinct items in a group. Null values are not counted. This function is not available in the following cases: workbooks created before Tableau Desktop 8.2 that use Microsoft Excel or text file data sources, workbooks that use the legacy connection, and workbooks that use Microsoft Access data sources. Extract your data into an extract file to use this function. See Extract Your Data on page 773.</th>
</tr>
</thead>
</table>
| COVAR | COVAR (expression1, expression2) | Returns the *sample covariance* of two expressions. Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when larger values of one variable tend to correspond to larger values of the other variable, on average. Sample covariance uses the number of non-null data points n - 1 to normalize the covariance calculation, rather than n, which is used by the population covariance (available with the COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population. COVAR is available with the following data sources:  
  - Tableau data extracts (you can create an extract from any data source)  
  - Cloudera Hive  
  - EXASolution  
  - Firebird (version 3.0 and later)  
  - Google BigQuery  
  - Hortonworks Hadoop Hive  
  - IBM PDA (Netezza)  
  - Oracle  
  - PostgreSQL  
  - Presto  
  - SybaseIQ  
  - Teradata |
For other data sources, consider either extracting the data or using WINDOW_COVAR. See Table Calculation Functions on page 1347.

If expression1 and expression2 are the same—for example, COVAR([profit], [profit])—COVAR returns a value that indicates how widely values are distributed.

**Note:** The value of COVAR(X, X) is equivalent to the value of VAR(X) and also to the value of STDEV(X)^2.

**Example:**
The following formula returns the sample covariance of Sales and Profit.

```
COVAR([Sales], [Profit])
```

**COVAR**

<table>
<thead>
<tr>
<th>expression1, expression2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the <em>population covariance</em> of two expressions.</td>
</tr>
<tr>
<td>Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when larger values of one variable tend to correspond to larger values of the other variable, on average. Population covariance is sample covariance multiplied by (n-1)/n, where n is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the COVAR function) is appropriate.</td>
</tr>
</tbody>
</table>

COVARP is available with the following data sources:

- Tableau data extracts (you can create an extract
from any data source)

- Cloudera Hive
- EXASolution
- Firebird (version 3.0 and later)
- Google BigQuery
- Hortonworks Hadoop Hive
- IBM PDA (Netezza)
- Oracle
- PostgreSQL
- Presto
- SybaseIQ
- Teradata
- Vertica

For other data sources, consider either extracting the data or using WINDOW_COVARP. See Table Calculation Functions on page 1347.

If expression1 and expression2 are the same—for example, COVARP([profit], [profit])—COVARP returns a value that indicates how widely values are distributed.

**Note:** The value of COVARP(X, X) is equivalent to the value of VARP(X) and also to the value of STDEVP(X)^2.

Example:
The following formula returns the population covariance of **Sales** and **Profit**.

```
COVARP([Sales], [Profit])
```

<table>
<thead>
<tr>
<th>MAX</th>
<th>MAX (expression)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the maximum of an expression across all records. If the expression is a string value, this function returns the last value where last is defined by alphabetical order.</td>
</tr>
</tbody>
</table>
| MEDIAN | MEDIAN (expression) | Returns the median of an expression across all records. Median can only be used with numeric fields. Null values are ignored. This function is not available for workbooks created before Tableau Desktop 8.2 or that use legacy connections. It is also not available for connections using any of the following data sources:

- Access
- Amazon Redshift
- Cloudera Hadoop
- HP Vertica
- IBM DB2
- IBM PDA (Netezza)
- Microsoft SQL Server
- MySQL
- SAP HANA
- Teradata

For other data source types, you can extract your data into an extract file to use this function. See Extract Your Data on page 773. |
| MIN | MIN (expression) | Returns the minimum of an expression across all records. If the expression is a string value, this function returns the first value where first is defined by alphabetical order. |
| PERCENTILE | PERCENTILE (expression, number) | Returns the percentile value from the given expression corresponding to the specified number. The number must be between 0 and 1 (inclusive)—for example, 0.66, and must be a numeric constant.

This function is available for the following data sources:

- Non-legacy Microsoft Excel and Text File connections.
- Extracts and extract-only data source types (for example, Google Analytics, OData, or Salesforce).
- Sybase IQ 15.1 and later data sources. |
For other data source types, you can extract your data into an extract file to use this function. See Extract Your Data on page 773.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STDEV</td>
<td>Returns the statistical standard deviation of all values in the given expression based on a sample of the population.</td>
</tr>
<tr>
<td>STDEVP</td>
<td>Returns the statistical standard deviation of all values in the given expression based on a biased population.</td>
</tr>
<tr>
<td>SUM</td>
<td>Returns the sum of all values in the expression. SUM can be used with numeric fields only. Null values are ignored.</td>
</tr>
<tr>
<td>VAR</td>
<td>Returns the statistical variance of all values in the given expression based on a sample of the population.</td>
</tr>
<tr>
<td>VARP</td>
<td>Returns the statistical variance of all values in the given expression on the entire population.</td>
</tr>
</tbody>
</table>

Create an aggregate calculation

Follow along with the steps below to learn how to create an aggregate calculation.

1. In Tableau Desktop, connect to the Sample - Superstore saved data source, which comes with Tableau.

2. Navigate to a worksheet and select Analysis > Create Calculated Field.

3. In the calculation editor that opens, do the following:
   - Name the calculated field Margin.
   - Enter the following formula:
IIF(SUM([Sales]) !=0, SUM([Profit])/SUM([Sales]), 0)

**Note:** You can use the function reference to find and add aggregate functions and other functions (like the logical IIF function in this example) to the calculation formula. For more information, see *Use the functions reference in the calculation editor* on page 1615.

- When finished, click **OK**.

The new aggregate calculation appears under **Measures** in the **Data** pane. Just like your other fields, you can use it in one or more visualizations.

**Note:** Aggregation calculations are always measures.

When **Margin** is placed on a shelf or card in the worksheet, its name is changed to **AGG (Margin)**, which indicates that it is an aggregate calculation and cannot be aggregated any further.
Rules for aggregate calculations

The rules that apply to aggregate calculations are as follows:

- For any aggregate calculation, you cannot combine an aggregated value and a disaggregated value. For example, \( \text{SUM(Price)} \times \text{[Items]} \) is not a valid expression because \( \text{SUM(Price)} \) is aggregated and \( \text{Items} \) is not. However, \( \text{SUM(Price} \times \text{Items}) \) and \( \text{SUM(Price)} \times \text{SUM(Items)} \) are both valid.

- Constant terms in an expression act as aggregated or disaggregated values as appropriate. For example: \( \text{SUM(Price} \times 7) \) and \( \text{SUM(Price)} \times 7 \) are both valid expressions.

- All of the functions can be evaluated on aggregated values. However, the arguments to any given function must either all be aggregated or all disaggregated. For example: \( \text{MAX(SUM(Sales),Profit)} \) is not a valid expression because Sales is aggregated and Profit is not. However, \( \text{MAX(SUM(Sales),SUM(Profit))} \) is a valid expression.

- The result of an aggregate calculation is always a measure.
Like predefined aggregations, aggregate calculations are computed correctly for grand totals. Refer to Grand Totals for more information.

See Also

Understanding Calculations: Aggregate Calculations

Data Aggregation in Tableau on page 279

Functions in Tableau on page 1275

Tableau Functions (by Category) on page 1380

Tableau Functions (Alphabetical) on page 1449

Pass-Through Functions (RAWSQL)

These RAWSQL pass-through functions can be used to send SQL expressions directly to the database, without first being interpreted by Tableau. If you have custom database functions that Tableau doesn’t know about, you can use these pass-through functions to call these custom functions.

Your database usually will not understand the field names that are shown in Tableau. Because Tableau does not interpret the SQL expressions you include in the pass-through functions, using the Tableau field names in your expression may cause errors. You can use a substitution syntax to insert the correct field name or expression for a Tableau calculation into pass-through SQL. For example, if you had a function that computed the median of a set of values, you could call that function on the Tableau column [Sales] like this:

```sql
RAWSQLLAGG_REAL("MEDIAN('%1')", [Sales])
```

Because Tableau does not interpret the expression, you must define the aggregation. You can use the RAWSQLLAGG functions described below when you are using aggregated expressions.

RAWSQL pass-through functions will not work with published data sources.

These functions may return different results starting in Tableau Desktop 8.2 than they did in earlier versions of Tableau Desktop. This is because Tableau now uses ODBC for pass-through functions instead of OLE DB. ODBC truncates when returning real values as integer; OLE DB rounds when returning real values as integer.
RAWSQL Functions

The following RAWSQL functions are available in Tableau.

RAWSQL_BOOL("sql_expr", [arg1], ...[argN])

Returns a Boolean result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In the example, %1 is equal to [Sales] and %2 is equal to [Profit].

RAWSQL_BOOL("IIF( %1 > %2, True, False)", [Sales], [Profit])

RAWSQL_DATE("sql_expr", [arg1], ...[argN])

Returns a Date result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In this example, %1 is equal to [Order Date].

RAWSQL_DATE("%1", [Order Date])

RAWSQL_DATETIME("sql_expr", [arg1], ...[argN])

Returns a Date and Time result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].

Example

RAWSQL_DATETIME("MIN(%1)", [Delivery Date])
RAWSQL_INT(“sql_expr”, [arg1], ...[argN])

Returns an integer result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example

RAWSQL_INT(“500 + %1”, [Sales])

RAWSQL_REAL(“sql_expr”, [arg1], ...[argN])

Returns a numeric result from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]

Example

RAWSQL_REAL(“-123.98 * %1”, [Sales])

RAWSQL_SPATIAL

Returns a Spatial from a given SQL expression that is passed directly to the underlying data source. Use %n in the SQL expression as a substitution syntax for database values.

Example

In this example, %1 is equal to [Geometry].

RAWSQL_SPATIAL(“%1”, [Geometry])

RAWSQL_STR(“sql_expr”, [arg1], ...[argN])

Returns a string from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Customer Name].

Example

RAWSQL_STR(“%1”, [Customer Name])
RAWSQLAGG_BOOL("sql_expr", [arg1], ...[argN])

Returns a Boolean result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In the example, %1 is equal to [Sales] and %2 is equal to [Profit].
RAWSQLAGG_BOOL("SUM( %1) >SUM( %2)", [Sales], [Profit])

RAWSQLAGG_DATE("sql_expr", [arg1], ...[argN])

Returns a Date result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Order Date].

Example

RAWSQLAGG_DATE("MAX( %1)", [Order Date])

RAWSQLAGG_DATETIME("sql_expr", [arg1], ...[argN])

Returns a Date and Time result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].

Example

RAWSQLAGG_DATETIME("MIN( %1)", [Delivery Date])

RAWSQLAGG_INT("sql_expr", [arg1], ...[argN])

Returns an integer result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example

RAWSQLAGG_INT("500 + SUM( %1)", [Sales])
RAWSQLAGG_REAL(“sql_expr”, [arg1,] ...[argN])

Returns a numeric result from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]

Example

RAWSQLAGG_REAL(“SUM( %1)”, [Sales])

RAWSQLAGG_STR(“sql_expr”, [arg1,] ...[argN])

Returns a string from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Discount].

Example

RAWSQLAGG_STR(“AVG(%1)”, [Discount])

User Functions

This article introduces user functions and their uses in Tableau. It also demonstrates how to create a user calculation using an example.

In this article

Why use user functions below
List of user functions in Tableau
Create a user calculation example

Why use user functions

User functions can be used to create user filters or column-level security filters that affect visualizations published to Tableau Server or Tableau Online, so that only certain people can see your visualization.
For example, if you have a visualization that shows the sales performance for each employee in your department published on Tableau Server or Tableau Online, you might want to only allow employees to see their own sales numbers when they access that visualization.

In this case, you can use the ISMEMBEROF function to create a field that returns true if the username of the person signed in to the server is a member of a specified group (on the server), such as the "Managers" group, for example. Then when you filter the view using this calculated field, only a person who is part of that group can see the data.

The calculation in this case might look something like the following:

```
ISMEMBEROF('Managers')
```

**Note:** If your group or user names contain non-alphanumeric characters, you must use HTML URL Encoding for the special characters when using the functions below. For example, the function `ISMEMBEROF("USERS+")` needs to be written as `ISMEMBEROF("USERS%2B")`, because the '%2B' is the HTML URL Encoding for the '+' symbol. For information on HTML URL encoding, please see [HTML URL Encoding Reference](https://www.w3schools.com/tags/att_url_encoded.asp) at the W3schools Web Developer site.

### User functions available in Tableau:

<table>
<thead>
<tr>
<th>Functions</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FULLNAME</strong></td>
<td><code>FULLNAME()</code></td>
<td>Returns the full name for the current user. This is the Tableau Server or Tableau Online full name when the user is signed in; otherwise the local or network full name for the Tableau Desktop user.</td>
</tr>
</tbody>
</table>

Example:

```
[Manager]=FULLNAME()
```

If manager Dave Hallsten is signed in, this example returns True only if the Manager field in the view contained Dave Hallsten. When used as a filter, this calculated field can be
used to create a user filter that only shows data that is relevant to the person signed in to the server.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISFULLNAME</td>
<td>Returns true if the current user’s full name matches the specified full name, or false if it does not match. This function uses the Tableau Server or Online full name when the user is signed in; otherwise it uses the local or network full name for the Tableau Desktop user.</td>
<td><code>ISFULLNAME(&quot;Dave Hallsten&quot;)</code>&lt;br&gt;This example returns true if Dave Hallsten is the current user, otherwise it returns false.</td>
</tr>
<tr>
<td>ISMEMBEROF</td>
<td>Returns true if the person currently using Tableau is a member of a group that matches the given string. If the person currently using Tableau is signed in, the group membership is determined by groups on Tableau Server or Tableau Online. If the person is not signed in, this function returns false.</td>
<td><code>IF ISMEMBEROF(&quot;Sales&quot;) THEN &quot;Sales&quot; ELSE &quot;Other&quot; END</code>&lt;br&gt;Example:</td>
</tr>
<tr>
<td>ISUSERNAME</td>
<td>Returns true if the current user’s username matches the specified username, or false if it does not match. This function uses the Tableau Server or Online username when the user is signed in; otherwise it uses the local or network username for the Tableau Desktop user.</td>
<td><code>ISUSERNAME(&quot;dhallsten&quot;)</code>&lt;br&gt;Example:</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td></td>
</tr>
</tbody>
</table>
| **USERNAME** | Returns the username for the current user. This is the Tableau Server or Tableau Online username when the user is signed in; otherwise it is the local or network username for the Tableau Desktop user. Example: 

```
[Manager]=USERNAME() 
```

If the manager dhallsten was signed in, this function would only return True when the Manager field in the view is dhallsten. When used as a filter this calculated field can be used to create a user filter that only shows data that is relevant to the person signed in to the server. |
| **USERDOMAIN** | Returns the domain for the current user when the user is signed on to Tableau Server. Returns the Windows domain if the Tableau Desktop user is on a domain. Otherwise this function returns a null string. Example: 

```
[Manager]=USERNAME() AND [Domain]=USERDOMAIN() 
```

This example returns true if dhallsten is the current user; otherwise it returns false. |

---

**Create a user calculation**

User calculations work directly with the users and groups you have set up on Tableau Server or Tableau Online. You can create user calculations to use as filters so users only see the data that is relevant to them.

For example, if you have a map viz similar to the following, which shows sales data for 48 US states, you can create a user calculation to show only parts of the map that are relevant to each user, such as data relevant for a regional manager versus a national manager. (A national
manager should be able to see data for the entire country, while a regional manager should only be able to see data for the region they manage).

When the national manager is signed in, they see the following visualization:

When the western regional manager is signed in, they see only sales for their region:
To create a user function that performs similarly to this example, follow the steps below.

Before you begin

To follow along with this example you must have access to Tableau Server or Online. You must also be a Server or Site Administrator.

Step 1: Create the users and groups

1. Sign in to Tableau Server or Tableau Online.
2. In Tableau Server or Tableau Online, add the following users:
   - Kelly Williams
   - Chuck Magee
   - Cassandra Brandow
   - Anna Andreadi
   For more information, see Add Users to a Site in the Tableau Server Help.
3. Create a new group called National Managers.
   For more information, see Create a Local Group in the Tableau Server Help.
4. Add yourself to the National Managers group.
   For more information, see Add Users to a Group in the Tableau Server Help.

Step 2: Create the visualization

1. Open Tableau Desktop, and connect to the Sample-Superstore data source, which comes with Tableau.
2. In the bottom left corner of the workspace, click the Data Source tab.
3. On the Data Source page, from the Connections pane on the left, drag the People sheet to the join area.
4. Click the join icon and select Left.
5. Navigate to a new worksheet.

6. In the **Data** pane, under Dimensions, double-click **State**. A map view is created.

7. From the **Data** pane, under Measures, drag **Sales** to **Color** on the Marks card.

8. On the Columns shelf, select the Longitude field and hold down Control (Command on Mac) on your keyboard to copy it. Drag the copy to the right of the original on the Columns shelf.
9. On the Marks card, click the second (bottom) **Longitude** tab.

10. From the **Data** pane, drag **Region** to **Color** on the Marks card.

    The map view on the right updates with new colors.

11. On the Marks card, click the mark type drop-down and select **Map**.
12. On the Marks card, click **Color**, and, under **Opacity**, adjust the slider to **50%**.

13. On the Marks card, click the first **Longitude** tab.

14. On the Marks card, click **Color > Edit Colors**, and then select **Gray** from the color palette drop-down list.

   The map view on the left updates.
15. On the Columns shelf, right-click the **Longitude** field on the right and select **Dual Axis**.

The map looks like the following:
Step 3: Create the User Calculation

1. Select Analysis > Create Calculated Field.

2. In the calculation editor that opens, do the following:

   - Name the calculated field, User Filter.
   - Enter the following formula:

     \[
     \text{[Region (People)]} = \text{USERNAME()} \text{ OR ISMEMBEROF ("National Managers")}
     \]

     This calculation checks if a person is included in the Region (People) field, or if a person is included in the National Managers group. If so, it returns true.

   - When finished, click OK.

The new user calculation appears under Dimensions in the Data pane. Just like your other fields, you can use it in one or more visualizations.
Step 4: Add the user calculation to the Filters shelf

1. From the Data pane, under Dimensions, drag User Filter to the Filters shelf.

2. In the Filter dialog box that opens, select True, and then click OK.

   **Note:** If you are not signed in to Tableau Server or Tableau Online, the True option is not visible. In Tableau Desktop, sign in to Tableau Server or Tableau Online to select it. See Sign in to Tableau Server or Online on page 310 for more information.

Step 5: Test the calculation

1. In Tableau Desktop, in the bottom-right corner of the workspace, click the Filter as User drop-down and change the user to Anna Andreadi.
   
   The map updates to show only the West region of the United States because Anna is assigned to the West region in the People sheet.

2. Select the Filter as User drop-down again and change the user to Kelly Williams.
   
   The map updates to show only the Central region of the United States because Kelly is assigned to the Central region in the People sheet.

3. Select the Filter as User drop-down again and change the user to Chuck Magee.
   
   The map updates to show only the East region of the United States because Chuck is assigned to the East region in the People sheet.

4. Select the Filter as User drop-down again and change the user to Cassandra Brandow.
   
   The map updates to show only the South region of the United States because Cassandra is assigned to the South region in the People sheet.

5. Select the Filter as User drop-down one more time and change the user back to yourself.
   
   The map updates to show all data because you are a part of the National Managers group on the server.

This behavior persists when you publish the view to Tableau Server or Tableau Online. Users not listed in the National Managers group, or in the People sheet in the Sample Superstore data source see only a blank visualization.
See Also

User Calculations free training video
Functions in Tableau on page 1275
Tableau Functions (by Category) on page 1380
Tableau Functions (Alphabetical) on page 1449

Table Calculation Functions

This article introduces table calculation functions and their uses in Tableau. It also demonstrates how to create a table calculation using the calculation editor.

In this article

Why use table calculation functions below
Table calculation functions available in Tableau below
Create a table calculation using the calculation editor on page 1368

Why use table calculation functions

Table calculation functions allow you to perform computations on values in a table.
For example, you can calculate the percent of total an individual sale is for the year, or for several years.

Table calculation functions available in Tableau

FIRST( )

Returns the number of rows from the current row to the first row in the partition. For example, the view below shows quarterly sales. When FIRST() is computed within the Date partition, the offset of the first row from the second row is -1.
Example

When the current row index is 3, \( \text{FIRST}() = -2 \).

\[ \text{INDEX()} \]

Returns the index of the current row in the partition, without any sorting with regard to value. The first row index starts at 1. For example, the table below shows quarterly sales. When \( \text{INDEX}() \) is computed within the Date partition, the index of each row is 1, 2, 3, 4..., etc.

Example

For the third row in the partition, \( \text{INDEX()} = 3 \).

\[ \text{LAST()} \]

Returns the number of rows from the current row to the last row in the partition. For example, the table below shows quarterly sales. When \( \text{LAST}() \) is computed within the Date partition, the offset of the last row from the second row is 5.
Example

When the current row index is 3 of 7, \( \text{LAST}() = 4 \).

**LOOKUP(expression, [offset])**

Returns the value of the expression in a target row, specified as a relative offset from the current row. Use `FIRST() + n` and `LAST() - n` as part of your offset definition for a target relative to the first/last rows in the partition. If `offset` is omitted, the row to compare to can be set on the field menu. This function returns NULL if the target row cannot be determined.

The view below shows quarterly sales. When `LOOKUP (SUM(Sales), 2)` is computed within the Date partition, each row shows the sales value from 2 quarters into the future.
Example

LOOKUP(SUM([Profit]), FIRST()+2) computes the SUM(Profit) in the third row of the partition.

PREVIOUS_VALUE(expression)

Returns the value of this calculation in the previous row. Returns the given expression if the current row is the first row of the partition.

Example

SUM([Profit]) * PREVIOUS_VALUE(1) computes the running product of SUM(Profit).

RANK(expression, ['asc' | 'desc'])
Returns the standard competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.

Example

The following image shows the effect of the various ranking functions (RANK, RANK_DENSE, RANK_MODIFIED, RANK_PERCENTILE, and RANK_UNIQUE) on a set of values. The data set contains information on 14 students (StudentA through StudentN); the Age column shows the current age of each student (all students are between 17 and 20 years of age). The remaining columns show the effect of each rank function on the set of age values, always assuming the default order (ascending or descending) for the function.

![](image)

RANK_DENSE(expression, ['asc' | 'desc'])

Returns the dense rank for the current row in the partition. Identical values are assigned an identical rank, but no gaps are inserted into the number sequence. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (3, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.
### RANK_MODIFIED(expression, ['asc' | 'desc'])

Returns the modified competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 3, 3, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.

### RANK_PERCENTILE(expression, ['asc' | 'desc'])

Returns the percentile rank for the current row in the partition. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is ascending.

With this function, the set of values (6, 9, 9, 14) would be ranked (0.25, 0.75, 0.75, 1.00).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.

### RANK_UNIQUE(expression, ['asc' | 'desc'])

Returns the unique rank for the current row in the partition. Identical values are assigned different ranks. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 3, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.
RUNNING_AVG(expression)

Returns the running average of the given expression, from the first row in the partition to the current row.

The view below shows quarterly sales. When `RUNNING_AVG(SUM([Sales]))` is computed within the Date partition, the result is a running average of the sales values for each quarter.

Example

`RUNNING_AVG(SUM([Profit]))` computes the running average of SUM(Profit).

RUNNING_COUNT(expression)

Returns the running count of the given expression, from the first row in the partition to the current row.

Example

`RUNNING_COUNT(SUM([Profit]))` computes the running count of SUM(Profit).

RUNNING_MAX(expression)
Returns the running maximum of the given expression, from the first row in the partition to the current row.

<table>
<thead>
<tr>
<th>Year of Order Date</th>
<th>Quarter of Order Date</th>
<th>Central</th>
<th>East</th>
<th>South</th>
<th>West</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>Q1</td>
<td>$180.877</td>
<td>$231.411</td>
<td>$133.934</td>
<td>$186.961</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$197.213</td>
<td>$204.914</td>
<td>$337.813</td>
<td>$213.507</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$302.679</td>
<td>$166.201</td>
<td>$283.806</td>
<td>$206.612</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$297.208</td>
<td>$226.903</td>
<td>$214.645</td>
<td>$293.291</td>
</tr>
<tr>
<td>2010</td>
<td>Q1</td>
<td>$180.509</td>
<td>$180.123</td>
<td>$273.543</td>
<td>$251.145</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$195.796</td>
<td>$224.062</td>
<td>$261.291</td>
<td>$195.876</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$116.613</td>
<td>$60.063</td>
<td>$184.601</td>
<td>$102.751</td>
</tr>
</tbody>
</table>

RUNNING_MAX(SUM([Profit])) computes the running maximum of SUM(Profit).

RUNNING_MIN(expression)

Returns the running minimum of the given expression, from the first row in the partition to the current row.
Example

$\text{RUNNING\_MIN\left(\text{SUM([Profit]}\right))}$ computes the running minimum of $\text{SUM(Profit)}$.

**RUNNING\_SUM(expression)**

Returns the running sum of the given expression, from the first row in the partition to the current row.
Example

`RUNNING_SUM(SUM([Profit]))` computes the running sum of `SUM(Profit)`

**SIZE()**

Returns the number of rows in the partition. For example, the view below shows quarterly sales. Within the Date partition, there are seven rows so the `SIZE()` of the Date partition is 7.

![Quarterly Sales Table](image)

Example

`SIZE() = 5` when the current partition contains five rows.

**SCRIPT_BOOL**

Returns a Boolean result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use `.argn` (with a leading period) to reference parameters (`.arg1, .arg2, etc.`).

In Python expressions, use `_argn` (with a leading underscore).

Examples

In this R example, `.arg1` is equal to `SUM([Profit])`:

```r
SCRIPT_BOOL("is.finite(.arg1)", SUM([Profit]))
```

The next example returns True for store IDs in Washington state, and False otherwise. This example could be the definition for a calculated field titled `IsStoreInWA`.

```r
SCRIPT_BOOL('grepl(".*_WA", .arg1, perl=TRUE)', ATTR([Store ID]))
```

A command for Python would take this form:
SCRIPT_BOOL("return map(lambda x : x > 0, _arg1)", SUM([Profit]))

SCRIPT_INT

Returns an integer result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use .argn (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use _argn (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

SCRIPT_INT("is.finite(.arg1)", SUM([Profit]))

In the next example, k-means clustering is used to create three clusters:

SCRIPT_INT('result <- kmeans(data.frame(.arg1,.arg2,.arg3,.arg4), 3);result$cluster;', SUM([Petal length]), SUM([Petal width]),SUM ([Sepal length]),SUM([Sepal width]))

A command for Python would take this form:

SCRIPT_INT("return map(lambda x : int(x * 5), _arg1)", SUM ([Profit]))

SCRIPT_REAL

Returns a real result from the specified expression. The expression is passed directly to a running external service instance. In

R expressions, use .argn (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use _argn (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

SCRIPT_REAL("is.finite(.arg1)", SUM([Profit]))
The next example converts temperature values from Celsius to Fahrenheit.

```r
SCRIPT_REAL('library(udunits2);ud.convert(.arg1, "celsius", "degree_fahrenheit")', AVG([Temperature]))
```

A command for Python would take this form:

```python
SCRIPT_REAL("return map(lambda x : x * 0.5, _arg1)", SUM ([Profit]))
```

**SCRIPT_STR**

Returns a string result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use `.arg n` (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use `_arg n` (with a leading underscore).

**Examples**

In this R example, `.arg1` is equal to `SUM([Profit])`:

```r
SCRIPT_STR("is.finite(.arg1)", SUM([Profit]))
```

The next example extracts a state abbreviation from a more complicated string (in the original form 13XSL_CA, A13_WA):

```r
SCRIPT_STR('gsub(".*_", "", .arg1)', ATTR([Store ID]))
```

A command for Python would take this form:

```python
SCRIPT_STR("return map(lambda x : x[:2], _arg1)", ATTR([Region]))
```

**TOTAL(expression)**

Returns the total for the given expression in a table calculation partition.

**Example**

Assume you are starting with this view:
You open the calculation editor and create a new field which you name **Totality**:

```
TOTAL(SUM([Sales]))
```

You then drop **Totality** on Text, to replace `SUM(Sales)`. Your view changes such that it sums values based on the default **Compute Using** value:
This raises the question, What is the default Compute Using value? If you right-click (Control-click on a Mac) Totality in the Data pane and choose Edit, there is now an additional bit of information available:

The default Compute Using value is Table (Across). The result is that Totality is summing the values across each row of your table. Thus, the value that you see across each row is the sum of the values from the original version of the table.

The values in the 2011/Q1 row in the original table were $8601, $6579, $44262, and $15006. The values in the table after Totality replaces SUM(Sales) are all $74,448, which is the sum of the four original values.

Notice the triangle next to Totality after you drop it on Text:
This indicates that this field is using a table calculation. You can right-click the field and choose **Edit Table Calculation** to redirect your function to a different **Compute Using** value. For example, you could set it to **Table (Down)**. In that case, your table would look like this:

```
WINDOW_AVG(expression, [start, end])
```

Returns the average of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.
For example, the view below shows quarterly sales. A window average within the Date partition returns the average sales across all dates.

Example

\[ \text{WINDOW\_AVG(SUM([Profit]), FIRST()+1, 0)} \]
computes the average of SUM(Profit) from the second row to the current row.

\[ \text{WINDOW\_CORR(expression1, expression2, [start, end])} \]

Returns the Pearson correlation coefficient of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()=n for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.

The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and -1 is an exact negative relationship.

There is an equivalent aggregation function: CORR. See Tableau Functions (Alphabetical) on page 1449.

Example

The following formula returns the Pearson correlation of \text{SUM(Profit)} and \text{SUM(Sales)} from the five previous rows to the current row.

\[ \text{WINDOW\_CORR(SUM[Profit]), SUM([Sales]), -5, 0)} \]
WINDOW_COUNT(expression, [start, end])

Returns the count of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

WINDOW_COUNT(SUM([Profit]), FIRST()+1, 0) computes the count of SUM(Profit) from the second row to the current row

WINDOW_COVAR(expression1, expression2, [start, end])

Returns the sample covariance of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end arguments are omitted, the window is the entire partition.

Sample covariance uses the number of non-null data points n - 1 to normalize the covariance calculation, rather than n, which is used by the population covariance (with the WINDOW_COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population.

There is an equivalent aggregation function: COVAR. See Tableau Functions (Alphabetical) on page 1449.

Example

The following formula returns the sample covariance of SUM(Profit) and SUM(Sales) from the two previous rows to the current row.

WINDOW_COVAR(SUM([Profit]), SUM([Sales]), -2, 0)

WINDOW_COVARP(expression1, expression2, [start, end])

Returns the population covariance of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.
Population covariance is sample covariance multiplied by \((n-1)/n\), where \(n\) is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the WINDOW_COVAR function) is appropriate.

There is an equivalent aggregation function: COVARP. Tableau Functions (Alphabetical) on page 1449.

**Example**

The following formula returns the population covariance of SUM(Profit) and SUM(Sales) from the two previous rows to the current row.

\[
\text{WINDOW_COVARP} \left( \text{SUM}([\text{Profit}]), \text{SUM}([\text{Sales}]), -2, 0 \right)
\]

**WINDOW_MEDIAN(expression, [start, end])**

Returns the median of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+\(n\) and \(\text{LAST}()-\(n\) for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly profit. A window median within the Date partition returns the median profit across all dates.
Example

Example

WINDOW_MEDIAN(SUM([Profit]), FIRST()+1, 0) computes the median of SUM (Profit) from the second row to the current row.

**WINDOW_MAX(expression, [start, end])**

Returns the maximum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window maximum within the Date partition returns the maximum sales across all dates.

Example

WINDOW_MAX(SUM([Sales]), FIRST(), LAST()) computes the maximum of SUM (Profit) from the second row to the current row.

**WINDOW_MIN(expression, [start, end])**

Returns the minimum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.
For example, the view below shows quarterly sales. A window minimum within the Date partition returns the minimum sales across all dates.

Example

\[
\text{WINDOW\_MIN}(\text{SUM}([\text{Profit}]), \text{FIRST}()+1, 0)
\]
computes the minimum of \(\text{SUM}(\text{Profit})\) from the second row to the current row.

**WINDOW\_PERCENTILE(expression, number, [start, end])**

Returns the value corresponding to the specified percentile within the window. The window is defined by means of offsets from the current row. Use FIRST()\(+n\) and LAST()\(-n\) for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

\[
\text{WINDOW\_PERCENTILE}(\text{SUM}([\text{Profit}]), 0.75, -2, 0)
\]
returns the 75th percentile for \(\text{SUM}(\text{Profit})\) from the two previous rows to the current row.

**WINDOW\_STDEV(expression, [start, end])**

Returns the sample standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()\(+n\) and LAST()\(-n\) for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.
Example

\texttt{WINDOW\_STDEV(SUM([Profit]), FIRST()+1, 0)} computes the standard deviation of \texttt{SUM(Profit)} from the second row to the current row.

\textbf{WINDOW\_STDEVP(expression, [start, end])}

Returns the biased standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use \texttt{FIRST()+n} and \texttt{LAST()-n} for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

\texttt{WINDOW\_STDEVP(SUM([Profit]), FIRST()+1, 0)} computes the standard deviation of \texttt{SUM(Profit)} from the second row to the current row.

\textbf{WINDOW\_SUM(expression, [start, end])}

Returns the sum of the expression within the window. The window is defined by means of offsets from the current row. Use \texttt{FIRST()+n} and \texttt{LAST()-n} for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window sum computed within the Date partition returns the summation of sales across all quarters.
Example

\textsc{WINDOW\_SUM(\sum([\text{Profit}]), \text{FIRST()}+1, 0)} computes the sum of \(\sum(\text{Profit})\) from the second row to the current row.

\textbf{WINDOW\_VAR(expression, [start, end])}

Returns the sample variance of the expression within the window. The window is defined by means of offsets from the current row. Use \text{FIRST()}+n and \text{LAST()}-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

\textsc{WINDOW\_VAR((\sum([\text{Profit}]), \text{FIRST()}+1, 0)} computes the variance of \(\sum(\text{Profit})\) from the second row to the current row.

\textbf{WINDOW\_VARP(expression, [start, end])}

Returns the biased variance of the expression within the window. The window is defined by means of offsets from the current row. Use \text{FIRST()}+n and \text{LAST()}-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

\textsc{WINDOW\_VARP(\sum([\text{Profit}]), \text{FIRST()}+1, 0)} computes the variance of \(\sum(\text{Profit})\) from the second row to the current row.

Create a table calculation using the calculation editor

Follow along with the steps below to learn how to create a table calculation using the calculation editor.

\textbf{Note:} There are several ways to create table calculations in Tableau. This example demonstrates only one of those ways. For more information, see \textit{Transform Values with Table Calculations} on page 1524.
Step 1: Create the visualization

1. In Tableau Desktop, connect to the Sample-Superstore saved data source, which comes with Tableau.
2. Navigate to a worksheet.
3. From the Data pane, under Dimensions, drag Order Date to the Columns shelf.
4. From the Data pane, under Dimensions, drag Sub-Category to the Rows shelf.
5. From the Data pane, under Measures, drag Sales to Text on the Marks card.
   Your visualization updates to a text table.

![Visualization](image)

Step 2: Create the table calculation

1. Select Analysis > Create Calculated Field.
2. In the calculation editor that opens, do the following:
• Name the calculated field, Running Sum of Profit.

• Enter the following formula:

   \[
   \text{RUNNING\_SUM} (\text{SUM}([\text{Profit}]))
   \]

   This formula calculates the running sum of profit sales. It is computed across the entire table.

• When finished, click **OK**.

   The new table calculation field appears under Measures in the Data pane. Just like your other fields, you can use it in one or more visualizations.

**Step 3: Use the table calculation in the visualization**

1. From the Data pane, under Measures, drag **Running Sum of Profit** to **Color** on the Marks card.

2. On the Marks card, click the Mark Type drop-down and select **Square**.
The visualization updates to a highlight table:

<table>
<thead>
<tr>
<th>Sub-Category</th>
<th>2014</th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories</td>
<td>$25,014</td>
<td>$40,524</td>
<td>$41,896</td>
<td>$59,945</td>
</tr>
<tr>
<td>Appliances</td>
<td>$15,314</td>
<td>$23,241</td>
<td>$26,050</td>
<td>$42,927</td>
</tr>
<tr>
<td>Art</td>
<td>$6,058</td>
<td>$6,237</td>
<td>$5,961</td>
<td>$8,863</td>
</tr>
<tr>
<td>Binders</td>
<td>$43,488</td>
<td>$37,453</td>
<td>$49,683</td>
<td>$72,788</td>
</tr>
<tr>
<td>Bookcases</td>
<td>$20,037</td>
<td>$38,544</td>
<td>$26,275</td>
<td>$30,024</td>
</tr>
<tr>
<td>Chairs</td>
<td>$77,242</td>
<td>$71,735</td>
<td>$83,919</td>
<td>$95,554</td>
</tr>
<tr>
<td>Copiers</td>
<td>$10,850</td>
<td>$26,179</td>
<td>$49,599</td>
<td>$62,899</td>
</tr>
<tr>
<td>Envelopes</td>
<td>$3,856</td>
<td>$4,512</td>
<td>$4,730</td>
<td>$3,379</td>
</tr>
<tr>
<td>Fasteners</td>
<td>$661</td>
<td>$545</td>
<td>$960</td>
<td>$858</td>
</tr>
<tr>
<td>Furnishings</td>
<td>$13,826</td>
<td>$21,090</td>
<td>$27,874</td>
<td>$26,915</td>
</tr>
<tr>
<td>Labels</td>
<td>$2,841</td>
<td>$2,956</td>
<td>$2,827</td>
<td>$3,861</td>
</tr>
<tr>
<td>Machines</td>
<td>$62,023</td>
<td>$27,764</td>
<td>$55,907</td>
<td>$43,545</td>
</tr>
<tr>
<td>Paper</td>
<td>$14,835</td>
<td>$15,288</td>
<td>$20,662</td>
<td>$27,695</td>
</tr>
<tr>
<td>Phones</td>
<td>$77,391</td>
<td>$88,394</td>
<td>$78,962</td>
<td>$105,341</td>
</tr>
<tr>
<td>Storage</td>
<td>$50,329</td>
<td>$45,048</td>
<td>$58,789</td>
<td>$69,678</td>
</tr>
<tr>
<td>Supplies</td>
<td>$14,394</td>
<td>$1,952</td>
<td>$14,278</td>
<td>$16,049</td>
</tr>
<tr>
<td>Tables</td>
<td>$46,088</td>
<td>$39,150</td>
<td>$60,833</td>
<td>$60,894</td>
</tr>
</tbody>
</table>

**Step 4: Edit the table calculation**

1. On the Marks card, right-click **Running Sum of Profit** and select **Edit Table Calculation**.

2. In the Table Calculation dialog box that opens, under Compute Using, select **Table (down)**.
The visualization updates to the following:

![Image of a Tableau worksheet]

### See Also

- [Create a table calculation](#) on page 1534
- [Table Calculation Types](#) on page 1537
- [Customize Table Calculations](#) on page 1564
- [Quick Table Calculations](#) on page 1559
- [Functions in Tableau](#) on page 1275
- [Tableau Functions (by Category)](#) on page 1380
- [Tableau Functions (Alphabetical)](#) on page 1449
Additional Functions

**REGEXP_REPLACE(string, pattern, replacement)**

Returns a copy of the given string where the regular expression pattern is replaced by the replacement string. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern and the replacement must be constants.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the [Regular Expressions](#) page in the online ICU User Guide.

**Example**

```plaintext
REGEXP_REPLACE('abc 123', '\s', '-') = 'abc-123'
```

**REGEXP_MATCH(string, pattern)**

Returns true if a substring of the specified string matches the regular expression pattern. This function is available for Text File, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Impala 2.3.0 (through Cloudera Hadoop data sources), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the [Regular Expressions](#) page in the online ICU User Guide.

**Example**

```plaintext
REGEXP_MATCH('-(?<=\[1234\].[The.Market])\-', '\s*(\w*\.)(\w*\s*)') = true
```
REGEXP_EXTRACT(string, pattern)

Returns the portion of the string that matches the regular expression pattern. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source’s documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_EXTRACT('abc 123', '[a-z]+s+(\d+)') = '123'

REGEXP_EXTRACT_NTH(string, pattern, index)

Returns the portion of the string that matches the regular expression pattern. The substring is matched to the nth capturing group, where n is the given index. If index is 0, the entire string is returned. This function is available for Text File, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source’s documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_EXTRACT_NTH('abc 123', '((a-z)+\s+\d+)', 2) = '123'
Hadoop Hive Specific Functions

**Note**: Only the PARSE_URL and PARSE_URL_QUERY functions are available for Cloudera Impala data sources.

**GET_JSON_OBJECT**(JSON string, JSON path)
Returns the JSON object within the JSON string based on the JSON path.

**PARSE_URL**(string, url_part)
Returns a component of the given URL string where the component is defined by url_part. Valid url_part values include: 'HOST', 'PATH', 'QUERY', 'REF', 'PROTOCOL', 'AUTHORITY', 'FILE' and 'USERINFO'.

**Example**

PARSE_URL('http://www.tableau.com', 'HOST') = 'www.tableau.com'

**PARSE_URL_QUERY**(string, key)
Returns the value of the specified query parameter in the given URL string. The query parameter is defined by the key.

**Example**

PARSE_URL_QUERY('http://www.tableau.com?page=1&cat=4', 'page') = '1'

**XPATH_BOOLEAN**(XML string, XPath expression string)
Returns true if the XPath expression matches a node or evaluates to true.

**Example**
XPATH_BOOLEAN('values> <value id="0">1</value><value id="1">5</value>', 'values/value [@id="1"] = 5') = true

XPATH_DOUBLE(XML string, XPath expression string)
Returns the floating-point value of the XPath expression.

Example
XPATH_DOUBLE('values> <value>1.0</value><value>5.5</value> </values>', 'sum(value/*)') = 6.5

XPATH_FLOAT(XML string, XPath expression string)
Returns the floating-point value of the XPath expression.

Example
XPATH_FLOAT('values> <value>1.0</value><value>5.5</value> </values>', 'sum(value/*)') = 6.5

XPATH_INT(XML string, XPath expression string)
Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example
XPATH_INT('values> <value>1</value><value>5</value> </values>', 'sum(value/*)') = 6

XPATH_LONG(XML string, XPath expression string)
Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example
XPATH_LONG('<values><value>1</value><value>5</value></values>','sum(value/*)') = 6

XPATH_SHORT(XML string, XPath expression string)
Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example
XPATH_SHORT('<values><value>1</value><value>5</value></values>','sum(value/*)') = 6

XPATH_STRING(XML string, XPath expression string)
Returns the text of the first matching node.

Example
XPATH_STRING('<sites><url domain="org">http://www.w3.org</url><url domain="com">http://www.tableau.com</url></sites>', 'sites/url[@domain="com"]') = 'http://www.tableau.com'

Google BigQuery Specific Functions

DOMAIN(string_url)
Given a URL string, returns the domain as a string.

Example
DOMAIN('http://www.google.com:80/index.html') = 'google.com'

GROUP_CONCAT(expression)
Concatenates values from each record into a single comma-delimited string. This function acts like a SUM() for strings.

Example
GROUP_CONCAT(Region) = "Central,East,West"

HOST(string_url)
Given a URL string, returns the host name as a string.

Example

LOG2(number)
Returns the logarithm base 2 of a number.

Example
LOG2(16) = '4.00'

LTRIM_THIS(string, string)
Returns the first string with any leading occurrence of the second string removed.

Example
LTRIM_THIS('[Sales-]','[') = 'Sales-

RTRIM_THIS(string, string)
Returns the first string with any trailing occurrence of the second string removed.

Example
RTRIM_THIS('[Market-]','[') = '[Market']
TIMESTAMP_TO_USEC(expression)
Converts a TIMESTAMP data type to a UNIX timestamp in microseconds.

Example
TIMESTAMP_TO_USEC(#2012-10-01 01:02:03#) = 1349053323000000

USEC_TO_TIMESTAMP(expression)
Converts a UNIX timestamp in microseconds to a TIMESTAMP data type.

Example
USEC_TO_TIMESTAMP(1349053323000000) = #2012-10-01 01:02:03#

TLD(string_url)
Given a URL string, returns the top level domain plus any country domain in the URL.

Example
TLD('http://www.google.com:80/index.html') = '.com'

TLD('http://www.google.co.uk:80/index.html') = '.co.uk'

FORMAT() Function Workarounds in Tableau
Tableau has no FORMAT() function for formatting fields, but it does provide a variety of means to change the structure and appearance of fields in a workbook:

- For information on formatting geographic fields, see Assign Geographic Roles on page 1941.
- For information on formatting date or number fields, see Set the default number format on page 975. And for information creating custom date formats, see Custom Date Formats on page 1071.
- For information on symbols and conventions that you can use to specify field formats,
see **Literal expression syntax** on page 1271.

- For information on formatting numbers and null values, see **Format Numbers and Null Values** on page 2389.

Tableau also provides a range of string functions that you can use to customize the appearance of string fields in a view. See **String Functions** on page 1288.

**Tableau Functions (by Category)**

The Tableau functions in this reference are organized by category. Click a category to browse its functions. Or press Ctrl+F (Command-F on a Mac) to open a search box that you can use to search the page for a specific function.

**Number functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
</table>
| **ABS**  | ABS(number)| Returns the absolute value of the given number.  
Examples:  
ABS(-7) = 7  
ABS([Budget Variance])  
The second example returns the absolute value for all the numbers contained in the **Budget Variance** field. |
| **ACOS** | ACOS(number)| Returns the arc cosine of the given number. The result is in radians.  
Example:  
ACOS(-1) = 3.14159265358979 |
| **ASIN** | ASIN(number)| Returns the arc sine of a given number. The result is in radians.  
Example: |
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIN</td>
<td>$\arcsin(1) = 1.5707963267949$</td>
<td></td>
</tr>
<tr>
<td>ATAN</td>
<td>$\arctan$ of a given number. The result is in radians.</td>
<td>$\arctan(180) = 1.5652408283942$</td>
</tr>
<tr>
<td>ATAN2</td>
<td>$\arctan$ of two given numbers ($x$ and $y$). The result is in radians.</td>
<td>$\arctan2(2, 1) = 1.10714871779409$</td>
</tr>
<tr>
<td>CEILING</td>
<td>Rounds a number to the nearest integer of equal or greater value.</td>
<td>$\text{CEILING}(3.1415) = 4$</td>
</tr>
</tbody>
</table>

### Availability by data source:

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Access</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Supported</td>
</tr>
<tr>
<td>Text File</td>
<td>Supported</td>
</tr>
<tr>
<td>Statistical File</td>
<td>Supported</td>
</tr>
<tr>
<td>Tableau Server</td>
<td>Supported</td>
</tr>
<tr>
<td>Actian Vector</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Supported</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Amazon Aurora</td>
<td></td>
</tr>
<tr>
<td>Amazon EMR Hadoop Hive</td>
<td></td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td></td>
</tr>
<tr>
<td>Aster Database</td>
<td></td>
</tr>
<tr>
<td>Cloudera Hadoop</td>
<td></td>
</tr>
<tr>
<td>DataStax Enterprise</td>
<td></td>
</tr>
<tr>
<td>EXASOL</td>
<td></td>
</tr>
<tr>
<td>Firebird</td>
<td></td>
</tr>
<tr>
<td>Google Analytics</td>
<td></td>
</tr>
<tr>
<td>Google BigQuery</td>
<td></td>
</tr>
<tr>
<td>Google Cloud SQL</td>
<td></td>
</tr>
<tr>
<td>Hortonworks Hadoop Hive</td>
<td></td>
</tr>
<tr>
<td>IBM BigInsights</td>
<td></td>
</tr>
<tr>
<td>IBM DB2</td>
<td></td>
</tr>
<tr>
<td>IBM PDA (Netezza)</td>
<td></td>
</tr>
<tr>
<td>MapR Hadoop Hive</td>
<td></td>
</tr>
<tr>
<td>MarkLogic</td>
<td></td>
</tr>
<tr>
<td>Microsoft Analysis Services</td>
<td></td>
</tr>
<tr>
<td>Microsoft PowerPivot</td>
<td></td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td></td>
</tr>
<tr>
<td>MySQL</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td></td>
</tr>
<tr>
<td>Oracle Essbase</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>COS</td>
<td>Returns the cosine of an angle. Specify the angle in radians.</td>
</tr>
<tr>
<td>COT</td>
<td>Returns the cotangent of an angle. Specify the angle in radians.</td>
</tr>
<tr>
<td>DEGREES</td>
<td>Converts a given number in radians to degrees.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>DEGREES</strong></td>
<td>( \text{DEGREES(PI( )/4)} = 45.0 )</td>
</tr>
<tr>
<td><strong>DIV</strong></td>
<td>( \text{DIV(integer1, integer2)} ) Returns the integer part of a division operation, in which integer1 is divided by integer2.</td>
</tr>
<tr>
<td><strong>EXP</strong></td>
<td>( \text{EXP(number)} ) Returns e raised to the power of the given number.</td>
</tr>
<tr>
<td><strong>FLOOR</strong></td>
<td>( \text{FLOOR(number)} ) Rounds a number to the nearest integer of equal or lesser value.</td>
</tr>
</tbody>
</table>

**Availability by data source:**

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Access</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Supported</td>
</tr>
<tr>
<td>Text File</td>
<td>Supported</td>
</tr>
<tr>
<td>Statistical File</td>
<td>Supported</td>
</tr>
<tr>
<td>Tableau Server</td>
<td>Supported</td>
</tr>
<tr>
<td>Actian Vector</td>
<td>Not supported</td>
</tr>
<tr>
<td>Platform</td>
<td>Support Status</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Amazon Aurora</td>
<td>Not supported</td>
</tr>
<tr>
<td>Amazon EMR Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td>Not supported</td>
</tr>
<tr>
<td>Aster Database</td>
<td>Not supported</td>
</tr>
<tr>
<td>Cloudera Hadoop</td>
<td>Supported</td>
</tr>
<tr>
<td>DataStax Enterprise</td>
<td>Supported</td>
</tr>
<tr>
<td>EXASOL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Firebird</td>
<td>Not supported</td>
</tr>
<tr>
<td>Google Analytics</td>
<td>Supported</td>
</tr>
<tr>
<td>Google BigQuery</td>
<td>Supported</td>
</tr>
<tr>
<td>Google Cloud SQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Hortonworks Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>IBM BigInsights</td>
<td>Not supported</td>
</tr>
<tr>
<td>IBM DB2</td>
<td>Not supported</td>
</tr>
<tr>
<td>IBM Netezza</td>
<td>Not supported</td>
</tr>
<tr>
<td>MapR Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>MarkLogic</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft Analysis Services</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft PowerPivot</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>Not supported</td>
</tr>
<tr>
<td>MySQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Oracle</td>
<td>Not supported</td>
</tr>
<tr>
<td>Oracle Essbase</td>
<td>Not supported</td>
</tr>
<tr>
<td>Database</td>
<td>Support Status</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>ParAccel</td>
<td>Not supported</td>
</tr>
<tr>
<td>Pivotal Greenplum</td>
<td>Not supported</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Progress OpenEdge</td>
<td>Not supported</td>
</tr>
<tr>
<td>Salesforce</td>
<td>Supported</td>
</tr>
<tr>
<td>SAP HANA</td>
<td>Not supported</td>
</tr>
<tr>
<td>SAP Sybase ASE</td>
<td>Not supported</td>
</tr>
<tr>
<td>SAP Sybase IQ</td>
<td>Not supported</td>
</tr>
<tr>
<td>Spark SQL</td>
<td>Supported</td>
</tr>
<tr>
<td>Splunk</td>
<td>Not supported</td>
</tr>
<tr>
<td>Teradata</td>
<td>Not supported</td>
</tr>
<tr>
<td>Teradata OLAP Connector</td>
<td>Not supported</td>
</tr>
<tr>
<td>Vertica</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

HEXBINX \( \text{HEXBINX}(\text{number, number}) \)

Maps an x, y coordinate to the x-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately.

HEXBINX and HEXBINY are binning and plotting functions for hexagonal bins. Hexagonal bins are an efficient and elegant option for visualizing data in an x/y plane such as a map. Because the bins are hexagonal, each bin closely approximates a circle and minimizes variation in the distance from the data point to the center of the bin. This makes the clustering both more accurate and informative.

Example:

HEXBINX([Longitude], [Latitude])
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HEXBINY</strong></td>
<td>Maps an x, y coordinate to the y-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately. Example: HEBIN([Longitude], [Latitude])</td>
<td></td>
</tr>
<tr>
<td><strong>LN</strong></td>
<td>Returns the natural logarithm of a number. Returns Null if number is less than or equal to 0.</td>
<td></td>
</tr>
<tr>
<td><strong>LOG</strong></td>
<td>Returns the logarithm of a number for the given base. If the base value is omitted, base 10 is used.</td>
<td></td>
</tr>
</tbody>
</table>
| **MAX** | Returns the maximum of the two arguments, which must be of the same type. Returns Null if either argument is Null. MAX can also be applied to a single field in an aggregate calculation. Examples: | MAX(4,7)  
MAX(Sales,Profit)  
MAX([First Name],[Last Name]) |
| **MIN** | Returns the minimum of the two arguments, which must be of the same type. Returns Null if either argument is Null. MIN can also be applied to a single field in an aggregate calculation. Examples: | MIN(4,7)  
MIN(Sales,Profit)  
MIN([First Name],[Last Name]) |
<p>| <strong>PI</strong> | Returns the numeric constant pi: 3.14159. | |
| <strong>POWER</strong> | Raises the number to the specified power. | |</p>
<table>
<thead>
<tr>
<th><strong>power</strong></th>
<th><strong>Examples:</strong></th>
</tr>
</thead>
</table>
|  | $\text{POWER}(5,2) = 5^2 = 25$  
$\text{POWER}($Temperature, $2)$ |
|  | You can also use the $^*$ symbol:  
$5^2 = \text{POWER}(5,2) = 25$ |
| **RADIANS** | **RADIANS (number)** | Converts the given number from degrees to radians.  
**Example:**  
$\text{RADIANS}(180) = 3.14159$ |
| **ROUND** | **ROUND (number, [decimals])** | Rounds numbers to a specified number of digits. The `decimals` argument specifies how many decimal points of precision to include in the final result. If `decimals` is omitted, `number` is rounded to the nearest integer.  
**Example:**  
This example rounds every `Sales` value to an integer:  
$\text{ROUND}(\text{Sales})$  
Some databases, such as SQL Server, allow specification of a negative length, where $-1$ rounds `number` to $10$'s, $-2$ rounds to $100$'s, and so on. This is not true of all databases. For example, it is not true of Excel or Access. |
| **SIGN** | **SIGN(number)** | Returns the sign of a number: The possible return values are $-1$ if the number is negative, $0$ if the number is zero, or $1$ if the number is positive.  
**Example:**  
If the average of the profit field is negative, then |
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
</table>
| SIN      | Returns the sine of an angle. Specify the angle in radians. Examples: | \[
\begin{align*}
\text{SIN}(0) &= 1.0 \\
\text{SIN}(\pi/4) &= 0.707106781186548
\end{align*}
\] |
| SQRT     | Returns the square root of a number. Example: | \[
\text{SQRT}(25) = 5
\] |
| SQUARE   | Returns the square of a number. Example: | \[
\text{SQUARE}(5) = 25
\] |
| TAN      | Returns the tangent of an angle. Specify the angle in radians. Example: | \[
\text{TAN}(\pi/4) = 1.0
\] |
| ZN       | Returns the expression if it is not null, otherwise returns zero. Use this function to use zero values instead of null values. Example: | \[
\text{ZN}([\text{Profit}]) = [\text{Profit}]
\] |

String functions
<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Definition</th>
</tr>
</thead>
</table>
| ASCII    | ASCII (string) | Returns the ASCII code for the first character of string.  
Example:  
`ASCII('A') = 65` |
| CHAR     | CHAR(number) | Returns the character encoded by the ASCII code number.  
Example:  
`CHAR(65) = 'A'` |
| CONTAINS  | CONTAINS (string, substring) | Returns true if the given string contains the specified substring.  
Example:  
`CONTAINS("Calculation", "alcu") = true` |
| ENDSWITH | ENDSWITH (string, substring) | Returns true if the given string ends with the specified substring. Trailing white spaces are ignored.  
Example:  
`ENDSWITH("Tableau", "leau") = true` |
| FIND     | FIND (string, substring, [start]) | Returns the index position of substring in string, or 0 if the substring isn't found. If the optional argument start is added, the function ignores any instances of substring that appear before the index position start. The first character in the string is position 1. |
| FINDNTH | FINDNTH (string, substring, occurrence) | Returns the position of the nth occurrence of substring within the specified string, where n is defined by the occurrence argument. 

**Note:** FINDNTH is not available for all data sources. 

Example: 

FINDNTH("Calculation", "a", 2) = 7 |
| LEFT | LEFT (string, number) | Returns the left-most number of characters in the string. 

Example: 

LEFT("Matador", 4) = "Mata" |
| LEN | LEN(string) | Returns the length of the string. 

Example: 

LEN("Matador") = 7 |
| LOWER | LOWER (string) | Returns string, with all characters lowercase. 

Example: |
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTRIM(string)</td>
<td>Returns the string with any leading spaces removed.</td>
<td>LTRIM(&quot; Matador &quot;) = &quot;Matador &quot;</td>
</tr>
<tr>
<td>MAX(a, b)</td>
<td>Returns the maximum of a and b (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, MAX finds the value that is highest in the sort sequence defined by the database for that column. It returns Null if either argument is Null.</td>
<td>MAX (&quot;Apple&quot;, &quot;Banana&quot;) = &quot;Banana&quot;</td>
</tr>
<tr>
<td>MID(string, start, [length])</td>
<td>Returns the string starting at index position start. The first character in the string is position 1. If the optional argument length is added, the returned string includes only that number of characters.</td>
<td>MID(&quot;Calculation&quot;, 2) = &quot;alculation&quot; MID(&quot;Calculation&quot;, 2, 5) = &quot;alcul&quot;</td>
</tr>
<tr>
<td>MIN(a, b)</td>
<td>Returns the minimum of a and b (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, MIN finds the value that is lowest in the sort sequence. It returns Null if either argument is Null.</td>
<td></td>
</tr>
</tbody>
</table>

LOWER("ProductVersion") = "productversion"
| **REPLACE** | REPLACE (string, substring, replacement) | Searches string for substring and replaces it with replacement. If substring is not found, the string is not changed.  
Example:  
```
REPLACE("Version8.5", "8.5", "9.0") = "Version9.0"
``` |
| **RIGHT** | RIGHT (string, number) | Returns the right-most number of characters in string.  
Example:  
```
RIGHT("Calculation", 4) = "tion"
``` |
| **RTRIM** | RTRIM (string) | Returns string with any trailing spaces removed.  
Example:  
```
RTRIM(" Calculation ") = " Calculation"
``` |
| **SPACE** | SPACE(number) | Returns a string that is composed of the specified number of repeated spaces.  
Example:  
```
SPACE(1) = " "
``` |
| **SPLIT** | SPLIT (string, delimiter, ) | Returns a substring from a string, using a delimiter character to divide the string into a sequence of tokens. |
The string is interpreted as an alternating sequence of delimiters and tokens. So for the string `abc-defgh-i-jkl`, where the delimiter character is `'-`, the tokens are `abc`, `defgh`, `i`, and `jkl`. Think of these as tokens 1 through 4. SPLIT returns the token corresponding to the token number. When the token number is positive, tokens are counted starting from the left end of the string; when the token number is negative, tokens are counted starting from the right.

Examples:

```
SPLIT ('a-b-c-d', '-', 2) = 'b'
SPLIT ('a|b|c|d', '|', -2) = 'c'
```

**Note:** The split and custom split commands are available for the following data sources types: Tableau data extracts, Microsoft Excel, Text File, PDF File, Salesforce, OData, Microsoft Azure Market Place, Google Analytics, Vertica, Oracle, MySQL, PostgreSQL, Teradata, Amazon Redshift, Aster Data, Google Big Query, Cloudera Hadoop Hive, Hortonworks Hive, and Microsoft SQL Server.

Some data sources impose limits on splitting string. The following table shows which data sources support negative token numbers (splitting from the right) and whether there is a limit on the number of splits allow per data source. A SPLIT function that specifies a negative token number and would be legal with other data sources will return this error with these data sources: “Splitting from right is not support by the data source.”

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Left/Right</th>
<th>Max-imum</th>
<th>Version lim-</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 1394 -
<table>
<thead>
<tr>
<th></th>
<th>Strains</th>
<th>Number of Splits</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tableau Data Extract</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Text file</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Salesforce</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>OData</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Google Analytics</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Tableau Data Server</td>
<td>Both</td>
<td>Infinite</td>
<td>Supported in version 9.0.</td>
</tr>
<tr>
<td>Vertica</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>MySQL</td>
<td>Both</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teradata</td>
<td>Left only</td>
<td>10</td>
<td>Version 14 and later</td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Aster Data-</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Base</td>
<td>Google BigQuery</td>
<td>Hortonworks Hadoop Hive Cloudera Hadoop</td>
<td>Microsoft SQL Server</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------</td>
<td>----------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td></td>
<td>Left only 10</td>
<td>Left only 10</td>
<td>Both 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Impala supported starting in version 2.3.0.</td>
<td>2008 and later</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>STARTSWITH</strong></th>
<th><strong>STARTSWITH</strong> (string, substring)</th>
<th>Returns true if string starts with substring. Leading white spaces are ignored. Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STARTSWITH(“Joker”, “Jo”) = true</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>TRIM</strong></th>
<th><strong>TRIM</strong> (string)</th>
<th>Returns the string with leading and trailing spaces removed. Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TRIM(&quot; Calculation &quot;) = &quot;Calculation&quot;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>UPPER</strong></th>
<th><strong>UPPER</strong> (string)</th>
<th>Returns string, with all characters uppercase. Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UPPER(&quot;Calculation&quot;) =</td>
<td></td>
</tr>
</tbody>
</table>
Date functions

Tableau provides a variety of date functions. Many of the examples use the # symbol with date expressions. See Literal expression syntax on page 1271 for an explanation of this symbol. Additionally, many date functions use date_part, which is a constant string argument. The valid date_part values that you can use are:

<table>
<thead>
<tr>
<th>date_part</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>'year'</td>
<td>Four-digit year</td>
</tr>
<tr>
<td>'quarter'</td>
<td>1-4</td>
</tr>
<tr>
<td>'month'</td>
<td>1-12 or &quot;January&quot;, &quot;February&quot;, and so on</td>
</tr>
<tr>
<td>'dayofyear'</td>
<td>Day of the year; Jan 1 is 1, Feb 1 is 32, and so on</td>
</tr>
<tr>
<td>'day'</td>
<td>1-31</td>
</tr>
<tr>
<td>'weekday'</td>
<td>1-7 or &quot;Sunday&quot;, &quot;Monday&quot;, and so on</td>
</tr>
<tr>
<td>'week'</td>
<td>1-52</td>
</tr>
<tr>
<td>'hour'</td>
<td>0-23</td>
</tr>
<tr>
<td>'minute'</td>
<td>0-59</td>
</tr>
<tr>
<td>'second'</td>
<td>0-60</td>
</tr>
</tbody>
</table>

**Note:** Date functions do not take account of the configured fiscal year start. See Fiscal Dates on page 1068.

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATEADD</td>
<td>DATEADD</td>
<td>Returns the specified date with the specified</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><strong>DATEADD</strong>&lt;br&gt;(date_part, interval, date)</td>
<td>number interval added to the specified date_part of that date. Supports ISO 8601 dates. Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DATEADD('month', 3, #2004-04-15#) = 2004-07-15 12:00:00 AM</td>
<td></td>
</tr>
<tr>
<td>This expression adds three months to the date #2004-04-15#.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DATEDIFF</strong>&lt;br&gt;(date_part, date1, date2, [start_of_week])</td>
<td>Returns the difference between date1 and date2 expressed in units of date_part. The start_of_week parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If it is omitted, the start of week is determined by the data source. See Date Properties for a Data Source on page 1059. Supports ISO 8601 dates. Examples:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'monday') = 1 DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'sunday') = 0</td>
<td></td>
</tr>
<tr>
<td>The first expression returns 1 because when start_of_week is 'monday', then 22 September (a Sunday) and 24 September (a Tuesday) are in different weeks. The second expression returns 0 because when start_of_week is 'sunday' then 22 September (a Sunday) and 24 September (a Tuesday) are in the same week.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| **DATENAME** | Returns date_part of date as a string. The start_of_week parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If start_of_week is omitted, the start of week is determined by the data source. See Date Properties for a Data Source on page 1059.

**Supports ISO 8601 dates.**

**Examples:**

```
DATENAME('year', #2004-04-15#) = "2004"
DATENAME('month', #2004-04-15#) = "April"
```

| **DATEPART** | Returns date_part of date as an integer.

The start_of_week parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If start_of_week is omitted, the start of week is determined by the data source. See Date Properties for a Data Source on page 1059.

**Note:** When the date_part is weekday, the start_of_week parameter is ignored. This is because Tableau relies on a fixed weekday ordering to apply offsets.

**Supports ISO 8601 dates.**

**Examples:**

```
```
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATETRUNC</strong></td>
<td>Truncates the specified date to the accuracy specified by the <code>date_part</code>. This function returns a new date. For example, when you truncate a date that is in the middle of the month at the month level, this function returns the first day of the month. The <code>start_of_week</code> parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If <code>start_of_week</code> is omitted, the start of week is determined by the data source. See <strong>Date Properties for a Data Source</strong> on page 1059. Supports ISO 8601 dates. Examples:</td>
</tr>
<tr>
<td><strong>DAY</strong></td>
<td>Returns the day of the given date as an integer. Examples:</td>
</tr>
<tr>
<td><strong>ISDATE</strong></td>
<td>Returns true if a given string is a valid date. Examples:</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>ISDATE</td>
<td>Returns true if the date is valid.</td>
</tr>
<tr>
<td>MAKEDATE</td>
<td>Returns a date value constructed from the specified year, month, and date. Available for Tableau Data Extracts. Check for availability in other data sources. Example:</td>
</tr>
<tr>
<td>MAKEDATETIME</td>
<td>Returns a datetime that combines a date and a time. The date can be a date, datetime, or a string type. The time must be a datetime. This function is available only for MySQL-compatible connections (which for Tableau are, in addition to MySQL, Amazon Aurora and Amazon Aurora). Examples:</td>
</tr>
<tr>
<td>MAKETIME</td>
<td>Returns a date value constructed from the specified hour, minute, and second. Available for Tableau Data Extracts. Check for availability in other data sources. Example:</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| MAX      | Returns the maximum of a and b (a and b must be of the same type). Returns Null if either argument is Null. | MAX(#2004-01-01#, #2004-03-01#) = 2004-03-01 12:00:00 AM  
MAX([ShipDate1], [ShipDate2]) |
| MIN      | Returns the minimum of a and b (a and b must be of the same type). Returns Null if either argument is Null. | MIN(#2004-01-01#, #2004-03-01#) = 2004-01-01 12:00:00 AM  
MIN([ShipDate1], [ShipDate2]) |
| MONTH    | Returns the month of the given date as an integer. | MONTH(#2004-04-15#) = 4 |
| NOW      | Returns the current date and time. The return varies depending on the nature of the connection:  
- For a live, unpublished connection, NOW returns the data source server time. |
- For a live, published connection, NOW returns the data source server time.
- For an unpublished extract, NOW returns the local system time.
- For a published extract, NOW returns the local time of the Tableau Server Data Engine. When there are multiple worker machines in different time zones, this can produce inconsistent results.

Example:

```
NOW( ) = 2004-04-15 1:08:21 PM
```

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TODAY</td>
<td>TODAY( )</td>
<td>Returns the current date.</td>
</tr>
<tr>
<td>YEAR</td>
<td>YEAR (date)</td>
<td>Returns the year of the given date as an integer.</td>
</tr>
</tbody>
</table>

Examples:

```
TODAY( ) = 2004-04-15
YEAR(#2004-04-15#) = 2004
```

**Logical functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>IF &lt;expr1&gt; AND &lt;expr2&gt; THEN &lt;then&gt; END</td>
<td>Performs a logical conjunction on two expressions.</td>
</tr>
</tbody>
</table>

Example:

```
IF (ATTR([Market]) = "Africa" AND
```
CASE <expression> WHEN <value1> THEN <return1> WHEN <value2> THEN <return2> ... ELSE <default return> END

Performs logical tests and returns appropriate values. The CASE function evaluates expression, compares it to a sequence of values, value1, value2, etc., and returns a result. When a value that matches expression is encountered, CASE returns the corresponding return value. If no match is found, the default return expression is used. If there is no default return and no values match, then Null is returned.

CASE is often easier to use than IIF or IF THEN ELSE. Typically, you use an IF function to perform a sequence of arbitrary tests, and you use a CASE function to search for a match to an expression. But a CASE function can always be rewritten as an IF function, although the CASE function will generally be more concise.

Many times you can use a group to get the same results as a complicated case function.

Examples:

CASE [Region] WHEN 'West' THEN 1 WHEN 'East' THEN 2 ELSE 3 END

CASE LEFT(DATENAME('weekday', [Order Date]),3) WHEN 'Sun' THEN 0 WHEN 'Mon' THEN 1 WHEN 'Tue' THEN 2 WHEN 'Wed' THEN 3 WHEN 'Thu' THEN 4 WHEN 'Fri' THEN 5 WHEN 'Sat' THEN 6 END
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| **ELSE** | IF <expr> THEN <then> ELSE <else> END | Tests a series of expressions returning the <then> value for the first true <expr>.  
Example:  

```
If [Profit] > 0 THEN 'Profitable'
ELSE 'Loss' END
``` |
| **ELSEIF** | IF <expr> THEN <then> ELSEIF <expr2> THEN <then2>... ELSE <else> END | Tests a series of expressions returning the <then> value for the first true <expr>.  
Example:  

```
IF [Profit] > 0 THEN 'Profitable'
ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END
``` |
| **END** | IF <expr> THEN <then> ELSEIF <expr2> THEN <then2>... ELSE <else> END | Tests a series of expressions returning the <then> value for the first true <expr>. Must be placed at the end of an expression.  
Example:  

```
IF [Profit] > 0 THEN 'Profitable'
ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END
``` |
| **IF** | IF <expr> THEN <then> ELSEIF <expr2> THEN <then2>... ELSE <else> END | Tests a series of expressions returning the <then> value for the first true <expr>.  
Example:  

```
IF [Profit] > 0 THEN 'Profitable'
ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END
``` |
<p>| <strong>IFNULL</strong> | IFNULL(expr1, &lt;expr2&gt;) | Returns &lt;expr1&gt; if it is not null, otherwise returns &lt;expr2&gt;. |</p>
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>expr2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IIF</td>
<td>Checks whether a condition is met, and returns one value if TRUE, another value if FALSE, and an optional third value or NULL if unknown.</td>
<td>[ \text{IIF([Profit] &gt; 0, 'Profit', 'Loss')} ]</td>
</tr>
<tr>
<td>ISDATE</td>
<td>Returns true if a given string is a valid date.</td>
<td>[ \text{ISDATE(&quot;2004-04-15&quot;) = True} ]</td>
</tr>
<tr>
<td>ISNULL</td>
<td>Returns true if the expression does not contain valid data (Null).</td>
<td>[ \text{ISNULL([Profit])} ]</td>
</tr>
<tr>
<td>MAX</td>
<td>Returns the maximum of a single expression across all records or the maximum of two expressions for each record.</td>
<td>[ \text{MAX([Sales])} ]</td>
</tr>
<tr>
<td>MIN</td>
<td>Returns the minimum of an expression across all records or the minimum of two expressions for each record.</td>
<td>[ \text{MIN([Sales])} ]</td>
</tr>
<tr>
<td><strong>expr2)</strong></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>MIN([Profit])</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>NOT</strong></th>
<th>Performs logical negation on an expression.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF NOT &lt;expr&gt; THEN &lt;then&gt; END</td>
<td>Example:</td>
</tr>
<tr>
<td>IF NOT [Profit] &gt; 0 THEN &quot;Unprofitable&quot; END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>OR</strong></th>
<th>Performs a logical disjunction on two expressions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF &lt;expr1&gt; OR &lt;expr2&gt; THEN &lt;then&gt; END</td>
<td>Example:</td>
</tr>
<tr>
<td>IF [Profit] &lt; 0 OR [Profit] = 0 THEN &quot;Needs Improvement&quot; END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>THEN</strong></th>
<th>Tests a series of expressions returning the &lt;then&gt; value for the first true &lt;expr&gt;.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF &lt;expr&gt; THEN &lt;then&gt; [ELSEIF expr2&gt; THEN &lt;then2&gt;... ] [ELSE &lt;else&gt; ] END</td>
<td>Example:</td>
</tr>
<tr>
<td>IF [Profit] &gt; 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Break even' ELSE 'unprofitable' END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>WHEN</strong></th>
<th>Finds the first &lt;value&gt; that matches &lt;expr&gt; and returns the corresponding &lt;return&gt;.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASE &lt;expr&gt; WHEN &lt;Value1&gt; THEN &lt;return1&gt; ... [ELSE &lt;else&gt;] END</td>
<td>Example:</td>
</tr>
<tr>
<td>CASE [RomanNumbereral] WHEN 'I' THEN 1 WHEN 'II' THEN 2 ELSE 3 END</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>ZN</strong></th>
<th>Returns &lt;expression&gt; if it is not null, otherwise returns zero.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZN(expression)</td>
<td></td>
</tr>
</tbody>
</table>
Aggregate functions

Aggregations and floating-point arithmetic: The results of some aggregations may not always be exactly as expected. For example, you may find that the Sum function returns a value such as -1.42e-14 for a column of numbers that you know should sum to exactly 0. This happens because the Institute of Electrical and Electronics Engineers (IEEE) 754 floating-point standard requires that numbers be stored in binary format, which means that numbers are sometimes rounded at extremely fine levels of precision. You can eliminate this potential distraction by using the ROUND function (see Number Functions on page 1275) or by formatting the number to show fewer decimal places.

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTR</td>
<td>ATTR(expression)</td>
<td>Returns the value of the expression if it has a single value for all rows. Otherwise returns an asterisk. Null values are ignored.</td>
</tr>
<tr>
<td>AVG</td>
<td>AVG(expression)</td>
<td>Returns the average of all the values in the expression. AVG can be used with numeric fields only. Null values are ignored.</td>
</tr>
<tr>
<td>COLLECT</td>
<td>COLLECT(spatial)</td>
<td>An aggregate calculation that combines the values in the argument field. Null values are ignored. <strong>Note:</strong> The COLLECT function can only be used with spatial fields. Example: COLLECT ([Geometry])</td>
</tr>
<tr>
<td>CORR</td>
<td>CORR</td>
<td>Returns the Pearson correlation coefficient of two</td>
</tr>
</tbody>
</table>
The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and −1 is an exact negative relationship.

CORR is available with the following data sources:

- Tableau data extracts (you can create an extract from any data source)
- Cloudera Hive
- EXASolution
- Firebird (version 3.0 and later)
- Google BigQuery
- Hortonworks Hadoop Hive
- Oracle
- PostgreSQL
- Presto
- SybaseIQ
- Teradata
- Vertica

For other data sources, consider either extracting the data or using WINDOW_CORR. See Table Calculation Functions on page 1347.

**Note:** The square of a CORR result is equivalent to the R-Squared value for a linear trend line model. See Trend Line Model Terms on page 1671.

**Example:**

You can use CORR to visualize correlation in a disaggregated scatter plot. The way to do this is to use a table-scoped level of detail expression. For example:
With a level of detail expression, the correlation is run over all rows. If you used a formula like `CORR(Sales, Profit)` (without the surrounding brackets to make it a level of detail expression), the view would show the correlation of each individual point in the scatter plot with each other point, which is undefined.

See **Table-Scoped** on page 1580

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COUNT</strong></td>
<td><code>COUNT(expression)</code> Returns the number of items in a group. Null values are not counted.</td>
</tr>
<tr>
<td><strong>COUNTD</strong></td>
<td><code>COUNTD(expression)</code> Returns the number of distinct items in a group. Null values are not counted. This function is not available in the following cases: workbooks created before Tableau Desktop 8.2 that use Microsoft Excel or text file data sources, workbooks that use the legacy connection, and workbooks that use Microsoft Access data sources. Extract your data into an extract file to use this function. See <strong>Extract Your Data</strong> on page 773.</td>
</tr>
<tr>
<td><strong>COVAR</strong></td>
<td><code>COVAR(expression1, expression2)</code> Returns the sample covariance of two expressions. Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when larger values of one variable tend to correspond to larger values of the other variable, on average. Sample covariance uses the number of non-null data points ( n - 1 ) to normalize the covariance calculation, rather than ( n ), which is used by the population covariance (available with the COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population. COVAR is available with the following data sources:</td>
</tr>
<tr>
<td><strong>Tableau data extracts (you can create an extract from any data source)</strong></td>
<td></td>
</tr>
<tr>
<td>Cloudera Hive</td>
<td></td>
</tr>
<tr>
<td>EXASolution</td>
<td></td>
</tr>
<tr>
<td>Firebird (version 3.0 and later)</td>
<td></td>
</tr>
<tr>
<td>Google BigQuery</td>
<td></td>
</tr>
<tr>
<td>Hortonworks Hadoop Hive</td>
<td></td>
</tr>
<tr>
<td>IBM PDA (Netezza)</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td></td>
</tr>
<tr>
<td>PostgreSQL</td>
<td></td>
</tr>
<tr>
<td>Presto</td>
<td></td>
</tr>
<tr>
<td>SybaseIQ</td>
<td></td>
</tr>
<tr>
<td>Teradata</td>
<td></td>
</tr>
<tr>
<td>Vertica</td>
<td></td>
</tr>
</tbody>
</table>

For other data sources, consider either extracting the data or using WINDOW_COVAR. See **Table Calculation Functions** on page 1347.

If expression1 and expression2 are the same—for example, COVAR([profit], [profit])—COVAR returns a value that indicates how widely values are distributed.

**Note:** The value of COVAR(X, X) is equivalent to the value of VAR(X) and also to the value of STDEV(X)^2.

Example:

The following formula returns the sample covariance of **Sales** and **Profit**.

\[
\text{COVAR}([\text{Sales}], [\text{Profit}])
\]

| **COVARP** | **COVARP** <br>(expression1, expression2) |
| **COVARP** <br>(expression1, expression2) |
| Returns the population covariance of two expressions. |
| Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when |
larger values of one variable tend to correspond to larger values of the other variable, on average. Population covariance is sample covariance multiplied by (n-1)/n, where n is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the COVAR function) is appropriate.

COVARP is available with the following data sources:

- Tableau data extracts (you can create an extract from any data source)
- Cloudera Hive
- EXASolution
- Firebird (version 3.0 and later)
- Google BigQuery
- Hortonworks Hadoop Hive
- IBM PDA (Netezza)
- Oracle
- PostgreSQL
- Presto
- SybaseIQ
- Teradata
- Vertica

For other data sources, consider either extracting the data or using WINDOW_COVARP. See Table Calculation Functions on page 1347.

If expression1 and expression2 are the same—for example, COVARP([profit], [profit])—COVARP returns a value that indicates how widely values are distributed.

**Note:** The value of COVARP(X, X) is equivalent to the value of VARP(X) and also to the value of STDEVP(X)^2.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MAX</strong></td>
<td>Returns the maximum of an expression across all records. If the expression is a string value, this function returns the last value where last is defined by alphabetical order.</td>
</tr>
<tr>
<td><strong>MIN</strong></td>
<td>Returns the minimum of an expression across all records. If the expression is a string value, this function returns the first value where first is defined by alphabetical order.</td>
</tr>
</tbody>
</table>
| **MEDIAN** | Returns the median of an expression across all records. Median can only be used with numeric fields. Null values are ignored. This function is not available for workbooks created before Tableau Desktop 8.2 or that use legacy connections. It is also not available for connections using any of the following data sources:  
- Access  
- Amazon Redshift  
- Cloudera Hadoop  
- HP Vertica  
- IBM DB2  
- IBM PDA (Netezza)  
- Microsoft SQL Server  
- MySQL  
- SAP HANA  
- Teradata  
For other data source types, you can extract your data into an extract file to use this function. See Extract Your Data on page 773. |
| **PERCENTILE** | Returns the percentile value from the given expression |

Example:
The following formula returns the population covariance of **Sales** and **Profit**.

\[
\text{COVARP}([\text{Sales}], [\text{Profit}])
\]
(expression, number) corresponding to the specified number. The number must be between 0 and 1 (inclusive)—for example, 0.66, and must be a numeric constant.

This function is available for the following data sources.

- Non-legacy Microsoft Excel and Text File connections.
- Extracts and extract-only data source types (for example, Google Analytics, OData, or Salesforce).
- Sybase IQ 15.1 and later data sources.
- Oracle 10 and later data sources.
- Cloudera Hive and Hortonworks Hadoop Hive data sources.
- EXASolution 4.2 and later data sources.

For other data source types, you can extract your data into an extract file to use this function. See Extract Your Data on page 773.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STDEV</td>
<td>Returns the statistical standard deviation of all values in the given expression based on a sample of the population.</td>
</tr>
<tr>
<td>STDEVP</td>
<td>Returns the statistical standard deviation of all values in the given expression based on a biased population.</td>
</tr>
<tr>
<td>SUM</td>
<td>Returns the sum of all values in the expression. SUM can be used with numeric fields only. Null values are ignored.</td>
</tr>
<tr>
<td>VAR</td>
<td>Returns the statistical variance of all values in the given expression based on a sample of the population.</td>
</tr>
<tr>
<td>VARP</td>
<td>Returns the statistical variance of all values in the given expression on the entire population.</td>
</tr>
</tbody>
</table>

User functions
<table>
<thead>
<tr>
<th>Functions</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FULLNAME</strong></td>
<td><code>FULLNAME()</code></td>
<td>Returns the full name for the current user. This is the Tableau Server or Tableau Online full name when the user is signed in; otherwise the local or network full name for the Tableau Desktop user.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>[Manager] = FULLNAME()</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If manager Dave Hallsten is signed in, this example returns True only if the Manager field in the view contained Dave Hallsten. When used as a filter, this calculated field can be used to create a user filter that only shows data that is relevant to the person signed in to the server.</td>
</tr>
<tr>
<td><strong>ISFULLNAME</strong></td>
<td><code>ISFULLNAME(string)</code></td>
<td>Returns true if the current user's full name matches the specified full name, or false if it does not match. This function uses the Tableau Server or Online full name when the user is signed in; otherwise it uses the local or network full name for the Tableau Desktop user.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>ISFULLNAME(&quot;Dave Hallsten&quot;)</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This example returns true if Dave Hallsten is the current user, otherwise it returns false.</td>
</tr>
<tr>
<td><strong>ISMEMBEROF</strong></td>
<td><code>ISMEMBEROF(string)</code></td>
<td>Returns true if the person currently using Tableau is a member of a group that matches the given string. If the person currently using Tableau is signed in, the group membership is determined by groups on Tableau Server or Tableau Online. If the person is not signed in, this function returns false.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>ISUSERNAME</td>
<td>Returns true if the current user’s username matches the specified username, or false if it does not match. This function uses the Tableau Server or Online username when the user is signed in; otherwise it uses the local or network username for the Tableau Desktop user.</td>
<td><code>ISUSERNAME(“dhallsten”)</code></td>
</tr>
<tr>
<td>USERDOMAIN</td>
<td>Returns the domain for the current user when the user is signed on to Tableau Server. Returns the Windows domain if the Tableau Desktop user is on a domain. Otherwise this function returns a null string.</td>
<td><code>[Manager]=USERNAME() AND [Domain]=USERDOMAIN()</code></td>
</tr>
<tr>
<td>USERNAME</td>
<td>Returns the username for the current user. This is the Tableau Server or Tableau Online username when the user is signed in; otherwise it is the local or network username for the Tableau Desktop user.</td>
<td><code>[Manager]=USERNAME()</code></td>
</tr>
</tbody>
</table>
Table calculations

**FIRST( )**

Returns the number of rows from the current row to the first row in the partition. For example, the view below shows quarterly sales. When FIRST( ) is computed within the Date partition, the offset of the first row from the second row is -1.

<table>
<thead>
<tr>
<th>Year of Order Date</th>
<th>Quarter of Order Date</th>
<th>Region</th>
<th>FIRST()</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>Q1</td>
<td>Central</td>
<td>$160,877</td>
</tr>
<tr>
<td></td>
<td></td>
<td>East</td>
<td>$231,411</td>
</tr>
<tr>
<td></td>
<td></td>
<td>South</td>
<td>$133,334</td>
</tr>
<tr>
<td></td>
<td></td>
<td>West</td>
<td>$185,961</td>
</tr>
<tr>
<td>2010</td>
<td>Q1</td>
<td>Central</td>
<td>$180,809</td>
</tr>
<tr>
<td></td>
<td></td>
<td>East</td>
<td>$180,123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>South</td>
<td>$273,943</td>
</tr>
<tr>
<td></td>
<td></td>
<td>West</td>
<td>$251,145</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>Central</td>
<td>$165,785</td>
</tr>
<tr>
<td></td>
<td></td>
<td>East</td>
<td>$224,302</td>
</tr>
<tr>
<td></td>
<td></td>
<td>South</td>
<td>$251,391</td>
</tr>
<tr>
<td></td>
<td></td>
<td>West</td>
<td>$196,976</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>Central</td>
<td>$116,613</td>
</tr>
<tr>
<td></td>
<td></td>
<td>East</td>
<td>$50,363</td>
</tr>
<tr>
<td></td>
<td></td>
<td>South</td>
<td>$194,601</td>
</tr>
<tr>
<td></td>
<td></td>
<td>West</td>
<td>$102,731</td>
</tr>
</tbody>
</table>

**Example**

When the current row index is 3, **FIRST( ) = -2**.

**INDEX( )**

Returns the index of the current row in the partition, without any sorting with regard to value. The first row index starts at 1. For example, the table below shows quarterly sales. When INDEX() is computed within the Date partition, the index of each row is 1, 2, 3, 4..., etc.
Example

For the third row in the partition, $\text{INDEX}() = 3$.

LAST()

Returns the number of rows from the current row to the last row in the partition. For example, the table below shows quarterly sales. When LAST() is computed within the Date partition, the offset of the last row from the second row is 5.

Example

When the current row index is 3 of 7, $\text{LAST}() = 4$.

LOOKUP(expression, [offset])

Returns the value of the expression in a target row, specified as a relative offset from the current row. Use FIRST() + n and LAST() - n as part of your offset definition for a target relative to the first/last rows in the partition. If offset is omitted, the row to compare to can be set on the field menu. This function returns NULL if the target row cannot be determined.
The view below shows quarterly sales. When $\text{LOOKUP}(\text{SUM}(\text{Sales}), 2)$ is computed within the Date partition, each row shows the sales value from 2 quarters into the future.

<table>
<thead>
<tr>
<th>Year of Order Date</th>
<th>Quarter of Order Date</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Central</td>
</tr>
<tr>
<td></td>
<td></td>
<td>East</td>
</tr>
<tr>
<td></td>
<td></td>
<td>South</td>
</tr>
<tr>
<td></td>
<td></td>
<td>West</td>
</tr>
<tr>
<td>2009</td>
<td>Q1</td>
<td>$160,877</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$231,411</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$133,934</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$185,961</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$197,213</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$204,914</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$337,813</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$213,507</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$302,678</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$165,201</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$283,806</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$206,512</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$297,208</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$226,983</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$214,846</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$230,291</td>
</tr>
<tr>
<td>2010</td>
<td>Q1</td>
<td>$180,609</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$180,123</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$273,943</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$251,145</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$195,785</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$224,882</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$251,391</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$195,976</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$116,613</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$50,363</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$194,601</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$102,731</td>
</tr>
</tbody>
</table>

Example

$\text{LOOKUP}(\text{SUM}([\text{Profit}]), \text{FIRST}()+2)$ computes the SUM(Profit) in the third row of the partition.

**PREVIOUS_VALUE**(expression)

Returns the value of this calculation in the previous row. Returns the given expression if the current row is the first row of the partition.

Example

$\text{SUM}([\text{Profit}]) \times \text{PREVIOUS\_VALUE}(1)$ computes the running product of SUM(Profit).
RANK(expression, ['asc' | 'desc'])

Returns the standard competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.

Example

The following image shows the effect of the various ranking functions (RANK, RANK_DENSE, RANK_MODIFIED, RANK_PERCENTILE, and RANK_UNIQUE) on a set of values. The data set contains information on 14 students (StudentA through StudentN); the Age column shows the current age of each student (all students are between 17 and 20 years of age). The remaining columns show the effect of each rank function on the set of age values, always assuming the default order (ascending or descending) for the function.

<table>
<thead>
<tr>
<th>Student</th>
<th>Age</th>
<th>RANK(expression)</th>
<th>RANK_DENSE(expression)</th>
<th>RANK_MODIFIED(expression)</th>
<th>RANK_PERCENTILE(expression)</th>
<th>RANK_UNIQUE(expression)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>18</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>70%</td>
<td>10%</td>
</tr>
<tr>
<td>Student</td>
<td>19</td>
<td>9</td>
<td>2</td>
<td>8</td>
<td>90%</td>
<td>96%</td>
</tr>
<tr>
<td>Student</td>
<td>18</td>
<td>8</td>
<td>3</td>
<td>9</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Student</td>
<td>17</td>
<td>13</td>
<td>4</td>
<td>14</td>
<td>144%</td>
<td>144%</td>
</tr>
<tr>
<td>Student</td>
<td>13</td>
<td>9</td>
<td>5</td>
<td>12</td>
<td>59%</td>
<td>59%</td>
</tr>
<tr>
<td>Student</td>
<td>14</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>70%</td>
<td>70%</td>
</tr>
<tr>
<td>Student</td>
<td>19</td>
<td>6</td>
<td>3</td>
<td>12</td>
<td>59%</td>
<td>59%</td>
</tr>
<tr>
<td>Student</td>
<td>13</td>
<td>9</td>
<td>5</td>
<td>12</td>
<td>59%</td>
<td>59%</td>
</tr>
<tr>
<td>Student</td>
<td>14</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>70%</td>
<td>70%</td>
</tr>
<tr>
<td>Student</td>
<td>19</td>
<td>6</td>
<td>3</td>
<td>12</td>
<td>59%</td>
<td>59%</td>
</tr>
<tr>
<td>Student</td>
<td>13</td>
<td>9</td>
<td>5</td>
<td>12</td>
<td>59%</td>
<td>59%</td>
</tr>
<tr>
<td>Student</td>
<td>14</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>70%</td>
<td>70%</td>
</tr>
<tr>
<td>Student</td>
<td>19</td>
<td>6</td>
<td>3</td>
<td>12</td>
<td>59%</td>
<td>59%</td>
</tr>
</tbody>
</table>

RANK_DENSE(expression, ['asc' | 'desc'])

Returns the dense rank for the current row in the partition. Identical values are assigned an identical rank, but no gaps are inserted into the number sequence. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (3, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.
For information on different ranking options, see Rank calculation on page 1549.

RANK_MODIFIED(expression, ['asc' | 'desc'])

Returns the modified competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 3, 3, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.

RANK_PERCENTILE(expression, ['asc' | 'desc'])

Returns the percentile rank for the current row in the partition. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is ascending.

With this function, the set of values (6, 9, 9, 14) would be ranked (0.25, 0.75, 0.75, 1.00).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.

RANK_UNIQUE(expression, ['asc' | 'desc'])

Returns the unique rank for the current row in the partition. Identical values are assigned different ranks. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 3, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.
RUNNING_AVG(expression)

Returns the running average of the given expression, from the first row in the partition to the current row.

The view below shows quarterly sales. When `RUNNING_AVG(SUM([Sales]))` is computed within the Date partition, the result is a running average of the sales values for each quarter.

Example

`RUNNING_AVG(SUM([Profit]))` computes the running average of SUM(Profit).

RUNNING_COUNT(expression)

Returns the running count of the given expression, from the first row in the partition to the current row.

Example

`RUNNING_COUNT(SUM([Profit]))` computes the running count of SUM(Profit).

RUNNING_MAX(expression)

- 1422 -
Returns the running maximum of the given expression, from the first row in the partition to the current row.

<table>
<thead>
<tr>
<th>Year of Order Date</th>
<th>Quarter of Order Date</th>
<th>Central</th>
<th>East</th>
<th>South</th>
<th>West</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>Q1</td>
<td>$150.877</td>
<td>$201.411</td>
<td>$133.954</td>
<td>$186.961</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$157.213</td>
<td>$204.914</td>
<td>$337.813</td>
<td>$213.507</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$302.679</td>
<td>$166.201</td>
<td>$293.806</td>
<td>$206.612</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$297.208</td>
<td>$229.893</td>
<td>$214.645</td>
<td>$320.291</td>
</tr>
<tr>
<td>2010</td>
<td>Q1</td>
<td>$180.509</td>
<td>$180.123</td>
<td>$273.543</td>
<td>$251.145</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$195.795</td>
<td>$224.662</td>
<td>$261.281</td>
<td>$195.970</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$116.613</td>
<td>$60.303</td>
<td>$194.601</td>
<td>$102.731</td>
</tr>
</tbody>
</table>

Example

RUNNING_MAX(SUM([Profit])) computes the running maximum of SUM(Profit).

RUNNING_MIN(expression)

Returns the running minimum of the given expression, from the first row in the partition to the current row.
Example

RUNNING_MIN(SUM([Profit])) computes the running minimum of SUM(Profit).

RUNNING_SUM(expression)

Returns the running sum of the given expression, from the first row in the partition to the current row.
Example

\texttt{RUNNING\_SUM(SUM([Profit]))} computes the running sum of \texttt{SUM(Profit)}

\textbf{SIZE()}

Returns the number of rows in the partition. For example, the view below shows quarterly sales. Within the Date partition, there are seven rows so the \texttt{Size()} of the Date partition is 7.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{example_view.png}
\caption{Example view showing quarterly sales by region.}
\end{figure}

Example

\texttt{SIZE() = 5} when the current partition contains five rows.

\textbf{SCRIPT\_BOOL}

Returns a Boolean result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use \texttt{.argn} (with a leading period) to reference parameters (.arg1, .arg2, etc.).

In Python expressions, use \texttt{_argn} (with a leading underscore).

\textbf{Examples}

In this R example, \texttt{.arg1} is equal to \texttt{SUM([Profit])}:

\begin{verbatim}
SCRIPT\_BOOL("is.finite(.arg1)", SUM([Profit]))
\end{verbatim}

The next example returns True for store IDs in Washington state, and False otherwise. This example could be the definition for a calculated field titled \texttt{IsStoreInWA}.

\begin{verbatim}
SCRIPT\_BOOL('grepl(".*_WA", .arg1, perl=TRUE)\',ATTR([Store ID]))
\end{verbatim}

A command for Python would take this form:
SCRIPT_BOOL("return map(lambda x : x > 0, _arg1)", SUM([Profit]))

**SCRIPT_INT**

Returns an integer result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use `.argn` (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use `_argn` (with a leading underscore).

**Examples**

In this R example, `.arg1` is equal to SUM([Profit]):

```r
SCRIPT_INT("is.finite(.arg1)", SUM([Profit]))
```

In the next example, k-means clustering is used to create three clusters:

```r
SCRIPT_INT('result <- kmeans(data.frame(.arg1,.arg2,.arg3,.arg4), 3);result$cluster;'", SUM([Petal length]), SUM([Petal width]), SUM([Sepal length]), SUM([Sepal width]))
```

A command for Python would take this form:

```python
SCRIPT_INT("return map(lambda x : int(x * 5), _arg1)", SUM([Profit]))
```

**SCRIPT_REAL**

Returns a real result from the specified expression. The expression is passed directly to a running external service instance. In

R expressions, use `.argn` (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use `_argn` (with a leading underscore).

**Examples**

In this R example, `.arg1` is equal to SUM([Profit]):

```r
SCRIPT_REAL("is.finite(.arg1)", SUM([Profit]))
```

The next example converts temperature values from Celsius to Fahrenheit.
A command for Python would take this form:

```
SCRIPT_REAL("return map(lambda x : x * 0.5, _arg1)", SUM ([Profit]))
```

---

**SCRIPT_STR**

Returns a string result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use \texttt{.argn} (with a leading period) to reference parameters (\texttt{.arg1}, \texttt{.arg2}, etc.)

In Python expressions, use \texttt{_argn} (with a leading underscore).

**Examples**

In this R example, \texttt{.arg1} is equal to \texttt{SUM([Profit])}:

```
SCRIPT_STR("is.finite(.arg1)", SUM([Profit]))
```

The next example extracts a state abbreviation from a more complicated string (in the original form \texttt{13XSL\_CA, A13\_WA}):

```
SCRIPT_STR('gsub(".*_","", .arg1)',ATTR([Store ID]))
```

A command for Python would take this form:

```
SCRIPT_STR("return map(lambda x : x[:2], _arg1)", ATTR([Region]))
```

---

**TOTAL(expression)**

Returns the total for the given expression in a table calculation partition.

**Example**

Assume you are starting with this view:
You open the calculation editor and create a new field which you name **Totality**:

```
TOTAL(SUM([Sales]))
```

You then drop **Totality** on Text, to replace **SUM(Sales)**. Your view changes such that it sums values based on the default **Compute Using** value:
This raises the question, What is the default **Compute Using** value? If you right-click (Control-click on a Mac) **Totality** in the Data pane and choose **Edit**, there is now an additional bit of information available:

![Edit Totality](image)

The default **Compute Using** value is **Table (Across)**. The result is that **Totality** is summing the values across each row of your table. Thus, the value that you see across each row is the sum of the values from the original version of the table.

The values in the 2011/Q1 row in the original table were $8601, $6579, $44262, and $15006. The values in the table after **Totality** replaces **SUM(Sales)** are all $74,448, which is the sum of the four original values.

Notice the triangle next to Totality after you drop it on Text:
This indicates that this field is using a table calculation. You can right-click the field and choose Edit Table Calculation to redirect your function to a different Compute Using value. For example, you could set it to Table (Down). In that case, your table would look like this:

![Table Image]

**WINDOW_AVG(expression, [start, end])**

Returns the average of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.
For example, the view below shows quarterly sales. A window average within the Date partition returns the average sales across all dates.

Example

WINDOW_AVG(SUM([Profit]), FIRST()+1, 0) computes the average of SUM(Profit) from the second row to the current row.

**WINDOW_CORR(expression1, expression2, [start, end])**

Returns the Pearson correlation coefficient of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.

The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and -1 is an exact negative relationship.

There is an equivalent aggregation function: CORR. See Tableau Functions (Alphabetical) on page 1449.

Example

The following formula returns the Pearson correlation of SUM(Profit) and SUM(Sales) from the five previous rows to the current row.

WINDOW_CORR(SUM([Profit]), SUM([Sales]), -5, 0)
WINDOW_COUNT(expression, [start, end])

Returns the count of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

WINDOW_COUNT(SUM([Profit]), FIRST()+1, 0) computes the count of SUM(Profit) from the second row to the current row.

WINDOW_COVAR(expression1, expression2, [start, end])

Returns the sample covariance of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end arguments are omitted, the window is the entire partition.

Sample covariance uses the number of non-null data points n - 1 to normalize the covariance calculation, rather than n, which is used by the population covariance (with the WINDOW_COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population.

There is an equivalent aggregation function: COVAR. See Tableau Functions (Alphabetical) on page 1449.

Example

The following formula returns the sample covariance of SUM(Profit) and SUM(Sales) from the two previous rows to the current row.

WINDOW_COVAR(SUM([Profit]), SUM([Sales]), -2, 0)

WINDOW_COVARP(expression1, expression2, [start, end])

Returns the population covariance of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.
Population covariance is sample covariance multiplied by \((n-1)/n\), where \(n\) is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the `WINDOW_COVAR` function) is appropriate.

There is an equivalent aggregation function: `COVARP`. Tableau Functions (Alphabetical) on page 1449.

Example

The following formula returns the population covariance of `SUM(Profit)` and `SUM(Sales)` from the two previous rows to the current row.

`WINDOW_COVARP(SUM([Profit]), SUM([Sales]), -2, 0)`

`WINDOW_MEDIAN(expression, [start, end])`

Returns the median of the expression within the window. The window is defined by means of offsets from the current row. Use `FIRST()+n` and `LAST()-n` for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly profit. A window median within the Date partition returns the median profit across all dates.
Example

\( \text{WINDOW\_MEDIAN}(\text{SUM}([\text{Profit}]), \text{FIRST}()+1, 0) \) computes the median of \( \text{SUM} \) (Profit) from the second row to the current row.

**WINDOW\_MAX(expression, [start, end])**

Returns the maximum of the expression within the window. The window is defined by means of offsets from the current row. Use \( \text{FIRST}()+n \) and \( \text{LAST}()-n \) for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window maximum within the Date partition returns the maximum sales across all dates.

Example

\( \text{WINDOW\_MAX}(\text{SUM}([\text{Sales}]), \text{FIRST}(), \text{LAST}()) \) computes the maximum of \( \text{SUM} \) (Sales) from the second row to the current row.

**WINDOW\_MIN(expression, [start, end])**

Returns the minimum of the expression within the window. The window is defined by means of offsets from the current row. Use \( \text{FIRST}()+n \) and \( \text{LAST}()-n \) for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.
For example, the view below shows quarterly sales. A window minimum within the Date partition returns the minimum sales across all dates.

![Quarterly Sales View](image)

**Example**

\[
\text{WINDOW\_MIN} (\text{SUM}([\text{Profit}]), \text{FIRST()} + 1, 0)
\]

computes the minimum of SUM(Profit) from the second row to the current row.

**WINDOW\_PERCENTILE(expression, number, [start, end])**

Returns the value corresponding to the specified percentile within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

**Example**

\[
\text{WINDOW\_PERCENTILE} (\text{SUM}([\text{Profit}]), 0.75, -2, 0)
\]

returns the 75th percentile for SUM(Profit) from the two previous rows to the current row.

**WINDOW\_STDEV(expression, [start, end])**

Returns the sample standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.
Example

WINDOW_STDEV(SUM([Profit]), FIRST()+1, 0) computes the standard deviation of SUM(Profit) from the second row to the current row.

WINDOW_STDEVP(expression, [start, end])

Returns the biased standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

WINDOW_STDEVP(SUM([Profit]), FIRST()+1, 0) computes the standard deviation of SUM(Profit) from the second row to the current row.

WINDOW_SUM(expression, [start, end])

Returns the sum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window sum computed within the Date partition returns the summation of sales across all quarters.
Example

WINDOW_SUM(SUM([Profit]), FIRST() + 1, 0) computes the sum of SUM(Profit) from the second row to the current row.

WINDOW_VAR(expression, [start, end])

Returns the sample variance of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

WINDOW_VAR(SUM([Profit]), FIRST()+1, 0) computes the variance of SUM(Profit) from the second row to the current row.

WINDOW_VARP(expression, [start, end])

Returns the biased variance of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

WINDOW_VARP(SUM([Profit]), FIRST()+1, 0) computes the variance of SUM(Profit) from the second row to the current row.

Pass-Through functions (RAWSQL)

These RAWSQL pass-through functions can be used to send SQL expressions directly to the database, without first being interpreted by Tableau. If you have custom database functions that Tableau doesn’t know about, you can use these pass-through functions to call these custom functions.

Your database usually will not understand the field names that are shown in Tableau. Because Tableau does not interpret the SQL expressions you include in the pass-through functions, using the Tableau field names in your expression may cause errors. You can use a substitution
syntax to insert the correct field name or expression for a Tableau calculation into pass-through SQL. For example, if you had a function that computed the median of a set of values, you could call that function on the Tableau column [Sales] like this:

```sql
RAWSQLAGG_REAL("MEDIAN(%1)", [Sales])
```

Because Tableau does not interpret the expression, you must define the aggregation. You can use the RAWSQLAGG functions described below when you are using aggregated expressions.

RAWSQL pass-through functions will not work with published data sources.

These functions may return different results starting in Tableau Desktop 8.2 than they did in earlier versions of Tableau Desktop. This is because Tableau now uses ODBC for pass-through functions instead of OLE DB. ODBC truncates when returning real values as integer; OLE DB rounds when returning real values as integer.

RAWSQL Functions

The following RAWSQL functions are available in Tableau.

```sql
RAWSQL_BOOL("sql_expr", [arg1], ...[argN])
```

Returns a Boolean result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In the example, %1 is equal to [Sales] and %2 is equal to [Profit].

```sql
RAWSQL_BOOL("IIF( %1 > %2, True, False)", [Sales], [Profit])
```

```sql
RAWSQL_DATE("sql_expr", [arg1], ...[argN])
```
Returns a Date result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In this example, %1 is equal to [Order Date].

RAWSQL_DATE("%1", [Order Date])

RAWSQL_DATETIME("sql_expr", [arg1], ...[argN])

Returns a Date and Time result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].

Example

RAWSQL_DATETIME("MIN(%1)", [Delivery Date])

RAWSQL_INT("sql_expr", [arg1], ...[argN])

Returns an integer result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example

RAWSQL_INT("500 + %1", [Sales])

RAWSQL_REAL("sql_expr", [arg1], ...[argN])

Returns a numeric result from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]
Example

RAWSQL_REAL("-123.98 * %1", [Sales])

RAWSQL_SPATIAL

Returns a Spatial from a given SQL expression that is passed directly to the underlying data source. Use %n in the SQL expression as a substitution syntax for database values.

Example

In this example, %1 is equal to [Geometry].

RAWSQL_SPATIAL("%1", [Geometry])

RAWSQL_STR("sql_expr", [arg1], ...[argN])

Returns a string from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Customer Name].

Example

RAWSQL_STR("%1", [Customer Name])

RAWSQLAGG_BOOL("sql_expr", [arg1], ...[argN])

Returns a Boolean result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In the example, %1 is equal to [Sales] and %2 is equal to [Profit].

RAWSQLAGG_BOOL("SUM( %1) >SUM( %2)", [Sales], [Profit])
RAWSQLAGG_DATE(“sql_expr”, [arg1], ...[argN])

Returns a Date result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Order Date].

Example

RAWSQLAGG_DATE(“MAX(%1)”, [Order Date])

RAWSQLAGG_DATETIME(“sql_expr”, [arg1], ...[argN])

Returns a Date and Time result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].

Example

RAWSQLAGG_DATETIME(“MIN(%1)”, [Delivery Date])

RAWSQLAGG_INT(“sql_expr”, [arg1], ...[argN])

Returns an integer result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example

RAWSQLAGG_INT(“500 + SUM(%1)”, [Sales])

RAWSQLAGG_REAL(“sql_expr”, [arg1], ...[argN])

Returns a numeric result from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]
Example

RAWSQLAGG_REAL("SUM( %1)", [Sales])

RAWSQLAGG_STR("sql_expr", [arg1,...[argN]])

Returns a string from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Discount].

Example

RAWSQLAGG_STR("AVG(%1)", [Discount])

Additional functions

Regular Expressions

REGEXP_REPLACE(string, pattern, replacement)

Returns a copy of the given string where the regular expression pattern is replaced by the replacement string. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern and the replacement must be constants.

For information on regular expression syntax, see your data source’s documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_REPLACE('abc 123', '\s', '-') = 'abc-123'
REGEXP_MATCH(string, pattern)

Returns true if a substring of the specified string matches the regular expression pattern. This function is available for Text File, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Impala 2.3.0 (through Cloudera Hadoop data sources), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_MATCH('-(\[1234].[The.Market])-', '^[\s*(\w*\.)\s*(*\w*)]*')=true

REGEXP_EXTRACT(string, pattern)

Returns the portion of the string that matches the regular expression pattern. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_EXTRACT('abc 123', '[a-z]+\s+(\d+)') = '123'
REGEXP_EXTRACT_NTH(string, pattern, index)

Returns the portion of the string that matches the regular expression pattern. The substring is matched to the nth capturing group, where n is the given index. If index is 0, the entire string is returned. This function is available for Text File, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_EXTRACT_NTH('abc 123', '([a-z]+)\s+(\d+)', 2) = '123'

Hadoop Hive Specific Functions

**Note:** Only the PARSE_URL and PARSE_URL_QUERY functions are available for Cloudera Impala data sources.

GET_JSON_OBJECT(JSON string, JSON path)

Returns the JSON object within the JSON string based on the JSON path.

PARSE_URL(string, url_part)

Returns a component of the given URL string where the component is defined by url_part. Valid url_part values include: 'HOST', 'PATH', 'QUERY', 'REF', 'PROTOCOL', 'AUTHORITY', 'FILE' and 'USERINFO'.
Example

PARSE_URL('http://www.tableau.com', 'HOST') = 'www.tableau.com'

PARSE_URL_QUERY(string, key)

Returns the value of the specified query parameter in the given URL string. The query parameter is defined by the key.

Example

PARSE_URL_QUERY('http://www.tableau.com?page=1&cat=4', 'page') = '1'

XPATH_BOOLEAN(XML string, XPath expression string)

Returns true if the XPath expression matches a node or evaluates to true.

Example

XPATH_BOOLEAN('<values> <value id="0">1</value><value id="1">5</value> </values>', 'values/value [@id="1"] = 5') = true

XPATH_DOUBLE(XML string, XPath expression string)

Returns the floating-point value of the XPath expression.

Example

XPATH_DOUBLE('<values><value>1.0</value><value>5.5</value></values>', 'sum(value/*)') = 6.5

XPATH_FLOAT(XML string, XPath expression string)

Returns the floating-point value of the XPath expression.
Example

XPATH_FLOAT('<values><value>1.0</value><value>5.5</value></values>','sum(value/*')) = 6.5

XPATH_INT(XML string, XPath expression string)
Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_INT('<values><value>1</value><value>5</value></values>','sum(value/*')) = 6

XPATH_LONG(XML string, XPath expression string)
Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_LONG('<values><value>1</value><value>5</value></values>','sum(value/*')) = 6

XPATH SHORT/XML string, XPath expression string)
Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_SHORT('<values><value>1</value><value>5</value></values>','sum(value/*')) = 6

XPATH STRING/XML string, XPath expression string)
Returns the text of the first matching node.
Example

XPATH_STRING('<sites><url domain="org">http://www.w3.org</url> <url domain="com">http://www.tableau.com</url></sites>', 'sites/url[@domain="com"]') = 'http://www.tableau.com'

Google BigQuery Specific Functions

DOMAIN(string_url)

Given a URL string, returns the domain as a string.

Example

DOMAIN('http://www.google.com:80/index.html') = 'google.com'

GROUP_CONCAT(expression)

Concatenates values from each record into a single comma-delimited string. This function acts like a SUM() for strings.

Example

GROUP_CONCAT(Region) = "Central,East,West"

HOST(string_url)

Given a URL string, returns the host name as a string.

Example


LOG2(number)
Returns the logarithm base 2 of a number.

Example

\text{LOG2}(16) = '4.00'

\text{LTRIM\_THIS}(\text{string}, \text{string})

Returns the first string with any leading occurrence of the second string removed.

Example

\text{LTRIM\_THIS}('[-Sales-]','-'') = 'Sales-'

\text{RTRIM\_THIS}(\text{string}, \text{string})

Returns the first string with any trailing occurrence of the second string removed.

Example

\text{RTRIM\_THIS}('[-Market-]','-'') = '[-Market'

\text{TIMESTAMP\_TO\_USEC}(\text{expression})

Converts a \text{TIMESTAMP} data type to a UNIX timestamp in microseconds.

Example

\text{TIMESTAMP\_TO\_USEC}('#2012-10-01 01:02:03#')=1349053323000000

\text{USEC\_TO\_TIMESTAMP}(\text{expression})

Converts a UNIX timestamp in microseconds to a \text{TIMESTAMP} data type.

Example

\text{USEC\_TO\_TIMESTAMP}(1349053323000000) = #2012-10-01 01:02:03#
TLD(string_url)

Given a URL string, returns the top level domain plus any country domain in the URL.

Example

TLD('http://www.google.com:80/index.html') = '.com'
TLD('http://www.google.co.uk:80/index.html') = '.co.uk'

Want to learn more about functions?

Read the functions topics.

See Also

Tableau Functions (Alphabetical)

The Tableau functions in this reference are organized alphabetically. Click a letter to see functions that start with it. If no functions start with that letter, the functions that start with the next letter in the alphabet are shown. You can also press Ctrl+F (Command-F on a Mac) to open a search box that you can use to search the page for a specific function.

ABS(number)

Returns the absolute value of the given number.

Examples

\[ \text{ABS}(-7) = 7 \]
ABS([Budget Variance])

The second example returns the absolute value for all the numbers contained in the Budget Variance field.

**ACOS(number)**

Returns the arc cosine of the given number. The result is in radians.

Example

ACOS(-1) = 3.14159265358979

**ASCII(string)**

Return the ASCII code for the first character of string.

Example

ASCII('A') = 65

**ASIN(number)**

Returns the arc sine of a given number. The result is in radians.

Example

ASIN(1) = 1.5707963267949

**ATAN(number)**

Returns the arc tangent of a given number. The result is in radians.

Example

ATAN(180) = 1.5652408283942
ATAN2(y number, x number)

Returns the arc tangent of two given numbers (x and y). The result is in radians.

Example

ATAN2(2, 1) = 1.10714871779409

ATTR(expression)

Returns the value of the expression if it has a single value for all rows. Otherwise returns an asterisk. Null values are ignored.

AVG(expression)

Returns the average of all the values in the expression. AVG can be used with numeric fields only. Null values are ignored.

CASE expression WHEN value1 THEN return1
WHEN value2 THEN return2...ELSE default return END

Use the CASE function to perform logical tests and return appropriate values. CASE is often easier to use than IF or IF THEN ELSE. The CASE function evaluates expression, compares it to a sequence of values, value1, value2, etc., and returns a result. When a value that matches expression is encountered, CASE returns the corresponding return value. If no match is found, the default return expression is used. If there is no default return and no values match, then Null is returned.

Typically, you use an IF function to perform a sequence of arbitrary tests, and you use a CASE function to search for a match to an expression. But a CASE function can always be rewritten as an IF function, although the CASE function will generally be more concise.
Many times you can use a group to get the same results as a complicated case function.

Examples

```sql
CASE [Region] WHEN 'West' THEN 1 WHEN 'East' THEN 2 ELSE 3 END
```

```sql
CASE LEFT(DATENAME('weekday',[Order Date]),3) WHEN 'Sun' THEN 0 WHEN 'Mon' THEN 1 WHEN 'Tue' THEN 2 WHEN 'Wed' THEN 3 WHEN 'Thu' THEN 4 WHEN 'Fri' THEN 5 WHEN 'Sat' THEN 6 END
```

**CEILING(number)**

Rounds a number to the nearest integer of equal or greater value.

Example

```
CEILING(3.1415) = 4
```

Availability by data source

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Access</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Supported</td>
</tr>
<tr>
<td>Text File</td>
<td>Supported</td>
</tr>
<tr>
<td>Statistical File</td>
<td>Supported</td>
</tr>
<tr>
<td>Tableau Server</td>
<td>Supported</td>
</tr>
<tr>
<td>Actian Vectorwise</td>
<td>Not supported</td>
</tr>
<tr>
<td>Amazon Aurora</td>
<td>Not supported</td>
</tr>
<tr>
<td>Application</td>
<td>Support Status</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Amazon EMR Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td>Not supported</td>
</tr>
<tr>
<td>Aster Database</td>
<td>Not supported</td>
</tr>
<tr>
<td>Cloudera Hadoop</td>
<td>Supported</td>
</tr>
<tr>
<td>DataStax Enterprise</td>
<td>Supported</td>
</tr>
<tr>
<td>EXASOL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Firebird</td>
<td>Not supported</td>
</tr>
<tr>
<td>Google Analytics</td>
<td>Supported</td>
</tr>
<tr>
<td>Google BigQuery</td>
<td>Supported</td>
</tr>
<tr>
<td>Google Cloud SQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Hortonworks Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>IBM BigInsights</td>
<td>Not supported</td>
</tr>
<tr>
<td>IBM DB2</td>
<td>Not supported</td>
</tr>
<tr>
<td>IBM PDA (Netezza)</td>
<td>Not supported</td>
</tr>
<tr>
<td>MapR Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>MarkLogic</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft Analysis Services</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft PowerPivot</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>Not supported</td>
</tr>
<tr>
<td>MySQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Oracle</td>
<td>Not supported</td>
</tr>
<tr>
<td>Database</td>
<td>Support</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Oracle Essbase</td>
<td>Not supported</td>
</tr>
<tr>
<td>Actian Matrix (ParAccel)</td>
<td>Not supported</td>
</tr>
<tr>
<td>Pivotal Greenplum</td>
<td>Not supported</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Progress OpenEdge</td>
<td>Not supported</td>
</tr>
<tr>
<td>Salesforce</td>
<td>Supported</td>
</tr>
<tr>
<td>SAP HANA</td>
<td>Not supported</td>
</tr>
<tr>
<td>SAP Sybase ASE</td>
<td>Not supported</td>
</tr>
<tr>
<td>SAP Sybase IQ</td>
<td>Not supported</td>
</tr>
<tr>
<td>Spark SQL</td>
<td>Supported</td>
</tr>
<tr>
<td>Splunk</td>
<td>Not supported</td>
</tr>
<tr>
<td>Teradata</td>
<td>Not supported</td>
</tr>
<tr>
<td>Teradata OLAP Connector</td>
<td>Not supported</td>
</tr>
<tr>
<td>Vertica</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

**CHAR(number)**

Returns the character encoded by the ASCII code `number`.

**Example**

```
CHAR(65) = 'A'
```

**COLLECT (spatial)**

An aggregate calculation that combines the values in the argument field. Null values are ignored.
Note: The COLLECT function can only be used with spatial fields.

Example

COLLECT ([Geometry])

CONTAINS(string, substring)

Returns true if the given string contains the specified substring.

Example

CONTAINS(“Calculation”, “alcu”) = true

CORR(expression 1, expression2)

Returns the Pearson correlation coefficient of two expressions.

The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and −1 is an exact negative relationship.

CORR is available with the following data sources:

- Tableau data extracts (you can create an extract from any data source)
- Cloudera Hive
- EXASOL
- Firebird (version 3.0 and later)
- Google BigQuery
- Hortonworks Hadoop Hive
- IBM PDA (Netezza)
- Oracle
- PostgreSQL
- Presto
- SybaseIQ
- Teradata
- Vertica
For other data sources, consider either extracting the data or using WINDOW_CORR. See Table Calculation Functions on page 1347.

**Note:** The square of a CORR result is equivalent to the R-Squared value for a linear trend line model. See Trend Line Model Terms on page 1671.

**Example**

You can use CORR to visualize correlation in a disaggregated scatter plot. The way to do this is to use a table-scoped level of detail expression. For example:

```
{CORR(Sales, Profit)}
```

With a level of detail expression, the correlation is run over all rows. If you used a formula like CORR(Sales, Profit) (without the surrounding brackets to make it a level of detail expression), the view would show the correlation of each individual point in the scatter plot with each other point, which is undefined.

See Table-Scoped on page 1580

**COS(number)**

Returns the cosine of an angle. Specify the angle in radians.

**Example**

```
COS(PI( )/4) = 0.707106781186548
```

**COT(number)**

Returns the cotangent of an angle. Specify the angle in radians.

**Example**

```
COT(PI( )/4) = 1
```
COUNT(expression)

Returns the number of items in a group. Null values are not counted.

COUNTD(expression)

Returns the number of distinct items in a group. Null values are not counted. This function is not available in the following cases: workbooks created before Tableau Desktop 8.2 that use Microsoft Excel or text file data sources, workbooks that use the legacy connection, and workbooks that use Microsoft Access data sources. Extract your data into an extract file to use this function. See Extract Your Data on page 773.

COVAR(expression1, expression2)

Returns the sample covariance of two expressions.

Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when larger values of one variable tend to correspond to larger values of the other variable, on average. Sample covariance uses the number of non-null data points n - 1 to normalize the covariance calculation, rather than n, which is used by the population covariance (available with the COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population.

COVAR is available with the following data sources:

- Tableau data extracts (you can create an extract from any data source)
- Cloudera Hive
- EXASOL
- Firebird (version 3.0 and later)
- Google BigQuery
- Hortonworks Hadoop Hive
- IBM PDA (Netezza)
- Oracle
- PostgreSQL
- Presto
- SybaseIQ
- Teradata
- Vertica

For other data sources, consider either extracting the data or using WINDOW_COVAR. See Table Calculation Functions on page 1347.

If expression1 and expression2 are the same—for example, COVAR([profit], [profit])—COVAR returns a value that indicates how widely values are distributed.

**Note:** The value of COVAR(X, X) is equivalent to the value of VAR(X) and also to the value of STDEV(X)^2.

**Example**

The following formula returns the sample covariance of Sales and Profit.

\[
\text{COVAR}([\text{Sales}], [\text{Profit}])
\]

**COVARP(expression 1, expression2)**

Returns the *population covariance* of two expressions.

Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when larger values of one variable tend to correspond to larger values of the other variable, on average. Population covariance is sample covariance multiplied by \((n-1)/n\), where \(n\) is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the COVAR function) is appropriate.

COVARP is available with the following data sources:

- Tableau data extracts (you can create an extract from any data source)
- Cloudera Hive
- EXASOL
- Firebird (version 3.0 and later)
- Google BigQuery
- Hortonworks Hadoop Hive
- Oracle
- PostgreSQL
For other data sources, consider either extracting the data or using WINDOW_COVARP. See Table Calculation Functions on page 1347.

If expression1 and expression2 are the same—for example, COVARP([profit], [profit])—COVARP returns a value that indicates how widely values are distributed.

Note: The value of COVARP(X, X) is equivalent to the value of VARP(X) and also to the value of STDEVP(X)^2.

Example

The following formula returns the population covariance of Sales and Profit.

\[ \text{COVARP}([\text{Sales}], [\text{Profit}]) \]

---

**DATE(expression)**

Returns a date given a number, string, or date expression.

**Examples**

\[ \text{DATE([Employee Start Date])} \]

\[ \text{DATE("April 15, 2004")} = \#\text{April 15, 2004}\# \]

\[ \text{DATE("4/15/2004")} \]

\[ \text{DATE(#2006-06-15 14:52#)} = \#2006-06-15# \]

Quotation marks are required in the second and third examples.
DATEADD(date_part, interval, date)

Returns the specified date with the specified number interval added to the specified date_part of that date.

Example

DATEADD('month', 3, #2004-04-15#) = 2004-07-15 12:00:00 AM

This expression adds three months to the date #2004-04-15#.

DATEDIFF(date_part, date1, date2, [start_of_week])

Returns the difference between date1 and date2 expressed in units of date_part.

The start_of_week parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If it is omitted, the start of week is determined by the data source. See Date Properties for a Data Source on page 1059.

Example

DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'monday')= 1
DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'sunday')= 0

The first expression returns 1 because when start_of_week is 'monday', then 22 September (a Sunday) and 24 September (a Tuesday) are in different weeks. The second expression returns 0 because when start_of_week is 'sunday' then 22 September (a Sunday) and 24 September (a Tuesday) are in the same week.

DATENAME(date_part, date, [start_of_week])

Returns date_part of date as a string. The start_of_week parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible
values are 'monday', 'tuesday', etc. If start_of_week is omitted, the start of week is determined by the data source. See Date Properties for a Data Source on page 1059.

Examples

DATENAME('year', #2004-04-15#) = "2004"
DATENAME('month', #2004-04-15#) = "April"

**DATEPARSE(format, string)**

Converts a string to a datetime in the specified format. Support for some locale-specific formats is determined by the computer’s system settings. Letters that appear in the data and do not need to be parsed should be surrounded by single quotes ("'"). For formats that do not have delimiters between values (for example, MMddyy), verify that they are parsed as expected. The format must be a constant string, not a field value. This function returns Null if the data does not match the format.

This function is available for several connectors. For more information, see Convert a Field to a Date Field on page 740.

Examples

DATEPARSE ("dd.MMMM.yyyy", "15.April.2004") = #April 15, 2004#
DATEPARSE ("h'h' m'm' s's'", "10h 5m 3s") = #10:05:03#

**DATEPART(date_part, date, [start_of_week])**

Returns date_part of date as an integer.

The start_of_week parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If start_of_week is omitted, the start of week is determined by the data source. See Date Properties for a Data Source on page 1059.

**Note:** When the date_part is weekday, the start_of_week parameter is ignored. This is because Tableau relies on a fixed weekday ordering to apply offsets.
Examples

DATEPART('year', #2004-04-15#) = 2004
DATEPART('month', #2004-04-15#) = 4

**DATETIME(expression)**

Returns a datetime given a number, string, or date expression.

**Example**

DATETIME("April 15, 2005 07:59:00") = April 15, 2005 07:59:00

**DATETRUNC(date_part, date, [start_of_week])**

Truncates the specified date to the accuracy specified by the date_part. This function returns a new date. For example, when you truncate a date that is in the middle of the month at the month level, this function returns the first day of the month. The start_of_week parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If start_of_week is omitted, the start of week is determined by the data source. See Date Properties for a Data Source on page 1059.

**Examples**

DATETRUNC('quarter', #2004-08-15#) = 2004-07-01 12:00:00 AM
DATETRUNC('month', #2004-04-15#) = 2004-04-01 12:00:00 AM

**DAY(date)**

Returns the day of the given date as an integer.

**Example**

DAY(#2004-04-12#) = 12
DEGREES(number)
Converts a given number in radians to degrees.

Example
DEGREES(PI( )/4) = 45.0

DIV(integer1, integer2)
Returns the integer part of a division operation, in which integer1 is divided by integer2.

Example
DIV(11,2) = 5

DOMAIN(string_url)
Note: Supported only when connected to Google BigQuery
Given a URL string, returns the domain as a string.

Example
DOMAIN('http://www.google.com:80/index.html') = 'google.com'

ENDSWITH(string, substring)
Returns true if the given string ends with the specified substring. Trailing white spaces are ignored.

Example
ENDSWITH("Tableau", "leau") = true
**EXP(number)**

Returns $e$ raised to the power of the given number.

**Examples**

EXP(2) = 7.389

EXP(-[Growth Rate]*[Time])

---

**FIND(string, substring, [start])**

Returns the index position of substring in string, or 0 if the substring isn't found. If the optional argument start is added, the function ignores any instances of substring that appear before the index position start. The first character in the string is position 1.

**Examples**

FIND("Calculation", "alcu") = 2

FIND("Calculation", "Computer") = 0

FIND("Calculation", "a", 3) = 7

FIND("Calculation", "a", 2) = 2

FIND("Calculation", "a", 8) = 0

---

**FINDNTH(string, substring, occurrence)**

Returns the position of the nth occurrence of substring within the specified string, where n is defined by the occurrence argument.

**Note:** FINDNTH is not available for all data sources.
Example

\[
\text{FINDNTH("Calculation", "a", 2) = 7}
\]

**FIRST()**

Returns the number of rows from the current row to the first row in the partition. For example, the view below shows quarterly sales. When FIRST() is computed within the Date partition, the offset of the first row from the second row is -1.

<table>
<thead>
<tr>
<th>Year of Order Date</th>
<th>Quarter of Order Date</th>
<th>Central</th>
<th>East</th>
<th>South</th>
<th>West</th>
<th>First()</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>Q1</td>
<td>$185,877</td>
<td>$231,411</td>
<td>$133,934</td>
<td>$185,961</td>
<td>$185,877</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$197,213</td>
<td>$204,914</td>
<td>$337,013</td>
<td>$213,507</td>
<td>$197,213</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$302,670</td>
<td>$185,201</td>
<td>$293,908</td>
<td>$206,512</td>
<td>$302,670</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$297,208</td>
<td>$226,983</td>
<td>$214,846</td>
<td>$230,291</td>
<td>$297,208</td>
</tr>
<tr>
<td>2010</td>
<td>Q1</td>
<td>$188,600</td>
<td>$180,123</td>
<td>$273,943</td>
<td>$261,145</td>
<td>$188,600</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$195,596</td>
<td>$224,032</td>
<td>$251,391</td>
<td>$195,976</td>
<td>$195,596</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$116,613</td>
<td>$50,363</td>
<td>$194,601</td>
<td>$102,731</td>
<td>$116,613</td>
</tr>
</tbody>
</table>

Example

When the current row index is 3, FIRST() = -2.

**FLOAT(expression)**

Casts its argument as a floating point number.

Examples

\[
\text{FLOAT(3) = 3.000}
\]

FLOAT([Age]) converts every value in the Age field to a floating point number.

**FLOOR(number)**

Rounds a number to the nearest integer of equal or lesser value.
Example

\[ \text{FLOOR}(3.1415) = 3 \]

### Availability by data source

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Access</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Supported</td>
</tr>
<tr>
<td>Text File</td>
<td>Supported</td>
</tr>
<tr>
<td>Statistical File</td>
<td>Supported</td>
</tr>
<tr>
<td>Tableau Server</td>
<td>Supported</td>
</tr>
<tr>
<td>Actian Vector</td>
<td>Not supported</td>
</tr>
<tr>
<td>Amazon Aurora</td>
<td>Not supported</td>
</tr>
<tr>
<td>Amazon EMR Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td>Not supported</td>
</tr>
<tr>
<td>Aster Database</td>
<td>Not supported</td>
</tr>
<tr>
<td>Cloudera Hadoop</td>
<td>Supported</td>
</tr>
<tr>
<td>DataStax Enterprise</td>
<td>Supported</td>
</tr>
<tr>
<td>EXASOL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Firebird</td>
<td>Not supported</td>
</tr>
<tr>
<td>Google Analytics</td>
<td>Supported</td>
</tr>
<tr>
<td>System</td>
<td>Support Status</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Google BigQuery</td>
<td>Supported</td>
</tr>
<tr>
<td>Google Cloud SQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Hortonworks Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>IBM BigInsights</td>
<td>Not supported</td>
</tr>
<tr>
<td>IBM DB2</td>
<td>Not supported</td>
</tr>
<tr>
<td>IBM Netezza</td>
<td>Not supported</td>
</tr>
<tr>
<td>MapR Hadoop Hive</td>
<td>Supported</td>
</tr>
<tr>
<td>MarkLogic</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft Analysis Services</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft PowerPivot</td>
<td>Not supported</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>Not supported</td>
</tr>
<tr>
<td>MySQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Oracle</td>
<td>Not supported</td>
</tr>
<tr>
<td>Oracle Essbase</td>
<td>Not supported</td>
</tr>
<tr>
<td>ParAccel</td>
<td>Not supported</td>
</tr>
<tr>
<td>Pivotal Greenplum</td>
<td>Not supported</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Not supported</td>
</tr>
<tr>
<td>Progress OpenEdge</td>
<td>Not supported</td>
</tr>
<tr>
<td>Salesforce</td>
<td>Supported</td>
</tr>
<tr>
<td>SAP HANA</td>
<td>Not supported</td>
</tr>
<tr>
<td>SAP Sybase ASE</td>
<td>Not supported</td>
</tr>
<tr>
<td>Database</td>
<td>Support</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>SAP Sybase IQ</td>
<td>Not supported</td>
</tr>
<tr>
<td>Spark SQL</td>
<td>Supported</td>
</tr>
<tr>
<td>Splunk</td>
<td>Not supported</td>
</tr>
<tr>
<td>Teradata</td>
<td>Not supported</td>
</tr>
<tr>
<td>Teradata OLAP Connector</td>
<td>Not supported</td>
</tr>
<tr>
<td>Vertica</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

**FULLNAME( )**

Returns the full name for the current user. This is the Tableau Server or Tableau Online full name when the user is signed in; otherwise the local or network full name for the Tableau Desktop user.

**Example**

```sql
[Manager]=FULLNAME( )
```

If manager Dave Hallsten is signed in, this example returns True only if the Manager field in the view contained Dave Hallsten. When used as a filter, this calculated field can be used to create a user filter that only shows data that is relevant to the person signed in to the server.

**GET_JSON_OBJECT(JSON string, JSON path)**

*Note: Supported only when connected to Hadoop Hive.*

Returns the JSON object within the JSON string based on the JSON path.
GROUP_CONCAT(expression)

*Note*: Supported only when connected to Google BigQuery

Concatenates values from each record into a single comma-delimited string. This function acts like a SUM() for strings.

**Example**

GROUP_CONCAT(Region) = "Central,East,West"

---

**HEXBINX(number, number)**

Maps an x, y coordinate to the x-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately.

HEXBINX and HEXBINY are binning and plotting functions for hexagonal bins. Hexagonal bins are an efficient and elegant option for visualizing data in an x/y plane such as a map. Because the bins are hexagonal, each bin closely approximates a circle and minimizes variation in the distance from the data point to the center of the bin. This makes the clustering both more accurate and informative.

**Example**

HEXBINX([Longitude], [Latitude])

---

**HEXBINY(number, number)**

Maps an x, y coordinate to the y-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately.

**Example**

HEXBINY([Longitude], [Latitude])
HOST(string_url)

*Note: Supported only when connected to Google BigQuery*

Given a URL string, returns the host name as a string.

**Example**


**IF test THEN value END / IF test THEN value ELSE else END**

Use the IF THEN ELSE function to perform logical tests and return appropriate values. The IF THEN ELSE function evaluates a sequence of test conditions and returns the value for the first condition that is true. If no condition is true, the ELSE value is returned. Each test must be a boolean: either a boolean field in the data source or the result of a logical expression. The final ELSE is optional, but if it is not provided and there is no true test expression, then the function returns Null. All of the value expressions must be of the same type.

**Examples**

IF [Cost]>[Budget Cost] THEN 'Over Budget' ELSE 'Under Budget' END

IF [Budget Sales]!=0 THEN [Sales]/[Budget Sales] END

**IF test1 THEN value1 ELSEIF test2 THEN value2 ELSE else END**

Use this version of the IF function to perform logical tests recursively. There is no built-in limit to the number of ELSEIF values you can use with an IF function, though individual databases may impose a limit on IF function complexity. While an IF function can be rewritten as a series of nested IIF statements, there are differences in how the expressions will be evaluated. In
particular, an IF statement distinguishes TRUE, FALSE and UNKNOWN, whereas an IF statement only worries about TRUE and not true (which includes both FALSE and UNKNOWN).

Example

When you create bins from a measure, Tableau creates bins of equal size by default. For example, say you have a measure that represents age. When you create bins from that measure, Tableau makes all the bins of equal size. You can specify how big you want the bins to be, but you cannot specify a separate range of values for each bin. A way around this constraint is to create a calculated field to define bins. Then you can create one bin for ages 0 - 20, another for ages 21 - 32, and so on. The following procedure shows how you could do that.

1. Create a new calculated field by choosing Analysis > Create Calculated Field.

2. Name the field Age Groups and type the following in the definition area

   IF
   
   [Age] < 21 THEN 'Under 21'
   
   ELSEIF
   
   [Age] <= 32 THEN '21-32'
   
   ELSEIF
   
   [Age] <= 42 THEN '33-42'
   
   ELSEIF
   
   [Age] <= 52 THEN '43-52'
   
   ELSEIF
   
   [Age] <= 64 THEN '53-64'
   
   ELSE '65+'

   END

   Confirm that the status message indicates that the formula is valid, and then click OK.

3. From the Measures area of the Data pane, drag Number of Records to Rows.
4. From the Dimensions area of the **Data** pane, drag **Age Groups** to **Columns**.

The records are now divided among the six bins that you defined:

![Bar chart showing age groups]

Unfortunately, the **Under 21** bin is at far right, when you would expect it to be at far left. Tableau’s smart enough to put the bins with entirely numerical names in the right order, but it can’t guess that the bin name beginning with 'Under' belongs at the left. Fix the problem with a manual sort.

5. Click the down arrow at the right side of the **Age Groups** field on **Columns** and then click **Sort**. Choose **Manual** and then move the **Under 21** bin up to the top of the list:
IIF(test, then, else, [unknown])

Use the IIF function to perform logical tests and return appropriate values. The first argument, test, must be a boolean: either a boolean field in the data source, or the result of a logical expression using operators (or a logical comparison of AND, OR, or NOT). If test evaluates to TRUE, then IIF returns the then value. If test evaluates to FALSE, then IIF returns the else value.

A boolean comparison may also yield the value UNKNOWN (neither TRUE nor FALSE), usually due to the presence of Null values in test. The final argument to IIF is returned in the event of an UNKNOWN result for the comparison. If this argument is left out, Null is returned.
Examples

IIF(7>5, 'Seven is greater than five', 'Seven is less than five')
IIF([Cost]>[Budget Cost], 'Over Budget', 'Under Budget')
IIF([Budget Sales]!=0, [Sales]/[Budget Sales], 0)
IIF(Sales>=[Budget Sales], 'Over Cost Budget and Over Sales Budget', 'Over Cost Budget and Under Sales Budget', 'Under Cost Budget')

IFNULL(expression1, expression2)

The IFNULL function returns the first expression if the result is not null, and returns the second expression if it is null.

Example

IFNULL([Profit], 0) = [Profit]

INDEX()

Returns the index of the current row in the partition, without any sorting with regard to value. The first row index starts at 1. For example, the table below shows quarterly sales. When INDEX() is computed within the Date partition, the index of each row is 1, 2, 3, 4..., etc.

Example

For the third row in the partition, INDEX() = 3.
INT(expression)

Casts its argument as an integer. For expressions, this function truncates results to the closest integer toward zero.

Examples

\[
\begin{align*}
\text{INT}(8.0/3.0) &= 2 \\
\text{INT}(4.0/1.5) &= 2 \\
\text{INT}(0.50/1.0) &= 0 \\
\text{INT}(-9.7) &= -9
\end{align*}
\]

When a string is converted to an integer it is first converted to a float and then rounded.

ISDATE(string)

The ISDATE function returns TRUE if the string argument can be converted to a date and FALSE if it cannot.

Examples

\[
\begin{align*}
\text{ISDATE('January 1, 2003')} &= \text{TRUE} \\
\text{ISDATE('Jan 1 2003')} &= \text{TRUE} \\
\text{ISDATE('1/1/03')} &= \text{TRUE} \\
\text{ISDATE('Janxx 1 2003')} &= \text{FALSE}
\end{align*}
\]

ISFULLNAME(string)

Returns true if the current user's full name matches the specified full name, or false if it does not match. This function uses the Tableau Server or Online full name when the user is signed in; otherwise it uses the local or network full name for the Tableau Desktop user.
Example

ISFULLNAME(“Dave Hallsten”)

This example returns true if Dave Hallsten is the current user, otherwise it returns false.

ISNULL(expression)

The ISNULL function returns TRUE if the expression is Null and FALSE if it is not.

Example

The following example uses ISNULL in combination with IIF to replace null values with 0's.

IIF(ISNULL([Sales]), 0, [Sales] )

ISMEMBEROF(string)

Returns true if the person currently using Tableau is a member of a group that matches the given string. If the person currently using Tableau is signed in, the group membership is determined by groups on Tableau Server or Tableau Online. If the person is not signed in, this function returns false.

Example

IF ISMEMBEROF(“Sales”) THEN “Sales” ELSE “Other” END

ISUSERNAME(string)

Returns true if the current user's username matches the specified username, or false if it does not match. This function uses the Tableau Server or Online username when the user is signed in; otherwise it uses the local or network username for the Tableau Desktop user.

Example

ISUSERNAME(“dhallsten”)

This example returns true if dhallsten is the current user; otherwise it returns false.
LAST()

Returns the number of rows from the current row to the last row in the partition. For example, the table below shows quarterly sales. When LAST() is computed within the Date partition, the offset of the last row from the second row is 5.

<table>
<thead>
<tr>
<th>Year of Order Date</th>
<th>Quarter of Order Date</th>
<th>Region</th>
<th>Central</th>
<th>East</th>
<th>South</th>
<th>West</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>Q1</td>
<td>$180,877</td>
<td>$231,411</td>
<td>$133,934</td>
<td>$185,961</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$197,213</td>
<td>$204,914</td>
<td>$337,613</td>
<td>$213,507</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$302,678</td>
<td>$165,201</td>
<td>$293,068</td>
<td>$206,512</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>$297,208</td>
<td>$228,983</td>
<td>$214,846</td>
<td>$230,291</td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td>Q1</td>
<td>$185,600</td>
<td>$180,123</td>
<td>$273,043</td>
<td>$251,145</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>$195,785</td>
<td>$224,032</td>
<td>$251,391</td>
<td>$195,976</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>$116,613</td>
<td>$50,363</td>
<td>$194,601</td>
<td>$102,731</td>
<td></td>
</tr>
</tbody>
</table>

Example

When the current row index is 3 of 7, LAST() = 4.

LEFT(string, number)

Returns the left-most number of characters in the string.

Example

LEFT("Matador", 4) = "Mata"

LEN(string)

Returns the length of the string.

Example

LEN("Matador") = 7
**LN(number)**

Returns the natural logarithm of a number. Returns *null* if number is less than or equal to 0.

**LOG(number [, base])**

Returns the logarithm of a number for the given base. If the base value is omitted, base 10 is used.

**LOG2(number)**

*Note: Supported only when connected to Google BigQuery*

Returns the logarithm base 2 of a number.

**Example**

LOG2(16) = '4.00'

**LOOKUP(expression, [offset])**

Returns the value of the expression in a target row, specified as a relative offset from the current row. Use FIRST() + n and LAST() - n as part of your offset definition for a target relative to the first/last rows in the partition. If *offset* is omitted, the row to compare to can be set on the field menu. This function returns NULL if the target row cannot be determined.

The view below shows quarterly sales. When LOOKUP (SUM(Sales), 2) is computed within the Date partition, each row shows the sales value from 2 quarters into the future.
Example

```
LOOKUP(SUM([Profit]), FIRST()+2) computes the SUM(Profit) in the third row of the partition.
```

**LOWER(string)**

Returns `string`, with all characters lowercase.

Example

```
LOWER("ProductVersion") = "productversion"
```

**LTRIM(string)**

Returns the string with any leading spaces removed.
Example

LTRIM(" Matador ") = "Matador "

LTRIM THIS(string, string)

*Note: Supported only when connected to Google BigQuery*

Returns the first string with any leading occurrence of the second string removed.

Example

LTRIM THIS('[-Sales-]','[-]') = 'Sales-

MAKEDATE(year, month, day)

Returns a date value constructed from the specified year, month, and date.

Available for Tableau Data Extracts. Check for availability in other data sources.

Example

MAKEDATE(2004, 4, 15) = #April 15, 2004#

MAKEDATETIME(date, time)

Returns a datetime that combines a date and a time. The date can be a date, datetime, or a string type. The time must be a datetime. This function is available only for MySQL-compatible connections (which for Tableau are, in addition to MySQL, Amazon Aurora and Amazon Aurora).

Examples

MAKEDATETIME("1899-12-30", #07:59:00#) = #12/30/1899 7:59:00 AM#

MAKEDATETIME([Date], [Time]) = #1/1/2001 6:00:00 AM#
MAKETIME(hour, minute, second)

Returns a date value constructed from the specified hour, minute, and second.
Available for Tableau Data Extracts. Check for availability in other data sources.

Example

MAKETIME(14, 52, 40) = #14:52:40#

MAX(a, b)

Returns the maximum of a and b (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, MAX finds the value that is highest in the sort sequence defined by the database for that column. It returns Null if either argument is Null.

Example

MAX ("Apple","Banana") = "Banana"

MAX(expression) or MAX(expr1, expr2)

Usually applied to numbers but also works on dates. Returns the maximum of a and b (a and b must be of the same type). Returns Null if either argument is Null.

Examples

MAX(#2004-01-01#, #2004-03-01#) = 2004-03-01 12:00:00 AM
MAX([ShipDate1], [ShipDate2])

MAX(number, number)

Returns the maximum of the two arguments, which must be of the same type. Returns Null if either argument is Null. MAX can also be applied to a single field in an aggregate calculation.
Examples

\[ \text{MAX}(4, 7) \]
\[ \text{MAX}(\text{Sales}, \text{Profit}) \]
\[ \text{MAX}([\text{First Name}], [\text{Last Name}]) \]

**MEDIAN(expression)**

Returns the median of an expression across all records. Median can only be used with numeric fields. Null values are ignored. This function is not available for workbooks created before Tableau Desktop 8.2 or that use legacy connections. It is also not available for connections using any of the following data sources:

- Access
- Amazon Redshift
- Cloudera Hadoop
- IBM DB2
- IBM PDA (Netezza)
- Microsoft SQL Server
- MySQL
- SAP HANA
- Teradata
- Vertica

For other data source types, you can extract your data into an extract file to use this function. See [Extract Your Data](#) on page 773.

**MID(string, start, [length])**

Returns the string starting at index position \( \text{start} \). The first character in the string is position 1. If the optional argument \( \text{length} \) is added, the returned string includes only that number of characters.

Examples

\[ \text{MID}("\text{Calculation}", 2) = "\text{alculation}" \]
\[ \text{MID}("\text{Calculation}", 2, 5) = "\text{alcu}" \]
**MIN(a, b)**

Returns the minimum of a and b (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, MIN finds the value that is lowest in the sort sequence. It returns **Null** if either argument is **Null**.

**Example**

\[
\text{MIN} ("Apple","Banana") = "Apple"
\]

**MIN(expression) or MIN(expr1, expr2)**

Usually applied to numbers but also works on dates. Returns the minimum of a and b (a and b must be of the same type). Returns **Null** if either argument is **Null**.

**Examples**

\[
\text{MIN} (\#2004-01-01#,\#2004-03-01#) = 2004-01-01 12:00:00 AM
\]

\[
\text{MIN} ([\text{ShipDate1}], [\text{ShipDate2}])
\]

**MIN(number, number)**

Returns the minimum of the two arguments, which must be of the same type. Returns **Null** if either argument is **Null**. MIN can also be applied to a single field in an aggregate calculation.

**Examples**

\[
\text{MIN}(4,7)
\]

\[
\text{MIN(}\text{Sales},\text{Profit}\text{)}
\]

\[
\text{MIN(}[\text{First Name}],[\text{Last Name}])
\]

**MONTH(date)**

Returns the month of the given date as an integer.
Example

MONTH(#2004-04-15#) = 4

---

**NOW( )**

Returns the current date and time.

The return varies depending on the nature of the connection:

- For a live, unpublished connection, NOW returns the data source server time.
- For a live, published connection, NOW returns the data source server time.
- For an unpublished extract, NOW returns the local system time.
- For a published extract, NOW returns the local time of the Tableau Server Data Engine. When there are multiple worker machines indifferent time zones, this can produce inconsistent results.

Example

NOW( ) = 2004-04-15 1:08:21 PM

---

**PARSE_URL(string, url_part)**

*Note: Supported only when connected to Hadoop Hive and Cloudera Impala.*

Returns a component of the given URL string where the component is defined by url_part. Valid url_part values include: 'HOST', 'PATH', 'QUERY', 'REF', 'PROTOCOL', 'AUTHORITY', 'FILE' and 'USERINFO'.

Example

PARSE_URL('http://www.tableau.com', 'HOST') = 'www.tableau.com'
PARSE_URL_QUERY(string, key)

**Note:** Supported only when connected to Hadoop Hive and Cloudera Impala.

Returns the value of the specified query parameter in the given URL string. The query parameter is defined by the key.

**Example**

PARSE_URL_QUERY('http://www.tableau.com?page=1&cat=4', 'page') = '1'

PERCENTILE(expression, number)

Returns the percentile value from the given expression corresponding to the specified number. The number must be between 0 and 1 (inclusive)—for example, 0.66, and must be a numeric constant.

This function is available for the following data sources.

- Non-legacy Microsoft Excel and Text File connections.
- Extracts and extract-only data source types (for example, Google Analytics, OData, or Salesforce).
- Sybase IQ 15.1 and later data sources.
- Oracle 10 and later data sources.
- Cloudera Hive and Hortonworks Hadoop Hive data sources.
- EXASOL 4.2 and later data sources.

For other data source types, you can extract your data into an extract file to use this function. See **Extract Your Data** on page 773.

PI()

Returns the numeric constant pi: 3.14159.
POWER(number, power)

Raises the number to the specified power.

Examples

POWER(5, 2) = 5^2 = 25
POWER(Temperature, 2)

You can also use the ^ symbol:
5^2 = POWER(5, 2) = 25

PREVIOUS_VALUE(expression)

Returns the value of this calculation in the previous row. Returns the given expression if the current row is the first row of the partition.

Example

SUM([Profit]) * PREVIOUS_VALUE(1) computes the running product of SUM(Profit).

RADIANS(number)

Converts the given number from degrees to radians.

Example

RADIANS(180) = 3.14159

RANK(expression, ['asc' | 'desc'])

Returns the standard competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.
With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.

Example

The following image shows the effect of the various ranking functions (RANK, RANK_DENSE, RANK_MODIFIED, RANK_PERCENTILE, and RANK_UNIQUE) on a set of values. The data set contains information on 14 students (StudentA through StudentN); the Age column shows the current age of each student (all students are between 17 and 20 years of age). The remaining columns show the effect of each rank function on the set of age values, always assuming the default order (ascending or descending) for the function.

RANK_DENSE(expression, ['asc' | 'desc'])

Returns the dense rank for the current row in the partition. Identical values are assigned an identical rank, but no gaps are inserted into the number sequence. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (3, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.
RANK_MODIFIED(expression, ['asc' | 'desc'])

Returns the modified competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 3, 3, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.

RANK_PERCENTILE(expression, ['asc' | 'desc'])

Returns the percentile rank for the current row in the partition. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is ascending.

With this function, the set of values (6, 9, 9, 14) would be ranked (0.25, 0.75, 0.75, 1.00).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.

RANK_UNIQUE(expression, ['asc' | 'desc'])

Returns the unique rank for the current row in the partition. Identical values are assigned different ranks. Use the optional 'asc' | 'desc' argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 3, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see Rank calculation on page 1549.
RAWSQL_BOOL("sql_expr", [arg1], ...[argN])

Returns a Boolean result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example
In the example, %1 is equal to [Sales] and %2 is equal to [Profit].
RAWSQL_BOOL("IIF ( %1 > %2, True, False)", [Sales], [Profit])

RAWSQL_DATE("sql_expr", [arg1], ...[argN])

Returns a Date result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example
In this example, %1 is equal to [Order Date].
RAWSQL_DATE("%1", [Order Date])

RAWSQL_DATETIME("sql_expr", [arg1], ... [argN])

Returns a Date and Time result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].

Example
RAWSQL_DATETIME("MIN(%1)", [Delivery Date])
RAWSQL_INT("sql_expr", [arg1], ...[argN])

Returns an integer result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example
RAWSQL_INT("500 + %1", [Sales])

RAWSQL_REAL("sql_expr", [arg1], ...[argN])

Returns a numeric result from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]

Example
RAWSQL_REAL("-123.98 * %1", [Sales])

RAWSQL_SPATIAL

Returns a Spatial from a given SQL expression that is passed directly to the underlying data source. Use %n in the SQL expression as a substitution syntax for database values.

Example
In this example, %1 is equal to [Geometry].

RAWSQL_SPATIAL("%1", [Geometry])

RAWSQL_STR("sql_expr", [arg1], ...[argN])

Returns a string from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Customer Name].
Example

`RAWSQL_STR(“%1”, [Customer Name])`

`RAWSQLAGG_BOOL(“sql_expr”, [arg1], … [argN])`

Returns a Boolean result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In the example, %1 is equal to [Sales] and %2 is equal to [Profit].

`RAWSQLAGG_BOOL(“SUM( %1) >SUM( %2)”, [Sales], [Profit])`

`RAWSQLAGG_DATE(“sql_expr”, [arg1], … [argN])`

Returns a Date result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Order Date].

Example

`RAWSQLAGG_DATE(“MAX(%1)”, [Order Date])`

`RAWSQLAGG_DATETIME(“sql_expr”, [arg1], … [argN])`

Returns a Date and Time result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].
Example

RAWSQLAGG_DATETIME("MIN(%1)", [Delivery Date])

RAWSQLAGG_INT("sql_expr", [arg1,] ...[argN])

Returns an integer result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example

RAWSQLAGG_INT("500 + SUM(%1)", [Sales])

RAWSQLAGG_REAL("sql_expr", [arg1,] ...[argN])

Returns a numeric result from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]

Example

RAWSQLAGG_REAL("SUM( %1)", [Sales])

RAWSQLAGG_STR("sql_expr", [arg1,] ...[argN])

Returns a string from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Discount].

Example

RAWSQLAGG_STR("AVG( %1)", [Discount])
REGEXP_REPLACE(string, pattern, replacement)

Returns a copy of the given string where the regular expression pattern is replaced by the replacement string. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern and the replacement must be constants.

For information on regular expression syntax, see your data source’s documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_REPLACE('abc 123', '\s', '-') = 'abc-123'

REGEXP_MATCH(string, pattern)

Returns true if a substring of the specified string matches the regular expression pattern. This function is available for Text File, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Impala 2.3.0 (through Cloudera Hadoop data sources), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source’s documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_MATCH('-(\([1234].\[The.Market]\)\-\', '\s*(\w*\.)(\w*\s*)\'}=true
**REGEXP_EXTRACT***(string, pattern)***

Returns the portion of the string that matches the regular expression pattern. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

**Example**

REGEXP_EXTRACT('abc 123', '[a-z]+\s+(\d+)') = '123'

**REGEXP_EXTRACT_NTH***(string, pattern, index)***

Returns the portion of the string that matches the regular expression pattern. The substring is matched to the nth capturing group, where n is the given index. If index is 0, the entire string is returned. This function is available for Text File, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.
Example
REGEXP_EXTRACT_NTH('abc 123', '([a-z]+)\s+(\d+)', 2) = '123'

REPLACE(string, substring, replacement)

Searches string for substring and replaces it with replacement. If substring is not found, the string is not changed.

Example
REPLACE("Version8.5", "8.5", "9.0") = "Version9.0"

RIGHT(string, number)

Returns the right-most number of characters in string.

Example
RIGHT("Calculation", 4) = "tion"

ROUND(number, [decimals])

Rounds numbers to a specified number of digits. The decimals argument specifies how many decimal points of precision to include in the final result. If decimals is omitted, number is rounded to the nearest integer.

Example
This example rounds every Sales value to an integer:

ROUND(Sales)

Some databases, such as SQL Server, allow specification of a negative length, where -1 rounds number to 10's, -2 rounds to 100's, and so on. This is not true of all databases. For example, it is not true of Excel or Access.
RTRIM(string)

Returns string with any trailing spaces removed.

Example

RTRIM(" Calculation ") = " Calculation"

RTRIM_THIS(string, string)

Note: Supported only when connected to Google BigQuery

Returns the first string with any trailing occurrence of the second string removed.

Example

RTRIM_THIS('[-Market-],'-'] = '[-Market'

RUNNING_AVG(expression)

Returns the running average of the given expression, from the first row in the partition to the current row.

The view below shows quarterly sales. When RUNNIMG_AVG(SUM([Sales])) is computed within the Date partition, the result is a running average of the sales values for each quarter.
Example

RUNNING_AVG(SUM([Profit])) computes the running average of SUM(Profit).

RUNNING_COUNT(expression)

Returns the running count of the given expression, from the first row in the partition to the current row.

Example

RUNNING_COUNT(SUM([Profit])) computes the running count of SUM(Profit).

RUNNING_MAX(expression)

Returns the running maximum of the given expression, from the first row in the partition to the current row.
Example

RUNNING_MAX(SUM([Profit])) computes the running maximum of SUM(Profit).

RUNNING_MIN(expression)

Returns the running minimum of the given expression, from the first row in the partition to the current row.
Example

RUNNING_MIN(SUM([Profit])) computes the running minimum of SUM(Profit).

RUNNING_SUM(expression)

Returns the running sum of the given expression, from the first row in the partition to the current row.

Example

RUNNING_SUM(SUM([Profit])) computes the running sum of SUM(Profit)

SCRIPT_BOOL

Returns a Boolean result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use .argn (with a leading period) to reference parameters (.arg1, .arg2, etc.).

In Python expressions, use _argn (with a leading underscore).
Examples

In this R example, `.arg1` is equal to `SUM([Profit])`:

```r
SCRIPT_BOOL("is.finite(.arg1)", SUM([Profit]))
```

The next example returns `True` for store IDs in Washington state, and `False` otherwise. This example could be the definition for a calculated field titled `IsStoreInWA`.

```r
SCRIPT_BOOL('grepl(".*_WA", .arg1, perl=TRUE)',ATTR([Store ID]))
```

A command for Python would take this form:

```py
SCRIPT_BOOL("return map(lambda x : x > 0, _arg1)", SUM([Profit]))
```

**SCRIPT_INT**

Returns an integer result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use `.argn` (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use `_argn` (with a leading underscore).

Examples

In this R example, `.arg1` is equal to `SUM([Profit])`:

```r
SCRIPT_INT("is.finite(.arg1)", SUM([Profit]))
```

In the next example, k-means clustering is used to create three clusters:

```r
SCRIPT_INT('result <- kmeans(data.frame(.arg1,.arg2,.arg3,.arg4),
3);result$cluster;', SUM([Petal length]), SUM([Petal width]),SUM ([Sepal length]),SUM([Sepal width]))
```

A command for Python would take this form:

```py
SCRIPT_INT("return map(lambda x : int(x * 5), _arg1)", SUM ([Profit]))
```
SCRIPT_REAL

Returns a real result from the specified expression. The expression is passed directly to a running external service instance. In R expressions, use `.argn` (with a leading period) to reference parameters (.arg1, .arg2, etc.) In Python expressions, use `_argn` (with a leading underscore).

Examples

In this R example, `.arg1` is equal to `SUM([Profit])`:

```
SCRIPT_REAL("is.finite(.arg1)", SUM([Profit]))
```

The next example converts temperature values from Celsius to Fahrenheit.

```
SCRIPT_REAL('library(udunits2);ud.convert(.arg1, "celsius", "degree_fahrenheit")', AVG([Temperature]))
```

A command for Python would take this form:

```
SCRIPT_REAL("return map(lambda x : x * 0.5, _arg1)", SUM ([Profit]))
```

SCRIPT_STR

Returns a string result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use `.argn` (with a leading period) to reference parameters (.arg1, .arg2, etc.) In Python expressions, use `_argn` (with a leading underscore).

Examples

In this R example, `.arg1` is equal to `SUM([Profit])`:

```
SCRIPT_STR("is.finite(.arg1)", SUM([Profit]))
```

The next example extracts a state abbreviation from a more complicated string (in the original form 13XSL_CA, A13_WA):

```
SCRIPT_STR('gsub(".* ", ",", .arg1)', ATTR([Store ID]))
```
A command for Python would take this form:

```
SCRIPT_STR("return map(lambda x : x[:2], _arg1)", ATTR([Region]))
```

**SIGN(number)**

Returns the sign of a number: The possible return values are -1 if the number is negative, 0 if the number is zero, or 1 if the number is positive.

**Example**

If the average of the profit field is negative, then

```
SIGN(AVG(Profit)) = -1
```

**SIN(number)**

Returns the sine of an angle. Specify the angle in radians.

**Example**

```
SIN(0) = 1.0
SIN(PI()/4) = 0.707106781186548
```

**SIZE()**

Returns the number of rows in the partition. For example, the view below shows quarterly sales. Within the Date partition, there are seven rows so the Size() of the Date partition is 7.
Example

SIZE() = 5 when the current partition contains five rows.

**SPACE(number)**

Returns a string that is composed of the specified number of repeated spaces.

Example

SPACE(1) = " "

**SPLIT(string, delimiter, token number)**

Returns a substring from a string, using a delimiter character to divide the string into a sequence of tokens.

The string is interpreted as an alternating sequence of delimiters and tokens. So for the string abc-defgh-i-jkl, where the delimiter character is ‘-’, the tokens are abc, defgh, i, and jlk. Think of these as tokens 1 through 4. SPLIT returns the token corresponding to the token number. When the token number is positive, tokens are counted starting from the left end of the string; when the token number is negative, tokens are counted starting from the right.

Examples

SPLIT (‘a-b-c-d’, ‘-’, 2) = ‘b’
SPLIT (‘a|b|c|d’, ‘|’, -2) = ‘c’

**Note:** The split and custom split commands are available for the following data sources types: Tableau data extracts, Microsoft Excel, Text File, PDF File, Salesforce, OData, Microsoft Azure Market Place, Google Analytics, Vertica, Oracle, MySQL, PostgreSQL, Teradata, Amazon Redshift, Aster Data, Google Big Query, Cloudera Hadoop Hive, Hortonworks Hive, and Microsoft SQL Server.

Some data sources impose limits on splitting string. The following table shows which data sources support negative token numbers (splitting from the right) and whether there is a limit on the number of splits allowed per data source. A SPLIT function that specifies a negative token,
number and would be legal with other data sources will return this error with these data sources: “Splitting from right is not support by the data source.”

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Left/Right Constraints</th>
<th>Maximum Number of Splits</th>
<th>Version Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tableau Data Extract</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Text file</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Salesforce</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>OData</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Google Analytics</td>
<td>Both</td>
<td>Infinite</td>
<td></td>
</tr>
<tr>
<td>Tableau Data Server</td>
<td>Both</td>
<td>Infinite</td>
<td>Supported in version 9.0.</td>
</tr>
<tr>
<td>Vertica</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>MySQL</td>
<td>Both</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Left only prior to version 9.0; both for version 9.0 and above</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Teradata</td>
<td>Left only</td>
<td>10</td>
<td>Version 14 and later</td>
</tr>
<tr>
<td>Amazon Red-shift</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>Support</td>
<td>Min Version</td>
<td></td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>Aster Database</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Google BigQuery</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Hortonworks Hadoop Hive</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Cloudera Hadoop</td>
<td>Left only</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>Both</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2008 and later</td>
<td></td>
</tr>
</tbody>
</table>

**SQRT(number)**

Returns the square root of a number.

Example

\[
SQRT(25) = 5
\]

**SQUARE(number)**

Returns the square of a number.

Example

\[
SQUARE(5) = 25
\]

**STARTSWITH(string, substring)**

Returns true if `string` starts with `substring`. Leading white spaces are ignored.
Example
```
STARTSWITH("Joker", "Jo") = true
```

**STDEV(expression)**

Returns the statistical standard deviation of all values in the given expression based on a sample of the population.

**STDEVP(expression)**

Returns the statistical standard deviation of all values in the given expression based on a biased population.

**STR(expression)**

Casts its argument as a string.

Example
```
STR([Age]) takes all of the values in the measure called Age and converts them to strings.
```

**SUM(expression)**

Returns the sum of all values in the expression. SUM can be used with numeric fields only. Null values are ignored.

**TAN(number)**

Returns the tangent of an angle. Specify the angle in radians.
Example

\[ \tan\left(\frac{\pi}{4}\right) = 1.0 \]

**TIMESTAMP_TO_USEC(expression)**

*Note: Supported only when connected to Google BigQuery*

Converts a TIMESTAMP data type to a UNIX timestamp in microseconds.

Example

\[ \text{TIMESTAMP\_TO\_USEC} \left(\#2012-10-01 01:02:03\right) = 1349053323000000 \]

**TLD(string_url)**

*Note: Supported only when connected to Google BigQuery*

Given a URL string, returns the top level domain plus any country domain in the URL.

Example

\[ \text{TLD('http://www.google.com:80/index.html')} = \text{'.com'} \]
\[ \text{TLD('http://www.google.co.uk:80/index.html')} = \text{'.co.uk'} \]

**TODAY()**

Returns the current date.

Example

\[ \text{TODAY()} = 2004-04-15 \]

**TOTAL(expression)**

Returns the total for the given expression in a table calculation partition.
Example

Assume you are starting with this view:

You open the calculation editor and create a new field which you name **Totality**:

```
TOTAL(SUM([Sales]))
```

You then drop **Totality** on Text, to replace **SUM(Sales)**. Your view changes such that it sums values based on the default **Compute Using** value:
This raises the question, What is the default **Compute Using** value? If you right-click (Control-click on a Mac) **Totality** in the Data pane and choose **Edit**, there is now an additional bit of information available:

```
TOTAL(SUM([Sales]))
```

The default **Compute Using** value is **Table (Across)**. The result is that **Totality** is summing the values across each row of your table. Thus, the value that you see across each row is the sum of the values from the original version of the table.

The values in the 2011/Q1 row in the original table were $8601, $6579, $44262, and $15006. The values in the table after **Totality** replaces **SUM(Sales)** are all $74,448, which is the sum of the four original values.

Notice the triangle next to Totality after you drop it on Text:
This indicates that this field is using a table calculation. You can right-click the field and choose Edit Table Calculation to redirect your function to a different Compute Using value. For example, you could set it to Table (Down). In that case, your table would look like this:

```
TRIM(string)
Returns the string with leading and trailing spaces removed. For example, TRIM("Calculation") = "Calculation"
```

Back to Top
UPPER(string)

Returns string, with all characters uppercase.

Example

UPPER("Calculation") = "CALCULATION"

USEC_TO_TIMESTAMP(expression)

Note: Supported only when connected to Google BigQuery

Converts a UNIX timestamp in microsseconds to a TIMESTAMP data type.

Example

USEC_TO_TIMESTAMP(1349053323000000) = #2012-10-01 01:02:03#

USERDOMAIN()

Returns the domain for the current user when the user is signed on to Tableau Server. Returns the Windows domain if the Tableau Desktop user is on a domain. Otherwise this function returns a null string.

Example

[Manager] = USERNAME() AND [Domain] = USERDOMAIN()

USERNAME()

Returns the username for the current user. This is the Tableau Server or Tableau Online username when the user is signed in; otherwise it is the local or network username for the Tableau Desktop user.

Example

[Manager] = USERNAME()
If the manager dhallsten was signed in, this function would only return True when the Manager field in the view is dhallsten. When used as a filter this calculated field can be used to create a user filter that only shows data that is relevant to the person signed in to the server.

VAR(expression)

Returns the statistical variance of all values in the given expression based on a sample of the population.

VPARP(expression)

Returns the statistical variance of all values in the given expression on the entire population.

WINDOW_AVG(expression, [start, end])

Returns the average of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window average within the Date partition returns the average sales across all dates.
Example

`WINDOW_AVG(SUM([Profit]), FIRST()+1, 0)` computes the average of `SUM(Profit)` from the second row to the current row.

**WINDOW_CORR(expression1, expression2, [start, end])**

Returns the Pearson correlation coefficient of two expressions within the window. The window is defined as offsets from the current row. Use `FIRST()+n` and `LAST()~-n` for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.

The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and -1 is an exact negative relationship.

There is an equivalent aggregation function: `CORR`.

Example

The following formula returns the Pearson correlation of `SUM(Profit)` and `SUM(Sales)` from the five previous rows to the current row.

`WINDOW_CORR(SUM([Profit]), SUM([Sales]), -5, 0)`
**WINDOW_COUNT(expression, [start, end])**

Returns the count of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

**Example**

```sql
WINDOW_COUNT(SUM([Profit]), FIRST()+1, 0) computes the count of SUM(Profit) from the second row to the current row
```

**WINDOW_COVAR(expression1, expression2, [start, end])**

Returns the *sample covariance* of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end arguments are omitted, the window is the entire partition.

Sample covariance uses the number of non-null data points n - 1 to normalize the covariance calculation, rather than n, which is used by the population covariance (with the WINDOW_COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population.

There is an equivalent aggregation function: COVAR.

**Example**

The following formula returns the sample covariance of `SUM(Profit)` and `SUM(Sales)` from the two previous rows to the current row.

```sql
WINDOW_COVAR(SUM([Profit]), SUM([Sales]), -2, 0)
```
WINDOW_COVARP(expression1, expression2, [start, end])

Returns the population covariance of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.

Population covariance is sample covariance multiplied by \((n-1)/n\), where \(n\) is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the WINDOW_COVAR function) is appropriate.

There is an equivalent aggregation function: COVARP. See Tableau Functions (Alphabetical) on page 1449.

Example

The following formula returns the population covariance of SUM(Profit) and SUM(Sales) from the two previous rows to the current row.

WINDOW_COVARP(SUM([Profit]), SUM([Sales]), -2, 0)

WINDOW_MEDIAN(expression, [start, end])

Returns the median of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly profit. A window median within the Date partition returns the median profit across all dates.
Example

\[ \text{WINDOW\_MEDIAN} \left( \text{SUM}([\text{Profit}]), \text{FIRST}()+1, 0 \right) \]
computes the median of \( \text{SUM} \) of \( \text{Profit} \) from the second row to the current row.

\[ \text{WINDOW\_MAX} \left( \text{expression}, [\text{start, end}] \right) \]

Returns the maximum of the expression within the window. The window is defined by means of offsets from the current row. Use \( \text{FIRST}()+n \) and \( \text{LAST}()-n \) for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window maximum within the Date partition returns the maximum sales across all dates.
Example

WINDOW_MAX(SUM([Profit]), FIRST()+1, 0) computes the maximum of SUM (Profit) from the second row to the current row.

WINDOW_MIN(expression, [start, end])

Returns the minimum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window minimum within the Date partition returns the minimum sales across all dates.
Example

WINDOW_MIN(SUM([Profit]), FIRST()+1, 0) computes the minimum of SUM(Profit) from the second row to the current row.

**WINDOW_PERCENTILE(expression, number, [start, end])**

Returns the value corresponding to the specified percentile within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

WINDOW_PERCENTILE(SUM([Profit]), 0.75, -2, 0) returns the 75th percentile for SUM(Profit) from the two previous rows to the current row.

**WINDOW_STDEV(expression, [start, end])**

Returns the sample standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.
Example

$\text{WINDOW\_STDEV}(\text{SUM}([\text{Profit}]), \text{FIRST()}+1, 0)$ computes the standard deviation of $\text{SUM(Profit)}$ from the second row to the current row.

**WINDOW\_STDEVP(expression, [start, end])**

Returns the biased standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use $\text{FIRST()}+n$ and $\text{LAST()}-n$ for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

$\text{WINDOW\_STDEVP}(\text{SUM}([\text{Profit}]), \text{FIRST()}+1, 0)$ computes the standard deviation of $\text{SUM(Profit)}$ from the second row to the current row.

**WINDOW\_SUM(expression, [start, end])**

Returns the sum of the expression within the window. The window is defined by means of offsets from the current row. Use $\text{FIRST()}+n$ and $\text{LAST()}-n$ for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window sum computed within the Date partition returns the summation of sales across all quarters.
Example

`WINDOW_SUM(SUM([Profit]), FIRST()+1, 0)` computes the sum of `SUM(Profit)` from the second row to the current row.

**WINDOW_VAR(expression, [start, end])**

Returns the sample variance of the expression within the window. The window is defined by means of offsets from the current row. Use `FIRST()+n` and `LAST()-n` for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_VAR((SUM([Profit])), FIRST()+1, 0)` computes the variance of `SUM(Profit)` from the second row to the current row.

**WINDOW_VARP(expression, [start, end])**

Returns the biased variance of the expression within the window. The window is defined by means of offsets from the current row. Use `FIRST()+n` and `LAST()-n` for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_VARP(SUM([Profit]), FIRST()+1, 0)` computes the variance of `SUM(Profit)` from the second row to the current row.

**XPATH_BOOLEAN(XML string, XPath expression string)**

*Note: Supported only when connected to Hadoop Hive*

Returns true if the XPath expression matches a node or evaluates to true.
Example

XPATH_BOOLEAN(''<values> <value id="0">1</value><value id="1">5</value>'', 'values/value [@id="1"] = 5') = true

XPATH_DOUBLE(XML string, XPath expression string)

*Note:* Supported only when connected to Hadoop Hive

Returns the floating-point value of the XPath expression.

Example

XPATH_DOUBLE(''<values><value>1.0</value><value>5.5</value></values>'', 'sum(value/*)') = 6.5

XPATH_FLOAT(XML string, XPath expression string)

*Note:* Supported only when connected to Hadoop Hive

Returns the floating-point value of the XPath expression.

Example

XPATH_FLOAT(''<values><value>1.0</value><value>5.5</value></values>'', 'sum(value/*)') = 6.5

XPATH_INT(XML string, XPath expression string)

*Note:* Supported only when connected to Hadoop Hive
Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

\[
\text{XPATH\_INT(''<values><value>1</value><value>5</value> </values>', 'sum(value/*)')} = 6
\]

**XPATH\_LONG(XML string, XPath expression string)**

*Note: Supported only when connected to Hadoop Hive*

Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

\[
\text{XPATH\_LONG(''<values><value>1</value><value>5</value> </values>', 'sum(value/*)')} = 6
\]

**XPATH\_SHORT(XML string, XPath expression string)**

*Note: Supported only when connected to Hadoop Hive*

Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

\[
\text{XPATH\_SHORT(''<values><value>1</value><value>5</value> </values>', 'sum(value/*)')} = 6
\]

**XPATH\_STRING(XML string, XPath expression string)**

*Note: Supported only when connected to Hadoop Hive*
Returns the text of the first matching node.

Example

XPATH_STRING(<sites><url domain="org">http://www.w3.org</url> <url domain="com">http://www.tableau.com</url></sites>, 'sites/url[@domain="com"]') = 'http://www.tableau.com'

YEAR (date)

Returns the year of the given date as an integer.

Example

YEAR(#2004-04-15#) = 2004

ZN(expression)

Returns the expression if it is not null, otherwise returns zero. Use this function to use zero values instead of null values.

Example

ZN([Profit]) = [Profit]

Want to learn more about functions?

Read the functions topics.
See Also

Tableau Functions (by Category) on page 1380

Functions in Tableau on page 1275

Transform Values with Table Calculations

This article explains the basics of table calculations and how to create them in Tableau.

In this article

- **What is a table calculation?** below
- The basics: addressing and partitioning on the next page
- Create a table calculation on page 1534
- Edit a table calculation on page 1536
- Remove a table calculation on page 1536

What is a table calculation?

For a 4 minute introduction to table calculations in Tableau, click this video link.

For related resources and videos, see Intro to Tableau Calculations. Use your tableau.com account to sign in.

A table calculation is a transformation you apply to the values in a visualization. Table calculations are a special type of calculated field that computes on the local data in Tableau. They are calculated based on what is currently in the visualization and do not consider any measures or dimensions that are filtered out of the visualization.

You can use table calculations for a variety of purposes, including:

- Transforming values to rankings
- Transforming values to show running totals
- Transforming values to show percent of total
For any Tableau visualization, there is a virtual table that is determined by the dimensions in the view. This table is not the same as the tables in your data source. Specifically, the virtual table is determined by the dimensions within the “level of detail,” which means the dimensions on any of the following shelves or cards in a Tableau worksheet:

The basics: addressing and partitioning

When you add a table calculation, you must use all dimensions in the level of detail either for partitioning (scoping) or for addressing (direction).

The dimensions that define how to group the calculation (the scope of data it is performed on) are called **partitioning fields**. The table calculation is performed separately within each partition.

The remaining dimensions, upon which the table calculation is performed, are called **addressing fields**, and determine the direction of the calculation.

Partitioning fields break the view up into multiple sub-views (or sub-tables), and then the table calculation is applied to the marks within each such partition. The direction in which the calculation moves (for example, in calculating a running sum, or computing the difference between values) is determined by the addressing fields. So when you order the fields in the Specific Dimensions section of the Table Calculation dialog box from top to bottom, you are
specifying the direction in which the calculation moves through the various marks in the partition.

When you add a table calculation using the Compute Using options, Tableau identifies some dimensions as addressing and others as partitioning automatically, as a result of your selections. But when you use Specific Dimensions, then it’s up to you to determine which dimensions are for addressing and which for partitioning.

**Table (across)**

Computes across the length of the table and restarts after every partition.

For example, in the following table, the calculation is computed across columns (YEAR(Order Date)) for every row (MONTH(Order Date)).

![Table (across)](image)

**Table (down)**

Computes down the length of the table and restarts after every partition.

For example, in the following table, the calculation is computed down rows (MONTH(Order Date)) for every column (YEAR(Order Date)).
Table (across then down)

Computes across the length of the table, and then down the length of the table.

For example, in the following table, the calculation is computed across columns (YEAR(Order Date)), down a row (MONTH(Order Date)), and then across columns again for the entire table.
Table (down then across)

Computes down the length of the table, and then across the length of the table.

For example, in the following table, the calculation is computed down rows (MONTH(Order Date)), across a column (YEAR(Order Date)), and then down rows again.
Pane (down)

Computes down an entire pane.

For example, in the following table, the calculation is computed down rows (MONTH(Order Date)) for a single pane.
Pane (across then down)

 Computes across an entire pane and then down the pane.

For example, in the following table, the calculation is computed across columns (YEAR(Order Date)) for the length of the pane, down a row (MONTH(Order Date)), and then across columns for the length of the pane again.

![Table](image)

Pane (down then across)

 Computes down an entire pane and then across the pane.
For example, in the following table, the calculation is computed down rows ($\text{MONTH(Order Date)}$) for the length of the pane, across a column ($\text{YEAR(Order Date)}$), and then down the length of the pane again.

```
<table>
<thead>
<tr>
<th>Quarter of Order</th>
<th>Month of Order</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td>-179.517</td>
<td>-1.0924</td>
<td>-1.483</td>
<td></td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>-$943.36</td>
<td>-$59,63</td>
<td>-$49,25</td>
<td>-$24,420</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>$50,130</td>
<td>$29,255</td>
<td>$28,013</td>
<td>$13,255</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>-$400</td>
<td>$14,041</td>
<td>$682</td>
<td></td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>-$4,647</td>
<td>-$4,064</td>
<td>$17,442</td>
<td>$5,339</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>$10,947</td>
<td>-$5,334</td>
<td>-$17,261</td>
<td>$2,609</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>-$53,012</td>
<td>-$26,155</td>
<td>-$24,480</td>
<td></td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>-$6,037</td>
<td>$8,133</td>
<td>-$5,175</td>
<td>$13,088</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$53,868</td>
<td>$27,698</td>
<td>$39,643</td>
<td>$29,973</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>-$336,641</td>
<td>-$15,455</td>
<td>-$19,444</td>
<td></td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$47,175</td>
<td>$44,568</td>
<td>$23,729</td>
<td>$34,533</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>-$9,083</td>
<td>-$11,053</td>
<td>$10,045</td>
<td>-$21,852</td>
</tr>
</tbody>
</table>
```

**Cell**

Computes within a single cell.
Specific Dimensions

Computes only within the dimensions you specify.

For example, in the following visualization the dimensions, Month of Order Date and Quarter of Order Date, are the addressing fields (since they are selected), and Year of Order Date is the partitioning field (since it is not selected). So the calculation transforms the difference from each month across all quarters within a year. The calculation starts over for every year.

Note that if all dimensions are selected, then the entire table is in scope.
At the level

The At the level option is only available when you select Specific Dimensions in the Table Calculations dialog box, and when more than one dimension is selected in the field immediately below the Compute Using options—that is, when more than one dimension is defined as an addressing field.

This option is not available when you’re defining a table calculation with Compute Using, because those values establish partitions by position. But with Specific Dimensions, because the visual structure and the table calculation are not necessarily aligned, the At the level option is available to let you fine-tune your calculation.

Use this setting to set a break (that is, restart of the calculation) in the view, based on a particular dimension. How is this different from just using that dimension for partitioning? In fact, it is partitioning, but it’s partitioning by position rather than by value, which is how partitioning is defined with the Compute Using options.

The choices available from the At the level drop-down list in the example above are:

<table>
<thead>
<tr>
<th>Choice</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deepest</td>
<td>Specifies that the calculation should be performed at the level of finest granularity. This is the default option.</td>
</tr>
<tr>
<td>Quarter of</td>
<td>Specifies that the calculation should be performed at the quarter level.</td>
</tr>
</tbody>
</table>
### Create a table calculation

To learn how to create a table calculation, follow along with the steps in the example below. To learn how to create quick table calculations, see Quick Table Calculations on page 1559.

### Step 1: Build the visualization

1. Open Tableau and connect to the **Sample-Superstore** saved data source.
2. Navigate to a new worksheet.
3. From the **Data** pane, under Dimensions, drag **Order Date** to the **Rows** shelf.
   
   The dimension updates to **YEAR(Order Date)**.
4. On the Rows shelf, right-click **YEAR(Order Date)** and select **Quarter**.
5. On the Rows shelf, click the + icon on **QUARTER(Order Date)**.
   
   MONTH(Order Date) is added to the shelf.
6. From the **Data** pane, under Dimensions, drag **Order Date** to the **Columns** shelf.
   
   The dimension updates to **YEAR(Order Date)** again.
7. From the **Data** pane, under Measures, drag **Sales** to **Text** on the Marks card.
The updates to look like this:

![Tableau screenshot showing the updated view](image)

**Step 2: Add the table calculation**

1. On the Marks card, right-click SUM(Sales) and select **Add Table Calculation**.

2. In the Table Calculation dialog box that opens, do the following:
   
   - **For Calculation Type**: select **Difference From**.
     
     For more information about the types of table calculations you can use in Tableau, and how you can configure them, see [Table Calculation Types](#) on page 1537.

   - **For Compute Using**, select **Table (across)**.
     
     For more information about these options, see [The basics: addressing and partitioning](#) on page 1525 section.

     Note that as you select how to compute the calculation, the visualization updates with visual indicators to guide you.

   - **When finished**, click the X in the top corner of the Table Calculation dialog box to exit it.

   The calculation is applied to the values in the visualization.
Check your work!

Edit a table calculation

To edit a table calculation:

1. Right-click the measure in the view with the table calculation applied to it and select **Edit Table Calculation**.

2. In the Table Calculation dialog box that appears, make your changes.

3. When finished, click the X in the top corner of the Table Calculation dialog box to exit it.

Remove a table calculation

To remove a table calculation:

- Right-click the measure in the view with the table calculation applied to it and select **Clear Table Calculation**.

  The table calculation is removed from the measure and the visualization updates with the original values.

See Also

**Table calculations** on page 1241
Quick Table Calculations on page 1559

Table Calculation Types below

Table Calculation Types

This article describes the types of table calculations available in Tableau and when to use them. It uses simple examples to demonstrate how each calculation transforms data in a table. For more information on how to create and configure table calculations, see Create a table calculation on page 1534.

In this article

<table>
<thead>
<tr>
<th>Difference From</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving</td>
</tr>
<tr>
<td>Percent Difference From</td>
</tr>
<tr>
<td>Percent From</td>
</tr>
<tr>
<td>Percent of Total</td>
</tr>
<tr>
<td>Percentile</td>
</tr>
<tr>
<td>Rank</td>
</tr>
<tr>
<td>Running Total</td>
</tr>
</tbody>
</table>

Difference From calculation

A Difference From table calculation computes the difference between the current value and another value in the table for each mark in the visualization.

With a Difference From, Percent Difference From, or Percent From calculation, there are always two values to consider: the current value, and the value from which the difference should be calculated. In most cases, you want to calculate the difference between the current value and the previous value, as in the procedure above. But in some cases you may want something different.

To specify from which value the difference should be calculated:

1. Right-click a measure in the view and select Add Table Calculation.
2. In the Table Calculation dialog box, for Relative to, select one of the following options:
<table>
<thead>
<tr>
<th>Previous</th>
<th>Calculates the difference between the current value and the previous value in the partition. This is the default value.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td>Calculates the difference between the current value and the next value in the partition.</td>
</tr>
<tr>
<td>First</td>
<td>Calculates the difference between the current value and the first value in the partition.</td>
</tr>
<tr>
<td>Last</td>
<td>Calculates the difference between the current value and the last value in the partition.</td>
</tr>
</tbody>
</table>

Example

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.

You can use a Difference From table calculation to calculate how sales fluctuate (how much they go up or down) between the years for each month.
You can see that in January, there was a 368 USD difference between sales in 2012 and 2013, and a 26,161 USD difference between sales in 2013 and 2014.

**Moving calculation**

For each mark in the view, a **Moving Calculation** table calculation (sometimes referred to as a *rolling* calculation) determines the value for a mark in the view by performing an aggregation (sum, average, minimum, or maximum) across a specified number of values before and/or after the current value.

A moving calculation is typically used to smooth short-term fluctuations in your data so that you can see long-term trends. For example, with securities data there are so many fluctuations every day that it is hard to see the big picture through all the ups and downs. You can use a moving calculation to define a range of values to summarize using an aggregation of your choice.

**Example**

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.
You can use a Moving calculation to find out how sales totals are trending over time. To do this, you can transform each monthly total so that it averages the monthly total for it and the two previous months over time.

You can see the average sales over time. For example, the value listed for December 2011 is the average sales for October, November, and December, 2011. The value listed for January, 2012 is the average sales for November and December, 2011, and January, 2012.
Add Secondary Calculation

With Running Total and Moving Calculation table calculations, you have the option to transform values twice to obtain the result you want—that is, to add a secondary table calculation on top of the primary table calculation. For example, you could add an initial table calculation to calculate the running total for sales per month within each individual year, and then a secondary calculation to calculate the year-over-year percent difference for each month from one year to the next.

For an example showing how to create a secondary calculation, see Running Total calculation on page 1551.

Percent Difference From calculation

A Percent Difference From table calculation computes the difference between the current value and another value in the table as a percentage for each mark in the visualization.

With a Difference From, Percent Difference From, or Percent From calculation, there are always two values to consider: the current value, and the value from which the difference should be calculated. In most cases, you want to calculate the difference between the current value and the previous value, as in the procedure above. But in some cases you may want something different.

To specify from which value the difference should be calculated:

1. Right-click a measure in the view and select Add Table Calculation.
2. In the Table Calculation dialog box, for Relative to, select one of the following options:

<table>
<thead>
<tr>
<th>Calculation Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous</td>
<td>Calculates the difference between the current value and the previous value in the partition. This is the default value.</td>
</tr>
<tr>
<td>Next</td>
<td>Calculates the difference between the current value and the next value in the partition.</td>
</tr>
<tr>
<td>First</td>
<td>Calculates the difference between the current value and the first value in the partition.</td>
</tr>
<tr>
<td>Last</td>
<td>Calculates the difference between the current value and the last value in the partition.</td>
</tr>
</tbody>
</table>
Example

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.

You can use a Percent Difference From table calculation to calculate how sales fluctuate (how much they go up or down) between the years for each month. Values are calculated as percentages.
You can see that between January and February, 2011, there was a -66% difference in sales, but between February and March, 2011, there was a huge improvement of 1,058% sales.

**Percent From calculation**

A **Percent From** table calculation computes a value as a percentage of some other value—typically, as a percentage of the previous value in the table—for each mark in the visualization.

With a **Difference From**, **Percent Difference From**, or **Percent From** calculation, there are always two values to consider: the current value, and the value from which the difference should be calculated. In most cases, you want to calculate the difference between the current value and the previous value, as in the procedure above. But in some cases you may want something different.

To specify from which value the difference should be calculated:

1. Right-click a measure in the view and select **Add Table Calculation**.
2. In the Table Calculation dialog box, for **Relative to**, select one of the following options:

<table>
<thead>
<tr>
<th>Previous</th>
<th>Calculates the difference between the current value and the previous value in the partition. This is the default value.</th>
</tr>
</thead>
</table>

- 1543 -
<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td>Calculates the difference between the current value and the next value in the partition.</td>
</tr>
<tr>
<td>First</td>
<td>Calculates the difference between the current value and the first value in the partition.</td>
</tr>
<tr>
<td>Last</td>
<td>Calculates the difference between the current value and the last value in the partition.</td>
</tr>
</tbody>
</table>

**Example**

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.

You can use a Percent From table calculation to calculate the percentage of a previous value. For example, you can calculate what percentage of sales in January 2011, was made in February 2011.
You can see that February, 2011 made 34% of the sales made in January, 2011; March, 2011 made 1,158% of the sales made in February, and so on.

**Percent of Total calculation**

For each mark in the view, a **Percent of Total** table calculation computes a value as a percentage of all values in the current partition.

**Example**

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.
You can use a Percent of Total table calculation to calculate the percentage of total sales each month makes within a quarter. For example, you can see that January, 2011 makes up 18.73% of sales made in Q1.
Or you can calculate the percentage of total sales each month makes within a year. For example, you can see that January, 2011 makes up 2.88% of sales made in 2011.

Percentile calculation

For each mark in the view, a Percentile table calculation computes a percentile rank for each value in a partition.

Example

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.
You can use a Percentile table calculation to rank the total sales for each month in a year as a percentage, rather than a whole number (for example, 1 through 10).

Since February made a very small amount of sales in 2012 compared to the overall total, it is ranked as 0.0% (or number 1 out of 12, since this example is Ascending, and therefore ranked from least to most). Sales in January, 2012 were a bit higher and were therefore ranked as 9.1% (or number 2 out of 12 months). Since November made the most sales in 2012, it is ranked as 100% (or number 12 out of 12).
Descending vs. Ascending

**Ascending** order ranks values from least to most. **Descending** order ranks values from most to least.

**Rank calculation**

For each mark in the view, a **Rank** table calculation computes a ranking for each value in a partition.

**Example**

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.

<table>
<thead>
<tr>
<th>Quarter of Year</th>
<th>Month of Order</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td>$13,946</td>
<td>$18,174</td>
<td>$18,542</td>
<td>$44,703</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>$4,811</td>
<td>$12,211</td>
<td>$22,868</td>
<td>$20,284</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>$55,691</td>
<td>$38,467</td>
<td>$51,186</td>
<td>$53,909</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>$28,295</td>
<td>$34,195</td>
<td>$39,249</td>
<td>$40,112</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>$23,648</td>
<td>$30,132</td>
<td>$56,691</td>
<td>$45,651</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>$34,595</td>
<td>$24,797</td>
<td>$39,430</td>
<td>$48,260</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>$33,946</td>
<td>$28,765</td>
<td>$38,441</td>
<td>$48,428</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>$27,909</td>
<td>$36,898</td>
<td>$33,266</td>
<td>$61,156</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$81,777</td>
<td>$64,596</td>
<td>$72,908</td>
<td>$90,499</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>$31,453</td>
<td>$31,405</td>
<td>$56,463</td>
<td>$77,794</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$78,629</td>
<td>$75,973</td>
<td>$82,192</td>
<td>$112,336</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>$69,546</td>
<td>$74,920</td>
<td>$97,237</td>
<td>$90,475</td>
</tr>
</tbody>
</table>

You can use a Rank table calculation to calculate a ranking for each month in a year.
You can see that, since November made the most amount of sales in 2012, it is ranked as number 1 (because the rank is in descending order, meaning it is ordered from most to least). Since February made the least amount of sales in 2012, it is ranked number 12.

**Descending vs. Ascending**

Ascending order ranks values from least to most. Descending order ranks values from most to least. For Rank table calculation, the default value is **Descending**.

**Rank Type**

One issue with Rank calculations is that there may be more than one mark with the same value. What would happen, for example, if Tables in the Central region and Appliances in the South region both had sales of exactly $36,729? Tableau lets you specify how to handle such cases by including an additional field in the Table Calculation dialog box when you set Calculation Type to Rank.

The choices are listed below. The number sequence at the beginning of each option show how each option would rank a hypothetical set of four values where two of the values are identical:

<table>
<thead>
<tr>
<th>Option</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competition (1, 2, 2, 4)</td>
<td>Identical values are assigned an identical rank. The highest value is ranked 1 and then the next two,</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Modified Competition (1, 3, 3, 4)</td>
<td>Identical values are assigned an identical rank. The highest value is ranked 1 and then the next two, identical values, are both are ranked 3. The next value is then ranked 4.</td>
</tr>
<tr>
<td>Dense (1, 2, 2, 3)</td>
<td>Duplicate values are all given the same rank, which is the next number in the ranking sequence. The next value after the duplicate values is computed as though the duplicate values were a single value.</td>
</tr>
<tr>
<td>Unique (1, 2, 3, 4)</td>
<td>Duplicate values are given unique rankings, according to the direction in which the ranking is being computed.</td>
</tr>
</tbody>
</table>

**Running Total calculation**

For each mark in the view, a **Running Total** table calculation aggregates values cumulatively in a partition. It can do this by summing values, averaging values, or replacing all values with either the lowest or highest actual value.

Suppose you are starting with the following text view, which shows sales totals broken out by year (from left to right) and by quarter and month (from top to bottom):
Instead of absolute sales values, you want to see a running total of sales for each year, such that each month’s sales are added to all previous months’ sales.

Create the Basic View

1. Connect to the **Sample - Superstore** data source.

2. Click and drag the **Order Date** field in the Data pane and drag it to the Columns shelf.

   The default date level is **YEAR(Order Date).**

3. Click and drag **Order Date** again and drop it this time on the Rows shelf.

4. Click on the right side of the field to open the context menu. Then choose **Quarter.**
You will see two options named **Quarter**. Be sure to choose the first one.

The field should now read **QUARTER(Order Date)**.

Note: If you are creating the view on the web, the menu looks a bit different.

5. Click and drag **Order Date** a third time and drop it on the Rows shelf to the right of **QUARTER(Order Date)**.

6. Click on the right side of the field to open the context menu and this time choose **Month** (again, choose the first of two options named **Month**). The field should now read **MONTH(Order Date)**.

7. Drag **Sales** from the Data pane and drop it on Text on the Marks card.

You now have the basic view, showing Sales by Order Date over a four-year period, by month, quarter, and year.
Add a Running Total table calculation to the basic view

1. Click the SUM(Sales) field on the Marks card and choose Add table calculation.
2. In the Table Calculation dialog box, choose Running Total as the Calculation Type.
3. Choose Table (Down) from the Compute Using list.

   The highlighting in the view shows how this Compute Using value sets the scope of the calculation in the view:

   Comparing the values in the original text view with the values in this view shows that the result is correct. The monthly values ascend steadily and the December value (484,247) is the same value you see if you show column grand totals (from the Analysis menu, select Totals > Show column grand totals).
4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

The Running Total doesn’t have to be a sum

For a Running Total table calculation, Tableau can update values cumulatively in other ways than summing. Choose one of the options from the drop-down list just below the Calculation Type field:
<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>Each value is added to the previous value.</td>
</tr>
<tr>
<td>Average</td>
<td>The running total averages the current and all previous values.</td>
</tr>
<tr>
<td>Minimum</td>
<td>All values are replaced with the lowest value in the original partition.</td>
</tr>
<tr>
<td>Maximum</td>
<td>All values are replaced with the highest value in the original partition.</td>
</tr>
</tbody>
</table>

**Restarting every option**

The **Restarting every** option is only available when you select **Specific Dimensions** in the Table Calculations dialog box and when more than one dimension is selected in the field immediately below the **Compute Using** options—that is, when more than one dimension is defined as an addressing field.

This option is not available when you’re defining a table calculation with **Compute Using**.

You can use this setting to set a break (that is, restart of the calculation) in the view, based on a particular dimension.

Restarting every can be useful in the following situations:

- With dates or other hierarchies, if you restart every month, as you bring in Year or Quarter, Tableau knows to partition automatically.

- With non-hierarchies, **Restarting every** affects the sorting. If you want to address on **Products** and partition by **State**, but you want the products sorted by **SUM(Sales)** within each state, you need to include **States** as an addressing field under **Specific Dimensions**, but then restart every state. Otherwise, the sort by **SUM(Sales)** would be based on each product's sum of sales across all states.

For example, if you take the result of the **Running Total** calculation you added above, you can see the effect of **Restarting every** by doing the following:

1. Click the **SUM(Sales)** field on the Marks card and choose **Edit table calculation**.
2. In the Table Calculation dialog box, choose **Specific Dimensions**.
Notice that two dimensions are now checked in the list box of dimensions: **Quarter of Order Date** and **Month of Order Date**. These are the addressing fields, and because more than one field is being used for addressing, Restarting every is now available.

The choices available from the At the level drop-down list are:

<table>
<thead>
<tr>
<th>None</th>
<th>Specifies that the calculation should be performed at the level of greatest granularity. This is the default option. This option does not change the view.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quarter of Order Date</td>
<td>Specifies that the calculation should be performed at the quarter level.</td>
</tr>
</tbody>
</table>

3. If you choose Quarter of Order Date, the view updates to show the effect of this change:

```
Quarter of Order Date

<table>
<thead>
<tr>
<th>Month of Order Date</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>January</td>
<td>$13,946</td>
<td>$18,174</td>
<td>$18,542</td>
<td>$44,703</td>
</tr>
<tr>
<td>February</td>
<td>$18,757</td>
<td>$30,385</td>
<td>$41,410</td>
<td>$64,987</td>
</tr>
<tr>
<td>March</td>
<td>$74,448</td>
<td>$68,852</td>
<td>$92,596</td>
<td>$118,896</td>
</tr>
<tr>
<td>Q2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>April</td>
<td>$28,295</td>
<td>$34,195</td>
<td>$39,249</td>
<td>$40,112</td>
</tr>
<tr>
<td>May</td>
<td>$51,944</td>
<td>$64,327</td>
<td>$95,940</td>
<td>$85,764</td>
</tr>
<tr>
<td>June</td>
<td>$86,539</td>
<td>$89,124</td>
<td>$135,370</td>
<td>$134,023</td>
</tr>
<tr>
<td>Q3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>July</td>
<td>$33,946</td>
<td>$28,765</td>
<td>$38,441</td>
<td>$48,428</td>
</tr>
<tr>
<td>August</td>
<td>$61,856</td>
<td>$55,654</td>
<td>$71,706</td>
<td>$109,944</td>
</tr>
<tr>
<td>September</td>
<td>$143,633</td>
<td>$130,260</td>
<td>$144,614</td>
<td>$200,433</td>
</tr>
<tr>
<td>Q4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>October</td>
<td>$31,453</td>
<td>$31,405</td>
<td>$56,463</td>
<td>$77,794</td>
</tr>
<tr>
<td>November</td>
<td>$110,082</td>
<td>$107,377</td>
<td>$138,655</td>
<td>$190,120</td>
</tr>
<tr>
<td>December</td>
<td>$179,628</td>
<td>$182,297</td>
<td>$235,893</td>
<td>$280,595</td>
</tr>
</tbody>
</table>
```

The calculation now restarts after every quarter. If you click out of the Table Calculations dialog box (to dismiss the highlighting) you can see this more clearly.

4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.
Add Secondary Calculation

With **Running Total** and **Moving Calculation** table calculations, you have the option to transform values twice to obtain the result you want—that is, to add a secondary table calculation on top of the primary table calculation. For example, you could add an initial table calculation to calculate the running total for sales per month within each individual year, and then a secondary calculation to calculate the year-over-year percent difference for each month from one year to the next.

To do this, first add the primary table calculation, as shown above. Then continue as follows:

1. Click the **SUM(Sales)** field on the Marks card and select **Edit table calculation**.
2. In the Table Calculation dialog box, click **Add Secondary Calculation**.
   
   The Table Calculation dialog box expands to show a second panel:
3. In the second panel, choose **Percent Difference From** as the **Secondary Calculation Type**.

4. You do not need to change the **Compute Using** selection: **Table (Across)** is the right option.

5. Click the X in the upper-right corner to dismiss the Table Calculation dialog box.

    Now your view shows what you needed: a year-over-year percent difference of a running total:
Quick Table Calculations

Quick table calculations allow you to quickly apply a common table calculation to your visualization using the most typical settings for that calculation type. This article demonstrates how to apply a quick table calculation to a visualization using an example.

The following quick table calculations are available in Tableau for you to use:

- Running total
- Difference
- Percent difference
- Percent of total
- Rank
- Percentile
- Moving average
- YTD total
- Compound growth rate
- Year of year growth
- YTD growth
How does a quick table calculation differ from a table calculation?

Quick table calculations are table calculations that you can apply quickly to your visualization in Tableau. They are applied to the visualization with the most typical settings for the calculation type you choose so that you can continue on with your analysis. With traditional table calculations, you can apply the same settings, but you must apply them manually.

Apply a quick table calculation to the visualization

Follow along with the steps below to learn how to apply a quick table calculation to a visualization.

Step 1: Set up the visualization

1. Open Tableau Desktop and connect to the Sample-Superstore data source, which comes with Tableau.
2. Navigate to a new worksheet.
3. From the Data pane, under Dimensions, drag Order Date to the Columns shelf.
4. From the Data pane, under Dimensions, drag State to the Rows shelf.
5. From the Data pane, under Measures, drag Sales to Text on the Marks Card.
6. From the Data pane, under Measures, drag Profit to Color on the Marks Card.
7. On the Marks card, click the Mark Type drop-down and select Square.
The visualization updates to look like this:
Step 2: Apply the quick table calculation

1. On the Marks card, right-click **SUM(Profit)** and select **Quick Table Calculation > Moving Average**.

Note: You can only perform quick table calculations on measures in the view.

A delta symbol appears on the field to indicate that a quick table calculation is being applied to the field. The colors in the visualization update to show the moving average of profit across the years.
Step 3: (Optional) Customize the quick table calculation

1. On the Marks card, right-click Sum(Profit) and select Edit Table Calculation.

2. In the dialog box that opens, you can configure the following options:
   - The calculation type
   - How to aggregate the values
   - How to compute the calculation (how to address and partition the calculation)

For more information about these options, see Table Calculation Types on page 1537 and The basics: addressing and partitioning on page 1525

The visualization updates as you make changes to the calculation. Highlighting and numbering are used to demonstrate how the calculation is being computed. For example, in the following image, the calculation is being computed across the table, for each State.
If this setting is changed, the visualization and highlighting update to indicate the change.

See Also

Understanding Calculations: Table calculations

Transform Values with Table Calculations on page 1524

Customize Table Calculations below

Customize Table Calculations

You can always customize a table calculation by editing it in the Table Calculations dialog box, but there are other, more specialized ways to customize a table calculation.

Customizing a Table Calculation Using Its Context Menu

Click any field in the view to see a context menu listing ways to customize the field. For a field in the view that has a table calculation, you can change the Compute Using option—that is, the option that determines the direction and scope of the calculation relative to the visual structure
of the view. Do this by clicking the field and then choosing an option from the **Compute Using** list.

For **Difference From**, **Percent Difference From**, and **Percent From** table calculations, you can also specify a different field from which the difference should be computed. Do this by clicking the field and then choosing an option from the **Relative to** list. The options are **Previous**, **Next**, **First**, and **Last**.

**Customizing a Table Calculation Using the Calculation Editor**

You can customize a table calculation by dragging it into the calculation editor:

![Customizing a Table Calculation](image)

When you edit a table calculation in the calculation editor, you can click **Default Table Calculation** in the lower-right corner of the editor to open the calculation in the Table Calculation dialog box. This will allow you to create a new named calculated field that uses the same table calculation as the one you are basing the calculation on.
Nested Table Calculations

A nested table calculation can be one of two types of calculated fields:

- A calculated field that includes more than one calculated field with a table calculation (as in the example below), or
- A calculated field that itself has a table calculation and includes at least one calculated field with a table calculation.

With nested table calculations, you can set **Compute Using** configurations for individual calculations independently.

Here is a scenario you can try, using the Sample - Superstore data source that is included with Tableau Desktop, which results in a nested table calculation.

1. Drag **Sub-Category** to Columns and **Region** to Rows.
2. Create a calculated field, **1-nest**, with the definition `TOTAL(SUM([Sales]))`.
   
   TOTAL is a table calculation function, so this calculated field automatically has a table calculation—when you use it in the view, the field will have the tell-tale table triangle, indicating a table calculation:

   ![1-nest](image)

   For information on table calculation functions, see **Table Calculation Functions** on page 1347.

3. Create a second calculated field, **2-nest**, with the definition `TOTAL(SUM([Profit]))`.
4. Create a third calculated field, **3-nest**, with the definition `[1-nest] + [2-nest]`.
5. Drag **3-nest** and drop it to the right of **Sub-Category** on Columns.
6. Click **3-nest** on Columns and choose **Edit Table Calculation**.
   
   In the Table Calculations dialog box, you can now separately configure the underlying
Create Level of Detail Expressions in Tableau

Level of Detail expressions (also known as LOD expressions) allow you to compute values at the data source level and the visualization level. However, LOD expressions give you even more control on the level of granularity you want to compute. They can be performed at a more granular level (INCLUDE), a less granular level (EXCLUDE), or an entirely independent level (FIXED).

This article explains the types of LOD expressions you can use in Tableau, as well as when to use them, and how to format them. It also uses an example to demonstrate how to create a simple LOD expression.

In this article

- How to create LOD expressions on the next page
- Types of LOD expressions on page 1571
- LOD expression syntax on page 1581
How to create LOD expressions

Follow along with the steps below to learn how to create and use an LOD expression in Tableau.

Step 1: Set up the Visualization

1. Open Tableau Desktop and connect to the Sample-Superstore saved data source.
2. Navigate to a new worksheet.
3. From the Data pane, under Dimensions, drag Region to the Columns Shelf.
4. From the Data pane, under Measures, drag Sales to the Rows Shelf.
A bar chart showing the sum of sales for each region appears.

Step 2: Create the LOD expression

Instead of the sum of all sales per region, perhaps you want to also see the average sales per customer for each region. You can use an LOD expression to do this.

1. Select Analysis > Create Calculated Field.
2. In the Calculation editor that opens, do the following:
- Name the calculation, Sales Per Customer.
- Enter the following LOD expression:

```plaintext
{ INCLUDE [Customer Name] : SUM([Sales]) }
```

3. When finished, click OK.

The newly created LOD expression is added to the Data pane, under Measures. To learn more about the types of LOD expressions you can use, see the Types of LOD expressions on the next page section.

**Step 3: Use the LOD expression in the visualization**

1. From the Data pane, under Measures, drag Sales Per Customer to the Rows shelf and place it to the left of SUM(Sales).

2. On the Rows shelf, right-click Sales Per Customer and select Measure (Sum) > Average.

You can now see both the sum of all sales and the average sales per customer for each region. For example, you can see that in the Central region, the sales totaled approximately 500,000 USD with an average sale for each customer being
approximately 800 USD.

Types of LOD expressions

There are three types of LOD expressions you can create in Tableau:

- **FIXED** on the next page
- **INCLUDE** on page 1574
- **EXCLUDE** on page 1578

You can also scope an LOD expression to the table. This is called a **Table-Scoped** on page 1580 LOD expression.
FIXED

FIXED level of detail expressions compute a value using the specified dimensions, without reference to the dimensions in the view.

Example

The following FIXED level of detail expression computes the sum of sales per region:

{FIXED [Region] : SUM([Sales])}

This level of detail expression, named [Sales by Region], is then placed on Text to show total sales per region.

The view level of detail is [Region] plus [State], but because FIXED level of detail expressions do not consider the view level of detail, the calculation only uses the dimension referenced in the calculation, which in this case is Region. Because of this, you can see that the values for the individual states in each region are identical. For more information about why this happens, see Aggregation and Level of Detail Expressions on page 1590.
If the INCLUDE keyword had been used in the level of detail expression instead of FIXED, the values would be different for each state, because Tableau would add the dimension in the expression ([Region]) with any additional dimensions in the view ([State]) when determining values for the expression. The result would be as follows:
INCLUDE

INCLUDE level of detail expressions compute values using the specified dimensions in addition to whatever dimensions are in the view.

INCLUDE level of detail expressions can be useful when you want to calculate at a fine level of detail in the database and then re-aggregate and show at a coarser level of detail in your view. Fields based on INCLUDE level of detail expressions will change as you add or remove dimensions from the view.

Example 1

The following INCLUDE level of detail expression computes total sales per customer:

```plaintext
{ INCLUDE [Customer Name] : SUM([Sales]) }
```
When that calculation is placed on the **Rows** shelf, aggregated as AVG, and the [**Region**] dimension is placed on the **Columns** shelf, the view shows the average customer sales amount per region:

If the [**Sales**] measure is then dragged to the **Rows** shelf, the result illustrates the difference between the total sale for each region and the average sale per customer for each region:
Example 2

The following INCLUDE level of detail expression calculates sum of sales on a per-state basis:

{ INCLUDE [State] : SUM(Sales) }

The calculation is placed on the Rows shelf and is aggregated as an average. The resulting visualization averages the sum of sales by state across categories.
When **Segment** is added to the **Columns** shelf and the calculation is moved to **Label**, the LOD expression results update. Now you can see how the average sum of sales per state varies across categories and segments.
EXCLUDE

EXCLUDE level of detail expressions declare dimensions to omit from the view level of detail.

EXCLUDE level of detail expressions are useful for ‘percent of total’ or ‘difference from overall average’ scenarios. They are comparable to such features as Totals and Reference Lines.

EXCLUDE level of detail expression cannot be used in row-level expressions (where there are no dimensions to omit), but can be used to modify either a view level calculation or anything in between (that is, you can use an EXCLUDE calculation to remove dimension from some other level of detail expression).

Example 1

The following EXCLUDE level of detail expression computes the average sales total per month and then excludes the month component:

```
{EXCLUDE [Order Date (Month / Year)] : AVG({FIXED [Order Date (Month / Year)] : SUM([Sales])})}
```
Notice that this is a nested level of detail expression—that is, a level of detail expression within another level of detail expression.

Saved as [average of sales by month], the calculation can then be subtracted from the sum of sales per month by means of an ad-hoc calculation on the Rows shelf:

\[
\text{SUM}([\text{Sales}]) - \text{SUM}([\text{average of sales by month}])
\]

With Month([Order Date]) on the Columns shelf, this creates a view that shows the difference between actual sales per month over a four-year period and the average monthly sales for the entire four-year period:

Example 2

The following level of detail expression excludes [Region] from a calculation of the sum of [Sales]:

\[
\{\text{EXCLUDE [Region]}: \text{SUM}([\text{Sales}])\}
\]

The expression is saved as [ExcludeRegion].

To illustrate how this expression might be useful, first consider the following view, which breaks out the sum of sales by region and by month:
Dropping [ExcludeRegion] on Color shades the view to show total sales by month but without the regional component:

**Table-Scoped**

It is possible to define a level of detail expression at the table level without using any of the scoping keywords. For example, the following expression returns the minimum (earliest) order date for the entire table:

```
{MIN([Order Date])}
```

This is equivalent to a FIXED level of detail expression with no dimension declaration:
LOD expression syntax

Level of Detail Expression Syntax

A level of detail expression has the following structure:

\{ [FIXED | INCLUDE | EXCLUDE] <dimension declaration> : <aggregate expression> \}

The elements in a level of detail expression are described in the following table.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{}</td>
<td>The entire level of detail expression is enclosed in curly braces.</td>
</tr>
<tr>
<td>[FIXED</td>
<td>INCLUDE</td>
</tr>
<tr>
<td></td>
<td>• FIXED</td>
</tr>
<tr>
<td></td>
<td>FIXED level of detail expressions compute values using the specified dimensions without reference to the view level of detail—that is, without reference to any other dimensions in the view.</td>
</tr>
<tr>
<td></td>
<td>FIXED level of detail expressions also ignore all the filters in the view other than context filters, data source filters, and extract filters.</td>
</tr>
<tr>
<td></td>
<td>Example: { FIXED [Region] : SUM([Sales]) }</td>
</tr>
<tr>
<td></td>
<td>For more information about FIXED level of detail expressions, and for some example FIXED level of detail scenarios, see the FIXED on</td>
</tr>
</tbody>
</table>
• **INCLUDE**

INCLUDE level of detail expressions compute values using the specified dimensions in addition to whatever dimensions are in the view.

INCLUDE level of detail expressions are most useful when including a dimension that isn’t in the view.

**Example:**
```
{ INCLUDE [Customer Name] : SUM([Sales]) }
```

For more information about INCLUDE level of detail expressions, and for some example INCLUDE level of detail scenarios, see the INCLUDE on page 1574 section.

• **EXCLUDE**

EXCLUDE level of detail expressions explicitly remove dimensions from the expression—that is, they subtract dimensions from the view level of detail.

EXCLUDE level of detail expressions are most useful for eliminating a dimension in the view.

**Example:**
```
{ EXCLUDE [Region]: SUM([Sales]) }
```

For more information about EXCLUDE level of detail expressions, and for some example EXCLUDE level of detail scenarios, see the EXCLUDE on page 1578 section.
### Table-Scoped

In the case of a table-scoped level of detail expression, no scoping keyword is required. For more information, see the Table-Scoped on page 1580 section.

| <dimension declaration> | Specifies one or more dimensions to which the aggregate expression is to be joined. Use commas to separate dimensions. For example:

\[
[\text{Segment}], \ [\text{Category}], \ [\text{Region}]
\]

For level of detail expressions, you can use any expression that evaluates as dimension in a dimensionality declaration, including Date expressions.

This example will aggregate the sum of Sales at the Year level:

\[
\{\text{FIXED YEAR}([\text{Order Date}]) : \text{SUM} (\text{Sales})\}
\]

This example will aggregate the sum of Sales for the [Order Date] dimension, truncated to the day date part. Because it is an INCLUDE expression, it will also use the dimensions in the view to aggregate the value:

\[
\{\text{INCLUDE DATETRUNC('day', [Order Date]) : AVG(Profit)}\}
\]

**Note:** It is strongly recommended that you drag fields into the calculation.
editor when creating dimension declarations, instead of typing them. For example, if you see YEAR ([Order Date]) on a shelf and then type that as the dimension declaration, it will not match the field on the shelf. But if you drag the field from the shelf into the expression, it will become DATEPART('year', [Order Date]), and that will match the field on the shelf.

With named calculations (that is, calculations that you save to the Data pane, as opposed to ad-hoc calculations, which you do not name), Tableau cannot match the name of a calculation to its definition. So if you create a named calculation, MyCalculation, defined as follows:

MyCalculation = YEAR([Order Date])

And then you created the following EXCLUDE level of detail expression and used it in the view:

{EXCLUDE YEAR([Order Date]) : SUM(Sales)}

Then MyCalculation would not be excluded. Similarly, if the EXCLUDE expression specified MyCalculation:

{EXCLUDE MyCalculation : SUM(Sales)}

Then YEAR([Order Date]) would not be
A colon separates the dimension declaration from the aggregate expression.

- \(<aggregate\ expression>\) - The aggregate expression is the calculation performed to define the target dimensionality.

See Also

- Introduction to Level of Detail Expressions
- A Deeper Look at LOD Expressions
- Diving into LOD Expressions
- Top 15 LOD Expressions
- Understanding LOD Expressions

How Level of Detail Expressions Work in Tableau below

How Level of Detail Expressions Work in Tableau

This article explains how level of detail expressions are computed and how they function in Tableau. For more information about LOD expressions and how they work, see the Understanding Level of Detail (LOD) Expressions whitepaper on the Tableau website.

In this article

- Row Level Expressions and View Level Expressions on the next page
- Limitations for Level of Detail Expressions on page 1587
- Level of Detail Expressions Can Be Dimensions or Measures on page 1588
- Filters and Level of Detail Expressions on page 1588
- Aggregation and Level of Detail Expressions on page 1590
Row Level Expressions and View Level Expressions

In Tableau, expressions referencing unaggregated datasource columns are computed for each row in the underlying table. In this case, the dimensionality of the expression is row level. An example of a row-level expression is:

\[
\frac{\text{[Sales]}}{\text{[Profit]}}
\]

This calculation will be evaluated in each row of the database. For each row, the Sales value in that row will be divided by the Profit value in that row, producing a new column with the result of the multiplication (a profit ratio).

If you create a calculation with this definition, save it with the name \[\text{ProfitRatio}\], and then drag it from the Data pane to a shelf, Tableau typically aggregates the calculated field for the view:

\[
\text{SUM[ProfitRatio]}
\]

By contrast, expressions referencing aggregated data source columns are computed at the dimensionality defined by the dimensions in the view. In this case, the dimensionality of the expression is view level. An example of a view-level expression is:

\[
\frac{\text{SUM(Sales)}}{\text{SUM(Profit)}}
\]

If you drag this calculation to a shelf (or type it directly on a shelf as an ad-hoc calculation), Tableau encloses it in an AGG function:

\[
\text{AGG(SUM(Sales) / SUM(Profit))}
\]

This is what is known as an aggregate calculation. For details, see Aggregate Functions in Tableau on page 1320.

Dimension and set fields placed on any of the locations highlighted in the following image contribute to the view level of detail:
Before level of detail expressions were supported in Tableau, it was not possible to create calculations at a level of detail other than the view level. For example, if you attempt to save the following expression, Tableau displays the error message: "Cannot mix aggregate and non-aggregate arguments with this function":

\[ [\text{Sales}] - \text{AVG}([\text{Sales}]) \]

The user's intent in this case was to compare store sales for each individual store to the average of sales for all stores. This can now be accomplished with a level of detail expression:

\[ [\text{Sales}] - \{\text{AVG}([\text{Sales}]) \} \]

This is what is known as a table-scoped level of detail expression. See Table-Scoped on page 1580.

**Limitations for Level of Detail Expressions**

The following limitations and constraints apply for level of detail expressions. Also see Data Source Constraints for Level of Detail Expressions on page 1593.
- Level of detail expressions that reference floating-point measures can behave unreliably when used in a view that requires comparison of the values in the expression. For details, see Understanding data types in calculations on page 1273.

- Level of detail expressions are not shown on the Data Source page. See Data Source Page on page 139.

- When referencing a parameter in a dimensionality declaration, always use the parameter name, and not the parameter value.

- With data blending, the linking field from the primary data source must be in the view before you can use a level of detail expression from the secondary data source. See Troubleshoot Data Blending on page 696.

In addition, some data sources have complexity limits. Tableau will not disable calculations for these databases, but query errors are a possibility if calculations become too complex.

### Level of Detail Expressions Can Be Dimensions or Measures

When you save a level of detail expression, Tableau adds it to either the Dimensions or the Measures area in the Data pane.

**FIXED** level of detail expressions can result in measures or dimensions, depending on the underlying field in the aggregate expression. So MIN([Date])} will be a dimension because [Date] is a dimension, and {fixed Store : SUM([Sales])} will be a measure because [Sales] is a measure. When a **FIXED** level of detail expression is saved as a measure you have the option of moving it to dimensions.

**INCLUDE** and **EXCLUDE** level of detail expressions are always measures.

### Filters and Level of Detail Expressions

There are several different kinds of filters in Tableau and they get executed in the following order from top to bottom.
The text on the right shows where level of detail expressions are evaluated in this sequence.

Extract Filters (in orange) are only relevant if you’re creating a Tableau Extract from a data source. Table calculations filters (dark blue) are applied after calculations are executed and therefore hide marks without filtering out the underlying data used in the calculations.

If you’re familiar with SQL, you can think of measure filters as equivalent to the HAVING clause in a query, and dimension filters as equivalent to the WHERE clause.

FIXED calculations are applied before dimension filters, so unless you promote the fields on your Filter shelf to Improve View Performance with Context Filters on page 1194, they will be ignored. For example, consider if you have the following calculation on one shelf in a view, along with [State] on a different shelf:

\[
\text{SUM([Sales])} / \text{ATTR({FIXED: SUM([Sales])})}
\]

This calculation will give you the ratio of a state’s sales to total sales.

If you then put [State] on the Filters shelf to hide some of the states, the filter will affect only the numerator in the calculation. Since the denominator is a FIXED level of detail expression, it will still divide the sales for the states still in the view against the total sales for all states—including the ones that have been filtered out of the view.

INCLUDE and EXCLUDE level of detail expressions are considered after Dimension filters. So if you want filters to apply to your FIXED level of detail expression but don’t want to use Context Filters, consider rewriting them as INCLUDE or EXCLUDE expressions.
Aggregation and Level of Detail Expressions

The level of detail of the view determines the number of marks in your view. When you add a level of detail expression to the view, Tableau must reconcile two levels of detail—the one in the view, and the one in your expression.

The behavior of a level of detail expression in the view varies depending on whether the expression's level of detail is coarser, finer, or the same as the level of detail in the view. What do we mean by “coarser” or “finer” in this case?

Level of Detail Expression is Coarser Than View Level of Detail

An expression has a coarser level of detail than the view when it references a subset of the dimensions in the view. For example, for a view that contained the dimensions [Category] and [Segment], you could create a level of detail expression that uses only one of these dimensions:

{FIXED [Segment] : SUM([Sales])}

In this case, the expression has a coarser level of detail than the view. It bases its values on one dimension ([Segment]), whereas the view is basing its view on two dimensions ([Segment] and [Category]).

The result is that using the level of detail expression in the view causes certain values to be replicated—that is, to appear multiple times.
Replicated values are useful for comparing specific values against average values within a category. For example, the following calculation subtracts average sales for a customer from the average sales overall:

\[
\text{[Sales]} - \{\text{FIXED [Customer Name]} : \text{AVG([Sales])}\}
\]

When values are being replicated, changing the aggregation for the relevant field in the view (for example, from \text{AVG} to \text{SUM}) will not change the result of the aggregation.

### Level of Detail Expression is Finer Than View Level of Detail

An expression has a finer level of detail than the view when it references a superset of the dimensions in the view. When you use such an expression in the view, Tableau will aggregate results up to the view level. For example, the following level of detail expression references two dimensions:

\[
\{\text{FIXED [Segment], [Category]} : \text{SUM([Sales])}\}
\]

When this expression is used in a view that has only \text{[Segment]} as its level of detail, the values must be aggregated. Here's what you would see if you dragged that expression to a shelf:

\[
\text{AVG}([[\text{FIXED [Segment]}, [Category]] : \text{SUM([Sales])}])
\]

An aggregation—in this case, average—is automatically assigned by Tableau. You can change the aggregation as needed.
Adding a Level of Detail Expression to the View

Whether a level of detail expression is aggregated or replicated in the view is determined by the expression type (FIXED, INCLUDE, or EXCLUDE) and whether the expression’s granularity is coarser or finer than the view’s.

- INCLUDE level of detail expressions will have either the same level of detail as the view or a finer level of detail than the view. Therefore, values will never be replicated.
- FIXED level of detail expressions can have a finer level of detail than the view, a coarser level of detail, or the same level of detail. The need to aggregate the results of a FIXED level of detail depends on what dimensions are in the view.
- EXCLUDE level of detail expressions always cause replicated values to appear in the view. When calculations including EXCLUDE level of detail expressions are placed on a shelf, Tableau defaults to the ATTR aggregation (as opposed to SUM or AVG) to indicate that the expression is not actually being aggregated and that changing the aggregation will have no effect on the view.

Level of detail expressions are always automatically wrapped in an aggregate when they are added to a shelf in the view unless they’re used as dimensions. So if you double-click on a shelf and type

\{FIXED[Segment], [Category] : SUM([Sales])\}

and then press Enter to commit the expression, what you now see on the shelf is

SUM({FIXED[Segment], [Category] : SUM([Sales])})

But if you double-click into the shelf to edit the expression, what you see in edit mode is the original expression.

If you wrap a level of detail expression in an aggregation when you create it, Tableau will use the aggregation you specified rather than assigning one when any calculation including that expression is placed on a shelf. When no aggregation is needed (because the expression’s level of detail is coarser than the view’s), the aggregation you specified is still shown when the expression is on a shelf, but it is ignored.
Data Source Constraints for Level of Detail Expressions

For some data sources, only more recent versions support level of detail expressions. Some data sources do not support level of detail expressions at all.

In addition, some data sources have complexity limits. Tableau will not disable calculations for these databases, but query errors are a possibility if calculations become too complex.

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actian Vectorwise</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Amazon EMR Hadoop Hive</td>
<td>Supported for Hive 0.13 and later.</td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td>Supported.</td>
</tr>
<tr>
<td>Aster Database</td>
<td>Supported for version 4.5 and later.</td>
</tr>
<tr>
<td>Cloudera Hadoop</td>
<td>Supported for Hive 0.13 and later.</td>
</tr>
<tr>
<td>Cloudera Impala</td>
<td>Supported for Impala 1.2.2 and later.</td>
</tr>
<tr>
<td>Cubes (multidimensional data sources)</td>
<td>Not supported.</td>
</tr>
<tr>
<td>DataStax Enterprise</td>
<td>Not supported.</td>
</tr>
<tr>
<td>EXASOL</td>
<td>Supported.</td>
</tr>
<tr>
<td>Firebird</td>
<td>Supported for version 2.0 and later.</td>
</tr>
<tr>
<td>Generic ODBC</td>
<td>Limited. Depends on the specific data source.</td>
</tr>
<tr>
<td>Google Big Query</td>
<td>Supported for standard SQL, not supported for legacy SQL.</td>
</tr>
<tr>
<td>Software</td>
<td>Supported Details</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Hortonworks Hadoop Hive</td>
<td>Supported for Hive 0.13 and later.</td>
</tr>
<tr>
<td></td>
<td>On version 1.1 of HIVE level of detail expressions that produce cross joins are not reliable.</td>
</tr>
<tr>
<td></td>
<td>Cross join occur when there is no explicit field to join on. For example, for a level of detail expression ({\text{fixed [Product Type]} : \sum(\text{sales})}) when the view only contains one dimension [Ship Mode], Tableau creates a cross-join. A cross join produces rows which combine each row from the first table with each row from the second table.</td>
</tr>
<tr>
<td>IBM BigInsights</td>
<td>Supported.</td>
</tr>
<tr>
<td>IBM DB2</td>
<td>Supported for version 8.1 and later.</td>
</tr>
<tr>
<td>MarkLogic</td>
<td>Supported for version 7.0 and later.</td>
</tr>
<tr>
<td>Microsoft Access</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Microsoft Jet-based connections (legacy connectors for Microsoft Excel, Microsoft Access, and text)</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>SQL Server 2005 and later.</td>
</tr>
<tr>
<td>MySQL</td>
<td>Supported.</td>
</tr>
<tr>
<td>IBM PDA (Netezza)</td>
<td>Supported version 7.0 and later.</td>
</tr>
<tr>
<td>Oracle</td>
<td>Supported version 9i and later.</td>
</tr>
<tr>
<td>Software</td>
<td>Version Supported</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Actian Matrix (ParAccel)</td>
<td>Supported version 3.1 and later.</td>
</tr>
<tr>
<td>Pivotal Greenplum</td>
<td>Supported for version 3.1 and later.</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Supported version 7 and later.</td>
</tr>
<tr>
<td>Progress OpenEdge</td>
<td>Supported.</td>
</tr>
<tr>
<td>SAP HANA</td>
<td>Supported.</td>
</tr>
<tr>
<td>SAP Sybase ASE</td>
<td>Supported.</td>
</tr>
<tr>
<td>SAP Sybase IQ</td>
<td>Supported version 15.1 and later.</td>
</tr>
<tr>
<td>Spark SQL</td>
<td>Supported.</td>
</tr>
<tr>
<td>Splunk</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Tableau Data Extract</td>
<td>Supported.</td>
</tr>
<tr>
<td>Teradata</td>
<td>Supported.</td>
</tr>
<tr>
<td>Vertica</td>
<td>Supported for version 6.1 and later.</td>
</tr>
</tbody>
</table>

**See Also**

*Create Level of Detail Expressions in Tableau* on page 1567

*Understanding Level of Detail (LOD) Expressions*

**Table-Scoped Level of Detail Expressions**

It is possible to define a level of detail expression at the table level without using any of the scoping keywords. For example, the following expression returns the minimum (earliest) order date for the entire table:

\[
\{\text{MIN}([\text{Order Date}])\}
\]

This is equivalent to a FIXED level of detail expression with no dimension declaration:
FIXED Level of Detail Expressions

FIXED level of detail expressions compute a value using the specified dimensions, without reference to the dimensions in the view.

The following examples can be recreated using the Sample - Superstore data source.

Example 1

The following view shows the interval between a customer’s first purchase date and any subsequent purchase:

Step-by-Step

Here’s how to build the view above, using the Sample - Superstore data source provided with Tableau Desktop.
1. Create two calculated fields: a FIXED level of detail expression, and a date subtraction.

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Purchase Date</td>
<td>{FIXED [Customer Name] : MIN([Order Date])}</td>
</tr>
<tr>
<td>Days Since First Purchase</td>
<td>DATETRUNC('day', [Order Date]) - DATETRUNC('day', [First Purchase Date])</td>
</tr>
</tbody>
</table>

2. Drag Days Since First Purchase from the Measures area of the Data pane to the Dimensions area.

Because this calculated field contains a subtraction operation, the result is a number, and so Tableau assigns it to the measure category. But you will be using it as a dimension.

3. Drag Days Since First Purchase to Columns.

4. Click Days Since First Purchase on Columns and choose Continuous.

5. Drag Sales to Rows.

6. Change the aggregation for Sales on Rows from SUM to AVG.

7. Add a quick table calculation to Sales on Rows: Running Total.

8. Drag First Purchase Date to Color.

9. Click the + in the YEAR(First Purchase Date) field on Color to add the next level down in the date hierarchy: QUARTER(First Purchase Date).
10. Tableau won't automatically put two fields on Color, but you can make it happen by clicking the icon to the left of the QUARTER(First Purchase Date) field and choosing Color:

11. Drag the Color legend over to the left side of the view, below the Marks card.

Your view should now look like this:
The visualization yields insights as you interact with it. Select individual quarters in the color legend on the left to see how customers acquired in specific individual quarters continued to spend in subsequent quarters. Notice that customers acquired in the early going (2013) tend to spend at a higher rate, even allowing for the fact that they had more time to do so—thus, the blue lines (for 2013) are higher up on the x-axis than other lines. If you had used a standard date value on the y-axis, the view would have been easier to create, but the lines would not all start at the same coordinates and so it would be more difficult to compare acquisition rates.

You can also drag either YEAR(First Purchase Date) or QUARTER(First Purchase Date) off color to see seasonal or annual trends more clearly.

Example 2

The following FIXED level of detail expression computes the sum of sales per region:

{FIXED [Region] : SUM([Sales])}

This level of detail expression, named [Sales by Region], is then placed on Text to show total sales per region:
The view level of detail is [Region] plus [State], but because FIXED level of detail expressions do not consider the view level of detail, the calculation only uses the [Region] dimension, and so the values for the individual states in each region are identical. See Level of Detail Expressions and Aggregation on page 1607 for an explanation of why this is so.

If the INCLUDE keyword had been used in the level of detail expression instead of FIXED, the values would be different for each state, because Tableau would add the dimension in the expression ([Region]) with any additional dimensions in the view ([State]) in determining values for the expression. The result would be:
INCLUDE Level of Detail Expressions

INCLUDE level of detail expressions compute values using the specified dimensions in addition to whatever dimensions are in the view.

INCLUDE level of detail expressions can be useful when you want to calculate at a fine level of detail in the database and then re-aggregate and show at a coarser level of detail in your view. Fields based on INCLUDE level of detail expressions will change as you add or remove dimensions from the view.

Example 1

The following INCLUDE level of detail expression computes total sales per customer:

{ INCLUDE [Customer Name] : SUM([Sales]) }
When that calculation is placed on the **Rows** shelf, aggregated as AVG, and the [Region] dimension is placed on the **Columns** shelf, the view shows the average customer sales amount per region:

If the [Sales] measure is then dragged to the **Rows** shelf, the result illustrates the difference between the average sale (somewhere between $200 and $250 per region) and the average sale per customer (between $750 and $1100 per region):
Example 2

The following INCLUDE level of detail expression calculates sum of sales on a per-state basis:

{ INCLUDE [State] : SUM(Sales) }

Type the calculation directly on the Rows shelf, and then change the aggregation to AVG using the field's context menu. The resulting view averages the sum of sales by state across categories.
The view can be further enhanced by adding [Segment] to Columns and then copying the ad-hoc calculation from Rows to Label (using Ctrl+drag). In the image below we also dragged Segment to Color and then edited the colors for aesthetic effect. Now we can see how the average sum of sales per state varies across categories and segments.
EXCLUDE Level of Detail Expressions

EXCLUDE level of detail expressions declare dimensions to omit from the view level of detail.

EXCLUDE level of detail expressions are useful for ‘percent of total’ or ‘difference from overall average’ scenarios. They are comparable to such features as Totals and Reference Lines.

EXCLUDE level of detail expression cannot be used in row-level expressions (where there are no dimensions to omit), but can be used to modify either a view level calculation or anything in between (that is, you can use an EXCLUDE calculation to remove dimension from some other level of detail expression).

Example 1

The following EXCLUDE level of detail expression computes the average sales total per month and then excludes the month component:

\{EXCLUDE [Order Date (Month / Year)] : AVG({FIXED [Order Date (Month / Year)] : SUM([Sales])})\}
Notice that this is a nested level of detail expression—that is, a level of detail expression within another level of detail expression.

Saved as [average of sales by month], the calculation can then be subtracted from the sum of sales per month by means of an ad-hoc calculation on the Rows shelf:

With Month([Order Date]) on the Columns shelf, this creates a view that shows the difference between actual sales per month over a four-year period and the average monthly sales for the entire four-year period:

![Graph showing the difference between actual sales and the average monthly sales]

**Example 2**

The following level of detail expression excludes [Region] from a calculation of the sum of [Sales]:

{EXCLUDE [Region]: SUM([Sales])}

The expression is saved as [ExcludeRegion].

To illustrate how this expression might be useful, first consider the following view, which breaks out the sum of sales by region and by month:
Dropping `[ExcludeRegion]` on Color shades the view to show total sales by month but without the regional component:

Level of Detail Expressions and Aggregation

The level of detail of the view determines the number of marks in your view. When you add a level of detail expression to the view, Tableau must reconcile two levels of detail—the one in the view, and the one in your expression.

The behavior of a level of detail expression in the view varies depending on whether the expression's level of detail is coarser, finer, or the same as the level of detail in the view. What do we mean by “coarser” or “finer” in this case?
Level of Detail Expression is Coarser Than View Level of Detail

An expression has a coarser level of detail than the view when it references a subset of the dimensions in the view. For example, for a view that contained the dimensions [Category] and [Segment], you could create a level of detail expression that uses only one of these dimensions:

{FIXED [Segment] : SUM([Sales])}

In this case, the expression has a coarser level of detail than the view. It bases its values on one dimension ([Segment]), whereas the view is basing its view on two dimensions ([Segment] and [Category]).

The result is that using the level of detail expression in the view causes certain values to be replicated—that is, to appear multiple times.

Replicated values are useful for comparing specific values against average values within a category. For example the following calculation subtracts average sales for a customer from the average sales overall:

[Sales] - {FIXED [Customer Name] : AVG([Sales])}

When values are being replicated, changing the aggregation for the relevant field in the view (for example, from AVG to SUM) will not change the result of the aggregation.
Level of Detail Expression is Finer Than View Level of Detail

An expression has a finer level of detail than the view when it references a superset of the dimensions in the view. When you use such an expression in the view, Tableau will aggregate results up to the view level. For example, the following level of detail expression references two dimensions:

\{\text{FIXED} \ [\text{Segment}], \ [\text{Category}] : \text{SUM}([\text{Sales}])\}

When this expression is used in a view that has only [Segment] as its level of detail, the values must be aggregated. Here’s what you would see if you dragged that expression to a shelf:

\text{AVG}([\text{FIXED} \ [\text{Segment}]], \ [\text{Category}] : \text{SUM}([\text{Sales}]))}

An aggregation—in this case, average—is automatically assigned by Tableau. You can change the aggregation as needed.

Adding a Level of Detail Expression to the View

Whether a level of detail expression is aggregated or replicated in the view is determined by the expression type (FIXED, INCLUDE, or EXCLUDE) and whether the expression’s granularity is coarser or finer than the view’s.

- **INCLUDE** level of detail expressions will have either the same level of detail as the view or a finer level of detail than the view. Therefore, values will never be replicated.

- **FIXED** level of detail expressions can have a finer level of detail than the view, a coarser level of detail, or the same level of detail. The need to aggregate the results of a FIXED level of detail depends on what dimensions are in the view.

- **EXCLUDE** level of detail expressions always cause replicated values to appear in the view. When calculations including EXCLUDE level of detail expressions are placed on a shelf, Tableau defaults to the ATTR aggregation (as opposed to SUM or AVG) to indicate that the expression is not actually being aggregated and that changing the aggregation will have no effect on the view.

Level of detail expressions are always automatically wrapped in an aggregate when they are added to a shelf in the view unless they’re used as dimensions. So if you double-click on a shelf and type
\{\text{FIXED}[\text{Segment}], \ [\text{Category}] : \ \text{SUM}\{[\text{Sales}])\} \}

and then press Enter to commit the expression, what you now see on the shelf is

\text{SUM}\{\{\text{FIXED}[\text{Segment}], \ [\text{Category}] : \ \text{SUM}\{[\text{Sales}])\}\}\}

But if you double-click into the shelf to edit the expression, what you see in edit mode is the original expression.

If you wrap a level of detail expression in an aggregation when you create it, Tableau will use the aggregation you specified rather than assigning one when any calculation including that expression is placed on a shelf. When no aggregation is needed (because the expression’s level of detail is coarser than the view’s), the aggregation you specified is still shown when the expression is on a shelf, but it is ignored.

Filters and Level of Detail Expressions

There are several different kinds of filters in Tableau and they get executed in the following order from top to bottom.

![Filter Diagram]

The text on the right shows where level of detail expressions are evaluated in this sequence.

Extract Filters (in orange) are only relevant if you’re creating a Tableau Extract from a data source. Table calculations filters (dark blue) are applied after calculations are executed and therefore hide marks without filtering out the underlying data used in the calculations.

If you’re familiar with SQL, you can think of measure filters as equivalent to the HAVING clause in a query, and dimension filters as equivalent to the WHERE clause.
FIXED calculations are applied before dimension filters, so unless you promote the fields on your Filter shelf to Improve View Performance with Context Filters on page 1194, they will be ignored. For example, consider if you have the following calculation on one shelf in a view, along with [State] on a different shelf:

\[
\text{SUM}([\text{Sales}]) \div \text{ATTR}({\text{FIXED}} : \text{SUM}([\text{Sales}]))
\]

This calculation will give you the ratio of a state’s sales to total sales.

If you then put [State] on the Filters shelf to hide some of the states, the filter will affect only the numerator in the calculation. Since the denominator is a FIXED level of detail expression, it will still divide the sales for the states still in the view against the total sales for all states—including the ones that have been filtered out of the view.

INCLUDE and EXCLUDE level of detail expressions are considered after Dimension filters. So if you want filters to apply to your FIXED level of detail expression but don’t want to use Context Filters, consider rewriting them as INCLUDE or EXCLUDE expressions.

Data Source Constraints for Level of Detail Expressions

For some data sources, only more recent versions support level of detail expressions. Some data sources do not support level of detail expressions at all.

In addition, some data sources have complexity limits. Tableau will not disable calculations for these databases, but query errors are a possibility if calculations become too complex.

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actian Vector</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Amazon EMR</td>
<td>Supported for Hive 0.13 and later.</td>
</tr>
<tr>
<td>Hadoop Hive</td>
<td>Supported for Hive 0.13 and later.</td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td>Supported.</td>
</tr>
<tr>
<td>Aster Database</td>
<td>Supported for version 4.5 and later.</td>
</tr>
<tr>
<td>Cloudera Hadoop</td>
<td>Supported for Hive 0.13 and later.</td>
</tr>
<tr>
<td>Cloudera Impala</td>
<td>Supported for Impala 1.2.2 and later.</td>
</tr>
<tr>
<td><strong>Cubes (multidimensional data sources)</strong></td>
<td>Not supported.</td>
</tr>
<tr>
<td><strong>DataStax Enterprise</strong></td>
<td>Not supported.</td>
</tr>
<tr>
<td><strong>EXASolution</strong></td>
<td>Supported.</td>
</tr>
<tr>
<td><strong>Firebird</strong></td>
<td>Supported for version 2.0 and later.</td>
</tr>
<tr>
<td><strong>Generic ODBC</strong></td>
<td>Limited. Depends on the specific data source.</td>
</tr>
<tr>
<td><strong>Google Big Query</strong></td>
<td>Supported for standard SQL, not supported for legacy SQL.</td>
</tr>
<tr>
<td><strong>Hortonworks Hadoop Hive</strong></td>
<td>Supported for Hive 0.13 and later. On version 1.1 of HIVE level of detail expressions that produce cross joins are not reliable. Cross join occur when there is no explicit field to join on. For example, for a level of detail expression `{fixed [Product Type] : sum(sales)} when the view only contains one dimension [Ship Mode], Tableau creates a cross-join. A cross join produces rows which combine each row from the first table with each row from the second table.</td>
</tr>
<tr>
<td><strong>IBM BigInsights</strong></td>
<td>Supported.</td>
</tr>
<tr>
<td><strong>IBM DB2</strong></td>
<td>Supported for version 8.1 and later.</td>
</tr>
<tr>
<td><strong>MarkLogic</strong></td>
<td>Supported for version 7.0 and later.</td>
</tr>
<tr>
<td><strong>Microsoft Access</strong></td>
<td>Not supported.</td>
</tr>
<tr>
<td><strong>Microsoft Jet-based connections (legacy connectors for</strong></td>
<td>Not supported.</td>
</tr>
<tr>
<td>Software</td>
<td>Compatibility</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Microsoft Excel,</td>
<td>SQL Server 2005 and later.</td>
</tr>
<tr>
<td>Microsoft Access,</td>
<td></td>
</tr>
<tr>
<td>and text)</td>
<td></td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>SQL Server 2005 and later.</td>
</tr>
<tr>
<td>MySQL</td>
<td>Supported.</td>
</tr>
<tr>
<td>IBM PDA (Netezza)</td>
<td>Supported version 7.0 and later.</td>
</tr>
<tr>
<td>Oracle</td>
<td>Supported version 9i and later.</td>
</tr>
<tr>
<td>Actian Matrix (ParAccel)</td>
<td>Supported version 3.1 and later.</td>
</tr>
<tr>
<td>Pivotal Greenplum</td>
<td>Supported for version 3.1 and later.</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>Supported version 7 and later.</td>
</tr>
<tr>
<td>Progress OpenEdge</td>
<td>Supported.</td>
</tr>
<tr>
<td>SAP HANA</td>
<td>Supported.</td>
</tr>
<tr>
<td>SAP Sybase ASE</td>
<td>Supported.</td>
</tr>
<tr>
<td>SAP Sybase IQ</td>
<td>Supported version 15.1 and later.</td>
</tr>
<tr>
<td>Spark SQL</td>
<td>Supported.</td>
</tr>
<tr>
<td>Splunk</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Tableau Data Extract</td>
<td>Supported.</td>
</tr>
<tr>
<td>Teradata</td>
<td>Supported.</td>
</tr>
<tr>
<td>Vertica</td>
<td>Supported for version 6.1 and later.</td>
</tr>
</tbody>
</table>
Tips for Working with Calculated Fields in Tableau

To help you become more efficient with creating and editing calculated fields in Tableau, this article lists several tips for working in the calculation editor.

In this article

- Drag and drop fields into the calculation editor below
- Drag and drop formulas from the calculation editor to the Data pane on the next page
- Use the functions reference in the calculation editor on the next page
- Take advantage of auto-complete for formulas on page 1616
- Drag table calculations into the calculation editor to edit them on page 1616
- Resize text in the calculation editor on page 1617
- See which sheets are using a calculated field on page 1617

Drag and drop fields into the calculation editor

When creating fields in the calculation editor, you can drag existing fields from the Data pane into the editor at any time.
Drag and drop formulas from the calculation editor to the Data pane

When typing a calculation in the calculation editor, you can highlight all or part of the formula and drag it to the Data pane to create a new calculated field. You can then rename the field by typing a name. For more information, see Ad-Hoc Calculations on page 1618.

Use the functions reference in the calculation editor

When typing a calculation in the calculation editor, you can use the functions reference to browse all the functions available in Tableau.

To open the functions reference:

- In the calculation editor, click the triangle icon on the right-side of the editor.

To add a function from the reference to a formula:
In the function reference, double-click a function.

Take advantage of auto-complete for formulas

As you type a formula in the calculation editor, Tableau suggests options to complete items in your formula. Tableau suggests functions, fields in your data source, parameters, sets, and bins that begin with or contain the string you type. The list of suggestions update as you type.

To add an item from auto-complete to a formula:

- Press Enter on your keyboard to select the highlighted suggestion.

Note: You can use the up and down arrows on your keyboard to move between items in the auto-complete list.

Drag table calculations into the calculation editor to edit them

When you create a table calculation, you can drag it into the calculation editor to review or make changes to the formula.

To edit a table calculation in the calculation editor:
1. Click the Data pane drop-down and select Create Calculated Field.
2. From the worksheet, drag the table calculation into the calculation editor.
3. When finished, click OK.

Resize text in the calculation editor

You can adjust the size of the text in the calculation editor as you create or edit calculations.

To increase text size in the calculation editor:

- Press the CTRL and + keys on your keyboard (Command + on a Mac)

To decrease text size in the calculation editor:

- Press the CTRL and – keys on your keyboard (Command – on a Mac).

Note: Text size persists until you close the editor. The next time you open the editor, text is at the default size.

See which sheets are using a calculated field

As you edit a calculated field, you can click Sheets Affected to see which other sheets are using the field. These sheets will also be updated when you commit your changes.
Ad-Hoc Calculations

Ad-hoc calculations are calculations that you can create and update as you work with a field on a shelf in the view. Ad-hoc calculations are also known as type-in or in-line calculations.

In this article:

- Create an Ad-Hoc Calculation below
- Multi-Line Ad-Hoc Calculations on page 1620
- Ad-Hoc Calculations and Aggregation on page 1620
- Ad-Hoc Calculations for Insight and Experimentation on page 1620
- Managing Ad-Hoc Calculations on page 1620

Create an Ad-Hoc Calculation

Double-click on an existing field to start editing.
You can also double-click on an empty shelf or on an empty part of a shelf to create a new calculation.

<table>
<thead>
<tr>
<th>Columns</th>
<th>YEAR(Order Date)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows</td>
<td></td>
</tr>
</tbody>
</table>

Type to update the expression, or drag new fields into the expression from the Data pane or elsewhere in the view.

<table>
<thead>
<tr>
<th>Columns</th>
<th>MONTH(Order Date)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows</td>
<td>SUM([Profit])/SUM([Sales])</td>
</tr>
</tbody>
</table>

Press Enter, Tab, or click outside the expression to commit the expression, close the calculation, and update the view.

Press Ctrl+Enter (or Command+Enter on a Mac) to commit the expression and update the view without closing the calculation.

<table>
<thead>
<tr>
<th>Columns</th>
<th>YEAR(Order Date)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows</td>
<td>AGG(SUM([Profit])/S..</td>
</tr>
</tbody>
</table>

Ad-hoc calculations are supported on the **Rows**, **Columns**, **Marks**, and **Measure Values** shelves; they are not supported on the **Filters** or **Pages** shelves.

If you are working in Tableau Desktop, errors in ad-hoc calculations are underlined in red. Hover over the error to see a suggestion for resolving it.

**Note:** Ad-hoc calculations are not named, but are saved when you close the workbook. If you want to save an ad-hoc calculation for use in other workbook sheets, copy it to the Data pane. You will be prompted to name the calculation. (On server it just deposits it w/o rename prompt.) Once you name an ad-hoc calculation, it is the same as a calculation you created with the calculation editor, and is available on other sheets in the workbook. See **Get Started with Calculations in Tableau** on page 1221.
Multi-Line Ad-Hoc Calculations

As you're typing an ad-hoc calculation, you can press Shift+Enter to start a new line. However, only the current line is ever visible in an ad-hoc calculation, so this can be confusing for anyone viewing or editing the calculation who has no way of knowing that it contains multiple lines.

The first line of a multi-line ad-hoc calculation can be a comment that serves as a title for the calculation. This is the only line that is visible on the shelf after the calculation is committed:

//City and State

[City] + ' , ' + [State]

Ad-Hoc Calculations and Aggregation

If Tableau determines that the expression you enter is a measure (that is, returns a number), it automatically adds an aggregation to the expression when you commit the expression. For example, if you type DATEDIFF('day',[Ship Date],[Order Date]) in an ad-hoc calculation and then press Enter, what you will see is the following:

SUM(DATEDIFF('day',[Ship Date],[Order Date]))

If you use a field that is already an aggregated field (for example, SUM([Profit])) in an ad-hoc calculation, the result is an aggregate calculation. For example, when you commit an ad-hoc calculation SUM([Profit])/SUM([Sales]), the result is:

AGG(SUM([Profit])/SUM([Sales]))

For more information on aggregate calculations, see Aggregate Functions in Tableau on page 1320.

Ad-Hoc Calculations for Insight and Experimentation

Typically you create ad-hoc calculations on-the-fly to do things like

- Test a hunch
- Try a what-if scenario
- Debug a complex calculation

Managing Ad-Hoc Calculations

Use the following keystrokes to manage ad-hoc calculations.
• Double-click in an existing field on the Rows, Columns, Marks, or Measure Values shelf to open it as an ad-hoc calculation.

• Double-click elsewhere on any of these shelves to create a new ad-hoc calculation from scratch.

• Press Esc to cancel an ad-hoc calculation.

• Press Enter to commit an ad-hoc calculation, which updates the view and closes the ad-hoc calculation. Press Ctrl+Enter to commit the change and update the view without closing the ad-hoc calculation.

• If you are working in Tableau Desktop, and there is a field to the right of the current ad-hoc calculation on the same shelf, pressing Tab opens that field as an ad-hoc calculation. If there is no field to the right of the current ad-hoc calculation on the shelf, pressing Tab opens a new ad-hoc calculation. Shift+Tab has the same functionality, except it moves to the left.

• When you double-click a named field on a shelf to edit it, you are not changing the original named field.

In addition, the following rules govern the use of ad-hoc calculations.

• Only one ad-hoc calculation can be open at a time.

• If a field that has a geographic role or a fiscal year setting associated with it is added to an ad-hoc calculation, the ad-hoc calculation inherits that role or setting.

• The right-click context menu for an ad-hoc calculation supports the same options that would be available for any other field in the view of the same type, including the ability to change aggregation, show a filter, or format.

• Ad-hoc calculations are not available when you create groups, sets, hierarchies, or parameters.

• Ad-hoc calculations are valid for creating trend lines, forecasts, and reference lines, bands, and distributions.

Example - Spotlightting Using Calculations

Spotlighting is a technique for showing discrete thresholds based on the values of a measure. For instance, you might want to color-code sales so that those over 10,000 appear green and those below 10,000 appear red. A spotlighting calculation is just a special case of a calculation that results in a discrete measure. A discrete measure is a calculation that is a dependent
variable (and therefore a measure), but which results in a discrete result (as opposed to a continuous result). Thus the name discrete measure. Here is an example:

![Image of a discrete measure formula]

The formula in this example defines a discrete measure named **Sales Spotlight**. Discrete measures always appear with a blue abc icon in the Data pane. **Sales Spotlight** is classified as a measure in Tableau because it is a function of another measure; it is discrete because it produces discrete values ("Good" and "Bad"). Here is an example of this categorical measure in use:

![Image of a Tableau sheet with a discrete measure]

- 1622 -
Here, **Sales Spotlight** is on Color in the Marks card. It appears with the **AGG** prefix because it is an aggregate calculation. Values above 10,000 and below 10,000 are assigned different colors.

**Pass Expressions to External Services**

Tableau supports a set of functions that you can use to pass expressions to external services for integration with R, MATLAB, and Python.

**In This Article**

**Available External Services** below

**Configure an External Service Connection** on the next page

**Share a Workbook That Requires an External Service Connection** on page 1626

**Publish a Workbook That Requires an External Service Connection** on page 1626

Also see **Troubleshooting External Service Connections** on page 1628.

**Available External Services**

Tableau supports integration with the following services:

- **R**
  
  R is an open source software programming language and a software environment for statistical computing and graphics.

  **Watch a Video:** To see related concepts demonstrated in Tableau, watch the free training videos **How to Integrate R and Tableau** (4:40 minutes) and **Using R with Tableau** (4:44 minutes). Use your **tableau.com** account to sign in.

- **MATLAB**
  
  MATLAB is a high-level language that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran. You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. For information on how to configure your MATLAB server to work with Tableau please contact **MATLAB support**.
To read more about the possibilities of MATLAB in Tableau, see Put your MATLAB models and algorithms to work in Tableau.

- **Python**

  Python is a widely used high-level programming language for general-purpose programming. By sending Python commands to an external service, you can do things like predicting customer churn or running sentiment analysis.

  Tableau Python Server (TabPy) is part of Tableau's expanding range of extensibility options. To install TabPy, visit this GitHub page. To read more about the possibilities of Python in Tableau, see Building advanced-analytics applications with TabPy.

**SCRIPT functions for expressions**

In Tableau Desktop, a set of four SCRIPT functions are available for passing expressions to external services and obtaining a result. The functions are:

- `SCRIPT_BOOL`
- `SCRIPT_INT`
- `SCRIPT_REAL`
- `SCRIPT_STR`

See [SCRIPT_BOOL](#) on page 1356 for details and examples. Because these SCRIPT functions are table calculations functions, addressing and partitioning concepts apply. (For an explanation of these concepts, see The basics: addressing and partitioning on page 1525.) Tableau makes one call to an external service per partition.

Because connecting to an external service involves some overhead, try to pass values as vectors rather than as individual values whenever possible. For example if you set addressing to Cell (for example, by clicking the field in the view and choosing Compute Along > Cell), Tableau will make a separate call per row to the external service; depending on the size of the data, this can result in a very large number of individual calls. If you instead use a column that identifies each row that you would use in the level of detail, you could compute along that column so that Tableau could pass those values in a single call.

**Configure an External Service Connection**

To use the SCRIPT functions, you must establish a connection to an external server.
For R, you need access to a server that allows applications to access R functionality. See Rserve for details. For information on installing, running, and configuring Rserve, as well as on optimizing R scripts and R security, see the Tableau Community post R Implementation Notes. And for more information about using R with Tableau, see the blog post Tableau 8.1 and R.

**Note**: For R integration, Tableau has been tested with R versions 3.1 and 3.2, and with Rserver version 0.6-8.

To configure an external service connection:

1. In Tableau Desktop, click the Help menu, and then select **Settings and Performance** > **Manage External Service connection** to open the External Service Connection dialog box:

2. Enter or select a server name using a domain or an IP address. The drop-down list includes **localhost** and the server you most recently connected to.

3. Specify a port.

   **Note**: Port 6311 is the default port for Rserve servers.

4. If the server requires credentials, specify a Username and Password.

5. Click **Test Connection**.

6. Click **OK**.
If no connection can be established, an error message is displayed. Click **Show Details** in the message to see any diagnostic information returned by the server.

**Share a Workbook That Requires an External Service Connection**

You may need to send a workbook that contains external service functionality to other users, who may be using different copies of Tableau Desktop on other computers. Or, users may download a workbook from Tableau Server that contains external service functionality. Before users will be able to use the external service functionality in workbooks they have received or downloaded to Tableau Desktop, they must configure external service connections on their computers.

**Publish a Workbook That Requires an External Service Connection**

Before you publish a workbook that relies on an external service connection to Tableau Server, you should verify that the server is configured to be able to run scripts, and also configure Tableau Server to have its own external service connection.

**Configure Tableau Server to Be Able to Run Scripts**

If you publish a workbook that relies on an external service connection, you must verify that the server is configured to allow scripts. The following two settings are relevant:

- **vizqlserver.allow_insecure_scripts**
  
  This setting allows affects scripts of all kinds (initial sql, custom sql, script calcs) in published workbooks. The default value is false. Setting it to true allows scripts that include string parameters (that is, scripts that reference a Tableau parameter) to run. Scripts that reference a Tableau parameter can constitute a vulnerability because of the potential for injection of code.

- **vizqlserver.script.disabled**
  
  This parameter determines whether external service calls from a workbook on Tableau Server can run. The default value is true, which seems like it would mean that scripts are disabled. But in fact true means scripts can run. To prevent scripts from running, set this parameter to false. This parameter does not affect the ability to upload workbooks with scripts, only the ability to execute script commands once a workbook is uploaded.

Use the Tableau Services Manager (TSM) command line tool to configure these settings. For details, see **tsm configuration** in the Tableau Server help.
Configure Tableau Server for an External Service Connection

You do this by configuring settings with `tsm configuration set`. The settings are equivalent to the values you set in the External Service Connection dialog box. They are:

- `vizqlserver.extsvc.host`
- `vizqlserver.extsvc.port`
- `vizqlserver.extsvc.username`
- `vizqlserver.extsvc.password`

An additional setting, `vizqlserver.extsvc.connect_timeout_ms`, is available for extending the timeout value, in milliseconds, for connections to Rserve. Raise the value of this setting if Tableau is timing out before the server can respond.

For more information on these settings, see `tsm configuration set options` in the Tableau Server help.

**Note:** If you configured Tableau Server for an external service connection prior to Tableau 10.0, these settings had different names. The configuration `vizqlserver.extsvc` was introduced in Tableau Server 10.1 as a replacement for `vizqlserver.rserve`. However, there is reverse compatibility in Tableau Server starting with version 10.1. In these more recent versions of Tableau Server, `vizqlserver.extsvc` settings will be checked for first, but if there are no `vizqlserver.extsvc` settings then Tableau Server will check for `vizqlserver.rserve` settings.

The settings `vizqlserver.extsvc.username` and `vizqlserver.extsvc.password` should be omitted if Tableau Desktop connects to Rserve without a username and password.

For information on how to use the above `tsm configuration set options` to configure Tableau Server, see `Change Tableau Server’s Configuration from the Command Line`.

Tableau cannot verify that workbooks that use an external service will render properly on Tableau Server. There might be scenarios where a required statistical library is available on a user’s machine but not on the external service instance that Tableau Server is using.

For views that cannot be rendered in Tableau Server because of a script error, you see a warning error when you publish the workbook:
This worksheet contains external service scripts, which cannot be viewed on the target platform until the administrator configures an external service connection.

You cannot publish a workbook that contains external service scripting to Tableau Online.

Because Tableau Server provides an authentication mechanism, it can be more secure to expose external service functionality to users through Tableau Server than in Tableau Desktop.

Troubleshooting External Service Connections

This topic describes errors you can receive when Tableau is connected to an external service.

Note: Tableau technical support cannot assist with writing, troubleshooting, or debugging external scripts.

In This Article

Errors That Can Occur When Communicating with Any External Service below

Errors that Can Occur Only with an Rserve Server on page 1632

Other Issues on page 1634

Errors That Can Occur When Communicating with Any External Service

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Comments</th>
</tr>
</thead>
</table>
| Unsupported type passed as an argument to SCRIPT function. | Tableau can only export the following Tableau data types to R:  
  - Number (Decimal)  
  - Number (Whole)  
  - Boolean  
  - String |
<table>
<thead>
<tr>
<th>Unexpected number of results returned by SCRIPT function. Function expected %2 values; %1 values were returned.</th>
<th>The script result must be either a scalar or vector of length one that is replicated for all rows, or a vector of length equal to the number of rows in the Tableau result table.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The result returned by the SCRIPT function is of an unexpected type.</td>
<td>Occurs when an invalid data type is received. Tableau can only import the following data types from an Rserve server:</td>
</tr>
<tr>
<td></td>
<td>• RDouble</td>
</tr>
<tr>
<td></td>
<td>• RInteger</td>
</tr>
<tr>
<td></td>
<td>• RLogical</td>
</tr>
<tr>
<td></td>
<td>• RCharacter</td>
</tr>
<tr>
<td></td>
<td>For TabPy, Tableau can import the following data types:</td>
</tr>
<tr>
<td></td>
<td>• List of floating point numbers</td>
</tr>
<tr>
<td></td>
<td>• List of integers</td>
</tr>
<tr>
<td></td>
<td>• List of Booleans</td>
</tr>
<tr>
<td></td>
<td>• List of strings</td>
</tr>
<tr>
<td></td>
<td>This error is also reported if the result is null or if there was a script execution error for which Tableau could not collect an explanation from the external server.</td>
</tr>
<tr>
<td>This worksheet contains external service scripts, which are not supported on the target platform.</td>
<td>This error is reported when you attempt to publish a worksheet containing an external service call to a Tableau server that does not allow external scripts because <code>vizqlserver.script.disabled</code> is set to <code>false</code>.</td>
</tr>
<tr>
<td>This worksheet contains external service scripts, which cannot be viewed on the target platform until the administrator configures an external service connection.</td>
<td>This error is reported when the server is configured to prevent Desktop from publishing worksheets with external service scripts because <code>vizqlserver.script.disabled</code> is set to false. (The setting is named contrary to its meaning: true means Desktop can publish worksheets with external service scripts, false means Desktop cannot publish worksheets with external service scripts.) Setting <code>vizqlserver.script.disabled</code> to false will not prevent Tableau Online users from using <code>tabcmd</code> to publish workbooks with external service scripts; however, the resulting views will generate this error when opened in a browser.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>An error occurred while communicating with the external service.</td>
<td>Tableau runs all external scripts inside of the &quot;try&quot; external function. This error is displayed along with an external-service-generated error message when the “try” function traps an evaluation error.</td>
</tr>
<tr>
<td>This external service connection does not support authentication. Try connecting without specifying a password.</td>
<td></td>
</tr>
<tr>
<td>Authentication failed. Please provide a valid external service username and password.</td>
<td></td>
</tr>
<tr>
<td>An unsupported authentication type is enabled in the external service.</td>
<td>Either disable external service authentication or change to plaintext password authentication.</td>
</tr>
<tr>
<td>No external service connection configured. Specify a server name and try again.</td>
<td>See <em>Pass Expressions to External Services</em> on page 1623.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>The calculation &quot;%1&quot; contains a SCRIPT_ function that requires an internal service connection. Configure your external service connection to enable custom scripts.</td>
<td>See <em>Pass Expressions to External Services</em> on page 1623.</td>
</tr>
<tr>
<td>The workbook you are attempting to publish contains SCRIPT_ functions that require an external service connection. Custom scripts are not allowed in public workbooks.</td>
<td>You cannot publish workbooks containing external service scripts to Tableau Public.</td>
</tr>
<tr>
<td>Tableau Public does not support running custom scripts that require external services. To take advantage of external services, upgrade to Tableau</td>
<td>The &quot;Tableau Public&quot; in this error refers to Tableau Desktop Public.</td>
</tr>
</tbody>
</table>
Desktop Professional Edition.

Tableau Reader does not support running custom scripts that require external services. To take advantage of external services, upgrade to Tableau Desktop Professional Edition.

The external service is busy or not responding.

Tableau has timed out a read pending on connection to Rserve – the default timeout is 250ms. An IPC connection Read of the Rserve protocol header has thrown an exception.

A tabadmin set setting, vizqlserver.esxtsvc.connect_timeout_ms, is available for extending the timeout value, in milliseconds, for connections to Microsoft’s RServer. Raise the value of this setting if Tableau is timing out before the server can respond. For more information, see tabadmin set options in the Tableau Server help.

Errors that Can Occur Only with an Rserve Server

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unrecognized Rserve signature.</td>
<td>The Rserve header signature must be &quot;Rsrv&quot;.</td>
</tr>
<tr>
<td>Unrecognized</td>
<td>The Rserve header version must be &quot;0103&quot;.</td>
</tr>
<tr>
<td>Rserve version.</td>
<td>The Rserve header protocol must be &quot;QAP1&quot;.</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>Unrecognized Rserve protocol.</td>
<td>The transport protocol type when reading a result was not SEXP as expected.</td>
</tr>
<tr>
<td>Authentication failure when connecting to R.</td>
<td>Tableau attempted and failed to authenticate with Rserve. Verify that you entered a valid password.</td>
</tr>
<tr>
<td>Incorrect number of bytes in parameter/body.</td>
<td></td>
</tr>
<tr>
<td>The length of data which Tableau read does not equal the length promised by the header.</td>
<td></td>
</tr>
<tr>
<td>Unrecognized response type.</td>
<td></td>
</tr>
<tr>
<td>Excessively long &lt;type&gt;vector.</td>
<td>Unreasonably large number measuring the length of data sent to or from Rserve possibly indicating a corrupt protocol header.</td>
</tr>
<tr>
<td>Invalid Rserve command.</td>
<td>Tableau may have improperly implemented the Rserve protocol.</td>
</tr>
<tr>
<td>Response from server was Error &quot;&lt;&lt; (uint32_t)status &lt;&lt; &quot;. See Rsrv.h for details.</td>
<td>Various error conditions are documented in the comments in Rsrv.h.</td>
</tr>
<tr>
<td>Excessively long header offset.</td>
<td>This may be due to a garbled header with an unreasonable offset to response data.</td>
</tr>
</tbody>
</table>
Other Issues

SCRIPT Functions Run Even in Logical Statements That Evaluate as False

A function that sends an expression to a running external service instance will be executed even when it is within a logical statement that would otherwise prevent it from being executed. This is true for logical functions such as IF, IIF, and CASE. For example:

IF 1==0 THEN

[[R script code]]

ELSE

"1 does not equal 0"

END

Spot Trends

Reference Lines, Bands, Distributions, and Boxes

You can add a reference line, band, distribution, or box plot to identify a specific value, region, or range on a continuous axis in a Tableau view. For example, if you are analyzing the monthly sales for several products, you can include a reference line at the average sales mark so you can see how each product performed against the average.

Tableau lets you add as many reference lines, bands, distributions, and box plots to a view as you require.
In this article

**Types of Reference Lines, Bands, Distributions, and Boxes**

- Add a Reference Line on page 1639
- Add Reference Bands on page 1646
- Add Reference Distributions on page 1651
- Add a Bullet Graph on page 1653
- Add a Box Plot on page 1656
- Edit Existing Reference Lines, Bands, and Distributions on page 1660
- Remove Reference Lines, Bands, or Distributions on page 1661

**Types of Reference Lines, Bands, Distributions, and Boxes**

You can add reference lines, bands, distributions, or (in Tableau Desktop but not on the web) box plots to any continuous axis in the view.

- **Reference Lines** - You can add a reference line at a constant or computed value on the axis. Computed values can be based on a specified field. You can also include confidence intervals with a reference line.
Reference Bands - Reference bands shade an area behind the marks in the view between two constant or computed values on the axis.
• **Reference Distributions** - Reference distributions add a gradient of shading to indicate the distribution of values along the axis. Distribution can be defined by percentages, percentiles, quantiles (as in the following image), or standard deviation.
Reference distributions can also be used to create bullet charts. See Add a Bullet Graph on page 1653 later in this article for specifics.

- **Box Plots** - Box plots (also known as box and whisker charts) are a standardized graphic for describing the distribution of values along an axis. Box plots show quartiles (also known as hinges) and whiskers. Tableau provides different box plot styles, and allows you to configure the location of the whiskers and other details.
Add a Reference Line

You can add a reference line to any continuous axis in the view.

**To add a reference line:**

1. Drag **Reference Line** from the **Analytics** pane into the view. Tableau shows the possible destinations. The range of choices varies depending on the type of item and the current view.

   In a simple case, the drop target area offers three options:
The view above is from a web editing session. In Tableau Desktop, the process is the same but the user interface looks a bit different. The terms **Table**, **Pane** and **Cell** define the scope for the item:

For a more complicated view—for example, if the view contains a line chart with multiple or dual axes—Tableau shows you an expanded drop target area:
If you drop the item in one of the three larger boxes in the header (for example, the Table box), a separate reference line is added for each continuous field in the view:

But if you drop the item in any of the lower boxes that are aligned with a specific continuous field, the line is added on the corresponding axis, with the specified scope.

When you drop the line in the target area, Tableau displays a dialog box:
2. The **Line** option is already selected at the top of the dialog box.

3. Select a continuous field from the **Value** field to use as the basis for your reference line. You can also select a parameter.

   You cannot select a continuous field that isn't currently in the view as the basis for your reference line. If you want to use such a continuous field, do the following:

   a. Drag the continuous field from the Data pane to the Details target on the Marks card.

   b. Change the continuous field's aggregation if necessary.
This will not change the view, but it will allow you to use that continuous field as the basis for your reference band.

c. Click on the reference line in the view and choose Edit to re-open the Edit Line dialog box.

4. Select an aggregation. The aggregations that are displayed depend on the continuous field you select:

- **Total** - places a line at the aggregate of all the values in either the cell, pane, or the entire view. This option is particularly useful when computing a weighted average rather than an average of averages. It is also useful when working with a calculation with a custom aggregation. The total is computed using the underlying data and behaves the same as selecting one of the totals option the Analysis menu.
- **Sum** - places a line at the SUM of all the values in either the cell, pane, or entire view.
- **Constant** - places a line at the specified value on the axis.
- **Minimum** - places a line at the minimum value.
- **Maximum** - places a line at the maximum value.
- **Average** - places a line at the average value along the axis.
- **Median** - places a line at the median value.

5. Select how you want to label the line:

- **None** - select this option to not show a label for the reference line.
- **Value** - select this option to show a label corresponding to the line's value on the axis.
- **Computation** - select this option to display the name of the continuous field that is the basis for your reference line and any computation that is performed.
- **Custom** - select this option to build a custom label in the text box. You can use the menu to the right of the text box to insert values such as the computation or the value. You can also type text directly into the box, so you could create a value such as `<Field Name> = <Value>`.

6. Specify whether to display the line with a confidence interval, just the line, or just the
confidence interval.

Confidence interval distribution bands shade the region in which the population average will fall \( n \) of the time, where \( n \) is the value you select in the drop-down on the right. You can choose one of the listed numeric values or select a parameter:

![Image showing the dropdown menu for confidence interval settings]

The higher the value you select, the wider the bands will be.

7. In Tableau Desktop, you can also specify formatting options for the line.
8. Optionally, add a fill color above and below the line.

When you are displaying a line and a confidence interval, the shading will be darker within the confidence interval, and lighter beyond it:
When you are displaying a confidence interval without a line, the fill colors are disregarded, though your settings are retained and then applied if you decide later to show a line.

9. Specify whether to **Show recalculated line for highlighted or selected data points**.
   For more information, see **Compare marks data with recalculated lines** on page 2600.

Add Reference Bands

Reference bands are shaded areas behind the marks in the view between two constant or computed values on the axis. You can add reference bands to any continuous axis in the view.

**To add a reference band:**

1. Drag **Reference Band** from the **Analytics** pane into the view. Tableau shows the possible destinations. The range of choices varies depending on the type of item and the
current view.

In a simple case, the drop target area would offer just three options:

The terms **Table**, **Pane**, and **Cell** define the scope for the item:

For a more complicated view—for example, if the view contains multiple or dual axes—Tableau shows you an expanded drop target area that looks like this:
If you drop the item in one of the three larger boxes in the header (for example, the Table box), a separate set of bands is added for each continuous field in the view:

But if you drop the item in any of the lower boxes aligned with a specific continuous field, the band is added on the corresponding axis, with the specified scope.

When you drop the band in the target area, Tableau displays a dialog box:
2. The Band area is already selected at the top of the dialog box.

3. Select two continuous fields to use as the basis for your reference band one in each Value field. You can also select a parameter from the drop-down lists. Do not select the same continuous field and aggregation in both areas.

You cannot select a continuous field that isn't currently in the view as the basis for your reference band. If you want to use such a continuous field, do the following:
a. Drag the continuous field from the Data pane to the Details target on the Marks card.

b. Change the continuous field's aggregation if necessary.
   This will not change the view, but it will allow you to use that continuous field as the basis for your reference band.

c. Click on the reference band in the view and choose Edit to re-open the Edit Band dialog box, and select the continuous field in in the Value (From) area and one in the Value (To) area.

4. Select a computation for each value. The aggregations that are displayed depend on the continuous field you select:
   - **Total** - extends the band to a value that is at the aggregate of all the values in either the cell, pane, or the entire view. This option is particularly useful when computing a weighted average rather than an average of averages. It is also useful when working with a calculation with a custom aggregation. The total is computed using the underlying data and behaves the same as selecting one of the totals option the Analysis menu.
   - **Sum** - extends the band to a value that is at the SUM of all the values in either the cell, pane, or entire view.
   - **Constant** - extends the band to a value that is at the specified value on the axis.
   - **Minimum** - extends the band to a value that is at the minimum value.
   - **Maximum** - extends the band to a value that is at the maximum value.
   - **Average** - extends the band to a value that is at the average value along the axis.
   - **Median** - extends the band to a value that is at the median value.

5. Select how you want to label the bands:
   - **None** – select this option to not show a label for the reference band.
   - **Value** – select this option to show a label corresponding to the band’s value on the axis.
   - **Computation** – select this option to display the name of the continuous field that is the basis for your reference band and any computation that is performed.
   - **Custom** – select this option to build a custom label in the text box. You can use the
menu to the right of the text box to insert values such as the computation or the value. You can also type text directly into the box, so you could create a value such as `<Field Name> = <Value>`.

3. In Tableau Desktop, you can also specify formatting options for the bands. You can mark the two values with a line or select a shading color for the band.

<table>
<thead>
<tr>
<th>Formatting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line:</td>
</tr>
<tr>
<td>Fill:</td>
</tr>
</tbody>
</table>

4. Specify whether to **Show recalculated line for highlighted or selected data points**. For more information, see *Compare marks data with recalculated lines* on page 2600.

Add Reference Distributions

When you add a reference distribution, you specify one, two, or more values. With one value, the result is a line; with two or more values the result is a set of one, two, or more bands.

**To add a reference distribution:**

1. Drag **Distribution Band** from the **Analytics** pane into the view. Tableau shows the possible destinations. The range of choices varies depending on the type of item and the current view.

2. Select a scope for the distribution. The terms **Table**, **Pane**, and **Cell** define the scope for the item:
3. Select the computation that will be used to create the distribution:

- **Percentages** - shades the interval between the specified percentage values. Use a comma to separate two or more percentage values (for example, 60, 80), and then specify which measure and aggregation to use for the percentages.

- **Percentiles** - shades intervals at the specified percentiles. Choose *Enter a value* from the Value drop-down list, and then enter two or more numerical values, delimited by commas (for example, 60, 80 or 25, 50, 75).

- **Quantiles** - breaks the view into the specified number of tiles using shading and lines. When you select this computation, you must also specify the number of tiles (from 3 to 10, inclusive). For example, if you select 3, Tableau calculates the
boundaries between the first, second and third terciles by calling the general quantile function and asking for the 33.33 and the 66.66 quantiles. It then shades the three terciles differently.

Tableau uses estimation type 7 in the R standard to compute quantiles and percentiles.

- **Standard Deviation** - places lines and shading to indicated the specified number of standard deviations above and below the mean. When you select this option you must specify the factor, which is the number of standard deviations and whether the computation is on a sample or the population.

4. Specify how you want to label the distribution bands:

- **None** – select this option to not show a label for the distribution bands.
- **Value** – select this option to show a label corresponding to each distribution band's value on the axis.
- **Computation** – select this option to display the name of the continuous field that is the basis for your distribution bands and any computation that is performed.
- **Custom** – select this option to build a custom label in the text box. You can use the menu to the right of the text box to insert values such as the computation or the value. You can also type text directly into the box, so you could create a value such as `<Field Name> = <Value>`.

5. Specify whether to **Show recalculated band for highlighted or selected data points**. For more information, see **Compare marks data with recalculated lines** on page 2600 in the Tableau Desktop online help.

Add a Bullet Graph

Reference distributions can also be used to create bullet graphs. A bullet graph is a variation of a bar graph developed to replace dashboard gauges and meters. The bullet graph is generally used to compare a primary measure to one or more other measures in the context of qualitative ranges of performance such as poor, satisfactory, and good. You can create a bullet graph by adding a distribution to indicate the qualitative ranges of performance, and a line to indicate the target. The following procedure uses Show Me to make this process easier.

1. Select one or more dimensions, and two measures in the **Data** pane. The bullet graph will compare measure values. For example, budget vs. actual; actual vs. target; etc. Select multiple fields in the **Data** pane by holding down the Ctrl key as you click fields. If
you are using the Superstore sample workbook, you can select the fields show below:

2. Click the **Show Me** button in the toolbar.

3. Select **Bullet Graph** in the Show Me pane.
Tableau adds a reference distribution that is defined at 60% and 80% of the Average of the measure on Detail. It also adds a reference line that marks the Average of that same measure. The other measure is placed on the Rows shelf.
You can edit either of these to change its definition. For example, you may want to add 100% to the set of distribution band values, or draw a line at a constant value. Click on the outer edge or a distribution band, or on the line, and choose **Edit**.

**Add a Box Plot**

In Tableau Desktop, but not on the web, you can add box plots to a continuous axis.

Use box plots, also known as box-and-whisker plots, to show the distribution of values along an axis.

Boxes indicate the middle 50 percent of the data (that is, the middle two quartiles of the data's distribution).

You can configure lines, called *whiskers*, to display all points within 1.5 times the interquartile range (in other words, all points within 1.5 times the width of the adjoining box), or all points at the maximum extent of the data, as shown in the following image:

![Boxplot Example](image)

Boxplots are also available from the Show Me pane when you have at least one measure in the view:
For information on Show Me, see Use Show Me to Start a View on page 2170

To add a box plot:

1. Right-click (Control-click on a Mac) on a quantitative axis and select Add Reference Line.
2. In the Add Reference Line, Band, or Box dialog box, select **Box Plot**.
3. Under Plot Options, specify placement for the whiskers:

   - **Data within 1.5 times the IQR** - places whiskers at a location that is 1.5 times the interquartile range—that is, 1.5 times further out than the width of the adjoining box. This is also known as a schematic box plot.

   - **Maximum extent of the data** - places whiskers at the farthest data point (mark) in the distribution. This is also known as a skeletal box plot.

4. Specify whether to **Hide underlying marks (except outliers)**—that is, whether to hide all marks except those beyond the whiskers.

5. Configure the appearance of the plot by selecting a **Style**, **Fill**, **Border**, and **Whiskers**.
Box Plot Alternatives: Show Me Vs. Add Reference Line, Band, or Box

The difference between adding a box plot using Show Me and adding a box plot using Add Reference Line is that with Show Me, the box plot is your visualization, whereas with Add Reference Line, Band, or Box, you are adding a box plot to an existing visualization. For example, you could create the following view by first selecting a circle view in Show Me, and then adding a box plot from Add Reference Line:

Edit Existing Reference Lines, Bands, and Distributions

You can edit existing lines, bands, or distributions. To do this, click on a line or on the outer edge of a band and choose **Edit** to reopen the edit dialog box for that object.
Remove Reference Lines, Bands, or Distributions

To remove a reference line, band, or distribution, click on a line or on the outer edge of a band and choose Remove. You can also drag a line or band off the view.

Add Trend Lines to a Visualization

You can show trend lines in a visualization to highlight trends in your data. You can publish a view that contains trend lines, and you add trend lines to a view as you edit it on the web.

When you add trend lines to a view, you can specify how you want them to look and behave.

For a 5-minute walkthrough, see the Trend Lines free training video. Use your tableau.com account to sign in. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

In this article

Add trend lines
Edit trend lines
Remove trend lines
See a description of a trend line
Trend line model types
Trend line model terms
Assess trend line significance
Commonly asked questions

Add trend lines to a view

To add a trend line to a visualization:

1. Select the Analytics pane.

2. From the Analytics pane, drag Trend Line into the view, and then drop it on the Linear, Logarithmic, Exponential, Polynomial, or Power model types.

For more information on each of these model types, see Trend Line Model Types on page 1669.
About adding trend lines (and when you can't add them)

To add trend lines to a view, both axes must contain a field that can be interpreted as a number. For example, you cannot add a trend line to a view that has the Product Category dimension, which contains strings, on the Columns shelf and the Profit measure on the Rows shelf. However, you can add a trend line to a view of sales over time because both sales and time can be interpreted as numeric values.

For multidimensional data sources, the date hierarchies actually contain strings rather than numbers. Therefore, trend lines are not allowed. Additionally, the 'm/d/yy' and 'mmmm yyyy' date formats on all data sources do not allow trend lines.

If you have trend lines turned on and you modify the view in a way where trend lines are not allowed, the trend lines do not show. When you change the view back to a state that allows trend lines, they reappear.

Tableau automatically stacks bar marks in many cases. However, trend lines cannot be turned on for stacked bars. You can turn off stacked marks by clearing the Analysis > Stack Marks option.
Edit a trend line

Once you add a trend line to the visualization, you can edit it to fit your analysis.

To edit a trend line:

**In Tableau Desktop**: Right-click a trend line in the visualization, and select **Edit Trend Lines**.

**In web editing mode**:

1. In the visualization, click the trend line, and then hover your cursor over it.
2. In the tooltip that appears, select **Edit** to open the **Trend Line Options** dialog box.

**Note**: To edit a trend line in Tableau Online or Tableau Server, you must have web editing permissions.

You can configure the following options in the Trend Line Options dialog box:
- Select a model type. For more information, see **Trend Line Model Types** on page 1669.

- Select which fields to use as factors in the trend line model. For more information, see **Choose which fields to use as factors in the trend line model** below.

- Decide whether to exclude color, using the **Allow a trend line per color** option. When you have color encodings in your view, you can use this option to add a single trend line that models all of the data, ignoring the color encoding.

- Decide whether to **Show Confidence Bands**. Tableau confidence bands show upper and lower 95% confidence lines by default when you add trend lines. Confidence lines are not supported for Exponential models.

- Select whether to **Force the y-intercept to zero**. This option is useful when you know that you want your trend line to begin at zero. This option is available only when both the **Rows** shelf and the **Columns** shelf contain a continuous field, as with a scatterplot.

- Decide whether to **show recalculated lines** when you select or highlight data in the visualization.

**Choose which fields to use as factors in the trend line model**

For trend models that are considering multiple fields, you can eliminate specific fields as factors in the trend line model.

Often you will want to remove factors because you want the trend line model to be based on the entire row in the table rather than broken up by the members or values of a field. Consider the following example. The view below shows the monthly sales for various products categories, broken out by region.
You can see that a separate model is created for each region.

Now remove Region as a factor in the model by deselecting it in the Trend Lines Options dialog box.
You can see that the trend line model within a category is now the same across all regions. This allows you to compare actual sales against a trend line that is the same for all regions.
Remove Trend Lines

To remove a trend line from a visualization, drag it off of the visualization area. You can also click a trend line and select Remove.

To remove all trend lines from the view, select Analysis > Trend Lines > Show Trend Lines.

Note: In Tableau Desktop, trend line options are retained so that if you choose Show Trend Lines again from the Analysis menu, the options are as you last set them. However, if you close the workbook with trend lines turned off, trend line options revert to defaults.

See a description of a trend line or trend line model

After you add trend lines, you can display statistics on the trend line. For example, you can see the formula as well as r-squared and p values. For more information on the model types and terms used in the descriptions, see the Trend Line Model Terms on page 1671 and Trend Line Model Types on page 1669 sections.

To see a description of a trend line:
- Hover over any part of a trend line to see its description.

Tableau Desktop only
- Right-click the trend line in the visualization, and then select Describe Trend Line.

To view a full description of the model being used in the current view:
- Right-click a trend line in the visualization, and then select Describe Trend Model.
Trend Line Model Types

These model types are available for trend lines: Linear below, Logarithmic below, Exponential on the next page, Power on the next page, and Polynomial on page 1671.

In the following formulas, X represents the explanatory variable, and Y the response variable.

**Linear**

With the linear model type the formula is:

\[ Y = b0 + b1 \times X \]

where \( b1 \) is the slope and \( b0 \) is the intercept of the line.

**Logarithmic**

With the logarithmic model type, the formula is:

\[ Y = b0 + b1 \times \ln(X) \]
Because a logarithm is not defined for number less than zero, any marks for which the explanatory variable is negative are filtered before estimation of the model. Avoid using a model that discards some data unless you know that the data being filtered out is invalid. The trend line description reports how many marks were filtered before model estimation.

**Exponential**

With the exponential model type, the formula is:

\[ Y = \exp(b_0) \times \exp(b_1 \times X) \]

With an exponential model, the response variable is transformed by the natural log before estimation of the model so the marks plotted in your view are found by plugging in various explanatory values to find values of \( \ln(Y) \).

\[ \ln(Y) = b_0 + b_1 \times X \]

These values are then exponentiated to plot the trend line. What you see is the exponential model in the following form:

\[ Y = b_2 \times \exp(b_1 \times X) \]

Where \( b_2 \) is the value of \( \exp(b_0) \). Because a logarithm is not defined for numbers less than zero, any marks for which the response variable is negative are filtered before model estimation.

**Power**

With the power model type, the formula is:

\[ Y = b_0 \times X^{b_1} \]

With a power model, both variables are transformed by the natural log before estimation of the model resulting in this formula:

\[ \ln(Y) = \ln(b_0) + b_1 \times \ln(X) \]

These values are then exponentiated to plot the trend line.

Because a logarithm is not defined for numbers less than zero, any marks for which the response variable or explanatory variable is negative are filtered before model estimation.
Polynomial

With the polynomial model type, the response variable is transformed into a polynomial series of the specified degree. The formula is:

\[ Y = b0 + b1 \times X + b2 \times X^2 + \ldots \]

With a polynomial model type, you must also select a **Degree** between 2 and 8. The higher polynomial degrees exaggerate the differences between the values of your data. If your data increases very rapidly, the lower order terms may have almost no variation compared to the higher order terms, rendering the model impossible to estimate accurately. Also, more complicated higher order polynomial models require more data to estimate. Check the model description of the individual trends line for a red warning message indicating that an accurate model of this type is not possible.

---

Trend Line Model Terms

When you view the description for a trend line model, there are several values listed. This section discusses what each of these values means.

**Model formula**

This is the formula for the full trend line model. The formula reflects whether you have specified to exclude factors from the model.

**Number of modeled observations**

The number of rows used in the view.

**Number of filtered observations**

The number of observations excluded from the model.

**Model degrees of freedom**

The number of parameters needed to completely specify the model. Linear, logarithmic, and exponential trends have model degrees of freedom of 2. Polynomial trends have model degrees of freedom of 1 plus the degree of the polynomial. For example a cubic trend has model degrees of freedom of 4, since we need parameters for the cubed, squared, linear and constant terms.
Residual degrees of freedom (DF)
For a fixed model, this value is defined as the number of observations minus the number of parameters estimated in the model.

SSE (sum squared error)
The errors are the difference between the observed value and the value predicted by the model. In the Analysis of Variance table, this column is actually the difference between the SSE of the simpler model in that particular row and the full model, which uses all the factors. This SSE also corresponds to the sum of the differences squared of the predicted values from the smaller model and the full model.

MSE (mean squared error)
The term MSE refers to "mean squared error" which is the SSE quantity divided by its corresponding degrees of freedom.

R-Squared
R-squared is a measure of how well the data fits the linear model. It is the ratio of the variance of the model's error, or unexplained variance, to the total variance of the data.

When the y-intercept is determined by the model, R-squared is derived using the following equation:

$$1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

When the y-intercept is forced to 0, R-squared is derived using this equation instead:

$$1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} y_i^2}$$

In the latter case, the equation will not necessarily match Excel. This is because R-squared is not well defined in this case, and Tableau’s behavior matches that of R instead of that of Excel.
Note: The R-Squared value for a linear trend line model is equivalent to the square of the result from the CORR function. See Tableau Functions (Alphabetical) on page 1449 for syntax and examples for CORR.

Standard error

The square root of the MSE of the full model. An estimate of the standard deviation (variability) of the "random errors" in the model formula.

p-value (significance)

The probability that an F random variable with the above degrees of freedom exceeds the observed F in this row of the Analysis of Variance table.

Analysis of Variance

This table, also known as the ANOVA table, lists information for each factor in the trend line model. The values are a comparison of the model without the factor in question to the entire model, which includes all factors.

Individual trend lines

This table provides information about each trend line in the view. Looking at the list you can see which, if any, are the most statistically significant. This table also lists coefficient statistics for each trend line. A row describes each coefficient in each trend line model. For example, a linear model with an intercept requires two rows for each trend line. In the Line column, the p-value and the DF for each line span all the coefficient rows. The DF column under the shows the residual degrees of freedom available during the estimation of each line.

Terms

The name of the independent term.

Value

The estimated value of the coefficient for the independent term.
StdErr
A measure of the spread of the sampling distribution of the coefficient estimate. This error shrinks as the quality and quantity of the information used in the estimate grows.

t-value
The statistic used to test the null hypothesis that the true value of the coefficient is zero.

p-value
The probability of observing a t-value that large or larger in magnitude if the true value of the coefficient is zero. So, a p-value of .05 gives us 95% confidence that the true value is not zero.

Assess Trend Line Significance
To see relevant information for any trend line in the view, hover the cursor over it:

The first line in the tooltip shows the equation used to compute a value of Profit from a value of Year of Order Date.

The second line, the R-Squared value, shows the ratio of variance in the data, as explained by the model, to the total variance in the data. For more details, see Trend Line Model Terms on page 1671.
The third line, the P-value, reports the probability that the equation in the first line was a result of random chance. The smaller the p-value, the more significant the model is. A p-value of 0.05 or less is often considered sufficient.

**Entire Model Significance**

Once you’ve added a trend line to a view, you typically want to know the goodness of fit of the model, which is a measure of the quality of the model's predictions. In addition, you may be interested in the significance of each factor contributing to the model. To view these numbers, open the Describe Trend Model dialog box, right-click (Control-click on a Mac) in the view and select Trend Lines >Describe Trend Model.

When you are testing significance, you are concerned with the p-values. The smaller the p-value, the more significant the model or factor is. It is possible to have a model that has statistical significance but which contains an individual trend line or a term of an individual trend line that does not contribute to the overall significance.

Under Trend Lines Model, look for the line that shows the p-value (significance) of the model: The smaller the p-value, the less likely it is that the difference in the unexplained variance between models with and without the relevant measure or measures was a result of random chance.

![Describe Trend Model](image)

This p-value for a model compares the fit of the entire model to the fit of a model composed solely of the grand mean (the average of data in the data view). That is, it assesses the explanatory power of the quantitative term f(x) in the model formula, which can be linear, polynomial, exponential, or logarithmic with the factors fixed. It is common to assess
significance using the "95% confidence" rule. Thus, as noted above, a p-value of 0.05 or less is considered good.

Significance of Categorical Factors

In the Analysis of Variance table, sometimes referred to as the ANOVA table, each field that is used as a factor in the model is listed. For each field, among other values, you can see the p-value. In this case, the p-value indicates how much that field adds to the significance of the entire model. The smaller the p-value the less likely it is that the difference in the unexplained variance between models with and without the field was a result of random chance. The values displayed for each field are derived by comparing the entire model to a model that does not include the field in question.

The following image shows the Analysis of Variance table for a view of quarterly sales for the past two years of three different product categories.

<table>
<thead>
<tr>
<th>Field</th>
<th>DF</th>
<th>SSE</th>
<th>MSE</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>160</td>
<td>1.65811e+012</td>
<td>1.03632e+010</td>
<td>20.9732</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Region</td>
<td>136</td>
<td>2.64345e+012</td>
<td>1.94371e+010</td>
<td>39.3372</td>
<td>&lt;0.0001</td>
</tr>
</tbody>
</table>

As you can see, the p-values for **Category** and **Region** are both quite small. Both of these factors are statistically significant in this model.

For information on specific trend line terms, see Trend Line Model Terms on page 1671.

For ANOVA models, trend lines are defined by the mathematical formula:

\[ Y = \text{factor 1} \ast \text{factor 2} \ast \ldots \text{factorN} \ast f(x) + e \]

The term \( Y \) is called the *response variable* and corresponds to the value you are trying to predict. The term \( x \) is the *explanatory variable*, and \( e \) (epsilon) is random error. The factors in the expression correspond to the categorical fields in the view. In addition, each factor is represented as a matrix. The \( \ast \) is a particular kind of matrix multiplication operator that takes two matrices with the same number of rows and returns a new matrix with the same number of rows. That means that in the expression \( \text{factor 1} \ast \text{factor 2} \), all combinations of the members of factor 1 and factor 2 are introduced. For example, if factor 1 and factor 2 both have three members, then a total of nine variables are introduced into the model formula by this operator.
Trend Line Assumptions

The p-values reported in Tableau trend lines depend on some assumptions about the data. The first assumption is that, whenever a test is performed, the model for the mean is (at least approximately) correct.

The second assumption is that the "random errors" referred to in the model formula (see Trend Line Model Types on page 1669) are independent across different observations, and that they all have the same distribution. This constraint would be violated if the response variable had much more variability around the true trend line in one category than in another.

Assumptions Required to Compute Trend Lines

The Assumptions required to compute (using Ordinary Least Squares) each individual trend line are:

- Your model is an accurate functional simplification of the true data generating process (for example, no linear model for a log linear relationship).
- Your errors average to zero and are uncorrelated with your independent variable (for example, no error measuring the independent variable).
- Your errors have constant variance and are not correlated with each other (for example, no increase in error spread as your independent variable increases).
- Explanatory variables are not exact linear functions of each other (perfect multicollinearity).

Commonly Asked Questions About Trend Lines

This section describes some commonly asked questions regarding trend lines in Tableau.

How do I change the confidence level used in the model?

Tableau does not enforce a confidence level. It simply reports the significance of the whole model, or of a specific field, by showing the p-value. The p-value will measure the probability of obtaining the same trend result without taking the dimensions into account. For example, a trend of sales per time p-value of 0.05 means that there is 5% chance that the same value could be obtained without taking the time into consideration.
What does it mean if the p-value for the model is significant but the p-value for the specific field in the Analysis of Variance table is not significant?

The p-value in the Analysis of Variance table indicates whether the field adds or detracts from the significance of the entire model. The smaller the p-value the less likely it is that the difference in the unexplained variance between models with and without the field was a result of random chance. The values displayed for each field are derived by comparing the entire model to a model that does not include the field in question. So, for the situation where the p-value for the model is significant but the p-value for the specific field is not, you know that the model is statistically significant, but you cannot be confident that the specific field in question adds anything to it. Consider whether you might not be better off removing the factor from the model.

What does it mean if the p-value for the specific field in the Analysis of Variance table is significant but the p-value for the model is not significant?

This could happen in a case when there is no "trend" within each pane. For example, the lines are flat, but the mean varies across a given factor.

Drop Lines

Use drop lines to distinguish marks and call out their position in the view. For example, in a view that is dense with scatter marks, you can turn on drop lines to show the position of a particular data point. When you add drop lines, a line is extended from the marks to one of the axes. You can show drop lines all the time or only when a mark is selected.
Note: Drop lines do not display when you publish a view to Tableau Server or Tableau Online.

To add drop lines to the view, right-click (control-click on Mac) the pane and select \texttt{Drop Lines > Show Drop Lines}.

By default, drop lines are set to only show when the mark is selected. You can change this setting and specify other options in the Drop Lines dialog box.

Right-click (Control-click on Mac) the pane and select \texttt{Drop Lines > Edit Drop Lines} to open the Drop Lines dialog box.

In the Drop Lines dialog box select an axis to draw the line to, whether to always show the drop lines, and whether to show labels.
Find Clusters in Data

Cluster analysis partitions marks in the view into clusters, where the marks within each cluster are more similar to one another than they are to marks in other clusters.

Watch a Video: To see related concepts demonstrated in Tableau, watch Clustering, a 2-minute free training video. Use your tableau.com account to sign in.
For an example that demonstrates the process of creating clusters with sample data, see **Example: Create clusters using World Economic Indicators data** on page 1696.

In this article

- Create clusters
- Clustering constraints
- Edit clusters
- Create a group from cluster results
- Refit saved clusters
- How clustering works in Tableau
- Information on statistics models used for clusters
- Example: Create clusters using World Economic Indicators data

Create clusters

To find clusters in a view in Tableau, follow these steps.

1. Create a view.

2. Drag **Cluster** from the **Analytics** pane into the view, and drop it on in the target area in the view:
You can also double-click **Cluster** to find clusters in the view.

When you drop or double-click **Cluster**:

- Tableau creates a **Clusters** group on **Color**, and colors the marks in your view by cluster. If there is already a field on **Color**, Tableau moves that field to **Detail** and replaces it on **Color** with the clustering results.
  
  Tableau assigns each mark in the view to one of the clusters. In some cases, marks that do not fit well into a cluster are assigned to a "Not Clustered" cluster.

- Tableau displays the Clusters dialog box, where you can customize the cluster.

![Clusters dialog box](image)

3. Customize the cluster results by doing either of the following in the Clusters dialog box.

  - Drag new fields from the **Data** pane into the Variables area of the Clusters dialog box. You can also drag fields out of the Variables area to remove them.

    When you add variables, measures are aggregated using the default aggregation for the field; dimensions are aggregated using ATTR, which is the standard way that Tableau aggregates dimensions.

    To change the aggregation for a variable, right-click it.
Specify the number of clusters (between 2 and 50). If you do not specify a value, Tableau will automatically create up to 25 clusters.

4. When you finish customizing the cluster results, click the X in the upper-right corner of the Clusters dialog box to close it:

![X icon]

**Note:** You can move the cluster field from Color to another shelf in the view. However, you cannot move the cluster field from the Filters shelf to the Data pane.

To rename the resulting clusters, you must first save the cluster as a group. For details, see Create a group from cluster results on page 1686 and Edit clusters on the next page.

**Clustering constraints**

Clustering is available in Tableau Desktop, but is not available for authoring on the web (Tableau Server, Tableau Online). Clustering is also not available when any of the following conditions apply:

- When you are using a cube (multidimensional) data source.
- When there is a blended dimension in the view.
- When there are no fields that can be used as variables (inputs) for clustering in the view.
- When there are no dimensions present in an aggregated view.

When any of those conditions apply, you will not be able to drag Clusters from the Analytics pane to the view.

In addition, the following field types cannot be used as variables (inputs) for clustering:

- Table calculations
- Blended calculations
- Ad-hoc calculations
- Generated latitude/longitude values
Edit clusters

To edit an existing cluster, right-click (Control-click on a Mac) a Clusters field on Color and select Edit clusters.

To change the names used for each cluster, you will first need to drag the Clusters field to the Data pane and save it as a group. For details, see Create a group from cluster results on page 1686.
Right-click the cluster group and select **Edit Group** to make changes to each cluster.

Select a cluster group in the list of Groups and click **Rename** to change the name.
Create a group from cluster results

If you drag a cluster to the Data pane, it becomes a group dimension in which the individual members (Cluster 1, Cluster 2, etc.) contain the marks that the cluster algorithm has determined are more similar to each other than they are to other marks.

After you drag a cluster group to the Data pane, you can use it in other worksheets.

Drag Clusters from the Marks card to the Data pane to create a Tableau group:

After you create a group from clusters, the group and the original clusters are separate and distinct. Editing the clusters does not affect the group, and editing the group does not affect the cluster results. The group has the same characteristics as any other Tableau group. It is part of the data source. Unlike the original clusters, you can use the group in other worksheets in the
workbook. So if you rename the saved cluster group, that renaming is not applied to the original clustering in the view. See Correct Data Errors or Combine Dimension Members by Grouping Your Data on page 1002.

Constraints on saving clusters as groups

You will not be able to save Clusters to the Data pane under any of the following circumstances:

- When the measures in the view are disaggregated and the measures you are using as clustering variables are not the same as the measures in the view. For details, see How to Disaggregate Data on page 289.
- When the Clusters you want to save are on the Filters shelf.
- When Measure Names or Measure Values is in the view.
- When there is a blended dimension in the view.

Refit saved clusters

When you save a Clusters field as a group, it is saved with its analytic model. You can use your cluster groups in other worksheets and workbooks, however, they don't automatically refresh.

In this example, a saved cluster group and its analytic model has been applied to a different worksheet. As a result, some of the marks are not included in the clustering yet (indicated by gray marks).
If the underlying data changes, you can use the **Refit** option to refresh and recompute the data for a saved clusters group.

**To refit a saved cluster**

- Right-click a clusters group in the Data pane, and then click **Refit**.

Here's an example of updated clustering after refitting the saved cluster:
When you refit saved clusters, new clusters will be created and existing aliases for each cluster group category will be replaced with new, generic cluster aliases. Be aware that refitting saved clusters may alter your visualizations that use existing clusters and aliases.

**How clustering works**

Cluster analysis partitions the marks in the view into clusters, where the marks within each cluster are more similar to one another than they are to marks in other clusters. Tableau distinguishes clusters using color.

**Note:** For additional insight into how clustering works in Tableau, see the blog post [Understanding Clustering in Tableau 10](#).

**The clustering algorithm**

Tableau uses the k-means algorithm for clustering. For a given number of clusters $k$, the algorithm partitions the data into $k$ clusters. Each cluster has a center (centroid) that is the
mean value of all the points in that cluster. K-means locates centers through an iterative procedure that minimizes distances between individual points in a cluster and the cluster center. In Tableau, you can specify a desired number of clusters, or have Tableau test different values of $k$ and suggest an optimal number of clusters (see Criteria used to determine the optimal number of clusters on page 1692).

K-means requires an initial specification of cluster centers. Starting with one cluster, the method chooses a variable whose mean is used as a threshold for splitting the data in two. The centroids of these two parts are then used to initialize k-means to optimize the membership of the two clusters. Next, one of the two clusters is chosen for splitting and a variable within that cluster is chosen whose mean is used as a threshold for splitting that cluster in two. K-means is then used to partition the data into three clusters, initialized with the centroids of the two parts of the split cluster and the centroid of the remaining cluster. This process is repeated until a set number of clusters is reached.

Tableau uses Lloyd’s algorithm with squared Euclidean distances to compute the k-means clustering for each $k$. Combined with the splitting procedure to determine the initial centers for each $k > 1$, the resulting clustering is deterministic, with the result dependent only on the number of clusters.

The algorithm starts by picking initial cluster centers:
It then partitions the marks by assigning each to its nearest center:
Then it refines the results by computing new centers for each partition by averaging all the points assigned to the same cluster:

![Diagram showing cluster refinement process](image)

It then reviews the assignment of marks to clusters and reassigns any marks that are now closer to a different center than before.

The clusters are redefined and marks are reassigned iteratively until no more changes are occurring.

**Criteria used to determine the optimal number of clusters**

Tableau uses the Calinski-Harabasz criterion to assess cluster quality. The Calinski-Harabasz criterion is defined as

\[ \frac{SS_B}{SS_W} \times \frac{(N-k)}{(k-1)} \]

where \( SS_B \) is the overall between-cluster variance, \( SS_W \) the overall within-cluster variance, \( k \) the number of clusters, and \( N \) the number of observations.
The greater the value of this ratio, the more cohesive the clusters (low within-cluster variance) and the more distinct/separate the individual clusters (high between-cluster variance).

Since the Calinski-Harabasz index is not defined for $k=1$, it cannot be used to detect one-cluster cases.

If a user does not specify the number of clusters, Tableau picks the number of clusters corresponding to the first local maximum of the Calinski-Harabasz index. By default, k-means will be run for up to 25 clusters if the first local maximum of the index is not reached for a smaller value of $k$. You can set a maximum value of 50 clusters.

**Note:** If a categorical variable (that is, a dimension) has more than 25 unique values, then Tableau will disregard that variable when computing clusters.

What values get assigned to the "Not Clustered" category?

When there are null values for a measure, Tableau assigns values for rows with null to a Not Clustered category. Categorical variables (that is, dimensions) that return * for ATTR (meaning that all values are not identical) are also not clustered.

Scaling

Tableau scales values automatically so that columns having a larger range of magnitudes don’t dominate the results. For example, an analyst could be using inflation and GDP as input variables for clustering, but because GDP values are in trillions of dollars, this could cause the inflation values to be almost completely disregarded in the computation. Tableau uses a scaling method called *min-max normalization*, in which the values of each variable is mapped to a value between 0 and 1 by subtracting its minimum and dividing by its range.

Information on statistics models used for clusters

The Describe Clusters dialog box provides information about the models that Tableau computed for clustering. You can use these statistics to assess the quality of the clustering.

When the view includes clustering, you can open the Describe Clusters dialog box by right-clicking Clusters on the Marks card (Control-clicking on a Mac) and choosing Describe Clusters. The information in the Describe Clusters dialog box is read-only, though you can click Copy to Clipboard and then paste the screen contents into a writeable document.
Describe Clusters - Summary Tab

The Summary tab identifies the inputs that were used to generate the clusters and provides some statistics that characterize the clusters.

Inputs for Clustering

Variables

Identifies the fields Tableau uses to compute clusters. These are the fields listed in the Variables box in the Clusters dialog box.

Level of Detail

Identifies the fields that are contributing to the view’s level of detail—that is, the fields that determine the level of aggregation. For details, see How dimensions affect the level of detail in the view on page 255.

Scaling

Identifies the scaling method used for pre-processing. Normalized is currently the only scaling method Tableau uses. The formula for this method, also known as min-max normalization, is $(x - \text{min}(x)) / (\text{max}(x) - \text{min}(x))$.

Summary Diagnostics

Number of Clusters

The number of individual clusters in the clustering.

Number of Points

The number of marks in the view.

Between-group sum of squares

A metric quantifying the separation between clusters as a sum of squared distances between each cluster’s center (average value), weighted by the number of data points assigned to the cluster, and the center of the data set. The larger the value, the better the separation between clusters.

Within-group sum of squares

A metric quantifying the cohesion of clusters as a sum of squared distances between the center of each cluster and the individual marks in the cluster. The smaller the value, the more cohesive the clusters.
Total sum of squares

Totals the between-group sum of squares and the within-group sum of squares. The ratio (between-group sum of squares)/(total sum of squares) gives the proportion of variance explained by the model. Values are between 0 and 1; larger values typically indicate a better model. However, you can increase this ratio just by increasing the number of clusters, so it could be misleading if you compare a five-cluster model with a three-cluster model using just this value.

Cluster Statistics

For each cluster in the clustering, the following information is provided.

# Items
The number of marks within the cluster.

Centers
The average value within each cluster (shown for numeric items).

Most Common
The most common value within each cluster (only shown for categorical items).

Describe Clusters - Models Tab

Analysis of variance (ANOVA) is a collection of statistical models and associated procedures useful for analyzing variation within and between observations that have been partitioned into groups or clusters. In this case, analysis of variance is computed per variable, and the resulting analysis of variance table can be used to determine which variables are most effective for distinguishing clusters.

Relevant analysis of variance statistics for clustering include:

F-statistic
The F-statistic for one-way, or single-factor, ANOVA is the fraction of variance explained by a variable. It is the ratio of the between-group variance to the total variance.

The larger the F-statistic, the better the corresponding variable is distinguishing between clusters.
p-value

The p-value is the probability that the F-distribution of all possible values of the F-statistic takes on a value greater than the actual F-statistic for a variable. If the p-value falls below a specified significance level, then the null hypothesis (that the individual elements of the variable are random samples from a single population) can be rejected. The degrees of freedom for this F-distribution are \((k - 1, N - k)\), where \(k\) is the number of clusters and \(N\) is the number of items (rows) clustered.

The lower the p-value, the more the expected values of the elements of the corresponding variable differ among clusters.

Model Sum of Squares and Degrees of Freedom

The Model Sum of Squares is the ratio of the between-group sum of squares to the model degrees of freedom. The between group sum of squares is a measure of the variation between cluster means. If the cluster means are close to each other (and therefore close to the overall mean), this value will be small. The model has \(k-1\) degrees of freedom, where \(k\) is the number of clusters.

Error Sum of Squares and Degrees of Freedom

The Error Sum of Squares is the ratio of within-group sum of squares to the error degrees of freedom. The within-group sum-of-squares measures the variation between observations within each cluster. The error has \(N-k\) degrees of freedom, where \(N\) is the total number of observations (rows) clustered and \(k\) is the number of clusters.

The Error Sum of Squares can be thought of as the overall Mean Square Error, assuming that each cluster center represents the "truth" for each cluster.

Example: Create clusters using World Economic Indicators data

The Tableau clustering feature partitions marks in the view into clusters, where the marks within each cluster are more similar to one another than they are to marks in other clusters. This example shows how a researcher might use clustering to find an optimal set of marks (in this case, countries) in a data source.
The objective

As life expectancy increases around the world, and as older people remain more active, senior tourism can be a lucrative market for companies that know how to find and appeal to potential customers. The World Indicators sample data set that comes with Tableau contains the kind of data that might help companies identify the countries where there are enough of the right kind of customers.

Finding the right countries

Here is an example of how Tableau clustering could help such a company identify the countries where a senior tourism business could succeed. Imagine you are the analyst. Here is how you might proceed.

1. Open the World Indicators sample data source in Tableau Desktop.
2. Double-click Country in the Data pane.
   Tableau automatically creates a map view, with a mark in each country.
3. On the Marks card, change the mark type to Map:

   ![Map Mark Type]

   You should now see a map projection where all countries are filled with a solid color:
4. The next step is to identify the fields that you will use as variables for clustering. Here are the fields you choose:

<table>
<thead>
<tr>
<th>Field</th>
<th>Reason for inclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Life Expectancy Female and Life Expectancy Male</td>
<td>Where people are living longer, there are more likely to be people who are interested in traveling later in life.</td>
</tr>
<tr>
<td>Population Urban</td>
<td>It is easier to market services in areas with greater population density.</td>
</tr>
<tr>
<td>Population 65+</td>
<td>The target population is older residents with the time and funds to travel.</td>
</tr>
<tr>
<td>TourismPerCapita</td>
<td>This is a measure that you must create as a named calculated field. The formula is:</td>
</tr>
<tr>
<td></td>
<td>SUM([Tourism Outbound])/SUM([Population Total])</td>
</tr>
</tbody>
</table>
Tourism Outbound aggregates the money (in US dollars) that residents of a country spend annually on international travel. But this total must be divided by the population of each country to determine the average amount each resident spends on international travel.

There is no guarantee that these are the ideal fields to choose, or that these fields will produce cluster results that are clear and unambiguous. Clustering is an iterative process—experimentation leads to discovery which leads, in turn, to more experimentation.

5. Drag these five fields from the Data pane to Detail on the Marks card.

6. Click to open the Analytics pane:

7. Drag Cluster from the Analytics pane and drop it in the view:
Tableau displays the Clusters dialog box and adds the measures in the view to the list of variables:

It also updates the view by adding clusters to Color. In this case, Tableau finds two distinct clusters, and is unable to assign certain countries (colored reddish-pink) to either cluster:
Note: See How clustering works on page 1689 for details on data that Tableau assigns to "Not clustered."

8. You decide that two clusters isn't enough—you don't have the resources to set up shop in half the countries in the world. So you type 4 in the Number of Clusters field in the Clusters dialog box.

The map becomes more interesting:
But how do these clusters relate to the variables you have chosen? Which one correlates best with the factors that support senior tourism? It’s time to look at the statistics behind the clusters.

9. Close the Clusters dialog box by clicking the X in its upper-right corner:

![X button]

10. Click the Clusters field on the Marks card and choose Describe Clusters.

The table at the bottom of the Models tab in the Describe Clusters dialog box shows the average value for each variable in each cluster:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1</td>
<td>69</td>
<td>74.226</td>
<td>54.429</td>
<td>0.003063</td>
<td>0.5205</td>
<td>187.12</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>52</td>
<td>56.75</td>
<td>54.429</td>
<td>0.003063</td>
<td>0.33127</td>
<td>20.775</td>
</tr>
<tr>
<td>Cluster 3</td>
<td>29</td>
<td>79.164</td>
<td>71.706</td>
<td>0.15493</td>
<td>0.05532</td>
<td>320.92</td>
</tr>
<tr>
<td>Cluster 4</td>
<td>20</td>
<td>82.287</td>
<td>76.355</td>
<td>0.12006</td>
<td>0.87123</td>
<td>1300.4</td>
</tr>
<tr>
<td>Not Clustered</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cluster 4 has the highest life expectancy (both male and female), the highest concentration of urban population, and the highest expenditure for international tourism: $1360.40 per capita. The only variable for which Cluster 4 does not have the highest
value is **Population 65+**, where Cluster 3 has the advantage: 0.15493 (just under 16%) to 0.11606 (just over 11%) in Cluster 4.

The clustering algorithm does not know whether you are looking for the maximum value for these variables, the minimum value, or something in the middle—it just looks for correlation. But you know that higher values for these variables is the signal you’re looking for, and Cluster 4 is the best choice.

11. You could attempt to pick out the Cluster 4 countries from the map, but there is an easier way. Close the **Describe Clusters** dialog box and then click **Cluster 4** on the Color legend and choose **Keep Only**.

12. Choose Text Table from ShowMe.

You now see a list of the countries in Cluster 4:
This list is not the end of the process. You might try clustering again with a somewhat different set of variables and maybe a different number of clusters, or you might add some countries to the list and remove others, based on other factors. For example, if your tours are mostly to tropical locales, you might remove countries like Curacao and the Bahamas from the list, because tropical tours might not appeal to residents of those countries.
Another option is to filter your data before you re-cluster, to only show countries with populations above a certain threshold, or to target countries in a particular geographical area.

Forecasting

You can add a forecast to a view when there is at least one date dimension and one measure in the view.

To turn forecasting on, right-click (control-click on Mac) on the visualization and choose Forecast >Show Forecast, or choose Analysis >Forecast >Show Forecast.

When no date dimension is present, you can add a forecast if there is a dimension field in the view that has integer values. For details, see Forecasting When No Date is in the View on page 1717.

You can forecast quantitative time-series data using exponential smoothing models in Tableau Desktop. With exponential smoothing, recent observations are given relatively more weight than older observations. These models capture the evolving trend or seasonality of your data and extrapolate them into the future. Forecasting is fully automatic, yet configurable. Many forecast results can become fields in your visualizations.

When a forecast is showing, future values for the measure are shown next to the actual values.

Watch a video: To see related concepts demonstrated in Tableau, watch Forecasting, a 6-minute free training video. Use your tableau.com account to sign in.

Forecasting Constraints

Forecasting is not supported for Multidimensional data sources. In Tableau Desktop, multidimensional data sources are supported only in Windows.

You can publish a view that contains a forecast, and see the forecast when you view or edit the view on the web, but you cannot modify or add a forecast when you are editing a view on the web.

In addition, you cannot add a forecast to a view if it contains any of the following:

- Table calculations
- Disaggregated measures
Percent calculations
Grand Totals or Subtotals
Date values with aggregation set to Exact Date

How Forecasting Works in Tableau

Forecasting in Tableau uses a technique known as exponential smoothing. Forecast algorithms try to find a regular pattern in measures that can be continued into the future.

Watch a video: To see related concepts demonstrated in Tableau, watch Forecasting, a 6-minute free training video. Use your tableau.com account to sign in.

You typically add a forecast to a view that contains a date field and at least one measure. However, in the absence of a date, Tableau can create a forecast for a view that contains a dimension with integer values in addition to at least one measure.

For details on creating a forecast, see Create a Forecast on page 1713. For details on forecasting using an integer dimension, see Forecasting When No Date is in the View on page 1717.

In this article

Overview below
Exponential Smoothing and Trend on the next page
Seasonality on page 1708
Model Types on page 1709
Forecasting with Time on page 1710
Granularity and Trimming on page 1711
Getting More Data on page 1712

Overview

All forecast algorithms are simple models of a real-world data generating process (DGP). For a high quality forecast, a simple pattern in the DGP must match the pattern described by the model reasonably well. Quality metrics measure how well the model matches the DGP. If the quality is low, the precision measured by the confidence bands is not important because it measures the precision of an inaccurate estimate.
Tableau automatically selects the best of up to eight models, the best being the one that generates the highest quality forecast. The smoothing parameters of each model are optimized before Tableau assesses forecast quality. The optimization method is global. Therefore, choosing locally optimal smoothing parameters that are not also globally optimal is not impossible. However, initial value parameters are selected according to best practices but are not further optimized. So it is possible for initial value parameters to be less than optimal. The eight models available in Tableau are among those described at the following location on the OTexts web site: A taxonomy of exponential smoothing methods.

When there is not enough data in the visualization, Tableau automatically tries to forecast at a finer temporal granularity, and then aggregates the forecast back to the granularity of the visualization. Tableau provides prediction bands which may be simulated or calculated from a closed form equation. All models with a multiplicative component or with aggregated forecasts have simulated bands, while all other models use the closed form equations.

Exponential Smoothing and Trend

*Exponential smoothing* models iteratively forecast future values of a regular time series of values from weighted averages of past values of the series. The simplest model, *Simple Exponential Smoothing*, computes the next level or smoothed value from a weighted average of the last actual value and the last level value. The method is exponential because the value of each level is influenced by every preceding actual value to an exponentially decreasing degree—more recent values are given greater weight.

Exponential smoothing models with trend or seasonal components are effective when the measure to be forecast exhibits trend or seasonality over the period of time on which the forecast is based. *Trend* is a tendency in the data to increase or decrease over time. *Seasonality* is a repeating, predictable variation in value, such as an annual fluctuation in temperature relative to the season.

In general, the more data points you have in your time series, the better the resulting forecast will be. Having enough data is particularly important if you want to model seasonality, because the model is more complicated and requires more proof in the form of data to achieve a reasonable level of precision. On the other hand, if you forecast using data generated by two or more different DGPs, you will get a lower quality forecast because a model can only match one.
Seasonality

Tableau tests for a seasonal cycle with the length most typical for the time aggregation of the time series for which the forecast is estimated. So if you aggregate by months, Tableau will look for a 12-month cycle; if you aggregate by quarters, Tableau will search for a four-quarter cycle; and if you aggregate by days, Tableau will search for weekly seasonality. Therefore, if there is a six-month cycle in your monthly time series, Tableau will probably find a 12-month pattern that contains two similar sub-patterns. However, if there is a seven-month cycle in your monthly time series, Tableau will probably find no cycle at all. Luckily, seven-month cycles are uncommon.

Tableau can use either of two methods for deriving season length. The original temporal method uses the natural season length of the temporal granularity (TG) of the view. Temporal granularity means the finest unit of time expressed by the view. For example, if the view contains either a continuous green date truncated to month or discrete blue year and month date parts, the temporal granularity of the view is month. The new non-temporal method, introduced with Tableau 9.3, uses periodic regression to check season lengths from 2 to 60 for candidate lengths.

Tableau automatically selects the most appropriate method for a given view. When Tableau is using a date to order the measures in a view, if the temporal granularity is quarterly, monthly, weekly, daily or hourly, the season lengths are almost certainly 4, 12, 13, 7 or 24, respectively. So only the length natural to the TG is used to construct the five seasonal exponential smoothing models supported by Tableau. The AIC of the five seasonal models and the three non-seasonal models are compared and the lowest returned. (For an explanation of the AIC metric, see Forecast Descriptions.)

When Tableau is using an integer dimension for forecasting, the second method is used. In this case there is no temporal granularity (TG), so potential season lengths must be derived from the data.

The second method is also used if the temporal granularity is yearly. Yearly series rarely have seasonality, but, if they do, it must also be derived from the data.

The second method is also used for views with temporal granularity of minute or second. If such series have seasonality, the season lengths are likely 60. However, when measuring a regular real world process, the process may have a regular repetition which does not correspond to the clock. So, for minutes and seconds, Tableau also checks for a length different from 60 in the data. This does not mean that Tableau can model two different season lengths at the same time. Rather, ten seasonal models are estimated, five with a season length of 60 and another
five with the season length derived from the data. Whichever of the ten seasonal models or three non-seasonal models has the lowest AIC, that model is used to compute the forecast.

For series ordered by year, minute, or second, a single season length from the data is tested if the pattern is fairly clear. For integer ordered series, up to nine somewhat less clear potential season lengths are estimated for all five seasonal models, and the model with the lowest AIC is returned. If there are no likely season length candidates, only the non-seasonal models are estimated.

Since all selection is automatic when Tableau is deriving potential season lengths from the data, the default Model Type of “Automatic” in the Forecast Options Dialog Model Type menu does not change. Selecting “Automatic without seasonality” improves performance by eliminating all season length searching and estimation of seasonal models.

The heuristic that Tableau uses to decide when to use season lengths derived from the data depends on the distribution of errors for the periodic regression of each candidate season length. Since the assembly of season length candidates by periodic regression usually produces one or two clear winning lengths if seasonality actually exists in the data, the return of a single candidate indicates likely seasonality. In this case, Tableau estimates seasonal models with this candidate for year, minute and second granularity. The return of less than the maximum of ten candidates indicates possible seasonality. In this case, Tableau estimates seasonal models with all returned candidates for integer ordered views. The return of the maximum number of candidates indicates that errors for most length are similar. Therefore, the existence of any seasonality is unlikely. In this case, Tableau estimates only non-seasonal models for an integer-ordered or yearly ordered series, and only the seasonal models with a natural season length for other temporally ordered views.

For Model Type “Automatic” in integer-, year-, minute- and second-ordered views, candidate season lengths are always derived from the data whether or not they are used. Since model estimation is much more time consuming than periodic regression, the performance impact should be moderate.

Model Types

In the Forecast Options dialog box, you can choose the model type Tableau users for forecasting. The **Automatic** setting is typically optimal for most views. If you choose **Custom**, then you can specify the trend and season characteristics independently, choosing either **None, Additive, or Multiplicative**: 

Back to top
An additive model is one in which the contributions of the model components are summed, whereas a multiplicative model is one in which at least some component contributions are multiplied. Multiplicative models can significantly improve forecast quality for data where the trend or seasonality is affected by the level (magnitude) of the data:

Keep in mind that you do not need to create a custom model to generate a forecast that is multiplicative: the Automatic setting can determine if a multiplicative forecast is appropriate for your data. However, a multiplicative model cannot be computed when the measure to be forecast has one or more values that are less than or equal to zero.

Forecasting with Time

When you are forecasting with a date, there can be only one base date in the view. Part dates are supported, but all parts must refer to the same underlying field. Dates can be on Rows, Columns, or Marks (with the exception of the Tooltip target).

Tableau supports three types of dates, two of which can be used for forecasting:

- Truncated dates and reference a particular point in history with specific temporal granularity, such as February 2017. They are usually continuous, with a green
background in the view. Truncated dates are valid for forecasting.

- Date parts refer to a particular member of a temporal measure such as February. Each date part is represented by a different, usually discrete field (with a blue background). Forecasting requires at least a Year date part. Specifically, it can use any of the following sets of date parts for forecasting:
  - Year
  - Year + quarter
  - Year + month
  - Year + quarter + month
  - Year + week
  - Custom: Month/Year, Month/Day/Year

Other date parts, such as Quarter or Quarter + month, are not valid for forecasting. See Convert Fields between Discrete and Continuous on page 980 for more details about different date types.

- Exact dates refer to a particular point in history with maximum temporal granularity such as February 1, 2012 at 14:23:45.0. Exact dates are invalid for forecasting.

It is also possible to forecast without a date. See Forecasting When No Date is in the View on page 1717.

Granularity and Trimming

When you create a forecast, you select a date dimension that specifies a unit of time at which date values are to be measured. Tableau dates support a range of such time units, including Year, Quarter, Month, and Day. The unit you choose for the date value is known as the granularity of the date.

The data in your measure typically does not align precisely with your unit of granularity. You might set your date value to quarters, but your actual data may terminate in the middle of a quarter—for example, at the end of November. This can cause a problem because the value for this fractional quarter is treated by the forecasting model as a full quarter, which will typically have a lower value than a full quarter would. If the forecasting model is allowed to consider this data, the resulting forecast will be inaccurate. The solution is to trim the data, such that the trailing periods that could mislead the forecast are ignored. Use the Ignore Last option in the
Forecast Options dialog box to remove—or trim—such partial periods. The default is to trim one period.

---

**Getting More Data**

Tableau requires at least five data points in the time series to estimate a trend, and enough data points for at least two seasons or one season plus five periods to estimate seasonality. For example, at least nine data points are required to estimate a model with a four quarter seasonal cycle (4 + 5), and at least 24 to estimate a model with a twelve month seasonal cycle (2 * 12).

If you turn on forecasting for a view that does not have enough data points to support a good forecast, Tableau can sometimes retrieve enough data points to produce a valid forecast by querying the datasource for a finer level of granularity:

- If your view contains fewer than nine years of data, by default, Tableau will query the data source for quarterly data, estimate a quarterly forecast, and aggregate to a yearly forecast to display in your view. If there are still not enough data points, Tableau will estimate a monthly forecast and return the aggregated yearly forecast to your view.

- If your view contains fewer than nine quarters of data, by default Tableau will estimate a monthly forecast and return the aggregated quarterly forecast results to your view.

- If your view contains fewer than nine weeks of data, by default, Tableau will estimate a daily forecast and return the aggregated weekly forecast results to your view.

- If your view contains fewer than nine days of data, by default, Tableau will estimate an hourly forecast and return the aggregated daily forecast results to your view.

- If your view contains fewer than nine hours of data, by default, Tableau will estimate an minutely forecast and return the aggregated hourly forecast results to your view.

- If your view contains fewer than nine minutes of data, by default, Tableau will estimate an secondly forecast and return the aggregated minutely forecast results to your view.

These adjustments happen behind the scene and require no configuration. Tableau does not change the appearance of your visualization, and does not actually change your date value. However, the summary of the forecast time period in the Forecast Describe and Forecast Options dialog will reflect the actual granularity used.

Tableau can only get more data when the aggregation for the measure you are forecasting is SUM or COUNT. See *Data Aggregation in Tableau* on page 279 for information on available aggregation types and information on how to change the aggregation type.
Create a Forecast

To create a forecast, your view must be using at least one date dimension and one measure.

To turn forecasting on, right-click (control-click on Mac) on the visualization and choose **Forecast >Show Forecast**, or choose **Analysis >Forecast >Show Forecast**.

**Watch a video:** To see related concepts demonstrated in Tableau, watch **Forecasting**, a 6-minute free training video. Use your **tableau.com** account to sign in.

Each of the following examples indicate the structure that supports creating a forecast.

- The field you want to forecast is on the **Rows** shelf and a continuous date field is on the **Columns** shelf.

- The field you want to forecast is on the **Columns** shelf and a continuous date field is on the **Rows** shelf.

- The field you want to forecast on either the **Rows** or **Columns** shelf, and discrete dates are on either the **Rows** or **Columns** shelf. At least one of the included date levels must be Year.

- The field you want to forecast is on the Marks card, and a continuous date or discrete date set is on **Rows, Columns or Marks**.

**Note:** You can also create a forecast when no date dimension is present if there is a dimension in the view that has integer values. See **Forecasting When No Date is in the View** on page 1717.

With forecasting on, Tableau visualizes estimated future values of the measure, in addition to actual historical values. The estimated values are shown by default in a lighter shade of the color used for the historical data:
The shaded area in the image above shows the 95% prediction interval for the forecast. That is, the model has determined that there is a 95% likelihood that the value of sales will be within the shaded area for the forecast period. You can configure the confidence level percentile for the prediction bands, and whether prediction bands are included in the forecast, using the **Show prediction intervals** setting in the Forecast Options dialog box:

Clear the check box if you do not want to display prediction bands in forecasts. To set the prediction interval, select one of the values or enter a custom value. The lower the percentile you set for the confidence level, the narrower the prediction bands will be.

How your prediction intervals are displayed depends on the mark type of your forecasted marks:

<table>
<thead>
<tr>
<th>Forecast mark type</th>
<th>Prediction intervals displayed using</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>Bands</td>
</tr>
</tbody>
</table>
Shape, square, circle, bar, or pie

Whiskers

In the following example, forecast data is indicated by lighter shaded circles, and the prediction intervals are indicated by lines ending in whiskers:

Enhancing Forecasts

For each forecast value, consider verifying the quality or precision of your forecast by dragging another instance of the forecast measure from the Data pane to the Detail shelf on the Marks card and then after right-clicking the field to open the content menu, choosing one of the available options:
For descriptions of these options, see **Forecast Field Results** on the next page.

You can repeat the process to add additional result types for each forecast value. See "Changing the Forecast Result" type in **Forecast Field Results** on the next page for information on changing the result type.

By adding such result types to the Details shelf, you add information about the forecast to tooltips for all marks that are based on forecasted data.

<table>
<thead>
<tr>
<th>Forecast Indicator:</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Month of Ship Date:</td>
<td>April 2015</td>
</tr>
<tr>
<td>Precision % of Sales:</td>
<td>±38.00%</td>
</tr>
<tr>
<td>Quality of Sales:</td>
<td>64</td>
</tr>
<tr>
<td>Sales:</td>
<td>$54,753</td>
</tr>
</tbody>
</table>
Forecasting When No Date is in the View

If a valid date is not in the view, Tableau will look for a dimension in the view that has integer values. If it finds such a dimension it will use that to forecast additional values for measures in the view. As with a date, when an integer dimension is selected to order the measures to be forecast, it is no longer used to partition the data. If there is more than one such integer dimension, Tableau will go in this order:

- An integer dimension on the Columns shelf. If there is more than one such dimension, it will use the first one (farthest to the left on the shelf).
- An integer dimension on the Rows shelf.
- An integer dimension on the Pages shelf.
- An integer dimension on the Marks card.

When Tableau is using an integer dimension to forecast, the Forecast Option and Forecast Description dialog boxes will automatically specify that forecasting is aggregating by periods:

Forecast Field Results

Tableau provides several types of forecast results. To view these result types in the view, right-click (control-click on Mac) on the measure field, choose **Forecast Result**, and then choose
one of the options.

The options are:

- **Actual & Forecast**—Show the actual data extended by forecasted data.
- **Trend**—Show the forecast value with the seasonal component removed.
- **Precision**—Show the prediction interval distance from the forecast value for the configured confidence level.
- **Precision %**—Show precision as a percentage of the forecast value.
- **Quality**—Show the quality of the forecast, on a scale of 0 (worst) to 100 (best). This metric is scaled MASE, based on the MASE (Mean Absolute Scaled Error) of the forecast, which is the ratio of forecast error to the errors of a naïve forecast which assumes that the value of the current period will be the same as the value of the next period. The actual equation used for quality is:

\[
100 \times \max(1 - MASE, 0)
\]

The Quality for a naïve forecast would be 0. The advantage of the MASE metric over the more common MAPE is that MASE is defined for time series which contain zero, whereas MAPE is not. In addition, MASE weights errors equally while MAPE weights positive and/or extreme errors more heavily.

- **Upper Prediction Interval**—Shows the value above which the true future value will lie confidence level percent of the time assuming a high quality model. The confidence level percentage is controlled by the Prediction Interval setting in the Forecast Options dialog box. See **Configure Forecast Options** on page 1720.

- **Lower Prediction Interval**—Shows 90, 95, or 99 confidence level below the forecast value. The actual interval is controlled by the **Prediction Interval** setting in the Forecast Options dialog box.

- **Indicator**—Show the string **Actual** for rows that were already on the worksheet when forecasting was inactive and **Estimate** for rows that were added when forecasting was activated.

- **None**—Do not show forecast data for this measure.
Forecast description information is also included in the worksheet description. See Describing the View on page 1753.

Forecasting a New Measure

When you add a new measure to a visualization that already has forecasting enabled, Tableau attempts to forecast future values.

Changing the Forecast Result Type

To change the forecast result type for a measure, right-click (control-click on Mac) on the measure field, select Forecast Result, and then choose a result type.
Configure Forecast Options

Use the Forecast Options dialog box to configure forecast options, including:

- The length of the forecast
- The range and temporal aggregation of source data from which to generate the forecast
- The forecast model
- Prediction intervals

When forecasting is enabled, you can open the Forecast Options dialog box by choosing **Analysis > Forecast > Forecast Options**.

Forecast Length

The **Forecast Length** section determines how far into the future the forecast extends. Select one of the following:

- **Automatic**: Tableau determines the forecast length based on the data.
- **Exactly**: Extends the forecast for the specified number of units.
- **Until**: Extends the forecast to the specified point in the future.

Source Data

Use the **Source Data** section to specify.

- **Aggregate by**: Specifies the temporal granularity of the time series. With the default value (**Automatic**), Tableau chooses the best granularity for estimation. This will typically match the temporal granularity of the visualization (that is, the date dimension that the forecast is based on). However, it is sometimes possible and desirable to estimate the forecast model at a finer granularity than the visualization when the time series in the visualization is too short to allow estimation.

**Note**: When you are using an integer dimension instead of a date dimension for forecasting, the Aggregate by value is always Periods. See **Forecasting When No Date is in the View** on page 1717.

- **Ignore last**: Specifies the number of periods at the end of the actual data that should be ignored in estimating the forecast model. Forecast data is used instead of actual data for these time periods. Use this feature to trim off unreliable or partial trailing periods which
could mislead the forecast. When the estimation granularity specified in the Source Data section is finer than in the visualization, the trimmed periods are estimation periods. As a result, the trailing actual visualization period may become a forecast period, which is an aggregate of both actual and forecast periods of estimation granularity. In contrast, null values are not filled with zeros and must be filtered to allow forecast.

- **Fill in missing values with zeros**: If there are missing values in the measure you are attempting to forecast, you can specify that Tableau fill in these missing values with zero.

**Forecast Model**

The **Forecast Model** section specifies how the forecast model is to be produced.

Use the drop down to specify whether Tableau selects what it determines to be the best of all models (**Automatic**), the best of those with no seasonal component (**Automatic without seasonality**), or the model you specify (**Custom**).

When you choose the **Custom** option, two new fields appear in the Forecast Options dialog box, which you use to specify the trend and season characteristics for your model:

![Forecast Options dialog box](image)

The choices are the same for both fields:

- **None**: When you select **None** for Trend, the model does not assess the data for trend. When you select **None** for Season, the model does not assess the data for seasonality.

- **Additive**: An additive model is one in which the combined effect of several independent factors is the sum of the isolated effects of each factor. You can assess the data in your view for additive trend, additive seasonality, or both.

- **Multiplicative**: A multiplicative model is one in which the combined effect of several independent factors is the product of the isolated effects of each factor. You can assess the data in your view for multiplicative trend, multiplicative seasonality, or both.
If there is more than one time series in your visualization, the Custom option forces them all to be forecast using the same custom model. Constraining the models in this way usually results in lower quality models than would be produced by automatic model selection.

**Constraints on Multiplicative Models**

- You cannot use a multiplicative model when the measure to be forecast has one or more values that are less than or equal to zero, or even when some of the data points are too close to zero, relative to other data points.
- You cannot specify a model with multiplicative trend and additive season because the result may be numerically unstable.

**Prediction Interval**

You can set the prediction interval to 90, 95, or 99 percent, or enter a custom value. This value is used in two locations:

- In the prediction bands displayed with a forecast.
- For the prediction interval options *(Upper Prediction Interval* and *Lower Prediction Interval*)
Interval) that are available as forecast result types for a measure in the view:

```
Forecast Summary

The text box at the bottom of the Forecast Options dialog box provides a description of the current forecast. The forecast summary updates whenever you change any of the forecast options above. If there is a problem with the forecast, the text box provides an error message that may help you resolve the issue. See Resolving Forecasting Errors on page 1729.

Forecast Descriptions

The Describe Forecast dialog box describes the forecast models that Tableau computed for your visualization.

When forecasting is enabled, you can open this dialog by selecting Analysis > Forecast > Describe Forecast.
```
The information in the Describe Forecast dialog box is read-only, though you can click Copy to Clipboard and then paste the screen contents into a document.

The Describe Forecast dialog box has two tabs: a Summary tab and a Models tab.

Describe Forecast - Summary Tab

The Summary tab describes the forecast models Tableau has created, as well as the general patterns Tableau discovered in the data.

Options Used To Create Forecasts

This section summarizes the options Tableau used to create forecasts. These options were either picked automatically by Tableau or specified in the Forecast Options dialog box.

- **Time series**—The continuous date field used to define the time series. In some cases this value might not actually be a date. See Forecasting When No Date is in the View on page 1717.

- **Measures**—The measures for which values are estimated.

- **Forecast forward**—The length and date range of the forecast.

- **Forecast based on**—The date range of the actual data used to create the forecast.

- **Ignore last**—The number of periods at the end of the actual data that are disregarded—forecast data is displayed for these periods. This value is determined by the Ignore Last option in the Forecast Options dialog box.

- **Seasonal pattern**—The length of the seasonal cycle that Tableau found in the data, or None if no seasonal cycle was found in any forecast.

Forecast Summary Tables

For each measure that is forecasted, a summary table is displayed describing the forecast. If the view is broken into multiple panes using dimensions, a column is inserted into each table that identifies the dimensions. The fields in summary forecast tables are:

- **Initial**—The value and prediction interval of the first forecast period.

- **Change From Initial**—The difference between the first and the last forecast estimate points. The interval for those two points is shown in the column header. When values are shown as percentages, this field shows the percentage change from the first forecast
Period.

- **Seasonal Effect**—These fields are displayed for models identified as having seasonality—that is, a repeating pattern of variation over time. They show the high and low value of the seasonal component of the last full seasonal cycle in the combined time series of actual and forecast values. The seasonal component expresses the deviation from the trend and so varies around zero and sums to zero over the course of a season.

- **Contribution**—The extent to which Trend and Seasonality contribute to the forecast. These values are always expressed as percentages and add up to 100%.

- **Quality**—Indicates how well the forecast fits the actual data. Possible values are GOOD, OK, and POOR. A naïve forecast is defined as a forecast that estimates that the value of the next period will be identical to the value of the current period. Quality is expressed relative to a naïve forecast, such that OK means the forecast is likely to have less error than a naïve forecast, GOOD means that the forecast has less than half as much error, and POOR means that the forecast has more error.

Describe Forecast - Models Tab

The **Models** tab provides more exhaustive statistics and smoothing coefficient values for the Holt-Winters exponential smoothing models underlying the forecasts. For each measure that is forecasted, a table is displayed describing the forecast models Tableau created for the measure. If the view is broken into multiple panes using dimensions, a column is inserted into each table that identifies the dimensions. The table is organized into the following sections:

**Model**

Specifies whether the components **Level**, **Trend**, or **Season** are part of the model used to generate the forecast. The value for each component is one of the following:

- **None**—The component is not present in the model.
- **Additive**—The component is present and is added to the other components to create the overall forecast value.
- **Multiplicative**—The component is present and is multiplied by the other components to create the overall forecast value.

**Quality Metrics**

This set of values provides statistical information about the quality of the model.
<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE: Root mean squared error</td>
<td>$\sqrt{\left(\frac{1}{n}\right) \sum e(t)^2}$</td>
</tr>
<tr>
<td>MAE: Mean absolute error</td>
<td>$\frac{1}{n} \sum</td>
</tr>
<tr>
<td>MASE: Mean absolute scaled error.</td>
<td>$\frac{\frac{1}{n} \sum</td>
</tr>
</tbody>
</table>

**RMSE**

Root mean squared error (RMSE) is defined as the square root of the average squared error. It provides a measure of the standard deviation of the errors, which gives us a sense of the typical magnitude of the error. It is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum e(t)^2}$$

**MAE**

Mean absolute error (MAE) is the average of the absolute differences between the observed and predicted values. It is defined as:

$$\text{MAE} = \frac{1}{n} \sum |e(t)|$$

**MASE**

Mean absolute scaled error (MASE) measures the magnitude of the error compared to the magnitude of the error of a naive one-step ahead forecast as a ratio. A naive forecast assumes that whatever the value is today will be the same value tomorrow. So, a MASE of 0.5 means that your forecast is likely to have half as much error as a naive forecast, which is better than a MASE of 1.0, which is no better than a naive forecast. Since this is a normalized statistic that is defined for all values and weighs errors evenly, it is an excellent metric for comparing the quality of different forecast methods.

The advantage of MASE over the more common MAPE metric is that MASE is defined for time series that contain zero, whereas MAPE is not. Also, MASE weights errors equally, whereas MAPE weights positive and/or extreme errors more heavily.

**MAPE**

Mean absolute percentage error (MAPE) measures the magnitude of the error compared to the magnitude of your data, as a percentage. So, a MAPE of 20% is better than a MAPE of 60%. Errors are the differences between the response values, which the model estimates, and the actual response values for each explanatory value in your data. Since this is a normalized statistic, it can be used to compare the quality of different models computed in Tableau. However, it is unreliable for
some comparisons because it weights some kinds of error more heavily than others. Also, it is undefined for data with values of zero.

AIC: Akaike information criterion.
AIC is a model quality measure, developed by Hirotugu Akaike, that penalizes complex models to prevent overfitting. In this definition, $k$ is the number of estimated parameters, including initial states, and SSE is the sum of the squared errors.

In the preceding definitions, the variables are as follow:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>Index of a period in a time series.</td>
</tr>
<tr>
<td>$n$</td>
<td>Time series length.</td>
</tr>
<tr>
<td>$m$</td>
<td>Number of periods in a season/cycle.</td>
</tr>
<tr>
<td>$A(t)$</td>
<td>Actual value of the time series at period $t$.</td>
</tr>
<tr>
<td>$F(t)$</td>
<td>Fitted or forecast value at period $t$.</td>
</tr>
</tbody>
</table>

Residuals are: $e(t) = F(t) - A(t)$

**Smoothing Coefficients**

Depending on the rate of evolution in the level, trend, or seasonal components of the data, smoothing coefficients are optimized to weight more recent data values over older ones, such that within-sample one-step-ahead forecast errors are minimized. Alpha is the level smoothing coefficient, beta the trend smoothing coefficient, and gamma the seasonal smoothing coefficient. The closer a smoothing coefficient is to 1.00, the less smoothing is performed, allowing for rapid component changes and heavy reliance on recent data. The closer a smoothing coefficient is to 0.00, the more smoothing is performed, allowing for gradual component changes and less reliance on recent data.
Troubleshooting Forecasting

This topic discusses some issues that may arise as you use forecasting in Tableau.

Null Forecasts

A null forecast results when you are using forecasting and modify the view in a manner that is incompatible with forecasting. The most obvious indication that you have a null forecast is that no forecast data is displayed in your view and the text No Forecast is displayed. Hover the cursor over this message to see why Tableau is unable to create a forecast:

Another indication is that the Forecast Indicator field on the Marks card shows Actual values, but no Estimate values:

To diagnose a null forecast, open the Describe Forecast dialog box from the Analysis menu (Analysis > Forecast > Describe Forecast) to see the error message. Then see Resolving Forecasting Errors on the next page for a suggested resolution.

Tableau also displays the error message in the Forecast Options dialog box (Analysis > Forecast > Forecast Options).

Because a forecast cannot be computed for a time series with null date values in the middle, actions that filter data, either explicitly or implicitly can trigger a null forecast. The Keep Only and Exclude commands on tooltips are examples of actions that can filter data implicitly—in some cases, these commands are removed when a forecast is shown. For example, if you have a time series of sales for each quarter from 2008 until 2012 and you exclude the Sales value for the second quarter of 2010, you will get a Null forecast because the time series is
irregular. If, instead, you exclude the first quarter of 2008, you have shortened the time series but it remains regular. So, a valid forecast is still possible.

Resolving Forecasting Errors

If Tableau is unable to provide a forecast for your view, the problem can often be resolved by changing the Date value in the view (see Changing Date Levels on page 1061).

Forecasting errors can result when the aggregation level of the time series (months, weeks, etc.) is either too fine or too coarse for the data to be forecast. This can lead to the "too much data" or "too little data" errors described below. Date aggregation can trigger a "too many Nulls" scenario when forecasting attempts to extract more data from the measure than is possible. For example, if the underlying granularity of the sales data is months but you aggregate by weeks, the result may be a significant number of Null values.

Other problems arise when the view’s aggregation and the aggregation specified for the forecast (using the Aggregate by field in the Forecast Options dialog box) are not compatible. Tableau can create a forecast when the forecast aggregation is a finer level of detail than the view’s aggregation, but not when it is at a coarser level of detail; even when it is finer, the two values are only compatible if there is a strict hierarchy that Tableau can use (for example, quarters can be evenly divided into three months, but months can’t be evenly divided into weeks). Avoid these scenarios by setting Aggregate by to Automatic.

The following list shows errors that can be result from invalid forecasts in Tableau, and provides advice on how to resolve them.

<table>
<thead>
<tr>
<th>Error message</th>
<th>Suggestion for Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>A continuous date cannot be derived from the date fields in the view.</td>
<td>Forecasting requires a date field that can be interpreted continuously. If the date field is not explicitly continuous, then one of the included date levels must be Year. This error is returned if there are no dates in the view, or if the dates in the view don’t constitute a full hierarchy (for example, the date includes Year and Day, but not Month), or if they constitute a hierarchy that is not supported (for example, Year, Week, Day).</td>
</tr>
<tr>
<td>The time series is too short to forecast.</td>
<td>Expand the time series in your view to include more date values. This error is returned if there are fewer than four data points</td>
</tr>
<tr>
<td>Issue</td>
<td>Solution</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>A forecast cannot be computed for a time series with Null date values.</td>
<td>Eliminate any Null values from the date field or fields in the view, either by filtering the date field or by using a less detailed date granularity (for example, by switching from months to quarters).</td>
</tr>
<tr>
<td>A forecast cannot be computed when the view contains multiple distinct date fields.</td>
<td>This error is returned if there are multiple date fields in the view. For example, if both Order Date and Ship Date are in the same view, forecasting is not supported.</td>
</tr>
<tr>
<td>The selected 'Aggregate by' value in Forecast Options is not compatible with the visualization.</td>
<td>The date in the view must be compatible with the value of Aggregate by in the Forecast Options dialog box. For example, if Aggregate by is set to Weeks and the date in the view is set to Months, this error occurs. Change one of the dates so that the two are compatible, or set Aggregate by to Automatic.</td>
</tr>
<tr>
<td>A forecast cannot be computed because there are too many missing values.</td>
<td>This error is returned if more than 40% of the data in a pane is missing. Selecting Fill in missing values with zeros in the Forecast Options dialog box will not resolve this error. Aggregate your data to a higher level of detail by removing dimensions or changing the date level, for example from 'weeks' to 'months'. Otherwise, you must modify the source data or use data from a different source.</td>
</tr>
<tr>
<td>There is no measure to forecast.</td>
<td>This error is returned if no measure that can be forecast is present in the view. Forecast measures must be on the Rows or Columns shelf, or on the Marks card.</td>
</tr>
<tr>
<td>The measure to forecast must be a number.</td>
<td>Some measures cannot be interpreted numerically and therefore cannot be forecast.</td>
</tr>
<tr>
<td>A forecast cannot be computed for a dimension.</td>
<td>The value to be forecast must be a measure, and not a dimension.</td>
</tr>
<tr>
<td>There is too much data to compute a forecast.</td>
<td>Forecasting is not possible when the result set from the query is too large. The limit is about 10,000 rows. To fix the forecast, aggregate the time series value at a higher level (for example, Month instead of Week) or filter the data.</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>A forecast cannot be computed because the data is divided into too many rows, columns, or colors.</td>
<td>Simplify the view to resolve the error by filtering or removing some of the dimensions.</td>
</tr>
<tr>
<td>A forecast cannot be computed because the view contains table calculations.</td>
<td>Create a version of the view that does not contain table calculations.</td>
</tr>
<tr>
<td>A forecast cannot be computed because there is a measure on the Filters shelf.</td>
<td>Remove the measure from the Filters shelf.</td>
</tr>
<tr>
<td>A forecast cannot be computed because <strong>Aggregate Measures</strong> is not selected.</td>
<td><strong>Aggregate Measures</strong> is an option on the Analysis menu. See [Data Aggregation in Tableau](page 279) and [How to Disaggregate Data](page 289).</td>
</tr>
<tr>
<td>A forecast cannot be computed because the view contains percent calculations.</td>
<td><strong>Percentage of</strong> is an option on the Analysis menu. See [Calculate Percentages in Tableau](page 1211).</td>
</tr>
<tr>
<td>A forecast cannot be computed because Grand Totals or Subtotals is enabled.</td>
<td>These options are controlled from the <strong>Totals</strong> command in the Analysis menu. See [Show Totals in a Visualization](page 1754).</td>
</tr>
<tr>
<td>A multiplicative model cannot be computed because the measure to be forecast has one or more values that are less than or equal to zero.</td>
<td>You have created a custom model with Trend or Seasonality set to <strong>Multiplicative</strong>. Change this value, or set the Forecast Model to <strong>Automatic</strong>.</td>
</tr>
<tr>
<td>Error Description</td>
<td>Suggested Action</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>A model with multiplicative trend and additive season is not allowed because it is numerically unstable.</td>
<td>You have created a custom model configured as described in the error message. Change the settings for the custom model, or set the Forecast Model to <strong>Automatic</strong>.</td>
</tr>
<tr>
<td>A seasonal model cannot be computed because the time series is too short.</td>
<td>Expand the time series in your view to include more date values.</td>
</tr>
<tr>
<td>The selected multiplicative model cannot be computed because some of the data is too close to zero relative to the rest of the data.</td>
<td>You have created a custom model configured as described in the error message. Change the settings for the custom model, or set the Forecast Model to <strong>Automatic</strong>.</td>
</tr>
</tbody>
</table>
Explore and Analyze Data in a View

When you are ready to explore and analyze your view, Tableau offers a selection of dynamic data inspection tools that help you isolate the data of interest. For example, if you have a dense data view, you can focus on a particular region, select a group of outliers, and view the underlying data source rows for each mark.

Watch a Video: To see related concepts demonstrated in Tableau, watch Getting Started with Visual Analytics, a 6-minute free training video. Use your tableau.com account to sign in.

In this article

- Select marks to highlight data points in the view
- Analyze selected marks using tooltips
- Highlight data by category in tooltips
- Explore data in maps
- Add analytics objects to the view
- Compare marks data with recalculated lines

Select marks to highlight data points in the view

The simplest way to add highlighting to a view is to select the marks you want to highlight. When you manually select a mark in the view, all other marks are dimmed to draw attention to the selection. Your selection is saved with the workbook.

For more information on how to use selection tools, see Select Marks on page 2612. Also see Legend Highlighting on page 1797, Highlight Data Points in Context on page 1800, Highlight Toolbar Button on page 1803, and Highlight Actions on page 1794.

You can select multiple marks by holding down the Ctrl key on your keyboard (Command key on Mac) while you select each mark. You can also use one of the advanced selection tools to select marks within a specific shape.
You can also select marks and headers to filter what is being displayed in the view. For more details, see **Select to keep or exclude data points in your view** on page 1163 and **Select headers to filter data** on page 1164.

---

**Analyze selected marks using tooltips**

When you select one or multiple marks in a view, and then continue to hover in the same spot, several options for inspecting your data are available in the tooltip.

**Tableau Desktop version**

**Web version**
Tooltips provide additional information about the marks in your view, and provide the following data analysis options:

- Keep only the selected marks in the view. For more information, see Select to keep or exclude data points in your view on page 1163.

- Exclude the selected marks from the view. For more information, see Select to keep or exclude data points in your view on page 1163.

- Create a group based on the selected marks. For more information, see Correct Data Errors or Combine Dimension Members by Grouping Your Data on page 1002.

- Create a set that contains the selected marks (Tableau Desktop only). For more information, see Create Sets on page 1004.

- View the underlying data of the marks selected. For more information, see View Underlying Data on page 2616.

For related information on Tooltips, see Tooltips on page 214.

For details on related analysis tools, see View Toolbar on page 2610, Select Marks on page 2612, Pan and Zoom on page 2615, Undo and Redo on page 2616, and Drop Lines on page 1678.

Highlight data by category in tooltips

If you include discrete (categorical) dimensions or measures in your tooltip, you can use them to select marks in the view that have the same values. If Tooltip selection is turned on for your worksheet, then these fields show as underlined text when you mouse over them. Clicking on the tooltip selects all the marks in the view that have the same value.
In the example below, when you click on Bill Shonely (the value for Customer Name) in the tooltip, any mark in the view that also includes Bill Shonely as a customer is selected and all other marks are dimmed.

![Example of selecting marks based on customer name in tooltip]

To turn this functionality off or on, select the **Allow selection by category** check box in the Edit Tooltips dialog box. For more information about setting tooltip properties, see Add **tooltips to marks** on page 1133.

---

**Explore data in maps**

Tableau maps can help you quickly find locations and analyze data worldwide. There are many ways you can explore and interact with map views. You can zoom in and out, pan, and select marks with the view toolbar, and even search for locations worldwide with map search.
For more details, see Explore Data in Maps on page 2086, Search for Locations in Your Map on page 2088, and Measure Distances Between Data Points and Locations in a Map on page 2089.

Add analytics objects to the view (trend lines, forecasts, reference lines and bands)

Drag reference lines, box plots, trend lines forecasts, and other items into your view from the Analytics pane, which appears on the left side of the workspace. Toggle between the Data pane and the Analytics pane by clicking the tabs at the top of the side bar. For more details, see Apply Advanced Analysis to a View (Analytics Pane) on page 174.
Tableau Desktop Analytics pane

In Tableau Desktop, options for adding Analytics objects to the view are available the Analytics pane or menu, or in context in the view. For example, reference lines and bands are available when you edit an axis, and trend lines and forecasts are available from the Analysis menu.

The Analytics pane provides drag-and-drop access for the various options.

On the web, most Analytics objects are available from the Analytics pane.

Compare marks data with recalculated lines

If there is an analytics object in your view, such as an average line, constant line, trend line, reference line or distribution line or band, selecting one or multiple marks in the view lets you instantly compare the analytical data for the selected marks to all data in the view.
For example, selecting marks in a view that contains a trend line creates a second, recalculated trend line, the value of which is determined by the selected marks only, so you can compare that trend to the overall trend.

Recalculated lines are displayed by default when you select marks in a view that already contains analytics objects. In Tableau Desktop, if you don’t want recalculated lines to be created you can turn them off.

**Turn off recalculated lines (Tableau Desktop only):**

1. Select an analytics object in the view, such as a trend line, and click **Edit**.

2. In the **Edit** dialog box, clear **Show recalculated line for highlighted or selected data points**.
Alternatively, you can right-click (control-click on Mac) an analytics object in the view and clear **Show Recalculated Line**.

When you return to the view and select or highlight marks, recalculated lines will not appear. For more information about how to highlight marks, see **Select Marks to Highlight** on page 1796.

**Recalculated lines and highlight actions**

Recalculated lines also work with highlighting actions, both in worksheets and dashboards.

For example, if you create a highlight action in a dashboard, selecting marks in one sheet will highlight marks in the other sheets in the dashboard. If those other sheets have trend lines, reference lines, or other analytics objects, recalculated lines will appear as the views update.
View Toolbar

The view toolbar appears in the upper-left corner of the view and lets you select marks, zoom in and out, and pan in the view. By default, the view toolbar appears when you hover over a map view, but you can also show the view toolbar in other types of views, such as scatter plots or box plots.

For more information about showing and hiding the view toolbar in map views, see Customize How People Interact with your Map on page 2069.
Show the view toolbar

To show the View Toolbar use one of the following options:

Tableau Desktop

- In a worksheet or dashboard, right-click (control-click on Mac) anywhere in the view, and then select Show View Toolbar.

Tableau Server or Tableau Online

- From a worksheet, from the top menu, select Worksheet > Show View Toolbar.
- From a dashboard, select the zone where you want to show the View Toolbar and do one of the following:
  - From the top menu, select Worksheet > Show View Toolbar.
  - Click the drop-down arrow for the selected zone and select View Toolbar from the context menu.

Set when the view toolbar appears in the view

- Select Worksheet > Show View Toolbar (in a dashboard, select a view, then select Worksheet > Show View Toolbar), and then select one of the following options:
  - Automatic – The toolbar appears only when you hover over a map view.
  - Show on hover – The toolbar appears when you hover over the selected view. You can select this option for any type of view.
  - Hide – The toolbar does not appear in the selected view.

Note: To improve usability on mobile devices, the view toolbar always appears when interacting with scrollable views or multiple selections.

Hide the view toolbar

To hide the View Toolbar use one of the following options:
Tableau Desktop

- In a worksheet or dashboard, right-click (control-click on Mac) anywhere in the view, and then select **Hide View Toolbar**.

Tableau Server or Tableau Online

- For a worksheet or dashboard, from the top menu, select **Worksheet > Show View Toolbar > Hide**.
- From a dashboard, select the zone where you want to show the View Toolbar, and then right-click (control-click on Mac), click the drop-down arrow and then select **View Toolbar > Hide** from the context menu.

**Note:** If the view toolbar is hidden, you can still use keyboard shortcuts to select marks, zoom, and pan. For more information, see **Keyboard Shortcuts** on page 2761.

Select Marks

You can select marks in the view to inspect your data. When you select a mark or a subset of marks in the view, you can see information about the marks in the tooltip that appears. You can also quickly filter the marks you select from the view, as well as view their underlying data. For more information, see **Tooltips** on page 214.

Select marks

Click an individual mark to select it. After you select a mark, hold down the Ctrl key (Command key on Mac) to add more marks to a selection.

To select multiple marks using the default selection tool, click and drag across the view. In most views, the Rectangular selection tool is the default tool.

You can also use the Radial, Rectangular, and Lasso tools on the view toolbar to select multiple marks. For more information about the view toolbar, see **View Toolbar** on page 2610.

**Note:** In Tableau Server and Tableau Online, the View Toolbar is only available for maps.
Radial selection tool

The Radial tool selects marks within a circular area. To use the Radial tool, hover over the arrow on the view toolbar, click the Radial tool button, and then click and drag across the view.

You can also measure distance in a map view with the Radial tool. For more information, see Measure Distances Between Data Points and Locations in a Map on page 2089.

Rectangular selection tool

The Rectangular tool selects marks within a rectangular shape. To use the Rectangular tool, hover over the arrow on the view toolbar, click the Rectangular tool button, and then click and drag across the view.
Note: The Rectangular tool is the default tool in most views, and may not appear in the view toolbar. In this case, you can drag across the view to use the rectangular tool.

Lasso selection tool

With the Lasso tool, you can select multiple marks by drawing a freehand shape around them. This tool is useful when you want to include only certain marks, and exclude others around them.

To use the Lasso tool, hover over the arrow on the view toolbar, click the Lasso tool button, and then draw a freehand shape around the marks you want to select.
Pan and Zoom

The pan tool and zoom controls help you interact with the view and inspect your data. They are located in the upper-left corner of the view, on the view toolbar. For more information, see View Toolbar on page 2610.

You can use the zoom controls to zoom in and out, zoom to a specific area, and fix or reset the axes in the view. Use the pan tool to move quickly around the view.

Zoom in and out

On the view toolbar, click the Zoom In button + to zoom in and the Zoom Out button - to zoom out. If the view toolbar is hidden, double-click the view to zoom in; to zoom out, hold down Shift, and then double-click the view.

Zoom to a specific area

To zoom in to a specific area of the view, click the Zoom Area tool button on the view toolbar, and then drag to create the zoom area. If the view toolbar is hidden, hold down Ctrl +
Shift (Command-Shift on Mac) to use the Zoom Area tool.

Reset the view

After you zoom in or out, the axes in the view are fixed to a specific range. To quickly reset the axes so they automatically zoom the view to all of your data, do one of the following:

- In Tableau Desktop, click the Reset Axes button on the view toolbar.
- In Tableau Server or Tableau Online, click the Zoom Home button on the view toolbar.

Pan

To pan, do one of the following:

- Hold down Shift, and then drag across the view.
- On the view toolbar, hover over the arrow, select the Pan tool, and then click and drag across the view.

Undo and Redo

You can perform unlimited undo and redo of your actions. You can undo almost all actions in Tableau by pressing the **Undo** button on the toolbar. Likewise, you can redo almost all actions by pressing the **Redo** button on the toolbar.

In this regard, every workbook behaves like a web browser. You can quickly return to a previous view. Or you can browse all the views of a data source that you have created. Tableau saves the undo/redos history across all worksheets until you exit. The history is not saved between sessions.

Summary Card

The **Summary Card**, available on the **Show/Hide Cards** toolbar menu, provides a quick view of information about a selection or the entire data source.
When you select data in the view, the **Summary Card** updates to show you information only for the data within the selection:

By default, the **Summary Card** shows Sum, Average, Minimum, Maximum, and Median values for the data in the view. The summary card values update to show these values for the current selection of data points. (Average is computed by summing all relevant values and then dividing by the total number of values. Median is computed by sorting values from lowest to highest and then selecting the middle value.) The Count value at the top of the card indicates the number of marks in the view or selection.

You can use the drop-down menu for the **Summary Card** to show additional statistics:

- **Standard Deviation**
  
  A measure of data spread around its average, measured in the same units as the data itself. The sample standard deviation is an unbiased estimate of the population standard deviation given a slight correction. This standard deviation includes the correction.

- **First Quartile**
  
  A measure of location that is commonly used with other quartiles to provide a robust measure of spread. Robust in this case means not as sensitive to outliers as the standard deviation. The first quartile is the 25th percentile, typically the lower line in a boxplot.
• **Third Quartile**
  A measure of location that is commonly used with other quartiles to provide a robust measure of spread. Robust in this case means not as sensitive to outliers as the standard deviation. The third quartile is the 75th percentile, typically the upper line in a boxplot.

• **Skewness**
  A measure of the tendency of your data to have extreme values to one side. Positive skewness means the extreme values are to the right, while negative skewness means the extreme values are to the left.

• **Excess Kurtosis**
  A measure of the tendency of your data to have more extreme or outlying values than a normal distribution. A normal distribution has a kurtosis of 3 so this value is kurtosis minus three.

**View Underlying Data**

The View Data command lets you display the values for all rows in the data source that underlie a set of marks in the view. It also shows you the summary data based on the aggregations in the view. You can view data to verify the aggregated value associated with a mark, or to isolate and export the individual rows associated with data of interest, such as outliers.

You can view data for a selection of marks, for the fields in the Data pane, and when you’re connecting to data.

The View Data command works with all relational and multi-dimensional databases except Oracle Essbase and SAP® Business Information Warehouse databases. While you can view data with the Microsoft Analysis Services and the Teradata OLAP connector multi-dimensional databases, the database must be drill-through enabled; in addition there are some restrictions to the data you can see. Multi-dimensional data sources are supported only in Tableau Desktop on Windows.

In the view below, sales for two product dimensions (**Category** and **Sub-Category**) are displayed as a bar chart. Suppose you wanted to view data for the largest marks in each pane. To do this, you would do one of the following:

- In Tableau Desktop, select the marks, right-click (control-click on Mac) in the view, and select **View Data** on the context menu. Alternatively, you could select the **Analysis** > **View Data** menu item.
In Tableau Server or Tableau Online, select the marks and click **View Data** on the Tooltip menu.

### Tableau Desktop version vs. Web version

**Viewing data** may not return any records if you are using a field that contains floating point values as a dimension. This is due to the precision of the data source and mainly occurs when you are connected to Microsoft Excel, Microsoft Access, or text files.

**Summary Data**

Summarized data is shown on the **Summary** tab. Summarized data is a text table of the aggregated data for the fields shown in the view.
Full Data

All data for the selected marks are displayed on the **Full Data** tab. In the lower right of the dialog box you can see the number of rows in the underlying data.

If you’re using data blending, the **Full Data** tab shows only the data from the primary data source.

<table>
<thead>
<tr>
<th>Tableau Desktop version</th>
<th>Web version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In Tableau Desktop

Sort the data by clicking one or more column headers. To restore the original sort order, click the header repeatedly until it is no longer highlighted with a sort arrow.

By default, Show all fields is selected. Clear this option to only show the columns used on shelves (or fields referenced by a calculation used on a shelf) in the current worksheet.

If you want to export one or more data source rows, select the data points of interest by selecting the row and then clicking Copy to copy the selected data.

In Tableau Server or Tableau Online

Click the Download all rows as a text file link to download all rows of data and open them in .csv file. If you want to see all fields related to the selected marks, select the Show all columns check box and then click the link to download the rows of data.
View Data (Microsoft Analysis Services)

In Tableau Desktop, view Data with a Microsoft Analysis Services database works almost the same way it does with relational data sources. The difference is that a Microsoft Analysis Services cube is generally set up and configured by an administrator who decides whether it is enabled for drill-through and the fields that a user is allowed to see. That means that when you try to view data using a database that is not enabled, you may get an error message alerting you that the cube is not enabled for drill-through.

In addition, Microsoft Analysis Services databases limit viewing data to a single mark at a time. More precisely, viewing the data (which uses MDX drill-through) is not an option when the selection of mark(s) is defined by more than one value of a dimension.

When you are viewing underlying data for a field, the **Show all fields** option is checked and disabled by default. With a Microsoft Analysis Services database, only the fields specified by the administrator are shown, so you cannot choose to include all data source fields in the dialog box.

Describing the View

Occasionally you may want to succinctly summarize an analysis you have completed on a worksheet. You might then want to remind yourself of what it shows (the filters that are applied, etc.), and finally, you may want to share a summary of the analysis with someone else.

When you choose **Worksheet > Describe Sheet**, you can view a description of the workbook, data source, fields and layout of the current worksheet. This summary includes the Caption in the first line, but provides other important summary information. This information can be copied and exported to other applications using the Clipboard.

**Note:** If you have Trend Lines turned on, the Describe Sheet dialog box includes information about the trend line model. For more information, see *Assess Trend Line Significance* on page 1674. If you have Forecasting turned on, the Describe Sheet dialog box includes information about estimated data. For more information, see *Forecast Descriptions* on page 1723.
Showcase Insights

This section describes the various ways you spotlight data insights. Read the following articles for information on how to show totals, add annotations, and embed visualizations in tooltips when you hover over a data point.

Watch a video: For a 6-minute walk through on how to embed visualizations within a tooltip, watch the Viz in Tooltip free training video. Use your tableau.com account to sign in.

For a 7-minute walk through on how to create tooltips in Tableau, watch the Basic Tooltips free training video. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Show Totals in a Visualization

This article demonstrates how to show grand totals and subtotals in a visualization, as well as how to customize how those totals are calculated and where they appear in the visualization.
In this article

Show grand totals below
Show subtotals on page 1761
Move totals on page 1762
Configure total aggregation on page 1762

Show grand totals

To show grand totals in a visualization:

1. Click the Analytics pane.

2. In the Analytics pane, under Summarize, drag Totals into the Add Totals dialog, and drop it over either the Row Grand Totals or Column Grand Totals option.

Row grand totals appear automatically on the right-side of the visualization. Column grand totals appear automatically at the bottom of the visualization. For information how to move where totals appear, see Move totals on page 1762.
In order to turn on grand totals:

- The view must have at least one header – Headers are displayed whenever you place a dimension on the **Columns** shelf or the **Rows** shelf. If column headers are displayed, you can calculate grand totals for columns. If row headers are displayed, you can calculate grand totals for rows.

- Measures must be aggregated – The aggregation determines the values displayed for the totals. See **Grand totals and aggregations** on page 1758 for more information.

- Grand totals cannot be applied to continuous dimensions.

You can also display totals for graphical views of data. In the figure below, only column totals are calculated because the table contains only column headers.

**Note:** By default, totals are computed on the server if you are connected to a Microsoft Analysis Services data source, and locally if you are connected to an Essbase data source, using the aggregation specified in the cube. Refer to **Configure total aggregation** on page 1762 to learn more.
Options for calculating grand totals

When you first turn on grand totals, the totals are computed using disaggregated data in the underlying data source. Consider the following view:

The average that you see at the right of the first row under Grand Total is $339. But if you compute the average for the four values you see in the row ($329, $306, $390, $348), the result comes to $343.25, not $339. The discrepancy is due to the fact that Tableau is averaging the data in the underlying data source. In this case there are more than four numbers to average, perhaps many more. The result is derived by averaging all values that have the properties Ship Mode = First Class, Category = Furniture, without regard to region.

To see totals that correspond to the numbers you see in your view requires two averaging operations in Tableau: first, values for the individual marks (or cells) in the view must be derived—for example, by averaging all values that have the properties Ship Mode = First Class, Category = Furniture, and Region = Central. Then the results for each region must be derived, by averaging the results for the individual marks. Fortunately, you do not need to perform two operations. To display a result of this type, from the Analysis menu choose Totals > Total All Using > Average. Now the average is performed on the values you see, and not on the disaggregated data in the data source:
This kind of total is sometimes referred to as a two-pass total, because the average you see in the grand total column is aggregated twice—once to derive the column or row value, and then again across column or rows to derive the grand total.

**Grand totals and aggregations**

When you turn on grand totals, the initial values are computing using the current aggregation for the fields in the view. In this case, totals are based on the underlying data rather than the data in the view.

For example, if you are totaling the **SUM(Profit)** for several products, the grand total will be the sum of the sums of profit. For aggregations such as SUM, you can easily verify the grand total because a summation of a group of sums is still a summation. However, be aware that your results may be unexpected when using other aggregations, especially custom aggregations. For details, see **Configure total aggregation** on page 1762. You can verify any calculation such as an aggregation or a grand total by viewing the underlying disaggregated data.

The following table summarizes the standard aggregations and the grand totals that are calculated by default when, from the Analysis menu, **Totals >Total All Using** is set to the default value **Automatic**.

---

**Note:** Only Automatic totals are available for table calculations and fields from a secondary data source. Total aggregations cannot be applied to table calculations or fields from a secondary data source.
For more information, see [Showing Grand Totals with Blended Data](#) and [Grand Totals and Subtotals Do Not Show Expected Numbers With Table Calculations](#) in the Tableau Knowledge Base.

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>Calculation Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>Shows the sum of the values shown in the row or column.</td>
</tr>
<tr>
<td>Average</td>
<td>Shows the average of the values shown in the row or column.</td>
</tr>
<tr>
<td>Median</td>
<td>Shows the median for the values shown in the row or column.</td>
</tr>
<tr>
<td>Count; Count Distinct</td>
<td>Shows how many values or distinct values are displayed in the rows and columns in</td>
</tr>
<tr>
<td>Aggregation</td>
<td>Calculation Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Minimum</td>
<td>Shows the minimum value shown in the row or column.</td>
</tr>
<tr>
<td>Maximum</td>
<td>Shows the maximum value shown in the row or column.</td>
</tr>
<tr>
<td>Percentile</td>
<td>Shows the average percentile for all values shown in the row or column.</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>The grand total using standard deviation is the standard deviation of the values shown in the row or column.</td>
</tr>
<tr>
<td>Aggregation</td>
<td>Calculation Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Variance</td>
<td>The grand total using variance is not the variance of the rows and columns in which they reside but rather of the underlying data behind the row or column.</td>
</tr>
</tbody>
</table>

Show subtotals

To show subtotals in a visualization:

1. Click the **Analytics** pane.

2. In the **Analytics** pane, under Summarize, drag **Totals** into the Add Totals dialog, and
Move totals

By default, row grand totals and subtotals appear on the right of the view, and column grand totals and subtotals appear at the bottom of the view.

In Tableau Desktop, you can also select to display totals on the left or top of the view.

To move row totals to the left of the view:

- Select Analysis > Totals, and then select Row Totals to Left.

To move column totals to the top of the view:

- Select Analysis > Totals, and then select Column Totals to Top.

Configure total aggregation

When totals are turned on in the visualization (either grand totals, subtotals, or both), you can specify how totals should be computed. For example, you can choose to calculate totals using a sum, average, minimum, or maximum.
To configure all totals:

- Select Analysis > Totals > Total All Using, and then select an aggregation from the list.

To configure totals for a specific field:

- Right-click (Control-click on a Mac) a field in the view, select Total using, and then select an aggregation from the list.

When you choose Automatic, totals are based on the underlying data, which is disaggregated, and not on the data in the view. See Options for calculating grand totals on page 1757. For details on how Tableau computes totals using the current aggregation, see Grand totals and aggregations on page 1758.

When you choose any of the other values (Sum, Average, Minimum, or Maximum), all totals are computed using the selected aggregation. The computations are performed on the aggregated data you see in the view.

An additional value, Server, may be available. Server computation is not always available and sometimes the totals will be blank for specific members in the view. When using server computation keep in mind the following information:

- Server computation is only available for ASO cubes.
- Server computation is not available for dynamic hierarchies. This means that if the members in the view are part of a dimension or hierarchy that is tagged as dynamic, they cannot be included in the set of values you are using to compute the totals and will show up as blank in the view.
- If you are computing totals for a calculated field whose formula makes assumptions about other calculated members at different levels in the hierarchy, the totals will display as blank in the view.

See Also

Apply Advanced Analysis to a View (Analytics Pane) on page 174

Add Annotations

You can add annotations to a visualization to call out a specific mark, a specific point, such as a location on a map, or an area, such as a cluster of scatter marks.
**Note:** In web authoring, you can create and delete annotations by right-clicking the view (in a worksheet or dashboard), and then selecting point or mark annotation. Right-click the annotation again to edit or remove the annotation. Formatting and moving annotations are not currently supported on the web.

In this article

- **Add an annotation below**
- **Edit an annotation** on the next page
- **Rearrange an annotation** on page 1766
- **Format annotations** on page 1772
- **Remove an annotation** on page 1773

Add an annotation

To add an annotation to your viz:

1. In a worksheet, right-click (control-click on Mac) a data point or a spot on the viz where you want to add an annotation and select **Annotate**, and then select the type of annotation you want to add.

   There are three types of annotations in Tableau:
   - **Mark** - select this option to add an annotation that is associated with the selected mark. This option is only available if a data point (mark) is selected.
   - **Point** - select this option to annotate a specific point in the view.
   - **Area** - select this option to annotate an area in the viz, such as a cluster of outliers or a targeted region.

2. In the Edit Annotation dialog box that opens, type the text you want to show in the annotation.

   Use the **Insert** menu to insert dynamic variables into the annotation text. For example, the annotation can display data values that update as the underlying data changes. The dynamic variables that are available are dependent on whether you are annotating a mark, point, or area.

3. When finished, click **OK**.
The viz updates with your annotation.

Edit an annotation

To edit an annotation:
1. Right-click (control-click on a Mac) the annotation in the viz and select **Edit**.
2. In the Edit Annotation dialog box that opens, edit the annotation text and then click **OK**.

---

**Rearrange an annotation**

After you add an annotation, you can move it around, resize it, adjust the line, and move the text. Each type of annotation can be rearranged and modified in different ways. This section discusses how to rearrange, resize, and modify each type of annotation.

**Mark annotations**

When you select a mark annotation the body and line are selected and several resize handles display. Use these handles to resize the body and line.
To reposition the body

- Click and drag the body of the selected annotation to a new position.

To resize the body

- Click and drag the body resize handle left and right. The text and height are automatically adjusted to fit the width of the body.

To resize the line

- Click and drag the line resize handle.

Point Annotations
A point annotation marks a specific point in the view, such as a reference line or a value on an axis. Point annotations display as text with a line. When you select a point annotation, several resize handles display. Use these handles to reposition and resize the body and line.

To reposition the body:

- Click and drag the body of the selected annotation to a new position. As you move the body, the line is automatically resized so that it continues to point at the specific point you
selected.

To resize the body:

- Click and drag the side resize handles left and right. The text and height are automatically adjusted to fit the width of the body.

To move the line end point:

- Click and drag the end point of the line so that it points at a new location.
Area Annotations

An area annotation is a way to highlight or call out an area in the view. Area annotations are not associated with any particular mark, in fact, these annotations are commonly used to call out several marks. When you select an area annotation, several resize handles and two text handles display. Use these handles to reposition and resize the box and text.

To reposition the box

- Click and drag the box of the selected annotation to a new position.
To resize the box

- Click and drag one of the box resize handles.

To reposition the text

- Click and drag the center text handle to a new position.
To resize the text width

- Click and drag the right text handle left and right. The text height is automatically adjusted to fit the width.

Format annotations

You can modify the text, body, and line of an annotation. For example, you can specify whether the body should be a box, a single edge, or not shown at all. Additionally, you can specify
whether the lines on mark and point annotations end with an arrow, dot, or a simple line.

To format annotations:

1. Select one or more annotations, right-click (control-click on Mac) one of the selected annotations, and then select **Format**.

   The **Format** pane opens on the left-side of the workspace, over the **Data** pane.

2. In the **Format** pane, use the drop-downs to specify font properties, text alignment, line style, and shading.

   ![Format pane](image)

Remove an annotation

To remove annotations from the viz:

1. Select one or more annotations in the viz.

2. Right-click (control-click on Mac) one of the selected annotations and select **Remove**.

See Also

**Show and Hide Mark Labels** on page 2180

Create Views in Tooltips (Viz in Tooltip)

As you craft views and look for ways to reveal more details about data to your audience, you can embed visualizations within tooltips—aka "Viz in Tooltip."
When a user hovers over a mark, the tooltip displays relevant data and details from another visualization filtered to that mark.

You can show related vizzes in tooltips to help your audience engage with the data at a different or deeper level, while keeping them in the current context and maximizing the space available for the current view.

**Watch a video:** To see related concepts demonstrated in Tableau, watch Viz in Tooltip, a 6-minute free training video. Use your tableau.com account to sign in. For more advanced tips, see Next Level Viz in Tooltip, a 55-minute free video presentation.

In this article

- General steps to create a Viz in Tooltip
- Examples of Viz in Tooltip
- Configure a Viz in Tooltip
- Change the size
- Change the filters
- Hide or show a Viz in Tooltip sheet
- Tips on using and configuring Viz in Tooltips
General steps to create a Viz in Tooltip

1. Create a visualization in the source worksheet in Tableau Desktop. Viz in Tooltips can be viewed on the web, but can only be configured in Tableau Desktop.

2. Create a visualization in a target worksheet view to serve as the Viz in Tooltip. Give the worksheet a name that helps you identify it as a Viz in Tooltip.

3. In the source worksheet, click Tooltip in the Marks card. In the Tooltip Editor, insert a reference to the Viz in Tooltip target worksheet. For details, see Configure a Viz in Tooltip below.

4. Optional: Hide the target sheet for the Viz inTooltip. For details, see Create Views in Tooltips (Viz in Tooltip) on page 1773.

5. Test the resulting Viz in Tooltip by hovering over different marks in the source worksheet view. If the Viz in Tooltip is too large for the tooltip window, adjust the height and width of the target worksheet visualization. You might also consider simplifying the structure and detail in the target visualization. For details, see Change the size of the Viz in Tooltip on page 1779.

   Note: By default, Viz in Tooltip is filtered on All Fields. Change the level of detail for Viz in Tooltip by defining a filter on Selected Fields. For details, see Change the filter for the Viz in Tooltip on page 1780.

For information on how to edit regular tooltips, see Format tooltips (Tableau Desktop only) on page 2385. To see related concepts demonstrated in Tableau, watch Basic Tooltips, a 7-minute free training video. Use your tableau.com account to sign in.

Configure a Viz in Tooltip

You will need a source worksheet visualization and a target worksheet visualization to create a Viz in Tooltip.

These steps use the example of a source view that is a map showing sales profits by state, and a target sheet with a chart that shows profits by product sub-category.
Create the source and target visualizations in a worksheet

1. In Tableau Desktop, figure out the worksheet that you want to use as your source visualization. Or build a new visualization on a new worksheet. This will be your source view.

2. Create a new worksheet to serve as the target visualization.
   Keep the number of filters used in the target view to a minimum.

3. Name the target view so that you will be able to identify it in a list of other sheets.

Insert a reference to the target worksheet in the source worksheet tooltip

1. In the source sheet, click the Tooltip button in the Marks card to open the Tooltip Editor.

2. Click the Insert menu in the Tooltip Editor. In the Insert menu, select Sheets, and then select a target sheet.

   For example:
The markup for the Viz in Tooltip is automatically added. (In this example, the original markup fields have been removed.)
Example of markup automatically generated for Viz in Tooltip

This is the resulting markup for this example:

```xml
<Sheet name="Tooltip: Profit by Sub-Category" maxWidth="300" maxHeight="300" filter="<All Fields>" />
```

By default, the Viz in Tooltip is filtered on All Fields (filtered on all fields possible, and on the most specific level of detail).

Click **OK**.

3. Go back to the source sheet and test the Viz in Tooltip. Hover over different marks to see the resulting Viz in Tooltip. Make adjustments to the target view as necessary to improve the Viz in Tooltip.
The Viz in Tooltip in this example shows a message that indicates that some of the data in the target view is not being shown. If you see this message, you can adjust the height and width settings in the parameters for the Viz in Tooltip to make the view size larger.

Change the size of the Viz in Tooltip

You can manually change the \texttt{maxwidth} and \texttt{maxheight} values to resize the Viz in Tooltip. The default size is 300 by 300 pixels. To change the size, manually replace "300" with another value. If you need to set the value greater than 600 pixels, you might want to reconsider whether the target view is a good candidate for Viz in Tooltip.

1. In the source sheet, click the Tooltip button in the Marks card to open the Tooltip Editor.
2. Select the number value for \texttt{maxwidth} and \texttt{maxheight} and type a different value to replace it. For example:
3. Click OK

Example of a Viz in Tooltip set to 500 pixels width and height. The author also created more space for the header text in the in the target sheet view.

Change the filter for the Viz in Tooltip

By default, Viz in Tooltip is filtered on All Fields. This means the view is filtered on all dimensions in the current view (not including fields on the Filters shelf), at the most specific level of detail.

You can change the level of detail for Viz in Tooltip by defining a filter on selected fields, similar to filtering on Selected Fields in Filter Actions.

1. In the source sheet, click the Tooltip button in the Marks card to open the Tooltip Editor.

2. Place your cursor within the filter value (filter="<place cursor here>"), and then click the Insert menu to select an available field. Or, manually replace the <All
Fields> value with the name of a field in the view. For example:

```xml
<SHEET name="Tooltip: Profit by Sub-Category" maxWidth="300" maxHeight="500" filter="<State>"/>
```

You can also filter more than one selected field by separating the field names with a comma. For example:

```xml
<SHEET name="Tooltip: Profit by Sub-Category" maxWidth="300" maxHeight="500" filter="<Country>,<State>"/>
```

---

## Hide or show a Viz in Tooltip sheet

You can hide the worksheet used for Viz in Tooltip, the same way you might hide sheets used in stories or dashboards.

**Note**: If the source sheet is hidden for a dashboard, you will first need to first unhide that worksheet from its dashboard to access it. For details, see [Hide and Show Sheets in Dashboards or Stories](#) on page 2211.

To hide a Viz in Tooltip worksheet, in the target worksheet that is the Viz in Tooltip, click **Hide**.
To show the Viz in Tooltip worksheet again, in the source worksheet, click **Unhide All Sheets**.
Examples of Viz in Tooltip

When you create a Viz in Tooltip, users can hover over a mark to examine details-on-demand, within the context of the original view. A Viz in Tooltip is a static image of data from another view that is relevant to a mark in the current view. Hovering over or selecting the mark reveals data from another sheet—filtered for that mark—in its tooltip.

Use Viz in Tooltip to show:
• Data at another level of detail

![Sales By Category Chart]

• Different, but relevant data

![Sub-Categories Chart]

• How the value of a mark changes over time
- Multiple visualizations in one tooltip -
- A legend for the main view
When you show related views in tooltips, you can help your audience engage with the data at a deeper level, while maximizing the space available for the current view.

Tips and notes on using and configuring Viz in Tooltip

- You can use worksheets to create a Viz in Tooltip, but you cannot use dashboards or stories to create a Viz in Tooltip.

- Users will be able to see Viz in Tooltips on the web, but you must configure Viz in Tooltip in Tableau Desktop.

- A Viz in Tooltip is a static image of a target view, not an interactive sheet. A Viz in Tooltip cannot have its own Viz in Tooltip.

- You will need a source worksheet visualization and a target worksheet visualization to create a Viz in Tooltip. You will need to create a target view to make it available in the Tooltip Editor.

- Use a standard naming scheme for the target sheets you plan to show in tooltips, such as Tooltip: Name of View. Using a standard naming scheme will help you keep track of views that you are using in tooltips.

- For the target view, consider the size of the view data and how the view will look in the tooltip. You will be able to specify the size for the Viz in Tooltip when you configure it, but you will need to check to see how it is displaying and possibly make adjustments to the view before you publish it.

  Remember that the target view is displayed in the context of the source view. Keeping the target visualization simple can help with performance and reduce cognitive load.

- If you click Show Me in the source sheet and it changes the view structure, all tooltip edits including Viz in Tooltip references will be reset. You will need to reconfigure the Viz in Tooltip.

- One target sheet can be referenced by one Viz in Tooltip source sheet at a time, because filters are applied directly to the referenced sheet. When a sheet is already being used as a target sheet in a tooltip, it becomes unavailable for selection in the Tooltip Editor.

- By default, Viz in Tooltip is filtered on All Fields, which considers all fields in the view (at the most specific level of detail) when identifying matching records. You can change the level of detail for Viz in Tooltip by defining a filter on Selected Fields, similar to filtering on
Selected Fields in Filter Actions. For related details, see Create Advanced Highlight Actions on page 1805.

Filtering on Selected Fields does not work across different data sources. If the source and target views are using different data sources, filtering on All Fields will automatically detect the fields in common (if they share the same alias), and filter on them. Filtering on Selected Fields, however, will not work.
Actions

Tableau allows you to add context and interactivity to your data using actions. There are three kinds of actions in Tableau: Filter, Highlight, and URL actions (Tableau Desktop only).

Filter actions allow you to use the data in one view to filter data in another as you create guided analytical stories. Highlight actions help you call attention to specific results, and URL actions allow you to point to external resources, such as a web page, file, or another Tableau worksheet.

For example, in a dashboard showing home sales by neighborhood you could use actions to help you quickly see relevant information for a selected neighborhood. Select a neighborhood in one view which then highlights the related houses in a map view, filters a list of the houses sold, then opens an external web page showing census data for the neighborhood.

**Note:** Actions behave differently in workbooks that use cube data sources. Cube data sources do not accept actions from relational or other cube data sources. For example, suppose you have a workbook that contains a view that uses a MySQL data source, a second view that uses a cube data source A, and a third view that uses a cube data source B. Actions in a view that use the MySQL data source will not affect the views that use the cube data sources. However, actions in the view that use the cube A data source can affect the view that uses the MySQL data source. The view that uses the cube B data source will not be affected in this case.

Filter Actions

Filter actions send information between worksheets. Typically, a filter action sends information from a selected mark to another sheet showing related information. Behind the scenes, filter actions send data values from the relevant source fields as filters to the target sheet.

For example, in a view showing the sales price of houses, when you select a particular house, a filter action can show all comparable houses in a different view. The source fields for the filter might contain sales price and square footage.

Watch a video to see real-world examples of actions. Filter actions are covered at the 2:20 mark.
Create a filter action

1. Do either of the following:
   - On a worksheet, select **Worksheet > Actions**, and then continue to step 2.
   - On a dashboard, select **Dashboard > Actions**. Or, from the drop-down menu of a sheet, select **Use as Filter**.

2. In the Actions dialog box, click **Add Action**, and then select **Filter**. Or select an existing action, and choose **Edit**.
3. Specify a name for the action.

Tip: If you plan to launch an action from a menu, use a descriptive name so users will understand the action’s purpose. You can use variables in the name that are drawn from the values of the selected field.

4. Select a source sheet or data source. If you select a data source or dashboard, you can select related sheets you want to launch the action from.
5. Specify how the action runs:

- **Hover** - Rest the pointer over a mark in the view to run the action. This option works well for highlight and filter actions within a dashboard.

- **Select** - Click on a mark in the view to run the action. This option works well for all types of actions.

- **Menu** - Right-click (Windows) or Control-click (macOS) a selected mark in the view, and then select an option on the context menu. This option works well for filter and URL actions.

6. Select a target sheet. When you select a dashboard, you can select one or more sheets within it.
7. Specify what happens when the selection is cleared in the view:
   - **Leave the filter** - Continues to show filtered results on the target sheets.
   - **Show all values** - Changes the filter to include all values.
   - **Exclude all values** - Changes the filter to exclude all values. This option is useful when you're building dashboards that only show some sheets if a value in another sheet is selected.

8. Specify the data that you want to show on the target sheets. You can filter on All Fields or Selected Fields.

9. If you chose Selected Fields, click Add Filter.

In the Add Filter dialog box, select source and target data sources and fields. When you run the action from a specific mark on the source sheet, a filter is added to the target sheet that only includes values for the target field that match the source field.

In the comparable houses example, if the Source Field and Target Field are both set to Beds, when you click a sheet link for a house with three bedrooms, the target worksheet shows only three-bedroom houses.
Understanding available target fields

In the Add Filter dialog box, the fields available in the Target Field drop-down list are limited to the data type of the Source Field. For example, if you select a text field for the source, only text fields are available as targets.

If you are connected to a relational data source, you can add sheet links across data sources even if the field names don't match. For example, if one data source has a Latitude field while another has a Lat field, you can associate the fields using the drop down lists in the Add Filter dialog box. When using a multidimensional data source, the destination sheet must use the same data source as the source sheet, and the source and target field names must match. (In Tableau, multidimensional data sources are supported only in Windows.)

Highlight Actions

Highlight actions allow you to call attention to marks of interest by coloring specific marks and dimming all others. You can highlight marks in the view using a variety of tools. For example, you can manually select the marks you want to highlight, use the legend to select related marks, use the highlighter to search for marks in context or create an advanced highlight action.

The following table describes the different methods you can use to highlight marks in a view, dashboard, or story.
<table>
<thead>
<tr>
<th>Highlight method</th>
<th>Benefits</th>
<th>When you might use this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select marks in a view</td>
<td>• Manually select a group of marks to highlight in a view.</td>
<td>• When you want to manually highlight a selection of marks and dim all others.</td>
</tr>
<tr>
<td></td>
<td>• Your selection is saved with the workbook.</td>
<td>• Works well with small domains or views with a small amount of data.</td>
</tr>
<tr>
<td>Legends</td>
<td>• Supports one-way and two-way highlighting.</td>
<td>• When you want to focus on select members in a view and dim all others.</td>
</tr>
<tr>
<td></td>
<td>• Highlight on color, size, or shape.</td>
<td>• When you want to highlight using only the legend or the legend and the view.</td>
</tr>
<tr>
<td></td>
<td>• You can disable or enable the highlighting action for the workbook or sheets from the toolbar.</td>
<td>• Works well with small domains or views with a small amount of data.</td>
</tr>
<tr>
<td></td>
<td>• Your selection is saved with the workbook and can be included in dashboards and stories and when publishing.</td>
<td></td>
</tr>
<tr>
<td>Highlighter</td>
<td>• Search for data points in a view using keywords or select from a drop-down list.</td>
<td>• When you want to highlight a mark or group of marks for a discrete field that is included in the view.</td>
</tr>
<tr>
<td></td>
<td>• Highlight marks while maintaining the context of the other data points.</td>
<td>• When you want to do ad hoc comparisons with instant highlighting.</td>
</tr>
<tr>
<td></td>
<td>• Values automatically update when the underlying data is updated.</td>
<td>• Works well with large domains and large amounts of data.</td>
</tr>
<tr>
<td></td>
<td>• Highlighters added to worksheets also appear on dashboards and stories.</td>
<td></td>
</tr>
<tr>
<td>Actions</td>
<td>• Highlight data based on criteria</td>
<td>• When you want to build</td>
</tr>
</tbody>
</table>
Select Marks to Highlight

The simplest way to add highlighting to a view is to select the marks you want to highlight. When you manually select a mark in the view, all other marks are dimmed to draw attention to the selection. Your selection is saved with the workbook.

You can select multiple marks by holding down the Ctrl key on your keyboard (Command key on Mac) while you select each mark. You can also use one of the advanced selection tools to select marks within a specific shape. For more information on how to use advanced selection tools, see Select Marks on page 2612.
Legend Highlighting

You can use legend highlighting to focus your users' attention on specific marks in a view. When legend highlighting is on, marks associated with the legend item are highlighted and the other marks are dimmed.

You can enable either one-way or two-way highlighting to highlight marks in the view. The icon at the top of the legend indicates which mode you're using.

- One-way highlighting enables you to highlight marks using the values in the legend.
- Two-way highlighting enables you to highlight marks using either the legend or the view. This is the default. When you highlight marks in the view, the matching member in the legend is also highlighted.

You can also turn off highlighting for the workbook or the sheet from the toolbar. This action hides the highlight icon on the legend. For more information about using the toolbar highlighting option, see Highlight Toolbar Button on page 1803.
In this example, the views below show the relationship between order quantity and profit for several products. The view on the left uses the standard functionality for the color legend, where all marks are colored based on their shipping mode. The view on the right uses legend highlighting to call out the products that were delivered via Second Class shipping.

You can easily switch between legend highlighting and normal modes using the legend card menu. For color legends, if you like how a view is highlighted, you can assign the highlight colors to the color palette. The old colors are replaced with the highlight colors.

**Turn on legend highlighting**

1. Click the **Highlight** button 🖊 at the top of the legend.

   If you're using Tableau Desktop you can also turn on legend highlighting by selecting **Highlight Selected Items** on the legend card menu.
This example shows highlighting using the color legend. Color legend highlighting is turned on by default.

### Ship Mode

- First Class
- Same Day
- Second Class
- Standard Class

2. Select an item in the color legend.

After legend highlighting is turned on, you can quickly focus on specific data in the view by selecting different items in the color legend. When you turn on legend highlighting, a highlight action is created that you can modify in the Actions dialog box.

#### Turn off legend highlighting

Click the **Highlight** button at the top of the legend. This changes highlighting to one-way and you can use the legend to highlight matching marks in the view.

### Legend highlighting in Tableau Desktop

If you're using Tableau Desktop you can also turn off legend highlighting by selecting **Highlight Selected Items** on the legend card menu. When you turn legend highlighting off the action is removed from the Actions dialog box.

If you like how the view is highlighted and want to keep a specific member highlighted even when you turn off legend highlight mode, you can assign the highlight colors to the existing color palette. The original color legend is discarded and the highlight colors become the new color palette for the legend.

To assign the highlight colors to the color palette, select **Assign Highlight Colors to Palette** on the color legend card menu.
Highlight Data Points in Context

When you have a view with a large amount of data you might want to explore your data interactively and highlight a specific mark or group of marks while still maintaining the context of where those marks show in your view.

To do this you can turn on the Highlighter for one or more discrete fields that are included in your view and that affect the level of detail (see How dimensions affect the level of detail in the view on page 255 for more information).

You can use keywords to search for matching data points. The highlighter immediately highlights the marks that match or partially match your keyword search. If you update the underlying data source for your view the data shown in the highlighter is automatically updated too.

In the example below, the Highlighter is turned on for the College field. Entering a partial keyword search for Public returns two possible matches. In the view, Tableau highlights the group of marks that match the partial search: Public Affairs and Public Health.
You can turn on highlighters for as many discrete fields in our view as you need. However, you can only highlight using one value set at a time. If you want to include the highlight option on dashboards and stories, turn them on in the worksheet before you add them to the dashboard, or select the worksheet on the dashboard, and then select Analysis > Highlighters from the top menu and select from the fields in the list.

**Tableau Desktop only:** If you want to change the format for the highlighter card after you turn it on, select Format > Highlighter.

Highlighters work across worksheets on a dashboard if the same field is included across those worksheets. If the same field is not found, then no match is found for the highlighted value, and the values in those worksheets are dimmed. If you have multiple highlighters showing on a dashboard, only one highlighter can be active at a time. In this situation, the highlight shows for the highlighter that was used last.

**Turn on highlighting**

1. Right-click on a discrete field (dimension) that is included in the view and impacts the level of detail in the view. Then select Show Highlighter from the context menu.
2. Click in the highlighter dialog box and do one of the following:

- Enter a keyword to search for matching marks to highlight.
- Enter a partial key word to search for all relevant matches that contain your search text.
- Select an item in the drop-down list. You can select one item at a time.
- Hover over items in the drop-down list to highlight marks in the view for an ad hoc analysis experience.
Click the image to replay it

3. Repeat steps 1 and 2 to add additional highlighters.

You can also show mark labels on highlighted marks. To turn on mark labels, on the Marks card, click Label, select the Show mark labels check box and select Highlighted in the Marks to Label section.

Highlight Toolbar Button

Another way to add a highlight action is to use the highlight button in the toolbar. Similar to legend highlighting, the toolbar button lets you highlight a collection of related marks in the view and functions the same way that two-way highlighting does. To turn on highlighting, select the fields you want to use for highlighting on the toolbar menu. The drop-down list shows the active fields in the view. Then select a mark in the view to see the related data.
For example, the view below shows Sales vs. Profit by Region. If you turn on highlighting for Ship Mode, when a mark is selected, all other marks that were shipped via that mark's ship mode are highlighted. In this example you can quickly see all products from across the United States that were shipped via Second Class.
The toolbar menu also lets you highlight on All Fields or Dates and Times. All Fields considers all fields in the view when identifying matching records; Dates and Times considers all date and time fields in the view.

When you use the Highlight toolbar button in Tableau Desktop, an action is created in the Actions dialog box. You can modify the action to create more advanced highlighting behavior. For more information about editing actions, see Create Advanced Highlight Actions below.

Finally, you can use the toolbar button to disable highlighting across the entire workbook or for just the active sheet. These options disable highlighting for legends only. They do not turn off the ability to manually highlight marks or use the Highlighter control.

When you turn off highlighting from the toolbar button, the highlight icon is hidden on the legend, and the Highlight Selected items menu option is grayed out on the context menu for the legend.

Create Advanced Highlight Actions

In Tableau Desktop, you can define more advanced highlight actions using the Actions dialog box. There you can specify source and target sheets and the fields you want to use for highlighting. Follow the steps below to create a highlight action.

Create a highlight action

1. On a worksheet select Worksheet > Actions. From a dashboard, select Dashboard > Actions.

2. In the Actions dialog box click the Add Action button and then select Highlight.
3. Name the action to identify it in the Actions dialog box. Try to make the name descriptive, such as, *Highlight Products Shipped by Delivery Truck*. You can select variables from a drop-down list and use them in the name. Then they are filled in based on the values of the selected field.

4. Use the drop-down list to select the source sheet or data source. If you select a data source or a dashboard sheet, you can further select individual sheets within them.

5. Select how you want to trigger the action. You can select from the following options:
   - **Hover** - Rest the pointer over a mark in the view to run the action. This option works well for highlight and filter actions within a dashboard.
   - **Select** - Click a mark in the view to run the action. This option works well for all types of actions.
   - **Menu** - Right-click (control-click on Mac) a selected mark in the view and then
select an option on the context menu. This option works well for filter and URL actions.

6. Select a target sheet. If you select a dashboard, you can further select individual sheets within the dashboard.

7. Select the fields you want to use for highlighting. Select from the following options:

- **Selected Fields** - Marks in the target sheet are highlighted based on select fields. For example, highlighting using the Ship Mode field will result in an action that highlights all marks in the target sheet that have the same ship mode as the selected mark in the source sheet.

- **Dates and Times** - Marks in the target sheet are highlighted when their date and time match those of the marks selected in the source sheet. The source and target worksheets can only have one date field each, however the date fields can have different names.

- **All Fields** - Marks in the target sheet are highlighted when they match the marks selected in the source sheet. All fields are considered when determining a match.

8. When finished, click **OK** twice to close the dialog boxes and return to the view.
See Also

Highlight Actions on page 1794

Highlight Data Points in Context on page 1800

URL Actions

A URL action is a hyperlink that points to a web page, file, or other web-based resource outside of Tableau. You may use URL actions to create an email or to link to more information about your data that is hosted outside of your data source. To make the link relevant to your data, you can substitute field values of a selection into the URL as parameters.

Make a URL action that opens a web page

1. On a worksheet, select Worksheet > Actions. From a dashboard, select Dashboard > Actions.

2. In the Actions dialog box, click Add Action and then select URL.

3. In the next dialog box, specify a name for the link.

   Make the name descriptive of the action. If you choose to run the action using the menu, the name you specify here becomes the option that shows on the menu. For example,
when linking to more product details, the name could be “Show More Details.” You can use variables in the name, which will be filled in based on the values of the selected field.

4. Use the drop-down list to select a source sheet or data source. If you select a data source or dashboard you can select individual sheets within it (see Filtering Across Multiple Data Sources Using a Parameter in the Tableau Knowledge Base for more information about filtering data sources).

5. Select how users will run the action.

   **If you choose this option...**  
   
   **Hover**  
   Mouses over a mark in the view. This option works well for highlight and filter actions within a dashboard.

   **Select**  
   Clicks a mark in the view. This option works well for all types of actions.

   **Menu**  
   Right-clicks (control-clicks on Mac) a selected mark in the view, then selects an option on the context menu. This option works well for filter and URL actions.

6. Specify the URL.
You can specify a URL with an ftp, http, or https prefix. As a security best practice, other protocols and UNC paths are not supported.

**Note:** You can only specify an ftp address if you do not have a web object already embedded on the dashboard. If you have configured a web object on the same page where a ftp URL action has been created, then the ftp will fail to load.

Just as you can use variables in the name of the URL, you can also use field values and filter values as parameters in the URL. That means that you can send information about each selected mark or filter setting to a given website.

7. (Optional) Select any of the following options:

- **URL Encode Data Values** - select this option if your data contains values that use characters that are not allowable in a URL. For example if one of your data values contains an ampersand, such as “Sales & Finance,” the ampersand must be translated into characters that your browser understands (URL encoded) if you want to include that value in the URL.

- **Allow Multiple Values** - select this option if you are linking to a webpage that can take lists of values as parameters in the link. For example, say you select several products in a view and you want to see each product’s details hosted on a webpage. If the server can load multiple product details based on a list of identifiers (product ID or product name), you could use multi-select to send the list of identifiers as parameters.

When you allow multiple values, you must also define the item delimiter, which is the character that separates each item in the list (for example, a comma). You must also define the Delimiter Escape, which is used if the delimiter character is
Make a URL action that creates an email

1. Select **Worksheet > Actions**.

2. In the Actions dialog box, click the **Add Action** button and select **URL**.

3. In the Source Sheets drop-down list, select the sheet that contains the field with the email addresses you want to send to.

8. Click **OK** twice to close the dialog boxes and return to the view.

URL actions can also point to a web page object in a dashboard. Refer to **Actions and Dashboards** on page 1814 to learn more about how actions work with dashboards.
4. In the URL textbox, type `mailto:`, click the arrow button to select the field in your data that contains the email addresses. Type `?subject=` and enter text for the Subject line if you want to. For example, in dialog below, the subject is Request for information.

5. Type `&body=`, and click the arrow button to select the fields of information that you want to include in the body of the email.

In the example below, the “Email” field contains the email addresses, the subject is “City Information”, and the body text of the email consists of the city and state information that is associated with the email address.
6. (Optional) Display data from your workbook in the body of your email as a vertical list instead of the default horizontal list. For example, suppose you have a horizontal list of cities, such as Chicago, Paris, Barcelona, which you would rather display vertically, like this:

Chicago
Paris
Barcelona

To make the list vertical, in the Edit URL Action dialog box, do the following:

- Under URL Options, verify that the **URL Encode Data Values** check box is clear.

- Under URL Options, select the **Allow multiple values** check box.

- Type `%0a` in the **Item Delimiter** text box to add a line break. This set of characters is the URL-encoded character for a line break.

**Running Actions**

Depending on how the action is created, you can run an action using one of the following three methods:
• **Hover** - rest the pointer over a mark in the view to run the action. This option works well for highlight and filter actions within a dashboard.

• **Select** - click on a mark in the view to run the action. This option works well for all types of actions.

• **Menu** - right-click (control-click on Mac) a selected mark in the view and then select an option on the context menu. This option works well for filter and URL actions.

Links are not always visible for every worksheet and mark. Because links are mapped to specific fields in the data source, links will only be available for the worksheets that use the mapped fields. For example, if you add a hyperlink that uses both Latitude and Longitude as parameters in the link, the link will only be available to worksheets that use Latitude and Longitude in the view. Additionally, the link is only available on marks and headers that contain relevant values.

**Actions and Dashboards**

Actions often have unique behavior when the source or destination is a dashboard. This is because a dashboard can contain multiple views—so a single filter or highlight action can have broad impact. A dashboard is also unique in that it can contain web page objects. Web page objects are associated with URL actions, which you can use to embed a web page in a dashboard.

**Use a Single View to Filter Other Views in a Dashboard**

Imagine you have a dashboard that contains three views about profitability: a map, a bar chart, and a table of customer names. You can use a filter action to make one of the views in your
dashboard, such as the map, the "master." When your users select a region in the map, the data in the other views is filtered so that it relates to just that region.

1. On the dashboard, select the view you want to use as a filter.
2. On the view's shortcut menu, choose Use as Filter. You can perform the same action by clicking the Use as Filter icon.

You can also use filter actions to filter the data on a dashboard when the data comes from multiple data sources. For more information, see Filtering Across Multiple Data Sources Using a Parameter in the Tableau Knowledge Base.

For more information about filter actions, see Filter Actions on page 1789.

Use Multiple Views to Filter Other Views in a Dashboard

Similar to how you can use a single view to filter other views in a dashboard, you can also use multiple views as a filter. This is sometimes called having a "multi-master" dashboard. The trick is to not only use the master views as filters, but to also disable their ability to be filtered themselves.
1. Create or open a dashboard that has at least three views.

![Dashboard screenshot]

2. Select the first view that you want to use as a filter (such as a map), and from its shortcut menu, select **Use as Filter**.
3. Open the same view’s shortcut menu again and select **Ignore Actions**. This ensures that other filter actions, including the one you’ll create next, will not affect this view.
4. Repeat the steps 2 and 3 for any other views you want to use as a filter.

Now, selecting marks in a master view filters data in one or more detail views—all without affecting any other masters.
Use a URL Action to Display a Web Page Inside a Dashboard (Tableau Desktop only)

You can use a URL action with a web page object to display information from the web inside your dashboard. For example, you might have a dashboard that shows profits by country. In addition to showing the profit data in your dashboard, you also want to display supplemental information about the countries, and that information is on a web site. Here’s how to do it.

1. From your dashboard, select **Dashboard > Actions**.
2. In the Actions dialog box, click **Add Action** and then select **URL**.
3. In the next dialog box, specify a name for the link. If you choose to run the action using a menu, such as a menu option on a tooltip, the name you specify here is what’s displayed.
4. Under Source Sheets, select the view or data source that will initiate the action. For example, if you want the action to be initiated when a user clicks a link on a map’s tooltip, select the map view.

5. Select how the people viewing your dashboard will run the action, such as Menu.

6. Enter the URL, starting with the http:// or https:// prefix, such as http://www.example.com. You can use field values as parameters in your URL. For example, if Country is a field used by a view in your dashboard, you can use <Country> as a parameter in your URL (see URL Actions on page 1808 for more details on protocols and parameters you can use).

7. Click OK.

At this point, if you use this action (for example, by clicking a link on a tooltip), a web browser window will open and load the page you specified in step 6.

8. To make the web page load within your dashboard, drag a Web Page object onto your dashboard.

9. In the Edit URL dialog, select Sheet Name. This tells the web page object to receive its URL from the view that you specified in step 4.
10. Click **OK**.

Now when you launch the action, a web page automatically loads within the dashboard rather than opening a separate browser window.
Using Field and Filter Values in Actions

When you add an action in Tableau you often want to use values from your data as parameters in the name of the action as well as the action itself. Using fields as variables in the action name makes the menu item that launches the action specific to the selected mark. More commonly, using field and filter values as parameters in the URL of a URL action allows you to send information about a specific data point or filter setting to the destination webpage.

Using Field and Filter Values in URLs

Tableau lets you add field, filter information, and parameter values as variables into URL actions so when you follow the link the values of those fields and filters are included. For example, when linking to an online mapping service, you can insert the address field into the URL so launching the link from a specific data point shows the address associated with that record on a map.

1. In the Add URL Action dialog box, begin typing the URL for the link.
2. Place the cursor where you want to insert a field, parameter, or filter value.
3. Click the arrow to the right of the text box and select the field, parameter, or filter you want to add to the URL. The field, parameter, or filter name is added to the URL between angle brackets. You can continue adding field and filter values parameters as many times as you need to create the URL.
**Note:** The list of available fields only includes non-aggregated fields. To use aggregated field values as a parameter in a link, you must first create a calculated field and then use the name of that field in the link. The calculated field must also be used in the view in order for the link to be available. A good way to use these fields is by placing them on the Level of Detail shelf.

When inserting parameter values, the Display As value is what is sent with the URL action. To send the actual value instead, you should add the character ~na after the parameter name. For example, you may have a parameter that includes IP addresses. The Actual Value strings for the parameter include values such as 10.1.1.195 while the Display As strings include more friendly values such as Computer A (10.1.1.195). Inserting the parameter into the URL like this: http://<IPAddress>/page.htm will send the display values so the final destination URL would be http://Computer A (10.1.1.195)/page.htm. Obviously, that is not likely to create a valid URL. To send the actual value, the parameter in the URL should look like this: http://<IPAddress~na>/page.htm.

Using Field and Filter Values in Action Names

In addition to using field, parameter, and filter values in URLs, you can use this information as variables in the action names. The name of the action displays on the context menu when an action is launched using the menu. Using field and filter variables in the name is useful in making the action specific to the selected mark. In a view showing real estate information, you could name a URL action that points at satellite images from an online mapping service, “Show satellite image of <Address>.” When you right-click (control-click on Mac) a specific mark, the <Address> tag is replaced with the location value associated with that mark.

1. In the Add Action dialog box, begin typing the name for the action.
2. Place the cursor where you want to insert the field or filter value.
3. Click the arrow to the right of the text box and select the field or filter you want to add as a
variable. The field or filter name is added between angle brackets.
Add Axes for Multiple Measures in Views

There are several different ways to compare multiple measures in a single view. You can:

- Create individual axes for each measure.
- Blend two measures to share an axis.
- Add dual axes where there are two independent axes layered in the same pane.

In any of these cases you can customize the marks for each axis to use multiple mark types and add different levels of detail. Views that have customized marks are called combination or combo charts.

For details on how to edit axes, see Edit Axes on page 1838.

In this article

- Add individual axes for measures below
- Blend axes for multiple measures into a single axis on the next page
- Compare two measures using dual axes on page 1829
- Synchronize axes to use the same scale on page 1830
- Synchronize axes with measures of different data types on page 1831
- Customize the marks for a measure on page 1832
- Create a combo chart (assign different mark types to measures) on page 1834

Add individual axes for measures

To add individual axes for each measure, drag measures to the Rows and Columns shelves.

- Adding a continuous field on the Rows shelf adds an additional axis to the rows of the table.
- Adding a continuous field on the Columns shelf adds an additional axis to the columns of the table.

The example below shows quarterly sales and profit. The Sales and Profit axes are individual rows in the table and have independent scales.
The Order Date field on the Columns shelf is a discrete date dimension. Because it is discrete, it creates headers rather than an axis. For details on how to edit axes, see Edit Axes on page 1838.

Blend axes for multiple measures into a single axis

Measures can share a single axis so that all the marks are shown in a single pane.

To blend multiple measures, drag one measure or axis and drop it onto an existing axis.
Instead of adding rows and columns to the view, when you blend measures there is a single row or column and all of the values for each measure is shown along one continuous axis. For example, the view below shows quarterly sales and profit on a shared axis.
Note: If you drag a measure on to the canvas and only see a single ruler indicator instead of the double ruler indicator shown below, Tableau creates dual axes instead of a blended axis. For more information about how to create dual axes, see Compare two measures using dual axes on the next page.

Blending measures uses the Measure Names and Measure Values fields, which are generated fields that contain all of the measure names in your data source and all of the measure values. The shared axis is created using the Measure Values field. The Measure Names field is added to Color on the Marks card so that a line is drawn for each measure. Finally, the Measure Names field is filtered to only include the measures you want to blend.

Note: Blending axes is most appropriate when comparing measures that have a similar scale and units. If the scales of the two measures are drastically different, the trends may be distorted.
Compare two measures using dual axes

You can compare multiple measures using dual axes, which are two independent axes that are layered on top of each other. Dual axes are useful for analyzing two measures with different scales.

To add a measure as a dual axis, drag the field to the right side of the view and drop it when you see a black dashed line appear.

You can also right-click (control-click on Mac) the measure on the Columns or Rows shelf and select **Dual Axis**.

The result is a dual axis view where the Profit axis corresponds to the red line and the Sales axis corresponds to the yellow line.
You can add up to four layered axes: two on the Columns shelf and two on the Rows shelf.

---

**Synchronize axes to use the same scale**

To align the two axes in a dual axes chart to use the same scale, right-click (control-click on Mac) the secondary axis, and select *Synchronize Axis*. This aligns the scale of the secondary axis to the scale of the primary axis.

In this example, the Sales axis is the secondary axis and the Profit axis is the primary axis.

**Note:** To synchronize axes, the data types for both measures must be the same. If the data types for your measures are different, see the section below.

You can synchronize dual axes for numeric data types that don't match. For example, you can synchronize an axis that uses an integer data type and an axis that uses decimal data type.

If you would like to change which axis is the primary, and which axis is the secondary, select the field on the Columns or Rows shelf that is the secondary, and drag it in front of the primary field on the shelf until you see an orange triangle appear.
In this example, you can select the SUM(Sales) field on the Rows shelf, and drag it in front of the SUM(Profit) field. The Sales axis is now the primary and the Profit axis is the secondary.

Synchronize axes with measures of different data types

The Synchronize Axis option ensures that you make a scaled and correct comparison in a dual axes chart. However, sometimes this option may not be available (grayed out). This is because the data type of one of the axes is different from the other.

**Note:** In a chart with dual axes, starting with version 2018.1, you can synchronize dual axes for numeric data types that don't match. For example, you can synchronize an axis that uses an integer data type and an axis that uses decimal data type.

To resolve this issue, you must change the data type of one of the axes. Follow the example below to change the data type for an axis.

1. Click the new Worksheet icon to open a new worksheet.
2. Drag Order Date to Columns, then click the drop-down arrow on the field on the shelf and select Month from the context menu.
3. Drag Sales to Rows, then drag Quantity to the right side of the view and drop it when you see a black dashed line appear to create a dual axis.
   
   If you right-click on the Quantity Axis, you can see that the Synchronize Axis option is grayed out.
4. In the Data pane, click on the drop-down arrow on the Quantity field, and select Change Data Type > Number (decimal) in the context menu. This changes the data type for this field.
5. Replace the **Quantity** measure on the view.

6. Right-click the secondary axis, and then select **Synchronize Axis**.

---

**Customize the marks for a measure**

1. Select the Marks card for the measure that you want to customize. There is a Marks card for each measure on the Rows and Columns shelves.
2. Select a new mark type for the measure.

Any changes to the mark type, shape, size, color, detail and other mark properties will be applied to the selected measure.

For example, in the view below the Sales Marks card is active. The Mark Type has been changed to Bar and when Product Type is placed on Color on the Sales Marks card, the encoding and level of detail is only applied to the Sales marks. The Budget Sales mark is not broken down by Product Type.
Tips

- Select a field in the Marks card to modify only its properties.
- Select the All Marks card to modify properties for all measures at once.
- To change the order of a field's marks in the view, right-click its axis, and then select Move marks to front.

Create a combo chart (assign different mark types to measures)

When working with multiple measures in a view, you can customize the mark type for each distinct measure. Because each measure can have customized marks, you can customize the level of detail, size, shape, and color encoding for each measure too.
For example, you could create a view that uses two measures in dual axes. One measure shows Profit with a line mark, and the other measure shows Sales with bar marks. You could also choose to display the measures as individual axes or blended axes.

**To create a combo chart like this example:**

1. Connect to the Sample - Superstore data source.
2. Drag the Order Date measure to the Columns shelf, and then click Order Date change Year to Month.
3. Drag the Sales measure to the Rows shelf.

4. Drag the Profit measure to the right side of the view and drop it when you see a black dashed line appear.
5. Click the Sales card. Change the Sales mark type to **Bar**.

6. To adjust the colors used for the Sales and Profit marks, click the **All** card, click **Color**, and then click **Edit Colors**.

   Click the Profit data item, and then click red in the palette.

   Click the Sales data item, and then click yellow in the palette.

   Click **Apply**, and then click **OK**.
Edit Axes

When you add a continuous field (a field with a green background) to the Columns or Rows shelf, it adds an axis to the view. An axis shows data points that lie within a range of values. For each axis, you can specify the range, scale, and tick mark properties.

Double-click an axis to open the Edit Axis dialog box and change the axis configuration and formatting. If you can't edit the axis, see Why can't I edit my axis? on page 1840
In this example, the Profit field (a continuous measure) on the Rows shelf creates a vertical axis, and the Order Date field (a continuous date dimension) on the Columns shelf creates a horizontal axis.

In this article

Edit an axis range below
Why can't I edit my axis? on the next page
Hide and reshow axes and headers on page 1842
Change an axis range on page 1844
Change the axis scale to reversed or logarithmic on page 1846
Change the appearance of an axis on page 1848
Format tick marks on page 1850
Examples: Use Different Axis Ranges (Uniform, Independent, Fixed) on page 1852

Edit an axis range

To edit an axis range, double-click the axis that you want to edit.
**Note:** In Tableau Desktop, you can right-click (control-click on Mac) the axis, and then select **Edit Axis**. In web authoring, you can click the arrow button on an axis, and then select **Edit Axis**.

When you select an axis, the marks associated with the axis are not selected so that you can edit and format the axis without modifying the marks.

**To select the marks associated with the axis,** right-click the axis and select **Select Marks**. You can also select each mark individually. For more information, see **Select Marks** on page 2612.

---

**Why can't I edit my axis?**

If you can't edit an axis, it's most likely a header rather than axis.

- Continuous fields (green background) on the Rows and Columns shelves create axes in the view. When you right-click an axis, you will see this menu:
Discrete fields (blue background) on the Rows and Columns shelves create headers, not axes. When you right-click a header, you will see this menu:
You can edit the formatting of headers by right-clicking a header, and then selecting **Format**.

For more details on formatting headers and field labels, see *Format at the Worksheet Level on page 2361* and *Format Fields and Field Labels on page 2386*. For details on continuous and discrete fields, see *Dimensions and Measures, Blue and Green on page 250*.

**Hide and reshow axes and headers**

If you have hidden an axis or a header in the view, you can show it again from its related field in the view.

**To hide an axis**

Right-click (control-click on Mac) the axis in the view, and then clear the check mark next to the **Show Header** option.

**To show a previously hidden axis**

Right-click (control-click on Mac) the continuous field in the view that is associated with axis you want to show, click its drop down menu, and then click **Show Header**.
For related details, see *Axes* on page 209 in *Parts of the View* on page 204.

**To hide a header**

Right-click (control-click on Mac) a header in the view, and then clear the check mark next to the *Show Header* option.

**To show previously hidden header**

Right-click (control-click on Mac) the discrete field in the view that is associated with headers that you want to show, click its drop down menu, and then click *Show Header*.
Change an axis range

You can limit the axis range to focus the view on certain data points.

For example, assume your view shows sales over four years. The automatic axis might show a range from 0 to $750,000. If sales never go below $470,000, you can adjust the axis range to start at $470,000 to keep the focus on where the data points actually lie.

**To change an axis range:**
1. Double-click the axis that you want to edit. You can also right-click (control-click on Mac) the axis, and then select **Edit Axis**.

2. In the **Edit Axis** dialog box, select one of the following options:
   - **Automatic**. Automatically bases the axis range on the data used in the view.
   - **Uniform axis range for all rows or columns**. Sets the axis range uniformly to the maximum data range for all panes in the view.
   - **Independent axis ranges for each row or column**. Makes the axis range independent. The axis range varies for each pane in the view, depending on the range of data in each pane.
   - **Fixed**. Specifies to start and/or end the axis at a specific value. You can fix both ends of the axis or only one end. When you fix both ends of the axis, the axis range is determined by the values you specify. When you fix only one end of the axis, you must set the other end of the axis as automatic, uniform, or independent.

3. You can also specify whether to include zero. When you clear the **Include zero** check box, the axis range adjusts to show only the range of values in the data.

4. Click **OK**.
Change the axis scale to reversed or logarithmic

Axis scale options include Reversed and Logarithmic.

When you select Logarithmic, you can then specify Positive or Symmetric. Symmetric displays data that contains positive, 0, or negative values on a log scale axis, and is best used to visualize large negative values on a logarithmic scale, as well as large positive values, or both. For related details, see About the symmetric log axis transform on page 1848.

To change the scale of an axis:

1. Double-click the axis that you want to edit. You can also right-click (control-click on Mac) the axis, and then select Edit Axis.
2. In the **Edit Axis** dialog box, on the General tab, select **Reversed** or **Logarithmic** for the axis scale.

The option to choose Symmetric or Positive is only available when the axis range extends outside of the closed interval \([-10, 10]\).

**Logarithmic**: Select **Positive** or **Symmetric** for the log scale. The **Symmetric** option displays data that contains positive, 0, and negative values on a log scale axis. By default, the tick marks are drawn at powers of ten, but you can specify any base that is greater than 1.

3. Click the **Tick Marks** tab and select one of the following options:
   - **Automatic** - the major tick marks are drawn at powers of 10.
   - **Fixed** - the major tick marks are drawn at a specified exponent. Type a number into the Powers of text box.
   - **None** - major tick marks are not shown.

4. When finished, click **OK**.
You can also reverse the axis by selecting **Reverse** in the Scale area on the General tab of the Edit Axis dialog box.

**About the symmetric log axis transform**

The symmetric log axis applies an area hyperbolic sine transform to the axis:

\[
\text{asinh}(x) = \ln(x + \sqrt{x^2 + 1})
\]

For large values \(|x| > 1\),

\[
\text{asinh}(x) \sim \ln(x), \quad x > 0
\]

\[
\text{asinh}(x) \sim -\ln(-x), \quad x < 0
\]

Because \(\text{asinh}\) is defined for all real values (negative numbers in particular) this transform allows you to visualize large negative values on a logarithmic scale, as well as large positive values, or both.

However, while a log axis can be used to visualize the scale of very small values, a symmetric log axis cannot be used to visualize the scale of very small values.

For small values \(|x| < 1\),

\[
\text{asinh}(x) \approx x
\]

Using a symmetric log axis scale for small values has a similar effect to using a linear scale (i.e., a normal quantitative axis).

---

**Change the appearance of an axis**

Every axis has a title that is automatically generated based on the fields in the view. You can specify a custom axis title and add a subtitle using the Edit Axis dialog box. You can also specify the scale of the axis, such as whether to use a logarithmic scale or whether to reverse the axis.

**To change the appearance of an axis:**

1. Right-click (control-click on Mac) on the axis that you want to edit and select **Edit Axis**.
2. In the Edit Axis dialog box, type a new title in the **Title** text box.
3. To add a custom subtitle, clear the **Automatic** check box.

4. To change the scale of the axis, select one of the following options:
   - **Reversed**: select this option to reverse the order of values on the axis.
   - **Logarithmic**: select this option to use a logarithmic scale on the axis.
5. Click **OK**.
3. For both **Major** and **Minor tick marks**, select from one of the following options:
   - **Automatic** - select this option to automatically show tick marks based on the data in the view.
   - **Fixed** - select this option to specify how often the tick mark should display and the starting value.
   - **None** - select this option to hide the tick marks completely.

4. Click **OK**.

   ![Edit Axis [Sales] dialog box](image)
Examples: Use Different Axis Ranges (Uniform, Independent, Fixed)

In this example you will build three views using the same data. However, each view will use a different axis range format. These views use the Sample-Superstore data source to display the aggregated total sales for three product categories over the course of four years.

Build a view with a uniform axis range below
Build a view using independent axis ranges on page 1854
Build a view using a fixed axis range on page 1857

Build a view with a uniform axis range

A uniform axis range means that the same range is applied to each row or column in your view. The range is automatically generated based on the underlying data values.

1. Place the Order Date dimension on the Columns shelf and the Category dimension on the Rows shelf.

2. Place the Sales measure on the Rows shelf.

The measure is automatically aggregated as a summation and an axis is added to the view. By default the view uses a uniform axis range. Notice that the axis range is the
same, from zero to 250,000 for each product category.
Build a view using independent axis ranges

When you use an independent axis range, each row or column will have its own axis range based on the underlying data values.

1. Right-click (control-click on Mac) the SUM(Sales) axis in the view and select **Edit Axis**.

2. In the Edit Axis dialog box, select **Independent axis ranges for each row or column**.
Click the X to close the dialog box with the current settings.

The axis range for each product category are now independent from each other. The Technology and Office Supplies categories still range from zero to 250,000, but the Furniture category ranges from zero to a little over 200,000.
Build a view using a fixed axis range

You can fix one or both ends of an axis in order to customize how much of the data you can see in the view.

The first example shows how to fix both ends of the axis. The second shows how to fix only one end of the axis, and then define the axis range for all panes in the view.

To fix both ends of the axis:

1. Right-click (control-click on Mac) the SUM(Sales) axis in the view and select Edit Axis.

2. In the Edit Axis dialog box, select Fixed, and then define Start and End values either by typing into the text boxes or by dragging the sliders toward each other. For this example, type 0 for Start and 200,000 for End.

3. Click the X to close the dialog box with the current settings.

The categories now display only up to $200,000 in sales. The axes are marked with a pin symbol 🌟, which indicates that you have limited the axis range and that some data
might not be displayed.

To fix one end of the axis:
1. Right-click (control-click on Mac) the \textit{SUM}(Sales) axis in the view and select \textit{Edit Axis}.

2. In the Edit Axis dialog box, select \textit{Fixed}, click the \textit{Fixed End} drop-down menu, and then select \textit{Independent}.

3. Click the \textit{X} to close the dialog box with the current settings.

Notice that the categories now have slightly different axis ranges. For example, Office Supplies has an axis range from 0 to 250,000 dollars, while Furniture only has a range from 0 to 200,000 dollars.
For more information about the difference between automatic, uniform, and independent axis ranges, see Edit Axes on page 1838.
Geographic Data Analysis in Tableau

When it comes to plotting your data on a map, Tableau has got you covered. This section provides all the documentation you need to get started with creating maps in Tableau.

Start with these topics

<table>
<thead>
<tr>
<th>Get Started Mapping with Tableau below</th>
<th>Map Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping Concepts</td>
<td>Build Maps</td>
</tr>
<tr>
<td>The Mapping Workspace in Tableau on page 1899</td>
<td>Customize Maps</td>
</tr>
</tbody>
</table>

Watch a Video: To see related concepts demonstrated in Tableau, watch these free training videos: Getting Started with Mapping (3 minutes) and Maps in Tableau (4 minutes). Use your tableau.com account to sign in.

Get Started Mapping with Tableau

This tutorial walks you through some of the most common tasks you might perform when creating maps in Tableau.

You'll learn how to connect to and join geographic data; format that data in Tableau; create location hierarchies; build and present a basic map view; and apply key mapping features along the way.

If you’re new to building maps in Tableau, this a great place to start.

In this article

- **Step 1: Connect to your geographic data** on the next page
- **Step 2: Join your data** on page 1865
- **Step 3: Format your geographic data in Tableau** on page 1867
Step 1: Connect to your geographic data

Geographic data comes in many shapes and formats. When you open Tableau Desktop, the start page shows you the connectors available in the left Connect pane. These are how you will connect to your data.

You can work with geographic data by connecting to spatial files, or you can connect to location data stored in spreadsheets, text files, or on a server.

Spatial files, such as a shapefile or geoJSON file, contain actual geometries (points, lines, or polygons), whereas text files or spreadsheets contain point locations in latitude and longitude coordinates, or named locations that, when brought into Tableau, connect to the Tableau geocoding (stored geometries that your data references).

For a complete list of connections Tableau supports, see the list of Data Connections on the Tableau website.
For this tutorial, you are going to connect to an Excel file that comes with Tableau Desktop. It contains location names that Tableau can geocode. When you build a map view, the location names reference the geometries stored in the Tableau Map Service based on the geographic role you assign to the field. You’ll learn more about geographic roles later in this tutorial.

1. Open Tableau Desktop.
2. In the Connect pane, click Excel.
3. Navigate to Documents > My Tableau Repository > Data Sources, and then open the Sample - Superstore.xls file.

Once you connect to the data source, your screen will look like this:
This is called the Data Source page, and it is where you can prepare your location data for use in Tableau.

Some of the tasks you can perform on the Data Source page include the following, but you don't have to do all these things to create a map view:

- Adding additional connections and joining your data
- Adding multiple sheets to your data source
- Assigning or changing geographic roles to your fields
- Changing the data type of your columns (from numbers to strings, for example)
- Renaming columns
- Splitting columns, such as splitting a full address into multiple columns for street, city, state, and postal code

For more information about the Data Source page and some of the tasks you can perform while on it, see the topics in the Set Up Data Sources on page 646 section.

Step 2: Join your data

Your data is often held in multiple data sources or sheets. As long as those data sources or sheets have a column in common, you can join them in Tableau. Joining is a method for
combining the related data on those common fields. The result of combining data using a join is a virtual table that is typically extended horizontally by adding columns of data.

Joining is often necessary with geographic data, particularly spatial data. For example, you can join a KML file that contains custom geographies for school districts in Oregon, U.S. with an Excel spreadsheet that contains demographic information about those school districts.

For this example, you will join two sheets in the Sample-Superstore data source.

1. On the left side of the Data Source page, under Sheets, double-click **Orders**.
2. Under Sheets, double-click **People**.

   Tableau creates an inner-join between the two spreadsheets, using the Region column from both spreadsheets as the joining field. Now there is a sales person assigned to every location in your data source, as well as to regions.

To edit this join, click the join icon (the two circles). You can edit the join in the Join dialog box that opens. For more information about joining data in Tableau, see **Join Your**
Step 3: Format your geographic data in Tableau

After you set up your data source, you might need to prepare your geographic data for use in Tableau. Not all of these procedures will always be necessary to create a map view, but it's important information to know when it comes to preparing geographic data for use in Tableau.

Depending on the type of map you want to create, you must assign certain data types, data roles, and geographic roles to your fields (or columns).

For example, in most cases, your latitude and longitude fields should have a data type of number (decimal), a data role of measure, and be assigned the Latitude and Longitude geographic roles. All other geographic fields should have a data type of string, a data role of dimension, and be assigned the appropriate geographic roles.

Note: If you are connecting to a spatial file, a Geometry field is created. It should have a data role of measure.

This step demonstrates how to format your geographic data to meet this criteria.

Change the data type of a column

When you first connect to geographic data, Tableau assigns data types to all of your columns. These data types include Number (decimal), Number (whole), Date and Time, Date, String, and Boolean. Sometimes Tableau does not get these data types right, and you must edit them.
For example, Tableau might assign a Postal Code column a data type of Number (whole). To create map views, your Postal Code data must have a data type of String.

To change the data type of a column:

1. On the Data Source page, click the data type icon (the globe) for Postal Code and select String.

<table>
<thead>
<tr>
<th>State</th>
<th>Postal Code</th>
<th>Region</th>
<th>Product ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kentucky</td>
<td>42420</td>
<td>South</td>
<td>FUR-BK-10001798</td>
</tr>
<tr>
<td>Kentucky</td>
<td>42420</td>
<td>South</td>
<td>FUR-CH-10000454</td>
</tr>
<tr>
<td>California</td>
<td>90036</td>
<td>West</td>
<td>OFF-LA-10000240</td>
</tr>
<tr>
<td>Florida</td>
<td>33311</td>
<td>South</td>
<td>FUR-TA-10000577</td>
</tr>
<tr>
<td>Florida</td>
<td>33311</td>
<td>South</td>
<td>OFF-ST-10000760</td>
</tr>
<tr>
<td>California</td>
<td>90032</td>
<td>West</td>
<td>FUR-FU-10001487</td>
</tr>
<tr>
<td>California</td>
<td>90032</td>
<td>West</td>
<td>OFF-AR-10002833</td>
</tr>
<tr>
<td>California</td>
<td>90032</td>
<td>West</td>
<td>TEC-PH-10002275</td>
</tr>
<tr>
<td>California</td>
<td>90032</td>
<td>West</td>
<td>OFF-BI-10003910</td>
</tr>
<tr>
<td>California</td>
<td>90032</td>
<td>West</td>
<td>FUR-AP-10002892</td>
</tr>
</tbody>
</table>

For more information about data types, see Data Types on page 263.

Assign geographic roles to your geographic data

In Tableau, a geographic role associates each value in a field with a latitude and longitude value. When you assign the correct geographic role to a field, Tableau assigns latitude and longitude values to each location in that field by finding a match that is already built in to the installed geocoding database. This is how Tableau knows where to plot your locations on the map.

When you assign a geographic role to a field, such as State, Tableau creates a Latitude (generated) field and a Longitude (generated) field.
Geographic roles are sometimes automatically assigned to your data, such as in this example. You can tell a geographic role has been assigned to your data because the column includes a globe icon.

If a geographic role is not automatically assigned, you can manually assign one to your field. You don’t need to do so for this example, but it's important to know how so you can do it for your own data.

**To assign or edit a geographic role:**

1. On the Data Source page, click the globe icon.

2. Select **Geographic Role**, and then select a role that best matches your data.

   For example, in this case, the Country column does not have a geographic role assigned to it, so the Country/Region geographic role is assigned.

<table>
<thead>
<tr>
<th>Abc</th>
<th>Orders</th>
<th>Orders</th>
<th>Orders</th>
<th>Orders</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Country</td>
<td>City</td>
<td>State</td>
<td>Postal Code</td>
</tr>
<tr>
<td>United States</td>
<td>Henderson</td>
<td>Kentucky</td>
<td>42420</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>Henderson</td>
<td>Kentucky</td>
<td>42420</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>Los Angeles</td>
<td>California</td>
<td>90036</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>Fort Lauderdale</td>
<td>Florida</td>
<td>33311</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>Fort Lauderdale</td>
<td>Florida</td>
<td>33311</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>Los Angeles</td>
<td>California</td>
<td>90032</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>Los Angeles</td>
<td>California</td>
<td>90032</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>Los Angeles</td>
<td>California</td>
<td>90032</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>Los Angeles</td>
<td>California</td>
<td>90032</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>Los Angeles</td>
<td>California</td>
<td>90032</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If you have difficulties assigning geographic roles to your data, or have data that is not built in to the Tableau map server, there are a few things you can do to get that data into Tableau. See **Assign Geographic Roles** on page 1941.
Change from dimensions to measures

When you connect to geographic data, Tableau also assigns data roles to all of your columns. A column can be a *dimension or measure*. In most cases, your latitude and longitude columns should be measures. For special cases, such as if you want to plot every location in your data source on a map without the ability to drill up or down a level of detail (such as from City to State), they can be dimensions. A great example of this is a point distribution map.

The rest of your geographic data should be dimensions.

You don’t need to change the data role of a column for this example, but it’s important to know how so you can do it for your own data. Feel free to practice here. You can always undo any changes you make.

**To change the data role of a column:**

1. On the Data Source page, click **Sheet 1**.

Your workspace updates to look like this:

![Worksheet](image)

This is called a worksheet, and it is where you will build your map. On the left-side of the screen is the **Data pane**. All of the columns in your data source are listed as fields in this pane. For example, Country and State. These fields contain all the raw data in your columns. Note that Tableau has generated a Latitude and Longitude field (*Latitude (generated)* and *Longitude (generated)*). This is because you assigned geographic roles to your data.
The fields in the data pane are divided into measures and dimensions. The fields placed in the Dimensions section of the Data pane are often categorical data, such as Date and Customer ID, while the fields placed in the Measures section of the Data pane are often quantitative data, such as Sales and Quantity.

2. In the **Data** pane, under Dimensions, select a field, such as Row ID, and drag it down to the Measures section.
The field is added to the Measures section and changes from blue to green. You just converted a Dimension to a Measure. To convert a field from a measure to a dimension, drag the field from the Measures section up to the Dimensions section.

For more information, see Dimensions and Measures, Blue and Green on page 250.
Step 4: Create a geographic hierarchy

Now that you are in the worksheet space, you can create geographic hierarchies. This is not required to create a map view, but creating a geographic hierarchy will allow you to quickly drill into the levels of geographic detail your data contains, in the order you specify.

To create a geographic hierarchy:

1. In the Data pane, right-click the geographic field, **Country**, and then select **Hierarchy > Create Hierarchy**.

2. In the Create Hierarchy dialog box that opens, give the hierarchy a name, such as **Mapping Items**, and then click **OK**.

   At the bottom of the Dimensions section, the Mapping Items hierarchy is created with the Country field.

3. In the Data pane, drag the State field to the hierarchy and place it below the Country field.

4. Repeat step 3 for the City and Postal Code fields.

When you are finished, your hierarchy should be in the following order:

- Country
- State
Step 5: Build a basic map

Now that you have connected to and joined your data, formatted your data, and built a geographic hierarchy, you are now ready to start building your map. You will start by building a basic map view.

1. In the Data pane, double-click **Country**.

   The Country field is added to Detail on the Marks card, and Latitude (generated) and Longitude (generated) are added to the Columns and Rows shelves. A map view with one data point is created. Since a geographic role is assigned to Country, Tableau creates a map view. If you double-click any other field, such as a dimension or measure, Tableau adds that field to the Rows or Columns shelf, or the Marks card, depending on what you already have in the view. Geographic fields are always placed on Detail on the Marks card, however.

   Since this data source only contains one country, (United States), that is the only data point shown. You will need to add more levels of detail to see additional data points. Since you created a geographic hierarchy, this is easy.

2. On the Marks card, click the + icon on the **Country** field.
The State field is added to Detail on the Marks card and the map updates to include a data point for every state in the data source.

If you did not create a hierarchy, the + icon on the Country field will not be available. In this case, to add State as another level of detail, manually drag State from the Data pane to Detail on the Marks card.

Congratulations! You now have a basic map view that you can customize and build upon in the next steps.

Step 6: Change from points to polygons

The default map type in Tableau is often a point map. When you have geographic roles assigned to your geographic data, however, it's easy to change those data points to polygons.

**Note:** Filled maps are not available for cities or airports.
1. On the Marks card, click the Mark Type drop-down and select **Filled Map**.

The map updates to a polygon map.

---

**Step 7: Add visual detail**

You can add measures and dimensions to the Marks card to add visual detail to your view. In this example, you will add color and labels to the view.
Add color

- From Measures, drag Sales to Color on the Marks card.

Each state is colored by sum of sales. Since Sales is a measure, a qualitative color palette is used. If you place a dimension on color, then a categorical color palette is used.

Add labels

1. From Measures, drag Sales to Label on the Marks card.

   Each state is labeled with sum of sales. The numbers need a little bit of formatting, however.

2. In the Data pane, right-click Sales and select Default Properties > Number Format.

3. In the Default Number Format dialog box that opens, select Number (Custom), and then do the following:

   - For Decimal Places, enter 0.
   - For Units, select Thousands (K).
   - Click OK.
The labels and the color legend update with the specified format.

Step 8: Customize your background map

The background map is everything behind your marks (borders, oceans, location names, etc.) You can customize the style of this background map, as well as add map layers and data layers. In addition to customizing the background maps, you can also connect to your own WMS server or Mapbox map. For more information, see Use Web Map Service (WMS) Servers on page 2078 and Use Mapbox Maps on page 2073.

To customize your background map:

1. Select Map > Map Layers.

   The Map Layers pane appears on the left side of the workspace. This is where all background map customization happens.

2. In the Map Layers pane, click the Style drop-down and select Normal.

   The background map updates to look like this:
3. In the Map Layers pane, under Map Layers, select **Coastlines**, and then clear **Country/Region Borders**, **Country/Region Names**, **State/Province Borders**, and **State/Province Names**.
4. At the top of the Map Layers pane, click the X to return to the Data pane.
   The background map is now simplified to draw attention to your data.

Step 9: Create custom territories

As you build your map view, you might want to group existing locations together to create your own territories or regions, such as sales territories for your organization.

1. In the Data pane, right-click State and select Create > Group.
2. In the Create Group dialog box that opens, select California, Oregon, and Washington, and then click Group. Each group you create represents a territory.
   Note: To multi-select, hold down Ctrl (Command on Mac) as your select states.
3. Right-click the new group you just created and select Rename.
4. Rename the group, West Coast.
5. For the next territory, select Alabama, Florida, Georgia, Louisiana, Mississippi, South Carolina, and Texas, and then click Group.
6. Rename this group, South.

7. For the third territory, select Connecticut, Delaware, District of Columbia, Main, Maryland, Massachusetts, New Hampshire, New Jersey, New York, Pennsylvania, Rhode Island, Vermont, and finally, West Virginia, and then click Group.

8. Rename this group, East Coast.

9. Select Include Other to group the remaining states.

10. Rename the Other group, Central.

11. Click OK.

A State (group) field appears in the Data pane beneath your other mapping items.

12. From the Data pane, drag State (group) to Color on the Marks card.

The view updates to look like this:
Notice that each group has a different color.

13. On the Marks card, click the **Color** icon and select **Edit Colors**.

14. In the Edit Colors dialog box that appears, select **Assign Palette**, and then click **OK**.

The marks update with new colors.
15. From Measures, drag **Sales** to **Tooltip** on the Marks card.

When you hover over a state, a tooltip appears with the sales for that state, among other information. You'll learn how to edit this tooltip later.

16. On the Marks card, click the minus (-) icon on the **Country** field to remove State from the level of detail.

If you did not create a hierarchy, you can drag **State** from the view to remove it. You can remove any field by dragging it from the view.

The states no longer appear on the map. Notice how the sum of sales has updated for the labels and in the tooltip? This is because custom territories aggregate at the level of the group, rather than separately for each location within the group. So the sum of sales your are seeing in the West Coast group, for example, are the total sales for California, Oregon, and Washington combined.
Step 10: Create a dual axis map

So far you have created two map views: one that shows the sales per state, and one that shows the sales per region. Could you layer these maps on top of one another? Yes! In Tableau, you can create a map with two layers of marks. This is called a dual axis map in Tableau, and is often used to layer points over polygons. In this example, you will layer two polygons maps.

To create a dual axis map:

1. From the Data pane, drag **Longitude (generated)** to the **Columns** shelf and place it to the right of the first Longitude field.

   The view updates with two identical maps.
There are now three tabs on the Marks card: one for each map view, and one for both views (All). You can use these to control the visual detail of the map views. The top Longitude tab corresponds to the map on the left of the view, and the bottom Longitude tab corresponds to the map on the right of the view.

2. On the Marks card, select the top Longitude (generated) tab.
3. From Measures, drag **Sales** to **Color** on the top Longitude (generated) Marks card. The map on the left updates.

4. On the top Longitude (generated) Marks card, click the + icon on the **Country** field to drill back down to the State level of detail.

5. On the Marks card, click **Color**, and then select **Edit Colors**.

6. In the Edit Colors dialog box that opens, click the Palette drop-down, select **Gray**, and then click **OK**.

At this point, your maps look like this:

7. On the Columns shelf, right-click the **Longitude (generated)** field on the right and select **Dual Axis**.
8. On the Marks card, select the bottom **Longitude (generated)** tab.

9. On the bottom **Longitude (generated)** Marks card, drag both **SUM(Sales)** fields from the view to remove them.

   The labels for each map no longer overlap.

10. On the bottom **Longitude (generated)** Marks card, click **Color**, and then, for **Opacity**, enter **50%**.

    This is a crucial step if you want to be able to see the map on the bottom layer.

    The map view updates to look like this:
You can now see how each state performed within each group.

11. On the bottom **Longitude (generated) Marks card, click Tooltip**.
    
    An Edit Tooltip dialog box opens.

12. Copy the following text and paste it into the Edit Tooltip dialog box, and then click **OK**:
    
    Total <State (group)> region sales: <SUM(Sales)>
    
    The tooltip looks similar to this:

    ![Dual axis map](image)
    
    Congrats! You've created a dual axis map! There's only one thing left to do.
For more information about dual axis maps, see Dual Axis Mapping - Many Ways on the Tableau Community.

Step 11: Customize how others can interact with your map

Now that you have created your map view, you can customize how people will interact with it. For example, you might now want anyone to be able to zoom in or out of your map, or pan. Or perhaps you want to display a map scale? You can customize these two options and more in the Map Options dialog box.

To customize how others can interact with your map:

1. Select **Map > Map Options**.
2. In the Map Options dialog box that appears, do the following:
   - Select **Show Map Scale**.
   - Clear **Show Map Search**.
   - Clear **Show View Toolbar**.

A scale appears in the bottom-right corner of the map, and the map search icon and the toolbar in the top left corner of the map disappear. You can still pan and zoom using keyboard shortcuts. For more information, see Pan and Zoom on page 2615.

And that’s a wrap! Your map view is now ready to be presented or combined with another view in a dashboard. For more information about dashboards, see Create a Dashboard on page 2248.

Want to check your work? Download the example workbook from Tableau Public.

What's Next?

Learn some mapping visualization basics.

Learn about the types of maps you can build in Tableau and how to create them.

Looking for more advanced content? See the mapping series on the Tableau Community.
Mapping Concepts in Tableau

If you want to analyze your data geographically, you can plot your data on a map in Tableau. This topic explains why and when you should put your data on a map visualization. It also describes some of the types of maps you can create in Tableau, with links to topics that demonstrate how to create each one.

If you're new to maps in Tableau, this is a great place to start learning.

In this article

- **Why put your data on a map?** on the next page
- **When should you use a map to represent your data?** on the next page
- **What types of maps can you build in Tableau?** on page 1893
- **Resources to help you get started** on page 1899
Why put your data on a map?

There are many reasons to put your data on a map. Perhaps you have some location data in your data source? Or maybe you think a map could really make your data pop? Both of those are good enough reasons to create a map visualization, but it’s important to keep in mind that maps, like any other type of visualization, serve a particular purpose: they answer spatial questions.

You make a map in Tableau because you have a spatial question, and you need to use a map to understand the trends or patterns in your data.

But what is a spatial question? Some examples might be:

- Which state has the most farmers markets?
- Where are the regions in the U.S. with the high obesity rates?
- Which metro station is the busiest for each metro line in my city?
- Where did the storms move over time?
- Where are people checking out and returning bikes from their local bike share program?

All of these are spatial questions. However, is a map the best way to answer them?

When should you use a map to represent your data?

If you have a spatial question, a map view might be a great way to answer it. However, that might not always be the case.

Take for example, the first question from the list above: Which state has the most farmers markets?

If you had a data source with a list of farmers markets per state, you might create a map view like the one below. Can you easily tell the difference between New York and California? Which one has more farmers markets?
What if you create a bar chart instead? Now is it easy to spot the state with the most farmers markets?

The above example is one of many where a different type of visualization would be better to answer a spatial question than a map.

So when do you know if you should use a map view?
One rule of thumb is to ask yourself whether or not you could answer your question faster, or easier with another visualization. If the answer is yes, then perhaps a map view is not the best visualization for the data you’re using. If the answer is no, then take the following into account:

Maps that answer questions well have both appropriate data representation, and attractive data representation. In other words: the data is not misleading, and the map is appealing.

If your map is beautiful, but the data is misleading, or not very insightful, you run the risk of people misinterpreting your data. That’s why it’s important to create maps that represent your data accurately, as well as attractively.

What types of maps can you build in Tableau?

With Tableau, you can create the following common map types:

- **Proportional symbol maps** below
- **Choropleth maps** (filled maps) on the next page
- **Point distribution maps** on page 1895
- **Flow maps** (path maps) on page 1896
- **Spider maps** (origin-destination maps) on page 1897

Proportional symbol maps

Proportional symbol maps are great for showing quantitative data for individual locations. For example, you can plot earthquakes around the world and size them by magnitude.

For more information about proportional symbol maps, and to learn how to create them in Tableau, see Create Maps that Show Quantitative Values in Tableau on page 1973.
Choropleth maps (filled maps)

Also known as filled maps in Tableau, Choropleth maps are great for showing ratio data. For example, if you want to see obesity rates for every county across the United States, you might consider creating a choropleth map to see if you can spot any spatial trends.

For more information about Choropleth maps, and to learn how to create them in Tableau, see Create Maps that Show Ratio or Aggregated Data in Tableau on page 1986.
Point distribution maps

Point distribution maps can be used when you want to show approximate locations and are looking for visual clusters of data. For example, if you want to see where all the hailstorms were in the U.S. last year, you can create a point distribution map to see if you can spot any clusters.

For more information about point distribution maps, and to learn how to create them in Tableau, see Create Maps that Highlight Visual Clusters of Data in Tableau on page 1980.
Flow maps (path maps)

You can use flow maps to connect paths across a map and to see where something went over time. For example, you can track the paths of major storms across the world over a period of time.

For more information about flow maps, and to learn how to create them in Tableau, see Create Maps that Show a Path Over Time in Tableau on page 1994.
Spider maps (origin-destination maps)

You can use a spider map to show how an origin location and one or more destination locations interact. For example, you can connect paths between metro stations to plot them on a map, or you can track bike share rides from an origin to one or more destinations.

For more information about spider maps, and to learn how to create them in Tableau, see Create Maps that Show Paths Between Origins and Destinations in Tableau on page 2000.
Resources to help you get started

Before you get started with building map views in Tableau, review some of the following resources.

Get your geographic data into Tableau

- **Create Tableau Maps from Spatial Files** on page 1959: If you have Esri Shapefiles, Mapinfo tables, or KML files, start here.
- **Geocode Locations Tableau Does Not Recognize and Plot Them on a Map** on page 1925: If you have data that Tableau does not recognize, start here.
- **Blend Geographic Data** on page 1934: If you have geographic data you want to blend with another data source, start here.

Format your geographic fields

- **Assign Geographic Roles** on page 1941: Once you bring your geographic data into Tableau, you'll need to format that data for use in Tableau.

**Watch a Video**: To see related concepts demonstrated in Tableau, watch these free training videos: *Getting Started with Mapping* (3 minutes) and *Maps in Tableau* (4 minutes). Use your tableau.com account to sign in.

See also

- **Build-It-Yourself: Build a map view**
- **Customize How Your Map Looks** on page 2047
- **Use Mapbox Maps** on page 2073
- **Use Web Map Service (WMS) Servers** on page 2078

The Mapping Workspace in Tableau

The mapping workspace in Tableau consists of the following elements.
The Data Pane

The Data pane organizes your location data into dimensions and measures. Typically, your location fields (country, state, and city names, for example) should be dimensions, and your latitude and longitude fields should be measures. However, there are some scenarios where latitude and longitude fields can be dimensions.

In the Data pane, a globe icon next to a location field means that a geographic role is assigned to that field. For more information, see Assign Geographic Roles on page 1941.
A: Globe icon - indicates that a field is a geographic field and can be automatically mapped by Tableau.

B: Location fields - These categorical fields, often location names, can be found under Dimensions. They are often recognized by Tableau.

C: Latitude and Longitude generated fields - When you have geographic fields in your data source, Tableau often generates Latitude and Longitude fields. You can use these to build your map, or you can include your own latitude and longitude fields.
2

The Marks Card

The Marks card is where you control the granularity and visual detail of your map view. In the Marks card, you can drag locations to Detail to add more granularity, drag fields to Color, Size, or Label to add visual detail, or change the map type from a point map to a polygon map.

![Marks Card Diagram]

A: Add location fields to Detail to create a map and add granularity to the view.

B: Select the Automatic mark type for a point map and the Map mark type for a polygon (filled) map.

C: Add fields to Color, Size, and Label to change the visual detail of the marks (data points) in the view.

3

The View Toolbar

The view toolbar holds many of the tools you might need to explore your map data.

Note: Many of the tools on this toolbar have keyboard shortcuts. For more information, see Keyboard Shortcuts on page 2761.
A: Use the map search icon to search for locations in your map.

B: Use the plus and minus icons to zoom in and out of the map.

C: Use the pin icon to fix the map at a specific zoom level.

D: Use the zoom area icon to zoom in to a specific area in the map.

E: Use the pan icon to pan around the map.

F: Use the selection tools to select data points within a rectangle, circle, or drawn area. You can also measure distances with the circle selection tool. For more information, see Measure Distances Between Data Points and Locations in a Map on page 2089.

The Map Menu

The Map menu holds several maps-specific options:

- Select Map > Background Maps to add a map service (such as Mapbox or a WMS server) or to switch between the background maps you have available in your workbook. For more information, see Import your own background map on page 2049.

- Select Map > Background Images to add a static image instead of a background map to your view.

- Select Map > Geocoding to import locations that Tableau does not recognize into your workbook. For more information, see Geocode Locations Tableau Does Not Recognize and Plot Them on a Map on page 1925.
- 1904 -

- Select **Map > Edit Locations** to edit your location data to match the Tableau map data. For more information, see **Edit Locations** on page 1946.

- Select **Map > Map Layers** to customize your background map style, add or remove map layers, or add U.S. data layers. For more information see **Customize How Your Map Looks** on page 2047.

- Select **Map > Map Options** to control if people can pan and zoom, search for locations, or use the view toolbar. For more information, see **Customize How People Interact with your Map** on page 2069.

5  

**The Columns and Rows Shelves**

The Columns and Rows shelves are where you put your latitude and longitude fields. Longitude is placed on the Columns shelf, and Latitude is placed on the Rows shelf, regardless of whether they are your latitude and longitude fields, or the generated fields Tableau creates.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Longitude (generated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows</td>
<td>Latitude (generated)</td>
</tr>
</tbody>
</table>

6  

**The Map Visualization**

This is your map. As you makes changes to your map, the viz updates. This visualization is dynamic and can be interacted with. Hover over the marks to learn more information about each location. You can edit the information that displays in the tooltip that appears using the Tooltip button on the Marks card. For more information, see **Add tooltips to marks** on page 1133 and **Create Views in Tooltips (Viz in Tooltip)** on page 1773.

In addition to the options available in the mapping workspace, Tableau provides many other options for analyzing your data. You can create custom territories, filter data from your view, set up parameters, create sets, and much more.
Location Data that Tableau Supports for Building Map Views

Tableau supports worldwide airport codes, cities, countries, regions, territories, states, provinces, and some postcodes and second-level administrative districts (county-equivalents). U.S. area codes, Core Based Statistical Areas (CBSA), Metropolitan Statistical Areas (MSA), Congressional districts, and Zip codes are also supported by Tableau. Additionally, any latitude and longitude coordinates are supported, as long as they are in decimal degrees.

See About Tableau Maps on the Tableau website for a complete list of location data by country that Tableau supports.

What to do if Tableau doesn't recognize your location data

- If Tableau recognizes your location data and automatically assigns geographic roles to your fields (you'll know this has happened because there will be a globe icon next to them in the Data pane), then you're ready to build a map view. Just double-click one of those geographic fields and you've got a map.

- If Tableau does not immediately recognize your location data, and you can't build a map view, you'll need to assign geographic roles to your fields. See the Assign a geographic role to a field on page 1942 section to learn how.

  Note: This procedure only works if your location data is supported by Tableau. If your data isn't supported by Tableau, there are a couple of things you can try in the meantime to map that data in Tableau:

  - Edit location names in your data source to match Tableau location names
  - Custom geocode your data
  - Blend your geographic data
  - Use spatial data to build a map view
Note: Connecting to spatial data is only supported on Tableau Desktop version 10.2 and later.

See Also

About Tableau Maps Data
About Tableau Maps Partners

Spatial File

This topic describes how to connect Tableau Desktop to Shapefiles, MapInfo tables, KML (Keyhole Markup Language) files, and GeoJSON files.

Note: Connecting to spatial data is supported in Tableau Desktop version 10.2 and later.

If you have Tableau Desktop version 10.1 or earlier, and would like to use shapefile data to create map views, see Create Tableau Maps from Shapefiles in the Tableau Desktop 10.1 Help.

In this article

Before you connect below
Make the connection and set up the data source on the next page
Work with the Geometry column on the next page
About .ttde and .hhyper files on page 1908

Before you connect

Before you can connect to spatial files, make sure to include all of the following files in the same directory:

- **For Esri shapefiles**: The folder must contain a .shp, .shx, and .dbf file.
- **For MapInfo tables**: The folder must contain a .TAB, .DAT, .MAP and .ID or .MID/.MIF file.
- **For KML files**: The folder must contain the .kml file. (No other files are required.)
For GeoJSON files: The folder must contain the .geojson file. (No other files are required.)

Note: You can only connect to point geometries, linear geometries, and polygons in current versions of Tableau. You cannot connect to mixed geometry types.

Also note that if your data does not display diacritics (accent marks on characters) properly, check to make sure the file is UTF-8 encoded.

Make the connection and set up the data source

Start Tableau and under Connect, select Spatial file. Then do the following:

1. Navigate to the folder that contains your spatial data and select the spatial file you want to connect to.
2. Select Open.

Spatial file data source example

Here is an example of a spatial file data source using Tableau Desktop on a Mac computer:

![Spatial file data source example](image)

Work with the Geometry column

There are many tasks that you can optionally perform on your data before you start your analysis, such as hiding or renaming fields. Note, however, the following restrictions apply
when working with the Geometry column:

- You can't hide the Geometry column.
- You can't split the Geometry column.
- On the data source page, you can't create a calculated field using the Geometry column.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

See also

Create Tableau Maps from Spatial Files on page 1959

Connect to Spatial Data in Microsoft SQL Server

When you connect to Microsoft SQL Server, you can use spatial columns stored there to build map visualizations in Tableau.

This article uses examples to demonstrate how to:

- **Connect to spatial columns in Microsoft SQL Server** on the next page
- **Use Custom SQL and RAWSQL to perform advanced spatial analysis** on page 1910
- Troubleshoot common errors and performance issues
- **Build a map from Microsoft SQL Server spatial data**
Important Note

For Microsoft SQL Server, only **Geography-type** spatial data is supported.

The following **SRIDs** are also supported:

- NAD83 (EPSG:4269)
- ETRS89 (EPSG:4258)
- WGS84 (EPSG:4326)

Connect to spatial columns in Microsoft SQL Server

1. Open Tableau and connect to Microsoft SQL Server. For more information on how to connect to Microsoft SQL Server, see the **Microsoft SQL Server** on page 487 connector example.

2. On the Data Source page, in the left pane under Table, drag a table that contains spatial data onto the canvas.
You are now ready to build a map in Tableau using Microsoft SQL Server spatial data. For more information on how to create maps from spatial files, see Build a map view from spatial data on page 1961.

Use Custom SQL and RAWSQL to perform advanced spatial analysis

Connect to a Custom SQL query

1. Open Tableau and connect to Microsoft SQL Server. For more information on how to connect to Microsoft SQL Server, see the Microsoft SQL Server on page 487 connector example.

2. On the Data Source page, in the left pane, drag New Custom SQL onto the canvas.

3. Type or paste your query into the Edit Custom SQL dialog box that appears. For examples of Custom SQL you can use with spatial data, see Custom SQL query examples below.

4. When finished, click OK.

For more information, see Connect to a Custom SQL Query on page 723

Custom SQL query examples

The following queries are examples. Note that Custom SQL queries will vary from data source to data source.

- If your data has two sets of points, such as schools and parks, and you want to show where they intersect:

```
SELECT S.[common nam] as SchoolName, P.[common nam] as ParkName, S.geom.STBuffer(<Parameters.Radius>) as school_geom, P.geom as park_geom FROM TestSpatial.dbo.seattleelementaryschools S LEFT JOIN TestSpatial.dbo.allseattleparks P on S.geom.STBuffer(<Parameters.Radius>).STIntersects(P.geom) = 1
```

This example query will produce a row for each case where a park is within Radius distance (meters) from a school. The result of the query looks like this in the data grid:
Note: This example references a parameter named "Radius". It measures distance (meters) and has the following settings:

- Data type: Float
- Current Value: 600
- Allowable values: Range
- Minimum: 100
- Maximum: 2,000
- Step size: 50

For more information, see Use parameters in a custom SQL query on page 730.

- If you only want data around a certain point

```sql
SELECT * FROM [Test.spatial_<username>].[dbo].[SDOT_Collisions] C WHERE C.geom.STIntersects (geography::STGeomFromText('POINT(-122.344706 47.650388)', 4326).STBuffer(1000))=1
```

This example query limits data to 1000 meters around the POINT. The radius can be a parameter, like the parameter used in the first example.
Or, rather than a point and radius, a rectangle can be used. For example:

```
SELECT * FROM [Test_spatial_<username>].[dbo].[SDOT_Collisions] C
WHERE C.geom.STIntersects(geography::STGeomFromText
('POLYGON ((-122.3625 47.6642,-122.3625 47.6493,-
122.3427 47.6493,-122.3422 47.6642,-122.3625 47.6642)),
4326))=1
```

**Note:** the winding order of the polygon points (clockwise vs. counter-clockwise) determines which set of points you get. You can reverse the winding order with the SQL Server function `ReorientObject()`.

- If you get an error about mixed data types

```
SELECT *, F.geom.STGeometryType() as geomtype FROM [Test_spatial_<username>].[dbo].[us_historic_fire_perimeters_dd83] F
WHERE F.geom.STGeometryType() = 'MultiPolygon' OR
 F.geom.STGeometryType() = 'Polygon'
```

This example query filters out the geometry-type spatial data, since Tableau can not render geometry-type spatial data for Microsoft SQL Server connections. You can select other spatial types as well.

- If your data renders too slowly in Tableau

```
SELECT [Id], [OBJECTID], [fire_num], [year_], [acres],
[fire_name], [unit_id], [Shape_Leng], [Shape_Area],
[geom].Reduce(500) as SimpleGeom
FROM [Test_spatial_<username>].[dbo].[us_historic_fire_perimeters_dd83]
WHERE [geom].Reduce(500).STGeometryType() =
'MultiPolygon' OR [geom].Reduce(500).STGeometryType() =
'Polygon'
```
This example query may greatly reduce data size. (Some objects are reduced to Points, and these are discarded.)

**Use RAWSQL**

To use RAWSQL with spatial data, you can create a calculated field using the RAWSQL_SPATIAL function. For example:

```plaintext
RAWSQAL_SPATIAL("Select %1.STIntersection(%2.STBuffer(200))", [school_geom], [park_geom])
```

This formula returns unique spatial data based on the intersection of two values. For more information about using RAWSQL with spatial data, see **Pass-Through Functions (RAWSQL)** on page 1331. For more information about how to create calculated fields, see **Create a calculated field** on page 1222.

**Troubleshooting Microsoft SQL Server spatial connections**

**Performance issues**

When working with joined spatial columns with a large number of records in Microsoft SQL Server, duplicate marks can occur. This can result in slow performance.

To improve performance, extract the joined data source. For more information, see **Extract Your Data** on page 773.

**Common error messages**

**When the spatial table is using an unsupported spatial reference**

**Error message:**

An error occurred while communicating with the Microsoft SQL Server data source '<data source name>'.

Bad Connection: Tableau could not connect to the data source.

[Microsoft][ODBC Driver 13 for SQL Server][SQL Server]Operand type clash: geometry is incompatible with geography

[Microsoft][ODBC Driver 13 for SQL Server][SQL Server]Statement(s) could not be prepared.

**Solution:**
- Export the spatial data from Microsoft SQL Server to a Shapefile and connect to it in Tableau. Spatial file connections go through a data transformation process that supports thousands of projections.

- Export the data from Microsoft SQL Server and transform the data using a GIS tool like QGIS or ArcGIS. Then reload the table back into Microsoft SQL Server.

When the spatial table includes multiple types of spatial objects

**Error message:**
Unable to complete action
The spatial operation resulted in a MixedGeometry or MixedGeography, which Tableau does not support yet.
The spatial operation resulted in a MixedGeometry or MixedGeography, which Tableau does not support yet.

**Solution:**
Use Custom SQL to filter the data. For more information, see If you get an error about mixed data types on page 1912 Custom SQL example.

When measures are being aggregated in the visualization

**Error message:**
An error occurred while communicating with the Microsoft SQL Server data source 'data source name'
Bad Connection: Tableau could not connect to the data source.
Cannot perform an aggregate function on an expression containing an aggregate or a subquery.

**Solution:**
Disaggregate measures in the visualization: Select Analysis, and then clear Aggregate Measures.

Example - Build a map from Microsoft SQL Server spatial data
The following example demonstrates how to create the map below, which shows schools within 600 meters of parks in Seattle, Washington.
This example uses a Microsoft SQL Server connection to a database called, TestSpatial. The following tables from this database are used:

- seattleelementaryschools
- allseattleparks

Step 1: Connect

1. Open Tableau Desktop and connect to Microsoft SQL Server.
2. On the Data Source page, in the left pane, drag **New Custom SQL** onto the canvas.
3. Type or paste a query into the Edit Custom SQL dialog box that appears.

   For this example, the following query is used:

   ```sql
   SELECT S.[common nam] as SchoolName, P.[common nam] as ParkName, S.geom.STBuffer(<Parameters.Radius>) as school_geom, P.geom as park_geom FROM TestSpatial.dbo.seattleelementaryschools S LEFT JOIN
   ```
TestSpatial.dbo.allseattleparks P on S.geom.STBuffer((Parameters.Radius)).STIntersects(P.geom) = 1

**Note:** This example references a parameter named "Radius". For more information, see *Use parameters in a custom SQL query* on page 730.

The result of this query looks like the following:

Step 2: Build the map

1. Go to a new worksheet.
2. In the **Data** pane, double-click a spatial field.

In this example, **school_geom** is used. When double-clicked, it is automatically added to Detail on the Marks card and a map view is created.
3. From the **Data** pane, drag a dimension to **Label** on the Marks card. In this example, **School Name** is used.

4. On the **Rows** shelf, control-click (Command-click on a Mac) and drag **Latitude (generated)** to the right. This copies the field.

   A duplicate map appears below the first map and the Marks card updates with two **Latitude(generated)** tabs. The top tab is for the map on the top and the bottom tab is for the map on the bottom.
5. On the Marks card, click the bottom **Latitude (generated)** tab, and remove the fields by dragging them back to the Data pane.

   The bottom map updates to a blank map.

6. Select **Analysis > Create Calculated field**.

7. In the Calculation editor that opens, do the following:
   
   - Name the calculated field. In this example, the calculated field is named, Intersection.
   - Enter a RAWSQL formula. This example uses the following formula:

   ```rawsql
   RAWSQL_SPATIAL("Select %1.STIntersection(%2.STBuffer (200))", [school_geom], [park_geom])
   ```

   This formula returns unique spatial data based on the intersection of two values.
   - When finished, click **OK**.

8. Select **Analysis**, and then clear **Aggregate Measures**.

9. From the **Data** pane, drag the new calculated field (in this case, **Intersection**) to **Detail**
on the Marks card.

The bottom map updates with new marks. Notice that the marks are unique spatial shapes due to the RAWSQL formula.

10. From the **Data** pane, drag a dimension to **Detail** on the Marks card. This disaggregates the marks.

    In this example, the dimension, ParkName, is used.

11. From the **Data** pane, drag the same dimension to **Color** on the Marks card.

    In this example, each park is now assigned a color.
12. On the Rows shelf, right-click the **Latitude (generated)** field on the right and select **Dual Axis**.

The maps are now combined and the data is layered. For more information, see *Create Dual-Axis (Layered) Maps in Tableau* on page 2018.
13. Format the map. For more information, see Customize How Your Map Looks on page 2047

In this example, the following formatting is applied:

- The School Name marks are colored gray.
- The School Name mark labels are colored orange.
- The map background style is set to Dark.
- The Streets and Highways layer is shown on the background map.

The map is now finished.
Join Spatial Files in Tableau

This article demonstrates how to join spatial files in Tableau using an example.

You can connect to the following spatial data sources: Shapefiles, MapInfo tables, KML (Keyhole Markup Language) files, and GeoJSON files.

Important Note

Tableau supports joining two spatial data sources using their spatial features (geography or geometry). You can only create spatial joins between **points** and **polygons**.
In this article

Join spatial files below
Troubleshoot spatial joins on page 1925

Join spatial files

1. Open Tableau and connect to the first spatial data source. For more information on the types of spatial files you can connect to in Tableau, as well as how to connect to them, see the Spatial File on page 1906 connector example.

2. On the upper-left side of the Data Source page, under Connections, click Add.

3. In the Add a Data Source menu that appears, connect to the second spatial data source. The two spatial data sources are added to the canvas.

4. Click the Join icon.

5. In the Join dialog box that appears, do the following:
   - Select a join type.
     For more information about each of these types, see Overview of join types on page 659.
   - Under Data Source, select a spatial field to join by. (Note, Geometry is the default field name for spatial file sources except in SQL Server, where users create field names). Spatial fields have a globe icon 🌍 next to them.
   - For the second data source, select another spatial field.
   - Click the = sign and then select Intersects from the drop-down menu. You can
only intersect two spatial fields.

6. When finished, close the Join dialog box.

You are now ready to start analyzing your spatial data.
Spatial data can be used to create maps or other chart types in Tableau. For more information on building different chart types, see Build Charts and Analyze Data on page 843.

Troubleshoot spatial joins

SQL Server error: Geometry is incompatible with geography

Although SQL Server supports both a geography and a geometry data type, Tableau only supports geography data type from SQL Server from the following geographies: EPSG:WGS84 = 4326, EPSG:NAD83 = 4269, EPSG:ETRS89=4258. If you try to add a different geography, or a geometry field from SQL Server to your analysis, you'll receive an error.

Vertex ordering

Different spatial systems may order their vertices differently. If two spatial files contain two different vertex orders, it may cause problems with your analysis. Tableau interprets ordering specification according to the order of the spatial data source.

See Also:

Spatial File on page 1906
Join Your Data on page 657
Create Tableau Maps from Spatial Files on page 1959
Create Dual-Axis (Layered) Maps in Tableau on page 2018

Geocode Locations Tableau Does Not Recognize and Plot Them on a Map

If you have locations that Tableau can't map, such as street addresses, you can custom geocode those locations. Custom geocoding means that you assign latitude and longitude coordinates to your locations so Tableau can plot them accurately. It also allows you to create custom geographic roles that you can use as you create map views in Tableau.

For example, if your data contains country, state/province, and street address data, Tableau Desktop will geocode your data to the country and state/province level, but will not recognize the street address data. In this case, you can create a custom geographic role for the street address data so that you can plot it on a map view in Tableau.

Follow the steps below to learn how to custom geocode your data.
Here's what you need to do:

1. Create one or more CSV files with your location data
2. Create a schema.ini file (optional)
3. Import the files into Tableau
4. Assign geographic roles to your fields

**Before you start:** Custom geocoding your data can be tricky. We recommend that you check out the Assign Geographic Roles on page 1941 topic to learn more about geographic roles and the types of data Tableau recognizes before you get started.

Create a CSV file with your location data

The first step to custom geocoding your data is creating a CSV file to import into Tableau. The contents of this file differ depending on whether you are extending an existing geographic hierarchy, adding a new geographic role, or adding a new hierarchy to Tableau.

However, the following applies to all custom geocoding files:

- The .csv file must contain Latitude and Longitude columns.
- The latitude and longitude values you add must be real numbers. Make sure to include at least one decimal place when specifying these values.

Extend an Existing Role

The built-in geographic roles in Tableau contain hierarchies that can be extended to include locations relevant to your data. For example, the existing hierarchy of Country > State/Province may not contain all of the states or provinces in your data. You can extend this level to include missing states or provinces.

The import file for this type of geocoding should contain every level of the hierarchy above the level you are extending. For example, if you are extending (adding data to) the State/Province geographic role, the existing State/Province hierarchy has a column for Country and State/Province, along with Latitude and Longitude.
In the import file, the names of the columns define the geographic roles. When extending an existing role, the column names must match the existing geographic roles in the hierarchy that you are extending. This will ensure that the new locations are added to the proper roles and hierarchies.

See the table below for information on how to organize hierarchies in your import file.

<table>
<thead>
<tr>
<th>Built-In Hierarchy</th>
<th>Columns to include in the .csv file (in order; left to right)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country (Name)</td>
<td>Country (Name)</td>
</tr>
<tr>
<td></td>
<td>Latitude</td>
</tr>
<tr>
<td></td>
<td>Longitude</td>
</tr>
<tr>
<td>Country (Name), State/Province</td>
<td>Country (Name)</td>
</tr>
<tr>
<td></td>
<td>State/Province</td>
</tr>
<tr>
<td></td>
<td>Latitude</td>
</tr>
<tr>
<td></td>
<td>Longitude</td>
</tr>
<tr>
<td>Country (Name), State/Province, City</td>
<td>Country (Name)</td>
</tr>
<tr>
<td></td>
<td>State/Province</td>
</tr>
<tr>
<td></td>
<td>City</td>
</tr>
<tr>
<td></td>
<td>Latitude</td>
</tr>
<tr>
<td></td>
<td>Longitude</td>
</tr>
<tr>
<td>Country (Name), State/Province, County</td>
<td>Country (Name)</td>
</tr>
<tr>
<td></td>
<td>State/Province</td>
</tr>
<tr>
<td></td>
<td>County</td>
</tr>
</tbody>
</table>
| Country (Name), ZIP Code/Post-code | Country (Name)       
|                                   | ZIP Code/Postcode   
|                                   | Latitude           
|                                   | Longitude          
| Country (Name), Area Code         | Country (Name)      
|                                   | Area Code          
|                                   | Latitude           
|                                   | Longitude          
| Country (Name), CBSA              | Country (Name)      
|                                   | CBSA               
|                                   | Latitude           
|                                   | Longitude          

In addition to the Country (Name) column, you can optionally include the following columns: Country 2 char (ISO 3166-1), Country 3 char (ISO 3166-1), and Country (FIPS 10). If you include these columns, they should be just to the right of the Country (Name) column in any order.

**Note:** The highest level in the hierarchy is Country and cannot be extended to include higher levels such as Continent, etc.

### Add New Roles

To add new geographic roles to the existing geographic hierarchy in Tableau, format your import file to include the new roles along with their parent roles.

When adding new roles to an existing hierarchy, the import file for those roles needs to contain the columns for each level in the existing hierarchy. For more information on the columns to include in each geographic hierarchy, see the built in hierarchy table in the Extend An Existing Role section.
Below is an example of an import file containing the locations of crater impacts in North America. Importing the file below would add the geographic role Crater Name to the existing Country (Name), State hierarchy. Notice that the column name for country matches the existing Country (Name) geographic role.

<table>
<thead>
<tr>
<th>Crater Name</th>
<th>State</th>
<th>Country (Name)</th>
<th>latitude</th>
<th>longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>Oklahoma</td>
<td>U.S.A.</td>
<td>35.007752</td>
<td>-97.002877</td>
</tr>
<tr>
<td>Arch</td>
<td>Alaska</td>
<td>U.S.A.</td>
<td>64.200841</td>
<td>-149.493873</td>
</tr>
<tr>
<td>Barringer</td>
<td>Arizona</td>
<td>U.S.A.</td>
<td>34.048928</td>
<td>-111.093731</td>
</tr>
<tr>
<td>Bassinhead</td>
<td>Montana</td>
<td>U.S.A.</td>
<td>46.879682</td>
<td>-110.362566</td>
</tr>
<tr>
<td>Fort</td>
<td>Ontario</td>
<td>Canada</td>
<td>51.253776</td>
<td>-85.252314</td>
</tr>
<tr>
<td>Lake</td>
<td>Michigan</td>
<td>USA</td>
<td>44.31484</td>
<td>-85.622364</td>
</tr>
<tr>
<td>Grant</td>
<td>Saskatchewan</td>
<td>Canada</td>
<td>52.939916</td>
<td>-106.40864</td>
</tr>
<tr>
<td>Guadalupe</td>
<td>Quebec</td>
<td>Canada</td>
<td>46.813678</td>
<td>-71.267981</td>
</tr>
<tr>
<td>Guadalupe</td>
<td>Virginia</td>
<td>U.S.A.</td>
<td>37.431573</td>
<td>-78.868694</td>
</tr>
<tr>
<td>Guadalupe</td>
<td>Yukonian</td>
<td>Mexico</td>
<td>20.709879</td>
<td>-88.084338</td>
</tr>
</tbody>
</table>

Once you have created your import file, save the file as a Comma Delimited (.csv) file (Windows Comma Separated if on a Mac) in a folder on your computer.

(Optional): Create a schema.ini file

Sometimes when you attempt to import custom geographic information in Tableau using a .csv file, you may see a ".csv could not be used because it does not contain a unique column" error message.

This is because your .csv file might contain numerical data, such as numeric postcodes. Tableau will only accept text fields for new geographic roles. However, you can create a schema.ini file to tell Tableau that the numeric field you want to import should be treated as a text field.

To learn how to create a schema.ini file, see Create a schema.ini File on page 1932.

: Import custom geocoding file(s) into Tableau Desktop

Once you have created a .csv file with custom geocoding you can import that file into Tableau.

1. Open Tableau Desktop and navigate to a new or existing worksheet.
2. Select Map > Geocoding > Import Custom Geocoding.
3. In the Import Custom Geocoding dialog box, click the button to the right of the text field to browse to the folder your import file is saved in. All files in the folder will be imported into Tableau.
4. Click **Import**.

The custom geocoding data is imported into the workbook and the new geographic roles become available.

**Where does Tableau store my custom geocoding data?**

When you import custom geocoding, the data is stored in the Local Data folder in your Tableau Repository. To import custom geocoding, your Tableau Repository must be on a local hard drive. The custom geocoding is then available for all workbooks.

Save custom geocoding files in a separate folder on your local computer. If you have several different sets of custom geocoding files, save each of them under their own folder names.

You can remove the custom geocoding stored in your Tableau Repository by clicking **Map > Geocoding > Remove Custom Geocoding**. This will not remove the geocoding from a packaged workbook, but it will remove it from the Local Data folder in your My Tableau Repository.
What happens to my custom geocoding data when I save my workbook?

When you save your workbook as a packaged workbook, the custom geocoding data is packaged with the workbook. When you open a packaged workbook you can import the custom geocoding from that workbook into your own Tableau Repository.

**Note:** Importing a new custom geocoding file will replace any custom geographic roles previously imported into Tableau.

---

**Assign geographic roles to your fields**

Once you import custom geocoding into your workbook, the custom geographic roles become available. The next step is to assign those geographic roles to fields in your data source.

For example, if you decided to add a new geographic role that lists the latitude and longitude coordinates of crater impacts in North America, you can assign that geographic role to a field in your data source that lists the names of craters so when you create a map view with that field, the crater impacts are plotted in the correct locations on the map view.

To assign a custom geographic role to a field, in the Data pane, click the data type icon next to the field and select **Geographic Role**. Next, select a custom geographic role from the list.

(Back to top)
You're now ready to start building a map view with your custom geographic roles. For more information on how to build a map view, see Mapping Concepts in Tableau on page 1890.

See also:

Data Blending vs. Custom Geocoding on page 1939

Blend Geographic Data on page 1934

Create a schema.ini File

Sometimes when you attempt to import custom geographic information in Tableau using a .csv file, you may see a ".csv could not be used because it does not contain a unique column" error message.

This is because your .csv file might contain numerical data, such as numeric postcodes. Tableau will only accept text fields for new geographic roles. However, you can create a
schema.ini file to tell Tableau that the numeric field you want to import should be treated as a text field.

A schema.ini file is a Microsoft configuration file associated with the Microsoft JET engine. It tells JET how to interpret the contents of text files, including data structure, date formats, and other settings. For example, creating a schema.ini file that specifies that the numeric postcodes in your import file should be treated as text will allow you to plot the postcodes on a map.

To create a schema.ini file:

1. Open a text file.

2. In the text file, using the following syntax, specify the names and data types for each column in your import file:

   ![Schema ini file example](example.png)

   For example, if your import (.csv) file contained the columns **German Post Code**, **Latitude**, and **Longitude**, and was named **German Postal Codes.csv**, your schema.ini file would look like the following:

   ![Schema ini file example](example.png)

3. Save the file as **schema.ini** in the same folder as your .csv file.
You’re now ready to import your .csv and schema.ini files into Tableau.

See also:

Geocode Locations Tableau Does Not Recognize and Plot Them on a Map on page 1925

Blend Geographic Data

This example demonstrates how to blend geographic data in Tableau using two small sample data sources. Follow the steps in this topic to learn how to create a file that defines your geographic data, blend two data sources, and build the map view below using the two data sources.

The embedded view below shows famous theaters around the world, and was created by blending geographic data with another data source. Hover over the marks in the view to see information about each theater. If you would like to follow along with the steps in this topic, click Download in the view below, and then select Tableau Workbook. This will open the view in Tableau Desktop.
Step 1: Create a file that defines your geographic data

When you have a data source that contains geographic information that is not automatically geocoded in Tableau, the first step is to create a second data source that defines the latitudinal and longitudinal values for that information. You will later connect to this file in Tableau Desktop and blend it with your original data source.

Create a second data source with the following information:

- **Geographic name column**: This column includes any geographic information you want to plot on a map view and typically matches a geographic column name in your original data. For example, if your original data source has a column called Street Address, your second data source should also have a column called Street Address. The data in this column would then be all of the street addresses you want to plot on a map view.

  If the column does not match a geographic column name in your original data source, you may need to edit the relationships between your two data sources in Tableau Desktop. For more information, see Step 3.

- Finally, this column should not match the names of any geographic roles already in Tableau, such as **County**, **Area Code**, or **CBSA/MSA**.

  **Note**: The data in these columns can be strings or numbers. However, if your locations are numbers, especially numbers with leading zeros (for example, 00501), make sure the data type of the field is set to **String** in Tableau. For more information, see **Data Types** on page 263.

- **Latitude and Longitude columns**: The values in these columns need to be in decimal degrees (for example, 47.651808, -122.356881).

Below is an example of an original data source with geographic locations, and a second data source that defines those geographic locations.

<table>
<thead>
<tr>
<th>Original data source</th>
<th>Second data source</th>
</tr>
</thead>
</table>

Step 2: Connect to the data sources

1. Connect to the original data source.
2. Connect to the data source that defines your geographic data.

Step 3: Edit Relationships

(Optional) If the two data sources share the same column names, your data should blend automatically when you add fields to the view. However, if the file you created to define your geographic data does not share the same column name as the geographic data in your original data source, you will need to create a relationship between those two fields using the Edit Relationships option. For information about how to edit relationships, see Step 4: (Optional) Define or edit relationships on page 688.

Step 4: Plot the data on a map view

1. In the Data pane, select the second data source that defines your geographic data. Under Measures, assign the Latitude geographic role to the Latitude field and the
Longitude geographic role to the Longitude field. For more information, see Assign Geographic Roles on page 1941.

2. In the Data pane, select the original data source, and then drag the field you want to plot on a map to Detail on the Marks card.
3. In the **Data** pane, select the second data source, and then double-click **Latitude** and **Longitude** to add them to the **Columns** and **Rows** shelves.

The link icon ‼️ indicates that the **Street Address** field is blended with the **Street Address** field in the original data source. A broken link icon ❌️ indicates that the **Street Address** field is not blended with the **Street Address** field in the original data source.
The view is now complete. Each mark represents an address from the original data source.

See also:

**Data Blending vs. Custom Geocoding** below

**Data Blending vs. Custom Geocoding**

If you have geographic locations in your data that are not automatically geocoded in Tableau, there are two ways to plot them on a map view—data blending and custom geocoding. Both data blending and custom geocoding allow you to plot your own locations on a map. Data blending is easier to set up and you can work with data from any data source. Custom geocoding lets you
add to existing roles, and create hierarchies. Custom geocoding can be easier to use once it is set up and imported.

This topic explains the difference between blending geographic information with another data source and importing custom geocoding data into Tableau.

The following is a quick comparison of data blending and custom geocoding.

<table>
<thead>
<tr>
<th>Capability</th>
<th>Data Blending</th>
<th>Custom Geocoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plot your own locations on a map view</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Use any data</td>
<td>Yes</td>
<td>No, text files only</td>
</tr>
<tr>
<td>Add new geographic roles</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Add to an existing geographic role</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Create new geographic hierarchies</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Can be reused for other workbooks</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Data Blending**

Data blending is the easiest way to plot your data on a map view. Data blending works great if you are adding a single level of geographic information with a latitude and longitude. You can use any data source, unlike Custom geocoding where you can only use text files. You can share the geographic data source with other workbooks on Tableau Server. You can also speed up your map performance by creating an extract that contains only your geocoding data.

The following is a high-level procedure for blending geographic data. For a more detailed example, see **Blend Geographic Data** on page 1934.

**To blend geographic data**

1. Create a data source that defines your geographic data.
2. In Tableau Desktop, connect to the original data you want to map, and then connect to the data source that defines your geographic data.
3. Blend the two data sources together. For more information, see **Blend Your Data** on page 682.
4. Plot the data on a map view.
Custom Geocoding

Custom geocoding is a more flexible way to plot your data on a map. Custom geocoding is available for all workbooks on a computer once the custom geocoding data is imported. The custom geocoding data will be copied into any packaged workbook (.twbx) or published workbook that uses a custom geographic role. This will make the workbooks about 40 MB larger.

If you upgrade Tableau Desktop, you may need to refresh your custom geocoding to take advantage of any fixes made to the geocoding data in Tableau.

In custom geocoding, you can use additional columns to define larger geographic locations. For instance, if you are creating a set of US census tracts, you may need to define which US county they fall in. You can include additional columns to define larger geographic locations in the import file.

You can also use multiple files for multiple geographic roles that have a matching relationship, meaning they share larger geographic data, such as country or state/province. Once the custom geocoding data is imported, you will see additional geographic roles that can be assigned to your geographic data.

Finally, custom geocoding lets you add additional places to an existing geographic role, such as adding new cities to the city role. It also allows you to define a hierarchy of geographic roles that extends the built-in hierarchies (e.g. census tracts in counties) or defines a new hierarchy (e.g. sub-territories in territories).

For more information about custom geocoding, see Geocode Locations Tableau Does Not Recognize and Plot Them on a Map on page 1925. Additionally, you can learn how to custom geocode data by watching the Custom Geocoding training video on the Tableau website.

See also:

Assign Geographic Roles below

Create a schema.ini File on page 1932

Assign Geographic Roles

This article describes how to assign a geographic role to a field in Tableau so you can use it to create a map view.
A geographic role associates each value in a field with a latitude and longitude value. When you assign a geographic role to a field, Tableau assigns latitude and longitude values to each location in your data based on data that is already built in to the Tableau map server.

In this article

Assign a geographic role to a field

Assigning a geographic role based on the type of location (such as state versus postcode) helps insure that your data is plotted correctly on your map view. For example, you can assign the City geographic role to a field that contains a list of city names.

When a field is assigned a geographic role, Tableau creates a map view when you add the field to Detail on the Marks card. In other words, Tableau geocodes the information in that field.

To assign a geographic role to a field:

• In the Data pane, click the data type icon next to the field, select Geographic Role, and then select the geographic role you want to assign to the field.
When you assign a geographic role to a field, Tableau adds two fields to the **Measures** area of the **Data** pane: Latitude (generated) and Longitude (generated).

These fields contain latitude and longitude values and are assigned the Latitude and Longitude geographic roles. If you double-click each of these fields, Tableau adds them to the Columns and Rows shelves and creates a map view using the Tableau background map.
Types of geographic roles in Tableau

The following table describes the geographic roles available in Tableau. Many of the roles are international, but some are limited to the U.S. only.

You can assign geographic roles to your fields based on the type of geographic data they contain. For example, you can assign the Airport geographic role to a field that contains International Air Transport Association (IATA) codes.

If your location data does not fit into one of these roles, you may have to import custom geocoding to plot the data on a map. For more information, see Geocode Locations Tableau Does Not Recognize and Plot Them on a Map on page 1925.

<table>
<thead>
<tr>
<th>Geographic Role</th>
<th>Assign this role to a field if it contains:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport</td>
<td>International Air Transport Association (IATA) or International Civil Aviation Organization (ICAO) airport codes.</td>
</tr>
<tr>
<td>Area Code (U.S.)</td>
<td>U.S. telephone area codes; numbers only.</td>
</tr>
<tr>
<td>CBSA/MSA (U.S.)</td>
<td>U.S. Core Based Statistical Areas (CBSA), which includes Metropolitan Statistical Areas (MSA), as defined by the U.S. Office of Management and Budget. CBSA/MSA Codes and Names are recognized.</td>
</tr>
<tr>
<td>City</td>
<td>Worldwide cities with population of 15,000 or more. Names are in English, French, German, Spanish, Brazilian-Portuguese, Japanese, Korean, and Chinese.</td>
</tr>
<tr>
<td>Congressional District (U.S.)</td>
<td>U.S. congressional districts.</td>
</tr>
<tr>
<td>Country/Region</td>
<td>Worldwide countries, regions, and territories. Names are in English, French, German, Spanish, Brazilian-Portuguese, Japanese, Korean, and Chinese. Tableau also recognizes, FIPS 10, ISO 3166-1 alpha 2, and ISO 3166-1 alpha 3. Names are included in various forms, including long, short, and various abbreviations.</td>
</tr>
<tr>
<td>County</td>
<td>Second-level administrative divisions for select countries. For example, U.S. counties, French départements, German kreise, etc.</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Second-level administrative division definitions vary by country. In Tableau, all second-level administrative divisions are geocoded with the County geographic role. For more information, see the Location Data that Tableau Supports for Building Map Views on page 1905 section.</td>
</tr>
<tr>
<td>NUTS Europe</td>
<td>NUTS (Nomenclature of Territorial Units for Statistics) levels 1 - 3 codes. Codes and names, including synonyms, are supported.</td>
</tr>
<tr>
<td>Latitude</td>
<td>Latitude in decimal degrees. Only available for numeric fields.</td>
</tr>
<tr>
<td>Longitude</td>
<td>Longitude in decimal degrees. Only available for numeric fields.</td>
</tr>
<tr>
<td>State/Province</td>
<td>Worldwide state, province, and other first-level administrative divisions. Names are in English, French, German, Spanish, Brazilian-Portuguese, Japanese, Korean, and Chinese. <strong>Note:</strong> Some names are available only in their local form.</td>
</tr>
<tr>
<td>ZIP Code/Post-code</td>
<td>ZIP codes and postcodes for select countries. For example, U.S. five-digit zip codes, Australian four-digit postcodes, German five-digit postcodes, etc. For more information, see the Location Data that Tableau Supports for Building Map Views on page 1905 section.</td>
</tr>
</tbody>
</table>

See also:

- **Location Data that Tableau Supports for Building Map Views on page 1905**
- **Geocode Locations Tableau Does Not Recognize and Plot Them on a Map on page 1925**
- **Blend Geographic Data on page 1934**
- **Create Tableau Maps from Spatial Files on page 1959**
- **Edit Locations on the next page**
Edit Locations

Sometimes Tableau will not recognize one or more of the location names in your data. When this happens those values are marked as **unknown** in the lower right corner of the view. This often happens because your data values may be spelled incorrectly or use an abbreviation that Tableau does not recognize. When this happens, you can edit the unknown location names to map to known locations.

To edit unknown location names

1. In the bottom right corner of the view, click the special values indicator. This opens the Special Values dialog box.
2. In the Special Values dialog box, select **Edit Locations**.

![Special Values dialog box](image)

**Note:** If the special values indicator is not visible, you can also select **Map > Edit Locations**.

3. In the Edit Locations dialog box, click on one of the **Unrecognized** cells to match a known location to your unknown data. When you click on an unrecognized cell, a search box appears. As you begin typing in the search box, Tableau generates a list of possible locations. Select a location from the list.
Alternatively, you can select **Enter Latitude and Longitude** from the drop-down menu to manually map the value to a location on the map.

**Note:** When you type a latitude and longitude, enter the values in decimal format.

**Ambiguous Locations**

Both unrecognized and ambiguous locations can be listed in the Edit Locations dialog box. You can usually fix ambiguous locations by specifying the **Country/Region** and/or **State/Province** options in the **Geographic Roles** section of the Edit Locations dialog box. For example, if you have several cities that are ambiguous, you can specify a State/Province to fix them.
Restrict Locations in the List

By default, all possible locations are listed in the drop-down menu. To restrict the list to locations that have not yet been matched to your data, select the **Show only unmatched locations in drop-down list** option in the bottom left corner of the Edit Locations dialog box.

See also:

*Assign Geographic Roles* on page 1941

Build Maps in Tableau
What type of map do you want to create?

Simple Map

Map from Spatial Files
Build a Simple Map

You can build several different types of maps for your geographic analysis in Tableau. If you're new to maps, or simply want to take advantage of the built in mapping capabilities that Tableau provides, you can create a simple point or filled (polygon) map similar to the examples below.

**Prerequisites**: To build a simple map, your data source must contain location data (location names, or latitude and longitude coordinates). If your data source does not contain location data, see the Map Data section for ways you can connect to location data.
This topic illustrates how to create a simple map using an example. To follow along with the example below, open Tableau Desktop and connect to the Sample-Superstore data source, which comes with Tableau.

Build a simple point map

1. Navigate to a worksheet.

2. In the Data pane, under Dimensions, double-click State.

   A map view is automatically created because the State field is a geographic field. To learn more about geographic fields and how to create them, see Assign a geographic role to a field on page 1942.
3. From Measures, drag **Sales** to **Size** on the Marks card.

   The data points on the map update to show the amount of sales proportionally.

4. Select **Maps > Map Layers**.
5. In the Map Layers pane, do the following:
   - Click the Style drop-down and select **Normal**.
   - Under Map Layers, clear **Country/Region Names**.

The background map updates with the new settings.

---

Build a simple filled (polygon) map

1. Navigate to a new worksheet.
2. In the **Data** pane, under Dimensions, double-click **State**.

   A map view is automatically created.
3. On the Marks card, click the Mark Type drop-down and select **Map**.

The map view updates to a filled (polygon) map.
4. From Dimensions, drag Sales to Color on the Marks card.

The polygons on the map update to show the amount of sales using color.
See also:

- **Mapping Concepts in Tableau** on page 1890
- **Build Maps in Tableau** on page 1948
- **Customize How Your Map Looks** on page 2047

### Create Tableau Maps from Spatial Files

You can connect to the following spatial file types: Shapefiles, MapInfo tables, KML (Keyhole Markup Language) files, and GeoJSON files. You can then create point, line, or polygon maps using the data in those files.

**Note:** In current versions of Tableau, you can only connect to point geometries, linear geometries, or polygons. You cannot connect to mixed geometry types.

Connecting to spatial data is supported in Tableau Desktop version 10.2 and later. If you have Tableau Desktop version 10.1 or earlier, and would like to use shapefile data to create map views, see the **Create Tableau Maps from Shapefiles** topic in the Tableau Desktop 10.1 Help.

### In this article

- **Connect to spatial files** on the next page
- **Join spatial data** on page 1961
- **Build a map view from spatial data** on page 1961
- **Filter geometries to improve view performance** on page 1961
- **Customize the appearance of geometries** on page 1964
  - Add color
  - Hide polygon lines
  - Specify which polygons or data points appear on top on page 1967
  - Adjust the size of data points on page 1968
- **Build a dual-axis map from spatial data** on page 1969
Where to find spatial files

If you do not already have spatial files, you can find them at many open data portals. You can also find them on websites for your city or for a particular organization, if they provide them.

Here are some examples:

- LONDON DATASTORE
- EGIS South Africa
- U.S. Energy Information Administration
- USGS Water Resources
- Geospatial Information Authority of Japan
- Data.gov
- Census.gov

Connect to spatial files

1. In Tableau Desktop, click the New Data Source icon and select **Spatial file**.
2. Navigate to the folder that contains your spatial data, select the spatial file you want to connect to, and then click **Open**.

**Note:** To connect to spatial files, you must include all of the following files in the same directory:

- **For Esri shapefiles:** The folder must contain .shp, .shx, and .dbf files.
- **For MapInfo tables:** The folder must contain .TAB, .DAT, .MAP, and .ID or .MID or .MIF files.
- **For KML files:** The folder must contain the .kml file. (No other files are required.)
- **For GeoJSON files:** The folder must contain the .geojson file. (No other files are required.)

How Tableau interprets your spatial data

As soon as you connect to your spatial data, Tableau reads the spatial reference information of the data set and transforms the data into latitude and longitude coordinates. All data, regardless of the spatial reference system, is transformed to **WGS84 (EPSG:4326)**.

**Note:** If your data does not display diacritics (accent marks on characters) properly, check to make sure the file is **UTF-8** encoded.
The Geometry field

When you connect to spatial data, Tableau creates a Geometry field for your point geometries or your polygons. You use the Geometry field to create a map with your spatial data.

The values in this field display the geometry primitive—for example, Point for point geometries, Linestring or Multilinestring for linear geometries, and Polygon or Multipolygon for polygons. But this is simply an alias for the underlying spatial data.

For more information on how to build a map with the Geometry field, see the Build a map view from spatial data below section.

Join spatial data

Sometimes your spatial data includes only the geometry information, and does not include any demographic or other information. In this case, you can join a spatial data source with another data source type, or even another spatial file that includes the additional data you need for your analysis, as long as the files have a column (or field) in common.

For example, you can join a KML file that contains custom geographies for school districts in Oregon, U.S. with an Excel spreadsheet that contains demographic information about those school districts.

For more information on the different ways to join data sources, see Join Your Data on page 657.

Build a map view from spatial data

1. Open a new worksheet.

2. In the Data pane, under Measures, double-click the Geometry field.

   The Geometry field is added to Detail on the Marks card, and the Latitude (generated) and Longitude (generated) fields are added to the Columns and Rows shelves. A map view is created.

Filter geometries to improve view performance

When working with large spatial data sets, your view might take a long time to render. If this is the case, you can filter the number of geometries being added to the view using another dimension in your data source. For more information about filtering data, see Filter Data from Your Views on page 1162.

For example, in the image below, the view has been filtered down to a small subset of polygons using a dimension (Family Nam). The data source, from the IUCN List of Threatened Species.
contains data on endangered mammals around the world. Therefore, the dimension Family Name contains a list of mammal family names. This view has been narrowed down to one family name: rhinoceroses. Polygons for only rhinoceroses are shown in the view.

Without the filter, the polygons for every mammal in the data source are shown around the world, and the view takes a long time to render every time you perform an action, such as select a mark in the view.
Add levels of detail to the view

The Geometry field is a measure, and by default, is aggregated into a single mark using the COLLECT aggregation when it is added to the view. All of your polygons or marks will be in the view, but they will operate as a single mark. You therefore need to:

- Add additional levels of detail to the view to break it up into separate marks (based on the level of detail you specify)

or

- Dissaggregate the data all together so every single mark (polygon or data point) is separate.

To add additional levels of detail to the view:

- From Dimensions, drag one or more fields to Detail on the Marks card.

To disaggregate the data:
Click **Analysis**, and then clear **Aggregate Measures**.

Customize the appearance of geometries

You can customize the appearance of points, polygons, and lines by adding color, hiding polygon lines, specifying which polygons or data points appear on top, and adjusting the size of your data points.

**Add color**

To add color to your data points or polygons, drag a dimension or measure to **Color** on the Marks card.

For example, in the images above, the dimension (Presence), is placed on Color to represent the presence of an animal in a particular area.

**Hide polygon lines**

By default, polygon lines are shown when you create a polygon map from spatial data. If you want a cleaner view, you can remove them.

Take the following images, for example. The first image shows polygon lines. The second image does not show polygon lines.
Polygon lines shown by default.
Polygon lines removed.

To hide polygon lines:
1. On the Marks card, click **Color**.

2. Under Effects, select the **Border** drop-down, and then click **None**.

Specify which polygons or data points appear on top

Your polygons or data points might overlap or cover each other. You can specify which polygons or data points appear on top if you have a color or size legend in the view.

For example, in the image below, notice that there is a smaller polygon hidden behind the larger teal polygon in southern Africa.
You can rearrange the items in your legend to control which data points or polygons appear on top. To do so, in the legend, select the item you want to be on top, and then drag it to the top of the list.

Adjust the size of data points

If you’re using point geometries, you can adjust the size of the points on the map view. This is useful if you want to proportion your data points by quantitative values, such as by average.
sales, or profit.

**To adjust the size of your data points:**

1. From the **Data** pane, drag a measure to **Size** on the Marks card.
2. On the Marks card, click the Mark Type drop-down and select **Circle**.
3. Optional: From the **Data** pane, drag one or more dimensions to **Detail** on the Marks card to add more data points to your view.

**Note:** The level of detail determines which data points are sized. Add additional dimensions to Detail on the Marks card to add levels of detail (more data points), otherwise you might end up with one large data point.

For more information about how to add levels of detail to the view, see the **Add levels of detail to the view** on page 1963 section.

**Build a dual-axis map from spatial data**

If you join a spatial file either with another spatial file, or a different file type, you can create a dual-axis map using the geographic data from those files. This enables you to create more than one layer of your data on a map.

For example, the following is a dual-axis map view that was created using two spatial files. It contains two maps; one map shows the boroughs of New York City as polygons, and the other
shows data points for subway entrances around the city. The subway entrance data is layered over top of the city boroughs polygons.

1. In Tableau Desktop, open a new worksheet.
2. Connect to your data sources.
3. Create the first map view.

   See Build a map view from spatial data on page 1961 above for how to build a map view from spatial files.

4. On the Columns shelf, control-drag (command-drag on a Mac) the Longitude field to copy it, and place it to the right of the first Longitude field.

   **Important:** This example uses the Latitude (generated) and Longitude (generated) fields that Tableau creates when you connect to spatial data. If your data source contains its own Latitude and Longitude fields, you can use them instead of the Tableau generated fields, or in combination with the Tableau
You now have two identical map views. There are now three tabs on the Marks card: one for each map view, and one for both views (All). You can use these to control the visual detail of the map views. The top Longitude tab corresponds to the map on the left of the view, and the bottom Longitude tab corresponds to the map on the right of the view.

5. On the Marks card, click one of the Longitude tabs, and then remove all fields on that tab.

   One of your map views is now blank.

6. Create the second map view by dragging the appropriate fields from the Data pane to the blank Longitude tab on the Marks card.
7. When your two map views are finished, on the Columns shelf, right-click the **Longitude** field on the right and select **Dual Axis**.

Your map data is now layered on one map view.
To change which data appears on top, on the Columns shelf, drag the `Longitude` field on the right, and place it in front of the `Longitude` field on the left.

See also

- **Spatial File** on page 1906
- **Tackle your geospatial analysis with ease in Tableau 10.2** (Tableau blog post)
- **Join Your Data** on page 657
- **Mapping Concepts in Tableau** on page 1890
- **Create Dual-Axis (Layered) Maps in Tableau** on page 2018

Create Maps that Show Quantitative Values in Tableau

You can create maps in Tableau Desktop that show quantitative values, similar to the example below. These types of maps are called proportional symbol maps.

Proportional symbol maps are great for showing quantitative values for individual locations. They can show one or two quantitative values per location (one value encoded with size, and, if necessary, another encoded with color). For example, you can plot earthquakes recorded between 1981 to 2014 around the world, and size them by magnitude. You can also color the data points by magnitude for additional visual detail.
This topic illustrates how to create a proportional symbol map using an example. Follow the example below to learn how to set up your data source, and build the view for a proportional symbol map.

Your data source

To create a proportional symbol map, your data source should include the following types of information:

- Quantitative values
- Latitude and Longitude coordinates or location names (if recognized by Tableau)

It's also recommended that your data contain a large variation of values, otherwise your symbols will appear approximately the same size in the view.

The following table is a snippet of the Earthquake data source, which is included in the Create Proportional Symbol Maps in Tableau Example Workbook on Tableau Public. It contains columns for earthquake magnitude and magnitude to the power of ten, and columns for latitude and longitude values. It also contains a column for the date and earthquake ID for added clarity and organization.

<table>
<thead>
<tr>
<th>Earthquake</th>
<th>ID</th>
<th>Magnitude</th>
<th>Magnitude^10</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Time</td>
<td>centennial19730101114235</td>
<td>6.00000</td>
<td>17,488,747.0-4</td>
<td>-35.570</td>
<td>-15.427</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------</td>
<td>---------</td>
<td>----------------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>1/2/73</td>
<td>pde1973010200532030-0_66</td>
<td>5.50000</td>
<td>25,329,516.2-1</td>
<td>-9.854</td>
<td>117.427</td>
</tr>
<tr>
<td>1/3/73</td>
<td>pde1973010302294280-0_33</td>
<td>4.80000</td>
<td>6,492,506.21</td>
<td>1.548</td>
<td>126.305</td>
</tr>
<tr>
<td>1/4/73</td>
<td>pde1973010400314200-0_33</td>
<td>4.50000</td>
<td>3,405,062.89</td>
<td>41.305</td>
<td>-29.272</td>
</tr>
<tr>
<td>1/5/73</td>
<td>pde1973010500394820-0_36</td>
<td>4.70000</td>
<td>5,259,913.22</td>
<td>0.683</td>
<td>-80.018</td>
</tr>
<tr>
<td>1/6/73</td>
<td>pde1973010606185230-0_83</td>
<td>4.90000</td>
<td>7,979,226.63</td>
<td>-22.354</td>
<td>-69.310</td>
</tr>
</tbody>
</table>

Basic map building blocks:

<table>
<thead>
<tr>
<th>Columns shelf:</th>
<th>Longitude (continuous measure, longitude geographic role assigned)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows shelf:</td>
<td>Latitude (continuous measure, latitude geographic role assigned)</td>
</tr>
<tr>
<td>Detail:</td>
<td>One or more dimensions</td>
</tr>
<tr>
<td>Size:</td>
<td>Measure (aggregated)</td>
</tr>
<tr>
<td>Mark type:</td>
<td>Automatic</td>
</tr>
</tbody>
</table>

Build the map view

To follow along with this example, download the Create Proportional Symbol Maps in Tableau Example Workbook from Tableau Public, and open it in Tableau Desktop.

1. Open a new worksheet.

2. In the Data pane, under Measures, double-click **Latitude** and **Longitude**.

   Latitude is added to the Rows shelf, and Longitude is added to the Columns shelf. A map view with one data point is created.
3. From Dimensions, drag **ID** to **Detail** on the Marks card. If a warning dialog appears, click **Add all members**.

A lower level of detail is added to the view.

4. From Measures, drag **Magnitude\(^{10}\)** to **Size** on the Marks card.

Note that the Magnitude\(^{10}\) field is used to encode size, instead of the Magnitude field. This is because Magnitude\(^{10}\) contains a wider range of values, so the differences between values can be seen visually.
You now have a proportional symbol map. The larger data points represent earthquakes with larger magnitudes, and the smaller data points represent earthquakes with smaller magnitudes.

In most cases, this is as far as you need to go to show quantitative values for single locations. However, in this case, since there are so many data points in the view, more visual detail is needed to help you differentiate between the earthquake magnitudes, and to help you spot any trends.

5. From Measures, drag **Magnitude** to **Color** on the Marks card.

6. On the Marks card, click **Color > Edit Colors**.

7. In the Edit Colors dialog box, do the following:
   - Click the color drop-down and select the **Orange-Blue Diverging** palette from the list.
   - Select **Stepped Color**, and then enter 8. This creates eight colors: four shades of orange, and four shades of blue.
   - Select **Reversed**. This reverses the palette so that orange represents a higher magnitude than blue.
   - Click **Advanced**, select **Center**, and then enter 7.
This shifts the color palette and ensures that any earthquake over 7.0 magnitude will appear orange in color, and any earthquake under 7.0 magnitude will appear blue in color.

- Click OK.

8. On the Marks card, click Color again, and then do the following:
   - For Opacity, enter 70%.
   - Under Effects, click the Border drop-down menu and select a dark blue border color.

The map view updates with new colors. The dark orange data points represent earthquakes with higher magnitudes, while the dark blue data points represent earthquakes with lower magnitudes. The opacity of the marks is at 70% so you can see where the data points overlap.

![Map view with new colors](image)

9. On the Marks card, right-click the ID field and select Sort.

10. In the Sort dialog box, do the following:
    - For Sort Order, select Descending.
    - For Sort By, select Field, and then click the drop-down and select Magnitude.
    - Click OK.

This sorts the data points in the view so that the larger magnitudes appear on top.

Your proportional symbol map is now complete.
Point location and attribute meaning

It's important to note that symbols on a map can sometimes be misinterpreted as representing actual ground area. For example, if you have a map view that plots earth impact craters across North America, and sizes each symbol by the diameter (in kilometers) of the impact area, you might get a map view that looks like this:
In this particular case, it could be very easy to interpret the size of these data points as representing the actual ground area of the craters. Your audience might assume that most of Montana, U.S. was destroyed by a crater, which is not accurate. In reality, the crater in Montana was simply one of the larger craters in the data source and has been sized accordingly.

In cases such as these, it might be helpful to include annotations, or explanations of what the size actually represents to avoid misinterpretations. Even if it seems obvious.

See Also:

Mapping Concepts in Tableau on page 1890

Assign Geographic Roles on page 1941

Create Maps that Highlight Visual Clusters of Data in Tableau

You can create maps in Tableau Desktop that help you spot visual clusters, similar to the example below. These types of maps are called point distribution maps.

Point distribution maps are great for showing the how locations of your data points are distributed.
This topic illustrates how to create a point distribution map using an example. Follow the example in this topic to learn how to set up your data source, and build the view for a point distribution map.

Your data source

To create a point distribution map, your data source should include the following types of information:

- Latitude and longitude coordinates for all of your locations

For example, the following table is a snippet of the Hail data source, which is included in the Create Point Distribution Maps in Tableau Example Workbook on Tableau Public. It contains columns for the latitude and longitude coordinates of hail storms across the United States from 1955 to the 2013.

<table>
<thead>
<tr>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>32.2000</td>
<td>-101.5000</td>
</tr>
<tr>
<td>38.5800</td>
<td>-92.5800</td>
</tr>
<tr>
<td>41.2000</td>
<td>-89.6800</td>
</tr>
<tr>
<td>39.2800</td>
<td>-87.4000</td>
</tr>
</tbody>
</table>
Basic map building blocks

<table>
<thead>
<tr>
<th>Columns shelf:</th>
<th>Longitude (continuous dimension, longitude geographic role assigned)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows shelf:</td>
<td>Latitude (continuous dimension, latitude geographic role assigned)</td>
</tr>
<tr>
<td>Mark type:</td>
<td>Automatic</td>
</tr>
</tbody>
</table>

Build the map view

To follow along with this example, download the Create Point Distribution Maps in Tableau Example Workbook from Tableau Public, and open it in Tableau Desktop.

1. Open a new worksheet.

2. Ensure that the **Latitude** geographic role is assigned to your latitude field, and the **Longitude** geographic role is assigned to your longitude field.
   
   For more information, see Assign a geographic role to a field on page 1942.

3. In the Data pane, double-click **Latitude** and **Longitude** to add them to the view.
   
   The Latitude and Longitude fields are added to the Columns and Rows shelves, and a map view with one data point is created.
4. On the Columns shelf, right-click **Longitude** and select **Dimension**.
5. On the Rows shelf, right-click Latitude and select Dimension. The map view updates with every location in your data source.

   **Note:** You might need to filter some data points from the view.

6. On the Marks card, click Size, and then move the slider to the left.
Your point distribution map is now complete. A data point for every location in your data source is now plotted on the map (minus any you filtered from the view). You can now see that a majority of hailstorms occur in the eastern half of the United States.

See also:

- **Mapping Concepts in Tableau** on page 1890
- **Create Maps that Show Quantitative Values in Tableau** on page 1973
Create Maps that Show Ratio or Aggregated Data in Tableau

You can create maps in Tableau Desktop that show ratio or aggregated data, similar to the example below. These types of maps are called choropleth maps, or filled maps.

Choropleth maps are best for showing ratio or aggregated data for polygons. These polygons can be counties, regions, states, or any area or region that can be geocoded in Tableau. They can even be custom territories created in Tableau. For more information, see Create Territories on a Map on page 2061.

This topic illustrates how to create a choropleth map using an example. Follow the example below to learn how to set up your data source, and build the view for a choropleth map.

Your data source

To create a choropleth map, your data source should include the following types of information:

- Quantitative or qualitative values.
- Location names (if recognized by Tableau), or custom polygons. For more information, see Create Tableau Maps from Spatial Files on page 1959.

The following table is a snippet of the countyObesity + (Obesity_State_County) data source, which is included in the Create Choropleth Maps in Tableau Example Workbook on Tableau Public. It contains columns for State, County, and Obesity Percent - 2012.
<table>
<thead>
<tr>
<th>State</th>
<th>County</th>
<th>Obesity Percent - 2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama</td>
<td>Autauga County</td>
<td>30.9000</td>
</tr>
<tr>
<td>Alabama</td>
<td>Baldwin County</td>
<td>26.7000</td>
</tr>
<tr>
<td>Alabama</td>
<td>Barbour County</td>
<td>40.8000</td>
</tr>
<tr>
<td>Alabama</td>
<td>Bibb County</td>
<td>40.1000</td>
</tr>
<tr>
<td>Alabama</td>
<td>Blount County</td>
<td>32.4000</td>
</tr>
<tr>
<td>Alabama</td>
<td>Bullock County</td>
<td>44.5000</td>
</tr>
</tbody>
</table>

Basic map building blocks:

<table>
<thead>
<tr>
<th>Columns shelf:</th>
<th>Longitude (continuous measure, longitude geographic role assigned)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows shelf:</td>
<td>Latitude (continuous measure, latitude geographic role assigned)</td>
</tr>
<tr>
<td>Detail:</td>
<td>One or more geographic units (dimensions with geographic roles assigned)</td>
</tr>
<tr>
<td>Color:</td>
<td>Measure or dimension</td>
</tr>
<tr>
<td>Mark type:</td>
<td>Automatic or Map</td>
</tr>
</tbody>
</table>

Build the map view

To follow along with this example, download the Create Choropleth Maps in Tableau Example Workbook from Tableau Public, and open it in Tableau Desktop.

1. Open a new worksheet.

2. In the **Data** pane, under Dimensions, state_Obesity, double-click **State**.
   
   State is added to Detail on the Marks card and Longitude and Latitude are added to the Columns and Rows shelves. A map view is created with a data point for each state in your data source.

3. In the map view, select the **Alaska** and **Hawaii** data points, and then click **Exclude** on the tooltip that appears.
4. On the Marks card, click the plus icon on the **State** field to drill down to the County level of detail.

There is now a data point for every county in your data source.

5. From Measures, under countyObesity, drag **Percent-2012** to **Color** on the Marks card.

The map view changes to a filled map mark type and the polygons are colored green.
Note that the default aggregation for the Percent-2012 measure is SUM by default.

6. On the Marks card, right-click the **Percent-2012** field and select **Attribute**.
Since the data in this field is already a percentage, it does not make sense to aggregate it as a sum.

7. On the Marks card, click **Color** and select **Edit Colors**.

![Edit Colors dialog box](image)

8. In the Edit Color dialog box, click the color drop-down, select **Orange**, and then click **OK**.

9. On the Marks card, click **Color** and then, under Effects, click the **Border** drop-down and select **None**.
You now choropleth map is now complete. There is an obesity percentage for each polygon (county) on the map view.
Note about color distribution

The distribution you specify for the color of the polygons highly affects how people interpret your data. For example, the map you created in the **Build the map view on page 1987** section used the Tableau default color distribution. Color was distributed evenly across all values. The map view looked like this:
However, if you decide to distribute the values across five colors (stepped color, in Tableau), the map view might look like this:

Or maybe you want to see quartiles. Well, then your map might look like this:

Notice how the high obesity rates in the South are more visible when you use quartiles? The color distribution you choose depends on the information you're trying to show.
For example, if you wanted to show that obesity rates are higher in the South, you might choose the quartile example. However, if you want to show that obesity rates are high in the United States in general, you might choose the first example, which distributed color evenly. Neither map is right or wrong; they just tell different stories.

Note about level of detail

The level of detail you specify in a choropleth map determines the patterns you want to see in the data. If you want to see a simplified pattern, consider aggregating up to a higher level of detail. If you want to dig in to find smaller trends, consider aggregating down to a lower level of detail.

For example, the map view you created in the Build the map view on page 1987 section was aggregated at the County level of detail. The county level of detail was the lowest level of detail in the data source. At a higher level of detail, such as the State level, the map view might look similar to the following. Notice that the patterns in the data are simplified at this level of detail.

See also:

Mapping Concepts in Tableau on page 1890
Assign Geographic Roles on page 1941
Create Maps that Show Quantitative Values in Tableau on page 1973
Create Territories on a Map on page 2061
Create Maps that Show a Path Over Time in Tableau

You can create maps in Tableau Desktop that show a path over time, similar to the example below. These types of maps are called flow maps, or path maps.

Flow maps are great for when you want to show where something went over time, such as the path of a storm.

This topic illustrates how to create a flow map using an example. Follow the example below to learn how to set up your data source, and build the view for a flow map.

Your data source

Note: Starting in Tableau version 10.4, you can connect to spatial files that contain linear geometries. If you have spatial data with linear geometries, you may not need to perform the steps below. To learn how to create a map using spatial data with linear geometries, see Create Tableau Maps from Spatial Files on page 1959

To create a flow map, your data source should include the following types of information:
- Latitude and longitude coordinates for each data point in a path
- A column to define the order to connect the points (this can be date information, or manually applied numbers, such as 1, 2, 3, 4, 5)
- A unique ID for each path
- Enough data points to shape each path into a line

For example, the following table is a snippet of the Storm data source, which is included in the Create Flow Maps in Tableau Example Workbook on Tableau Public. It contains data on the paths of storms, and has columns for **Latitude** and **Longitude**, **Date**, and **Storm Name**. In this example, the Date column is used as an order to connect the data points, and the Storm Name column is used as a unique ID for each path.

Though only a few data points for the storm PAKHAR are shown in this example, the actual data source has enough entries to provide a detailed path for every storm recorded in 2012.

Note that the table also includes two optional columns: Basin and Wind Speed. These fields can be used to quickly filter and add visual detail to the view. You'll see how in the section, **Build the map view on the next page**.

<table>
<thead>
<tr>
<th>Storm Name</th>
<th>Date</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Basin</th>
<th>Wind Speed (kt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAKHAR</td>
<td>3/26/12 12:00:00 AM</td>
<td>9.5000</td>
<td>115.700</td>
<td>West Pacific</td>
<td>0</td>
</tr>
<tr>
<td>PAKHAR</td>
<td>3/26/12 6:00:00 AM</td>
<td>9.5000</td>
<td>115.400</td>
<td>West Pacific</td>
<td>0</td>
</tr>
<tr>
<td>PAKHAR</td>
<td>3/26/12 12:00:00 PM</td>
<td>9.5000</td>
<td>115.100</td>
<td>West Pacific</td>
<td>0</td>
</tr>
<tr>
<td>PAKHAR</td>
<td>3/26/12 6:00:00 PM</td>
<td>9.4000</td>
<td>114.800</td>
<td>West Pacific</td>
<td>0</td>
</tr>
<tr>
<td>PAKHAR</td>
<td>3/27/12 12:00:00 AM</td>
<td>9.4000</td>
<td>114.500</td>
<td>West Pacific</td>
<td>0</td>
</tr>
<tr>
<td>PAKHAR</td>
<td>3/27/12 6:00:00 AM</td>
<td>9.4000</td>
<td>114.300</td>
<td>West Pacific</td>
<td>35</td>
</tr>
</tbody>
</table>
Basic map building blocks:

<table>
<thead>
<tr>
<th><strong>Columns shelf</strong></th>
<th><strong>Rows shelf</strong></th>
<th><strong>Detail</strong></th>
<th><strong>Path</strong></th>
<th><strong>Mark type</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$Longitude$ (continuous measure, longitude geographic role assigned)</td>
<td>$Latitude$ (continuous measure, latitude geographic role assigned)</td>
<td>$Dimension$ (unique ID for each path)</td>
<td>$Date field or order field$ to define order to connect the data points</td>
<td>$Line$</td>
</tr>
</tbody>
</table>

Build the map view

To follow along with this example, download the Create Flow Maps in Tableau Example Workbook from Tableau Public, and open it in Tableau Desktop.

1. Open a new worksheet.

2. In the Data pane, under Measures, double-click $Latitude$ and $Longitude$.

   The Latitude and Longitude fields are added to the Columns and Rows shelves, and a map view with one data point is created.

3. From Dimensions, drag $Storm Name$ to $Detail$ on the Marks card.

   The map view updates with a data point for every storm in the data source. In the next steps, you will narrow the storms down to only those that occurred in the western Pacific Ocean in 2012.

4. From Dimensions, drag $Date$ to the Filters shelf.

5. In the Filter Field [Date] dialog box that appears, select $Years$, and then click $Next$.

6. In the Filter [Year of Date] dialog box that appears, click $2012$, and then click $OK$.

   The map view updates to show only the storms that occurred in 2012.

7. From Dimensions, drag $Basin$ to the Filters shelf.

8. In the Filter Field [Basin] dialog box that appears, select $West Pacific$, and then click $OK$.

   The map view updates to show only storms that occurred in the western part of the Pacific Ocean.
9. On the Marks card, click the mark-type drop-down and select **Line**.

A Path button appears on the Marks card, and the map view updates with a line connecting every data point.
10. From Dimensions, drag **Date** to **Path** on the Marks card.

The line disappears. This is because the Date field is set to discrete years. Since the date field in the Storm data source includes day, month, year, and time, this is not the correct level of detail for this field.

11. On the Marks card, right-click the **YEAR(Date)** field and select **Exact Date**.

Now the map view updates with a data point for every recorded date and time. You can now see the individual paths of each storm.
12. From Measures, drag **Wind Speed** to **Size** on the Marks card.

   The map view updates to show the varying wind speeds along each storm path.

13. On the Marks card, right-click the **SUM(Wind Speed)** field and select **Measure > Average**.

14. From Dimensions, drag **Storm Name** to **Color** on the Marks card.

   Each storm path is assigned a color and the flow map is now complete.

   You can now see the paths for every recorded storm that occurred in the West Pacific basin in 2012. You can also see at what point in their path their wind speeds were strongest.
Create Maps that Show Paths Between Origins and Destinations in Tableau

You can create maps in Tableau Desktop that show paths between origins and destinations, similar to the examples below. These types of maps are called spider maps, or origin-destination maps.

Spider maps are great for when you’re working with hubs that connect to many surrounding points. They are an excellent way to show the path between an origin and one or more destination locations.

There are several ways to create spider maps in Tableau. This topic illustrates how to create a spider map using two examples. Follow the examples below to learn how to set up your data source, and build the view for two different spider maps.
For other examples that might fit closer to your data, see the following workbooks on Tableau Public:

- Puget Sound Radio Tower Bandwidth
- Recruitment Strategies in the English Premier League
- 2014 U.S. Flight Departure Delays

Example 1: Metro station traffic in Paris, France

Learn how to build it!

Your data source

Note: Starting in Tableau version 10.4, you can connect to spatial files that contain linear geometries. If you have spatial data with linear geometries, you may not need to perform the steps below. To learn how to create a flow map using spatial data with linear geometries, see Create Tableau Maps from Spatial Files on page 1959

To create this type of spider map in Tableau, your data source should include the following information:

- A path ID for every unique path. See The Line Group (Path ID) column on the next page below for more information.
- Numbers to define the drawing order of each data point (location). See The Order of Points column on the next page below for more information.
- Latitude and longitude coordinates for every location. See the example table below.

It's also recommended that your data source contain a column with your location names, but it is not required.

The following table is a snippet of the Transports data source, which is included in the Create Spider Maps in Tableau Example 1 Workbook on Tableau Public. It contains metro traffic data for all metro lines in Paris, France. The first three metro stations for two metro lines are included in this example, and it contains the required columns Line Group (Path ID), Order of Points, Latitude, and Longitude. It also contains the additional columns Line, Station, and Traffic for added clarity and organization, but they’re not required to build the map view.
<table>
<thead>
<tr>
<th>Line</th>
<th>Line Group (Path ID)</th>
<th>Order of Points</th>
<th>Station</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>La Défense (Grande Arche)</td>
<td>48.891934</td>
<td>2.237883</td>
<td>14,275,382</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>Esplanade de la Défense</td>
<td>48.887843</td>
<td>2.250442</td>
<td>9,843,051</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>Pont de Neuilly</td>
<td>48.884509</td>
<td>2.259892</td>
<td>6,902,931</td>
</tr>
<tr>
<td>10 BOUCLE</td>
<td>10</td>
<td>1</td>
<td>Boulogne-Jean-Jaurès</td>
<td>48.842222</td>
<td>2.238836</td>
<td>3,847,782</td>
</tr>
<tr>
<td>10 BOUCLE</td>
<td>10</td>
<td>2</td>
<td>Porte d'Auteuil</td>
<td>48.848074</td>
<td>2.258648</td>
<td>687,237</td>
</tr>
<tr>
<td>10 BOUCLE</td>
<td>10</td>
<td>3</td>
<td>Michel-Ange-Auteuil</td>
<td>48.847740</td>
<td>2.264297</td>
<td>2,222,709</td>
</tr>
</tbody>
</table>

**The Line Group (Path ID) column**

For each path, there is a unique key or string, which is added to every location in that path. In this example, the Line Group (Path ID) column is used to identify each unique path. You will use this column to create your spider map.

For example, in the table above, there are two metro lines (1 and 10 Boucle), and each of those metro lines have a unique path ID listed in the Line Group (Path ID) column. For metro line 1, the Line Group is 1. For metro line 10 Boucle, the Line Group is 10. Every location in line one contains the line group 1, and every location in line 10 Boucle contains the line group 10.

**The Order of Points column**

This column defines the point order and the direction in which the line is drawn from the first data point to the last data point on your map, which, in this example is the start and end of the metro line. This field is very important if you have more than two locations in a path, because it draws your paths in the desired order (you can think of it like connecting the dots).
In the example table above, there are three stations on line 1. They are listed in order from La Défense (Grande Arche) to Pont de Neuilly using the numbers 1 through 3 in the **Order of Points** column. The same is true for line 10 Boucle.

Basic map building blocks:

<table>
<thead>
<tr>
<th>Columns shelf:</th>
<th>Longitude (continuous measure, longitude geographic role assigned)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows shelf:</td>
<td>Latitude (continuous measure, latitude geographic role assigned)</td>
</tr>
<tr>
<td>Detail:</td>
<td>Path ID field (discrete dimension)</td>
</tr>
<tr>
<td>Path:</td>
<td>Order field (continuous dimension)</td>
</tr>
<tr>
<td>Mark type:</td>
<td>Line</td>
</tr>
</tbody>
</table>

**Build the map view**

To follow along with this example, download the [Create Spider Maps in Tableau Example 1 Workbook](#) from Tableau Public, and open it in Tableau Desktop.

1. In the **Create Spider Maps in Tableau Example Workbook**, click the New Worksheet icon.
2. In the new worksheet, from Measures, drag **Longitude** to the **Columns** shelf, and **Latitude** to the **Rows** shelf.
3. From Dimensions, drag **Line Group (Path ID)** to **Detail** on the Marks card.
4. On the Marks card, click the Mark Type drop-down and select **Line**.

The map view updates with a line connecting all of the points. The Path button should appear on the Marks card.
5. From Measures, drag **Point Order** to **Path** on the Marks card.

   Point Order is aggregated as a sum.

6. On the Marks card, right-click the **SUM(Point Order)** field and select **Dimension**.

   The map view updates with a line for every metro line.
7. From Dimensions, drag **Line Group (Path ID)** to **Color** on the Marks card.

   Each line now has its own color associated with it, and a color legend is added to the view.

8. From Measures, drag **Longitude** to the **Columns** shelf and place it to the right of the first Longitude field.

   There are now two Longitude fields on the columns shelf. The view updates with two identical maps. The Marks card updates with two tabs: one for the map on the left, and one for the map on the right. You can customize each of these tabs to change the visual detail of each map view. There is an All tab to control the visual detail in both maps at the same time.
9. On the Marks card, click the bottom **AVG(Longitude)** tab.

10. On the Marks card, under the bottom AVG (Longitude) tab, click the Mark type drop-down and select **Automatic**.

    The map view on the right updates to be a point map.
11. On the Columns shelf, right-click the second **AVG (Longitude)** field (on the right), and select **Dual Axis**.
Your map views are now layered on top of one another.

12. From Measures, drag **Traffic** to **Size**, on the bottom AVG (Longitude) Marks card. The size of the data points update to show the amount of traffic per station.

13. On the Marks card, click **Size** and move the slider to the right.

14. On the Marks card, click **Color**, and then, under Effects, click the **Border** drop-down and select a color. The view is now complete. You can quickly find the stations on each metro line with the most traffic.

Filter the amount of information in the view

If you want to filter the amount of lines you see in the view:

- From Dimensions, drag **Line Group** to the Filters shelf.

You can also right-click the **Line Group** field and select **Show Filter**, to show a filter card in the view.
Example 2: Bike share data in Seattle, Washington

Learn how to build it!

Your data source

Note: Starting in Tableau version 10.4, you can connect to spatial files that contain linear geometries. If you have spatial data with linear geometries, you may not need to perform the steps below. To learn how to create a flow map using spatial data with linear geometries, see Create Tableau Maps from Spatial Files on page 1959

Similar to the first example, for this type of spider map, your data source should contain the following information:

- A Path ID for every unique path. See The Path ID column on page 2011 below for more information.
- Latitude and longitude coordinates for every location. See the example table below.

For this type of spider map, there should also be two rows in your data source for each path. That means you should have one row for your origin location data, and one row for your
destination location data, repeated for every path. This is a crucial step that enables Tableau to draw your paths correctly. See **Origin-destination rows** below for more information.

It’s also recommended that your data source contain a column with your location names, but it is not required.

This example is a snippet of the Seattle bike share data source, which is included in the Create Spider Maps in Tableau Example 2 Workbook. It has a column for **Origin-Destination**, **Station**, **Path ID**, **Latitude**, and **Longitude**. Only the last three columns in this data source are required to create a spider map, but the Origin-Destination and Location Name columns provide further clarity and organization.

<table>
<thead>
<tr>
<th>Origin-Destination</th>
<th>Station</th>
<th>Path ID</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin</td>
<td>BT-01</td>
<td>BT-01_BT-01</td>
<td>47.61841</td>
<td>-122.35101</td>
</tr>
<tr>
<td>Destination</td>
<td>BT-01</td>
<td>BT-01_BT-01</td>
<td>47.61841</td>
<td>-122.35101</td>
</tr>
<tr>
<td>Origin</td>
<td>BT-01</td>
<td>BT-01_BT-03</td>
<td>47.61841</td>
<td>-122.35101</td>
</tr>
<tr>
<td>Destination</td>
<td>BT-03</td>
<td>BT-01_BT-03</td>
<td>47.61576</td>
<td>-122.34843</td>
</tr>
<tr>
<td>Origin</td>
<td>BT-01</td>
<td>BT-01_BT-04</td>
<td>47.61841</td>
<td>-122.35101</td>
</tr>
<tr>
<td>Destination</td>
<td>BT-04</td>
<td>BT-01_BT-04</td>
<td>47.61613</td>
<td>-122.34108</td>
</tr>
<tr>
<td>Origin</td>
<td>BT-01</td>
<td>BT-01_BT-05</td>
<td>47.61841</td>
<td>-122.35101</td>
</tr>
<tr>
<td>Destination</td>
<td>BT-05</td>
<td>BT-01_BT-05</td>
<td>47.61303</td>
<td>-122.34410</td>
</tr>
</tbody>
</table>

**Origin-destination rows**

For each unique path you want to create, you need a row for your origin location, and a row for your destination location in your data source. This means your origin location will be paired with every destination location.

For example, when showing the path between an origin bike share location and several destination locations in a city, you need a row for the origin location, and a row for the destination location for **every single path**.

In the example above, the origin station BT-01 is paired with several different destination locations (BT-01, BT-03, BT-04, BT-05) to show that bikes were checked out of the BT-01 location and returned either to the same location or to a different location. Each origin-
destination pair is color-coded to show that they make up one path. There is a column for Origin-Destination to further illustrate this concept, but this column is not required.

**The Path ID column**

The Path ID column is used to identify each unique origin-to-destination path. You will use this column to create your spider map.

For each origin and destination location, there is a unique key or string that identifies them as a pair.

In the example below, for the first origin-destination path, the Path ID is BT-01_BT-01. For the second origin-destination path, the Path ID is BT-01_BT-03. Each path ID is listed twice, once for the origin location, and once for the destination location. Again, each pair is color-coded to indicate that they make up one path.

<table>
<thead>
<tr>
<th>Origin-Destination</th>
<th>Location Name</th>
<th>Path ID</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin</td>
<td>BT-01</td>
<td>BT-01_BT-01</td>
<td>47.61841</td>
<td>-122.35101</td>
</tr>
<tr>
<td>Destination</td>
<td>BT-01</td>
<td>BT-01_BT-01</td>
<td>47.61841</td>
<td>-122.35101</td>
</tr>
<tr>
<td>Origin</td>
<td>BT-01</td>
<td>BT-01_BT-03</td>
<td>47.61841</td>
<td>-122.35101</td>
</tr>
<tr>
<td>Destination</td>
<td>BT-03</td>
<td>BT-01_BT-03</td>
<td>47.61576</td>
<td>-122.34843</td>
</tr>
</tbody>
</table>

**Note:** Your path ID can be anything you want. However, if you want to create complex calculated fields to help filter the locations later on, it helps if your path IDs are consistent across all paths. A great way to do this is to create Path IDs that are a combination of your origin and destination location names, separated by a delimiter. For example, the Path ID for the origin location BT-01 and the destination location BT-03 is BT-01_BT-03. For an example of why you might want to do this, see the Option 2: Create a dynamic filter: on page 2015 section.

Basic map building blocks:

<table>
<thead>
<tr>
<th>Columns shelf:</th>
<th>Longitude (continuous dimension, longitude geographic role assigned)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows shelf:</td>
<td>Latitude (continuous dimension, latitude geographic role assigned)</td>
</tr>
</tbody>
</table>
Detail: | Path ID field (discrete dimension)
---|---
Mark type: | Line

Build the map view

After you set up your data source, you can connect to it in Tableau Desktop and build a spider map. To follow along with this example, download the Create Spider Maps in Tableau Example 2 Workbook from Tableau Public.

1. From Measures, drag Longitude to the Columns shelf, and Latitude to the Rows shelf.

2. On the Columns shelf, right-click the Longitude field and select Dimension. Do the same for the Latitude field on the Rows shelf.

This will ensure Tableau does not aggregate your origin and destination locations.

You should see all the locations in the data source as data points on your map. In this example, there is a data point for every bike share location in the data source.

3. On the Marks card, click the Marks Type drop-down, and select Line. The view updates to show a line connecting each data point, and the Marks card updates with a Path button.
4. From Dimensions, drag **Path ID** to **Detail** on the Marks card.

If you have only a few origin-destination pairs, your view might look something like the following:
However, if you have many origin-destination pairs, your view might look more like this:

This is very common, and can be fixed by filtering a large portion of your paths from the view. Continue to the following section to learn a couple of ways to do so.
Filter the amount of information in the view

If your data source contains a lot of origin-destination pairs, you can filter most of them from the view.

Option 1: Create a simple filter:

1. From Dimensions, drag Path ID to the Filters shelf.
2. In the Filter dialog box that opens, do the following:
   - Under the General tab, select None.
   - Click the Wildcard tab.
   - Under the Wildcard tab, for Match Value, enter BT-01, and then click Starts With.
   - Click OK.
   This filters the view down to only the paths that start with BT-01.

Option 2: Create a dynamic filter:

You can also create a calculated field and combine it with a parameter so you can switch between the paths you want to see right in the view. Follow the steps below to learn how.

Step 1: Create the parameter

1. On the Data pane, to the right of Dimensions, click the Data pane drop-down and select Create Parameter.
2. In the Create Parameter dialog box, do the following:
   - Name the parameter StationSelected.
   - For Data type, select String
   - For Allowable values, click List, select Add from field, and then select the Location Names field.
   - Click OK.

Step 2: Create the calculated field

1. Select Analysis > Create Calculated Field.
2. In the calculation editor, name the calculated field, Select by Origin-Destination, and then enter the following formula:
IF

LEFT([Path ID], FIND([Path ID], "") -1) = [StationSelected] THEN "Origin"

ELSEIF

RIGHT([Path ID], LEN([Path ID]) - FIND([Path ID], ")")
= [StationSelected] THEN "Destination"

ELSE

"Unselected stations"

END

In the table above, each Path ID (for example BT-01_BT-03) contains an underscore (_) as a delimiter to separate the origin location name (BT-01) from the destination location name (BT-03). This delimiter is used in the formula to tell Tableau which locations (selected in the parameter you created in step 1 of this procedure) are origin locations, and which are destination locations. The parameter you created above is also used in the formula (StationSelected).

3. Check that the calculation is valid, and then click OK.

Step 3: Add the calculated field to the Filters shelf

1. From Dimensions, drag the calculated field to the Filters shelf.

2. In the Filter dialog box, select Origin, and then click OK.

Step 4: Show the parameter control

In the Data pane, under Parameters, right-click the parameter you created and select Show Parameter Control.

You can now select an origin from the parameter control in the view, and the marks on your map view will update.
See also:

Tableau Community post: Origin-Destination Maps (or Flow Maps)

Create Maps that Show a Path Over Time in Tableau on page 1994

Mapping Concepts in Tableau on page 1890

Create Parameters on page 1031

Filter Data from Your Views on page 1162

Create Dual-Axis (Layered) Maps in Tableau

This article demonstrates how to create dual axis (layered) maps in Tableau using several examples. A dual-axis map is a map with two sets of geographic data overlaid on top of one another. For example, a filled map of U.S. states with data points for each city layered on top.

There are three ways to create a dual-axis map in Tableau:

- By using Tableau Latitude(generated) and Longitude(generated) fields
- By using custom latitude and longitude fields
- By using a combination of Tableau Latitude(generated) and Longitude (generated) fields, and custom latitude and longitude fields

Follow the steps below to learn how to create a dual-axis map using each of these methods.

Create a dual-axis map from Tableau Latitude (generated) and Longitude (generated) fields

1. Open Tableau Desktop.

2. In the Connect pane, under Saved Data Sources, connect to the Sample-Superstore data source.

3. In the Data pane, under Dimensions, double-click State.

A map view is created.
4. On the Marks card, click the Mark Type drop-down and select **Map**.

5. From the **Data** pane, under **Measures**, drag **Sales** to **Color** on the Marks card.
The map updates to show the sum of sales by state. The states with higher sales are darker blue, and the states with fewer sales are lighter blue.

6. On the **Columns** shelf, control-drag (command-drag on a Mac) the **Longitude (generated)** field to copy it, and place it to the right of the first Longitude field.
7. On the Marks card, select the top **Longitude (generated)** tab.

8. From the **Data** pane, under Dimensions, drag **Region** to **Color** on the Marks card.

   The map view on the left updates.
9. On the Columns shelf, right-click the **Longitude (generated)** field on the right and select **Dual Axis**.

The map views are now overlapping each other. You might not be able to see the map on the bottom layer.
10. On the Marks card, ensure that the bottom Longitude (generated) tab is selected, and then click **Color > Edit Colors**.

11. In the Edit Colors dialog box that opens, click the Palette drop-down, select **Gray**, and then click **OK**.

   Notice that the colors of the map update.
12. On the Marks card, click **Color** again.

13. In the Color pop-up dialog box, under Opacity, move the slider to approximately 75%.

The dual axis (layered) map is now complete. For each region, you can now see which states has the most sales.
**Tip:** To change which map is on top, rearrange the Longitude (generated) fields on the Columns shelf.

Create a dual-axis map from custom latitude and longitude fields

If your data source contains custom latitude and longitude fields, you can use them instead of the Tableau Latitude (generated) and Longitude (generated) fields to create a dual-axis map. Follow the steps below to learn how.

1. Open Tableau and connect to a data source with custom latitude and longitude values.
2. Navigate to a new worksheet.
3. In the Data pane, right-click the custom latitude field and select **Geographic Role > Latitude**.
Note that the Latitude geographic role may already be assigned to the field.

4. In the Data pane, right-click the custom longitude field and select Geographic Role > Longitude.

Note that the Longitude geographic role may already be assigned to the field.

5. From the Data pane, drag the custom latitude field to the Rows shelf.

6. From the Data pane, drag the custom longitude field to the Columns shelf.

7. From the Data pane, under Dimensions, drag a geographic field to Detail on the Marks card.

In this example, the geographic field, Country (Name), is used.
8. On the Marks card, click the Mark Type drop-down and select Map.

The map updates to a filled map.
9. On the Rows shelf, Ctrl-click (Command-click on a Mac) and drag the custom latitude field to the right. This copies the field.

A second, identical map is created and the Marks card updates to include three tabs. The middle tab is for the top map, and the bottom tab is for the bottom map.

10. On the Marks card, click the bottom tab and remove the geographic field by dragging it off.

11. From the Data pane, drag a new geographic field to Detail on the Marks card.

In this example, Airport (City) is used.
12. On the Marks card, click **Color** and select a color for the marks. In this example, the color orange is used.

13. On the Rows shelf, right-click the custom latitude field on the right and select **Dual Axis**. The two maps are now combined.
Create a dual-axis map from a combination of generated and custom latitude and longitude fields

Follow the steps in each scenario to learn how to create a dual-axis map from generated and custom latitude and longitude fields.

Best practices for creating dual-axis maps with two sets of latitude and longitude fields:

- The Latitude(generated) and Longitude(generated) fields must be placed on the Columns and Rows shelves.
- The custom latitude and longitude fields must be placed on Detail on the Marks card, and then converted to dimensions. For more information, see steps 11 and 12 in Scenario 1.
- The custom latitude and longitude fields must be assigned the Latitude and Longitude geographic roles. For more information, see Assign Geographic Roles on page 1941.
When joining data sources, use a Full Outer join. For more information, see Join Your Data on page 657.

Scenario 1: Use generated and custom latitude and longitude fields from a single data source

1. Open Tableau Desktop and connect to a data source.
   
   In this example, an excel spreadsheet that contains country and city names, as well as custom latitude and longitude columns for airport locations around the world, is used.

<table>
<thead>
<tr>
<th>Airports (IATA)</th>
<th>Airports (Name)</th>
<th>Airports (City)</th>
<th>Airports Country (Name)</th>
<th>Airports Latitude</th>
<th>Airports Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>GKA</td>
<td>GOROKA</td>
<td>GOROKA</td>
<td>PAPUA NEW GUINEA</td>
<td>-6.0817</td>
<td>145.392</td>
</tr>
<tr>
<td>LAE</td>
<td>null</td>
<td>LAE</td>
<td>PAPUA NEW GUINEA</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>MAG</td>
<td>MADANG</td>
<td>MADANG</td>
<td>PAPUA NEW GUINEA</td>
<td>-5.2069</td>
<td>145.789</td>
</tr>
<tr>
<td>MSU</td>
<td>MOUNT HAGEN</td>
<td>MOUNT HAGEN</td>
<td>PAPUA NEW GUINEA</td>
<td>-5.8261</td>
<td>144.296</td>
</tr>
<tr>
<td>LAB</td>
<td>NAIZAB</td>
<td>NAIZAB</td>
<td>PAPUA NEW GUINEA</td>
<td>-6.5697</td>
<td>146.726</td>
</tr>
<tr>
<td>POM</td>
<td>PORT MORESBY</td>
<td>PORT MORESBY</td>
<td>PAPUA NEW GUINEA</td>
<td>-9.4453</td>
<td>147.220</td>
</tr>
<tr>
<td>RAB</td>
<td>null</td>
<td>RABAUL</td>
<td>PAPUA NEW GUINEA</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

2. Navigate to a new worksheet.
   
   Notice that, in the Data pane, under Measures, there are two sets of latitude and longitude fields: the custom latitude and longitude fields from the data source, and the generated latitude and longitude fields that Tableau creates from your geographic fields.

   Measures
   - Altitude
   - Latitude
   - Longitude
   - Latitude (generated)
   - Longitude (generated)
   - Number of Records
   - Measure Values

3. From the Data pane, under Measures, drag Longitude (generated) to the Columns shelf.

4. From the Data pane, under Measures, drag Latitude (generated) to the Rows shelf.
5. From the Data pane, under Dimensions, drag a geographic location to Detail on the Marks card.

In this example, **Country (Name)** is used. A map view with a data point for every country in the data source is created.

6. On the Marks card, click the Mark type drop-down and select **Map**.
7. On the Rows shelf, Ctrl-click (Command-click on Mac) and drag the **Latitude (generated)** field to the right. This copies the field.

A second, identical map is created and the Marks card updates to include two tabs labeled Latitude(generated). The top tab is for the top map, and the bottom tab is for the bottom map.
8. On the Marks card, click the bottom **Latitude (generated) tab** and remove **Country (name)**.

9. From the Data pane, under Measures, drag the custom **Latitude** field to **Detail** on the Marks card.

10. From the Data pane, under Measures, drag the custom **Longitude** field to **Detail** on the
Marks card.

11. On the Marks card, right-click the custom \textit{Latitude} field and select \textbf{Dimension}.

12. Repeat step 11 for the custom \textit{Longitude} field.

13. On the Marks card, click \textbf{Color}, and then select a new color.

   In this example, the color orange is used.

14. On the Rows shelf, right-click the \textit{Latitude(generated)} field on the right and select \textbf{Dual Axis}.
The two maps are now combined. You can filter the view as needed or zoom in to a particular area.
Check your work! Watch steps 2 - 14 below:

<table>
<thead>
<tr>
<th>Airport (IATA)</th>
<th>Airport (Name)</th>
<th>Airport (City)</th>
<th>Country (Name)</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Altitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAE</td>
<td>GORONA</td>
<td>GORONA</td>
<td>PAPUA NEW GUINEA</td>
<td>-4.5817</td>
<td>145.308</td>
<td>350</td>
</tr>
<tr>
<td>LAF</td>
<td>LAF</td>
<td>LAF</td>
<td>PAPUA NEW GUINEA</td>
<td>0.0000</td>
<td>0.000</td>
<td>0</td>
</tr>
<tr>
<td>WAG</td>
<td>MADAMO</td>
<td>MADAMO</td>
<td>PAPUA NEW GUINEA</td>
<td>0.0000</td>
<td>145.789</td>
<td>20</td>
</tr>
<tr>
<td>HGU</td>
<td>MOUNT MAGEN</td>
<td>MOUNT MAGEN</td>
<td>PAPUA NEW GUINEA</td>
<td>-6.8261</td>
<td>144.206</td>
<td>5,388</td>
</tr>
<tr>
<td>LAC</td>
<td>NAZARE</td>
<td>NAZARE</td>
<td>PAPUA NEW GUINEA</td>
<td>-6.5967</td>
<td>146.702</td>
<td>390</td>
</tr>
<tr>
<td>POM</td>
<td>PORT MORESBY</td>
<td>PORT MORESBY</td>
<td>PAPUA NEW GUINEA</td>
<td>-9.4483</td>
<td>147.220</td>
<td>148</td>
</tr>
<tr>
<td>RAQ</td>
<td>RABULA</td>
<td>RABULA</td>
<td>PAPUA NEW GUINEA</td>
<td>0.0000</td>
<td>0.000</td>
<td>0</td>
</tr>
<tr>
<td>WAX</td>
<td>WAMU INTERNATIONAL</td>
<td>WAMU</td>
<td>PAPUA NEW GUINEA</td>
<td>-3.5830</td>
<td>143.694</td>
<td>19</td>
</tr>
<tr>
<td>NIA</td>
<td>ANGAA/ WASSÅK</td>
<td>GREENLAND</td>
<td>0.0000</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
</tr>
<tr>
<td>NIA</td>
<td>ANGAA/ WASSÅK</td>
<td>GREENLAND</td>
<td>0.0000</td>
<td>0.000</td>
<td>0.000</td>
<td>0</td>
</tr>
</tbody>
</table>
Scenario 2: Join spatial data with an independent data source and plot the data from both on the same map

1. Open Tableau Desktop and connect to spatial data. For more information, see Connect to spatial files on page 1960.
   In this example, a shape file with polygon data for U.S. states is used.

2. On the Data Source page, in the Connections pane, click Add.

3. Connect to another data source.
   In this example, an excel spreadsheet with custom latitude and longitude columns is used.

4. Join the new data source to the spatial data using a Full Outer join. For more information on how to join data, see Join Your Data on page 657.
   In this example, the excel spreadsheet is joined to the shape file using columns that contain state names from each data source.

5. Navigate to a new worksheet.
   Notice that, in the Data pane, there are two sets of Dimensions and two sets of Measures: one for each of your data sources.

6. From the Data pane, under Measures, drag Latitude(generated) to the Rows shelf.

7. From the Data pane, under Measures, drag Longitude(generated) to the Columns
shelf.

A blank map is created.

8. **From the Data pane**, drag either the Geometry field from your spatial data source, or a geographic field from your excel data source, to **Detail** on the Marks card.

The field you choose is for your bottom layer of data. For example, if you have polygon data in your spatial file, use the Geometry field so that the bottom layer will be filled polygons. If you have linear geometries in your spatial file, use a geographic field from your excel data source and format the map to be a filled or point map.

In this example, the **Geometry** field from the shape file is used. The map updates to a filled map of the United States.

9. **On the Rows shelf**, Ctrl-click (Command-click on a Mac) and drag the **Latitude (generated)** field to the right. This copies the field.

   A second, identical map is created and the Marks card updates to include two tabs labeled Latitude(generated). The top tab is for the top map, and the bottom tab is for the bottom map.

10. **On the Marks card**, click the bottom **Latitude (generated) tab** and remove the
geographic field.

In this example, COLLECT(Geometry) is removed.

11. In the Data pane, under Measures, right-click the custom latitude field and select Geographic Role > Latitude. For more information, see Assign Geographic Roles on page 1941.

12. In the Data pane, under Measures, right-click the custom longitude field and select Geographic Role > Longitude.

13. From the Data pane, under Measures, drag the custom latitude field to Detail on the Marks card.

In this example, the custom latitude field is named Y.

14. From the Data pane, under Measures, drag the custom longitude field to Detail on the Marks card.

In this example, the custom longitude field is named X.
15. On the Marks card, right-click the custom latitude field and select **Dimension**.

16. Repeat step 13 for the custom longitude field.
17. On the Marks card, click **Color**, and then select a new color. In this example, the color orange is used.

18. On the Rows shelf, right-click the **Latitude(generated)** field on the right and select **Dual Axis**.

   The two maps are now combined. You can filter the view as needed or zoom in to a particular area.
Check your work! Watch steps 1-18 below:

See Also

**Build a dual-axis map from spatial data** on page 1969

**Join Your Data** on page 657

**Spatial File** on page 1906

**Assign Geographic Roles** on page 1941

**Create Filled Maps with Pie Charts in Tableau**

The pie mark type can be useful to show simple proportions to a relative whole. For example, pie marks might be effective when you want to show the percentage of profit for a product by
geographic location.

In this example, the pie mark type is used with the filled maps mark type to show the percentage of profit for office supplies, furniture, and technology, relative to the total profits by state. It uses the Sample-Superstore data source that comes with Tableau Desktop.

Build the map view

1. In Tableau, open a new workbook and connect to the Sample-Superstore data source.
2. On the Data Source page, click Sheet 1 to go to a new worksheet.
3. In the Data pane, under Dimensions, double-click State.

   Tableau creates a symbol map, with a data point for each state in the Sample-Superstore data source.

4. On the Marks card, click the Mark-type drop-down and select the Map mark type.

   ![Mark type drop-down menu](image)

5. From Measures, drag Profit to Color on the Marks card.

6. From Measures, drag Latitude (generated) to the Rows shelf, and place it to the right of the other Latitude field.
You should now have two identical map views.

7. On the Rows shelf, right-click the second **Latitude** field and select **Dual Axis**.
   
   The second map is now layered on top of the first map. There are now three drop-downs on the Marks card: one for each map view, and one for both views (all). These are three separate marks cards that you can use to control the visual detail for each of the map views.

8. On the Marks card, click one of the **Latitude (generated)** tabs, and then click the Mark type drop-down and select the **Pie** mark type.

9. From Measures, drag **Sales** to **Size** on the Latitude (generated) Marks card you selected.
   
   The Sum of sales for each state is shown as text.

10. From Dimensions, drag **Category** to **Color** on the same Marks card.
    
    If the size of the pie charts is too small, click **Size** on the Marks card to adjust the size.
    
    The map view now shows the sum of profit, as well as the sum of sales for each category, for each state.
See also:

Mapping Concepts in Tableau on page 1890

Get Started Mapping with Tableau on page 1862

Customize Maps

Customize How Your Map Looks

This article describes how you can customize the appearance of a map view in Tableau.

Note: Many of the tasks in this article make use of the Map Layers pane. To open this pane, select Map > Map Layers. In older versions of Tableau, select Map > Map Options.

In this article

Customize how the background map looks:

- Select a Tableau background map style on the next page
- Import your own background map on page 2049
Select a Tableau background map style

When you are connected to the Tableau background map, you can choose between three built-in background map styles (Normal, Light, or Dark). You can see the three styles below:

Normal          Light           Dark

To select a Tableau background map style:

1. Select **Map > Map Layers**.
2. In the Map Layers pane on the left-hand side of the workspace, under Background, click the **Style** drop-down menu, and then select a background map style.
For more information about the built-in Tableau background maps, see Select Background Maps on page 2071.

Import your own background map

If the built-in Tableau background map styles don't meet your needs, you can import your own background map from a Web Map Service (WMS) server or a Mapbox map.

To import your own background map:

1. Select Map > Background Maps > Map Services.
2. In the Map Services dialog box, click Add, and then select to add WMS servers or Mapbox maps.
3. Follow the prompts to connect Tableau to the map service of your choice.
   For more information, see Use Web Map Service (WMS) Servers on page 2078 or Use Mapbox Maps on page 2073.
4. When finished, click Close.
5. Select Map > Background Maps, and then select the background map you want to use.
   The map updates to include the background map you choose.

Add a static background image

In addition to importing your own background map, you can add a static background image to your workbook and plot your data on it. For example, you can take a Google Map image and
plot your data on it.

For more information, see the following topics:

- Use Background Images in Your Views on page 2094
- Add Images of Google Maps and OpenStreetMap as Background Images in Tableau on page 2104
- Find Background Image Coordinates on page 2106

Add or remove map layers

If you are using the Tableau background map, or a WMS map or Mapbox map that contains custom layers, you can add or subtract layers on your background map to mark points of interest on the map. For example, you can overlay streets and highways or county boundaries on the map to give your data context.

**To add or subtract map layers:**

1. Select **Map > Map Layers**.
2. In the Map Layers pane, under the Map Layers section, select one or multiple map layers.

**Note:** Some map layers are only visible at specific zoom levels. If a map layer is unavailable at your current level of zoom, it will appear grayed out.
To use layers that are not available, zoom further in to the view.

The map layers that come automatically with the Tableau background map are described in the table below.
<table>
<thead>
<tr>
<th>Layer Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>Shows the base map including water and land areas.</td>
</tr>
<tr>
<td>Land Cover</td>
<td>Shades wilderness areas and parks to give the map more depth.</td>
</tr>
<tr>
<td>Coastlines</td>
<td>Displays coastlines.</td>
</tr>
<tr>
<td>Streets and Highways</td>
<td>Marks freeways and highways as well as small city streets. This layer includes the highway and street names as well.</td>
</tr>
<tr>
<td>Light Country/Region Borders</td>
<td>Shows a light gray outline of country/region borders and names.</td>
</tr>
<tr>
<td>Light Country/Region Names</td>
<td>Shows country and region names in a light gray.</td>
</tr>
<tr>
<td>Country/Region Borders</td>
<td>Highlights country and region borders in a darker gray.</td>
</tr>
<tr>
<td>Country/Region Names</td>
<td>Highlights country and region names in a darker gray.</td>
</tr>
<tr>
<td>Light State/Province Borders</td>
<td>Shows a light gray outline of state borders and names.</td>
</tr>
<tr>
<td>Light State/Province Names</td>
<td>Shows state and province names in a light gray.</td>
</tr>
<tr>
<td>State/Province Borders</td>
<td>Highlights state and province borders in a darker gray.</td>
</tr>
<tr>
<td>State/Province Names</td>
<td>Highlights state and province names in a darker gray.</td>
</tr>
<tr>
<td>US County Borders</td>
<td>Highlights U.S. county borders.</td>
</tr>
<tr>
<td>US County Names</td>
<td>Highlights U.S. county names.</td>
</tr>
<tr>
<td>Zip Code Boundaries</td>
<td>Marks U.S. zip code boundaries. You must zoom in to see this layer.</td>
</tr>
<tr>
<td>Zip Code Labels</td>
<td>Shows labels for U.S. zip codes. You must zoom in to see this layer.</td>
</tr>
<tr>
<td>Layer Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Area Code Boundaries</td>
<td>Marks the U.S. area code boundaries. You must zoom in to see this layer.</td>
</tr>
<tr>
<td>Area Code Labels</td>
<td>Shows labels for the U.S. area codes. You must zoom in to see this layer.</td>
</tr>
<tr>
<td>US Metro Boundaries (CBSA)</td>
<td>Marks U.S. Metropolitan Statistical Areas and Micropolitan area boundaries.</td>
</tr>
<tr>
<td>US Metro Labels (CBSA)</td>
<td>Shows labels for the U.S. Metropolitan Statistical Areas and Micropolitan areas.</td>
</tr>
<tr>
<td>Place Names</td>
<td>Displays the names of places from country/region names and borders to city names, bodies of water, parks, universities, and more. This layer is dependent on the zoom level.</td>
</tr>
</tbody>
</table>

Add U.S. data layers

If you are using the Tableau background map, you can turn on a variety of predefined data layers that show U.S. census information.

**Note:** Map data layers are only available for locations in the U.S.

To add data layers for locations outside the U.S. you can connect to a WMS server that contains the desired demographic information. For more information, see Use Web Map Service (WMS) Servers on page 2078.

To add a U.S. data layer to your map view:

1. Select **Map > Map Layers**
2. In the **Map Layers** pane, under **Data Layer**, do the following:
   - Click the **Layer** drop-down menu and select a data layer.
   - Click the **By** drop-down menu, and then select to layer the data by **State, County**, ...
Zip Code or Block Group.

- Click the Using drop-down menu to select a color scheme.

Once you select a data layer, it is added as shading to the map and a legend is shown to explain the colors of the layers. To hide or show this legend at any time, select Map > Map Legend.

Change the mark type

By default, when you add a geographic field to the view, Tableau creates a point map. You can change this to a polygon (filled) map or a line map.

**Note:** Filled maps are not available at the city or postcode level.

To change a point map to a filled or line map:
- On the Marks card, click the Mark Type drop-down and select **Map**.

![Map Mark Type dropdown](image1)

**Add levels of detail**

With maps, for each level of detail you add, the more granular your data becomes. For example, you might look at obesity rates at the state level, or you could drill down into the county level, like the examples below. Adding or subtracting levels of detail changes the make up of your map.

![Map example 1](image2)

![Map example 2](image3)
To add levels of detail to the view:

- From Dimensions, drag a geographic field to **Detail** on the Marks card.

Add color

There are two ways you can add color to your map view: You can color locations categorically, or you can color locations quantitatively.

**To color locations on your map categorically:**

- From the **Data** pane, drag a dimension to **Color** on the Marks card.

  The image below shows each state in the U.S. colored by region: West, Central, South, and East. The dimension, Region, is on Color on the Marks card.

To color each location on your map quantitatively:

- From the **Data** pane, drag a measure to **Color** on the Marks card.

  The image below shows each state in the U.S. colored by the amount of sales they
achieved. The measure, Sales, is on Color on the Marks card.

For more information about color, see Color Palettes and Effects on page 1145.

Add labels

You can add labels to your locations to provide extra context. For example, you can add labels for location name and sales.

To add labels to your data, from the Data pane, drag a dimension or measure to Label on the Marks card.

A label appears in the center of your location (if a polygon), or to the side of your location (if a data point).

You can add multiple labels.
Adjust the size of your data points

You can adjust the size of your data points to compare and contrast them, or make smaller data points easier to see.

To uniformly adjust the size of your data points:
On the Marks card, click **Size**, and then adjust the slider to the left or right.

To size your data points quantitatively:
• From Measures, drag a field to **Size** on the Marks card.

**Create custom tooltips**

You can create custom tooltips to show additional information about your locations when your audience hovers or clicks on them. You can type in your own information to appear for all marks, or add a field that will update with information specific to each mark.
To add a field to a tooltip:

- From the Data pane, drag the field to **Tooltip** on the Marks card.

To edit a tooltip:

1. On the Marks card, click **Tooltip**.
2. In the Edit Tooltip dialog box, format the tooltip how you would like it to appear.

For more information about customizing tooltips, see **Format tooltips (Tableau Desktop only)** on page 2385 and **Add tooltips to marks** on page 1133.

See also:

- **Mapping Concepts in Tableau** on page 1890
- **Use Web Map Service (WMS) Servers** on page 2078
- **Use Mapbox Maps** on page 2073
- **Select Background Maps** on page 2071

Create Territories on a Map

When you’re analyzing data on a map view, you might want to group *existing* locations together to create your own territories or regions, such as sales territories for your organization.

In Tableau, there are several ways to create territories:

- **Option 1:** Select and group locations on a map on the next page
- **Option 2:** Create a territory from a geographic field on page 2065
- **Option 3:** Geocode a territory field using another geographic field on page 2065
Option 1: Select and group locations on a map

If you have created a map view, you can select and group locations (marks) on the map to create your territories.

For example, the following map view shows the total sales by postcode for a company with sales across Australia.

The first thing you might be thinking is: That's a lot of postcodes! In fact, there is a mark in this map view for every single 4-digit postcode in Australia, and each of those marks has an aggregation for SUM of Sales. If you're looking for sales information for every single postcode in Australia, than this is the view for you.

However, you might not want to analyze every individual location in your map view. Instead you might have your own areas, regions, or territories that you want to analyze as a whole. For example, maybe your organization splits Australia into five different sales territories, and you want to analyze the total sales for each of them.

In this case, because you've already created a map view, you can select locations directly on the map and group them to create your own territories. Follow the steps below to learn how.
Select locations on the map to group into a territory

On the map view, select the locations (marks) that you want to group into your own territory. For more information about how to select marks, see Select Marks on page 2612.

Create the first group (your first territory)

When you select marks in the view, or hover over a selection, a tooltip appears. On that tooltip, click the Group icon, and then select the level of detail you want to group the locations by.

A new group field is added in the Data pane with a group icon next to it. To change the locations within the group at any time, right-click the group field in the Data pane and select Edit group.

The new field is also added to Color on the Marks card.
Notice that the marks you selected are now all the same color.

: Continue grouping data to add territories

Repeat steps 1 and 2 to create additional territories. Each territory is added as a group within the group field you created in step 2. You can create as many territories as you want.

: Add the group field to the view

From the Data pane, drag the newly created group field to Detail on the Marks card. Next, to make each territory appear as a single mark, you must also remove the related geographic field(s) from the view.

For example, if you grouped postcodes at the Post Code, Country level, you must remove the Post Code and Country fields from the view.

In the image below, notice that the Post Code and Country fields are no longer on Detail on the Marks card because they have been replaced by the Post Code & Country (group) field.

In this example, there are five groups within the Post Code & Country (group) field, so five territories appear on the map. Aggregations, such as SUM (Sales), are computed at the level of the territory, rather than separately for each location within the territory.

You can now see the SUM (Sales) for each territory by color. Territory 5 has the most sales.
Option 2: Create a territory from a geographic field

You can also create territories by creating groups in the Data pane.

1. In the Data pane, right-click a geographic field (such as City or State) and select Create > Group.

2. In the Create Group dialog box, select the locations you want in your first territory and click Group. Each group you create represents a territory.

3. Repeat step 2 until you've created all of your territories, and then click OK. The new group field is added to the Data pane.

4. From the Data pane, drag the newly created group field to Detail on the Marks card. You can also drag the field to Color or Label on the Marks card to help differentiate between each territory.

Change locations in your territory groups

If you want to change the locations in your territory groups at any time, right-click the group field in the Data pane and select Edit group.

Option 3: Geocode a territory field using another geographic field

If your data source already contains a custom territory field, you can quickly geocode it to make a map in Tableau using an existing geographic field in your data source.
For example, the following data source contains a custom territory field for geographic areas called **Zone géographique**. It contains the values, **North**, **Central** and **South**.

If you were to bring this field into Tableau by itself, Tableau would not be able to recognize it as geographic data.

However, this data source also contains three geographic fields that Tableau does recognize: **Country**, **Region**, and **City**.

Each row in this data source contains a value for country, region, city and geographic zone. Therefore, each location is assigned a value from the custom territory field.

<table>
<thead>
<tr>
<th>Country</th>
<th>Region</th>
<th>City</th>
<th>Zone géographique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leeds</td>
<td>Angleterre</td>
<td>Royaume-Uni</td>
<td>Nord</td>
</tr>
<tr>
<td>Leeds</td>
<td>Angleterre</td>
<td>Royaume-Uni</td>
<td>Nord</td>
</tr>
<tr>
<td>West Bromwich</td>
<td>Angleterre</td>
<td>Royaume-Uni</td>
<td>Nord</td>
</tr>
<tr>
<td>West Bromwich</td>
<td>Angleterre</td>
<td>Royaume-Uni</td>
<td>Nord</td>
</tr>
<tr>
<td>West Bromwich</td>
<td>Angleterre</td>
<td>Royaume-Uni</td>
<td>Nord</td>
</tr>
<tr>
<td>Le Bouscat</td>
<td>Aquitaine</td>
<td>France</td>
<td>Centre</td>
</tr>
<tr>
<td>Le Bouscat</td>
<td>Aquitaine</td>
<td>France</td>
<td>Centre</td>
</tr>
<tr>
<td>Le Bouscat</td>
<td>Aquitaine</td>
<td>France</td>
<td>Centre</td>
</tr>
</tbody>
</table>

Because the Zone géographique field is included in the workbook with other geographic fields that contain locations that Tableau recognizes, you can geocode it at the level of detail of one of those fields.

Follow the steps below to geocode a territory field using another geographic field:

1. **Assign a geographic role to the territory field**

   On the data source page, click the data type icon for the custom territory field and select **Geographic Role > Create from**, and then select a level of detail to aggregate the field.
The options you see are based on the available geographic fields in your data source. Choose the level of aggregation you want in the view. If you're unsure how they will affect your view, try several different levels of detail. You can always undo and try again.

After you assign a geographic role to the territory field, it's added to its respective place in the location hierarchy. You can see it in the Data pane when you go to any worksheet.

**Note:** You can also click the data type icon next to the field in the Data pane to geocode a territory field using the above procedure.
: Add the territory field to the view

Create a map view and add the newly geocoded territory field to the Marks card. You can place it on Detail, Color, Label, or Tooltip. Note that fields on Detail determine the level of aggregation in the view.

The custom territory field functions similar to your other geographic fields.

In the example below, the custom territory field (Zone géographique) is on Detail and Color on the Marks card. The average profit is shown for each territory (since the lowest level of detail is the territory field).

If you add another geographic field to Detail on the Marks card, such as the country field (Pays), the aggregation will update to show information for that level of detail.

Blend custom territory data

If your data source does not contain a territory field, you can blend a territory field with another geographic field in your data source. For more information, see Blend Geographic Data on page 1934. After you have blended your custom territory field with another field, follow the procedure in Step 1: Assign a geographic role to the custom territory field to geocode your territory field.
Customize How People Interact with your Map

When you create a map view, there are several default ways you can explore and interact with the view. You can zoom in and out of the view, pan, select marks, and even search for locations worldwide with map search.

However, sometimes you want to limit some of the ways your audience can interact with your map.

You can customize how your audience interacts with your view in the following ways using the Map Options dialog box.

In this article

Show a map scale below
Hide map search on the next page
Hide the view toolbar on the next page
Turn off pan and zoom on the next page

Show a map scale

You can display a scale on your map so that your audience can understand distances between your data points. This scale appears in the bottom-right corner of the view, and updates as you zoom in and out of the map.

Note: A map scale will not display on non-Web Mercator projections, such as a WMS using Plate Carrée.

To show a map scale on your map:

1. Select Map > Map Options.
2. In the Map Options dialog box that appears in the view, select Show Map Scale.
3. Under Units, click the drop-down and select the units of measurement you want.
   - If you want the scale to show measurements in meters and kilometers, select Metric.
   - If you want the scale to show measurements in feet and miles, select U.S.
   - If you want the units of measurement to be determined by your workbook locale, select Automatic.
Note: This setting also determines the units that the Radial tool uses to measure distances. For more information, see Change the units of measurement on page 2093 in the Measure Distances between Data Points and Locations on a Map View topic.

Hide map search

You can hide the map search icon so your audience cannot search for locations in your map view.

To hide the map search icon:

1. Select Map > Map Options.
2. In the Map Options dialog box that appears in the view, clear Show Map Search.

Note: When you hide the map search icon, you also hide the locate me button for views published on the web. For more information about the locate me button, see Interact with Maps in the Tableau Online Help.

Hide the view toolbar

You can hide the view toolbar in a map view so your audience cannot lock the map in place or zoom the map to all of your data.

To hide the view toolbar:

1. Select Map > Map Options.
2. In the Map Options dialog box that appears in the view, clear Show View Toolbar.

Note: When you hide the view toolbar, you can still use keyboard shortcuts to zoom in and out of the view, pan, and select marks. For more information, see View Toolbar on page 2610.

Turn off pan and zoom

You can turn off pan and zoom in your map view, as well as in background images, so your audience cannot pan, or zoom in or out of the view.

To turn off pan and zoom:
1. Select **Map > Map Options**.

2. In the Map Options dialog box that appears in the view, clear **Allow Pan and Zoom**.

When you turn off pan and zoom, the pan tool and all zoom controls are removed from the view toolbar, and the rectangular tool becomes the default tool. Keyboard shortcuts for zooming in and out of the view, or panning, no longer work. If the map search icon is not hidden, you can still navigate the view by searching for locations with the map search box.

See also:

- **Search for Locations in Your Map** on page 2088
- **View Toolbar** on page 2610
- **Select Marks** on page 2612
- **Pan and Zoom** on page 2615

Select Background Maps

Tableau comes with a set of online and offline background maps that you can access to create map views.

By default, Tableau connects to an online map provider.

The three background maps that come with Tableau are described below. They can be found in the **Maps > Background Maps** menu. If the workbook author has added a WMS server or Mapbox map to the workbook, they will be listed using the name the author gave them in the **Maps > Background Maps** menu as well.

- **None**
  
  Displays data between latitude and longitude axes.

- **Offline**
  
  Available only for Tableau Desktop, this background map stores the images that make up the map in a cache on your machine for improved performance and offline access. For more information, see the offline maps section below.

- **Tableau**
  
  Connects to the Tableau background map. By default, all map views connect to this background map, unless you specify otherwise.
To select a new background map:

- In Tableau Desktop, Tableau Server, or Tableau Online, select **Map > Background Maps** and then select the background map you want to use.

Set a default background map in Tableau Desktop

By default, all map views connect to the Tableau background map. In Tableau Desktop, if you do not want maps views to automatically be created with the Tableau background map, you can specify a different background map as the default.

**To specify a default background map:**

1. In Tableau Desktop, select **Map > Background Maps**, and then select the background map that you want to make the default.

2. Select **Map > Background Maps > Set as Default** to set the selected background map to the default.

   The background map is automatically saved as a Tableau Map Source (.tms) and placed in the Mapsources folder of your My Tableau Repository. It is now the default background map for all new worksheets.

Use the Offline background map in Tableau Desktop

You can create and inspect data in a map view offline using the offline background map that comes with Tableau Desktop.

**To use the offline background map:**

- In Tableau Desktop, select **Map > Background Maps > Offline**

**Note:** The offline background map uses map images stored on your machine. You can find these images in the following locations:

- **On Windows:** C:\Program Files\Tableau\<Tableau Version>\Local\Maps
- **On Mac:** /Applications/<Tableau Version>.app/Contents/install/local/maps

There are several actions, however, that require Tableau to retrieve a map image that may not be stored. If the new map image is not stored on your machine, you won’t be able to load the map until you reconnect to the online map that comes with Tableau.
You may need to reconnect to the online map if you would like to do one or more of the following:

- **Turn layers on or off** - if you decide to turn on a layer that isn’t stored in the cache, Tableau will need to connect to retrieve the necessary information.

- **Zoom in or out** - zooming in or out on a map requires different map images. If the images at the specified zoom level don’t exist in the cache, Tableau will need to retrieve the updated maps.

- **Pan** - panning sometimes requires new map images. If you are working offline and don’t have the necessary map images and legends stored in the cache, the new images and legends will not load.

**To reconnect to the Tableau online map:**

- On Tableau Desktop, select **Map > Background Maps > Tableau**

**About the Tableau background map**

The Tableau background map includes updated map and demographic data, as well as stylistic improvements. It has taken the place of the Tableau classic background map. Any map view created with an earlier version of Tableau Desktop, and with a connection to the Tableau classic background map, will automatically connect to the Tableau background map when opened in Tableau Desktop version 9.0 or later.

**Note:** If you create a map view with data layers in Tableau Desktop version 8.3 or earlier using the Tableau classic background map, and then open that view in Tableau Desktop 9.0 or later, the color of the data layers may appear differently due to data layering and color ramp (color palette) improvements made to the Tableau background map.

**Use Mapbox Maps**

If you have access to Mapbox maps, you can add them to your workbooks or use them to create map views in Tableau Desktop.

When you publish a view that uses Mapbox maps to Tableau Server, Tableau Online, or Tableau Public, your audience can view your data and your Mapbox map without having a Mapbox account.
Add a Mapbox map to your workbook

In Tableau Desktop, you can add a Mapbox map to your workbook and use it as a background map.

After you add a Mapbox map to your workbook, the map is saved with the workbook and available to anyone with whom you share the workbook. You can also save a Mapbox map as a Tableau Map Source (.tms) file that you can share with others so they can quickly connect to it and use it in their own workbooks. For more information, see Save a Map Source on page 2084.

1. In Tableau Desktop, select Map > Background Maps > Map Services.
2. In the Map Services dialog box, select Add > Mapbox Map.
3. In the Add Mapbox Map dialog box, you can add a Mapbox GL map or a classic Mapbox map. To do so, you need the following from your Mapbox account:
   - API access token
   - Username
   - One or more map IDs
   - Style URL (Mapbox GL maps Only)

For more information about any of the above items, see the Access Tokens, Maps, and Styles sections of the Mapbox API Help.

Add a Mapbox GL map

To add a Mapbox GL map, click Mapbox GL in the Add Mapbox Map dialog box, and then do the following:

Note: Mapbox GL is selected by default.

- For Style name, enter a name for the Mapbox map. This name can be anything you want, and will appear in the Background maps menu after you add the map.
- For Url, enter a style URL for the Mapbox map you want to add.
  This URL contains the style ID for your Mapbox map, your access token, and your username. It might look similar to the following:
When you add the correct style URL to this field, the **API access token**, **Username**, and **Layer ID** fields automatically populate.

If you don't have a style URL for the Mapbox map, you must enter a Mapbox access token, username, and layer (map) ID to add your Mapbox map.

**Add a classic Mapbox map**

To add a classic Mapbox map, click **Classic** in the Add Mapbox Map dialog box, and then do the following:

- For **Style name**, enter a name for the Mapbox map. This name can be anything you want, and will appear in the Background maps menu after you add the map.

- For **API access token**, enter the API access token for the Mapbox map you want
to add.

![Add Mapbox Map dialog](image)

- Choose a Mapbox preset style, or add one or more custom map layers:

  **To use a Mapbox preset style:**

  Click **Mapbox preset style**, and then select a style from the drop-down menu.

  **To add one or more map layers:**

  Click **Custom**, and then enter one or more map IDs in the space indicated.
4. When finished, click **OK** to exit the Add Mapbox Map dialog box, and then click **Close** to return to the view.

**Create a map view using a Mapbox map**

To create a map view using a Mapbox map, select **Map > Background Maps**, and then select the Mapbox map you want to use. Next, build the map view. For information, see [Mapping Concepts in Tableau](#) on page 1890.

![Mapbox Map View](image)

**Add or subtract map layers from the view**

If you chose to add one or more custom layers by map ID when you connected to your Mapbox map in Tableau, you can add or subtract those layers from the view using the **Map Layers** pane. For more information, see [Add or remove map layers](#) on page 2050 in the Customize How Your Map Looks topic.

By default, all map layers appear in the view when you first add a Mapbox map to a workbook.

**Note:** When you use a Mapbox preset style, you can’t toggle the layers within the map.

For example, if you use the Mapbox Streets map, you can’t toggle streets, labels,
building footprints, or administrative boundaries like you can with the Tableau map service. This is because Tableau only receives the image tiles that make up the Mapbox map.

Map layers are different from data layers in Tableau. Data Layers are pre-built tiles that include demographic information by various levels, like State, County, and Block Group. Data layers and Mapbox are independent of one another in Tableau. For more information about how data layers work, see Customize How Your Map Looks on page 2047.

See also:

- Use Web Map Service (WMS) Servers below
- Save a Map Source on page 2084
- Import a Map Source on page 2085
- Select Background Maps on page 2071
- Customize How Your Map Looks on page 2047

Use Web Map Service (WMS) Servers

In Tableau Desktop, you can connect to map servers with the Web Map Service (WMS) protocol. WMS is a standard protocol for requesting and receiving geographically referenced imagery.

You can connect to any WMS server that supports the WMS 1.0.0, 1.1.0, or 1.1.1 standards.

In this article

- Connect to a WMS server in Tableau Desktop
- Use a WMS background map to build a map view in Tableau
- Save a WMS server as a Tableau map source
- Performance considerations for WMS servers in Tableau
- Supported Spatial Reference Systems for WMS servers
Connect to a WMS server

1. In Tableau Desktop, select **Map > Background Maps > Map Services**.
2. In the Map Services dialog box, click **Add > WMS Servers**.
3. In the Add WMS Servers dialog box, type the URL for the server you want to connect to in Tableau, and then click **OK**.

You can add as many map servers as you want to a workbook. Each WMS server you add appears as a background map in the **Background Maps** menu.

**Save a WMS server as a Tableau map source**

After you add a WMS server to your workbook, it is saved with the workbook and available to anyone you share the workbook with. You can also save a WMS server as a Tableau Map Source (tms) file, which you can share with others so they can quickly connect to it and use it in their own workbooks.

For more information about how to save a map as a Tableau map source, see **Save a Map Source** on page 2084.

**Use a WMS background map**

After you connect to a WMS server, you can create a map view using the WMS background map that Tableau creates.

**To create a map view using a WMS server:**
1. In Tableau Desktop, select **Map > Background Maps**, and then select a WMS background map to use in the view.

![Map](image)

2. Add a geographic field to the view.

   For more information, see **Mapping Concepts in Tableau** on page 1890.

![Map with Geographic Field](image)

3. Select **Map > Map Layers**, and then select the map layers you want to show in the view.

   For more information, see **Customize How Your Map Looks** on page 2047.
Performance considerations

The content, speed, and performance of a WMS server is reliant on the network and WMS provider. If your Internet connection is interrupted or if you're working behind a restrictive firewall, you can use the Offline background map that comes installed with Tableau Desktop to avoid any performance issues while you build your map view. You can switch back to your WMS server at any time.

**Note:** With the offline background map, you might only be able to access up to four levels of zoom. For more information, see [Select Background Maps](#) on page 2071.

Supported Spatial Reference Systems

Tableau automatically supports the following list of Spatial Reference Systems (SRS) / European Petroleum Survey Group (EPSG) codes. WMS servers must support at least one of the following spatial reference systems to be compatible with Tableau mapping features.

**Supported SRS/EPDG Codes in Tableau:**

- 3857 - WGS 84 / Pseudo-Mercator [DEFAULT]
- 4326 - WGS 84
- 4269 - NAD83
- 3824 - TWD97
- 3889 - IGRS
- 4019 - Unknown datum based on the GRS 80 ellipsoid
4023 - MOLDREF99
4030 - Unknown datum based upon the WGS 84 ellipsoid
4031 - Unknown datum based upon the GEM 10C ellipsoid
4046 - RGRDC 2005
4075 - SREF98
4081 - REGCAN95
4126 - LKS94 (ETRS89)
4130 - Moznet
4140 - NAD83 (CSRS98)
4148 - Hartebeesthoek94
4151 - CHTRF95
4152 - NAD83 (HARN)
4163 - Yemen NGN96
4166 - Korean 1995
4167 - NZGD2000
4170 - SIRGAS 1995
4171 - RGF93
4172 - POSGAR
4173 - IRENET95
4176 - Australian Antarctic
4180 - EST97
4189 - REGVEN
4190 - POSGAR 98
4258 - ETRS89
4283 - GDA94
4319 - KUDAMS
4612 - JGD2000
4617 - NAD83(CSRS)
4619 - SWEREF99
4627 - RGR92
4640 - RRAF 1991
4645 - RGNC 1991
4659 - ISN93
4661 - LKS92
4667 - IKBD-92
4669 - LKS94
4670 - IGM95
4674 - SIRGAS 2000
4686 - MAGNA-SIRGAS
4687 - RGPF
4693 - Nakhl-e Ghanem
4694 - POSGAR 94
4702 - Mauritania 1999
4737 - Korea 2000
4742 - GDM2000
4747 - GR96
4749 - RGNC91-93
4755 - DGN95
4756 - VN-2000
4757 - SVY21
4758 - JAD2001
4759 - NAD83(NSRS2007)
4761 - HTRS96
4762 - BDA2000
See also:

**Use Mapbox Maps** on page 2073

**Save a Map Source** below

**Import a Map Source** on the next page

**Select Background Maps** on page 2071

**Customize How Your Map Looks** on page 2047

**Save a Map Source**

After you add a Mapbox map or WMS server to your workbook, you can save it as a Tableau map source (.tms).

Saving a Mapbox map or WMS server as a Tableau Map Source allows you to share your map with others so they can quickly import it into their own workbooks and use it to create new map views. For more information about importing a Tableau Map Source into a workbook, see **Import a Map Source** on the next page.

1. Select **Map > Background Maps > Map Services**. This opens the Map Services dialog box.

2. Select the map that you want to save as a Tableau map source, and then click **Export**. This opens the Export Connection dialog box.

3. Type a name for the file, choose a location, and then click **Save**.

The Tableau map source includes any default map layer settings you have specified in the workbook. For example, the map source will include any custom set of map layers you have specified to show by default. For more information, see **Customize How Your Map Looks** on page 2047 in the Set Map Layers topic.
Note: If you change the default settings for the map layer options, you should export the map again to include the new settings in the Tableau map source file.

See also:

Select Background Maps on page 2071

Import a Map Source

You can import a Tableau Map Source (.tms) that someone has shared with you into a workbook, and then use it to create custom map views.

1. Select Map > Background Maps > Map Services. This opens the Map Services dialog box.

2. Click Import. This opens the Import Connection dialog box.

3. Navigate to the saved Tableau map source file (.tms) that you want to import, select it, and then click Open. This closes the Import Connection dialog box and adds the Tableau map source to the list of maps in the Map Services dialog box.

   If you want to edit the newly added Tableau map source, select it from the list, and then click Edit. Otherwise, click Close to return to the view.

You can import as many Tableau map sources as you want into a workbook. Each Tableau map source you add appears as a background map in the Background Maps menu and is automatically selected as the default background map until another Tableau map source is imported, until you connect to a WMS server or Mapbox map, or until you select to use a different background map.

Note: When you create a new map view, the background map that is selected in the Background Maps menu is used to create the map for that view.

See also:

Save a Map Source on the previous page

Select Background Maps on page 2071
Explore Data in Maps

Tableau maps can help you quickly find locations and analyze data worldwide. There are many ways you can explore and interact with map views. You can zoom in and out, pan, and select marks with the view toolbar, and even search for locations worldwide with map search.

If you don't see the view toolbar or the map search icon, or if you can't pan or zoom, it's probably because the workbook author has chosen to customize how others can interact with the view.

The workbook author can choose to hide the view toolbar and the map search icon so they do not appear in the view. Similarly, the workbook author can choose to turn off pan and zoom to control how you interact with the view.

For more information, see Customize How People Interact with your Map.

Select marks, pan, and zoom

You can use the view toolbar in the upper left corner of the view to select marks, pan, and zoom in and out of a map view. For more information, see Select Marks and Pan and Zoom.
Search for locations

You can use the map search icon 🌍 to search for locations in your map view. For more information, see Set Map Search Options.

Find your current location

If you’re exploring a map view on Tableau Server or Tableau Online, you can use the locate me button to quickly pan and zoom to your current location. To do so, click the map search icon in the top-left corner of the view, and then, next to the search box that appears, click the locate me button.

**Note:** Finding your current location is not supported on Tableau Desktop.
Note: When you click the locate me button, you might be prompted by your web browser to allow Tableau access to find your current location. If you block access, Tableau will be unable to zoom to your current location.

Return to the initial view

After zooming in and out of the view, or panning to a new location, you can return to the initial view of your map. To do so:

On Tableau Desktop, click the reset axes button on the view toolbar.

On Tableau Server or Tableau Online, click the zoom home button on the view toolbar.

Search for Locations in Your Map

Map search helps you find locations in a map view so you can quickly explore and inspect data.

The map search icon appears in the top left corner of the view. When you click the search icon, a search box appears.
When you begin to type in the search box, map search suggests possible locations that are in your map view. The suggestions are based on location names and text in your data source.

You can search for the following location types:

- Continent
- Country
- State or province
- County
- City
- Postcode

Select a location from the list of suggestions to pan and zoom to that location on the map.

Hide Map Search

By default, the map search icon appears in the top left corner of the view. When you publish a view to Tableau Server or Tableau Online, or share the view through Tableau Reader, the map search icon remains in the view. If you do not want viewers to search for locations in your map view, you can hide the map search icon.

For more information about showing and hiding map search, see Customize How People Interact with your Map on page 2069.

Measure Distances Between Data Points and Locations in a Map

While exploring data in a map view, you might have questions about how that data relates to its surrounding geography, locations, or landmarks. To answer these types of questions, you can use the Radial tool to measure approximate distances in your map view.

The following view shows the number of earthquakes of magnitude 6.0 and higher that have occurred around the world between January, 1994 and February, 2014.
At a global scale, this view is very powerful, but you or your audience might want to zoom in and explore certain areas more closely. For example, maybe you want to see how many earthquakes have occurred in Taiwan in the past 10 years, specifically within 100 miles of its capital, Taipei.

To do so, you can use the Radial tool to find all the earthquakes within approximately 100 miles of the capital city.

Follow the steps below to learn how to measure distance with the Radial tool.

**Step 1: Zoom in to an area or location**

The first step to measuring distance in maps with the Radial tool is to zoom in to an area or location in the map view. For more information about how to zoom in and out of the view, see [Pan and Zoom](#) on page 2615.

You can also use map search to quickly navigate to a location in your map. For more information about how to use map search, see [Search for Locations in Your Map](#) on page 2088.

**Note:** You must zoom in to the map several times before you can measure distance with the Radial tool. The measured distance will not appear if you are zoomed too far out of the map. For more information, see the [Measurement accuracy](#) on page 2093 section.
Step 2: Select the Radial tool

After you have zoomed in to a particular area or location in your view, select the Radial tool on the view toolbar, and then click and drag across the view. The measured distance appears to the right of the circle that appears when you drag across the view.

For more information about how to use the Radial tool, see Select Marks on page 2612.

**Note:** If you do not see a measured distance, you must zoom in further to a location or area in the view.
Note: If the view toolbar is hidden, press S on your keyboard to use the Radial tool.

As you drag, the Radial tool selects marks that are located within the radius of the circle. In this example, the radius is 100 miles, and it is centered over Taipei. This means that, according to this data, all of the selected earthquakes (25) have occurred within approximately 100 miles of Taipei in the past 10 years.
Measurement accuracy

By default, Radial tool measurements have a small margin of error because, in the map projection, distances become exaggerated and stretch as you move away from the equator. This means that the Radial tool can only measure approximate distances.

The Radial tool can measure distance more accurately the closer you are to the equator, and the further you zoom in to the view.

**Note:** By design, the Radial tool does not display a measured distance when you are zoomed too far out of the view, because the measurement might be inaccurate.

Change the units of measurement

By default, your workbook locale determines which units the Radial tool uses to measure distance. If your workbook locale is set to a country that uses the Imperial system, the Radial
tool measures distance in feet and miles. If your workbook locale is set to a country that uses the Metric system, the Radial tool measures distance in meters and kilometers.

You can change the units the Radial tool uses to measure distance for any map view in your workbook. To do so, select Map > Map Options. Next, under Units, click the drop-down menu and choose from the following:

- If you want to measure distances in meters and kilometers, select Metric.
- If you want to measure distances in feet and miles, select U.S.
- If you want the units of measurement to be determined by your workbook locale, select Automatic.

The units you choose will be saved with the workbook and will remain if you publish the map view to Tableau Server, Tableau Online, or Tableau Public.

See also:
Show a map scale on page 2069

Map Storing

When you create map views using the online map provider, Tableau stores the images that make up the map in a cache. That way, as you continue your analysis you don’t have to wait for the maps to be retrieved. In addition, by storing the maps, you can do a certain amount of work when you are offline. For more information, see Select Background Maps on page 2071.

The cache for the maps are stored with your temporary internet files and can be cleared at any time by deleting the temporary files from your browser.

Stored map images and legends remain valid for about thirty days. After that time, Tableau will not use the stored image; instead, it will require you to reconnect and fetch an updated map. This is to prevent the map images from becoming outdated.

Use Background Images in Your Views

Background images are images that you display underneath your data in order to add more context to the marks in the view. A common use of background images is adding custom map images that correspond to a coordinate system in your data.
For example, you might have data that corresponds to several floors in a building. You can use background images to overlay that data on the actual floor plan of the building to give more context. Other examples of using background images include showing a model of the sea floor, images of web pages for analyzing web logs, and even levels from video games to visualize player statistics.

While Tableau allows you to load dynamic maps from the online and offline provider, background images allow you to use your own custom images whether they are special maps or any other image that corresponds to your data.

In this article

Add background images to your workbook below
Build a view with a background image on page 2098
Edit a background image on page 2099
Enable or disable background images on page 2100
Add show and hide conditions to background images on page 2100
Remove a background image on page 2103

Add background images to your workbook

When you add a background image to the view, you need to specify a coordinate system by mapping both the X and Y axes to the values of fields in your database. If you are adding a map, the X and Y axes should be longitude and latitude expressed as a decimal. However, you can map the axes to any relevant fields based on your own coordinate system.

To add a background image:

1. Select **Map > Background Images** and then select a data source.
2. In the Background Images dialog box, click **Add Image**.

![Background Images dialog box](image)

3. In the Add Background Image dialog box do the following:
   - Type a name for the image into the **Name** text box.
   - Click **Browse** to navigate to and select the image you want to add to the background. You can also type a URL to link to an image hosted online.
   - Select the field to map to the x-axis of the image and specify the left and right values. When adding a map, the longitude values should be mapped to the x-axis using decimal values (instead of degrees/minutes/seconds or N/S/E/W).
   - Select the field to map to the y-axis of the image and specify the top and bottom values. When adding a map, the latitude values should be mapped to the y-axis using decimal values (instead of degrees/minutes/seconds or N/S/E/W).
   - You can adjust the intensity of the image using the Washout slider. The farther the
slider moves to the right, the more faded the image will appear behind your data.

4. You can specify the following options using the **Options** tab:

   - **Lock Aspect Ratio** - select this option to maintain the original dimensions of the image for any manipulations of the axes. Deselecting this option allows the image’s shape to be distorted.

   - **Always Show Entire Image** - select this option to avoid cropping the image when the data encompasses only a portion of the image. If you lock both the axis in a view, this option may be negated.

   - Add conditions for when to show the image. Refer to **Filter Data from Your**
Views on page 1162 to learn more about defining conditions.

5. Click **OK**.

When you add the x and y fields to the Rows and Columns shelf in the view, the background image displays behind the data. If the background image does not display, make sure that you are using the disaggregated measures for the x and y fields. To disaggregate all measures, select **Analysis > Aggregate Measures**. To change each measure individually, right-click the field on the shelf and select **Dimension**. Finally, if you’ve used the generated Latitude and Longitude fields for the x and y fields, you’ll need to disable the built in maps before your background image will display. Select **Map > Background Maps > None** to disable the built-in maps.

In order to make the marks in a view more visible when placed on top of a background image, each mark is surrounded by a solid contrasting color called a halo. You can turn mark halos off by selecting **Format > Show Mark Halos**.

**Build a view with a background image**

After you add a background image, you need to build the view in a way that matches the x and y mappings you specified for the image. That is, the fields you specified as x and y must be on the
proper shelves. Follow the steps below to set up the view correctly:

1. Place the field mapped to the x-axis on the **Columns** shelf.
   If you are working with maps, the longitude field should be on the columns shelf. It may seem backward at first, however, the fields on the columns shelf determine the values distributed across the x-axis.

2. Place the field mapped to the y-axis on the **Rows** shelf.
   If you are working with maps, the latitude field should be on the rows shelf. It may seem backward at first, however, the fields on the rows shelf determine the values distributed across the y-axis.

**Edit a background image**

After adding a background image, you can always go back and edit the x and y field mappings as well as any of the options on the Options tab.

To edit an image:

1. Select **Map > Background Images**.

2. In the Background Images dialog box, select the image you want to edit and click **Edit** (you can also just double-click the image name).

3. In the Edit Background Image dialog box, make the changes to the image and click **OK**.
Enable or disable background images

Although you can add multiple images to a workbook, you may want to only use a subset of the images for a particular set of worksheets. For example, you may want to show a map of the entire United States of America on one view, and maps of individual states in other views.

Use the check boxes in the Background Images dialog box to enable and disable the images for the current worksheet. You can show several images by enabling multiple images on a single worksheet. For example, you may have several images that you want to tile in the background to make a larger background image.

To enable or disable a background image:

1. Select **Map > Background Images**.
2. In the Background Images dialog box, select the check boxes next to the images you want enabled.
3. Click **OK**.

Add show and hide conditions to background images

When you add a background image and enable it, the image will be shown automatically on any worksheet that has the required fields used in the view. To avoid showing an image on all the worksheets, you can specify show/hide conditions. Show/Hide conditions are conditional
statements that you define to specify when to show the image. For example, you may have a
floor plan image for a multi-story building. While each image is associated with the same
coordinates (the corners of the building), you do not want to show the first floor map when
looking at the third floor information. In this case, you can specify a condition to only show the
first floor image when the Floor field is equal to one.

**To specify show/hide conditions:**

1. Select **Map > Background Images** and then select a data source.
2. In the Background Images dialog box, select the image you want to add a condition to
   and click **Edit**.
3. In the subsequent dialog box, select the **Options** tab.
4. Click the **Add** button at the bottom of the dialog box.

![Add Background Image dialog box]

5. Select a field to base the condition on. In the example described above, the field is Floor.
6. Specify when you want to show the image by selecting a value of the field. For this example, one is selected.
7. Click **OK**.

   A condition statement is added to the image. In the building floor plan example, the condition statement is Only show the image when Floor is equal to One.

8. Click **OK** twice to close the Background Image dialog boxes and apply the changes.

---

When you add multiple conditions, the background image will only show when all conditions are met. For example, if a background image has two conditions on Property Name and Floor, it will only show when Property is Greenwood Estates and Floor is 3.

---

**Remove a background image**

When you no longer want to use a background image you can either disable it or remove it, making it unavailable to all worksheets.

**To remove an image:**

1. Select **Map > Background Images**.

2. In the Background Images dialog box, select the image you want to remove and click **Remove**.

3. Click **OK**.
See Also:

- **Find Background Image Coordinates** on page 2106
- **Geographic Data Analysis in Tableau** on page 1862

**Add Images of Google Maps and OpenStreetMap as Background Images in Tableau**

You can add images of Google Maps or OpenStreetMap as static background images in Tableau, and then plot data points onto them. To plot data points onto a background image of GoogleMaps or OpenStreetMap, you need both latitude and longitude coordinates of your data.

This topic describes how to obtain latitude and longitude coordinates for Google Maps and OpenStreetMap, as well as how to plot them on a background image in Tableau.

**Step 1: Find your map coordinates**

Follow the steps below to find map coordinates for Google Maps or OpenStreetMaps.

**Option 1: Map data points onto Google Maps**

1. Go to [http://www.gorissen.info/Pierre/maps/googleMapLocation.php](http://www.gorissen.info/Pierre/maps/googleMapLocation.php) and locate the geographic area that you want to plot data points.

2. Once you have located the area you want, hover over the easternmost side of the area and write down the longitude coordinate (displayed in the bottom-right corner of the map). Do the same for the westernmost side of the area.

3. Hover over the northernmost part of the area and write down latitude coordinate. Do the same for the southernmost part of the area.

You should now have two latitude coordinates (one top and one bottom) and two longitude coordinates (one east and one west).
You will use these coordinates to define the scale of your background image in Tableau, and to ensure that you can plot data points onto the image using latitude and longitude coordinates.

4. Take a screenshot of your map and save it to your computer.

5. Follow the steps in "Step 2: Add your map image as a background image below" to add the image as a background image in Tableau.

Option 2: Map data points onto a map in OpenStreetMap

1. Go to http://www.openstreetmap.org and zoom in on the geographic area you want to plot data points.

2. On the top menu bar, click Export.

   The Export pane provides the latitude and longitude coordinates of the area in the view. You can edit these coordinates to select a different area.

3. Write down the coordinates displayed in the box.

4. On the right side of the page, click the share icon, and then click the Download button.

5. Follow the steps listed in "Step 2: Add your map image as a background image below" to add the image as a background image in Tableau.

Step 2: Add your map image as a background image

1. Open Tableau Desktop and connect to the data you want to plot on the map image.

2. Select Maps > Background Images, and select the data source you want to use to plot data on your background image.

3. In the Background Images dialog box, click Add Image.

4. In the Add Background Image dialog box, click Browse, and then navigate to the location of the map image you created earlier in Step 1: Find your map coordinates.

5. In the Add Background Image dialog box, do the following:
   - For the X Field drop-down list, select Longitude (generated).
   - In the Left box, type the easternmost longitude in your map image.
   - In the Right box, type the westernmost longitude in your map image.
   - For the Y Field drop-down list, select Latitude (generated).
• In the **Bottom** box, type the southernmost latitude in your map image.

• In the **Top** box, type the northernmost latitude in your map image.

6. When finished, click **OK** in both the Add Background Image and Background Images dialog boxes.

7. Select **Maps > Background Maps > None**.

   The background image shows up only if there is no background map.

8. From the Measures pane, drag **Latitude (generated)** to the **Rows** shelf and **Longitude (generated)** to the **Columns** shelf.

   Your background image should appear in the view. If it does not, or if it appears at the wrong scale, complete the following steps to fix your axes.

9. In the view, double-click the Y axis.

10. In the Edit Axis dialog box, under Range, select **Fixed**.

11. In the **Fixed start** text box, enter the bottom most longitude in your map image.

12. In the **Fixed end** text box, enter the top most longitude in your map image.

13. Click **OK**.

14. Repeat steps 9 - 13 for the other axis.

To learn how to plot your data on the background image, see steps 1, 2, and 4 in **Find Background Image Coordinates** below.

See also:

**Use Background Images in Your Views** on page 2094

**Find Background Image Coordinates** below

**Use Mapbox Maps** on page 2073

**Use Web Map Service (WMS) Servers** on page 2078

**Select Background Maps** on page 2071

**Find Background Image Coordinates**

In many scenarios, you might want to map your data onto a background image instead of on a Tableau map. This article explains how to plot data points on a background image.
There are several steps to this process:

1. Create a table in your data source for X and Y coordinates.
2. Connect to your data source in Tableau Desktop and join the coordinate table with the rest of your data.
3. Import your background image and build the view.
4. Annotate points on the background image.
5. Add coordinates to the coordinates table in your data source.

: Create the coordinate table

Create a table in your data source that contains columns to identify a unique mark. In this example, there are three columns:

- The first column holds the unique identifier for each mark.
- The other two columns are for the X and Y coordinates.

After you create your columns, you join the new coordinate table to the original data source. The join is based on the unique identifier used in both the original data source and the coordinate table (the identifier that represents a single mark).

1. Select a scale that is appropriate for your background image. If the image is wider than it is tall, use X: 0-100 and Y: 0-50.
2. For this example, open the Sample Superstore Excel file from your Tableau repository in Excel and select a new sheet tab.
3. On the new tab, put a single row in a table that contains one of the unique identifiers and the X and Y end points you selected for your two scales.

In this case, **Product Sub-category** is a column in Sample Superstore, and one of its members is **Tables**. The X endpoint is **100**, and the Y endpoint is **50** because that is the scale for the background image.
4. Change the tab name to **Coords** and save the file.

**Connect to the coordinate table**

1. In a new Tableau workbook, select **Connect to Data**.
2. In the **Connect to Data** dialog box, select **Microsoft Excel** and then click **Next**.
3. On the data source page, do the following:
   - Drag the Orders sheet to the Join area.
   - Drag the Coords sheet to the Join area.
   - Click the join icon to modify the existing join.
   - In the Join dialog box that opens, select **Left**, and then select **Product Sub-Category** for the **Data Source** column, and **Product Name (Coords)** for the **Coords** column.

4. Close the Join dialog box, and then select the sheet tab to go to a worksheet.
Import the background image

1. In the new worksheet, select Maps > Background Images, Sample - Superstore.
   This opens the Background Images dialog box.

2. In the Background Images dialog box, click Add Image.

3. In the Add Background Image dialog box, browse to and select the file.

4. For X Field, select X in the list.

   **Note**: If you do not see the X field in the list, it means that you need to change the
   X field to be a continuous number.

   To change the field to be a number, right-click the X field in the Data pane, and
   select Change Data Type > Number (Whole). To convert the field to be
   continuous, right-click the X field in the Data pane, and select Convert to
   Continuous.

   Repeat these steps for the Y field.

5. In the Right text box, type 100 (100 is the number you used in the coordinate table).

6. For Y Field, select Y in the list.

7. In the Top text box, type 50 (50 is the number you used in the coordinate table). Be sure
   to type in the correct text box.
8. Click OK, and then click OK in the **Background Images** dialog box. (Disregard the **Valid** setting of No in the **Background Images** dialog box.)

Find the coordinates

1. From the Data pane, drag X to the **Columns** shelf and Y to the **Rows** shelf.
2. On the **Columns** shelf, right-click X and select **Dimension**.
3. On the **Rows** shelf, right-click Y and select **Dimension**.

**Note:** This step is only necessary if the X and Y fields were brought in to Tableau as Measures.

The background image appears in the view.
4. Right-click the X axis and select **Edit Axis**.

![Edit Axis dialog box]

5. In the **Edit Axis** dialog box, select **Fixed**, and then in the **Start** text box, type **0**, and in the **End** text box, type **100**.

6. Click **OK**.

7. Right-click the Y axis and select **Edit Axis**.
8. In the Edit Axis dialog box, select Fixed, and then in the Start text box, type 0, and in the End text box, type 50.

9. Click OK.

Now the axes start at zero.

10. Right-click anywhere on the image where you want to find the coordinates, and select Annotate > Point.

For more information about annotating marks or points in the view, see the Add Annotations on page 1763 topic in the Tableau Help.
11. In the **Edit Annotation** dialog box, click **OK**.

12. Repeat steps 10 and 11 for each point you want to annotate.

The coordinates appear on the background image as a callout. You can move a callout to a location where it shows up better; just click and drag it. You can also resize the callout by dragging one of the size handles. You can move a coordinate point to a new location by clicking and dragging the arrowhead; the coordinates in the callout change to reflect the new location.
13. Add these coordinates to the Coords sheet in the Sample - Superstore Excel file and save the changes.
In Tableau, select **Data > Sample - Superstore > Refresh**.

The refreshed view displays marks at the coordinate locations.

If you don't want to continue to display the callouts, you can select and remove them on the background image.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Product Sub-category</td>
<td>X</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Tables</td>
<td>100</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Labels</td>
<td>17.16</td>
<td>41.42</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Paper</td>
<td>9.00</td>
<td>29.81</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Bookcases</td>
<td>23.95</td>
<td>23.21</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Office machines</td>
<td>32.27</td>
<td>22.77</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Other Ideas

Try placing another Measure on the Size shelf and put either a Dimension or Measure on Color on the Marks card. Add your own custom shapes. Develop powerful and unique analyses by creating Actions between this view and your other analyses.

If you want to hide the X and Y scales, right-click the respective field on the Columns or Rows shelf and clear Show Header.
Build Data Views for Accessibility

If you want to make your views accessible to as many people as possible, or if you work in an environment that is subject to US Section 508 requirements, other accessibility-related laws and regulations, you can use Tableau to build data views that conform to the Web Content Accessibility guidelines (WCAG 2.0 AA). This includes building views that are accessible to users who use screen readers, braille keyboards, keyboard-only navigation, and so on.

View controls that support accessibility

To create an accessible view, use Tableau Desktop to create a view that includes WCAG-conformant elements, and then publish and embed that view in a web page that conforms to Web Content Accessibility guidelines (WCAG 2.0 AA).

The example below shows a view with a scatter plot chart and includes various elements that you can use to be WCAG conformant. For more details, see Best Practices for Designing Accessible Views on page 2119 and Author Views for Accessibility on page 2127.
Note: The color blind palette can help you select colors that can be recognized by users with visual impairments. When you assign colors to different dimension values make sure that they provide enough contrast and aren't too close to each other on the light-dark spectrum.

Additional support for accessible views

In addition to the controls shown in the illustration, Tableau also supports the following features to help you create accessible views:

- Keyboard navigation
- Programmatic context for assistive technologies (using ARIA roles)
- A text equivalent for charts and visualizations
- Compliance with contrast standards
- Authentication when signing in to Tableau Server for embedded views

Note: If you use a data source that requires authentication to access it, that authentication page does not conform to the WCAG 2.0 AA guidelines. To prevent the
authentication page from showing, you can embed the credentials for the data source when you publish. For information about how to embed credentials for the data source, see Set Credentials for Accessing Your Published Data on page 2530, "Set the authentication type", Embedded password section.

Additional resources

The following external resources can help you when designing your views for accessibility.

- Web Content Accessibility Guidelines (WCAG 2.0)
- Color Contrast Tips and Tools
- Color Contrast Analyzer

Disclaimer: Although we make every effort to ensure these links to external websites are accurate, up to date, and relevant, Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the external site for answers to questions regarding its content.

For information, guidelines and examples about how to create accessible views, refer to the following topics in this section.

Best Practices for Designing Accessible Views

You created a great view, and you want to make sure that all of your users can see and understand the data that you've put together- but some users have visual or physical impairments. So what can you do?

Tableau supports several controls to enable you to build a view that complies with US Section 508 requirements when embedded in a web page that meets the Web Content Accessibility guidelines (WCAG 2.0 AA). In addition to following the steps in Author Views for Accessibility on page 2127, make sure that content embedded in your web site also conforms to the following WCAG 2.0 AA principles:

- Perceivable - Information and user interface components must be presented to users in a way that they can perceive. Consider including text alternatives and alternate ways to present the content.
Operable - The user interface components and navigation must be accessible to users from the different devices or methods that they use to interact with the view.

Understandable - The information presented in the view must be understandable to your users. For example, using clearly distinguishable names and labels for different elements shown in your view.

Follow the best practices described in this article and incorporate the steps described in Author Views for Accessibility on page 2127, to build views that are accessible to all of your users when published on Tableau Server or Tableau Online and then embedded in a WCAG 2.0 AA conformant web page.

In this article

- Keep it simple below
- Titles and captions on page 2122
- Additional text on page 2123
- Color and contrast on page 2125
- Publishing your view on page 2126

Keep it simple

WCAG 2.0 AA principle: Understandable

You might have a lot of information that you want to communicate with your view. However, dense views can be difficult to understand or navigate using a screen reader or keyboard. Use the following guidelines to help you communicate everything that you want to without overwhelming your users with an over-packed view.

- Aggregate your data whenever possible to help reduce the number of marks you are showing. Also, showing more than 1000 marks in the view can cause the view to be rendered by the server instead of the browser and server-rendered views are not yet supported for WCAG conformance.

Users can also access the View Data page (enabled by default) to review the underlying data for the marks or can download the data from that page to an accessible application to view it that way.

Example: This example shows two different bar chart views to illustrate the difference between a detailed and aggregated view.
Not easily accessible - Too many marks

- This much detail makes the view harder to understand.
- The view is showing over 5000 marks and does not include enough text to indicate what the different marks represent.
- It is too difficult to use for users who need screen readers.

More accessible - Aggregated view

- This example shows the same view with the data aggregated at a higher level.
- Key data points are still included, but they are now easier for users to read and understand.
- The number of marks is reduced from over 5000 to about 20.
- Users can still read the underlying details for the marks by placing focus on the view and then pressing Enter to open the View Data page.

- Consider using simple graphic elements like bar charts or line charts that allow you to use text, color, and shapes to add additional context to the view.
- Limit the number of marks to only those that emphasize the most important data points.
For information and examples about how to build this type of view, see Keep it simple on page 2128 in Author Views for Accessibility on page 2127.

Titles and captions

**WCAG 2.0 AA principle:** Perceivable, Understandable

Providing good descriptive text in titles and captions provides context to users who are using assistive technology, and can help them understand the data in your view. Use the following guidelines to adapt the visual nature of Tableau to meet the needs of all of your users.

- Think about your views as a supplement to the text you use to describe it.
- Use text in titles and captions to describe your visualizations and what you are showing.
- Use simple, easy-to-understand language. Avoid jargon, acronyms, or abbreviations.
- Don’t include words like "image of" or "picture of" in your text descriptions, because screen readers sometimes already include this information.
- Avoid using all capital letters (for example in headings or titles), because they can be difficult to read.

**Example:** This example shows two different bar charts. One using very little text and the other using titles and captions to add context

**Not easily accessible - Too little text**

**More accessible - Adding descriptive text to provide context**

- A one-word title is not descriptive enough.
- This example shows the same view, but includes additional
- The marks are differentiated by color and size. But without additional text, the context for these marks can be difficult to understand.

- There is no caption or other explanatory text in both the title and the caption.

- The same contrast color scale and size differentiators are used, but labels are added to the bar marks to give additional context.

For information and examples about how to build a view that includes text for context, see Show more text and make it helpful on page 2139 in Author Views for Accessibility on page 2127.

Additional text

**WCAG 2.0 AA principle:** Perceivable, Understandable

Using text beyond just titles and captions throughout your view can help users understand the context of the different elements that you are showing, as well as help describe the relationship between the different controls (such as legends and filters) and your data.

Use the following guidelines when adding additional text:

- Use text in headings on legends or filters to describe the control and what it does. You can also use text zones on a dashboard to add additional context for your visualizations to further describe what you are showing.

- Refer to controls by label whenever possible. For example changing the label for a legend from **Subcategory** to **Color key for product type** can help users understand the relationship between the controls and the data.

- If you include link text in your view, use text that describes where the link will take the user. For example, use link text like "Global Warming statistics for 1990-2000". Avoid wording like "Click here", "More", or "More information". These link text examples are too generic and can be confusing to users.

- Consider using Natural Language Generation (NLG) tools to help produce data-driven, textual narratives for visualizations. For example **Wordsmith** from Automated Insights, **Narratives for Tableau** from NarrativeScience, or **Savvy** from Yseop.
Example: This example shows two bubble charts. One that uses only the text that shows by default when creating a view and one that adds context-specific text throughout the view to help convey meaning in the view.

Not easily accessible - Too little text

More accessible - More descriptive text

- This view uses the default text only for the sheet title and the default labels for the filters and legend.
- The marks are differentiated by size and color only.
- There is no caption or other explanatory text included in the view to help provide context.

- This example shows the same view, but includes additional explanatory text.
- Additional text was added to the title and caption to explain the relationship of the marks and provide additional context about what the view is showing.
- Mark labels are added to show profit numbers so that users don’t have to rely on the color only to understand this information.
For information and examples about how to build a view that includes additional text for context, see Show more text and make it helpful on page 2139 in Author Views for Accessibility on page 2127.

Color and contrast

**WCAG 2.0 AA principle:** Perceivable, Understandable

You can use color to help distinguish marks in your view. However, for users with visual impairments, using color alone doesn’t always provide enough of a distinction, especially when there are many marks in a view. Use the following guidelines when using color in your views:

- Tableau provides a color-blind palette that you can use to help you select appropriate colors for your view. For any color palette that you use, try to provide enough contrast and assign colors that differ from each other on the light-dark spectrum.
- For line marks, use additional options like shapes, size and labels to help distinguish them.
- Use contrast analyzer tools to help select the best text colors and backgrounds with sufficient contrast ratios. Make sure that color contrast for text is strong and meets the contrast ratio standards of 4.5:1 (3:1 for large text).

**Example:** This example shows two line charts. One that uses color only to differentiate the lines and the other that uses more appropriate colors plus shapes to differentiate marks

- Not easily accessible - Using only color to differentiate marks
- More accessible - Using color and shape to differentiate marks
This view uses only color to distinguish the lines from one another.

The marks in the view are not using a color-palette that is fully accessible to users with visual impairments.

This example shows the same view, but uses both color and shape to identify the marks.

A color legend and a shape legend with clear titles identify what the colors and shapes represent in the view.

For information and examples about how to build a view that uses color to help distinguish marks, see Use color thoughtfully and provide contrast on page 2147 in Author Views for Accessibility on the next page.

Publishing your view

**WCAG 2.0 AA principle:** Perceivable, Operable, Understandable

To make your views available to your users, publish your view to Tableau Server or Tableau Online, in the toolbar menu, click **Share** and copy the **Embed Code** and then paste it in your WCAG conformant web page to embed your view and make it available to your users.

**Note:** Tableau does not support WCAG conformance for the Alerts, Custom Views, and Subscriptions buttons on the toolbar, so you must alter the embed code slightly to turn them off before embedding your view on your web page. Tableau does not support WCAG conformance for web editing, so Web Edit permissions must be turned off in the published workbook.
For information about how to publish and embed workbooks, find the embed code to copy into your web pages, and turn off the toolbar, see Publish and embed the view on page 2159 in Author Views for Accessibility below.

**Author Views for Accessibility**

This article provides more detail on the design guidelines discussed in Best Practices for Designing Accessible Views on page 2119, and steps on how to create these views in Tableau Desktop (version 10.2 and later).

In this article

- General steps
- Building the views
- Walkthrough: Make a viz more accessible
- Publish and embed the view

**General steps**

As a content author, you can create WCAG-conformant views following these general steps:

**Step 1:** Author views in Tableau Desktop (version 10.2 and later) following the best practices.

**Step 2:** Publish the views to Tableau Server or Tableau Online (version 10.2 and later).

**Step 3:** Embed the views in a web page for users to access the content.

Most types of views that you create following these best practice guidelines in Tableau Desktop will be WCAG-conformant; they are treated by screen readers as images with a title and a description. Users will be able to use keyboard navigation to open the View Data window and access the underlying data for the view. However, map views in Tableau are not currently supported for WCAG conformance.

**Building the views**

Many of the principles that apply to creating effective views in Tableau also apply to creating WCAG-conformant views. The main difference in creating conformant views is that you should always provide other types of mark encoding when you use color in the view. You will also want to use text more liberally in the view to provide context for the view and its marks.
- Keep it simple
- Show more text and make it helpful
- Use color thoughtfully and provide contrast
- Provide other visual cues: Position, size, and shape

Keep it simple

When creating accessible views, keep it simple. Doing so helps people more easily see the relationships and intersections in your data.

How can you make a view more simple? Emphasize only the most relevant and important data. This means reducing the marks in the view to only what is needed. Limiting the level of detail and granularity in the view also helps to prevent cognitive overload.

When you design your views, keep in mind that your users will be accessing the underlying data in the view. The data you include in the view is what screen readers will read when users access the underlying data in the Summary tab of the View Data window.

Guidelines that support simplicity:

- Limit the number of marks in the view
- Orient your views for legibility
- Limit the number of colors and shapes in a single view
- Use filters to reduce the number of marks in the view at a given time

Limit the number of marks in the view

Keep the number of marks below 1000 to ensure the view will be rendered in the browser and not on the server.

Note: Server-rendered views are not WCAG-conformant; only client-rendered views are WCAG-conformant. To ensure an embedded view is rendered in the browser, make sure the view has less than 1000 marks (including items in legends and filters).

You can see the number marks a view has when you are authoring the view in Tableau Desktop (bottom left corner of the window).

Example of a view with too many marks

The following example shows a view with over 20,000 marks. The view shows a more granular level of detail in the data (versus aggregated). Multiple dimensions (Region, Category, Sub-Category) and measures (Sales, Profit, Profit per Order, and Profit Ratio) have been added to
the view. Product Name is on Detail, which adds a mark for every product included in the data set.

Example of a view with fewer marks

The next example shows a more accessible view (fewer than 100 marks). The data is aggregated and shows a less granular level of detail in the data. Two dimensions (Region and Product Category) and one measure (Sales) are used in the view.

For more details about these concepts, see Data Aggregation in Tableau on page 279 and How dimensions affect the level of detail in the view on page 255.
There are several ways you can limit the number of marks:

- Create views of aggregated data rather than showing more granular levels of detail in the view. Don't place dimensions that contain thousands of values on Detail in the Marks card.

- If you want to use a dimension that is in a hierarchy, but not make the hierarchy available in the view, use a copy of that particular dimension. Copy the field you want to use from the hierarchy and build the view with the copied version of the dimension.

Doing this prevents users from expanding the hierarchy in the embedded view, which might increase the number of marks in the view and result in the view being rendered on the server instead of the browser.

1. Right-click a dimension in the hierarchy, and then click Duplicate. A copy of the dimension will appear under Dimensions.

2. Right-click the copied dimension, and then click Rename. Give the copied dimension a unique, meaningful name. Now use that dimension to build the view.

- Limit the number of measures and dimensions used to create the view.

**Less accessible**: The next example shows a view with four measures on Rows and three dimensions on Columns. More fields create a more complex view.
More accessible: This view has two dimensions on Columns and one measure on Rows. Fewer fields in the view structure reduces cognitive overload.

Orient your views for legibility

Horizontally-oriented labels and headers are easier to read than vertically-oriented labels.

Less accessible: In this example, the text is being displayed vertically.
More accessible: In this example, space has been added to the view to allow the text to be displayed horizontally.

To change the orientation of the view and its headers and labels, you can click the Swap button on the toolbar, or you can resize the view to allow more room for horizontal labels. In this example, the table was resized. For details, see Resize Tables and Cells on page 2406.
Limit the number of colors and shapes in a view

If you are going to show a dimension on Color or Shape, don't use more than 10 colors and shapes in one view so that users can distinguish each color and shape and see important patterns.

Don't do this: In the following example, the line colors have started repeating after the 10th mark applied to Product type.

Do this: In the next example, filters have been used to limit the number of marks that can be shown at a time. As a result, the number of colors and shapes in the view is never more than ten.
Use filters to reduce the number of marks in a view

Using filters helps to focus the number of marks in the view to only what you want users to see. For more details about creating filters, see Filter Data from Your Views.

You can also show the following filter modes to allow users to control what data is in the view. The filter modes that currently are supported for WCAG conformance in Tableau are:

- **Single Value (list)**: a filter with radio buttons. Only one item can be selected at a time. Giving your users single-value filters is a great way to reduce the number of marks in the view.

- **Multiple Values (list)**: a filter with a list of items (with check boxes) that can be selected at the same time. Multiple-value filters will allow more marks to be shown at the same time.
- Single Value (dropdown): a filter with a dropdown list of items. Only one item can be selected at a time. Giving your users single-value filters is a great way to reduce the number of marks in the view.

- Multiple Values (dropdown): a filter with a dropdown list of items that can be selected at the same time. Multiple-value filters will allow more marks to be shown at the same time.
To show a filter, right-click the field you want to use as a filter, and then select Show Filter.

To select the filter mode, in the drop-down menu for a filter, select a Single Value (list), Single Value (dropdown), Multiple Values (list), or Multiple Values (dropdown) filter.
For a Single Value (list) or Single Value (dropdown filter), you can remove the All option from the filter. In the drop-down menu for the filter, select Customize and then clear the Show "All" Value option.

The following image shows the Customize option in a filter.
When you are using multiple filters in a worksheet, keep the following guidance in mind:

- When you are creating a view, test the browser zoom up to 200% to make sure all of the components in the view display as intended.

- Avoid positioning filters horizontally in a worksheet. In instances where the browser window is not wide enough, components in the view (such as filters) will overlap with the visualization.

- Don't reposition the filters after you add them to the view. If you reposition the filters or change their order, the tab order may change.

This example shows a view with a **Single Value (list)** filter for the Product category and a **Multiple Values (list)** filter for Product order date. The **Single Value (list)** filter only allows one product type to be shown at a time (**All** is hidden).
Show more text and make it helpful

In your views, make sure you show titles, captions, and mark labels. You can edit titles and captions to provide more context and details that help users understand the view.

- Be precise and consistent in how you name every object in the view.
- Provide text that explains how components of a visualization relate.
- Simplify text to remove redundancies.

You can also rename measures and dimensions in the Data pane to make sure field names in the view are meaningful.

Here are the different elements in a view that you can show and edit:

- **Titles (worksheet, filters, and legends).** Show titles for the worksheet, filters, and legends. The default title text is the sheet name or field name referenced by the filter or legend. You can edit any title to provide more context.

- **Captions (in worksheets).** Show the caption in the worksheet. The default caption text gives a summary of what is shown the view. You can add to this text to provide more context.
• **Labels (Marks card, Label button).** By default, labels are not shown in a view, so you must to choose the option to show them.

**Less accessible:** This example shows a view with limited text. Many text elements are not shown: the worksheet caption, mark labels, a descriptive title, and additional text about the view in the Title area. Also, the filter and legend titles have not been edited to provide more context.

![Graph showing profit and sales data](image)

**More accessible:** This example shows the same view with multiple text elements shown to provide more detail and context. Titles and captions are shown in the worksheet. The worksheet title was edited to include text that describes how data is indicated in the view. The worksheet caption includes the default summary text that is automatically generated by Tableau. The legend titles have been edited to make more sense in this view. Mark labels show Profit values so that users don't have to rely only on seeing the color.
You also can build views that show only text, such as this example of a Text Table view that shows similar data.

<table>
<thead>
<tr>
<th>Product category</th>
<th>Central</th>
<th>East</th>
<th>South</th>
<th>West</th>
</tr>
</thead>
<tbody>
<tr>
<td>Furniture</td>
<td>$4,157</td>
<td>$4,829</td>
<td>$10,850</td>
<td>$36,034</td>
</tr>
<tr>
<td>Office Supplies</td>
<td>$3,155</td>
<td>$2,571</td>
<td>$17,017</td>
<td>$30,073</td>
</tr>
<tr>
<td>Technology</td>
<td>$2,636</td>
<td>$3,148</td>
<td>$16,978</td>
<td>$30,166</td>
</tr>
</tbody>
</table>

To show and edit titles and captions for the worksheet
1. Select **Worksheet > Show Title**, and then select **Worksheet > Show Caption**.

![Worksheet menu](image)

2. In the Title box at the top of the view, click the drop-down menu, and then click **Edit title**.

![Title editing](image)

Type text in the text area and format it, and then click **OK**. You can add text in addition to markup text, or replace the markup text.

3. In the Caption box, click the drop-down menu, and then click **Edit caption**.
Type text in the text area and format the text, click the drop-down menu, and then click **OK**. You can add text in addition to the default text, or replace the default text.

**To show mark labels in the view**

- Click **Label** in the Marks card, and then click **Show mark labels**.

**To edit field names in the Data pane**
1. Right-click the dimension or measure, and then click Rename.

2. Edit the name, and then press Enter.

For more details, see Rename fields.

To edit filter or legend titles

The text associated with legends and filters should clearly describe their purpose. Edit titles for filters and legends to indicate function. Refer to filters by name in any instructions that you provide in the worksheet title or caption.
1. Click the drop-down menu for the filter, and then click **Edit Title**.

2. Edit the text, and then click **OK**.
Text Size

The default text sizes for elements in a Tableau workbooks are WCAG-conformant. However, if you want to change text size, you can select Format > Workbook to change the default text sizes for the workbook. For more details, see Format Your Work.
Use color thoughtfully and provide contrast

Different color palettes are available depending on the type of field that is being placed on Color in the Marks card. For any color palette that you use, ensure that you are providing enough contrast and assign colors that differ from each other on the light-dark spectrum.

- For dimensions (discrete), a suggested practice is to use the Color Blind palette. You may need to adjust the colors assigned to different dimension values. Try to provide enough contrast and assign colors that differ from each other on the light-dark spectrum. For monochromacy, try using the Seattle Grays palette with Stepped Color set to 5.

- For measures (continuous), try using the Blue or Orange-Blue Diverging palette with
the **Stepped Color** option set to 5. You may need to adjust the number of steps used. For monochromacy, consider using the **Gray** palette with **Stepped Color** set to 5.

For text color, general guidelines are that the visual presentation of text and images of text has a contrast ratio of at least 4.5:1 (large text 3:1). Tableau's default color formatting follows these prescribed contrast ratios. You can use a contrast analyzer tool to test the contrast ratios of text colors and backgrounds.

**Important**: When you use color in the view, you must also provide other types of encoding for marks, such as labels, size, shape, and position. Don't rely on color alone to communicate differences in marks.

**Dimension on Color**

**Less accessible**: This view shows an example of a discrete dimension being placed on Color, which is set to the Tableau 20 palette. The line path provides helpful position information, but the colors do not provide clear distinction between the line marks.

**More accessible**: In the next example, the Color Blind palette is being used. Shape is also being used to reinforce what is being shown in color.
Measure on Color

**Less accessible:** This view shows an example of a continuous measure placed on Color, which is set to the Blue-Teal palette with gradient colors. The view does not provide enough contrast between colors and there is insufficient contrast on the light-dark spectrum. The use of size helps this view, but this color palette can make it hard for people with color vision deficiencies to see contrast differences between marks.
More accessible: This view uses the Orange-Blue Diverging palette with stepped colors. This palette provides improved contrast between marks.

<table>
<thead>
<tr>
<th>Product category</th>
<th>Product type</th>
<th>Central</th>
<th>East</th>
<th>South</th>
<th>West</th>
</tr>
</thead>
<tbody>
<tr>
<td>Furniture</td>
<td>Bookcases</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chairs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Furnishings</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tables</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Office Supplies</td>
<td>Appliances</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Art</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Binders</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Envelopes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fasteners</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Labels</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Paper</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Supplies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technology</td>
<td>Accessories</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Copiers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Machines</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phones</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To select a palette

You must already have a measure or dimension on Color in the view for this option to be available.

1. Click Color in the Marks card, and then click Edit Colors.

2. For Select Color Palette, click the drop-down arrow and select the palette.

   This example shows selecting the Color Blind palette.
For the Color Blind palette, click Assign Palette, and then click OK. Otherwise, click OK.

**To change the color of a value:**

1. Click a value on the left, under Select Data Item.

2. Click a new color in the palette on the right. Hover over a swatch to identify the color. Repeat for as many values that you want to change.

3. Click OK to exit the Edit Colors dialog box.

The following example shows the Orange-Blue Diverging palette with Stepped Color set to 5 steps.
3. For the Color Blind palette, click **Assign Palette**, and then click **OK**. Otherwise, click **OK**.

For details about colors and palettes, see **Color Properties**.

---

**Provide visual cues beyond color: position, size, and shape**

When you use color for field values, you must also provide other types of encoding for marks, such as labels, and position, size, and shape. Use position, size, or shape for the same field values to reinforce the differences between marks. These mechanisms all communicate information about the data to your users.

If you are using color to show measure or dimension values, make sure that you provide enough light-dark spectrum contrast for the colors that you assign to different marks.

**Less accessible**: This example shows the same scatter plot with no use of color, shape, or size to create contrast. The position of the marks provides meaningful information, but the view could make use of more visual cues.
More accessible: This example shows a calculated field named Quadrant being used on color and shape. Position, color, and shape are used to distinguish each mark. The axes also clarify the mark position and meaning.
Putting it all together

This next example shows how text, shape, color, and size can all work together to create an accessible view. Sales are represented by the size of each circle (mark). Profit represented by color and text. The title and caption for the view are shown and have been edited to provide more context. The filter and legend titles have also been edited to provide more context.

Walkthrough: The journey from less accessible to more accessible

Let’s convert a viz that isn’t very accessible into one that is. The data in this starting view has a more granular level of detail and a structure that could contribute to cognitive overload with a screen reader.
This view is designed to show total sales for each sales region, along with the profit ratio, to determine which types of products seem to be selling well but are not actually profitable. Profit and Profit per Order are also shown.

We'll make changes to it based on the concepts and best practices mentioned previously in this article.

**Step 1: Reduce the level of detail in the view.**

Because the Product Name dimension is on Detail, every product in the data set is being represented with a mark in the view.

- Remove Product Name from Detail. This change reduces the number of marks in the view from over 20,000 to 272.

Reducing the number of marks in the view helps to ensure the embedded view will be rendered on the browser instead of the server.
Step 2: Create a more aggregated view of the data.

1. Reduce the number of measures on Columns and the number of dimensions on Rows.
2. Remove Region (we can use a filter for Region). The number of marks changes to 68.
3. Replace Category and Sub-Category (both dimensions that are part of the Product hierarchy) with copies (not part of a hierarchy) that have been renamed Product category and Product type. Remove the Sub-Category filter.

Doing this prevents users from expanding the hierarchy in the embedded view, which might increase the number of marks in the view and result in the view being rendered on the browser instead of the server.

The view is already more aggregated and simplified, but we’ll keep going.
- Move the Profit, Profit per Order, and Profit Ratio from the Columns shelf to Tooltip.

The number of marks is now 17. The underlying details in the data will still be available in the View Data window, but the reduced number of marks ensures the view will be WCAG-conformant because it will be rendered in the browser.

Step 3: Check the color palette settings.

The palette is set to Orange-Blue Diverging.

- Set Stepped Color to 5.
Step 4: Update and show text in the view.

1. Change the Title to make sense with the new view.
2. Show the worksheet Caption.
3. Show Mark labels.
4. Drag Profit Ratio to Label in the Marks card.
5. Edit the filter and legend titles.

Step 5: Open the View Data window to the Summary tab.

This is the text that screen readers will read for this view. The four measures are included in the underlying data for each product type.
Publish and embed the view

After you create your views, to make them WCAG-conformant, you must:

- Publish views to Tableau Server or Tableau Online, and then embed the view in a WCAG-compliant web page. For information about how to publish a view, see Comprehensive Steps for Publishing a Workbook on page 2521.

- Web editing is not supported in WCAG-conformant views, so Web Edit permissions must be turned off in the published workbook.

- In the embedded views, hide the Custom Views, Subscriptions, and Alerts buttons in the toolbar.

- Make sure users have permissions to access to the embedded views and are able to view the underlying data in the View Data window.

Turn off Web Edit permissions

To prevent your users from seeing the Edit option in the view toolbar, set the Web Edit permission to Deny when you publish your workbook. You should also let your Tableau Online administrator know when you publish workbooks that should not allow users to access web editing mode.
1. In Desktop, select Server > Publish Workbook.

2. Under Permissions, select Edit, select All Users, and then select Edit again.

3. For the Web Edit permissions, select Deny. Click Apply, then click Ok.

Embed the view

When you are ready to embed a view, go to the view in Tableau Server or Tableau Online and click the Share button in the view toolbar. Then copy the embed code and paste it into the HTML code of a WCAG-conformant web page.

In the view toolbar, the following options are supported for WCAG conformance: Undo, Redo, Revert, Refresh, Pause, Share, Download, and Full Screen.
Custom Views, Subscribe, and Alerts are not supported, so you will need to hide them in the embedded view.

To hide the View, Subscribe, and Alert buttons, set the `customViews`, `subscriptions`, and the `alerts` parameters in the embed code to no.

For example, in JavaScript:

The view toolbar is supported for conformance except the Subscribe and Alerts buttons, so you need to hide them in the embedded view. To hide the Subscribe and Alert buttons, set the `subscriptions` parameter and the `alerts` parameter in the embed code to no. For example:

```html
<param name='customViews' value='no' />
<param name='subscriptions' value='no' />
<param name='alerts' value='no' />
```

For more information about how to embed views, see Embed Views into Webpages.

Sample embed code

```html
<script type='text/javascript' src='http://Your_Server/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder' style='width: 1256px; height: 818px;'></div>

<object class='tableauViz' width='1256' height='818' style='display:none;'><param name='host_url' value='http%3A%2F%2FYour_Server%2F' />
<param name='site_root' value='' />
<param name='name' value='name of the visualization' />
<param name='tabs' value='yes' />
<param name='subscriptions' value='no' />
<param name='alerts' value='no' />
<param name='customViews' value='no' />
<param name='showShareOptions' value='true' />
</object>
</div>
```

In a view URL:
https://myserver/t/Sales/views/MyCoSales/SalesScoreCard?embed=y&:customViews=no&:subscriptions=no&:alerts=no

Embedded views must be rendered on the browser (not the server)

Server-rendered views are not WCAG-conformant; only client-rendered views are WCAG-conformant. To ensure an embedded view is rendered in the browser, make sure the view has less than 1000 marks, or set render to true in the URL for the embedded view.

For example:
http://localhost/views/Supplies/MyView?:render=true

Client-side rendering is enabled (true) by default.

About permissions

Users who need access to the underlying data will need the proper site role and permissions to view and interact with your embedded views. This includes having access to the projects and workbooks that contain the views. At the very minimum, your users will need View and Download Summary Data permissions. The ability to view the underlying data is necessary for users who use screen readers.

Users viewing the embedded view should have:

- Site role of Viewer, Interactor, or Guest.
  
  **Note:** The Viewer site role is more limited and will not be able to use filters or view the Full Data tab.

- **View** permission for the project that the workbook is in.

- **View** permissions for the workbook and its views.

- **Download Summary Data** permission to read the underlying data in the view on the Summary tab of the View Data window.

- **Download Full Data** permission if the user needs to be able to read all of the available data in the data source on the Full Data tab of the View Data window.

For information about how to set permissions for workbooks and views, see Set Permissions for Workbooks and Views in the online help.

Test the view

Test your embedded view using keyboard shortcuts for embedded views. Provide your users with this information in your user documentation.
Keyboard access for embedded views

Embedded views that are WCAG-conformant use commonly supported WAI-ARIA standards for keyboard navigation.

In this article

**Keystrokes for navigating embedded Tableau views below**

**Example: Navigate an embedded view** on page 2167

Keystrokes for navigating embedded Tableau views

<table>
<thead>
<tr>
<th>Component</th>
<th>Keystrokes</th>
</tr>
</thead>
</table>
| All       | - **Tab**: Move focus to next component  
- **Shift+Tab**: Move focus to previous component |
| Sheet Tabs| - **Left/Up**: Move focus to previous  
- **Right/Down**: Move focus to next  
- **Enter**: Navigate to the focused tab |
| Visualization (includes the view area, filters, and legends) |  
| Legends | - **Up**: Move focus to previous  
- **Down**: Move focus to next  
- **Left**: (Single column) move focus to previous, |
(Multi-column) move focus to left

- **Right**: (Single column) move focus to next, (Multi-column) move focus to right

- **Home**: Move focus to first item

- **End**: Move focus to last item

- **Space**: Toggle item selection

- **Enter**: Toggle highlighting on and off

- **Esc**: Clear all item selections

**Note**: To view legend item text with more contrast, select individual legend items, and then toggle highlighting on and off.

### Filters

- **Left/Up**: Move focus to previous item

- **Right/Down**: Move focus to next item

- **Home**: Move focus to first
item

- **End**: Move focus to last item

- **Space**: Toggle selection (check/clear checkboxes, select/clear radio buttons)

- **Enter**: Apply changes, if Cancel or Apply buttons available

- **Esc**: Revert changes to default settings.

**Filter controls and search**

- **Shift+Tab**: Move focus from filter item to filter control buttons (search, Single Value (List)/Multiple Values (List), Single Value (Dropdown)/Multiple Values (Dropdown), Include/Exclude Values). Press **Space** to click a button.

- For the drop-down menu, use **arrow keys** to move focus. Press **Enter** to choose a menu item.

- For search, press **Space** on the Search button, and then enter search terms. Press **Esc** once to clear input. Press **Esc** again to dismiss Search.
| **View Data window** | • **Esc** Clear mark selections.  
• When the view has the focus, press **Enter**, or press **Ctrl + Shift + Enter** to show the View Data window.  
The screen reader you are using might require some combination of modifier keys (**Shift**, **Ctrl**, **Alt**, **Cmd**) in conjunction with **Enter**. See the documentation for your screen reader.  
In the View Data window (when screen reader is on):  
• **Arrow keys**: Move within table.  
• **Tab**: Move focus to download data link in the window. Press **Enter** to navigate to link. |
|-----------------------------------------------|---------------------------------------------------------------|
| **Toolbar**                                  | • **Right** Move focus to next item.  
**Left** Move focus to previous item.  
**Enter** Select toolbar button. |
In the toolbar button window:

- Use **Tab** to move focus between window items.
- For drop-down menus, use **arrow keys** to move focus.
  Press **Enter** to choose a menu item.

**Example: Navigate an embedded view**

This example shows how you can access embedded views using keyboard navigation.

1. When you first open a web page that contains an embedded view, press **Tab** to move the focus to view page.

   In a workbook that has multiple sheet tabs, press the arrow keys to move focus between the tabs. Then press **Enter** or the spacebar to open a specific view.
1. Press **Tab** to move focus to the visualization area.

This image shows the visualization area in a sheet. A blue border indicates the focus.

When the visualization has the focus, screen readers will treat the visualization area as a graphic. Screen readers will identify the workbook that contains the view, and read the view's title and captions.

2. Navigate to filters and legends in the view by pressing **Tab**.

To change a filter setting, press **Tab** to move focus to the filter, and then press the arrow keys to move between items.

- Press **Space** to toggle item selections. Press **Enter** to apply filter changes. Press **Esc** to revert changes to the default settings.

- Press **Enter** to apply filter changes. Press **Esc** to revert changes to the default settings.

To highlight legend items, press **Tab** to move focus to the legend, and then press the arrow keys to move between items.
- Press **Space** to toggle item selections.

**Note:** To view legend item text with more contrast, select individual legend items, and then toggle highlighting on and off.

- Press **Enter** to highlight the selected legend items in the view. Press **Enter** again to turn off highlighting.

- Press **Esc** to clear all item selections.

See the table of keystrokes at the beginning of this article for the list of keystroke options for legends and filters.

3. To access underlying data that can be read by your screen reader, tab the focus to the visualization area and then press **Enter** or press **Ctrl + Shift + Enter**.

**Note:** Your screen reader may require some combination of modifier keys (**Shift**, **Ctrl**, **Alt**, **Cmd**) in conjunction with **Enter**. See the documentation for your screen reader for details.
An example of the table shown in the Summary tab of the View Data window

The View Data window opens in a new browser window.

The View Data window includes a link to download the raw data as a text file. The Summary tab lists the data that is currently selected in the view, or all data in the view if no data is selected. To clear selected data in the view, press Esc.

Use keyboard navigation for table items to move through the data.

4. To access buttons in the toolbar, tab the focus to the toolbar area and then use the right arrow to move to the next toolbar button.

Press Enter to select a toolbar button.

Select Tab to move through options in the button’s window, including exiting the window.

Tips for Working in Tableau

This section describes several tips for working efficiently in Tableau.

Watch videos: For even more tips, see the Why is Tableau Doing That? topic in the Free Training Videos library.

Use Show Me to Start a View

Show Me creates a view based on the fields already used in the view and any fields you’ve selected in the Data pane. Open Show Me by clicking Show Me on the toolbar.

In this article

- Start a view based on fields that you select
- Create a Line Chart with Show Me on page 2173
- Create a Scatter Plot with Show Me on page 2177

Start a view based on fields that you select

To use Show Me, select the fields you want to analyze in the Data pane, and then select the type of view you want to create in the Show Me pane. Tableau automatically evaluates the
selected fields and gives you the option of several types of views that would be appropriate for those fields. Show Me also highlights the visualization type that best matches the data.

1. Select fields in the **Data** pane that you want to analyze. Hold the Ctrl key (Command key on a Mac) to make multiple selections.

   ![Dimensions Tree](image)

2. Click **Show Me** on the toolbar and then select the type of view you want to create.
Any view type that is not gray will generate a view of your data. As you hover over each view type, the description at the bottom shows the minimum requirements. In this example, Stacked Bars is selected.

3. View the Result. Tableau automatically creates a view of the data.
Create a Line Chart with Show Me

In this example, you use Show Me to create a line chart that displays profit as a function of time.

1. Select [Order Date] and Profit in the Data pane. Hold down the Ctrl key (or the command key on a Mac) as you select the fields.
2. Click **Show Me** on the toolbar to display the **Show Me** card, if it isn't already displayed.

3. In the **Show Me** pane, see what type of view Tableau suggests.
Because a date dimension and a measure are selected, Tableau suggests you build a line view. A line is often the optimal way to look at how measure values change over time.

The view below shows the resulting chart. Each point on the line represents the sum of profit for the corresponding year.
You can more easily see the points by clicking the **Show Mark Labels** button on the toolbar.
Create a Scatter Plot with Show Me

In this example you use Show Me to build a scatter plot to show sales versus profit for each product and customer.

1. Select Customer Name and Product Name (from the Dimensions area in the Data pane) and Profit and Sales from the Measures area. To select multiple fields, hold down the Ctrl key (or the command key on a Mac) as you click fields.
2. Click **Show Me** on the toolbar.

3. Select the scatter plot chart type from **Show Me**.
Tableau creates a scatter plot using the fields you selected. You can drag fields to further refine the view.
Show and Hide Mark Labels

You can add labels to the data points in your visualization. For example, in a view that shows product category sales over time as a line, you can label sales next to each point along the lines.

In this article

- Show mark labels on the next page
- Hide mark labels on the next page
- Choose which marks to label (Tableau Desktop only) on the next page
- Show and hide individual mark labels on page 2186
- Move mark labels on page 2187
Show mark labels

To show mark labels in a viz:

- On the Marks card, click **Label**, and then select **Show mark labels**.

![Show mark labels](image)

Hide mark labels

To hide mark labels in a viz:

- On the Marks card, click **Label**, and then clear **Show mark labels**.

Choose which marks to label (Tableau Desktop only)

After you show mark labels in a worksheet, you can specify which marks to label.

To specify which marks to label:
• On the Marks card, click **Label**.

• In the dialog box that opens, under Marks to Label, select one of the following options:
  
  • **All**
  
  Label all marks in the view.

  ![Image showing labeled marks]

  • **Min/Max**

  Label only the minimum and maximum values for a field in the view. When you select this option, you must specify a scope and field to label by.

  For example, the image below shows the mark labels scoped to pane, and specifies to show the minimum and maximum sum of sales values. This means that the minimum and maximum sales are labeled for each pane in the view. In this case, there are four panes in the view.
- Line Ends

You can label the start or the end of all lines in the view, or both.

- Selected
Label only the selected marks in the view. Labels appear when you select one or more marks in the view.

- **Highlighted**

Label only the highlighted marks in the view. Labels appear when you select a member in a legend (for example, Office Supplies, in the following image), when you select one or more marks in the view, or when you select marks using the Highlighter. For more information about how to use the Highlighter, see *Highlight Data Points in Context* on page 1800.
- Most Recent

Label the most recent marks in the view. When there is a time or date field in the view, you can label all marks that correspond to the most recent date or time in the view. When you select this option, you can specify a scope for the labels in the view.

Marks labeled are determined by the level of detail of the view.

For example, the following image shows labels for each category, because the Category field is on Color on the Marks card. There is no label for the Furniture category, because the most recent mark in that category is excluded from the view.
Show and hide individual mark labels

Rather than showing all mark labels or dynamically showing labels based on the view, you might want to show labels for only individual marks. You can use mark labels to call out the values of specific marks of interest, as well as hide overlapping mark labels.

To show or hide individual mark labels:

- In a worksheet, right-click (control-click on Mac) the mark you want to show or hide a mark label for, select Mark Label, and then select one of the following options:
  
  - **Automatic** - select this option to turn the label on and off depending on the view and the settings in the Label drop-down menu.
  
  - **Always Show** - select this option to show the mark label even when it would otherwise be hidden (based on the settings in the Label drop-down menu).
  
  - **Never Show** - select this option to hide the mark label even when it would
otherwise be shown (based on the settings in the Label drop-down menu).

---

**Move mark labels**

After you show a mark label in a view, you can reposition it to best fit your view and presentation. For example, in a stacked bar chart, the mark labels are automatically placed in the center of each bar. However, you may want to stagger the labels so that the longer ones don’t overlap.

**To move a mark label:**
1. Select the mark whose mark label you want to move.

2. Click and drag the move handle to a new location.

Format mark labels

When you select to show mark labels in the view, there are several formatting options to help you adjust the appearance of the labels. You can customize the text, adjust the font properties, and set an alignment for all labels.

To edit the label text:

1. On the Marks card, click **Label**.

2. In the dialog box that opens, under Label Appearance, click the text button.
3. Edit the text and then click **OK**.

To edit the label font:

1. On the Marks card, click **Label**.
2. In the dialog box that opens, under Label Appearance, click the Font drop-down.
3. In the Font drop-down menu, you can do the following:
   - Select a font type, size, and emphasis.
   - Adjust the opacity of the labels by moving the slider at the bottom of the menu.
   - Select a color for the labels
     - To select a specific color, Click a color from the options.
To match labels to the color of their marks, click **Match Mark Color**.

4. When finished, click **OK**.

To edit the label alignment:

1. On the Marks card, click **Label**.
2. In the dialog box that opens, under Label Appearance, click the Alignment drop-down.
3. In the Alignment drop-down menu, you can do the following:
   - Select to align the label horizontally (left, center, right).
   - Select to change the direction of the text.
   - Select to align the label vertically (bottom, middle, top).
   - Select to wrap the text for all labels in the viz.
4. When finished, click **OK**.

Overlap other marks and labels (Tableau Desktop only)

By default, mark labels do not overlap other marks or labels in the view. You can show all labels in the view, even the ones that overlap other marks and labels.

**To overlap other marks in the view:**

- On the Marks card, click **Label**, and then, under the Options section, select **Allow labels to overlap other marks**.

See Also

- *Add Annotations* on page 1763
- *Create Aliases to Rename Members in the View* on page 977
Move Marks

In a dual-axis view that contains many marks, some marks can become hidden behind others, making it difficult to see details in your view. Tableau provides a Move Marks option to move selected marks forward or backward, depending on the axis that you select.

To move marks forward or backward, right-click on one of your axes and select one of the following options:

- Move Marks to Front
- Move marks to Back

Example - Move marks forward

In this example, suppose you have a view that uses the Circle mark type on a dual axis and shows department Sales and Profit for each Category.

The shape of each mark represents Sales and Profit, while Category is encoded as Color. The right axis represents the Sales mark, while the left represents the Profit mark.

Because the profit marks are in front, it is difficult to see the Sales marks in the Office Supplies category.

To move the Sales marks in front of the Profit marks, do the following:

1. Right-click on the Sales axis.
2. Select Move marks to front from the context menu.
By moving the **Sales** mark to the front, you can now see that rubber bands are below $100,000 in sales, whereas they were nearly invisible before.
Stack Marks

Stacking marks is relevant when your data view includes numeric axes. That is, at least one measure has been placed on the **Rows** or **Columns** shelves. When marks are stacked, they are drawn cumulatively along an axis. When marks are not stacked, they are drawn independently along an axis. That is, they are overlapping.

Stacking marks is particularly useful for bar charts which is why Tableau automatically stacks bars. You might find that stacking marks is useful for other marks such as lines as well.

You can control whether marks are stacked or overlapping in any given view by selecting the **Analysis > Stack Marks** menu item.

You can either allow Tableau to automatically select whether the marks are stacked or you can specify **On** or **Off**. The default setting is **Automatic**. When you are in automatic mode, the Stack Marks menu shows whether stacked marks is on or off.

Selecting **On** or **Off** on the **Stack Marks** menu, switches into manual mode. Your selection remains throughout any changes you make to the view.

The following examples illustrate stacking marks.

In this article

<table>
<thead>
<tr>
<th>Example: Stacking Bars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Stacking Lines</td>
</tr>
</tbody>
</table>

**Example: Stacking Bars**

Consider the stacked bars view shown below. It was created by placing a dimension on the **Columns** shelf, placing a measure on the **Rows** shelf, and color-encoding the data by a dimension (that is, dropping a dimension on **Color** on the Marks card).
Example: Stacking Lines

Consider the data view shown below. It was created by placing a date dimension on the **Columns** shelf, placing a measure on the **Rows** shelf, and color-encoding the data by a dimension (that is, dropping a dimension on **Color** on the Marks card). Because the mark type is Line, the marks are not automatically stacked. Instead, they are drawn independently from the horizontal axis.
Measure Values and Measure Names

The Data pane always contains a number of fields that do not come from your original data, two of which are **Measure Values** and **Measure Names**. Tableau automatically creates these fields so that you can build certain types of views that involve multiple measures.

- The **Measure Values** field always appears at the bottom of the Measures area in the Data pane and contains all the measures in your data, collected into a single field with continuous values. Drag individual measure fields out of the Measure Values card to remove them from the view.

- The **Measure Names** field always appears at the bottom of the Dimensions area in the Data pane and contains the names of all measures in your data, collected into a single field with discrete values.
In this article

How you can use Measure Names and Measure Values
The Measure Values card
About Measure Names
Create a visualization using Measures Names and Measure Values
When Measure Names and Measure Values automatically get added to the view
How Measure Names and Measure Values work with Legends Per Measure

How you can use Measure Names and Measure Values

Tableau automatically creates these fields so that you can build certain types of views that involve multiple measures.

Measure Values and Measure Names are in most respects like other fields in Tableau:

- You can display a filter for Measure Names.
- You can format Measure Values—the formatting is then inherited as the default formatting for all measures.
- You can drop Measure Values and Measure Names on shelves.
- You can assign aliases to the values for Measure Names.
- You can change the manual sort order for the values for Measure Names.

The Measure Values card

When Measure Values is in the view, Tableau creates a Measure Values card, outlined in red, below, that lists the measures in the data source with their default aggregations.
You can remove individual measures from the view by dragging them out of the Measure Values card.

### About Measure Names

When you want to show multiple measures in a view, you can use the Measure Values and the Measure Names fields. When you add Measure Names to a view, all of the measure names appear as row or column headers in the view. The headers include each measure name. This feature becomes useful when you are working with a text table that shows multiple measures.
For example, suppose you have a text table containing the aggregated profit of each product category by region.

![Text Table Example](image)

Now suppose you want to show both the **Profit** and the **Sales** for each category and region. When you add **Sales** to the text table (by dragging it and dropping it in the view), the measures are combined and the **Measure Values** field is added to Text. The **Measure Names** field is automatically added to the **Rows** shelf.
Notice how the header names removed the aggregation label by default. You may want to include the aggregations or call it "Total Profit" and "Total Sales." To change the measure names, right-click (control-click on Mac) the Measure Names field on the Rows shelf and select **Edit Aliases**. Make the changes and click **OK**.
Create a visualization using Measures Names and Measure Values

You can use **Measure Values** and **Measure Names** to display the values for all measures in your data source, using their default aggregations, simultaneously. The view below was created with the Sample - Superstore data source. It shows the values of all measures in the Data pane, using their default aggregations.

**To create this view**

Create a new worksheet using the Superstore example workbook

Drag **Measure Values** to Rows

Click Text Table in Show Me.

Text Table from Show Me.
Of course, this view is not well suited for any more than a quick survey of your data because you can’t generate a great deal of insight by comparing numbers that aren’t measuring the same thing—Profit and Sales are both in dollars, but Discount and Profit Ratio are percentages.

When Measure Names and Measure Values automatically get added to the view

Tableau will automatically add Measure Names and Measure Values fields to the view in response to certain actions on your part.
Blending Axes

You can show multiple measures on the same continuous axis by blending axes. For details, see **Blend axes for multiple measures into a single axis** on page 1826. When you use this technique, Tableau automatically adds Measure Names and Measure Values to the view, allowing you to add additional measures to the blended axis.

Using Show Me When Multiple Measures Are in the View

When there are multiple measures in the view, Tableau will automatically add **Measure Names** and **Measure Values**—or just **Measure Names**—when you choose certain visualization types from Show Me.

- When you choose side-by-side bars when there are multiple measures in the view, Tableau adds both **Measure Values** and **Measure Names**. For example, your original view might use lines to show **Sales** and **Profit** over time:

![Diagram showing side-by-side bars with Measure Values and Measure Names]

- If you choose side-by-side lines from Show Me, the view updates to include **Measure Names** and **Measure Values**:
When you choose side-by-side circles from Show Me when there are multiple measures in the view, Tableau adds **Measure Values** and **Measure Names**:

When you choose dual lines from Show Me when there are multiple measures in the view, Tableau adds **Measure Names** but not **Measure Values**:
This is similar to blending axes, except that Tableau uses separate value ranges (one on the left of the view and one on the right) to make the lines align as closely as possible. With blended axes, Tableau uses a single value range and the lines may or may not align well.

How Measure Names and Measure Values work with Legends Per Measure

When Measure Values and Measure Names are in the view, you can color code data for each individual measure. For details, see Measure values and color legends on page 222.

Handling Null and Other Special Values

Some data requires special handling in Tableau. Specifically:

- null values
- unrecognized or ambiguous geographic locations
• negative or zero values when working with a logarithmic scale
• negative or zero values when working with treemaps

In this article

About special values
Null numbers and dates, and negative values on log axes
Unknown geographic locations
Zero and negative values in treemaps

About special values

When your data contains any of these special values, Tableau cannot plot them in the view. Instead, it displays an indicator in the lower right corner of the view. Click the indicator to see options for how to handle these values.
Null numbers and dates, and negative values on log axes

When you drag a measure or continuous date to the view, the values are shown along a continuous axis. If the field contains null values or if there are zeroes or negative values on a logarithmic axis, Tableau cannot plot them. Tableau displays an indicator in the lower right corner of the view. Click the indicator and choose from the following options:

- **Filter Data** - exclude the null values from the view using a filter. When you filter data, the null values are also excluded from any calculations used in the view.

- **Show Data at Default Position** - show the data at a default location on the axis. The null values will still be included in calculations. The default position depends on the data type. The table below defines the defaults.

<table>
<thead>
<tr>
<th>Type of Data</th>
<th>Default Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numbers</td>
<td>0</td>
</tr>
<tr>
<td>Dates</td>
<td>12/31/1899</td>
</tr>
<tr>
<td>Negative Values on a Log Axis</td>
<td>1</td>
</tr>
<tr>
<td>Unknown Geographic Location</td>
<td>(0,0)</td>
</tr>
</tbody>
</table>

**Note:** See [Logical Functions](#) on page 1314 for functions, such as `ISNULL()` and `IFNULL()`, that you can use to handle null values with a calculated field.

Unknown geographic locations

When working with maps and geographic fields, unknown or ambiguous locations are identified by the indicator in the lower right corner of the view. Click the indicator and choose from the following options:

- **Edit Locations** - correct the locations by mapping your data to known locations.

- **Filter Data** - exclude the unknown locations from the view using a filter. The locations will not be included in calculations.

- **Show Data at Default Position** - show the values at the default position of (0, 0) on the
Zero and negative values in treemaps

When working with treemaps, any null or zero values display in the indicator in the lower right corner of the view. Click the indicator and choose from the following options:

- **Filter Data** - exclude the unknown locations from the view using a filter. The locations will not be included in calculations.

- **Use Absolute Values** - use the absolute value to determine the size of the corresponding area in the view. For example, both values of 5 and -5 are shown as the same size.

If you don’t know how to handle the values, you can choose to leave the special values indicator. Generally, you should continue to show the indicator so that you know there is data that is not being shown in the view. However, to hide the indicator, right-click (control-click on a Mac) it and select **Hide Indicator**.
Show or Hide Missing Values or Empty Rows and Columns

When you’re working with dates or numeric bins, Tableau only shows the values that are represented in your data. If your data does not contain the complete range of values, the missing values will not be shown. For example, your data may contain data for January through May and September through December. However, there was no data recorded for June, July, and August. If you create a line chart in Tableau, the missing months will not be shown. You can optionally show the missing months to make it clear that there was no data recorded during that time.

**Note:** You can replace missing values with a calculated field using the ZN() function. See Number Functions on page 1275 for details.

**Missing Values Hidden (Default)**

By default, missing values in a date range or numeric bins are not shown.

**Missing Values Shown**

You can show the missing values to indicate incomplete data.
To show missing values in a range, right-click (control-click on Mac) the date or bin headers and select **Show Missing Values**.
Show and Hide Empty Rows and Columns

When you are working with fields that are not dates or numeric bins, Tableau hides missing values by default. For example, you may be showing workers and the number of hours worked in a particular month. If a worker didn’t work at all that month, there may not be a row in the database for that worker. When you drag the Worker field to the Rows shelf, the workers that didn’t work are hidden by default. You can show the empty rows by selecting Analysis > Table Layout > Show Empty Rows. Similarly, show the empty columns by selecting Analysis > Table Layout > Show Empty Columns.

Empty Rows Hidden (default)  
Bob did not work in January so there are no records in the database for him. By default he is not listed.

Empty RowsShown  
Even though Bob did not work in January, he is listed but no marks are drawn.

Hide and Show Sheets in Dashboards or Stories

To streamline the editing process for large workbooks, hide and show sheets as you work.

In this article

- What sheets can I hide? on the next page
- Hide a sheet on the next page
- Unhide a sheet on page 2213
What sheets can I hide?

While editing a workbook, you can hide any sheet contained in a dashboard or story. (That includes dashboard sheets in stories.)

Sheets that aren’t in a dashboard or story are always visible while editing workbooks. However, you can hide any sheet in workbooks you publish to Tableau Online or Tableau Desktop. See Comprehensive Steps for Publishing a Workbook on page 2521.

**Note:** If you’re creating a viz in a tooltip, see these steps to hide the target sheet.

Hide a sheet

1. Look for the sheet in the tabs at the bottom of the screen. Or, if you’re viewing a dashboard or story, look in the Sheets list at left:

2. Right-click (Windows) or Control-click (macOS) the sheet name, and select Hide Sheet.
Note: From the sheets list to the left of a dashboard or story, you can hide only sheets used in that particular dashboard or story. You can identify these sheets by the blue checkmark on the sheet icon.

Unhide a sheet

1. At the bottom of the screen, click the tab for the dashboard or story that contains the sheet. You can identify dashboards and stories by grid and book icons, respectively.

Note: If a story has a hidden dashboard that in turn has a hidden sheet, first go to the hidden dashboard from the story, and then to the hidden sheet from the dashboard.

2. In the list at left, right-click (Windows) or Control-click (macOS) the sheet, and clear the
**Hide Sheet** checkbox.

Temporarily show a hidden sheet

While editing a dashboard or story, you can temporarily show a hidden sheet to get a closer look and change its structure. The sheet is hidden again when you switch to another sheet.

Do either of the following:

- On the dashboard itself, select the item that references the hidden sheet. Then click the drop-down arrow in the top left or right corner, and select **Go to Sheet**.
• In the **Sheets** list to the left of the dashboard, click the icon to the right of the sheet name.

![Sheets list](image)

**Pause Updates When Building Large Views**

If you are building a view that involves a large amount of data, you might find it easier to turn off automatic updates, build the view, and then resume updates. Turning off automatic updates queues the queries instead of sending a separate query to the database every time you drag a field.

If you are using filters in your views, you might also want to pause automatic updates for filters. This enables you to pause the query for filter changes until you are finished organizing your data. Otherwise every time you make a change to a filter, Tableau queries the data source, potentially slowing down performance.

For more information, see [Turn off Automatic Updates to Boost Performance](#) on page 2457.

Follow the steps below to pause automatic updates while building the view.

1. Click the **Pause Auto Updates** button on the toolbar to turn off automatic updates for worksheets.
2. Place all of the selected fields on the shelves.
3. Specify filters to restrict the data to view only the values that you are interested in.
   - If you also want to pause automatic updates for filters, click the drop-down arrow on the **Pause Auto Updates** button on the toolbar and select **Auto Update Filters** to clear the check mark.
4. Click the **Resume Auto Updates** button on the toolbar to turn automatic updates back on for worksheets.

   To resume automatic updates for filters, click the drop-down arrow on the **Pause Auto Updates** button on the toolbar and select **Auto Update Filters** again.

---

## Cube Data Sources

Cube data sources (also known as multidimensional or OLAP data sources) have certain characteristics that differentiate them from relational data sources when you work with them in Tableau. This topic describes these differences, and also identifies some Tableau features that are not available when you connect to a cube data source. In many cases there are alternative approaches that you can use to compensate for the unavailability of these features with cube data sources, but you may also have the option to connect directly to a relational database that was used as the source for a cube data source. Talk to your database administrator to find out if this is an option.

**Note:** Cube data sources are supported only for Tableau Desktop on Windows—not for the Mac.

---

## What Are Cube Data Sources?

A cube data source is a data source in which hierarchies and aggregations have been created by the cube's designer in advance.

Cubes are very powerful and can return information very quickly, often much more quickly than a relational data source. However, the reason for a cube's speed is that all its aggregations and hierarchies are pre-built. These definitions remain static until the cube is rebuilt. Thus, cube data sources are not as flexible as relational data sources if the types of questions you need to ask were not anticipated by the original designer, or if they change after the cube was built.

The cube data sources supported in Tableau are

- Oracle Essbase
- Teradata OLAP
- Microsoft Analysis Services (MSAS)
- SAP NetWeaver Business Warehouse
Create Calculated Members Using MDX Formulas

When working with a cube data source, you can create calculated members using MDX formulas instead of creating Tableau formulas. MDX, which stands for Multidimensional Expressions, is a query language for OLAP databases. With MDX calculated members, you can create more complex calculations and reference both measures and dimensions. A calculated member can be either a calculated measure, which is a new field in the data source just like a calculated field, or a calculated dimension member, which is a new member within an existing hierarchy. For details, see How to Create a Calculated Member on page 2225.

Tableau Features That Are Affected When You Use a Cube Data Source

When you use a cube data source, not all Tableau features work the same as with relational data sources, or are even available. The following table details the differences.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Status for Cube Data Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data blending</td>
<td>Cube data sources can only be used as the primary data source for blending data in Tableau. They cannot be used as secondary data sources. For details, see Troubleshoot Data Blending on page 696.</td>
</tr>
<tr>
<td>Date dimensions</td>
<td>For cube data sources, date dimensions are typically organized into hierarchies that contain levels such as year, quarter, and month. In addition, some multidimensional data sources have time intelligence enabled, which makes it possible to look at data levels different ways, such as Months by Year, Months by Quarter, Weekends, etc. These levels are represented as attributes of the hierarchy. For details, see Dates and Times on page 1058.</td>
</tr>
<tr>
<td>Filters</td>
<td>When you display a filter for a cube dimension, all levels of the hierarchy for that dimension are included in the filter. For example:</td>
</tr>
</tbody>
</table>
You can use cube attributes as filters, to show just a single level instead of a hierarchy. In the Data pane, attributes appear in the Dimensions section and are indicated by this icon: 🔄

Slicing filters behave differently with a cube data source than with a relational data source. See Create Slicing Filters on page 2222 for details.

Some dimension filter card options are not available. For example, single value (drop-down), multiple values (custom list), etc. Instead, dimensional filters shown in the view retain their hierarchical look and feel, and cannot be changed into specific list types. You can create a Tableau set containing certain values from the hierarchy and then use the set as a filter in the view with the expected filter options (right-click the set in the Data pane, and then click Show Filter).

In addition, see Filter Your Data Carefully on page 2445 for
### Sets
Cube data sources support hierarchical sets, which filter data to the selected members and all of their descendants. For details, see [Examples of Sets](#) on page 1013.

### Aliases
Aliases for cube databases are created by the cube’s designer and can be activated in Tableau by selecting the data source from the Data menu and then choosing **Alias File**. Talk to your database administrator to find out whether your database has aliases available. Aliases are not support by Microsoft Analysis Services databases.

By default the alias for every member of every dimension is initially defined to be the original member name. For example, the figure below shows a bar chart built from an Essbase database. By default, the original member names are displayed (example on the left). As you can see, these names are not very intuitive. By selecting **Data > Aliases File** and selecting an appropriate alias file set up by the database administrator, meaningful names are displayed in the headers.

### Groups
You cannot create groups when working with a cube data source. Any concept of groupings should be pre-defined in the cube as dimensional attributes or cube sets.

However, you can write multidimensional expressions (MDX) directly in Tableau using a calculated member to create a group. For example:

```
[Customer].[Customer Geography].[France] + 
[Customer].[Customer Geography].[Germany]
```

For a demonstration of this process, see the Tableau on-demand video [Analysis with Cubes and MDX](#).

### Advanced analytics features
Level of detail expressions, trend lines, forecasting, and clustering are not supported for cube data sources.

### Publishing
Workbooks using cube data source can be published to Tableau Server, but do not support pass-through connections. This means that you cannot make a connection from Tableau Server using such a data
source. It also means you cannot create a workbook using the data source in Tableau Server. For details, see **Cube Data Sources** in the Tableau Server help.

Publishing a cube data source to Tableau Server gives you the ability to store the data source on the server. However, to use the data source, you must download the data source to Tableau Desktop and use it locally.

Workbooks that use cube data source cannot be published to Tableau Online.

### Actions

With Microsoft Analysis Services connections, drill-down actions defined in the cube are not available in Tableau.

Cube data sources do not accept actions from relational or other cube data sources. See **Actions** on page 1789.

### Aggregate calculation functions

Cube data sources are pre-aggregated and thus do not support aggregation functions, such as `SUM()`, `AVG()`, and `CNT()`.

It may be possible to use Table Calculations to perform aggregation operations on the cell-level results from the cube in Tableau.

### Bins

For cube data sources, the **Create Bin** command is not available for measures.

You can, however, write a calculation that takes cube cell results and bins them. For example:

```
str((INT([Internet Sales Amount]/1000)) * 1000)
```

### Type conversions

For cube data sources, some type conversion functions are not available in the calculation editor. As a rule, data type conversions should be defined in the cube prior to analysis.

In particular, changing the data type of a cube dimension to a date in Tableau sometimes gives incorrect information. This feature is supported for some cube dimensions, depending on how the dates are formatted in the cube.

You can write Multidimensional Expressions (MDX) directly in Tableau using a calculated member to change the data type of a cube dimension.
| **Custom multidimensional expression (MDX) statements** | When connected to a cube, you cannot make a connection to a custom MDX statement. All sub-cube definitions need to be created on the server by implementing necessary cube perspectives, partitions, dimensions, or cell security. |
| **Data source filters** | Data source filters are not available for cube data sources. All field values must be defined in the cube prior to analysis in Tableau. |
| **Extracts** | You cannot create extracts from most cube data sources. Cube and relational data sources have incompatible data structures, which makes extracting data from a cube and storing it in a relational data source, such as the data engine, impossible in most cases. Starting in 10.4, you can create SAP BW extracts without the need for a special license key from Tableau. For information about support for and limitations of SAP BW extracts, see [SAP NetWeaver Business Warehouse](page 549). |
| **Hierarchies** | For cube data sources, hierarchies must be defined in the cube prior to analysis. |
| **Cube KPI data type** | When connected to Microsoft Analysis Services, any KPI calculations defined in the cube are not available. However, you can write your own KPI calculations in Tableau. You can also use Tableau parameters to create highly flexible what-if KPI analysis. For more information, see [Visualize Key Progress Indicators](page 934). |
| **Cube lag functions** | For cube data sources, you cannot use cube lag functions in the Tableau calculation editor. You can use Tableau Table Calculations to calculate certain percentages and totals instead. For more information, see [Transform Values with Table Calculations](page 1524). Alternatively, you can use an MDX Lag function directly in Tableau using a calculated member. For example: 

```sql
Avg ( { [Date].[Calendar].CurrentMember.Lag(4) : }
```
Tableau string functions

With cube data sources, dimensions are not available in the calculation editor. However, you can write MDX inside Tableau using a calculated member to manipulate dimensional values. For example:

```
LEFT([Product].[Product Categories].DataMember.MemberValue,LEN([Product].[Product Categories].DataMember.MemberValue)-5)
```

Parameters

For cube data sources, you cannot use parameter values to filter dimensions in an MDX calculation.

Create Slicing Filters

Slicing filters, also known as slicers or calculation filters, are dimension filters that behave differently for multidimensional data sources than they do for relational data sources.

In Tableau, multidimensional data sources are supported only in Windows.

A slicing filter exists when:

- The dimension that you place on the Filters shelf is not also on any other shelf in the view.
- The filter is defined to include multiple values. For example:

![Filter Window](image)
For relational data sources, there is nothing remarkable about a slicing filter. Tableau recalculates values for each measure in the view using the configured aggregations.

But because multidimensional data sources contain data that is aggregated in the data source, no aggregation can be defined for the “slice” that the dimension filter cuts through the values of individual measures. Tableau therefore automatically performs a summation. And because multiple values are defined, the result is the sum of a group of sums.

For example, consider the view shown below, which shows profit by region and state.

![Tableau view showing profit by region and state](image)

A slicing filter that includes the Q3 and Q4 members of the Quarter dimension is then applied to the data.
In the resulting view, the profit for Oregon in the West region is 2,567. This number was calculated by summing data values for Qtr3 and Qtr4.

The summation symbol (Σ) you see on the Filters shelf in the view above indicates that this is a slicing filter. If the original aggregation for a field is non-additive (for example, Count Distinct) the values you see after applying a slicing filter may be unexpected.
How to Create a Calculated Member

If you are using a multidimensional data source, you can create calculated members using MDX formulas instead of Tableau formulas. A calculated member can be either a calculated measure, which is a new field in the data source just like a calculated field, or a calculated dimension member, which is a new member within an existing hierarchy. For example, if a dimension Product has three members (Soda, Coffee, and Crackers) you can define a new calculated member Beverages that sums the Soda and Coffee members. When you then place the Products dimension on the Rows shelf it displays four rows: Soda, Coffee, Crackers, and Beverages.

**Note:** In Tableau, multidimensional data sources are supported only in Windows.

Defining Calculated Members

You can define a calculated dimension member by selecting **Calculated Members** from the Data pane menu. In the Calculated Members dialog box that opens, you can create, delete, and edit calculated members.

To create a new calculated member do the following:
1. Click New to add a new row to the list of calculated members at the top of the dialog box.

2. Type a Name for the new calculated member in the Member Definition area of the dialog box.

3. Specify the Parent member for the new calculated member. All Member is selected by default. However, you can choose Selected Member to browse the hierarchy and select a specific parent member.

   **Note:** Specifying a parent member is not available if you are connected to Oracle Essbase.

4. Give the new member a solve order.

   Sometimes a single cell in your data source can be defined by two different formulas. The solve order defines the precedence given to each formula. Formulas with a lower solve order are solved first. The default solve order is zero.

5. If you are connected to a Microsoft Analysis Services data source, the calculation editor
contains a Run before SSAS check box. Choose this option to execute the Tableau calculation before any Microsoft Analysis Services calculations. For information on connecting to Microsoft Analysis Services data sources, see Microsoft Analysis Services on page 481.

6. Type or paste an MDX expression into the large white text box.
7. Click Check Formula to verify that the formula is valid.
8. When finished, click OK.

The new member displays in the Data pane either in the Measures area, if you chose [Measures] as the parent member, or in the Dimensions area under the specified parent member. You can use the new member just like any other field in the view.

**Multidimensional Hierarchies**

Multidimensional data sources contain hierarchies. For example, your database might contain a Product dimension that includes members such as product family, product department, and so on organized into a hierarchy, or you might have a Time dimension that includes years, quarters, and months.

![Data pane examples]

Note: In Tableau, multidimensional data sources are supported only in Windows.
Drilling Down and Up in a Cube Data Source Hierarchy

Multidimensional (cube) data sources contain hierarchies. One of the most useful ways to navigate hierarchies is to drill down or drill up. For example, if you are examining the sales totals for various years, you can then drill down and view sales for all of the months within each year. Alternatively, if you are examining sales totals for all months, you can then drill up and view the sales for each year.

Hierarchies in cube data sources

When connected to cube data sources, you cannot create or customize hierarchies in Tableau. Hierarchies must be created in the cube before you connect to it in Tableau.

Hierarchies appear with the following icon in the Data pane: ✈️

You can see an example of a hierarchy below:

```
  Product Categories
    Category
    Subcategory
    Product Name
```

Note: You can only drill up or down in fields that are part of a hierarchy.

Drill up and down for all dimension members in a hierarchy

To drill up or down for all dimension members in a hierarchy:

- On the Columns or Rows shelf, or the Marks card, click the plus sign (+) on a field to drill down, or the minus sign (-) to drill up.
Drill up and down for **individual** dimension members in a hierarchy

To drill down and drill up for individual dimension members in a hierarchy:

- Right-click a table header and select **Drill Down** or **Drill Up**.
This is often referred to as non-uniform drill down because you expose only the members of interest instead of exposing all the members of a given level.

For example, you can see the difference between drilling down for all dimension members and drilling down for individual dimension members below.

**Individual dimension member: Bikes**

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Sales Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessories</td>
<td></td>
<td>$739,312.49</td>
</tr>
<tr>
<td>Bikes</td>
<td>Mountain Bikes</td>
<td>$25,419,777.11</td>
</tr>
<tr>
<td></td>
<td>Road Bikes</td>
<td>$28,885,971.10</td>
</tr>
<tr>
<td></td>
<td>Touring Bikes</td>
<td>$7,831,266.22</td>
</tr>
<tr>
<td>Clothing</td>
<td></td>
<td>$1,331,553.75</td>
</tr>
<tr>
<td>Components</td>
<td></td>
<td>$8,210,376.13</td>
</tr>
</tbody>
</table>

**All dimension members**
One reason to use a non-uniform drill down is if your data source has a ragged hierarchy (asymmetric layout). You also might want to view the children for just the member of interest.

**Note:** Drilling down and drilling up results in filtering the data.
Perfect Pivoting

In Tableau, perfect pivoting refers to working with hierarchies in these ways:

- Using varying levels of detail including skipping levels (for example, Country and City, but not State).
- Using varying levels of detail on different worksheet shelves simultaneously (for example, Product Family on the Columns shelf and Product Department on Color).
- Using varying levels of detail out of order (for example, Quarter before Year).

For example, in the following view the Market hierarchy is broken up to show the State level as Rows and the Region level as Color.

Defining Unique Values

Sometimes, when you are building views in Tableau, a field will have multiple members with the same name. For example, you may have a view showing the average profit by month over several years. The month February appears multiple times (once for each year).
While the name, February, is repeated, each instance of February can either be considered similar or unique. If you consider them similar, they will appear in the same column if you decided to move the Year field to the Color. However, if you consider them unique, they will be treated as two different values.
It is generally okay to consider repeated names within date and time fields (like in the previous example) similar but if there are repeated names in the Customer Name field, you won’t want to consider the two customers as the same person.

To define how you want Tableau to determine whether repeated values are unique, right-click (control-click on Mac) on the dimension in the Data pane and select one of the following on the Unique Values context menu:

- By Key: each member is considered unique based on the key given to it by the system administrator when the database is set up. Members with the same name but different keys are treated as unique values.
- By Name: each member is considered unique based on the member name. Members with the same name (regardless of their keys) are treated as if they are the same.

By default, unique date and time values are determined by name and all other values are determined by key.

Utility Dimensions

Oracle Essbase databases sometimes have special dimensions used to model comparative values such as Actual vs. Budgeted or Current Year vs. Previous Years. These dimensions are
the utility dimensions and are often set up as Scenario or Years. For example, the members of a Scenario dimension are shown below.

In the above view, you can see Actual Sales, Budgeted Sales, and so on. However, what if you wanted to compare Actual Sales to Budgeted Sales in a bullet graph? In that case, you need to set the Scenario dimension to be used as the utility dimension. When you set a dimension as the utility dimension, you can then specify which member of the utility dimension to use for each measure in the view. For example, below is a bullet graph showing actual sales to budgeted sales by region.

You can see that the Sales measure is used twice in the view: once to show actual and once to show budgeted.

**To use a dimension as the utility dimension:**
1. Right-click the dimension in the Data pane and select **Set as utility dimension**.

The dimension in the Data pane (in this example, Scenario) can no longer be used as a dimension field in the view. The Measures area of the Data pane indicates that there is a utility dimension (in this example, (by Scenario) indicates the utility dimension).
2. Drag a measure to the view.

3. Open the field menu for the measure in the view and select the member of the utility dimension you want to anchor the measure to.
To remove a utility dimension:

Open the drop-down menu at the top of the Data pane and select **Clear utility dimension**.

When you remove the utility dimension, measures that reference it in the view are no longer valid.
Dashboards

A dashboard is a collection of several views, letting you compare a variety of data simultaneously. For example, if you have a set of views that you review every day, you can create a dashboard that displays all the views at once, rather than navigate to separate worksheets.

Like worksheets, you access dashboards from tabs at the bottom of a workbook. Data in sheets and dashboards is connected; when you modify a sheet, any dashboards containing it change, and vice versa. Both sheets and dashboards update with the latest available data from the data source.

Best Practices for Effective Dashboards

A well-designed dashboard can align your organization's efforts, help uncover key insights, and speed up decision-making. Use this topic for tips on best practices for creating effective dashboards in Tableau.

In this article

- **What’s your goal?** below
- **Design for the real world** on the next page
- **Add interactivity to encourage exploration** on page 2243
- **Save time** on page 2246

What's your goal?

Know your purpose and audience

The best visualizations have a clear purpose and work for their intended audience. What will you be trying to say with this dashboard? Are you presenting a conclusion or a key question?

In addition to knowing what you're trying to say, it's important to know who you're saying it to. Does your audience know this subject matter extremely well or will it be new to them? What kind of cues will they need? Thinking about these questions before you head into the design phase can help you create a successful dashboard.
Leverage the most-viewed spot

Most viewers scan web content starting at the top left of a web page. Once you know your dashboard’s main purpose, be sure to place your most important view so that it occupies or spans the upper-left corner of your dashboard. In the dashboard below, the author decided that the map view holds the key message.

Design for the real world

Author at your final display size

By default, Tableau dashboards are set to use a fixed size and if you keep this setting, be sure to construct your visualization at the size it will be viewed at. You can also set **Size** to **Automatic**, which makes Tableau automatically adapt the overall dimensions of a visualization based on screen size. This means that if you design a dashboard at 1300 x 700 pixels, Tableau will resize it for smaller displays—and sometimes this results in scrunched views or scrollbars. The **Range** sizing feature is helpful for avoiding this.
If you're using Tableau Desktop to create dashboards, you can also design for specific device layouts, so that on tablets, for example, your dashboard contains one set of views and objects, and on phones it displays another. See Create Dashboard Layouts for Different Device Types on page 2302 for steps.

Limit the number of views

In general, it's a good idea to limit the number of views you include in your dashboard to two or three. If you add too many views, visual clarity and the big picture can get lost in the details. If you find that the scope of your story needs to grow beyond two or three views, you can always create more dashboards.

Too many views can also interfere with the performance of your dashboard after it's published. See Make Visualizations Faster on page 2450 for more details on performance.
Be security-savvy

A dashboard can include more than just views, it can include objects—including embedded web pages. If you include web page objects in your dashboard, here are some tips to keep in mind:

- **HTTPS**: When you add a web page object to your dashboard, you're prompted to specify a URL. It’s a best practice to use HTTPS (https://) in your URL. This ensures that the connection from your dashboard to the web page is encrypted. Also, if Tableau Server is running HTTPS and you use HTTP in the URL, your users' browsers won't be able to display the web page that the URL points to. If you don't specify a protocol, HTTP will be assumed.
- **Plug-ins**: If the web page object requires a plug-in, such as Adobe Flash, the plug-in must be installed on the computer that's running Tableau. The Tableau component that handles plug-ins requires that Safari be installed as well.
- **Web view security options** (Tableau Desktop only): You can control whether the web page object has JavaScript and plug-ins enabled, among other things. To adjust them, select Help > Settings and Performance > Dashboard Web View Security, and then clear one or more of the options listed below.:
- Enable JavaScript. Allows JavaScript support in the web view. Clearing this option may cause some web pages that require JavaScript to function improperly in the dashboard.

- Enable Plug-ins. Enables any plug-ins the web page uses, such as an Adobe Flash or QuickTime player.

- Block Pop-ups. When selected, blocks pop-ups.

- Enable URL Hover Actions. Allows URL hover actions. For more information, see URL Actions on page 1808.

Note: Any changes you make to the security options apply to all web page objects in your workbook, including new web page objects you create, and all subsequent workbooks you open in Tableau Desktop. To see your changes, you may need to save and reopen the workbook.

Add interactivity to encourage exploration

Show filters

Filters help users specify which data is shown in the view.

To turn on filters for a field:

- In Tableau Desktop—Right-click the field in the Data window and select Show Filter.
- In Tableau Server or Tableau Online—In the toolbar click Show/Hide Cards > Filters.

You can customize each filter for different types of data. For example, you can show filters as multi-select check boxes, single select radio buttons, or drop-down lists, etc. You can include a search button, the option to show all fields, null controls, and more. You can also edit the title of a filter to give your viewers clear instructions for interacting with the data.
Enable highlighting

You can use the Highlight button on the toolbar to set up highlighting between views. When highlighting is turned on, a selection in one view will highlight related data in the other views. You can turn on highlighting for all fields or select specific fields. For more information about the different methods you can use to highlight data, see Highlight Actions.

You can also display a highlighter that allows your customers to highlight parts of a view based on what they enter or select.

To display a highlighter:

1. Go to the worksheet where the view is (or select Go to Sheet from the dashboard).
2. Right-click the field you want to highlight and choose Show Highlighter:
In the highlighter, your users will be able to select or enter terms to highlight data in the view:
Save time

Use the Go to Sheet command

The views in a dashboard are connected to the worksheets they represent. Tableau provides short-cut menus to help you quickly accomplish basic tasks, like jumping from the dashboard you're working on to the original view, where you can perform other actions.

To jump to a sheet:
1. In your dashboard, select the view you want to go to.

2. Select **Go to Sheet**.

---

**Hide sheets as you go**

When you're working with a large number of worksheets to build dashboards, as you finish off a dashboard, you can hide the sheets that went into the dashboard. This can make your workbook easier to navigate.

To hide a dashboard's sheets, right-click (control-click on Mac) the dashboard's tab at the bottom of the workbook and select **Hide All Sheets**.
Create a Dashboard

After you’ve created one or more views, you can pull them into a dashboard, add interactivity, and much more.

Tip: To quickly get up and running, see Dashboard Starters for Cloud-based Data on page 2252.

In this article

- Open a dashboard sheet below
- Add views on the next page
- Add objects on the next page
- Add interactivity on page 2250
- Edit a web dashboard (Tableau Server and Tableau Online only) on page 2251

Open a dashboard sheet

You create a dashboard in much the same way you create a new worksheet. After you create a dashboard you can add views and objects.
To open a new dashboard sheet and start creating a dashboard, click the **New Dashboard** icon at the bottom of the workbook:

![Dashboard icons]

The **Dashboard** area appears on the left and lists the sheets in your workbook.

**Add views**

After you have a dashboard sheet, click the views you built (listed under **Sheets** on the left) and drag them to your dashboard sheet on the right. A gray, shaded area indicates where you can drop views:

![Dashboard area with views]

Click the image to replay it.

**Add objects**

In addition to adding views to your dashboard, you can add objects, including web pages, blank space, and layout containers.
To add an object, select an item under **Objects** on the left, and drag it to the dashboard sheet on the right:

![Objects](image)

Depending on the object you're adding, you may be prompted for additional information. If you are adding a Web Page object, see *Best Practices for Effective Dashboards* on page 2239 for tips on web view security options. Be aware that some web pages don't allow themselves to be embedded—Google is one example.

**Tip:** Use **layout containers** to fine-tune how your dashboard resizes when users interact with it.

---

**Add interactivity**

Part of the power of dashboards is that you can set up filters and interactivity to associate the different views on your sheet and enhance your users' analysis. The people looking at your dashboard can stay in a single spot—they don't need to click outside the world you've created for them.

In a dashboard you can use the **Use as Filter** option to make one of the views act as a filter on all the other views in the dashboard. To do this, select the filter button.
If you're using Tableau Desktop, there are many other actions you can add to your dashboard. For details, see Actions and Dashboards on page 1814 and Using Field and Filter Values in Actions on page 1822.

Edit a web dashboard (Tableau Server and Tableau Online only)

On the Views page in Tableau Server or Tableau Online, you can open a dashboard for editing in the following ways:

- In list view, select the check box for the dashboard, then select Actions > Edit Sheet.
- Click a dashboard to open it, and then click Edit.
Note: If you don't see the Edit option, ask the content owner or your administrator for web editing permissions. (The Save options available to you also depend on your permissions.)

Although it's possible to edit a workbook that has device layouts, it's not advised. Because device layouts inherit their content, formatting, and action from the parent (that is, from the default dashboard), editing the default dashboard can impact a device layout. You'll be able to see any issues in a device layout, but because device layouts are read-only, you won't be able to edit it. To edit a dashboard that has device layouts, download the dashboard to Tableau Desktop and make your edits there.

Dashboard Starters for Cloud-based Data

Dashboard Starters help you quickly author and analyze data from cloud-based systems like Salesforce, ServiceNow, Marketo, and Eloqua. Simply create a new workbook and choose from several beautiful, informative designs that are tailor-made for key business metrics.

In this article

Create a Dashboard Starter on the next page
Change permissions to share Dashboard Starters with colleagues on page 2254
Replace sample data with your data on page 2254
Fix grayed-out views by replacing field names on page 2254
Create a Dashboard Starter

It's easy to create a Dashboard Starter. When the process is complete, you can edit the resulting workbook just like any other, quickly customizing it for your needs.

1. In Tableau Online, navigate to the Workbooks tab, and click the New Workbook button.

2. At the top of the “Connect to data” window, click Dashboard Starter.

3. From the list of pre-built designs, find an option that reflects the data source and business metrics you need, and click Use Dashboard.

4. To quickly see how a workbook looks with sample data, click Continue without signing in. Or click Continue to create a workbook with your data.

5. Specify a name and containing project for the workbook.

6. If you chose to create a workbook with your data, sign in to your data source. While Tableau prepares an extract of your data, sample data appears so you can explore the layout.

That's it—you've created a rich, interactive dashboard in seconds!
Change permissions to share Dashboard Starters with colleagues

To avoid exposure of confidential data, workbooks for Dashboard Starters are visible only to authors and administrators by default. To share a Dashboard Starter with your colleagues, follow these simple steps:

1. In Tableau Online, navigate to the workbook for the Dashboard Starter.
2. Select the workbook, click Actions, and choose Permissions.
3. Give View permissions to any user or group you want to see the dashboard. For more information, see Edit Permissions in Tableau Online Help.

Replace sample data with your data

If you chose to quickly load sample data in a dashboard, you can replace it with your data at any time.

1. In Tableau Online, navigate to the workbook for the Dashboard Starter.
2. On the Data Sources tab, select the data source. From the Actions menu, choose Edit Connection.
3. For authentication, select Embedded credentials in the connection, and either choose an existing user account or add a new one. Then click Save.
4. On the Refresh Schedules tab, select the schedule. From the Actions menu, choose Run Now.

Fix grayed-out views by replacing field names

If your organization has customized the data structure for a cloud-based system, you may need to match those changes in Dashboard Starters after your data loads in them. For example, if your organization has renamed the Salesforce “Account” field to “Customer”, you’ll need to make a corresponding change in Dashboard Starters to avoid grayed-out views like this:
Fortunately, the fix is pretty straightforward:

1. Above the dashboard, click **Edit**.

2. **Navigate directly to the grayed-out sheet.**

3. In the Dimensions and Measures lists at left, look for red exclamation points (!) next to field names, which indicate that your organization uses different names.

4. Right-click each of those fields, and choose **Replace References**. Then select the
Fix empty dashboards by changing default date ranges

If a dashboard looks completely empty, the likely cause is a default date range that doesn't correspond to the dates in your source data.

1. **Download the workbook**, and open it in Tableau Desktop.

2. Click the **Data Source** tab.

3. In the upper-left corner, click the arrow next to the data source name, and choose **Edit Connection**. Then sign in.
4. Specify a date range that reflects the dates in your data, and click **Connect**.

5. Choose **Server > Publish Data Source** to update extracts of the data on Tableau Online.

### Size and Lay Out Your Dashboard

After you create a dashboard, you might need to resize and reorganize it to work better for your users.

In this article

- **Control overall dashboard size** on the next page
- **Group items using layout containers** on page 2260
- **Tile or float dashboard items** on page 2266
- **Size, position, and reorder individual dashboard items** on page 2270
- **Add padding, borders, and background colors around items** on page 2272
Control overall dashboard size

Dashboard size options

**Fixed size (default):** The dashboard remains the same size, regardless of the size of the window used to display it. If the dashboard is larger than the window, it becomes scrollable. You can pick from a preset size, such as Desktop Browser (the default), Small Blog, and iPad.

Fixed size dashboards let you specify the exact location and position of objects, which can be useful if there are floating objects. Select this setting if you know the precise size at which your dashboard will be displayed.

Published dashboards that use a fixed size can load faster because they’re more likely to use a cached version on the server. (Dashboards with variable sizes need to be freshly rendered for every browser request.) For other performance tips, see *Optimize Workbook Performance* on page 2438.

**Range:** The dashboard scales between minimum and maximum sizes that you specify. If the window used to display the dashboard is smaller than the minimum size, scroll bars are displayed. If it's larger than the maximum size, white space is displayed.

Use this setting when you're designing for two different display sizes that need the same content and have similar shapes—such as small- and medium-sized browser windows. Range also works well for mobile dashboards with vertical layouts, where the width may change to account for different mobile device widths, but the height is fixed to allow for vertical scrolling.
Automatic: The dashboard automatically resizes to fill the window used to display it.

Use this setting if you want Tableau to take care of any resizing. For best results, use a tiled dashboard layout.

Tip: If you use Tableau Desktop, see Create Dashboard Layouts for Different Device Types on page 2302 to create unique layouts optimized for desktop computers, tablets, and phones. In addition to adapting to different screen sizes, each device layout can contain different items.

Set overall dashboard size

- Under Size on the Dashboard pane, select the dashboard's dimensions (such as Desktop Browser) or sizing behavior (for example, Automatic).
Group items using layout containers

Layout containers let you group related dashboard items together so you can quickly position them. As you change the size and placement of items inside a container, other container items automatically adjust.

Layout container types

A horizontal layout container resizes the width of the views and objects it contains; a vertical layout container adjusts height.

Horizontal layout container  Vertical layout container
The two views below are arranged in a horizontal layout container.

The three views below are stacked in a vertical layout container.

Add a layout container

1. Under **Objects** on the Dashboard pane, select **Horizontal** or **Vertical**.
2. Drag the container to the dashboard.
3. Add views and objects to the layout container.
Evenly distribute a layout container’s items

1. Select the layout container. If you have trouble doing this, select an individual item within the container and choose **Select Container** from its shortcut menu.

2. With the layout container selected, choose **Distribute Evenly** from its short-cut menu:
Items that are already within the layout container arrange themselves evenly; any items you add will do the same.

**Format a layout container (Tableau Desktop only)**

You can specify shading and border style for layout containers to visually group objects in the dashboard. By default, layout containers are transparent and have no border style.

Open the shortcut menu for the layout container you want to format, and then select **Format Container**.
Automatically resize sheets in layout containers

If you add multiple sheets with related data to a layout container, whenever marks are selected in one sheet, you can automatically resize related sheets.

In this example, when a mark is selected in the map, the bar chart updates to display profit and sales for that mark; when no marks are selected in the map, it automatically expands to fill the layout container.

**With mark selected**

**Without a selection**

1. Add multiple sheets with related data to a layout container.

2. From the drop-down menu of the sheet you want to expand, choose Use as Filter.
3. Choose **Dashboard > Actions**, and double-click the generated filter you just created.

4. In the **Target Sheets** section of the Edit Filter Action dialog box, select the other sheets in the layout container.

5. To control how target sheets resize when no marks are selected in the source sheet, select one of the following:
   - **Show all values** returns target sheets to their original size, showing all data.
   - **Exclude all values** collapses target sheets under their titles, hiding all data.

**Remove a layout container to independently edit items it contains**

1. Select the container either on the dashboard or in the **Item hierarchy** area of the Layout pane.

2. From the drop-down menu at the top of the container, select **Remove Container**.

**Tile or float dashboard items**

**Tiled vs. floating layouts**

Each object, layout container, and view that you place on a dashboard is either tiled (the default) or floating.
**Tiled layout**

Tiled items don't overlap; they become part of a single-layer grid that resizes based on the overall dashboard size.

**Floating layout**

Floating items can be layered over other objects. In the example below, a map floats over tiled views.

For best results, give floating objects and views a fixed size and position.

**Float or tile a new item**

1. Under **Objects** in the Dashboard pane, click the layout option you want to use: **Floating** or **Tiled**.
2. Drag the view or object onto the dashboard on the right.

Tip: You can also assign a floating layout to an item by holding down Shift on your keyboard as you drag it onto the dashboard.

Switch an existing item from tiled to floating

1. Select the item in the dashboard.

2. In the item's shortcut menu, select Floating:
Float transparent legends, filters, highlighters, and parameters

To visually connect filters, parameters, and highlighters to related data, float these items, which are transparent by default. Text always remains fully opaque, maintaining legibility.
If a floating object continues to display a color, check these object- and worksheet-level settings:

- Select the object, and on the Layout tab, click the Background color, and choose None.
- Click the Format menu, and then choose Legends, Filters, Highlighters, or Parameters. Then, in the Format pane at left, choose Shading > None.

For more information, see Format at the Worksheet Level on page 2361.

Size, position, and reorder individual dashboard items

If you want a high degree of control over the placement of every item in a dashboard, choose Fixed size or Range for overall dashboard sizing, and then set each item's size and position using the Layout pane.
Set an item's size and position

1. Select the dashboard item you want to position and size. In the Layout pane, the item's name appears under **Selected item**.

![Dashboard and Layout panes](image)

**Tip:** To quickly toggle between the Dashboard and Layout panes, press the T key.

2. Define the item's position x and y position in pixels as an offset from the top left corner of the dashboard. For example, to place an object in the top left corner of the dashboard, specify x = 0 and y = 0.

   The values you enter can be positive or negative but must be whole numbers.

3. To resize an item, click and drag it in the dashboard. You can also use the settings under **Size**.

4. To hide the title of the item you have selected, clear the **Show title** check box. (The similar **Show dashboard title** check box on the Dashboard pane controls the overall dashboard title.)

   **Tip:** To quickly position floating items, press arrow keys to move 1 pixel, or Shift+arrow keys to move 10 pixels. To quickly resize items, add Alt (Windows) or Option (macOS) to these shortcuts.

**Align items with a grid**

To present a visually consistent design, arrange and size dashboard items over a grid.
1. Choose Dashboard > Show Grid.

2. To change the grid size, choose Dashboard > Grid Options.

**Tip:** To quickly toggle the grid on and off, press the G key.

**Reorder floating objects**

The **Item hierarchy** area on the Layout tab shows the relationships between objects on your dashboard. Drag floating items in the hierarchy to change how they layer over each other. Items at the top of the list appear in the front, while items at the bottom appear in back.

**Add padding, borders, and background colors around items**

Padding lets you precisely space items on dashboard, while borders and background colors let you visually highlight them. Inner padding sets the spacing between item contents and the perimeter of the border and background color; outer padding provides additional spacing beyond the border and background color.

**Note:** If you can't change the border or background color for a particular dashboard item, change its formatting at the worksheet level.
A. Outer padding  
B. Blue border  
C. Inner padding with light blue background

1. Select an individual item, or your entire dashboard.

2. On the **Layout** tab at left, specify border style and color, background color and opacity,
Here are some handy dashboard spacing tips:

- To precisely align one dashboard item with another, deselect **All sides equal**, and adjust padding for only one side.
To create seamless designs, specify zero outer padding for adjoining items.

Refine Your Dashboard

After you’ve created and laid out your dashboard, take a moment to step back and evaluate it. This topic lists some areas to check and refine.

In this article

Is everything in the right spot? below
Review sizing and layout on page 2277
Highlight the essentials on page 2284

Is everything in the right spot?

Confirm placement

Make sure that the view with your key finding or main question spans or occupies the upper-left corner of your dashboard. In your views, the most important data should be on the X or Y axis and your less important data should be on color, size, or shape.
Check label orientation and readability

Important labels should be horizontal so that your viewers can easily read them.
Review sizing and layout

Test your work and adjust sizing

Once you finish your dashboard, it's a best practice to test it and experience it as your users will. For example, will they be using Tableau Reader on a laptop to view your dashboard, will the dashboard be published on Tableau Server, or will it be embedded in a web application and viewed from a tablet?

After you test your dashboard, you'll probably find some things to change—sizing and how and where items are positioned are usually among them.

Tableau dashboards are set to a fixed default size that's intended to work well on a typical desktop. However, when you publish (to the web, in a blog, for a presentation, etc.) you may find yourself more limited.

You can specify the overall size of the dashboard using the settings under Size in the Dashboard area on the left. Select a new size using the drop-down menu.

- **Automatic** - The dashboard automatically resizes to fill the window it's displayed in.
- **Fixed size** - The dashboard always remains a specific size. If the dashboard is larger than the window the dashboard becomes scrollable.
- **Range** - The dashboard scales between minimum and maximum sizes that you specify, after which scroll bars or white space display.

Customize image display (Tableau Desktop only)

When you add an Image object to a dashboard, you can customize how the image displays by selecting an option on the image's menu. For example, you can select whether to **Fit Image**, which scales the image to the size of image object on the dashboard.
By default, any high-resolution images that you use in a dashboard are displayed in standard resolution and will maintain the same size whether they're viewed on standard or Retina displays.

To take advantage of high-resolution images in a dashboard that are intended to be viewed on a Retina display, you can use the @2x naming convention:

1. Navigate to the high-resolution image on your computer with Retina display.
2. Rename your high-resolution image to include the @2x modifier by using `<image-name>@2x<file-extension>` format. For example, logo@2x.png.
3. In your workbook, click and drag an Image object onto your dashboard.
4. When you're prompted, select the image you just renamed using the @2x naming convention and then click Open.

**Note:** EMF image file formats are not compatible with Tableau Desktop on the Mac.

Use a layout container

A layout container can create a smoother experience for your users by helping dashboard objects reposition and resize when one view is filtering another.

The image below shows how a dashboard behaves when two views are placed in a vertical layout container versus how they behave when they are not placed in a layout container. Notice how, in the dashboard that uses a layout container, the views adjust vertically as different filters are applied.
## With a Layout Container

![Map of the United States with states colored differently](image)

<table>
<thead>
<tr>
<th>Sub-Category</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookcases</td>
<td>-4,436</td>
</tr>
<tr>
<td>Chairs</td>
<td>13,235</td>
</tr>
<tr>
<td>Furnishings</td>
<td>7,919</td>
</tr>
<tr>
<td>Tables</td>
<td>-9,728</td>
</tr>
<tr>
<td>Appliances</td>
<td>6,982</td>
</tr>
<tr>
<td>Art</td>
<td>3,454</td>
</tr>
<tr>
<td>Binders</td>
<td>17,996</td>
</tr>
<tr>
<td>Envelopes</td>
<td>3,264</td>
</tr>
<tr>
<td>Fasteners</td>
<td>577</td>
</tr>
<tr>
<td>Labels</td>
<td>3,076</td>
</tr>
<tr>
<td>Paper</td>
<td>15,536</td>
</tr>
<tr>
<td>Storage</td>
<td>7,104</td>
</tr>
<tr>
<td>Supplies</td>
<td>-1,658</td>
</tr>
<tr>
<td>Accessories</td>
<td>20,736</td>
</tr>
<tr>
<td>Copiers</td>
<td>24,084</td>
</tr>
<tr>
<td>Machines</td>
<td>2,141</td>
</tr>
<tr>
<td>Phones</td>
<td>23,837</td>
</tr>
<tr>
<td>Bookcases</td>
<td>635</td>
</tr>
<tr>
<td>Chairs</td>
<td>8,345</td>
</tr>
</tbody>
</table>

Click the image to replay it.

To add a layout container
1. Drag a horizontal or vertical layout container to the dashboard.

2. Add sheets and objects to the layout container. As you hover over the layout container, a blue box indicates that the object is being added to the flow of the layout container.
3. To position items evenly, choose **Distribute Evenly** from the layout container's shortcut menu:
The **Distribute Evenly** command is only available for layout containers that you explicitly add to a dashboard. It’s not available for layout containers that are automatically added when you add items.

**Buy space with a floating layout**

Tableau uses a tiled layout by default in dashboards, which means that each view, legend, and object is arranged in a single-layer grid, similar to a tiled floor.

If you want an item to overlap another, you can use a **Floating** layout for one or more items instead:
Legends can sometimes be floated over other items instead of remaining tiled.
Highlight the essentials

Show just what your users need

Legends, titles, captions, and filters can be helpful for your viewers. Make sure you give your users what they need and no more.

To change what's displayed for your users:

1. Select a view in the dashboard.
2. Click the drop-down menu in the upper-right corner of the selected view and select the items you want to show. For example, you can show the title, caption, legends, and a variety of filters.
Alternatively, you can right-click an item in the **Layout** section of the **Dashboard** pane to access all of these same commands.
Note: Filters are only available for the fields used in the original view.

Eliminate clutter

In general, you should try to use no more than two color palettes in a single dashboard, and ideally, just one palette if your data is quantitative. Remove unnecessary text, lines, or shading that don't provide actionable information for your viewers. Make sure every legend you provide is really needed, remembering to take into account the final, published size of your dashboard.
Make a sheet sorter (Tableau Desktop only)

A sheet sorter can be a time-saver for the people viewing your dashboard. It allows them to quickly display individual worksheets on the dashboard. For information, see Create a Sheet Selector for a Dashboard on the next page.
Create a Sheet Selector for a Dashboard

Create a drop-down menu that lets users select an individual view and expand it to fill a dashboard.

1. On an individual worksheet, right-click an empty area of the Data pane at left, and select Create Parameter.
2. In the Create Parameter dialog box, do the following:

   - Enter a name that will appear above the menu, like **Select a View**.
   - For **Data type**, select **String**.
   - For **Allowable values**, select **List**.
   - Under **List of values**, type **All** for the first value, and then add values with the
name of each view in the dashboard.

3. Click **OK**.

4. On any sheet, right-click an empty area of the Data pane at left, and select **Create Calculated Field**.

5. Give the calculation a descriptive name like **Display sheet**. In the formula text box, enter the name of the parameter you created above. Then click **OK**.
6. One by one, open each sheet you plan to add to your dashboard, and drag the new calculation to the Filters shelf. Then do the following in the Filter dialog box, and click OK.

   - Select Custom Value List.
   - Type All in the text box, and click the Add Item button.
   - Type the current view's name (like "Map") in the text box, and click the Add Item
7. Select **Dashboard > New Dashboard**.

8. From the **Objects** section at lower left, drag a **Vertical** or **Horizontal** layout container to the dashboard.
9. Now drag each sheet to the layout container, identified by the dark blue outline.

10. Right-click the title area of each view in the dashboard, and select **Hide Title**.
    (If titles are visible, they prevent selected views from filling the dashboard.)

11. From the drop-down menu at the top of a view, choose **Parameters > [new parameter name]**.
Your sheet selector is ready to go! For similar examples, see Use Parameters to Make Views More Interactive on page 1039.

Rebrand a Dashboard

Tableau Desktop builds visual best practices into its default formatting settings so that your visualizations look great from the start. But you might want to change formatting settings for a dashboard—for example, to align them with your organization's brand.

This article steps you through a dashboard rebranding process for a fictitious organization called Citywide Bike Share. Its brand includes turquoise, orange, and gray for its colors and Century Gothic for its font. The overall look is light and uncluttered and, of course, it's all about bikes!

Before

After
In this article

Change the fonts and font color of your titles below
Change the marks colors on the next page
Customize your tooltips on page 2298
Add an image or logo on page 2300

Change the fonts and font color of your titles

Fonts and colors are usually an important part of an organization's brand. The quickest way to make a fast, large-scale change to all the titles in your dashboard is to change them at the workbook level. A workbook is the largest possible container for formatting settings.

1. Click **Format > Workbook**.

2. Make your changes under **Fonts**, in the **Format Workbook** pane.

   Here we’ve changed the font to Century Gothic and the font color to turquoise. We've chosen to change every title in the workbook, but you could also change the look of individual titles.
If the font you want to use doesn't exist in Tableau by default, you can add it. See Use Custom Fonts on page 2381 for details.

Change the marks colors

Marks represent data in a view. In this rebranding scenario, the marks show bike share locations. Changing marks colors is another way to communicate your organization's brand to your audience. For certain types of marks, you can also add and format halos, which is a colored ring around a mark.

1. Go a view on your dashboard and click Go to Sheet from the view's shortcut menu.
2. In the view, click Color on the Marks card.
3. Make your new color selection.

Here we've changed the marks in the Chicago Ride Share map to use gray with a turquoise halo:
4. Repeat the above steps for every view in your dashboard.

If a color you want to use isn't available, you can create your own custom palette. See Create Custom Color Palettes on page 2411 for details. For example, here we've changed the colors in the bar chart by created a custom diverging palette.
Now, with just a few changes to fonts and marks colors, the dashboard branding is already taking shape.

Customize your tooltips

Tooltips appear when you hover the mouse pointer over a mark. Tooltips appear by default for most views. They’re a great way to reinforce your brand and tell a better story.

Here’s what the map tooltips looked like by default:
1. Go to the view's worksheet and click **Worksheet > Tooltip**.

2. Confirm that the tooltip has the details you want to display. You can change fonts, order, wording, alignment, and color.

   For this rebranding scenario, we reduced the number of items displayed in the tooltip so that it's easier for users to quickly see what they're most interested in. We also reworded a few items and added the color orange as a contrasting, brand-appropriate color:
Here’s what the map tooltips look like now:

Add an image or logo

Adding your own images can also be a way to bring your organization's brand into a workbook. For example, you can import a logo and use it at the top of your dashboard instead of displaying the default dashboard title.

To remove the title and add an image:

1. In the Dashboard pane, stop displaying the dashboard title by clearing the Show dashboard title check box in the lower left corner.

2. Drag the Image object to the area where you want to display your logo. You may also need to adjust or remove other dashboard objects.
Click the image to replay it.

Here’s our final, rebranded workbook:
Create Dashboard Layouts for Different Device Types

After you've built a dashboard you can create layouts for it in Tableau Desktop that are specific to particular types of devices. When you publish the dashboard to Tableau Server or Tableau Online, people who interact with it experience a dashboard designed specifically for their screen size, whether it's a phone, tablet, or desktop. As the author, you only have to create a single dashboard, and deliver a single URL.

In this article

~ Preview and add a device layout below ~
How the default dashboard relates to device layouts on page 2304
Customize a device layout on page 2305
Add more layouts and optimize for phones on page 2308
Publish the dashboard on page 2310
Test the dashboard on page 2311
Confirm which layout a device will display on page 2312

Preview and add a device layout

1. Open a dashboard.
2. On the Dashboard tab on the left, click Device Preview.
In device preview mode, these options appear above the dashboard:

3. Take a moment to click through the **Device types** and **Models** and explore the different screen sizes. Then set these options:

   - To see how the dashboard will look in landscape vs. portrait mode, click 📸. Usually, landscape is optimal for tablets and portrait is best for phones.
   - Select **Tableau Mobile app** to see how the dashboard will look with the app instead of the browser. This option is available for iOS or Android devices and shrinks the dashboard slightly, leaving space for the app controls.

4. Choose a **Device type**, such as **Tablet**.
5. In the upper-right corner, click the Add Layout button for the device type you selected (for example, Add Tablet Layout).

Note: Your device-specific dashboard will not update automatically if you make changes to an original view. Create a device-specific layout when all changes to your view have been made.

How the default dashboard relates to device layouts

After you add a device layout, the black frame changes to a lighter outline with angle brackets and you’ll see the layout appear on the Dashboard tab, under Default. The new device layout contains every item that the Default dashboard does and its size and layout are derived from Default as well.
Think of the Default dashboard as the parent, and the device layouts (desktop, tablet, and phone) as its children.

Any view, filter, action, legend or parameter that you want to add to a device layout must first exist in the Default dashboard. However, each device layout can contain independent web, image, text, and layout objects.

Customize a device layout

After you've added a device layout to your dashboard, you can customize it.

1. Click **Custom** and start removing and rearranging views and objects to create the look you want.
If you remove an item from the device layout, it's only removed from that layout. It still exists on the default dashboard and can be added back to the device layout again.

Anything you can add to your layout is listed on the left, under Layout. If an item has a blue check mark, it means that it's part of the device layout you're currently working on.
2. Click through the **Device model** options to see how the layout will appear on different models.

![Device model options]

Ultimately, it's the size of the web browser that loads the dashboard that determines which layout appears on the device. For details, see Confirm which layout a device will display on page 2312.

3. Explore the options under **Size**.

![Size options]
**Default**: The height and width of the device layout mimics whatever the default dashboard is using. For example, if you’re creating a tablet layout and the default dashboard is set to a fixed size of Desktop Browser (1000 x 800), setting Size to Default for the tablet layout will make it use 1000 x 800 as well.

**Fit all**: All items are automatically resized to fit the device frame size. The device frame size is determined by the Device type, Model, and orientation (portrait or landscape) settings.

**Fit width** (recommended for phones): Items are automatically resized to fit the width of the device frame, but the height is fixed. This is a great option for phone layouts and vertical scrolling.

### Add more layouts and optimize for phones

1. Add an additional layout by selecting a new **Device type**:

   ![Device type selection](image)

2. Click **Add Layout**.

3. (Optional) One technique for creating a dashboard layout for phones is to create duplicates of certain views in the default dashboard—one optimized for desktop viewing and a second optimized for phones.

   For example, to create a duplicate of a map view that has different functionality, go to the worksheet for the view, click its tab, and select **Duplicate Sheet**.
Next, customize the view for mobile viewing. For example, you may want the map to be zoomed in to a specific region by default, or you may want to disable panning, zooming, and other maps functionality. See Customize How People Interact with your Map on page 2069.

Finally, add the new view to the default dashboard so that it can be available to the device layouts you’re creating.
4. As you optimize for different device layouts, dashboard titles are another area to look at. Short titles work best for mobile viewing. To edit a title, double-click it.

5. White space is another visual element to consider if you’re designing for phones. While screen real estate on a phone is scarce and you want to make the most of it, you may also want to provide additional safe places for your users to tap or initiate scrolling, so they don’t select filters and other items unintentionally.

   To add white space, use padding or Blank objects. For more information, see Size and Lay Out Your Dashboard on page 2257.

6. (Optional) To let phone users quickly contact key people about dashboard content, add URL actions to objects that automatically trigger SMS messages and telephone calls. Use the link format sms:phone-number or tel:phone-number. Be sure to include country and area codes if necessary.

7. Add device layouts until you have all three possible layouts under Default: Desktop, Tablet, and Phone.

Creating a layout for each device type gives you the most control over your users’ experience as they view your dashboard from different devices. After you publish a dashboard with all three layouts, users won’t see the default dashboard layout; instead, they’ll always see the appropriate device-specific layout.

Publish the dashboard

1. Click Server > Publish Workbook. If you're not already signed in, you're prompted for your credentials.

2. In the Publish Workbook to Tableau Server dialog box, make sure the Show sheets as tabs check box is cleared.
When this check box is selected for device-specific dashboards, the tabs' sizing requirements interfere with the server's ability to correctly detect the size of the web browser and load the correct layout.

3. Click **Publish**.

**Test the dashboard**

After you publish the dashboard to Tableau Server or Tableau Online, test the dashboard by viewing it from different browser sizes.

1. Open the dashboard on Tableau Server or Tableau Online.

2. In the upper-right corner of the page, click **Share** and copy the contents of the **Link** text box.

3. Paste the string into a web browser URL. The string should include the following:

   `embed=y`
4. With the embed code string as your browser URL, test the different layouts by changing the size of your web browser window and refreshing it.

Confirm which layout a device will display

The dashboard layout a device displays is based on the smallest dimension (height or width) of the iframe in which the Tableau view appears. Sometimes Desktop, Tablet, or Phone layouts may appear on other types of devices. For example, a Tablet layout may appear on a desktop computer if the display or browser window is small.

<table>
<thead>
<tr>
<th>If the smallest iframe dimension is...</th>
<th>This device layout appears...</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 pixels or less</td>
<td>Phone</td>
</tr>
<tr>
<td>Between 501 and 800 pixels</td>
<td>Tablet</td>
</tr>
<tr>
<td>Greater than 800 pixels</td>
<td>Desktop</td>
</tr>
</tbody>
</table>

Use Dashboard Extensions

Extensions let you add unique features to dashboards or directly integrate them with applications outside Tableau. Adding extensions is easy; you incorporate them into dashboard layouts just like other dashboard objects.

Extensions expand dashboard functionality with the help of web applications created by third-party developers. If you’re a developer and want to create your own extensions, see the Tableau Extensions API documentation on GitHub.

**Note:** Tableau Server administrators can turn off dashboard extensions features.

The animation below shows an example extension. With other extensions, you'll see different options.
Add an extension to a dashboard below
Ensure that extensions run on Tableau Online or Tableau Server on page 2315
Data security and dashboard extensions on page 2316
Get support for dashboard extensions on page 2317

Add an extension to a dashboard

To add an extension, you first need to download the related .trex file, which specifies the extension’s properties, including the URL for the web-based application.

1. In a Tableau workbook, open a dashboard sheet.
2. From the Objects section, drag **Extension** to the dashboard.
3. In the “Choose an Extension” dialog box, do either of the following:
   - Click **My Extensions**, and navigate to a .trex file you previously downloaded.
   - Click **Extension Gallery**, and download a new extension. Then click **My Extensions**, and navigate to the downloaded .trex file.

4. If prompted, allow or deny the dashboard extension access to data in the workbook. For more information, see [Data security and dashboard extensions](#) on page 2316.

   If you allow access, follow any on-screen instructions for configuring the extension.

   **Note:** Extension objects will appear blank in prints, PDFs, and images of dashboards (including images in subscription emails).

**Configure a dashboard extension**

Some dashboard extensions provide configuration options that let you customize features.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Configure**.
2. Follow the on-screen instructions to configure the extension.

Reload a dashboard extension

If a dashboard extension becomes unresponsive, you might need to reload it, which is similar to refreshing a web page in a browser.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Reload**.
   
   The dashboard extension is refreshed and set to its original state.

2. If reloading the extension fails to return it to a useable state, try removing it from the dashboard and adding it again.

Ensure that extensions run on Tableau Online or Tableau Server

You can add extensions to workbooks you publish from Tableau Desktop or directly in the web-authoring mode of Tableau Online and Tableau Server. A Tableau administrator must allow extensions to run on a site and add extensions that access full underlying data to a safe list. Administrators should only allow extensions that you have tested and trust.

If you want to use a dashboard extension on Tableau Online or Tableau Server, direct your administrator to Manage Dashboard Extensions in Tableau Online or Manage Dashboard Extensions in Tableau Server.
Data security and dashboard extensions

Dashboard extensions are web applications, which can be running on web servers located outside of your local network. Before adding an extension or viewing a dashboard with one, be certain that you trust the website that hosts it. By default, dashboard extensions use the HTTPS protocol, which guarantees an encrypted channel for sending and receiving data, and ensures some privacy and security.

Allow or deny data access to an extension

Depending on how an extension is designed, it can access either visible data in a view, or full underlying data, table and field names from data sources, and information about data source connections. When you add an extension, or view a dashboard with one, you’re given an opportunity to allow or deny the extension to run and access this data.

If you’re viewing a dashboard with an extension that requires full data access, and that access has been denied, a message appears in place of the extension. If you trust the extension and want to use it, you can reset permissions and allow the extension to run.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Reset Permissions**.

2. Click either **Allow** to let the extension run and access data, or **Deny** to prevent the extension from running.

Enable JavaScript

Dashboard extensions interact with data using the Tableau Extensions API library, a JavaScript library. If you want to use extensions, be sure that JavaScript is enabled in the dashboard.

Get support for dashboard extensions

To get help for an extension, you'll need to contact the developer or company who created it.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose About.
2. Click Get Support to go to the support page of the extension developer.

Tableau doesn't provide support for extensions or for other programs that interface with the Extensions API. However, you can submit questions and ask for help in the Tableau developer community.
Stories

In Tableau, a **story** is a sequence of visualizations that work together to convey information. You can create stories to tell a data narrative, provide context, demonstrate how decisions relate to outcomes, or to simply make a compelling case.

A story is a sheet, so the methods you use to create, name, and manage worksheets and dashboards also apply to stories (for more details, see *Workbooks and Sheets* on page 226). At the same time, a story is also a collection of sheets, arranged in a sequence. Each individual sheet in a story is called a *story point*.

When you share a story—for example, by publishing a workbook to Tableau Public, Tableau Server, or Tableau Online—users can interact with the story to reveal new findings or ask new questions of the data.

---

**The Story Workspace**

As you work on a story, you can use the following controls, elements, and features.
A. **Options for adding a new story point**: Choose **Blank** to add a new point or **Duplicate** to use the current story point as the starting place for your next point.

B. **The Story pane**: Use this pane to drag dashboards, sheets, and text descriptions to your story sheet. This is also where you set the size of your story and display or hide the title.

C. **The Layout pane**: This is where you choose your navigator style and display or hide the forward and back arrows.

D. **The Story menu**: Use this menu in Tableau Desktop to format the story or copy or export the current story point as an image. You can also clear the entire story here or show or hide the navigator and story title.

E. **The Story toolbar**: This toolbar appears when you mouse-over the navigator area. Use it to revert changes, apply updates to a story point, delete a story point, or create a new story point out of the current, customized one.

F. **The navigator**: The navigator allows you to edit and organize your story points. It’s also how your audience will step through your story. To change the style of the navigator, use the Layout pane.

For more information on these options, see **Create a Story** on page 2325.
Best Practices for Telling Great Stories

A good data story brings data and facts to life. Use this article for tips on best practices to use with Tableau's story points feature.

In this article

What's your story's purpose? below
The seven types of data stories below
Keep it simple on page 2322
Use 'Fit to' in your dashboards on page 2323
Plan for fast load times on page 2324

What's your story's purpose?

Before you start to build your story, take some time to think about the purpose of your story and what you want your viewers' journey to be. Is it a call to action, is it a simple narrative, or are you presenting a case?

If you're presenting a case, decide whether you want to present data points that lead up to a conclusion at the end, or start with a conclusion then show the supporting data points. The latter approach works well for a busy audience.

Finally, sketching out your story first on paper or a whiteboard can help you quickly identify problems with your sequence.

The seven types of data stories

When you use the story feature, you are building a sequence of points. Each point can contain a view, dashboard, or even just text. Some stories show the same view throughout the story, with text annotations and different filters applied to different points to support the narrative arc.

The following table describes seven different data story approaches you can take and provides an example for each. Each data story type is also illustrated in the Data Story Examples Workbook on Tableau Public. A single story can also use more than one approach—see Example - A Story That Examines a Trend on page 2334.
<table>
<thead>
<tr>
<th>Data Story Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Change Over Time</strong></td>
<td><strong>What it does</strong>: Uses a chronology to illustrate a trend.</td>
</tr>
<tr>
<td></td>
<td><strong>Discussions it starts</strong>: Why did this happen, or why does it keep happening? What can we do prevent or make this happen?</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong>: Arsenal's Injury Crisis</td>
</tr>
<tr>
<td><strong>Drill Down</strong></td>
<td><strong>What it does</strong>: Sets context so that your audience better understands what's going on in a particular category.</td>
</tr>
<tr>
<td></td>
<td><strong>Discussions it starts</strong>: Why is this person, place or thing different? How does the performance of this person, place, or thing compare?</td>
</tr>
<tr>
<td></td>
<td><strong>Examples</strong>: Tell Me About Will, The Simpsons Vizipedia</td>
</tr>
<tr>
<td><strong>Zoom Out</strong></td>
<td><strong>What it does</strong>: Describes how something your audience cares about relates to the bigger picture.</td>
</tr>
<tr>
<td></td>
<td><strong>Discussion it starts</strong>: How does something you care about compare to the bigger picture? What effect does one area have on the bigger picture?</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong>: Vancouver Cyclists</td>
</tr>
<tr>
<td><strong>Contrast</strong></td>
<td><strong>What it does</strong>: Shows how two or more subjects differ.</td>
</tr>
<tr>
<td></td>
<td><strong>Discussions it starts</strong>: Why are these items different? How can we make A perform like B? Which area should we focus on and which area is doing fine?</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong>: The Pyramids of Egypt</td>
</tr>
</tbody>
</table>
| **Intersections** | **What it does:** Highlights important shifts when one category overtakes another.  

**Discussions it starts:** What causes these shifts? Are these shifts good or bad? How do these shifts affect other aspects of our plan?  

**Example:** US vs. THEM |

| **Factors** | **What it does:** Explains a subject by dividing it into types or categories.  

**Discussions it starts:** Is there a particular category we should focus on more? How much do these items affect the metric we care about?  

**Example:** Planet Earth |

| **Outliers** | **What it does:** Shows anomalies or where things are exceptionally different.  

**Discussions it starts:** Why is this item different?  

**Example:** SOS Children's Villages |

---

**Keep it simple**

A common error is trying to cram too many views and dashboards into a single story. The result is too many points for your viewers to take in.

The clarity of each story point is also important. Take a step back and consider your story from the perspective of someone who's never seen it. Every element should serve a purpose. If captions, titles, legends, or grid lines aren't necessary, get rid of them!

| **Before** | **After** |
Use 'Fit to' in your dashboards

Dashboards are a common ingredient in Tableau stories. For dashboards that you plan to include in your story, you can use the Fit to option under Size on the Dashboard pane. It will resize your dashboard so that it's the right size for the story you're creating.
Plan for fast load times

The most wonderful story in the world won’t have much impact if it takes too long to load once it’s published. People find long waits frustrating.

Filtering is a common culprit for slow load times. Although filters are effective in restricting the amount of data being analyzed, they also impact query performance. For example, Exclude filters tend to be slower than Keep Only filters. This is because Exclude filters load all of the data for a dimension instead of just what you want to keep. See Filter Your Data Carefully on page 2445 in the Tableau Desktop online help for more details. Knowing Tableau’s Order of Operations on page 267 can also shave time off your load times.
Some of the most critical performance decisions you make as an author begin before you even create your first view or story, in the data preparation stage. Take a moment to familiarize yourself with the data you’re working with. See Know Your Data at the Database Level on page 2438 and Test Your Data and Use Extracts on page 2440 in the Tableau Desktop online help for tips on what to look for and smart ways to work with your data.

Create a Story

Use stories to make your case more compelling by showing how facts are connected, and how decisions relate to outcomes. You can then publish your story to the web, or present it to an audience.

Each story point can be based on a different view or dashboard, or the entire story can be based on the same visualization seen at different stages, with different filters and annotations.

In this article

- Create a story point below
- Explore layout options on page 2329
- Format a story on page 2330
- Delete a story point on page 2333
- Present your story on page 2333

Create a story point

1. Click the New Story tab.

Tableau opens a new story as your starting point:
2. In the lower-left corner of the screen, choose a size for your story. Choose from one of the predefined sizes, or set a custom size, in pixels:
**Note:** Choose the size your story will be viewed at, not the size you're authoring in.

3. By default, your story gets its title from the sheet name. To edit it, right-click the sheet tab, and choose **Rename Sheet**.

   If you're using Tableau Desktop, you can also rename a story by double-clicking the title.

4. To start building your story, double-click a sheet on the left to add it to a story point.

   In Tableau Desktop, you can also drag sheets into your story point.
When you add a sheet to a story point, that sheet remains connected to the original sheet. If you modify the original sheet, your changes will automatically be reflected on the story points that use it.

If you are using Tableau Server or Tableau Online to author on the web and the original sheet has **Pause Auto Updates** enabled, the story sheet will be blank until auto-updates are resumed.

5. Click **Add a caption** to summarize the story point.

In Tableau Desktop, you can highlight a key takeaway for your viewers by dragging a text object to the story worksheet and typing a comment.

6. To further highlight the main idea of this story point, you can change a filter or sort on a field in the view. Then save your changes by clicking **Update** on the story toolbar above the navigator box:
7. Add another story point by doing one of the following:
   - Click **Blank** to use a fresh sheet for the next story point.
     
     ![Story and Layout tabs with options]
     
     - Start customizing a story point and click **Save as New** on the toolbar above the navigator box.
     
     ![Save as New button]
     
     - Click **Duplicate** to use the current story point as the basis for a new one.
     
   **Explore layout options**

   You can refine the look of your story using the options on the **Layout** tab.

   1. Click the **Layout** tab.
   2. Choose a navigator style that best suits your story, and show or hide the next and previous arrows.
Format a story

Resize captions (Tableau Desktop only)

Sometimes the text in one or more of your captions is too long to fit inside the height of the navigator. In this case, you can re-size the captions vertically and horizontally.

1. In the navigator, select a caption.
2. Drag the border left or right to resize the caption horizontally, down to resize vertically, or select a corner and drag diagonally to resize the caption both horizontally and vertically.

All captions in the navigator update to the new size.

Fit a dashboard to a story

You can fit a dashboard to the exact size of a story. For example, if your story is exactly 800 by 600 pixels, you can shrink or expand a dashboard to fit inside that space.

Click the **Size** drop-down menu and select the story you want the dashboard to fit inside.
Format a story's shading, title, and text objects (Tableau Desktop only)

To open the Format Story pane, select Format > Story.
Clear all formatting (Tableau Desktop only)

- To reset a story to its default format settings, click the **Clear** button at the bottom of the **Format Story** pane.

- To clear a single format setting, right-click (Windows) or control-click (macOS) the format setting you want to undo in the **Format Story** pane. Then select **Clear**.

  For example, if you want to clear the alignment of the story title, right-click (control-click...
on Mac) **Alignment** in the **Title** section, and then select **Clear**.

Delete a story point

Click the X in the toolbar above the point's caption:

Present your story

1. In Tableau Desktop, click the **Presentation Mode** button on the toolbar. Or, publish the story to Tableau Online or Tableau Server, and click the **Full Screen** button in the upper-right corner of the browser.
2. To step through your story, click the arrow to the right of the story points. Or, in Tableau Desktop, use the arrow keys on your keyboard.

3. To exit Presentation or Full Screen mode, press Esc.

Example - A Story That Examines a Trend

The example in this article walks you through building a story about earthquake trends over time.

The story feature in Tableau is a great way to showcase this type of analysis because it has a step-by-step format which lets you move your audience through time.

Rather than showing you how to create all the views and dashboards from scratch, this example starts from an existing workbook. What you’ll do is pull the story together. To follow along and access the pre-built views and dashboards, download the following workbook from Tableau Public: An Earthquake Trend Story.

In this article

- Frame the story below
- Build the story on the next page
- Create a story worksheet on the next page
- State the question on page 2336
- Start big on page 2336
- Drill down on page 2340
- Highlight outliers on page 2342
- Show a trend on page 2344
- Offer your analysis on page 2346
- Answer the question on page 2347

Frame the story

A successful story is well-framed, meaning its purpose is clear. In this example, the story’s purpose is to answer the following question: Are big earthquakes becoming more common?

There are several approaches you could take—see Best Practices for Telling Great Stories on page 2320 for a list—but the one used here as an overall approach is Change over
Time, because it works especially well for answering questions about trends. As you build the story you'll notice that other data story types, such as Drill Down and Outliers, are blended in to support the overall approach.

Build the story

Create a story worksheet

1. Use Tableau Desktop to open the Earthquake Trend Story workbook that you downloaded.

   If you also have Tableau Server or Tableau Online and you want to do your authoring on the web instead of in Tableau Desktop, publish the workbook to your Tableau server, click **Workbooks**, select the workbook, then under **Actions** choose **Edit Workbook**.

   Once you open the workbook, you'll see that it has three dashboards. You'll be using those dashboards to build your story. The workbook also has a finished version of the story.

   **Tip:** To see the individual views that are in a dashboard, right-click the dashboard's tab and select **Unhide all Sheets**.

2. Click the **New Story** tab.

   Tableau opens a new worksheet as your starting point.
3. Right-click the **Story 2** tab, choose **Rename Sheet**, and type **Earthquake story** as the worksheet name.

**State the question**

Story titles are in view at all times and they're a handy way to keep your story's purpose front and center. By default, Tableau uses the worksheet name as the story title. In Tableau Desktop you can override that by doing the following:

1. Double-click the title.

2. In the Edit Title dialog box, replace `<Sheet Name>` with the following:

   **Are big earthquakes on the rise?**

3. Click **OK**.

   If you're authoring in Tableau Server or Tableau Online, the story tab is the only place where you can change the title.

**Start big**

To help orient your audience, the first story point you create will show the broadest possible viewpoint—all earthquakes, across the entire planet.
1. On the Story pane, double-click **Map dashboard** to place it on the story sheet. If you’re using Tableau Desktop, you can also use drag-and-drop to add views and dashboards to a story sheet.

Notice how there’s a horizontal scroll bar and the legend isn’t fully displayed.
There's a special setting you can use on your dashboards to prevent this from happening.

2. Select **Map dashboard** and under **Size** on the Dashboard pane, select **Fit to Earthquake story**. This setting is designed to make dashboards the perfect size for a story.
Look at the Earthquake story again. You see that its size has been adjusted and the scroll bars are gone.

3. If you're using Tableau Desktop, add a description for this story point, such as *Exactly 131,834 earthquakes of magnitude 4.0 or greater have been recorded since 1973.*

4. Add caption text by clicking the area that reads **Write the story point description text here.**

5. Click **Update** on the caption to save your changes to the story point.
Drill down

Just like the plot of a good novel needs to move the action along, so does a data story. Starting with your next story point, you’ll use the drill-down technique in order to narrow down the scope of the story and keep the narrative moving.

1. To use your first story point as a baseline for your next, click **Duplicate** under **New Storypoint** on the left.

2. Change the **Magnitude** filter to **7.000 – 9.100** so that the map filters out smaller earthquakes. The map pans to show the Pacific "Ring of Fire," where the majority of the large earthquakes occurred.
3. Add a caption, such as *About two quakes each year qualify as “major”*

4. If you're using Tableau Desktop, edit the description to describe what you've done in this story point. For example: *Out of over 130,000 earthquakes since 2004, only 174 were of magnitude 7.0 or greater—about two major earthquakes each year. But many people wonder, "Are earthquakes happening more often?"*

5. Click **Update** in the story toolbar above the caption to save your changes.

In the next story point, you’re going to drill down further, narrowing the story's focus so that a specific type of earthquake—the "megaquake"—comes into view.

1. Click **Duplicate** in your second story point to use it as the baseline for your third story point.

2. Change the **Magnitude** filter to **8.000 – 9.100** so that the map filters out everything except the megaquakes.

3. Add the caption and description text.
• Caption: These megaquakes have drawn a lot of attention

• Description (Tableau Desktop only): Recent megaquakes of magnitude 8.0 and higher have often caused significant damage and loss of life. The undersea megaquakes near Indonesia and Japan also caused tsunamis that have killed many thousands of people.

4. Click **Update** to save your changes.

![Map showing Earthquakes](image)

Highlight outliers

In the next two story points, you’re going to further engage your audience by examining data points at the far end of the scale: the two most deadly earthquakes in recent history.

1. As you’ve done before, use **Duplicate** to create a new story point as your starting point.
2. Adjust **Magnitude** to **9.000–9.100** and you’ll see just two data points.
3. Select one of the marks, such as the Indian Ocean earthquake and tsunami of 2004 that
had a magnitude of 9.1.

4. Use the pan tool on the maps menu to center it in your story point.

![Map with震中位置](image)

5. Add caption and description text. For example:
   - Caption: *The Indian Ocean earthquake and tsunami of 2004*
   - Description (Tableau Desktop only): *The 2004 Indian Ocean earthquake was an undersea megathrust earthquake that occurred on December 26, 2004. It is the third largest earthquake ever recorded and had the longest duration of faulting ever observed, between 8.3 and 10 minutes.*

6. Click **Update** to save your changes.

7. Repeat the preceding steps for the Japanese earthquake and tsunami of 2011, using the following as caption and description text.
   - Caption: *The Japanese earthquake and tsunami of 2011*
   - Description (Tableau Desktop only): *The 2011 quake off the coast of Tōhoku was a magnitude 9.0 undersea megathrust earthquake. It was the most powerful known earthquake ever to have hit Japan, and the 5th most powerful earthquake ever recorded.*

Notice that you've already created a compelling visual story using just a single dashboard—all by filtering the data and zooming and panning the map.
We still haven’t answered the key question, however: Are big earthquakes on the rise? The next story points will dig in to that angle.

Show a trend

In the next story point, you’ll switch to a line chart (the Timeline dashboard) to show your audience a trend you spotted when you were initially creating views and dashboards.
1. Switch from the story you’re building to **Timeline dashboard**.

![Timeline dashboard]

2. On the Timeline dashboard, set size to **Fit to Earthquake story**.

3. Go back to your story and click **Blank** to create a fresh story point.

4. Double-click the **Timeline dashboard** to add it to your story sheet.

   More earthquakes are being reported over time since 1973. In fact, it’s increased significantly!

5. Add a caption, such as: **More and more earthquakes are being detected**

6. Use **Drag to add text** to add a description of the trend (Tableau Desktop only): **Since 1973, there’s been a steady increase in the number of earthquakes recorded. Since 2003, the trend has accelerated.**
Offer your analysis

From your earlier work in this story with the Map dashboard you know that there are regional differences in earthquake frequency. In your next story point, you'll pull in the Timeline by region dashboard, which breaks out earthquakes by region, and adds trend lines, which help reduce the variability in the data.

1. Click Blank to create a new story sheet.

2. Double-click the Timeline by region dashboard to the story sheet. The APAC region clearly stands out.

3. Add a caption then use Drag to add text to add a comment that points out the large number of earthquakes in the APAC region.
   - Caption: Especially on the eastern side of the Pacific Rim
   - Description (Tableau Desktop only): A rough categorization of earthquakes into
geographic regions (by longitude) shows that the most significant increase in recorded earthquakes has occurred around the Pacific Rim.

Answer the question

Thus far, your data story has concluded that earthquake frequency in the Pacific Rim has increased since 1973, but your original question was about whether big earthquakes are becoming more frequent.

To answer this question, in your final story point, you'll filter out weaker earthquakes and see what the resulting trend line is.

1. Click **Duplicate** to create a new story sheet.
2. Set the **Magnitude** filter to **5.000–9.100**. Notice how the trend lines have flattened out but there's still a slight increase.
3. Add a caption then use **Drag to add text** to add your answer to the story point.
Caption: *But the trend in big quakes is not as clear*

Description (Tableau Desktop only): *It appears that big earthquakes are increasing slightly. There should be more investigation, however, on whether this trend is real or the result of a small number of exceptionally strong recent earthquakes.*

As is often the case with a data story, the story ends with additional questions.

Yes, there’s a trend, but it’s slight. More big earthquakes (magnitude 5.000 - 9.100) have been reported in recent years, especially in the Asia-Pacific region, but could that be natural variation? That might be a good topic for another story.
Format Your Work

Formatting is an important part of both your analysis and presentation. You can format almost everything you see on a worksheet including the fonts, shading, alignment, borders, and graph lines. For example, in a text table you may want to add banded shading to help you visually separate consecutive groups of rows or columns. In a scatter view with reference lines you may want to change the line thickness and color. All of these settings can be changed using the Format window.

Most often you will want to specify format settings for the entire worksheet, all rows, or all columns. However, Tableau also allows you to format individual parts of the view as well. For example, you can format specific fields, resize the cells and the table, and edit individual axes.

Visual Best Practices

The fonts, colors, shading, alignment, borders, and grid lines in your visualization are important parts of both your analysis and the story you’re telling. Tableau products are designed so that you can create great-looking visualizations that use visual best practices by default, freeing you from the need to think about things like fonts and colors—unless you want to.

If you do want to customize, you can control the look of almost everything you see on a worksheet. This article outlines visual best practices and tips to keep in mind while you customize, from ideal workflow to how to get the most out of tooltips.

In this article

- **Format from largest to smallest** on the next page
- **Change color with purpose** on the next page
- **Limit colors** on page 2352
- **Use fonts that optimize online readability** on page 2353
- **Reinforce your story with tooltips** on page 2354
- **Consider your axes** on page 2356
Format from largest to smallest

As you change the look and feel of your work, use a "biggest to smallest" workflow. Start by formatting fonts and titles at the workbook level, then move on to the worksheet level. Save formatting the individual parts of a view for last. A workbook is the largest possible "container" for formatting changes and making changes at that level first will save you time.

See Format at the Workbook Level on page 2356, Format at the Worksheet Level on page 2361, and Format Text and Numbers on page 2370 for details.

Change color with purpose

If you want to change the colors used by your visualization, Tableau makes it easy: just click the Color card, then Edit Colors. For example, maybe you want to emphasize a key finding. You can use neutral colors with a single, bright color to highlight what you want your viewers to pay attention to. If you want to match your company’s brand, you can create a custom palette. See Create Custom Color Palettes on page 2411 for steps.

When you change colors, keep the following best practices in mind..
Discrete fields and categorical palettes

Discrete fields are ones where the field's values are unique. One example of this is customer names. Tableau automatically associates discrete fields with *categorical* palettes, which have colors that are designed to be distinct from one another yet also work well together in the visualization as a whole.

![Image of Edit Colors window]

Continuous fields and quantitative palettes

With continuous fields, the field's values are part of a whole (such as sales over time). Tableau automatically associates these fields with *quantitative* palettes, where colors are along a continuum or range. If the field's values are positive, a single-color range is best:

![Image of SUM(Sales) chart]

If there are both positive and negative values, choose a two-color range:
Limit colors

Color used correctly can enhance analysis. Too many colors can create visual overload for your users and impede analysis. In the view below, the stacked bars represent cell phone subscriptions and the trend line is internet usage over time. In the first image, Years is on the Colors shelf. Notice how hard it is to pick out the trend line among all the other colors and how easy it is to spot the trend line in the second image, where there are only two colors.
Use fonts that optimize online readability

The fonts in the Tableau typeface were designed to be paired with data visualizations and are optimized for legibility at small sizes. Tableau visualizations use them by default.
If you need to use a different font, choose one that optimizes readability online, such as the following:

- Arial
- Trebuchet MS
- Verdana
- Times New Roman
- Lucida sans

Reinforce your story with tooltips

A person looking at your dashboard will intuitively use his or her mouse to explore marks, and this makes tooltips appear. Tooltips appear by default for most views and they are a great way to reinforce your data story. You can customize your tooltips by clicking **Worksheet > Tooltip**.

For example, here’s a tooltip for a view on pageviews for a web site:
Here's an example of how the above tooltip could be rewritten and formatted to highlight what's important to the people looking at this view:

You can also rewrite your tooltips to tell a mini-story. For example, here's a tooltip about the number of oil rigs in Italy.
Below, the tooltip is rewritten as a phrase and key elements have been put in bold to draw the viewer's attention.

![As of 10/15/2002, Italy had 13 oil rigs](image)

Consider your axes

By default, an axis range in Tableau will automatically adjust, based on the data in your visualization. Filtering and other actions can change the axis range. If you are trying to create a specific comparison between two views, changing axes ranges can make analysis difficult for your users. If this is the case, you can set your axes to a specific, fixed range. For steps on how to do this, see Examples: Use Different Axis Ranges (Uniform, Independent, Fixed) on page 1852.

If you’re working with data that has a very large range, a fixed axis may not be practical. If this is the case, add grid lines to your visualization. Grid lines can help your users stay oriented when the axis automatically adjusts. See Format lines on page 2367 for more information.

Format at the Workbook Level

You can quickly change how fonts, titles, and lines look in every view in a workbook by specifying format settings at the workbook level, instead of the worksheet level.

For example, you might want to use a specific font, size, and color so that all views adhere to your company’s brand. You might also want to remove grid lines from your views—or make them more noticeable by increasing their pixel size or color.

You can also change the theme used by your workbook. Themes control items like the default font, colors, and line thickness. When you create a new workbook, it automatically uses the Default theme, which uses visual best practices.
In this article

- Change lines in your workbook below
- Change fonts in your workbook on the next page
- Upgrade or change your workbook theme (Tableau Desktop only) on page 2359
- Reset a workbook to its default settings on page 2360

Change lines in your workbook

You can change every type of line for all views in your workbook, or you can change just certain types of lines, such as trend lines. You can also turn off certain types of lines, including grid lines.

When you make changes to your workbook's line settings, a gray dot appears next to the setting in the Format Workbook pane. You can quickly switch back to default settings using the Reset to Defaults button.

In Tableau, line opacity is linked to line color. If you set a line's opacity at the workbook level, all worksheets in the workbook will receive the workbook's color setting for that line as well. If you need to change line opacity and maintain different color settings for a line type, change opacity at the worksheet level instead of at the workbook level.

1. On the Format menu, select Workbook.

2. The Format Workbook pane replaces the Data pane on the left and provides a series
of drop-down lists where you can change all line settings in a workbook.

To confirm that your change has been applied to every view in your workbook, you can use thumbnails. Right-click a thumbnail, and select **Refresh All Thumbnails**.

**Change fonts in your workbook**

You can change all fonts in your workbook or you can change fonts for only certain areas, such as just worksheet titles.

1. On the **Format** menu, select **Workbook**.

2. The **Format Workbook** pane replaces the **Data** pane on the left and provides a series of drop-down lists where you can change all font settings in a workbook, as well as the font settings for titles of worksheets, stories, and dashboards.
If you have made font changes at the worksheet level, such as on a filter card or a worksheet title, changing the font at the workbook level will overwrite those changes.

Upgrade or change your workbook theme (Tableau Desktop only)

If your workbook is set to a theme other than Default, it will retain that theme when you upgrade from one version of Tableau to another, but you can easily change the theme to Default. Conversely, you can change your workbook's theme from Default to a different, earlier theme.
If you upgraded from version 9.3 to version 10.x and your theme was set to Default in version 9.3, it will be set to Previous in version 10.x. After you switch to the Default theme, check your workbook to confirm that items are sized appropriately. You might need to make some manual adjustments.

1. On the **Format** menu, select **Workbook Theme**.
2. Choose a theme.

![Workbook Theme Menu](image)

Each theme is associated with a different version of Tableau Desktop.

<table>
<thead>
<tr>
<th>This workbook theme...</th>
<th>Is associated with these Tableau versions...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Version 10.x</td>
</tr>
<tr>
<td>Previous</td>
<td>Versions 8.0.x – 9.3.x</td>
</tr>
<tr>
<td>Modern</td>
<td>Versions 3.5 – 7.0.x</td>
</tr>
<tr>
<td>Classic</td>
<td>Versions 1.0 – 3.2</td>
</tr>
</tbody>
</table>

**Reset a workbook to its default settings**

When you make changes to your workbook's font settings, a gray dot appears next to the setting in the **Format Workbook** pane. You can quickly switch back to default settings using the **Reset to Defaults** button.
1. On the Format menu, select Workbook.

2. In the Format Workbook pane, click Reset to Defaults.

Format at the Worksheet Level

You can format settings for fonts, alignment, shading, borders, lines and tooltips at the worksheet level. For example, you might want to remove all the borders in a text table, or add shading to every other column in a view.

When you make formatting changes at this level, they apply only to the view you’re working on. See Format at the Workbook Level on page 2356 for how to make changes that apply to every view in your workbook.
In this article

Access worksheet formatting settings below
Format fonts below
Format text alignment on the next page
Format shading on page 2365
Format borders on page 2366
Format lines on page 2367
Format highlighters on page 2368
Format a filter card on page 2368
Format a parameter control card on page 2369
Copy and paste worksheet formatting on page 2370

Access worksheet formatting settings

1. Display a worksheet or dashboard.

2. From the Format menu, choose the part of the view that you want to format, such as Font, Borders, or Filters.

Format fonts

For a view, you can specify the font, style, size, and color for either the pane text or header text, or both. For example, in the view below, the header text is set to use the Tableau Bold font.
If you have totals or grand totals in the view, you can specify special font settings to make these values stand out from the rest of the data. This is particularly useful when you are working with a text table. The view below shows a text table in which the grand totals are formatted to be dark red.

Format text alignment

Tableau uses visual best practices to determine how text is aligned in a view, but you can also customize text. For example, you can change the direction of header text so that it is horizontal.
(normal) instead of vertical (up).

**Note:** Tableau adheres to regional standards when determining when to begin or end line breaks.

For each text area you can specify the following alignment options:

**Horizontal** - Controls whether text aligns to the left, right, or center.

**Vertical Alignment** - Controls whether text aligns at the top, middle, or bottom.

**Direction** - Rotates text so that it runs horizontally (normal), top-to-bottom (up), or bottom-to-top (down).

**Wrap** - Controls whether long headers wrap to the next line or are abbreviated, but does not control text marks.
Note: If cells are not large enough to show more than one row of text, turning on wrapping will have no visible effect. If this happens, you can hover the cursor over a cell until a double-sided arrow appears, and then click and drag down to expand the cell size.

Format shading

Shading settings control the background color of the pane and headers for totals, grand totals, as well as for the worksheet areas outside those areas.

You can also use shading to add banding, alternating color from row to row or column to column. Banding is useful for text tables because the alternating shading helps your eye distinguish between consecutive rows or columns.
## Table with Banding

<table>
<thead>
<tr>
<th>Street Name</th>
<th>Row 1</th>
<th>Row 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton St &amp; Washington Blvd</td>
<td>1,033</td>
<td>2,873</td>
</tr>
<tr>
<td>Canal St &amp; Jackson Blvd</td>
<td>942</td>
<td>2,355</td>
</tr>
<tr>
<td>Clinton St &amp; Madison St</td>
<td>994</td>
<td>1,946</td>
</tr>
<tr>
<td>Franklin St &amp; Jackson Blvd</td>
<td>1,649</td>
<td>1,829</td>
</tr>
<tr>
<td>Daley Center Plaza</td>
<td>1,147</td>
<td>1,772</td>
</tr>
<tr>
<td>Michigan Ave &amp; Lake St</td>
<td>2,919</td>
<td>1,522</td>
</tr>
<tr>
<td>Dearborn St &amp; Adams St</td>
<td>1,456</td>
<td>1,464</td>
</tr>
<tr>
<td>Clark St &amp; Randolph St</td>
<td>956</td>
<td>1,447</td>
</tr>
<tr>
<td>Clinton St &amp; Lake St</td>
<td>701</td>
<td>1,370</td>
</tr>
<tr>
<td>Dearborn St &amp; Monroe St</td>
<td>838</td>
<td>1,353</td>
</tr>
<tr>
<td>State St &amp; Kinzie St</td>
<td>1,957</td>
<td>1,304</td>
</tr>
<tr>
<td>Kingsbury St &amp; Erie St</td>
<td>877</td>
<td>1,049</td>
</tr>
</tbody>
</table>

For row and column banding, you can use the following options:

- **Pane and Header**: The color the bands use.
- **Band Size**: The thickness of the bands.
- **Level**: If you have nested tables with multiple dimensions on the rows and columns shelves, this option allows you to add banding at a particular level.

### Format borders

Borders are the lines that surround the table, pane, cells, and headers in a view. You can specify the border style, width, and color for the cell, pane, and header areas. Additionally, you
can format the row and column dividers. For example, in this view the **Row Divider** borders are formatted to use an orange color:

Row and column dividers serve to visually break up a view and are most commonly used in nested text tables. You can modify the style, width, color, and level of the borders that divide each row or each column using the row and column divider drop-downs. The level refers to the header level you want to divide by.

**Format lines**

You can control the appearance of the lines that are part of the view, such as grid lines and zero lines, as well as lines that help you inspect data, such as trend lines, reference lines, and drop lines.

For example, you can set trend lines to use a red color and an increased thickness:
Format highlighters

The highlighter on your worksheet can be formatted to use a different font, style, color, background color, font size, and border. Formatting highlighters allows you to better integrate them into your dashboard or worksheet style. You can also edit the title that displays on each highlighter that shows in the view.

For more information about using highlighters, see Highlight Data Points in Context on page 1800.

Format a filter card

Filter cards contain controls that let users interact with your view. You can change filter cards to use custom formatting. For example, the body text in the filters below is formatted to use the Tableau Bold font, in aqua.
**Note:** For filters and parameters, title formatting appears only on dashboards or in views published to the web.

**Format a parameter control card**

Parameter controls are similar to filter cards in that they contain controls that let users modify the view. If you create a parameter control, you can customize how it looks. For example, in the view below, the Sales Range parameter is formatted so that the sales amount appears in orange.
Copy and paste worksheet formatting

After you format a worksheet, you can copy its formatting settings and paste them into other worksheets. The settings that you can copy are anything you can set in the Format pane, with the exception of reference lines and annotations. Adjustments like manual sizing and level of zoom are not copied.

1. Select the worksheet from which you want to copy formatting.
2. Right-click (control-click on Mac) the worksheet tab and select Copy Formatting.
3. Select the worksheet you want to paste the formatting into.
4. Right-click (control-click on Mac) the worksheet tab and select Paste Formatting.

Format Text and Numbers

To learn how to format specific items, click the links below.

Format Text

The proper font selection can take your visualization from good to great, and Tableau is here to help. Whether you're changing the font size, position, color, or direction, Tableau provides a wide range of customization options for your text.

When you make formatting changes at this level, they apply only to the view you're working on. See Format at the Workbook Level on page 2356 for how to make changes that apply to every view in your workbook.

To access worksheet-level format settings, select the Format menu, then choose the part of the view, such as Font, or Border, that you want to format.

In this article

Format fonts on the next page
Format text alignment on page 2378
Format tooltips on page 2379
Edit worksheet titles, captions, and legend titles on page 2380
Format fonts

Select **Format > Font**. A **Format Font** control pane will open. This control pane will let you customize the text in the view. 

**Click the letters below for more information.**

![Format Font pane](image)

A. Worksheet

Editing the font at the worksheet level applies the changes you make to all text fields in view except the Header and Tooltip fields.

**Note:** Tableau treats Worksheet formatting as a new default, and any changes to any other fields (such as Title, Pane, or Header) will overwrite the Worksheet format.
B. Pane

Adjusting the font here will effect the pane view, which is where Tableau displays the results of your visualizations.
Select this option to adjust font for the headers in your view. Headers display the dimensions used in your viz.

D. Tooltip

Adjusting the font here will apply changes to all the text in the tooltip, the field that pops up when someone hovers over a mark in your view. If you want more control over the fields in the tooltip, see Format tooltips on page 2379
E. Title

Select this formatting option to quickly adjust the appearance of the Title. For more information, see *Edit worksheet titles, captions, and legend titles* on page 2380.

F. Total Pane
Adjusting the font for totals at the pane level will change both the subtotals and grand totals. For information on setting up totals, see Show Totals in a Visualization on page 1754.

G. Total Header

Adjusting the font for totals at the header changes the appearance of the Total and Grand Total label in the view.
### H. Grand Total Pane

Select this option to adjust the font for the grand total result in your view.

![Grand Total Pane Diagram](image)

### I. Grand Total Header

Select this option to adjust the font for the grand total label in your view.

![Grand Total Header Diagram](image)
J. Row and Column Text Formatting

If you have dimensions on both the row and column shelves, you can independently format the fonts through the **Rows** and **Columns** tabs on the Formatting pane.

The format font options are limited to the Header, Total, and Grand Total selections.
Format text alignment

Tableau uses visual best practices to determine how text is aligned in a view, but you can also customize text. For example, you can change the direction of header text so that it is horizontal (default) instead of vertical (up).

To format the text alignment, select **Format > Alignment** to open the text alignment pane.

For each text area you can specify the following alignment options:

- **Horizontal** - Controls whether text aligns to the left, right, or center.
- **Vertical Alignment** - Controls whether text aligns at the top, middle, or bottom.
- **Direction** - Rotates text so that it runs horizontally (default), top-to-bottom (up), or bottom-to-top (down).
**Wrap** - Controls whether long headers wrap to the next line or are abbreviated. It does not control text marks in the visualization.

**Note:** If cells are not large enough to show more than one row of text, turning on wrapping will have no visible effect. If this happens, you can hover the cursor over a cell until a double-sided arrow appears, and then click and drag down to expand the cell size.

**Note:** Tableau adheres to regional standards when determining when to begin or end line breaks.

**Format tooltips**

Tooltips are details that appear when you rest the pointer over one or more marks in the view. Tooltips also offer convenient tools to quickly filter or remove a selection, select marks that have the same value or view underlying data. You can edit the tooltip to include both static and dynamic text. You can also modify which fields are included in the automatic tooltip.

**To edit a tooltip:**

1. Select **Worksheet > Tooltip**. Tooltips are specified for each sheet and can be formatted using the formatting tools along the top of the Edit Tooltip dialog box. Use the **Insert** menu at the top of the dialog box to add dynamic text such as field values, sheet properties, and more.
2. Use the formatting tools along the top of the Edit Tooltip dialog box.

3. (Optional) Use the Insert drop-down list at the top of the dialog box to add dynamic text such as field values, sheet properties, and more.

   The All Fields option on the Insert menu adds all field names and values that are used in the view. Inserting the All Fields parameter updates the tooltip as you change the view. You can also use the All Fields option to exclude fields.

4. (Optional) Select the Include command buttons option to show filtering and view data options in the tooltip. For example, including command buttons will add Keep Only, Exclude, and View Data buttons to the bottom of the tooltip. These command buttons are available both in Tableau Desktop and when the view is published to the web or viewed on a mobile device.

5. (Optional) Select the Allow selection by category check box to select marks in a view that have the same value by clicking on a discrete field in a tooltip. For more information see Tooltips on page 214.

Edit worksheet titles, captions, and legend titles

1. Do one of the following:
   - On Tableau Server or Tableau Online web editing, double click the item you want to change.
   - On Tableau Desktop, Right-click (control-click on Mac) the item you want to
change and select **Edit <item>**, for example, **Edit Title**.

- On a worksheet, hover on the title, click the drop-down arrow on the right-hand side and select **Edit Title** or **Edit Caption** from the context menu.

2. In the Edit dialog box, modify the text and format the font, size, style, color, and alignment. Use the **Insert** menu to add dynamic text such as sheet properties and field values. Click **OK**.

![Edit Title dialog box](image)

To reset the title back to the default, click **Reset**.

**Use Custom Fonts**

Tableau provides a variety of fonts for you to use in your visualizations, including the fonts in its default typeface: Tableau. However you might want to use a font other than what's provided. For example, you may want to use a font that's unique to your company's brand.

To use a custom font in Tableau, the first step is to install it on the computers that are running Tableau. This includes Tableau Desktop and, if you're publishing to Tableau Server, any computers running Tableau Server. To ensure that Tableau Server can render the font correctly, make sure the font is "web safe" in all browsers. If a font specified in Tableau Desktop isn't also installed on Tableau Server, Tableau Server uses a default font.
Note: If you download a workbook from Tableau Server, custom fonts are not downloaded with the workbook.

For more information about installing fonts, see the following links:

- How to install or remove a font in Windows (Microsoft Knowledge Base)
- Mac Basics: Font Book (Apple Support)

After you install the font, it appears in the Formatting pane in Tableau Desktop, where you can use it to change fonts at the workbook, worksheet, or individual view part levels. See Format at the Workbook Level on page 2356, Format at the Worksheet Level on page 2361, and Format Text and Numbers on page 2370 for details.

Format Titles, Captions, Tooltips, and Legends

You can show titles, captions, and tooltips on any sheet. After you add a title or caption you can edit and format the text as well as modify the shading and border. If a title or caption is not showing, from the top menu, go to the menu for the current sheet type (Worksheet, Dashboard, or Story) and select Show Title or Show Caption.

A legend card appears in the worksheet when you encode marks by dropping them on the Color or Size cards. You can format the legend font, shading, border, and alignment. You can also edit the titles that appear on each legend.

For details on showing a visualization from a worksheet in a tooltip (Viz in Tooltip), Create Views in Tooltips (Viz in Tooltip) on page 1773.

In this article

- Edit worksheet titles, legend titles, or captions (Tableau Desktop only) on the next page
- Format borders and shading for a title, legend, or caption (Tableau Desktop only) on the next page
- Edit dashboard titles on page 2384
- Format tooltips (Tableau Desktop only) on page 2385
Edit worksheet titles, legend titles, or captions (Tableau Desktop only)

1. Do one of the following:
   - If you are editing a worksheet title on Tableau Server or Tableau Online web editing, double click the item you want to change. Worksheet captions and legend titles cannot be edited in web editing.
   - Right-click (control-click on Mac) the item you want to change and select Edit <item>, for example, Edit Title.
   - On a worksheet, hover on the title, click the drop-down arrow on the right-hand side and select Edit Title or Edit Caption from the context menu.

2. In the Edit dialog box, modify the text and format the font, size, style, color, and alignment. Use the Insert menu to add dynamic text such as sheet properties and field values. Click OK.

![Edit Title Dialog Box](image)

To reset the title back to the default, click Reset.

Format borders and shading for a title, legend, or caption (Tableau Desktop only)

1. Right-click (control-click on Mac) the title, caption, or legend and select Format <item>—for example, Format Title.
2. In the **Format** pane, use the drop-down lists to change the default shading and border.

![Format Title and Caption](image)

---

**Edit dashboard titles**

To edit a dashboard title:

1. In a dashboard, double-click the title.

2. In the Edit Title dialog box that appears, format the title as you like.
   
   You can enter a new title for the dashboard or a description, as well as change the font type, size, emphasis, color, and alignment. You can also type in a website URL and Tableau will automatically create a hyperlink.

   **Note**: If you edit the hyperlink, make sure to delete the entire hyperlink and then reenter the new one. If you do not do this, the text of the hyperlink will update, but the URL path will not.

3. When finished, click **OK**.

   The dashboard title updates with the changes.
Format tooltips (Tableau Desktop only)

Tooltips are details that appear when you rest the pointer over one or more marks in the view. Tooltips also offer convenient tools to quickly filter or remove a selection, select marks that have the same value or view underlying data.

Watch a video: To see related concepts demonstrated in Tableau, watch Basic Tooltips, a 7-minute free training video. Use your tableau.com account to sign in.

You can edit the tooltip to include both static and dynamic text. You can also modify which fields are included in the automatic tooltip.

To edit a tooltip:

1. Select Worksheet > Tooltip. Tooltips are specified for each sheet and can be formatted using the formatting tools along the top of the Edit Tooltip dialog box. Use the Insert menu at the top of the dialog box to add dynamic text such as field values, sheet properties, and more.

2. Use the formatting tools along the top of the Edit Tooltip dialog box.

3. (Optional) Use the Insert drop-down list at the top of the dialog box to add dynamic text such as field values, sheet properties, and more.
The All Fields option on the Insert menu adds all field names and values that are used in the view. Inserting the All Fields parameter updates the tooltip as you change the view. You can also use the All Fields option to exclude fields.

4. (Optional) Select the Include command buttons option to show filtering and view data options in the tooltip. For example, including command buttons will add Keep Only, Exclude, and View Data buttons to the bottom of the tooltip. These command buttons are available both in Tableau Desktop and when the view is published to the web or viewed on a mobile device.

5. (Optional) Select the Allow selection by category check box to select marks in a view that have the same value by clicking on a discrete field in a tooltip. For more information see Tooltips on page 214.

Format Fields and Field Labels

You can format a specific field in a view, as well as individual field labels. When formatting is done this way, your formatting changes are scoped to only where the field appears in the view, or only an individual label.

In this article

Format a field below
Format a field label on page 2388

Format a field

In the view below, the Month(Order Date) field has been formatted so that the headers use the Tableau Semibold font, in blue. Notice that the header values along the Profit axis are not affected.
To format a specific field:

1. Right-click (control-click on Mac) the field and select **Format**.

   The **Format** pane opens to settings for the selected field.

2. Make your changes in the **Format** pane.
For discrete fields, such as Region or Customer Name, you can specify font and alignment properties for both header and pane areas. For continuous fields, such as Profit or Sales, you can format font properties for the pane and axis as well as number and tick mark colors. For more information about other axis options, see Edit Axes on page 1838. The view is updated as you make changes so you can quickly see the colors and formats that work with your view.

Format a field label

Field labels are row and column headings that indicate the data fields used to create a view. By default, Tableau shows field labels, but you can choose to hide them. When field labels are showing they appear in three different parts of the view: rows, columns, and the corner. The view below shows an example of each of these types of field labels.

You can format the font, shading, alignment, and separators for each of these types of field labels.

To format a specific field label:

1. Right-click (control-click on Mac) the field label in the view and select Format.
2. In the Format pane, specify the settings of the font, shading, and alignment field labels.

Note: When you have multiple dimensions on the rows or columns shelves, the field labels appear adjacent to each other in the table. Each field label is separated from the others with a forward slash symbol. Specify a different separator in the Format pane.
Format Numbers and Null Values

When you format a measure, you can specify the number format for both the axis and the pane text. You can select from a set of standard formats, such as number, currency, scientific, and percentage; or you can define a custom number format using Microsoft Excel style format codes.

When a measure contains null values, the nulls are usually plotted as zero. You can use formatting, however, to handle the null values in a different way, such as hiding them.

In this article

<table>
<thead>
<tr>
<th>For Tableau Desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a number format below</td>
</tr>
<tr>
<td>Set the default number format for a field on page 2395</td>
</tr>
<tr>
<td>Format a measure as currency on page 2395</td>
</tr>
<tr>
<td>Use locale to specify number formats on page 2397</td>
</tr>
<tr>
<td>Format null values on page 2399</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>For Tableau Server or Tableau Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a number format on page 2401</td>
</tr>
</tbody>
</table>

For Tableau Desktop

Specify a number format

1. Right-click (control-click on Mac) a measure or axis in the view and select **Format**.
2. In the **Format** pane, click the **Numbers** drop-down menu.
3. Select a number format.

Some formats require additional settings. For example, if you select **Scientific**, you must
also specify the number of decimal places.

Here are the number formats and associated options available in Tableau.

<table>
<thead>
<tr>
<th>Number Format</th>
<th>Format Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td>format is automatically selected based on either the format specified by the data source or the data contained in the field.</td>
</tr>
<tr>
<td>Number (Standard)</td>
<td>Locale: number format changes based on the</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Number Format</strong></td>
<td><strong>Format Options</strong></td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>selected.</td>
<td>geographical location selected.</td>
</tr>
<tr>
<td><strong>Number (Custom):</strong> format is customized to your choice.</td>
<td>Decimal Places: the number of decimal places to display.</td>
</tr>
<tr>
<td></td>
<td>Negative Values: how negative values are displayed.</td>
</tr>
<tr>
<td></td>
<td>Units: the number is displayed using the specified units. For example, if the number is 20,000 and the units are thousands, the number will be displayed as 20K.</td>
</tr>
<tr>
<td>Number Format</td>
<td>Format Options</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td></td>
<td>Prefix/Suffix: characters that precede and follow each displayed number. Include thousands separators: whether the number shows separators every thousand (example: 100,000 vs. 100000).</td>
</tr>
<tr>
<td><strong>Currency (Standard):</strong> format and currency symbol is based on locale selected.</td>
<td>Locale: currency format based on the geographical location selected.</td>
</tr>
<tr>
<td><strong>Currency (Custom):</strong> format and currency symbol is</td>
<td>Decimal Places: the number of decimal places to</td>
</tr>
<tr>
<td>Number Format</td>
<td>Format Options</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>customized to your choice.</td>
<td>display.</td>
</tr>
<tr>
<td></td>
<td>Negative Values: how negative values are displayed.</td>
</tr>
<tr>
<td></td>
<td>Units: the number is displayed using the specified units. For example, if the number is 20,000 and the units are thousands, the number is displayed as 20K.</td>
</tr>
<tr>
<td></td>
<td>Prefix/Suffix: characters that precede and follow each displayed number.</td>
</tr>
<tr>
<td></td>
<td>Include thousands</td>
</tr>
<tr>
<td>Number Format</td>
<td>Format Options</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td></td>
<td>separators:  whether the number shows separators every thousand (example: 100,000 vs. 100000).</td>
</tr>
<tr>
<td>Scientific:</td>
<td>Decimal: the number of decimal places to display.</td>
</tr>
<tr>
<td>Percentage:</td>
<td>Decimal: the number of decimal places to display.</td>
</tr>
<tr>
<td>Custom:</td>
<td>Custom: type in the format you want to use. This format can be</td>
</tr>
<tr>
<td>Number Format</td>
<td>Format Options</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>the format options.</td>
<td>specified by a non-conditional Excel style number code. See Number format codes in the Microsoft Knowledge Base for details.</td>
</tr>
</tbody>
</table>

Set the default number format for a field

Right-click (control-click on Mac) the field in the Data pane and select Default Properties > Number Format.

In the subsequent dialog box, specify a number format to be used whenever the field is added to the view. The default number format is saved with the workbook. It is also exported when you export the connection information.

Note: Formatting numbers using the Format pane overrides any number formatting applied elsewhere.

Format a measure as currency

The view in the following image shows profit over time. Notice that the profit figures on the vertical axis are not formatted as currency.
To format the numbers as currency:

1. Right-click the Profit axis and choose Format.

2. On the Axis tab in the Format pane, under Scale, select the Numbers drop-down list, and then select one of the following:

   Currency (Standard) to add a dollar sign and two decimal places to the figures.
Currency (Custom) to specify the number of decimal places, how to show negative values, the units, whether to include a prefix or suffix, and whether to include a separator character.

Use locale to specify number formats

By default, Tableau uses your computer’s locale and language settings to format numbers. But you can explicitly set a different locale in the Format pane.

The following steps show how to set Swiss German currency, using the same view as in the previous section.

1. Right-click the Profit axis and select Format.
2. On the Axis tab, under Scale, select the Numbers drop-down list and then select Currency (Standard).
3. In the **Locale** drop-down list, items appear in a **Language (Country)** format. For this example, select **German (Switzerland)**.

The view updates to show the sales figures in Swiss Francs, formatted for the German language.
**Tip:** You can change the default currency setting so that every time you drag the Profit measure to a view it uses the settings you want. In the Measures pane, right-click Profit (or other monetary measure), and select Default Properties > Number Format. Then format the field as shown above.

Format null values

When a measure contains null values, they are usually plotted in a view as zero. However, sometimes that changes the view and you’d rather just suppress null values altogether. You can format each measure to handle null values in a unique way.

**To format null values for a specific field:**

1. Right-click the field in the view that has the null values (Control-click on a Mac) and choose Format.
2. Go to the Pane tab.
3. In the Special Values area, specify whether to show the null values using an indicator in the lower right corner of the view, plot them at a default value (e.g., zero for number fields), hide the values but connect lines, or hide and break lines to indicate that null values exist.
4. If you specify text in the **Text** field, it will appear in the view for null values when mark labels are turned on. See **Show and Hide Mark Labels** on page 2180.

**Note:** The Special Values area is not available for dimensions or discrete measures.
For Tableau Server or Tableau Online

Specify a number format

When authoring a view on the web, you can specify the number format for a field used in the view.

1. In web editing mode, right-click a measure in the view and select **Format Number**.

2. In the dialog box that appears, select a number format.
Some formats provide additional settings. For example, if you select **Currency**, you can also specify the number of decimal places, as well as the units, and whether or not to include separators, such as commas.

In this example, Sales is formatted as a Currency with zero decimal places and thousand (k) units. Sales numbers in the view update with these settings. Labels and tooltips update as well.
Here are the number formats and associated options available in Tableau.

<table>
<thead>
<tr>
<th>Number Format</th>
<th>Format Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Automatic</strong>:</td>
<td>None.</td>
</tr>
<tr>
<td>format is automatically selected based on either the format specified by the data source or the data contained in the field.</td>
<td></td>
</tr>
</tbody>
</table>

<p>| Number (Custom):     | <strong>Decimal Places</strong>: the number of decimal places to display. |
| format is customized to your choice. | <strong>Units</strong>: the number is displayed using the specified units. For example, if the number is 20,000 and the units are |</p>
<table>
<thead>
<tr>
<th>Number Format</th>
<th>Format Options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>thousands, the number will be displayed as 20K.</td>
</tr>
</tbody>
</table>

**Include separators**: whether the number shows separators every thousand (example: 100,000 vs. 100000).

<table>
<thead>
<tr>
<th>Currency (Custom): format and currency symbol is customized to your choice.</th>
<th>Decimal Places: the number of decimal places to display.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Units</strong>: the number is displayed using the specified units. For example, if</td>
</tr>
<tr>
<td>Number Format</td>
<td>Format Options</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td></td>
<td>the number is 20,000 and the units are thousands, the number is displayed as 20K.</td>
</tr>
<tr>
<td></td>
<td><strong>Include separators</strong>: whether the number shows separators every thousand (example: 100,000 vs. 100000).</td>
</tr>
</tbody>
</table>

<p>| Percentage (Custom): numbers are displayed as a percentage with the percent symbol. The value of 1 is interpreted as 100% and 0 as 0% | Decimal Places: the number of decimal places to display. |</p>
<table>
<thead>
<tr>
<th>Number Format</th>
<th>Format Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scientific (Custom): numbers are displayed in scientific notation.</td>
<td>Decimal Places: the number of decimal places to display.</td>
</tr>
</tbody>
</table>

Resize Tables and Cells

You can change the size of the rows, columns, and cells that compose a table. The best way to resize your table depends on the view type and the table components you want to resize.

In this article

Use commands to resize rows and columns below
Manually resize rows and columns on the next page
Resize an entire table on the next page
Resize cells on page 2408

Use commands to resize rows and columns

1. Select Format > Cell Size.
2. Choose the Taller, Shorter, Wider, or Narrower command. To quickly apply these commands, see Shortcuts for resizing rows and columns (Tableau Desktop) on page 2765.

For example, in the view below, we used the Wider and Taller commands to make the view more readable.
Note: For a given field, all members will have the same width and the same height. You can't resize individual field members.

Manually resize rows and columns

To manually resize the widths or heights of row and column headers or axes:

1. Place your cursor over the vertical or horizontal border of a header or axis.

2. When you see the resize cursor , click and drag the border left and right or up and down.

Resize an entire table

You can increase or decrease the size of the entire table by selecting Bigger or Smaller on the Format > Cell Size menu. For example, to increase the width of the columns and the height of the rows for the view shown below, you can select Format > Cell Size > Bigger.
This option increases both the width and height of the panes in a visually appropriate way. Notice that the size of the row headers increases horizontally when you resize the table.

Resize cells

Any table you can create in Tableau has the cell as its basic component. For a text table, the cell is what you would expect. It is the intersection of a row and a column, and is where the text is displayed.

However, depending on the view you construct, identifying the cell is not always possible or useful, such as in the case of a scatter plot.

Manipulating cells to enhance your data view is useful when dimensions are the inner fields on both the Rows and Columns shelves. In this case, there are two shortcuts you can select on the Format > Cell Size menu:

- **Square Cell** – Adjusts the view so the cell has a 1:1 aspect ratio. This results in a square cell, which is useful for heat maps.
- **Text Cell** – Adjusts the view so the cell has a 3:1 aspect ratio. This is useful for text tables.

For example, you can see the text table below with **Square Cell** and **Text Cell** applied. **Text Cell** enforces a cell aspect ratio of 3:1 and results in a compact table that is easy to read.
The heat map shown below is modified by selecting **Square Cell** on the **Format > Cell Size** menu. This enforces a cell aspect ratio of 1:1 and results in a compact table that is easy to analyze. You can also use the **Size** slider on the **Marks** card to adjust the size of each mark.

After changing the cell size, you can use **Ctrl+B** and **Ctrl+Shift+B** to decrease or increase the table size while maintaining the cell aspect ratio. On a Mac, the keystrokes are **Command-B** and **Shift-Command-B**.

**Define Table Structure**

In addition to the standard formatting, there are some other settings that define the table structure. You can modify these settings by selecting **Analysis > Table Layout > Advanced**
to open the Table Options dialog box. There you can specify the aspect ratio, the default number format, row and column attributes, and the default label orientation for labels along the bottom of the view. While these settings apply to the whole view, some can be overridden using the Format pane.

In this article

Set the aspect ratio below
Set the default number format below
Set row attributes below
Set column attributes on the next page

Set the aspect ratio

The aspect ratio refers to the ratio of the pane width to the pane height. You can choose to constrain the aspect ratio to a specified amount or not constrain it at all. An unconstrained axis range can be useful because it means that the axes don’t have to be the same length.

Anytime you manually resize a row or column, you are unconstraining the aspect ratio. The aspect ratio setting only applies to views containing continuous axes on both the row and column shelves. Nominative axes are not affected by the aspect ratio settings.

Set the default number format

You can define the number of decimal places to appear by default for numbers in the view. If you select Automatic, Tableau automatically decides the number of decimal places based on the data in the field. If you select Manual, you can decide to show up to 16 decimal places.

Set row attributes

Select from the following Row attributes:

- Maximum levels of row labels: determines the number of fields that can be added to the Rows shelf before the headers are combined on the same level.
- Maximum levels of horizontal row labels: determines the number of fields that can be placed on the Rows shelf before headers are automatically oriented vertically rather than horizontally.
Set column attributes

Select from the following column attributes:

- Maximum levels of column labels: determines the number of fields that can be placed on the Columns shelf before Tableau begins to combine the labels.

- Show innermost level at bottom of view when there is a vertical axis: displays the innermost level of column headers at the bottom of the view (as opposed to the top) when a vertical axis is added to the view.

- Default orientation of labels at bottom of view: determines whether labels at the bottom of the view are oriented horizontally or vertically by default. You can toggle between the horizontal and vertical options by pressing Ctrl + L on your keyboard (Command-L on a Mac).

Create Custom Color Palettes

Tableau Desktop comes with color palettes that have been carefully designed to work well together and effectively apply color to data in many situations, such as on maps, heat maps, bar charts, etc. You can also create and use your own custom color palettes by modifying the Preferences.tps file that comes with Tableau Desktop. For example, you can create a custom categorical palette that matches your company’s brand.

In this article

- About the preferences file below
- Edit the preferences file on the next page
- Create custom color palettes on page 2413
- Use discontinued (classic) color palettes on page 2421

About the preferences file

You can add as many custom palettes as you like to your Preferences.tps file, each with as many colors as you want. When you modify Preferences.tps to add colors, use the standard HTML format for the new colors (hexadecimal value #RRGGBB or Red Green Blue format). When you save the workbook and restart Tableau Desktop, the color palette names you added
to \texttt{Preferences.tps} appear in the \textbf{Select Color Palette} drop-down list (Edit Color dialog). You can use a new palette like you would any other.

Tableau doesn’t test or support custom color palettes, so be sure to back up your workbooks before you continue. Also, there is no guarantee that custom color palettes you create will work with future Tableau Desktop upgrades.

> When you edit your \texttt{Preferences.tps} file, be sure to use straight quotation marks (‘ ’ or “ ”) to delimit the palette name and type, not curly quotation marks (“ ” or ‘ ’).

\section*{Edit the preferences file}

The \texttt{Preferences.tps} file is located in your My Tableau Repository. The file is a basic XML file that you can open in a text editor to modify. An unedited preferences file looks like this:

\begin{verbatim}
<?xml version='1.0'?>
<workbook>
</workbook>
\end{verbatim}

\section*{To edit your preferences file:}

1. Go to the My Tableau Repository folder in your Documents directory, and open the \texttt{Preferences.tps} file.

2. Between the opening and closing \texttt{workbook} tags, insert opening and closing \texttt{preferences} tags.
3. Follow one of the next three procedures to create a custom color palette.

Create custom color palettes

Create a custom categorical color palette

A categorical color palette contains several distinct colors that can be assigned to discrete dimension members. For example, when you put a discrete dimension such as Region on the Color card, the categorical color legend is used.

The following is an example of what to add between the preferences tags to add a categorical color palette. Note that the type attribute is specified as regular, which identifies this palette as a categorical palette.

To create a custom categorical color palette:

1. In the Preferences.tps file, between the "preferences" tags, paste the following. Be sure to use straight quotation marks, not curly quotation marks, to delimit the palette's name and type.
<color-palette name="My Categorical Palette" type="regular" >

<color>#eb912b</color>
<color>#7099a5</color>
<color>#c71f34</color>
<color>#1d437d</color>
<color>#e8762b</color>
<color>#5b6591</color>
<color>#59879b</color>
</color-palette>

2. Save the Preferences.tps file and then restart Tableau Desktop.

3. Open a data source, such as the Superstore - Sample data source.

4. From the Dimensions pane, drag a discrete dimension, such as Region, to Color.

5. Click the color legend menu arrow and select Edit Colors.

6. In the Edit Colors dialog box, from the palette drop-down list, select your new custom palette.
7. Click the **Assign Palette** button to assign the custom colors to each respective field.

8. Click **OK**.
Create a custom sequential color palette

Another type of palette is the sequential color palette. Typically, this type of palette shows a single color, varying in intensity. This type of color palette is used for continuous fields, typically for measures.
The following is an example of what to add between the preferences tags to add a sequential color palette. Note that the type attribute is specified as ordered-sequential, which identifies this palette as a sequential palette. Also, for sequential palettes you must specify at least two variants of the color in the sequential color range.

To create a custom sequential color palette:

1. In the Preferences.tps file, between the preferences tags, paste the following. Be sure to use straight quotation marks, not curly quotation marks, to delimit the palette’s name and type.

   ```xml
   <color-palette name="My Sequential Palette" type="ordered-sequential">
       <color>#eb912b</color>
       <color>#eb9c42</color>
       <color>#ebad67</color>
       <color>#eabb86</color>
       <color>#eacba8</color>
       <color>#ebd8c2</color>
   </color-palette>
   ``

2. Save the Preferences.tps file and then restart Tableau Desktop.

3. Open a data source, such as the Superstore-Sample data source.

4. From the Measures pane, drag a measure (such as Sales) to Color.
5. Click the color legend menu arrow, and select **Edit Colors**.

6. In the Edit Colors dialog box, from the palette drop-down list, select your custom palette.

7. If you want each color gradation to be defined within a box, select the **Stepped Color** check box, and in the **Steps** text box, type the number of color steps you want to display in the bar.

8. Click the **Advanced** button.

9. Select the **Start** check box, and in the text box, type the low end number you want for the continuum.

10. Click the **Apply** button to see the result, and make adjustments as needed. The color will range from high to low intensity (or the reverse) based on the order you specify in the **Preferences.tps** file. The default for sequential color palettes in Tableau is to make the high end of the continuum intense and the low end pale, though selecting the **Reversed** check box will do the opposite.

Create a custom diverging color palette
The third type of color palette is a diverging color palette. A diverging palette shows two ranges of values using color intensity to show the magnitude of the number and the actual color to show which range the number is from. Diverging palettes are most commonly used to show the difference between positive and negative numbers.

![SUM(Profit)]($500,000 $523,589)

The following is an example of what to add between the preferences tags to add a diverging color palette. Note that the type attribute is specified as ordered-diverging, which identifies this palette as a diverging palette.

**To create a custom diverging color palette:**

1. In the **Preferences.tps** file, between the preferences tags, paste the following. Be sure to use straight quotation marks, not curly quotation marks, to delimit the palette’s name and type.

   ```xml
   <color-palette name="My Diverging Palette" type="ordered-diverging">
   <color>#eb912b</color>
   <color>#59879b</color>
   </color-palette>
   ```

2. Save the **Preferences.tps** file and then restart Tableau Desktop.

3. Open a data source, such as the **Superstore - Sample** data source.

4. Click the **Assign Palette** button. The colors in the palette are used in the order they appear in the Preferences file.
If you add a sequential or diverging palette, remember to change the "type" attribute from "regular" to one of the following:

- ordered-sequential
- ordered-diverging

(Optional): Assign a default custom palette to dimensions and measures and publish as a data source

After you save the workbook, the custom color palette information is embedded in the workbook (for Excel and text file-based workbooks, in the .twbx) and therefore only available for that particular workbook. This means that colors that are in use are shown for anybody opening the that particular workbook. If they don't have the modified preferences file, they can't use the color information for any new color encoding.

To allow new color encoding using the custom color palette or to standardize a custom color palette for the Tableau workbooks in your organization, you can create the custom color palette using one of the options above, and then publish it as a Tableau Server data source.
1. On the same computer from which you modified the Preferences.tps file, open Tableau Desktop.

2. Open the Superstore - Sample data source.

3. Right-click a field in the Data pane, and select Default Properties > Color.

4. In the Edit Colors dialog box, associate the field values with the custom color palette, and then click OK when finished.

5. From the Data menu, select the data source, select Publish to Server, and then complete process to publish the data source.

After publishing the data source to Tableau Server, connect any new workbooks to this data source to use the custom color palette.

**Use discontinued (classic) color palettes**

In version 10.0, Tableau created new color palettes, updated some existing ones (such as Tableau 10 and Tableau 20), and discontinued others. If you want to keep using a color palette that was discontinued, you can edit your Preferences.tps file to add the hex values for the palette. You can add as many color palettes as you like. See Hex values for discontinued color palettes below and Restore a discontinued (classic) palette on page 2436 for details.

**Hex values for discontinued color palettes**

The table below lists discontinued color palettes, along with the XML code and hex values you can use in your Preferences.tps file to restore them. If you use the code as-is, palette names will start with Classic to indicate that they're from version 9.3 and earlier. In many cases, there are updated versions of color palettes that have been discontinued. See the "Details" column for specifics.

<table>
<thead>
<tr>
<th>Version 9.x palette name</th>
<th>Details</th>
<th>Version 9.x HEX Values</th>
</tr>
</thead>
</table>
| Tableau 10              | Named Tableau Classic 10 in version 10.0 and higher. | <color-palette name="Classic 10" type = "regular"> <color>#17becf</color> <color>#bcbd22</color> <color>
<table>
<thead>
<tr>
<th>Tableau 10 Medium</th>
<th>Named Tableau Classic Medium in version 10.0 and higher.</th>
<th>Tableau 10 Light Discontinued.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;color-palette name=&quot;Classic 10 Medium&quot; type = &quot;regular&quot;&gt;</td>
<td>&lt;color-palette name=&quot;Classic 10 Light&quot; type = &quot;regular&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;color&gt;#6dccda&lt;/color&gt; &lt;color&gt;#cdcc5d&lt;/color&gt; &lt;color&gt;#a2a2a2&lt;/color&gt; &lt;color&gt;#f7b6d2&lt;/color&gt; &lt;color&gt;#c5b0d5&lt;/color&gt; &lt;color&gt;#ff9896&lt;/color&gt; &lt;color&gt;#98df8a&lt;/color&gt; &lt;color&gt;#ffbb78&lt;/color&gt;</td>
<td>&lt;color&gt;#9edae5&lt;/color&gt; &lt;color&gt;#dbdb8d&lt;/color&gt; &lt;color&gt;#c7c7c7&lt;/color&gt; &lt;color&gt;#f7b6d2&lt;/color&gt; &lt;color&gt;#c49c94&lt;/color&gt; &lt;color&gt;#c5b0d5&lt;/color&gt; &lt;color&gt;#ff9896&lt;/color&gt; &lt;color&gt;#f8df8a&lt;/color&gt; &lt;color&gt;#fb7878&lt;/color&gt;</td>
</tr>
<tr>
<td>Tableau 20 Classic 20</td>
<td>Named Tableau Classic 20 in version 10.0 and higher.</td>
<td>Gray 5</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td><img src="https://example.com" alt="Tableau 20 Classic 20" /></td>
<td>&lt;color-palette name=&quot;Classic 20&quot; type = &quot;regular&quot;&gt; &lt;color&gt;#9edae5&lt;/color&gt; &lt;color&gt;#17becf&lt;/color&gt; &lt;color&gt;#dbdb8d&lt;/color&gt; &lt;color&gt;#c7c7c7&lt;/color&gt; &lt;color&gt;#7f7f7f&lt;/color&gt; &lt;color&gt;#f7b6d2&lt;/color&gt; &lt;color&gt;#c49c94&lt;/color&gt; &lt;color&gt;#8c564b&lt;/color&gt; &lt;color&gt;#c5b0d5&lt;/color&gt; &lt;color&gt;#9467bd&lt;/color&gt; &lt;color&gt;#ff9896&lt;/color&gt; &lt;color&gt;#d62728&lt;/color&gt; &lt;color&gt;#98df8a&lt;/color&gt; &lt;color&gt;#2ca02c&lt;/color&gt; &lt;color&gt;#ffbb78&lt;/color&gt; &lt;color&gt;#ff7f0e&lt;/color&gt; &lt;color&gt;#1f77b4&lt;/color&gt; &lt;/color-palette&gt;</td>
<td></td>
</tr>
<tr>
<td>Discontinued. Check out Seattle Grays in 10.0 and higher for the updated version of this palette.</td>
<td>&lt;color-palette name=&quot;Classic Gray 5&quot; type = &quot;regular&quot;&gt; &lt;color&gt;#cfcfcf&lt;/color&gt; &lt;color&gt;#8f8782&lt;/color&gt; &lt;color&gt;#414451&lt;/color&gt; &lt;color&gt;#a5acaf&lt;/color&gt; &lt;color&gt;#60636a&lt;/color&gt; &lt;/color-palette&gt;</td>
<td></td>
</tr>
<tr>
<td>Palette</td>
<td>Status</td>
<td>HTML Code</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Color Blind 10</td>
<td>Discontinued.</td>
<td>`&lt;color-palette name=&quot;Classic Color Blind&quot; type = &quot;regular&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>Check out <strong>Color Blind</strong> in 10.0 and higher for the updated version of this palette.</td>
<td><code>&lt;color&gt;#cfcfcf&lt;/color&gt; &lt;color&gt;#ffbc79&lt;/color&gt; &lt;color&gt;a2c8ec&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#898989&lt;/color&gt; &lt;color&gt;c85200&lt;/color&gt; &lt;color&gt;5f9ed1&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#595959&lt;/color&gt; &lt;color&gt;ababab&lt;/color&gt; &lt;color&gt;ff800e&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#006ba4&lt;/color&gt;</code></td>
</tr>
<tr>
<td>Traffic Light</td>
<td>Discontinued.</td>
<td>`&lt;color-palette name=&quot;Classic Traffic Light&quot; type = &quot;regular&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>Check out <strong>Traffic Light</strong> in 10.0 and higher for the updated version of this palette.</td>
<td><code>&lt;color&gt;#9fcd99&lt;/color&gt; &lt;color&gt;ffdd71&lt;/color&gt; &lt;color&gt;f26c64&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#69b764&lt;/color&gt; &lt;color&gt;ffc156&lt;/color&gt; &lt;color&gt;d82526&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#309343&lt;/color&gt; &lt;color&gt;dba13a&lt;/color&gt; &lt;color&gt;b10318&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;/color-palette&gt;`</td>
</tr>
<tr>
<td>Purple-Gray 6</td>
<td>Discontinued.</td>
<td>`&lt;color-palette name=&quot;Classic Purple-Gray 6&quot; type = &quot;regular&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#d7d5c5&lt;/color&gt; &lt;color&gt;d098ee&lt;/color&gt; &lt;color&gt;995688&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#94917b&lt;/color&gt; &lt;color&gt;dc5fbd&lt;/color&gt; &lt;color&gt;7b66d2&lt;/color&gt;</code></td>
</tr>
<tr>
<td>Palette Name</td>
<td>Description</td>
<td>Code</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Purple-Gray 12</td>
<td>Discontinued. Check out <a href="#">Purple-Pink-Gray</a> in 10.0 and higher for the updated version of this palette.</td>
<td>&lt;color-palette name=&quot;Classic Purple-Gray 12&quot; type = &quot;regular&quot;&gt;</td>
</tr>
<tr>
<td>Green-Orange 6</td>
<td>Discontinued.</td>
<td>&lt;color-palette name=&quot;Classic Green-Orange 6&quot; type = &quot;regular&quot;&gt;</td>
</tr>
<tr>
<td>Green-Orange 12</td>
<td>Discontinued. Check out <a href="#">Green-Orange-Teal</a> in 10.0 and higher for the updated version of this palette.</td>
<td>&lt;color-palette name=&quot;Classic Green-Orange 12&quot; type = &quot;regular&quot;&gt;</td>
</tr>
<tr>
<td>Palette Name</td>
<td>Discontinued.</td>
<td>Color Palette XML</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Blue-Red 6</td>
<td>Discontinued.</td>
<td>&lt;color-palette name=&quot;Classic Blue-Red 6&quot; type = &quot;regular&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#98d9e4&lt;/color&gt; &lt;color&gt;#3cb7cc&lt;/color&gt; &lt;color&gt;#fff977&lt;/color&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#ff7f0f&lt;/color&gt; &lt;color&gt;#acd98d&lt;/color&gt; &lt;color&gt;#32a251&lt;/color&gt;</td>
</tr>
<tr>
<td>Blue-Red 12</td>
<td>Discontinued. Check out Blue-Red-Brown in 10.0 and higher for the updated version of this palette.</td>
<td>&lt;color-palette name=&quot;Classic Blue-Red 12&quot; type = &quot;regular&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#f4737a&lt;/color&gt; &lt;color&gt;#bd0a36&lt;/color&gt; &lt;color&gt;#ddc9b4&lt;/color&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#ac8763&lt;/color&gt; &lt;color&gt;#b5dffd&lt;/color&gt; &lt;color&gt;#6ba3d6&lt;/color&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#e9c39b&lt;/color&gt; &lt;color&gt;#ac613c&lt;/color&gt; &lt;color&gt;#fff6b0&lt;/color&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#f02720&lt;/color&gt; &lt;color&gt;#b5c8e2&lt;/color&gt; &lt;color&gt;#2c69b0&lt;/color&gt;</td>
</tr>
<tr>
<td>Cyclic</td>
<td>Discontinued.</td>
<td>&lt;color-palette name=&quot;Classic Cyclic&quot;</td>
</tr>
<tr>
<td>Color</td>
<td>Description</td>
<td>Code</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Green</td>
<td>Discontinued. Check out <strong>Green</strong> in 10.0 for the updated version of this palette.</td>
<td><code>&lt;color-palette name=&quot;Classic Green&quot; type = &quot;ordered-sequential&quot;&gt; &lt;color&gt;#09622a&lt;/color&gt; &lt;color&gt;#1a7232&lt;/color&gt; &lt;color&gt;#27823b&lt;/color&gt; &lt;color&gt;#339444&lt;/color&gt; &lt;color&gt;#69a761&lt;/color&gt; &lt;color&gt;#94bb83&lt;/color&gt; &lt;color&gt;#bccfb4&lt;/color&gt; &lt;/color-palette&gt;</code></td>
</tr>
<tr>
<td>Gray</td>
<td>Discontinued. Check out <strong>Gray</strong> in 10.0 for the updated version of this palette.</td>
<td><code>&lt;color-palette name=&quot;Classic Gray&quot; type = &quot;ordered-sequential&quot;&gt; &lt;color&gt;#1e1e1e&lt;/color&gt; &lt;color&gt;#282828&lt;/color&gt; &lt;color&gt;#333333&lt;/color&gt; &lt;color&gt;#3f3f3f&lt;/color&gt; &lt;color&gt;#4b4b4b&lt;/color&gt; &lt;color&gt;#585858&lt;/color&gt; &lt;/color-palette&gt;</code></td>
</tr>
<tr>
<td>Hue Circle</td>
<td>Check out <strong>Hue Circle</strong> in 10.0 and higher for the updated version of this palette.</td>
<td><code>&lt;color type = &quot;regular&quot;&gt; &lt;color textColor #6f63bb&lt;/color&gt; &lt;color textColor #8a60b0&lt;/color&gt; &lt;color textColor #ba43b4&lt;/color&gt; &lt;color textColor #c7519c&lt;/color&gt; &lt;color textColor #d63a3a&lt;/color&gt; &lt;color textColor #ff7f0e&lt;/color&gt; &lt;color textColor #ffaa0e&lt;/color&gt; &lt;color textColor #ffbf50&lt;/color&gt; &lt;color textColor #bcfd22&lt;/color&gt; &lt;color textColor #78a641&lt;/color&gt; &lt;color textColor #2ca030&lt;/color&gt; &lt;color textColor #12a2a8&lt;/color&gt; &lt;color textColor #1f83b4&lt;/color&gt; &lt;/color&gt;</code></td>
</tr>
<tr>
<td>Color</td>
<td>Discontinued.</td>
<td>Check out <strong>Color</strong> in 10.0 and higher for the updated version of this palette.</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Blue</td>
<td>Discontinued.</td>
<td>Check out <strong>Blue</strong> in 10.0 and higher for the updated version of this palette.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color-palette name=&quot;Classic Blue&quot; type = &quot;ordered-sequential&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#26456e&lt;/color&gt; &lt;color&gt;#1c5998&lt;/color&gt; &lt;color&gt;#1c73b1&lt;/color&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#3a87b7&lt;/color&gt; &lt;color&gt;#67add4&lt;/color&gt; &lt;color&gt;#7bc8e2&lt;/color&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#b4d4da&lt;/color&gt; &lt;/color-palette&gt;</td>
</tr>
<tr>
<td>Red</td>
<td>Discontinued.</td>
<td>Check out <strong>Red</strong> in 10.0 and higher for the updated version of this palette.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color-palette name=&quot;Classic Red&quot; type = &quot;ordered-sequential&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#9c0824&lt;/color&gt; &lt;color&gt;#b10c1d&lt;/color&gt; &lt;color&gt;#c21417&lt;/color&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#cf1719&lt;/color&gt; &lt;color&gt;#d8392c&lt;/color&gt; &lt;color&gt;#e35745&lt;/color&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#f57667&lt;/color&gt; &lt;color&gt;#f89a90&lt;/color&gt; &lt;color&gt;#eac0bd&lt;/color&gt;</td>
</tr>
<tr>
<td>Orange</td>
<td>Discontinued.</td>
<td>Check out <strong>Orange</strong> type = &quot;ordered-sequential&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;color&gt;#7b3014&lt;/color&gt; &lt;/color-palette&gt;</td>
</tr>
<tr>
<td>Orange in 10.0 and higher for the updated version of this palette.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>or&gt;#a33202&lt;/color&gt; &lt;color&gt;or&gt;#d74401&lt;/color&gt; &lt;color&gt;or&gt;#f06511&lt;/color&gt; &lt;color&gt;#fd8938&lt;/color&gt; &lt;color&gt;or&gt;#fdab67&lt;/color&gt; &lt;color&gt;or&gt;#f0c294&lt;/color&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;color-palette name=&quot;Classic Area Red&quot; type = &quot;ordered-sequential&quot;&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;color&gt;#bd1100&lt;/color&gt; &lt;color&gt;#c92b14&lt;/color&gt; &lt;color&gt;#d43e25&lt;/color&gt; &lt;color&gt;#e04e35&lt;/color&gt; &lt;color&gt;#ea5e45&lt;/color&gt; &lt;color&gt;#f46b55&lt;/color&gt; &lt;color&gt;#fd7864&lt;/color&gt; &lt;color&gt;#fe8b7a&lt;/color&gt; &lt;color&gt;#fd9c8f&lt;/color&gt; &lt;color&gt;#fbb3ab&lt;/color&gt; &lt;color&gt;#f5cac7&lt;/color&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discontinued.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Area Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discontinued.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Area Green</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discontinued.</td>
</tr>
</tbody>
</table>

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

| Area Green |
| Discontinued. |

<p>| Area Green |
| Discontinued. |</p>
<table>
<thead>
<tr>
<th>Palette Name</th>
<th>Status</th>
<th>HTML Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area Brown</td>
<td>Discontinued</td>
<td><code>&lt;color-palette name=&quot;Classic Area-Brown&quot; type = &quot;ordered-sequential&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#bb5137&lt;/color&gt; </code> <code>&lt;color&gt;#bb6348&lt;/color&gt; </code> <code>&lt;color&gt;#bb7359&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#c08262&lt;/color&gt; </code> <code>&lt;color&gt;#cc8f63&lt;/color&gt; </code> <code>&lt;color&gt;#d89c63&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#e4aa63&lt;/color&gt; </code> <code>&lt;color&gt;#f0b763&lt;/color&gt; </code> <code>&lt;color&gt;#f7c577&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#f6d29c&lt;/color&gt; </code> <code>&lt;color&gt;#f3e0c2&lt;/color&gt;</code></td>
</tr>
<tr>
<td>Red-Green Diverging</td>
<td>Discontinued</td>
<td><code>&lt;color-palette name=&quot;Classic Red-Green&quot; type = &quot;ordered-diverging&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#09622a&lt;/color&gt; </code> <code>&lt;color&gt;#1e7735&lt;/color&gt; </code> <code>&lt;color&gt;#2f8e41&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#69a761&lt;/color&gt; </code> <code>&lt;color&gt;#a2c18f&lt;/color&gt; </code> <code>&lt;color&gt;#cacaca&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#fc8375&lt;/color&gt; </code> <code>&lt;color&gt;#df513f&lt;/color&gt; </code> <code>&lt;color&gt;#d11719&lt;/color&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>&lt;color&gt;#bd1316&lt;/color&gt; </code> <code>&lt;color&gt;#9c0824&lt;/color&gt;</code></td>
</tr>
<tr>
<td>Red-Blue Diverging</td>
<td>Discontinued</td>
<td><code>&lt;color-palette name=&quot;Classic Red-Blue&quot; type = &quot;ordered-diverging&quot;&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Palette</td>
<td>Details</td>
<td>Code</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Blue Diverging</td>
<td>in 10.0 and higher for the updated version of this palette.</td>
<td><code>&lt;color&gt;#26456e&lt;/color&gt; &lt;color&gt;#1c5998&lt;/color&gt; &lt;color&gt;#1c73b1&lt;/color&gt; &lt;color&gt;#3a87b7&lt;/color&gt; &lt;color&gt;#67add4&lt;/color&gt; &lt;color&gt;#cacaca&lt;/color&gt; &lt;color&gt;#fc8375&lt;/color&gt; &lt;color&gt;#df513f&lt;/color&gt; &lt;color&gt;#d11719&lt;/color&gt; &lt;color&gt;#bd1316&lt;/color&gt; &lt;color&gt;#9c0824&lt;/color&gt;</code></td>
</tr>
<tr>
<td>Red-Black Diverging</td>
<td>Discontinued. Check out Red-Black Diverging in 10.0 and higher for the updated version of this palette.</td>
<td><code>&lt;color-palette name=&quot;Classic Red-Black&quot; type = &quot;ordered-diverging&quot;&gt; &lt;color&gt;#1e1e1e&lt;/color&gt; &lt;color&gt;#383838&lt;/color&gt; &lt;color&gt;#565656&lt;/color&gt; &lt;color&gt;#777777&lt;/color&gt; &lt;color&gt;#9b9b9b&lt;/color&gt; &lt;color&gt;#cacaca&lt;/color&gt; &lt;color&gt;#fc8375&lt;/color&gt; &lt;color&gt;#df513f&lt;/color&gt; &lt;color&gt;#d11719&lt;/color&gt; &lt;color&gt;#bd1316&lt;/color&gt; &lt;color&gt;#9c0824&lt;/color&gt; &lt;/color-palette&gt;</code></td>
</tr>
<tr>
<td>Area Red-Green Diverging</td>
<td>Discontinued.</td>
<td><code>&lt;color-palette name=&quot;Classic Area Red-Green&quot; type = &quot;ordered-diverging&quot;&gt; &lt;color&gt;#4a8c1c&lt;/color&gt; &lt;color&gt;#559633&lt;/color&gt; &lt;color&gt;#5ea049&lt;/color&gt; &lt;color&gt;#69aa56&lt;/color&gt; &lt;color&gt;#75b65d&lt;/color&gt; &lt;/color-palette&gt;</code></td>
</tr>
<tr>
<td>Orange-Blue Diverging</td>
<td>Discontinued. Check out Orange-Blue Diverging in 10.0 and higher for the updated version of this palette.</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;color-palette name=&quot;Classic Orange-Blue&quot; type = &quot;ordered-diverging&quot;&gt; or #26456e or #1c5998 or #1c73b1 or #3a87b7 or #67add4 or #7bc8e2 or #cacaca or #fdab67 or #fd8938 or #f06511 or #d74401 or #a33202 or #7b3014&lt;/color-palette&gt;</td>
<td></td>
</tr>
<tr>
<td>Palette</td>
<td>Description</td>
<td>Code</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Green-Blue Diverging</td>
<td>Discontinued. Check out <strong>Green-Blue Diverging</strong> in 10.0 and higher for the updated version of this palette.</td>
<td><code>&lt;color-palette name=&quot;Classic Green-Blue&quot; type = &quot;ordered-diverging&quot;&gt; &lt;color&gt;#26456e&lt;/color&gt; &lt;color&gt;#1c5998&lt;/color&gt; &lt;color&gt;#1c73b1&lt;/color&gt; &lt;color&gt;#3a87b7&lt;/color&gt; &lt;color&gt;#1c73b1&lt;/color&gt; &lt;color&gt;#67add4&lt;/color&gt; &lt;color&gt;#cacaca&lt;/color&gt; &lt;color&gt;#a2c18f&lt;/color&gt; &lt;color&gt;#69a761&lt;/color&gt; &lt;color&gt;#2f8e41&lt;/color&gt; &lt;color&gt;#1e7735&lt;/color&gt; &lt;color&gt;#09622a&lt;/color&gt; &lt;/color-palette&gt;</code></td>
</tr>
<tr>
<td>Red-White-Green Diverging</td>
<td>Discontinued. Check out <strong>Red-White-Green Diverging</strong> in 10.0 and higher for the updated version of this palette.</td>
<td><code>&lt;color-palette name=&quot;Classic Red-White-Green&quot; type = &quot;ordered-diverging&quot;&gt; &lt;color&gt;#09622a&lt;/color&gt; &lt;color&gt;#297839&lt;/color&gt; &lt;color&gt;#428f49&lt;/color&gt; &lt;color&gt;#74af72&lt;/color&gt; &lt;color&gt;#b9d7b7&lt;/color&gt; &lt;color&gt;ffffff&lt;/color&gt; &lt;color&gt;#fcb4a5&lt;/color&gt; &lt;color&gt;#e86753&lt;/color&gt; &lt;color&gt;#cc312b&lt;/color&gt; &lt;color&gt;#b41f27&lt;/color&gt; &lt;color&gt;#9c0824&lt;/color&gt; &lt;/color-palette&gt;</code></td>
</tr>
<tr>
<td>Red-Black-White Diverging</td>
<td>Discontinued. Check out <strong>Red-Black-White Diverging</strong> in 10.0 and higher for the updated version of this palette.</td>
<td><code>&lt;color-palette name=&quot;Classic Red-White-Black&quot; type = &quot;ordered-diverging&quot;&gt; &lt;color&gt;#1e1e1e&lt;/color&gt; &lt;color&gt;#393939&lt;/color&gt; &lt;color&gt;#575757&lt;/color&gt;</code></td>
</tr>
<tr>
<td>Palette</td>
<td>Updated Version</td>
<td>Discontinued</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td><strong>Orange-White-Blue Diverging</strong></td>
<td>for the updated version of this palette.</td>
<td>Discontinued. Check out <strong>Orange-Blue-White Diverging</strong> in 10.0 and higher for the updated version of this palette.</td>
</tr>
<tr>
<td><strong>Red-White-Black Light</strong></td>
<td>Discontinued.</td>
<td></td>
</tr>
</tbody>
</table>

```
<color-palette name="Classic Orange-White-Blue" type = "ordered-diverging">
  <color>#26456e</color> or #2e5f8a or #3679a8 or #6a9ec5 or #b7cde2 or #ffffff
  <color-palette name="Classic Red-White-Black Light" type = "ordered-diverging">
  <color>#c6c6c6</color> or #d1d1d1 or #dddddd
  ```
<table>
<thead>
<tr>
<th>Palette Name</th>
<th>Status</th>
<th>Color Palette</th>
</tr>
</thead>
</table>
| Orange-White-Blue Light Diverging    | Discontinued | `<color-palette name="Classic Orange-White-Blue Light" type = "ordered-diverging"> ...
| Red-White-Green Light Diverging      | Discontinued | `<color-palette name="Classic Red-White-Green Light" type = "ordered-diverging"> ...

Red-Green Light Diverging  Discontinued.

```
<color-palette name="Classic Red-Green Light" type = "ordered-diverging">
  <color>#b7e6a7</color> <color>#c1e6b4</color> <color>#cae6c0</color>  
  <color>#d4e6cc</color> <color>#dde6d9</color> <color>#e5e5e5</color>  
  <color>#ecdbdc</color> <color>#f2d1d2</color> <color>#f8c7c9</color>  
  <color>#fcbdc0</color> <color>#ffb2b6</color>
</color-palette>
```

Restore a discontinued (classic) palette

1. Go to the My Tableau Repository folder in your Documents directory, and open the Preferences.tps file.

2. Between the opening and closing workbook tags, insert opening and closing preferences tags.

```
<workbook>
  <preferences>
    </preferences>
</workbook>
```

3. In the Preferences.tps file, between the "preferences" tags, paste the XML code for the
discontinued palette (see **Hex values for discontinued color palettes** on page 2421). For example:

```html
color-palette name="Classic Blue Red 12" type = "regular">
    <color>#f4737a</color>  <color>#bd0a36</color>
    <color>#ddc9b4</color>  <color>#ac8763</color>
    <color>#b5dffd</color>  <color>#6ba3d6</color>
    <color>#e9c39b</color>  <color>#ac613c</color>
    <color>#ffb6b0</color>  <color>#f02720</color>
    <color>#b5c8e2</color>  <color>#2c69b0</color>
</color-palette>
```

Use straight quotation marks (as in ' ' or " "), not curly quotation marks, to delimit the palette's name and type.

4. Save the **Preferences.tps** file and then restart Tableau Desktop.

   When you open the Edit Colors dialog box and choose **Select Color Palette**, the color palette you added will be at the bottom of the palette list.
Optimize Workbook Performance

When we talk about performance, we mean the speed with which you can work in Tableau. That might mean speed of data analysis, for example, if you work in Tableau Desktop on a large corporate database that takes forever in real-time when you access it remotely. Or it might mean simply the speed of loading views or dashboards that you access on your desktop or from Tableau Server.

If you are working with small data volumes, many of the recommendations in this section are optional for you. Your workbook performance is probably as speedy as you expect. But if you work with hundreds of millions of records, designing an efficient workbook is critically important to the speed at which you can work. We recommend that you review the topics and tips in this section before you start building your first view.

Tableau provides a number of features designed to help you optimize performance. These features are discussed throughout the help within the descriptions of tasks and workflows where you would most likely encounter them. They are also collected here in case you want make performance your focus.

Making performance improvements almost always means making trade-offs. For example: Do you sacrifice speed to make sure you have the most current data? Is it important to include all the data for analysis, or will a speedier subset do?

The bigger and more complicated the data, the longer it takes to interpret and render, but there are things you can do to accelerate the process. Think of fine-tuning your workbook performance as working on a puzzle with multiple pieces. This section provides you with the pieces of information, and you pick and choose which pieces solve your particular puzzle.

This section starts by discussing the big picture and then drills down to specific functionality. We start with databases, data, and extracts, and then proceed through things you can do that affect your data source, your workbooks, your calculations, and visualizations. Next, we talk about how to record and analyze workbook performance. Finally we look at how to reduce upload times to a Tableau server.

Know Your Data at the Database Level

Understanding the impact that certain factors in the design of your database have on Tableau performance might give you information you can use to work with your database team to
optimize data at the database level.

Enable support for referential integrity

Databases that support referential integrity support the Tableau Assume Referential Integrity feature, which improves the performance of inner joins. Joins cost time and resources to process on the database server. When you join multiple tables in a data source, Tableau uses functionality that is generally invisible to the user, called join culling. Join culling queries only the relevant tables instead of all tables defined in your join. Join culling only occurs where referential integrity is defined between tables. For more information, see Assuming Referential Integrity on page 680.

Make sure database permissions support creating temp tables

Does your database grant users permission to create and drop temporary tables, also known as temp tables, and does the environment have sufficient spool space for the queries being run? Tableau creates temp tables to help improve performance and add functionality. The temp tables temporarily hold information specific to a connection. If the creation of temp tables in the database is restricted, the overall performance of workbooks and views isn’t as fast as it could be.

Create indexes for tables

Index the tables in your relational database. To successfully index your data set, identify the fields that you frequently filter on and add them to the index. If you have a field that you use as a context filter often, consider setting it as your primary index. If you are working with Access tables that have more than 200,000 rows of data, consider setting indexes on the tables. You can learn how to do this by searching for “index” in the Access online help. You can officially store 2 GB of data (approximately 1-2 million rows) in an Access database, but it performs poorly well below this limit.

Many database management system (DBMS) environments have management tools that will look at a query and recommend indexes that would help.

Break up your data

You can often improve performance if you partition a large database table into multiple smaller tables. For example, you can create a cluster of Access tables that addresses specific subsets
of your data.

**Use a database server**

If you have a lot of data, you might consider storing it in a database server, such as Oracle, MySQL, or Microsoft SQL Server. The Professional Edition of Tableau can connect to these larger database servers.

**Test Your Data and Use Extracts**

You probably have no control over the hardware, processing power, and disk space that your database runs on. Or how many people access the database at any given time, how many services are running, or how robust the network is. But you can test the *raw performance* of the data before you begin to build a workbook, so that you understand the environment you’re working in. And there are things you can do to affect the performance of the workbooks you create, starting with how you choose to connect to your data: the drivers you use, and whether your connection type is *live* or *extract*.

**Test as close to the data as possible**

A good way to test the raw performance of the data is to install Tableau Desktop on the computer where the data resides and to run some queries. This eliminates external factors such as network bandwidth and latency from performance results, and help you to understand the raw performance of the query in the data source.

Additionally, you can use the *localhost* name for the data instead of the DNS name to help determine if environmental factors such as slow name resolution or proxy servers are slowing performance.

**Connect with native database drivers**

When you connect to data with native database drivers, you will often experience significantly faster performance than when you connect to the same data with ODBC drivers. When you use native database drivers, make sure that you’re using the latest version. Database vendors are always working to improve their drivers. For a list of drivers, see *Drivers and Activation* on the Tableau website.
Work with extracts instead of live data

Depending on your data, you can choose between a live or extract connection on the data source page. A live connection is a direct connection to your data. A Tableau data extract is a compressed snapshot of data stored locally and loaded into memory as required to render a Tableau visualization. Extracts are designed to use all parts of your computer’s memory optimally.

There are several reasons to use an extract, but the main performance-related reason is if your query execution is slow. The extract data format is designed to provide a fast response to analytic queries. In this case, you can think of the extract as a query acceleration cache.

For more information about Tableau extracts, see Extract Your Data on page 773. Another reference is a three-part blog post about extracts that starts with the first post, Understanding Tableau Data Extracts.

Reduce the amount of data

When you create an extract, use filters to exclude data that you don’t need. Also, ask yourself if you need all of the records in a data source, or if you can limit the extract to a representative sample. For more information, see Extract Your Data on page 773.

Hide unused fields

Hidden fields are not included when you create an extract. Use the Hide All Unused Fields option to hide unnecessary fields before you create an extract. This makes the extract smaller, which improves performance. For more information, see Hide or Unhide Fields on page 971.

Optimize extracts

Since an extract is a columnar store, the wider the data set, the slower the query time. You can optimize an extract by selecting a data source on the Data menu and then selecting Extract > Optimize. Optimizing an extract creates secondary structure in the extract that speeds up future queries. For more information, see Materialize Calculations in Your Extracts on page 803.
Use extracts for file-based data

In general it’s best practice to import file-based data—text files such as CSV, Microsoft Excel spreadsheets, and Microsoft Access files, as well as statistical files like SPSS, SAS, and R—into Tableau. This makes queries perform much faster and also results in a much smaller file to store the data values.

If your data is a large text or Excel file, using an extract not only improves performance but also makes more functionality available to you. Note that if you connect Tableau to a large text file, you will be prompted to extract the data if Tableau discovers that the file is too large to perform well.

Avoid using custom SQL

In most cases, custom SQL runs slower than queries created by Tableau. Tableau cannot perform query optimizations on custom SQL. But in cases where you must use custom SQL, use an extract so that the query runs only once. For more information, see Connect to a Custom SQL Query on page 723.

Create Efficient Joins and Blends

Now that you’ve made the initial connection to your data, consider how you set up your data source—especially joins and blends—to make it efficient.

If your analysis requires data from different databases, including different workbooks or files, you should consider the impact of joining or blending your data. Joins and blends can improve performance, but they also require more processing power, and so take more time to execute.

A key factor in efficiency is reducing the amount of data you analyze. We’ll outline a few ways you can do this.

Should I join or blend my data?

When you consider joining tables or blending tables, think about where the data is coming from, the number of connections, and the number of records you have. If the workbook uses data from more than one database, you must either blend data or add another connection to the existing data source and create a cross-database join. If the workbook uses multiple tables from the same database, joining the tables can improve performance and filtering control.

For more information, see Blend Your Data on page 682.
Use blending and joins carefully

You should restrict joins to the fewest number of tables possible. In cases where you need access to many tables in a workbook, you may want to create separate data connections with joins that are tailored to a particular worksheet.

Blending queries the data from both data sources at the level of the linking fields, and then merges the results of both queries together in memory. For this reason, best practice is to avoid data blending on dimensions with many unique values (for example, Order ID, Customer ID, or exact date/time).

Reduce query time by aggregating, then blending data

If you have multiple connections to large amounts of data that take a long time to query, consider aggregating the tables, and then blending the data on the aggregate. For more information, see Blend on Summary Data on page 704. For information about aggregating data in an extract, see Materialize Calculations in Your Extracts on page 803.

Create a primary group

If you must blend two data sources because one contains the “fact” records and the other contains dimensional attributes, you may be able to improve performance by creating a primary group or alias. Primary groups are used for one-to-many relationships and primary aliases are used for one-to-one relationships. Note that both primary groups and aliases are not dynamic and must be updated if the data changes. Consequently, they aren’t a great solution for frequently updated data, but if you need a quick mapping, they can potentially eliminate the need for costly blends. For an example, see Bring a Field into the Primary Data Source on page 692.

Design for Performance While You Build a View

You’re connected to your data, and you’ve built your data source. The next step is to begin building a view in Tableau. There are a few things to keep in mind as you build your workbook so that it will perform faster while you’re building it, as well as after it is created.
Use the Describe field to get to know your data

Rather than dragging a dimension out onto the rows shelf so that you can eyeball the data—which requires Tableau to render the data as a crosstab, one of the slowest visualization types—use the **Describe field** option. Right-click the field in the **Data** pane to view a description of the data type as reported by the data source, as well as a sample of the dimension member values.

Keep workbooks a reasonable size

The fewer worksheets and data sources in a workbook, the faster it will perform. If you have a big topic to explore and are tempted to put everything into a single monolithic workbook, reconsider. Notice if your workbook starts to slow down with the addition of a new view and additional queries. If the performance starts to suffer as you enlarge the scope of your workbook, think about breaking your workbook into separate files.

Turn off Automatic Updates

When you place a field on a shelf, Tableau generates the view by automatically querying the data. If you are creating a dense data view, the queries might be time-consuming and significantly degrade system performance. In this case, you can turn off queries in Tableau while you build the view. You can then turn queries back on when you are ready to see the result. For more information, see **Turn off Automatic Updates to Boost Performance** on page 2457.

Look for warnings

Tableau displays a performance warning dialog box when you attempt to place a large dimension (with many members) on any shelf. The dialog box provides four choices, as shown below. If you choose to add all members, then you might experience a significant degradation in performance.
You might also see a warning when you attempt to create too many panes in a table. In this case, Tableau warns you that the requested table "contains more than the recommended maximum number of panes." It is best not to display more than the recommended number of panes, in part because you won’t get a useful view.

**Filter Your Data Carefully**

Filters are an effective way to restrict the amount of data for analysis. The less data to analyze, the faster the query executes. In Tableau, there are a variety of ways to filter your data. You can create a filter on a data source, you can implement a filter when you create an extract, you can filter dimensions, measures, and dates. You can create a context filter that acts as an independent filter—then any other filters you set are dependent filters because they process only the data that passes through the context filter. You can also show a filter in a view that users can interact with. For information about when filters are executed, see Tableau’s Order of Operations on page 267.

Filters provide enormous flexibility, but you should use them carefully. They can be computationally expensive, so reducing the number of filters can improve workbook performance.

**Filter your data in the data source**

When you create a filter on a data source, you reduce the amount of data in the data source. For systems that rely heavily on partitions or indexing, data source filters may yield tremendous control over the performance of queries issued by Tableau. For more information, see Filter Data from Data Sources on page 766.
Use Keep Only filters instead of Exclude filters

Because Exclude filters load all of the data for a dimension, they run more slowly than Keep Only filters. For more information, see Select to keep or exclude data points in your view on page 1163.

Add filters first

If you are working with a large amount of data and have automatic updates turned off, you can inadvertently create a very slow query when you add filters to the view. Rather than building the view first, and then specifying filters, specify the filters first and then drag fields to the view. That way, when you run the update or turn on automatic updates, the filters are evaluated first.

Use context filters sparingly

Think of a context filter as an independent filter; any other filters you set process only the data that passes through the context filter. Only use context filters when they limit the size of the data set significantly. If you are setting filters that significantly reduce the data set size, and that will be used for more than several data views, you should set those filters as context filters. For more information, see Improve View Performance with Context Filters on page 1194. For more information about performance improvement with context filters, see Speed up Context Filters on page 1195.

Use a set to filter data

If you want to filter a dimension to remove members based on a range of measure values, you should create a set rather than using a quantitative filter. For instance, you can create a set that only returns the Top 50 items in a dimension, rather than all of the items in a dimension.

When you create a group from a selection as described in Group Your Data on page 990, make sure you include only the columns of interest. Each additional column in the set decreases performance.

Aggregate your data using custom SQL

As an alternative to filters, another effective way to restrict the amount of data for analysis is to use custom SQL to aggregate measures before you bring the data into Tableau. Suppose your reviews data is stored in a SQL Server table, and for your analysis, you’re only interested in the
data from the “rating” and “reviewer ID” fields. To pre-aggregate the data so that you bring into Tableau only the fields you’re interested in, you might use the following custom SQL:

```
SELECT AVG([rental_reviews].[Rating]) AS [Rating], [rental_reviews].[id] AS [id]
FROM [dbo].[rental_reviews] [rental_reviews]
GROUP BY [id]
```

This custom SQL query reduces the amount of data brought into Tableau because it pre-aggregates the reviews and averages the values in the review ratings.

**Filter your cube data**

Here are some specific tips that apply to filtering cube data.

**Filter on a specific dimension**

If your cube has a single large dimension, you should set a filter directly on that dimension rather than setting a filter on another dimension or measure. For example, suppose you want to reduce the numbers of products being displayed in a view. It is much more efficient to set the filter directly on Products or to create a computed set based on Products (such as Top 10) rather to filter other fields such as Location or Profit.

Also, avoid selecting large numbers of members from a large dimension. When a dimension is large, keep the size of the filter to less than a thousand members.

**Create a set**

When creating a set from a selection as described in [Create Sets on page 1004](#), make sure you’ve included only the columns of interest in the Create Set dialog box. Each additional column in the group decreases performance. For example, if you create a set that contains all regions with sales between 8,000 and 15,000, but you include a column that doesn't affect the members of the set, you might notice a performance decrease. To remove extra columns, right-click the column and select **Remove This Column** from the context menu.
Don’t apply sorts to levels

Avoid applying sorts to levels within a very large hierarchy in a cube.

Use the Exclude command

If you are working with a dimension whose root level is greater than 1000 but less than 100,000, avoid using the Filter dialog box to filter the data. Instead, drag the dimension to a shelf and use the Exclude command in the headers context menus to limit the data that is displayed in the view.

Create Efficient Calculations

When your data doesn’t provide all the information you need to answer your questions, you can create calculated fields to help with your analysis.

Within a calculated field you can define a hardcoded constant (such as a tax rate), do very simple mathematical operations like subtraction or multiplication (for example, revenues minus cost), use more complex mathematical formulas, perform logical tests (IF/THEN, CASE), do type conversions, send expressions to external services such as R, and much more.

There are different calculation types in Tableau:

**Basic and aggregate calculations:** These types of calculations are generated as part of the query to the underlying data source and are calculated in the database. In general, basic and aggregate calculations scale very well, and there are many database tuning techniques that can improve their performance.

**Table calculations:** These calculations are calculated by Tableau on the query result set. While this means more work for Tableau, table calculations are generally done over a much smaller set of records than are in the original data source. If table calculation performance is a problem (possibly because the result set returned to Tableau is very large) consider pushing some aspects of the calculation back to the data source layer. One way to do this is to aggregate the data and then perform the calculation on the aggregated data.

**Level of detail (LOD) expressions:** LOD expressions are generated as part of the query to the underlying data source and are calculated in the database. They are expressed as a nested select, so they are dependent on database performance. A table calculation or blending might perform better than a LOD expression, or vice versa.
If you suspect performance is slow due to a LOD expression, you might try replacing it with a table calculation or a data blend to see if performance improves. For an example, see Example 2 in Tableau's Order of Operations on page 267.

LOD expressions can be affected by join culling, so look at Assuming Referential Integrity on page 680 if your queries run slowly when you use LOD expressions.

For more information, see the Tableau whitepaper Understanding Level of Detail (LOD) Expressions.

**Booleans and integers are faster**

When you create calculated fields, the data type you use has a significant impact on the calculation speed. Integers and Booleans are generally much faster than strings. If your calculation produces a binary result (for example, yes/no, pass/fail, over/under), be sure to return a Boolean result rather than a string.

**Use parameters for conditional calculations**

A common technique in Tableau is to show a parameter control so users can select a value that determines how a calculation is performed. Typically, to give the user easy-to-understand options, it makes sense to create the parameter as a string type. But numerical calculations are much faster than string calculations, so take advantage of the Display As feature of parameters: that is, show text labels but use underlying integer values for the calculation logic. For more information, see Create Parameters on page 1031.

**Convert date fields**

Users often have date data that is not stored in native date formats—for example, a date might be a string or a numeric timestamp. You can use the DATEPARSE function if your data supports it—this function is available for non-legacy Microsoft Excel and text file connections, MySQL, Oracle, PostgreSQL, and Tableau data extract data sources. Otherwise, parse the field into a date string such as “2012-01-01”. ISO strings are preferred because they are not locale-specific. Then pass the value to the DATE function. If the originating data is a numeric field, converting it first to a string and then to a date is very inefficient. It is much better to keep the data as numeric and use DATEADD and date literal values to perform the calculation. The performance gains can be significant with large data sets. See Date Functions on page 1298 for more information about these functions.
Use ELSEIF logic statements

When working with complex logic statements, remember that ELSEIF is faster than ELSE IF, because a nested IF computes a second IF statement rather than being computed as part of the first.

Aggregate measures

If the views you create are slow, make sure you are working with aggregated measures. With disaggregated data, you might be trying to view many rows of data at once. You can reduce the number of rows by aggregating the data. To do this, select Analysis > Aggregate Measures.

Calculation tips

There are lots of little things you can do that can improve calculation performance.

- Distinct counting values is one of the slowest aggregation types in almost all data sources. Use the COUNTD aggregation sparingly.
- Using parameters with a wide scope of impact (for example, in a custom SQL statement) can affect cache performance.
- Filtering on complex calculations can potentially cause indexes to be missed in the underlying data.
- Script functions like RAWSQL and SCRIPT_* for integrating with external services can be slow, particularly if there are lots of values that need to be passed back and forth from the DBMS/R server.
- Use NOW only if you need the time stamp level of detail. Use TODAY for date level calculations.
- Remember that all basic calculations are passed through to the underlying data—even literal calculations like label strings. If you need to create labels (for example, for column headers) and your data is very large, create a simple text/Excel file data source with just one record to hold them so they don’t add overhead on the big data source.

Make Visualizations Faster

You’ve put in a lot of work to make your visualization—view, dashboard, or story—make a point or tell a story. You don’t want to lose your audience while you wait for the visualization to load.
By following the guidelines in this topic, you can improve the speed of your visualizations.

Reduce the scope

Whether you're creating a view, dashboard, or story, it's tempting to pack a lot of information into your visualization because it's so easy to add more fields and calculations to the view and more sheets to the workbook. But the result can be that the visualization becomes slower and slower to render.

Remember that each worksheet runs one or more queries against the data, so the more sheets, the longer it takes to render the visualization.

Take advantage of the fact that Tableau is designed to deliver interactive visualizations to your users, and spread the data out across multiple visualizations, if you can. Be strategic when designing your visualization—the fewer sheets and data sources, the faster your visualization will perform.

Limit the number of filters you show in the view

Filters that you show in your view (formerly called quick filters) are a very powerful feature of Tableau that you can use to create rich, interactive visualizations for your users. See the filters highlighted on the right in the following screenshot:
When you add an interactive filter to a view, each filter in the view requires a query in order to populate the options. If you add a lot of interactive filters to your dashboard, it can cause the dashboard to take a long time to render.

Also, when you use “show relevant values” on a filter, it requires a query to update the shown values each time other filters are changed. Use this feature sparingly.

Reduce the number of marks on the view

While there is no hard and fast rule on what defines “too many marks,” be aware that more marks means that more processing power and memory is required to render them. You can find the number of marks by looking at the lower left of the Tableau Desktop window in the status bar. Watch out for large crosstabs and for maps with complex custom polygons. Keep in mind that too many data points on a view can also reduce the visual analytics value by causing information overload.
To avoid this problem, compile related views and connect them with action filters so that you can go from an overview to a more granular view as you explore the data. Make sure that you remove any unneeded dimensions from the Detail shelf. You can also try displaying your data in different types of views to see what’s most effective.

**Zoom without filtering**

When users zoom in on a visualization with a large number of marks, Tableau doesn’t filter out the marks you can’t see. What changes is the view of the data, not the total number of marks. If you only need a subset of the data, filter out the data you don’t need.

**Record and Analyze Workbook Performance**

Your workbook is done and you’re wondering if its performance is taking a little longer than it should. To find out what’s slowing it down, you can use a performance recording to evaluate your workbook. This is an especially good idea if you plan to share or publish the workbook.

The Performance Recording feature in Tableau records performance information about key events as you interact with a workbook. You can then view performance metrics in a workbook that Tableau creates to analyze and troubleshoot different events that are known to affect performance:

- Query execution
- Geocoding
- Connections to data sources
- Layout computations
- Extract generation
- Blending data
- Server blending (Tableau Server only)

Tableau support may ask that you create a performance workbook as they work with you to diagnose performance issues.

**Create a performance recording in Tableau Desktop**

To start recording performance, follow this step:
Help > Settings and Performance > Start Performance Recording
To stop recording, and then view a temporary workbook containing results from the recording session, follow this step:

Help > Settings and Performance > Stop Performance Recording
You can now view the performance workbook and begin your analysis.
If you are sending the recording to Tableau Support, save this workbook as a packaged workbook (.twbx) file, and then send it.

Interpret a performance recording workbook
A performance recording workbook is a Tableau dashboard that contains three views: Timeline, Events, and Query.

Timeline
The uppermost view in a performance recording dashboard shows the events that occurred during recording, arranged chronologically from left to right. The bottom axis shows elapsed time since Tableau started, in seconds.
In the Timeline view, the Workbook, Dashboard, and Worksheet columns identify the context for events. The Event column identifies the nature of the event, and the final column shows each event’s duration and how it compares chronologically to other recorded events:

Events
The middle view in a performance recording workbook shows the events, sorted by duration (greatest to least). Events with longer durations can help you identify where to look first if you want to speed up your workbook.
Different colors indicate different types of events. The range of events that can be recorded is:

- Computing layouts
  If layouts are taking too long, consider simplifying your workbook.
- Connecting to data source
  Slow connections could be due to network issues or issues with the database server.
- Executing query
  For live connections, if queries are taking too long, it could be because the underlying data structure isn’t optimized for Tableau. Consult your database server’s documentation. As an alternative, consider using an extract to speed performance.
  For extracts, if queries are taking too long, review your use of filters. If you have a lot of filters, would a context filter make more sense? If you have a dashboard that uses filters, consider using action filters, which can help with performance.
- Generating extract
  To speed up extract generation, consider only importing some data from the original data source. For example, you can filter on specific data fields, or create a sample based on a specified number of rows or percentage of the data.
- Geocoding
  To speed up geocoding performance, try using less data or filtering out data.
- Blending data
  To speed up data blending, try using less data or filtering out data.
- Server rendering
  You can speed up server rendering by running additional VizQL Server processes on additional machines.
Query

If you click on an **Executing Query** event in either the **Timeline** or **Events** section of a performance recording dashboard, the text for that query is displayed in the Query section.

If you are connected to a published data source, the query text is displayed in XML. If you are connected to the data source directly, the query is displayed in SQL like shown below:

```sql
SELECT "State"."ID" AS "ID",
       "StateSynonyms"."Name" AS "State_Name",
       "State"."ParentID" AS "State_ParentID"
FROM "StateSynonyms"
INNER JOIN "State" ON ("State"."ID" = "StateSynonyms"."ParentID") AND ("State"."MapCode" = "StateSynonyms"."MapCode"

If it makes sense, you can use the query text to work with your database team on optimizing at the database level. Sometimes the query is truncated and you’ll need to look in the Tableau log to find the full query. Most database servers can give you advice about how to optimize a query by adding indexes or other techniques. See your database server documentation for details.

Sometimes for efficiency, Tableau combines multiple queries into a single query against the data. In this case, you may see an **Executing Query** event for the Null worksheet and zero queries being executed for your named worksheets.

Reduce Upload Times to Tableau Server

You’ve created your workbook and analyzed its performance recording to make sure that it’s been optimized for speed. Now you’re ready to publish the workbook to Tableau Server. It isn’t always possible to reduce upload times to Tableau Server, but here are some considerations to keep in mind.

**Avoid publishing packaged workbooks**

It is a best practice to publish workbooks (.twb files) and data sources separately, rather than publishing them together as packaged workbook (.twbx files). This may not make your initial uploads of workbooks and data sources any faster than publishing them together, but it makes republishing workbooks faster (if you don’t have to also republish data sources), and it makes data sources available online for new workbooks. For data sources that are extracts, it also allows you to schedule automated refreshes.
Make extracts smaller

For Tableau extracts, you can help speed uploads by making your extract smaller. As you create extracts, consider:

- Removing unused fields from extracts.
- Making extracts smaller with sampling and filtering.
- Aggregating data for visible dimensions.

See also

Extract Your Data on page 773
Publish a Data Source on page 2516

Turn off Automatic Updates to Boost Performance

When you place a field on a shelf, Tableau generates the resulting view by querying the data source. When you create a dense data view that involves many fields, these queries can be time-consuming. You can instruct Tableau to turn off automatic updates to improve performance.

Automatic updates for worksheets

By default, automatic updates are turned on and the toolbar button is highlighted . However, it is sometimes more efficient for Tableau to execute the queries you need only for your final view, rather than for every intermediate step required to compose that view. You can turn off updates for worksheets by clicking the Pause Auto Updates toolbar button.

You can also turn automatic updates on and off by pressing F10 (Option-Command-0 on a Mac) on your keyboard.

You can update the view at any time while automatic updates are off by pressing F9 (Shift-Command-0 on a Mac) or clicking the Run Update button on the toolbar.
Note: It is possible to enter an invalid state when automatic updates are turned off. When this happens, the view is desaturated and invalid commands are disabled. The view and commands become available again when you click Run Update on the toolbar.

For example, the view below has automatic updates turned off. When the aggregation for Profit is changed from a summation to an average, the view is desaturated to let you know that you have made a change to the view that has made the current view invalid.

Automatic updates for filters

When you change a filter, even when you turn off automatic updates for worksheets, Tableau continues to query the data source to update the view.

When you work with a large data set, this continuous query action can degrade performance, so Tableau also gives you an option to pause automatic updates for filters.

If you pause automatic updates for filters, you can make as many changes as you need without having to wait for Tableau to update the filter each time.

To pause automatic updates for filters, click the drop-down arrow on the Pause Auto Updates button on the toolbar and select Auto Update Filters to clear the check mark.
To resume automatic updates for filters, click the drop-down arrow on the **Pause Auto Updates** button on the toolbar and select **Auto Update Filters** again. You can also click the **Run Update** button on the toolbar to manually update the view at any time.

**How Automatic Updates Affect Dashboards and Stories**

You configure automatic updates on a per-view basis. This means that you can have a dashboard where some views update automatically, and others don't. Similarly, you can have a story where some story points update automatically, and others don't. But when dashboards or stories are published to Tableau Server, automatic updates affect either none of the contributing views in a story or dashboard, or all of the views.
Save Your Work

You can save your work at anytime while analyzing or interacting with data in Tableau.

In this article

For Tableau Desktop

- **Automatically save a workbook** on the next page
- **Save a workbook** on the next page
- **Save a packaged workbook** on page 2462
- **Save a bookmark** on page 2462

For web authoring

- **Save a workbook** on page 2463
- **Save a copy of an existing workbook** on page 2464
- **Save changes as a custom view** on page 2465

For Tableau Desktop

In Tableau Desktop, there are several ways for you to save your work:

- **Automatically save a workbook** - Automatically saves the workbook in the same location as the original file. In the event of a crash, a recovered version is available.
- **Save a workbook** – Saves all open worksheets.
- **Save a packaged workbook** – Saves the workbook along with all referenced local file data sources and images into a single file.
- **Save a bookmark** – Saves the current worksheet.

You can share workbooks and bookmarks with your co-workers, provided they can access the relevant data sources that the workbook uses. If your co-workers do not have access to the data sources, you can save a packaged workbook.

Custom fields such as binned measures, calculated fields, groups, and sets are saved with workbooks and bookmarks.
Automatically save a workbook

Tableau Desktop automatically saves your work for you every few minutes - no more losing hours of work if Tableau Desktop closes unexpectedly. This feature is enabled by default, but you can turn it off from the toolbar under Help > Settings and Performance > Enable Autosave.

**Note:** If you don't see this option, your System Administrator may have disabled this feature.

If Tableau crashes, a recovered version of the workbook is automatically created with a .twbr extension and saved in the same location as the original file or in your My Tableau Repository/Workbooks folder. New workbooks are saved with the name "Book1" plus a numeric ID. When you reopen Tableau, a recovery dialog box shows a list of the recovered files that you can select and open to continue in your flow.

You can also delete unwanted files from this same dialog box.

For more information about how to turn this feature on or off during installation, see Plan your Tableau Desktop Deployment in the Tableau Desktop Deployment guide.

Save a workbook

When you open Tableau Desktop, it automatically creates a new workbook. Workbooks hold the work you create and consist of one or more worksheets. Each worksheet contains a particular view of your data.

**To save a Tableau workbook:**
1. Select **File > Save**.

2. Specify the workbook file name in the **Save As** dialog box.

By default, Tableau saves the file with the .twb extension. By default, Tableau saves your workbook in the **Workbooks** folder in your My Tableau Repository. You can find this repository in your Documents folder. However, you can save Tableau workbooks to any directory you choose.

Tableau file names cannot include any of the following characters: forward slash (/), backslash (\), greater-than sign (>), less-than sign (<), asterisk (*), question mark (?), quotation mark ("), pipe symbol (|), colon (:), or semicolon (;).

**To save a copy of a workbook you have open:**

- Select **File > Save As** and save the file with a new name.

**Save a packaged workbook**

Packaged workbooks contain the workbook along with a copy of any local file data sources and background images. The workbook is no longer linked to the original data sources and images. These workbooks are saved with a .twbx file extension. Other users can open the packaged workbook using Tableau Desktop or Tableau Reader, and do not need access to the data sources that the workbook includes.

To learn more about how to save your workbook as a packaged workbook, see **Packaged Workbooks** on page 2465.

**Save a bookmark**

You can save a single worksheet as a Tableau bookmark. When you save the bookmark, Tableau creates a snapshot of the worksheet. Bookmarks can be accessed from any workbook using the Bookmarks menu. When you open a bookmarked worksheet, it adds the worksheet to your workbook in the state that it was in when it was bookmarked. It will never update or change automatically. Bookmarks are convenient when you have worksheets that you use frequently.

**To save a Tableau bookmark:**

1. Select **Window > Bookmark > Create Bookmark**.

2. Specify the bookmark file name and location in the Create Bookmark dialog box.
Tableau saves the file with a .tbm extension. The default location is the Bookmarks folder in the Tableau Repository. However, you can save bookmarks to any location you choose. Bookmarks that are not stored in the Tableau repository do not appear on the **Bookmark** menu.

You can organize bookmarks into folders in the same way you organize files or documents. This can be useful when you have a large number of bookmarks to manage. For example, you might organize bookmarks based on employee name, product types, or sales results. You can organize bookmarks by creating a new folder, renaming an existing folder, renaming existing bookmark files, and so on.

Delete bookmarks the same way you would delete any other file on your computer. After you delete a bookmark from the Bookmarks folder in the Tableau Repository, it is removed from the **Bookmarks** menu the next time you start Tableau.

**Note:** While bookmarks are generally a snapshot of the worksheet and include the data connection, formatting, etc., a bookmark does not include parameter values and the current page setting on the Pages shelf.

### For web authoring

When creating, editing, and interacting with views on Tableau Server or Tableau Online, there are a couple of different ways for you to save your work:

- **Save a workbook** - saves a workbook in the project you specify.
- **Save a copy of a workbook** - saves a copy of the workbook in the project you specify.
- **Save changes as a custom view** - saves changes as a custom view, which is related to the original view and updates when the original view is updated.

### Save a workbook

When you create a new workbook, or edit an existing workbook on Tableau Server or Tableau Online, you can save your work at any time.

**To save a workbook:**
In web editing mode, select **File > Save**.

**Note:** If the workbook has never been saved, you must select **File > Save As**.

### Save a copy of an existing workbook

Sometimes you don’t want to overwrite an existing view with your changes. In cases like these, you can save a copy of an existing workbook. When you do this, the existing workbook remains unchanged, and a copy of it is created for you to edit as you wish.

**Note:** When you save a copy of an existing workbook, it is not updated when the original workbook is updated or republished. It is also not deleted when the original view is deleted. If you would like to create a view that updates with the original, see the **Save changes as a custom view** on the next page section.

### To save a copy of a workbook:

1. In web editing mode, select **File > Save As**.
2. In the Save Workbook dialog box that opens, do the following:
   1. For **Name**: Enter a name for the workbook.
   2. For **Project**: Select the project in which you would like to save the workbook.
   3. (Optional) Select **Show sheets as tabs** to display all worksheets, dashboards, and stories in the workbook as separate tabs.
4. Click **Save**.

**Note:** The person who published the data source to Tableau Server or Tableau Online may have also set up credentials for accessing the published data source included in the workbook you are about to save. These authentication options should appear in the Save Workbook dialog box. For more information, see the authentication types in the **Set Credentials for Accessing Your Published Data** topic.
Save changes as a custom view

If you notice you are making the same changes to a view every time you open it, you might want to consider saving the changes as a custom view. This option is not available in web editing mode, but can be accessed when you open a view to interact with it.

A custom view does not change the original, but is related to it. If the original view is updated or republished, the custom view is also updated.

You can also choose whether your custom views are visible to other users (public), or only to you (private).

For more information about custom views and how to create them, see Use Custom Views on page 2649.

See Also

Use Custom Views on page 2649

Packaged Workbooks

Workbooks often reference external resources. For example, workbooks might reference background images or local file data sources such as Excel files, Access files, and Tableau extract files (.hyper or .tde).

When you save a workbook, links to these resources are also saved. The next time you open the workbook, the views are automatically updated with any changes that may have occurred to the data and images. In most cases, you will want to save the workbook in this way. But if you plan to share the workbook with someone who does not have access to the referenced resources or to Tableau Server, you might want to save a packaged workbook instead.

Packaged workbooks contain the workbook along with a copy of any local file data sources and background images. The workbook is no longer linked to the original data sources and images. These workbooks are saved with a .twbx file extension. Other users can open the packaged workbook using Tableau Desktop or Tableau Reader.
Create a .twbx with file-based data sources

1. Select File > Save As.
2. Specify a file name for the packaged workbook in the Save As dialog box.
3. Select Tableau Packaged Workbooks on the Save as type drop-down list.
4. Click Save.

The default location is the Workbooks folder of the Tableau repository. However, you can save packaged workbooks to any directory you choose.

The following files are included in packaged workbooks:

- Background images
- Custom geocoding
- Custom shapes
- Local cube files
- Microsoft Access files
- Microsoft Excel files
- Tableau extract files (.hyper or .tde)
- Text files (.csv, .txt, etc.)
If you are sharing packaged workbooks that contain Microsoft Excel or Access 2007 data sources, the people opening the workbook must either have Microsoft Excel and Access 2007 or the Office 2007 Data Connectivity Components installed on their computers. The data connectivity components are available on the Tableau Tableau Drivers page.

Create a .twbx with non-file-based data sources

If the workbook contains connections to enterprise data sources or other non-file-based data sources, such as Microsoft SQL, Oracle, or MySQL, the data must be extracted from the data sources for it to be included in a packaged workbook (.twbx).

1. In the workbook, right-click the data source in the Data pane and choose Extract Data.

2. In the Extract Data dialog box, click the Extract button to extract all data from the data source.
After the extract completes, the data source icon changes to indicate that an extract is active for that data source. Instead of a single cylinder, there are two cylinders connected by an arrow.
3. **Optional:** Repeat the above steps for each data source in the workbook.

4. Select **File > Save As.**

5. From the **Save as type** drop-down menu, select **Tableau Packaged Workbook (*.twbx).**

After the extracts have been created for all non-file-based data sources and the packaged workbook has been saved, you can send your workbook.

**Create a .twbx with Tableau Server data sources**

If the workbook contains connections to a published Tableau Server data source, you must download a local copy of the Tableau Server data source, take an extract of it, and then replace the connection to the local copy for it to be included in a packaged workbook (.twbx).
1. In the workbook, right-click the published data source in the Data pane, and then select **Create Local Copy**.

A copy of the published data source is added to the Data pane.

2. Right-click the local copy, and select **Extract Data**.

3. In the Extract Data dialog box, click the **Extract** button to extract all data from the data source. Creating an extract of the data source allows the person you are sharing the workbook with to have access to a copy of the data source.
4. In the Data pane, right-click the published data source, and then select **Replace Data Source**.
5. Verify that the published data source will be replaced by the local data source, and then click **OK**.

6. Right-click the published data source, and then click **Close**.

7. Select **File > Save As**.

8. From the **Save as type** drop-down menu, select Tableau Packaged Workbook (*.twbx).
After the local copy and extract of the local copy is created and the packaged workbook saved, you can send your workbook.

Unpackage a .twbx

Packaged workbooks can be unpackaged.

- On a Windows computer, right-click the packaged workbook file (.twbx) in Windows Explorer and select Unpackage.
- On a Mac computer, rename the file with a .zip extension (for example, from myfile.twbx to myfile.zip) and then double-click it.

When you unpack a workbook, you get a regular workbook file (.twb), along with a folder that contains the data sources and images that were packaged with the workbook.

Save Workbooks to Tableau Public

If you want to share your data discoveries with the world outside of your organization, you can save your workbook to Tableau Public, a free cloud service. On Tableau Public, anyone can interact with your views, or download your workbooks or data sources. For information, go to the Tableau Public website.

Save a workbook

1. With your workbook open in Tableau Desktop, select Server > Tableau Public > Save to Tableau Public.

Note: This option is available only if you’ve created a viz that contains at least one field.
2. Sign in using your Tableau Public account.

![Sign in to Tableau Public](image)

If you don’t have an account, select the link to create a new one.

3. Type a name for the workbook and click **Save**.

When you save a workbook to Tableau Public, the publishing process creates an extract of the data connection.

**Tip:** The title becomes part of your view’s metadata. Use a unique title that will help others find it when they search. (The title shown in the image is a good example of how *not* to name your workbook.)

![Save Workbook to Tableau Public](image)

After the workbook is published, you are redirected to your account on the Tableau Public website.

On your profile page on Tableau Public, do any of the following to customize your profile:
• Hover the pointer over a viz to get access to actions such as selecting it as your featured viz, or hiding, downloading, or deleting it.

• Hover the pointer over a viz and then select View to open the viz’s home page. There you can select Edit Details to customize metadata such as workbook name and description, add a permalink, and change other settings.

• To get a link to share on social media or code to embed in a web page, display a view, and then click Share at the bottom of the view. (You can get links and embed code for other Tableau Public users’ views this way, too.)

Revert a Workbook to the Last Saved Version

You can revert a workbook at anytime to undo all of the changes you’ve made to it since you last saved it.

To revert to the last saved version of a workbook:

• In Tableau Desktop, select File > Revert to Saved, and then select Revert in the warning dialog box that opens.

• In web authoring mode, select File > Revert.

The Revert command is only available for workbooks (.twb) that do not have connections to Extract data sources.

Export Views

If you want to transfer your Tableau results into a presentation, report or web page, Tableau provides the following options.

Copy the view to another application

You can copy your view as an image and paste it into another application such as Microsoft PowerPoint, Word, or Excel. If you’re using Tableau Desktop on a Mac, a TIFF (Tagged Image File Format) is created for pasting purposes. On Windows, a BMP (Bitmap) is created. Either image format will include everything in your view, including Tableau fonts.
If you’re running Windows, you can create an EMF (Enhanced Metafile) image instead of a BMP, but if your view includes a Tableau font (such as Tableau Regular, Tableau Semibold, etc.), a font that’s similar to the Tableau font will be substituted. See Create an EMF of the view (Windows only) below for steps.

To create a vector-based file of your view that embeds the Tableau fonts, you can use File > Print to PDF. See Print the view to PDF on the next page for details.

Create an image of the view

1. Select Worksheet > Copy > Image.
2. In the Copy Image dialog box, select what you want to include in the image, along with the legend layout (if the view contains a legend).
3. Click Copy. Tableau copies the current data view as a BMP (Windows) or a TIFF (Mac) and places it on the Clipboard.
4. Open the target application and paste the image from the Clipboard.

Create an EMF of the view (Windows only)

You can create an EMF of your view instead of a BMP by using the Export command. If your view uses a Tableau font, it will be substituted with a similar (non-Tableau) font.

1. Select Worksheet > Export > Image.
2. In the Export Image dialog box, select what you want to include in the image, along with the legend layout.
3. Click Save and in the Save Image dialog, save the image as Enhanced Metafile (*.emf).

Export the view as an image file

The export image command saves the current view as an image file. You can export to an image file with the following three steps.

1. Select Worksheet > Export > Image.
2. In the Copy Image dialog box, select the contents you want to include in the image and the legend layout (if the view contains a legend).
3. Click **Save**.

4. In the Save Image dialog box, navigate to where you want to save the image file and type a file name into the text box. Select a file format from the Save as type drop-down menu.

5. Click **Save**.

**Print the view to PDF**

You can publish one or more views to PDF by selecting **File > Print to PDF**.

**Note:** When printing a sheet, filters in the view are not included. To show filters, create a dashboard containing the sheet and print the dashboard to PDF. However, when printing a dashboard to PDF, the contents of web page objects are not included.

**Export Data**

One way to export your work from Tableau to another application is to export the data. You can export the data in the Tableau data source, which can contain a portion or all of the records from your original data. Alternatively, you can also export the portion of data used to generate the view.

**Export data in the data source**

After you join tables from one or more connections and make general customizations (for example, create a calculated field, pivot fields, create groups, apply data source or extract filters, etc.) to your Tableau data source, you might want to share or reuse the data in its new form. You can do this by using one of the methods listed below.

- Export your data to .csv file on the next page
- Extract your data on the next page
- Export the data source on the next page

**Note:** The export may exclude some table calculations and level of detail expressions.
Export your data to .csv file

Because the .csv format is one of the most simple structured formats for data, it's supported by a wide range of tools, databases, and programming languages. Exporting your data in the Tableau data source using this format creates an independent data set and can be a convenient and flexible way to share your data with others.

There are two primary ways you can export your data in the data source to a .csv file in Tableau: from the Data Source page and from the view.

- **From the Data Source page**: On the Data Source page, select Data > Export Data to CSV to export all the data in your data source to .csv file.
- **From the view**: On the sheet tab, drag a field to the Columns or Rows shelf, click the View Data icon in the Data pane, and click the Export All button.

Extract your data

Another way to export all of your data or a subset of your data in the data source is to create an extract (.hyper) file. An extract functions as an independent data set, which you can use to connect directly from Tableau. For more information, see Extract Your Data on page 773.

Export the data source

After you connect to your data, you can export and save your data source as a Tableau data source (.tds) file. Saving the data source creates a shortcut to your remote data and allows you to avoid having to create a new connection to a specific data set each time. For more information, see Save Data Sources on page 832.
Export data used in the view

After you create a view, you can also export just the data used to generate that view.

The fields that are exported come from the fields on the shelves of the sheet. However, fields that function as external filters, in other words, the fields that appear only on the Filters shelf, are not included in the export. If you want to include other fields with the exported data without changing the baseline view, you can place those fields on the Detail shelf.

The various methods for exporting the data used to generate the view is listed below.

- Export data in the view to Microsoft Access or .csv below
- Export crosstab of data in the view to Excel on the next page
- Copy data in the view to clipboard on page 2481
- Copy crosstab of data in the view to clipboard on page 2482

Export data in the view to Microsoft Access or .csv

Export the data that is used to generate the view as an Access database (Windows only) or .csv file (Mac only).

1. In Tableau Desktop, select Worksheet > Export > Data.
2. Select a location and type a name for your Access database or .csv file.
3. Click Save.

   If you're on Windows, the Export Data to Access dialog box displays to give you the option to immediately use the new Access database and continue working in Access.
without interrupting your work flow.

Export crosstab of data in the view to Excel

You can export directly to Excel the data used to generate the view formatted as a crosstab. When you export your view as a crosstab, Tableau automatically opens the Excel application and pastes a crosstab version of the current view into a new Excel workbook.

Although this option provides a direct method for exporting your data to another application, performance of the export can be affected because it is simultaneously copying and formatting the data. If the view you are exporting contains a lot of data, a dialog box opens asking whether you want to export the formatting. In this case, if you choose to exclude the formatting from the export, performance of the export might improve.
1. Create a view.

   ![View Diagram]

2. Select Worksheet > Export > Crosstab to Excel.

   **Copy data in the view to clipboard**

   Copy the data used to generate the view so that you can paste it into another application.

   1. Create a view.

   ![View Diagram]

   2. Select Worksheet > Copy > Data.

   3. Open another application, such as Word, and paste the data into the document.
In this example, the fields placed on the Columns, Rows, and Color shelves are copied into the document. However, the **Customer Segment** field is not copied because it is an external filter because it appears only on the Filters shelf.

<table>
<thead>
<tr>
<th>Category</th>
<th>Customer Name</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Furniture</td>
<td>Aaron Bergman</td>
<td>$391</td>
</tr>
<tr>
<td>Furniture</td>
<td>Adam Shillingsburg</td>
<td>$2,077</td>
</tr>
<tr>
<td>Furniture</td>
<td>Adrian Barton</td>
<td>$1,280</td>
</tr>
<tr>
<td>Furniture</td>
<td>Aimee Bixby</td>
<td>$16</td>
</tr>
<tr>
<td>Office Supplies</td>
<td>Aaron Bergman</td>
<td>$274</td>
</tr>
<tr>
<td>Office Supplies</td>
<td>Adam Shillingsburg</td>
<td>$1,058</td>
</tr>
</tbody>
</table>

**Copy crosstab of data in the view to clipboard**

You can copy a crosstab version of a view so that you can paste or transfer the data into another application. The pasted data always appears as a crosstab, even if the initial view of the data in Tableau did not use a crosstab format.

Copying a crosstab is restricted by some general conditions:

- You must copy all records in the view. You cannot copy a subset of records.
- This option is valid for aggregated views only. It cannot be used on disaggregated views of data because a crosstab is by definition an aggregated view of data. This means the **Aggregate Measures** option on the Analysis menu must be selected in order for copying a crosstab to work properly.
- You cannot copy a crosstab if the view contains continuous dimensions such as continuous dates and times.
- Other restrictions may apply depending on the data in your view.

After the general conditions are met, copy the crosstab.
1. Create a view.

2. Select Worksheet > Copy > Crosstab.

3. Open another application, such as Excel, and paste the crosstab.

Copying Information Between Workbooks

You can combine resources from different workbooks into one, or pull information from a larger workbook into a standalone subset.

To share or extract a subset of information, you can use any of the following actions on worksheets, dashboards, and stories:

- Copy and paste selected sheets to another workbook.
- Import a saved workbook into your current workbook.
Export selected sheets to a new workbook.

In this article

- What gets copied or saved with selected sheets below
- How Tableau handles duplicate items below
- Import an entire Tableau workbook on page 2486
- Export and import sheets between workbooks below
- Copy and paste sheets between workbooks on page 2487

What gets copied or saved with selected sheets

When you copy, save, or export selected sheets, other workbook items that the views in those sheets depend on are also included:

- Any associated visible or hidden sheets.
- Data sources for fields used in the selected sheets.
- Calculations, parameters, groups, sets, actions, custom shapes, and so on.

How Tableau handles duplicate items

When you paste or import sheets from a different workbook, some items might already exist in the destination workbook, or some items might have the same name in both places. If Tableau encounters an exact duplicate item in the Data pane, such as a calculation, it does not paste or import that item into the destination workbook. However, if an item in the Data pane has the same name but is defined differently, Tableau imports and renames it.

Tableau also pastes or imports sheets and data sources with duplicate names—whether only the name is identical, or their names and contents are identical—and it renames the newer copy.

Export and import sheets between workbooks

If you want to extract a subset of information from a larger workbook to maintain as a standalone file, you can export or save selected sheets to a new workbook. You can then
import that workbook into an existing one to incorporate its sheets and other objects into the existing workbook.

**Note:** These steps describe how to share your work among Tableau workbooks. You can also export views to use outside of Tableau. For information, see Export Views on page 2475.

1. Open the workbook that contains the sheets you want to export to a new file.

2. Using the tabbed worksheet, filmstrip view, or sheet sorter view, right-click (Control-click on Mac) the sheet tab or thumbnail view, and then select Export to export a single sheet. Use Shift + click or Ctrl + click to select multiple sheets.

3. In the Save As dialog box, specify the file format you want to save (.twb or .twbx), select the location for the new workbook file, give it a name, and then click Save.
Import an entire Tableau workbook

After you save or export selected sheets to a new workbook (.twb) file, you can import the information into another workbook.

1. With the existing workbook open, select File > Import Workbook.

2. Select the workbook that contains the sheets you saved from another workbook, and click Open.
The following image shows the result of importing a workbook that contains a sheet with the same name as a sheet in the existing workbook. Tableau adds a number after the name of the imported sheet.

Copy and paste sheets between workbooks

Copying and pasting sheets is a quick way to combine information from different workbooks or create a new workbook. You can copy one or more sheets from the tabbed sheet view, filmstrip view or sheet sorter view. Use Shift + click or Ctrl + click to select multiple sheets.

To copy and paste a worksheet using the filmstrip view, do the following:

1. Open a workbook and click the **Filmstrip** button in the status bar.

2. Select the thumbnails of the sheets you want to copy, then right-click (Control-click on Mac) and select **Copy**.
Tableau copies the information in the file format (.twb or .twbx) of the workbook.

3. Open the destination workbook, or create a new workbook. Right-click (Control-click on Mac) on the tab for any sheet, and select **Paste**.

   Pasted sheets are placed after existing worksheets, dashboards, and stories.

   **Note:** The **Paste** option is not available when the active sheet is a story.

4. Save the changes.

For information about what gets copied when you copy and paste sheets, see the **What gets copied or saved with selected sheets** on page 2484 section.

### Print Tableau Views

You can print Tableau views. The first thing you should do before printing is specify how you want the printed page to look using the Page Setup dialog box. Then you can print to a printer or publish to a PDF.

In this article

- Set up the page on the next page
- Print on page 2491
- Print to PDF on page 2492
Set up the page

Before you print, there are several options you can set to specify how the worksheet will look when it is printed.

For example, you can select which elements to include, set the printed page orientation, specify where you want to put the legend, margins, and more. You define these settings in the Page Setup dialog box. You can set different page setup options for each worksheet in the workbook. That way you can have different titles, captions, legend settings, etc., for each worksheet you want to print.

To begin setting up your page, select **File > Page Setup**.

The Page Setup dialog box has the following tabs:

- **General** - Use the General tab to select the elements you want to show when you print. You can show or hide the title, view, caption, color legend, shape legend, size legend, and map legend.

  Specify how to handle headers and breaks. The headers refer to the headers in each of your views. When you select **Repeat headers and legends on each page**, the row and column headers will appear at the top of each printed page when a view breaks across several pages.

  Select **Break pages on pane boundaries** to prevent page breaks in the middle of a cell in a table.

  If you have used the Pages Shelf to build your view, you can select whether to print the current page only or all pages.

- **Layout** - Use the Layout tab in the Page Setup dialog box to specify the layout legend, page margins, and centering options for printing.

  - **Legend Layout** - If you include one or more legends, you can select one of the options for how you want the legends to appear on the printed page.

  - **Margins** - Specify top, bottom, left, and right margins by typing values into the text boxes.

  - **Centering** - Optionally, select whether to center the view horizontally or vertically—or both—on the page.

- **Print Scaling** - Use the Print Scaling tab to fit the view to a certain size or to change the page orientation. These options only affect printed documents. The scaling options you
specify here will not affect exported images or PDFs. However, the orientation settings will be used as the default when you publish the workbook to Tableau Server or Tableau Public.

- **Print Scaling**

You can scale your view to fit within a single page or scale it across multiple pages. Select from the following options:

- Automatic – Scales the view automatically based on the paper size.
- Scale to – Scales the view to the specified percentage of its original size.
- Fit to – Scales the view to fit within the specified area. Select the number of printed pages across and down. For example, if you have a really wide view that is not very tall, you can specify three pages across by one page down.

- **Page Orientation**

Use the page orientation settings to specify how you want the view oriented on the printed page. For example, if you have a view that is very wide but not very high, select **Landscape**. Select from the following page orientation options:

- Use Printer Setting – Use the page orientation that is already specified by the printer.
- Portrait – Presents the view so that it is oriented vertically on the printed page.
- Landscape – Presents the view so that it is oriented horizontally on the printed page.

The following diagram shows the difference between portrait and landscape page orientations.
These page orientation settings are used as the default settings when you publish the workbook to Tableau Server or Tableau Public.

Print

After you have configured the Set up the page on page 2489 settings, you can print by selecting File > Print. In the Print dialog, select a printer, decide whether to show selections, specify a print range, and select the number of copies you want to print.

The following options in the Print dialog box are unique to Tableau.

Show Selections

When this option is selected any selections you’ve made in the views will be maintained while printing.

Change the Print Range

When you print from a workbook with multiple worksheets, each worksheet represents one or more printed pages, depending on the page setup.

Select from the following print ranges:

- **Entire Workbook** - Prints all the worksheets in the workbook.
- **Active Sheet** - Prints only the sheet currently displayed in the workbook.
- **Selected Sheets** - Prints the selected sheets.
You can select multiple worksheets in a workbook by holding down the CTRL or Shift keys (or the % key on a Mac) while clicking the worksheet tabs that you want to select.

Print to PDF

In Tableau, you can publish views as PDF files rather than printing them as hard copies. You do not need to have Adobe Acrobat installed on your computer.

Publish as a PDF using a Windows computer

1. Specify page setup options for each sheet in your workbook.
2. Select File > Print to PDF.

![Print to PDF dialog box]

3. In the Print to PDF dialog box, select the print Range:
   - Entire Workbook - Publishes all the sheets in the workbook.
   - Active Sheet - Publishes only the sheet currently displayed in the workbook.
   - Selected Sheets - Publishes the selected sheets. To select multiple sheets in a Tableau workbook, hold down the Ctrl key as you select sheet tabs along the bottom of the Tableau workbook.
4. Select a Paper Size. If you select Unspecified, the paper size will expand to the necessary size to fit the entire view on a single page.
5. Select **View PDF File After Printing** if you want to automatically open the PDF after creating it. This option is only available if you have Adobe Acrobat Reader or Adobe Acrobat installed on your computer.

6. Select whether to **Show Selections**. When this option is selected the selections in the views are maintained in the PDF.

7. Click **OK** and specify where you want to save the PDF. Then click **Save**.

### Publish as a PDF using a Mac computer

1. Specify **page setup** options for each sheet in your workbook.

2. Select **File > Print**.

3. In the Print dialog box, click **Show Details** to select a print range:
   - **Entire Workbook** - Publishes all the sheets in the workbook.
   - **Active Sheet** - Publishes only the sheet currently displayed in the workbook.
   - **Selected Sheets** - Publishes the selected sheets. To select multiple sheets in a Tableau workbook, hold down the Command key as you select sheet tabs along the bottom of the Tableau workbook.

4. Click **PDF > Save as PDF**.
5. Specify where you want to save the PDF, then click Save.
Publish Data Sources and Workbooks

Suppose you create a view that exposes a new range of questions in the data you’re using, and you want to share this analysis with other people using this data. Or maybe you are your team’s Data Steward, in charge of building the data models approved for use by analysts, and meeting your organization’s requirements for security, compliance, performance, and so on.

You can share your work with the rest of your team by publishing it to Tableau Server or Tableau Online. After it’s published, you and your team can access it through your web browser or the Tableau mobile app. Publishing data sources can also help you to centralize data management.

In this article

- Why publish on the next page
- What you can publish on the next page
- Who can publish on page 2497

In other resources

For the steps on how to publish, see either of the following topics:

- Publish a Data Source on page 2516
- Comprehensive Steps for Publishing a Workbook on page 2521

Note: If you do not have Tableau Online or Server, you can share analysis on Tableau Public, a free cloud service. As the name suggests, views published to Tableau Public are publicly accessible. For more information, see public.tableau.com.
Why publish

You can publish data sources and workbooks when you want to widen the audience for your data analysis within your organization. By publishing you can begin to do the following:

- **Collaborate and share with others**
  Allow people in your organization to view, interact with, download, subscribe to, share, edit, and save published views, even if they do not use Tableau Desktop. Incorporate views into blog posts or websites.

- **Centralize data and database driver management**
  Create and publish data models that everyone can use. Centralized data management allows for sharing a single source for your Tableau data. All workbooks connected to the published data reflect updates to it.

  In addition, when you publish and connect to data on the server, people connecting to the data from Tableau Desktop do not need to install and maintain database drivers on their own computers.

- **Support mobility**
  Access your data from a different computer or location, through a web browser or the Tableau Mobile iOS app. Sign in to your organization’s Tableau Server from a private network offsite.

What you can publish

Content types you can publish include:

- **Data sources**: You can publish data sources that others can use to build new workbooks. A data source can contain a direct (or live) connection to your database or an extract you can refresh on a schedule.

  For information, see [Best Practices for Published Data Sources](#) on page 2499.

- **Workbooks**: Workbooks contain your views, dashboards, and stories, and data connection. You can include local resources, such as background images and custom geocoding, if they reside in a location that the server or other Tableau users cannot access.
Who can publish

To publish to Tableau Server or Tableau Online, your server or site administrator must grant you the following capabilities:

- A site role of **Creator** (formerly Publisher) on the site you're publishing to.
- **View** and **Save** capabilities set to **Allowed** on the project into which you publish.

If you use Tableau Desktop and are not sure whether you can publish to a server, or you are having trouble publishing, see your Tableau administrator. If you’re an administrator, see **Content Access and Ownership** in the Tableau Server help (or the **Tableau Online version**) for more information about site roles and permissions.

Simple Steps to Share a Workbook

When you want to share a workbook in an open-collaboration environment, you can get your work to your Tableau Server or Tableau Online site with a few simple clicks. There the people you want to share it with can view it, interact with it, and even edit it if their server permissions allow.

Before you share (publish) your workbook on the server, make sure you know the following things:

- If your organization uses Tableau Server, get the name of the server and how you sign in to it.

  If your organization uses Tableau Online, you can select the Quick Connect link, although you will need only your sign-in process.

  If your organization does not use Tableau Server or Tableau Online, you can create a Tableau Online trial site as part of the sharing steps.

- Any existing publishing or sharing guidelines your Tableau administrator might have, such as the name of the project you should use for sharing.
Share your workbook

1. With the workbook open in Tableau Desktop, click the Share button in the toolbar.

If you aren’t already signed in to Tableau Server or Tableau Online, do so now.
If you don’t have a site yet, here’s where you can create one on Tableau Online.

2. In the Publish Workbook dialog box complete the following recommended minimum settings:
   a. Select the project to publish to.
   b. Name the workbook according to whether you’re creating a new one or publishing over an existing one.
      Make sure the name will be understandable to your audience. Unlike most of these settings, you can’t change the name on the server after you publish. Instead you republish with the new name and remove the old workbook. By that time your viewers might have created bookmark links to the old one, which would no longer be valid.
   c. Under Data Sources, select Edit, and for Authentication, select Allow refresh access or Embed password.
      Only one of these choices appears in the list, depending on the data connection. If None shows, leave it set to that.
   d. Click Publish.
You can change some settings on the server after you publish. To learn about those we didn’t touch here, see the following articles:

- **Prepare for Publishing a Workbook** on page 2505
- **Comprehensive Steps for Publishing a Workbook** on page 2521
- **Set Credentials for Accessing Your Published Data** on page 2530

How others access the workbook on the server

After you publish, you can let your colleagues know it’s ready for them.

1. Make sure your colleagues or managers can access the site and project you published to.

2. Use the steps in **Share Web Views** on page 2664 to send them a direct link to your workbook.

3. After they sign in, they navigate to the project and workbook, where they can select it to open it for viewing, interacting, or editing.

Using the simple steps for sharing, you make the data access easy for others by embedding the sign-in credentials. These are separate from Tableau Server or Online credentials, and they still need to sign in to the server.

For more information, see **Set Credentials for Accessing Your Published Data** on page 2530.

To learn about creating sharable data sources and connecting your workbooks to them, see **Publish a Data Source** on page 2516.

Best Practices for Published Data Sources

Publishing data sources to Tableau Online or Tableau Server is integral to maintaining a single source for your data. Publishing also enables sharing data among colleagues; including those who don’t use Tableau Desktop, but have permission to edit workbooks in the web editing environment.

Updates to a published data source flow to all connected workbooks, whether the workbooks themselves are published or not.
In this article

- What makes up a published data source below
- Preparing a data source for publishing on the next page
- When to use an extract on the next page
- Publishing data separately or embedded in workbooks on page 2502
- Keeping extracts up-to-date on page 2504
- Additional resources on page 2504

What makes up a published data source

A Tableau data source consists of the following:

**The data connection information** that describes what data you want to bring in to Tableau for analysis. When you connect to the data in Tableau Desktop, you can create joins, including joins between tables from different data types. You can rename fields on the Data Source page to be more descriptive for the people who work with your published data source.

**An extract**, if you decide to create one. Guidelines for when to create an extract are included below, as well as in the additional resources.

**Information about how to access or refresh the data.** The connection also includes access information. Examples of this type of information include:

- The path to an original Excel file.
- Embedded credentials or OAuth access tokens for accessing the data directly.
- Alternatively, no credentials, so that users are prompted to enter them when they want to access the data (whether it’s to view a workbook that connects to it, or to connect a new workbook to it).

For more information, see **Set Credentials for Accessing Your Published Data** on page 2530.

**Customization and cleanup** that helps you and others use the data source efficiently. When you’re working with your view, you can add calculations, sets, groups, bins, and parameters; define any custom field formatting; hide unused fields; and so on.

All of these refinements become part of the metadata contained in the data source that you publish and maintain.
Preparing a data source for publishing

When you publish a data source, consider these best practices:

- Create the connection for the information you want to bring into Tableau and do any customization and cleanup that will help you and others use the data source efficiently.

- If appropriate, create an extract of the data you want to publish. For more information, see the following section, When to use an extract below.

- Develop a data source naming convention.

  After publishing a data source you cannot rename it directly. Instead, you need to publish a new copy with the new name, and then update all workbook connections. A well-considered naming convention can also help other users of the data deduce which data source to connect to.

- Consider designating the following roles among your Tableau users:
  
  - A data steward (or team) who creates and publishes the data sources for the Tableau community, which meet your organization’s data requirements.
  
  - A site administrator who manages published content, extract refreshes, and permissions on the server you publish to (Tableau Server or Tableau Online).

Central management helps to avoid data source proliferation. Authors who connect to managed data can be confident that the answers they find in it reflect the current state of the business.

When to use an extract

Under the following conditions you might be required or choose to publish an extract instead of connecting live.

Publishing data to Tableau Online that it cannot reach directly

Tableau Online in the cloud cannot reach data sources that you maintain on your local network. Depending on the connection, you might be required to publish an extract and set up a refresh schedule using Tableau Bridge.

Some cloud-hosted data sources always require extracts. These include Google Analytics, Salesforce.com, Oracle, OData, and some ODBC data sources. You can set up refresh schedules for some of these data sources directly on Tableau Online; for others you use Tableau Bridge.
Web data connector data sources always require extracts. If you connect to the data source using standard user name and password authentication, you can refresh it using Tableau Bridge. If you connect to the WDC data source using OAuth authentication, you will need to use an alternative method to refresh it.

For more about how Tableau Bridge supports both extract and live connections to data Tableau Online cannot reach directly, see Use Tableau Bridge to Expand Data Freshness Options in the Tableau Online Help.

**Improving performance**

Even if the server supports live connections to your data, an extract might make more sense. For example, if the database is large or the connection slow, you can extract a subset that includes only the pertinent information. The extract can be easier and faster to work with than connecting live.

In cases where you can use a live connection or an extract that you refresh on a schedule, you might want to experiment with both options to see which works best for you.

**Enabling functionality the data source does not inherently support**

For example, suppose you want to use the Median function with SQL Server data.

To learn more about creating data extracts, see Extract Your Data on page 773.

**Publishing data separately or embedded in workbooks**

You can publish data sources as standalone resources that workbooks connect to, or you can publish workbooks with the data sources included within them.

When you publish a workbook, if any connection specifies anything other than a Tableau data source published to the same project, the data is published as part of the workbook (sometimes referred to as embedded in the workbook).

When data is embedded in a workbook:

- Access to the data source is limited to the workbook in which you published it. Neither you nor other users can connect to that data from another workbook.
- You can set up extract refresh schedules as you do for data sources that you publish separately.
Each way of publishing has its advantages. The table below shows a few common points of comparison. It is not a comprehensive list, and these are generalizations. How these and other factors apply to you are specific to your environment.

<table>
<thead>
<tr>
<th>Published separately</th>
<th>Embedded in workbook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publishing data sources is a step toward centralizing data management. You can create policies geared toward minimizing data source proliferation and helping people find the right data for the work they do.</td>
<td>Each embedded data source has a separate connection to the data. Each has the potential to show something different than the other at any given time (and data source proliferation is common).</td>
</tr>
<tr>
<td>Meant to be shared; becomes available for other Tableau users to connect to.</td>
<td>Data is available only inside the workbook; it is not available for other Tableau Desktop users to connect to.</td>
</tr>
<tr>
<td>Without content management and self-service guidelines, seeing a long list of data sources to connect to can be confusing to users who rely on the data to do their work, and is more difficult to manage on the server.</td>
<td>Users create their own connections, and they know exactly what data they’re getting.</td>
</tr>
<tr>
<td>Someone who changes a shared data source might be uncertain or unaware of the effects that those changes have on connected workbooks.</td>
<td>Changing the data requires opening the workbook, where you can see the result of the change.</td>
</tr>
<tr>
<td>Even if effects of data source changes on connected workbooks are planned, updating those connected workbooks is cumbersome.</td>
<td>Same as above; however, if multiple workbooks use similar data and need to be updated, it might be worth connecting to a published data source instead.</td>
</tr>
<tr>
<td>Extracts can be refreshed on a schedule. You set up one refresh schedule for the extract, and all workbooks that connect to it always</td>
<td>Embedded extracts that aren’t refreshed can be useful for showing snapshots in time.</td>
</tr>
</tbody>
</table>
### Published separately vs. Embedded in workbook

<table>
<thead>
<tr>
<th>Published separately</th>
<th>Embedded in workbook</th>
</tr>
</thead>
<tbody>
<tr>
<td>show the most current data.</td>
<td>If you want to keep the data fresh, each workbook must have its own refresh schedule.</td>
</tr>
<tr>
<td>Generally helps you to optimize performance on the server or site.</td>
<td>Performance might be affected when the server contains multiple workbooks that connect to the same original data, and each workbook has its own refresh schedule.</td>
</tr>
</tbody>
</table>

### Keeping extracts up-to-date

When you publish a data source with an extract, you can refresh it on a schedule. The way you schedule refreshes depends on the data source type and whether you’re publishing to Tableau Server or Tableau Online.

For more information, see the following topics:

- [Keep Data Fresh on Tableau Online](#)
- [Keep Data Fresh on Tableau Server](#)

### Additional resources

- [Data Server](#)—Training video by Tableau, with a helpful overview of data sources and publishing.
- [Understanding Tableau Data Extracts](#)
  A version-agnostic, three-part series by Gordon Rose on the Tableau blog. It includes an in-depth look at the extract's file structure, guidelines for when to use extracts, and best practices.
- [O Extract, Where Art Thou? and TDE or Live? When to Use Tableau Data Extracts (or not)](#)
  Posts by Tableau Zen Master Jonathan Drummey on his blog Drawing with Numbers. Includes tips on extracts, explains the different file types, describes different publishing scenarios. (Read the comments, too.)
Prepare for Publishing a Workbook

When you publish a workbook, you need to make decisions that determine how others will access the views and the data behind them. You also need to take into account the experience you and others will have of interacting with the views on the server. Before you publish your workbook, use the information in this topic to help you make these decisions and improve the workbook’s performance.

Note: If you have a Tableau site administrator, check with them on whether your organization already has publishing guidelines that have already answered these questions.

In this article

- What makes up a typical published workbook
- Decide how to access the data and keep it up-to-date
- Assess the workbook performance and ease of use

What makes up a typical published workbook

During the publishing process, you select settings that determine how the pieces that make up the workbook interact with each other. The following image shows a simplified overview of a workbook’s structure.
A. The underlying (original) data that you connected to when you created the workbook. When you publish, you specify whether to include credentials for accessing this data, or whether to require users to enter credentials to access it.

B. The Tableau data source. This is where the action is. It contains the XML metadata that describes how to access the underlying data (A), field customizations or calculations you made in Tableau, when to refresh the extract if there is one.

In this image, the data is embedded in the workbook, and it contains an extract connection and refresh schedule, some calculations, and so on.

C. A view showing data from B, which you want to make available for your colleagues to edit or interact with on the server. When you publish, you can select the sheets—which include views, dashboards, and stories—you want to share.

Decide how to access the data and keep it up-to-date

During the publishing steps, you need to answer the following questions about the data connection. For help with the answers, see Best Practices for Published Data Sources on page 2499.

- Will you publish with a live connection to the data, or will you create an extract?
  
  If you publish live connections to Tableau Online, see Authorize Access to Cloud Data Published to Tableau Online on page 2533.

- Do you want to publish the workbook’s connections as separate, standalone Tableau data sources (and then connect the workbook to the published data); or embed the data
into the workbook?

**Tip:** If the workbook already connects to a Tableau data source, you keep the existing connection. That means you’ve just answered this question.

- Do you want the workbook to show the data when other users open the workbook, or do you want to require them to provide database credentials? (Either way, they must also have access to the site and project you publish to.)

This can be complicated depending on the data the workbook connects to. In many cases, the simple path to data described below works great. If it doesn’t work for you (or you’re not sure), see Set Credentials for Accessing Your Published Data on page 2530.

**The simple path to data authorization**

When you go through the publishing steps, you specify how your workbook users will access the data the workbook connects to. You do this in the **Data Sources** section of the Publish Workbook dialog box.

![Data Sources](image)

Depending on the connection type, the simple path might work for your environment. You would do one of the following:

**For a connection to a Tableau data source:** Keep the existing connection and embed the password.

**For other data connections:**

1. Before you open the Publish Workbook dialog box, create an extract and include only the data you need for the workbook.

2. In the Publish Workbook dialog box, embed the extract in the workbook and select **Allow**
refresh access. The latter embeds the credentials in the connection.

3. After you specify remaining settings and click Publish, set up a refresh schedule.

For the complete steps, see Comprehensive Steps for Publishing a Workbook on page 2521.

Scenarios where data authorization requires some extra effort

The simple path doesn’t work best for every situation. Some reasons you might need to divert from it include:

- Instead of embedding data into workbooks, your existing policies mandate publishing data sources separately and connecting workbooks to the Tableau published data sources.

  This requires a few more steps than embedding the data; however, where there is a choice between the two, publishing data separately and managing it on the server is considered a better practice.

- You added a user filter to enforce row-level security. In this scenario, publishing live connections is more common, and other steps are required to secure the filter.

- You’re publishing to Tableau Server, and you use SAP HANA or Impala single sign-on, or your workbook connects to data that can use impersonation-based authorization (SQL Server or Kerberos-enabled data).

- Other reasons that have to do with your organization’s data security or authorization policies, which your IT staff can help you with.

Even if whatever you decide turns out to be not the best path, you can easily correct it by republishing. In some cases, your site administrator can change the settings on the server directly.

Assess the workbook performance and ease of use

Keep performance and interaction best practices in mind as you build workbooks you want to publish. If changes take a long time to display while you’re working in Tableau Desktop, they will take as long or longer to display on the server. Simple steps you can take that can have a big impact include limiting the number of marks you add to a view, limiting the number of views you add to a dashboard, and removing unused fields from your data source.

For additional tips, see the following topics:
Make Workbooks Compatible Between Versions

You’ve created a workbook, and would like to share it or get edits from other Tableau users. When you save your workbook, you need to make sure that the person you send your work to can open it.

This article describes which combinations of Tableau Desktop and Tableau Server can work together and how to export your workbook as another version of Tableau Desktop.

In this article

When does compatibility matter? below
Compatibility between versions of Tableau Desktop on the next page
Compatibility between Tableau Desktop and Tableau Server on page 2511
Downgrade your workbook on page 2513

When does compatibility matter?

Compatibility issues typically arise only between release versions of Tableau products. For example, if you are using version 10.5 of Tableau Desktop and version 10.2 of Tableau Server. There are no compatibility issues with dot releases (maintenance versions)—for example, between Tableau Desktop version 10.4.1 and Tableau Server version 10.4.2.
There is one exception. Tableau sometimes introduces new connector support with a dot release. This can introduce compatibility issues between Tableau Desktop and Tableau Server if Tableau Desktop has been upgraded but Tableau Server hasn't been.

For details about data source support by release, see the Release Notes page.

Compatibility between versions of Tableau Desktop

Tableau Desktop users can share workbooks—for example, by emailing them or copying them to a location on the company network. But if two users are using different versions of Tableau Desktop, sharing isn't always possible. Tableau Desktop is backward compatible, which means:

- **A newer version of Tableau Desktop can open a workbook created with an older version.**
- **An older version of Tableau Desktop can't open a workbook created with a newer version unless it is downgraded to that older version.**

For example, if Amanda creates a Tableau workbook with Tableau Desktop version 10.3 and sends it to Zachary, who has Tableau Desktop version 10.5, Zachary will be able to open the workbook. He'll see this message:

*The workbook will be upgraded when it is saved. The upgraded file can't be read by earlier versions of the application.*

If Zachary edits the workbook and sends it back to Amanda, he will need to downgrade the workbook so that she can open it. If not, the load will fail when Amanda tries to open it in Tableau Desktop version 10.3, because the workbook is now a 10.5 workbook.

For more information about downgrading your workbook between Tableau Desktop versions, see Downgrade your workbook on page 2513 in this article.

Working with Tableau extracts

The same logic applies to Tableau extracts: a newer version of Tableau Desktop can open an extract created with an older version. But an older version of Tableau Desktop can’t open an extract created with a newer version.

However if you create an extract in an older version and open the workbook in Tableau Desktop 10.5 and later and refresh the extract, it will be converted to the .hyper format and the workbook will become incompatible with older versions of Tableau.
If you know you will be sharing your workbooks with users that have Tableau Desktop version 10.5 or later, save a copy of your workbook first before sharing it. If the workbook has already been opened in version 10.5 or later, you can downgrade the workbook to an earlier version if the extract hasn’t been refreshed already.

For more information about the .hyper extract format, see Extract Upgrade to .hyper Format. For more information about how to downgrade your workbooks to prior versions, see Downgrade your workbook on page 2513 in this article.

Compatibility between Tableau Desktop and Tableau Server

A workbook created with any version of Tableau Desktop can connect to any supported version of Tableau Server. But once you are connected, compatibility can be an issue depending on the version you are using and the action you’re performing.

Publishing from Tableau Desktop to Tableau Server

You can publish workbooks and data sources to Tableau Server if the version of Tableau Server is the same or newer than the version of Tableau Desktop. For example, you can publish workbooks and data sources from Tableau Desktop 10.0 to Tableau Server 10.0, Tableau Server 10.1, or Tableau Server 10.2.

But if Tableau Server is running an earlier versions than Tableau Desktop, you will be prompted to downgrade the workbook before you can publish it to Tableau Server. For example, you can downgrade a workbook created on Tableau Desktop version 10.5 to publish it to Tableau Server version 10.2. For more information about how to downgrade your workbook, see Downgrade your workbook on page 2513 in this article.

Note: If your workbook contains an extract using the .hyper format or if you are running Tableau Server version 10.1 and earlier, the downgrade workbook features are not available.

Downloading workbooks from Tableau Server or Tableau Online

A workbook downloaded from Tableau Server has the same version as the version of Tableau where it was last edited. For example, suppose Amanda is running Tableau Desktop 10.3 and
publishes a workbook to Tableau Server 2018.1. If Zachary is also running Tableau Desktop 10.3, he can connect to Tableau Server, download the workbook, and open it.

However, compatibility issues can arise:

- If someone else edits the workbook on the server, the workbook is updated to that version of Tableau Server.
- If someone downloads the workbook from Tableau Server to a newer version of Tableau Desktop and edits, saves, and publishes it, the workbook is updated to the newer version of Tableau Desktop.

For example, suppose Elizabeth publishes a workbook from Tableau Desktop 10.0 to Tableau Server 10.3. Ted downloads the workbook from Tableau Server to Tableau Desktop 10.3. He edits and republishes the workbook back to the server. The workbook is now updated to the newer version. Zachary and Elizabeth, who are both using Tableau Desktop 10.2 can download the workbook but can't open it because the workbook has been updated to Tableau version 10.3.

If you edit or create a workbook on a newer version of Tableau Server or Tableau Online, you can downgrade the workbook to be opened in an older version of Tableau Desktop.

For more information about downgrading your workbook between Tableau Server or Tableau Online and Tableau Desktop, see Downgrade your workbook on the next page in this article.

Connecting to published extracts

If using Tableau Server version 10.5 and later, published extracts are automatically upgraded when you refresh them to the new .hyper extract format. If this happens, the workbook can no longer be opened using previous versions of Tableau Desktop. For more information about the .hyper extract format, see Extract Upgrade to .hyper Format.

For previous versions of Tableau Server, any published workbook or data source with an extract created in Tableau Desktop retains its original version when it's refreshed or appended to on Tableau Server.

Connecting to Tableau Server-based data sources

Workbooks in Tableau Desktop can use data sources that have been published to later versions of Tableau Server. However, if you connect to a data source that uses features that aren't in that earlier version of Tableau Desktop, those fields are unavailable.
For example, suppose you create a workbook in version 8.2 of Tableau Desktop, and then connect to a data source on Tableau Server version 9.0 that contains fields that include level of detail expressions. In that case, those data fields are unavailable in the Tableau Desktop 8.2 workbook, because level of detail expressions were not available until Tableau 9.0.

Workbooks created in Tableau Desktop can always use data sources from an earlier version of Tableau Server. For example, a Tableau Desktop 10.1 workbook can use data sources from Tableau Server 10.0.

---

### Downgrade your workbook

If you need to publish your workbook to an earlier version of Tableau Server or share your workbook with someone using an earlier version of Tableau Desktop, you can downgrade your workbook to a previous version. The downgraded workbook can be opened in the selected version and later.

Any features not available in that previous version will be removed when the workbook is downgraded. The earliest version that you can downgrade to is Tableau Desktop 10.2.

**Note:** These features are not available for workbooks that include a Tableau Data Extract that has been refreshed in version 10.5. The extract will have been converted to use the new .hyper format and this format can't be downgraded back to a .tde format. For more information about the .hyper extract format, see Extract Upgrade to .hyper Format.

### Tableau Desktop to Tableau Desktop

To downgrade the workbook to be accessible on previous versions of Tableau Desktop, do the following:

1. In Tableau Desktop select **File > Export As** and select the version that you want to downgrade to.
2. Save the workbook to your My Tableau Repository or a selected location.

**Tableau Desktop to Tableau Server**

If you are using a newer version of Tableau Desktop and an older version of Tableau Server, you can downgrade the workbook to the older Tableau Server version during publishing.

When you publish the workbook, you will see a warning message that tells you that the workbook will be downgraded.
If you continue to publish the workbook, any features or functionality that aren't available in the older version are removed for compatibility.

If you open the downgraded workbook in the newer version of Tableau Desktop, you will see a message like the one shown below.

You may need to add back the features that were removed when the workbook was downgraded.

**Note:** This option doesn't apply to data sources.
Tableau Online or Tableau Server to Tableau Desktop

If you are using Tableau Server or Tableau Online, you can downgrade a workbook to be opened in an older version of Tableau Desktop.

In Tableau Server or Online, open the view you’d like to downgrade. Select Download > Tableau Workbook and select the version that you want to downgrade to. The workbook will download to your computer.

Any features or functionality that aren’t available in the older version are removed for compatibility.

Publish a Data Source

When you are ready to make a data source available to other Tableau Desktop users, you can publish it to Tableau Server or Tableau Online. If the data source is in a workbook that you published to Tableau Server or Tableau Online, you can make it available by saving it, provided it’s an embedded Excel or text file. For details, see Save a Data Source on the Web on page 2521.

Note: If you haven’t yet read about best practices for creating data sources and when to create an extract, see Best Practices for Published Data Sources on page 2499.

In this article

- Publishing steps below
- Publishing with a Web Data Connector on page 2520
- Using hidden fields in workbooks on page 2520

Publishing steps

The following steps give an overview of the publishing flow you will use regardless of the type of data or the server you publish to. Below these steps you can find supplemental information for authentication types and using the Tableau Bridge to refresh extracts of local data sources.
1. Select **Server > Publish Data Source**.

   If your workbook is connected to multiple data sources, select the one you want from the submenu.

![Server](image)

2. If you’re not already signed in to Tableau Server or Tableau Online, sign in now.

   How you sign in depends on how your administrator set up your environment. For information, see **Sign in to Tableau Server or Online** on page 310.

3. In the **Publish Data Source** dialog box, do the following:

   - Select the project you want to publish to and enter the data source name.
   - Add a description and tags that will help you and other users find it.

   Separate tags using either a comma or a space. To add a tag that contains a space, put it in quotation marks (e.g., “Sales Quotes”).
- For **Permissions**, accept the default project settings. Generally a site administrator manages permissions on the server. If you think your data source is an exception, work with your administrator to determine the best course of action, and see *Set Permissions as You Publish a Data Source or Workbook* on page 2526.

- For **Authentication**, if you need to provide credentials to access your data, you can specify how authentication should be handled when the data is published to the server.

The options available for accessing the data source depend on the type of data you publish and whether you are publishing to Tableau Server or Tableau Online.
Information appears at the bottom of the dialog box to let you know whether you need to take further action, such as adding Tableau Online to your data provider's authorized list.

For information about the authentication types, see Set Credentials for Accessing Your Published Data on page 2530.

4. If you are publishing file-based data that is on a Windows mapped drive, or using images that will not be available from the server, select Include external files.

When you include external files, copies of the files are put on the server as part of your data source. Copies of files are also put on the server and included as part of the data source when you publish extracts of multi-connection data sources that contain a connection to file-based data such as Excel. For more information about the implications of publishing extracts of multi-connection data sources, see Extracts of multi-connection data sources that contain connections to file-based data (Tableau Desktop only) on page 669

If you do not want to publish the external files to the server, change the connection information so that the data source references a full UNC path. For example, rather than connecting to D:\datasource.xls, you would connect to \filesrv\datasource.xls.

5. By default, during the publishing process, Tableau updates the workbook connection to use the new published data source. It also closes the local data source.

To continue using the local data source instead, clear the Update workbook to use the published data source check box.

Note: If you select Undo after publishing the data source, Tableau will revert to using the local data source, but the data source will remain published. In addition, Tableau does not replace a local data source when you publish a cube.
(multidimensional) data source to Tableau Server. (Tableau Online does not support publishing cube data sources.)

6. Click **Publish**.

7. (Optional) Set up a refresh schedule on the server.

   After you click **Publish**, you are redirected to the site you published to, open in the web browser. There, you are prompted to set up the refresh schedule. If you published with a live connection to on-premises data, or an extract of a data source to Tableau Online that Tableau Online cannot reach directly, you schedule refreshes using Tableau Bridge.

   For more information about refreshes and schedules, see any of the following topics:
   - Schedule Refreshes on Tableau Server
   - Schedule Refreshes on Tableau Online
   - Schedule Refreshes Using Tableau Bridge

**Publishing with a Web Data Connector**

To publish a web data connector data source, you need to *import* the web data connector to the server before you can set up a refresh schedule. You can do this only on Tableau Server.

You can refresh some web data connector data sources on Tableau Online, using Tableau Bridge.

For information, see Web Data Connectors in Tableau Server in the Tableau Server Help or Use Tableau Bridge to Expand Data Freshness Options in the Tableau Online Help.

**Using hidden fields in workbooks**

Workbooks connected to a published data source respect the state of hidden fields in the published the data source.

- If you create a new workbook that uses a published data source with hidden fields, those fields remain hidden in the workbook and cannot be used in calculations, sets, groups, and other object creation.

- If you work with an existing workbook that uses a published data source with hidden
fields, those hidden fields are displayed in red in the workbook to indicate that the fields, and therefore the views and calculations that use those fields, are invalid.

You can address this issue in one of the following ways:

- Show (unhide) the relevant fields in the data source, and then republish the data source.
- Update the relevant workbooks to exclude the hidden fields.

For information, see Hide or Unhide Fields on page 971.

See also

- Keep Data Fresh (Tableau Online)
- Data Sources (Tableau Server)

Save a Data Source on the Web

You can save data sources to share or reuse from your published workbooks on Tableau Server. To do this, the workbook must be connected to a live Excel or text data source. You'll also need to have the specific permissions to edit the workbook and publish on Tableau Server. For more about publishing permissions, see Who can publish content in the Tableau Server Help.

1. On Tableau Server, open a workbook that connects to a live Excel or text data source.
2. Click the tab for any worksheet.
3. From the Data pane, click the drop-down menu for the data source you want to share, and click Save.

Comprehensive Steps for Publishing a Workbook

The steps below describe the general publishing process. Use the information you gathered while reviewing Prepare for Publishing a Workbook on page 2505 to adapt these steps for your workbook and data.

If this is your first time publishing, you might publish as a test first, and work out any glitches before you let other users know the workbook is ready for them.
1. In Tableau Desktop, open the workbook you want to publish, or create a new one and add some data to the view.

2. Select Server > Publish Workbook.
   
   If the Publish Workbook option does not appear on the Server menu, make sure a worksheet or dashboard tab is active (not the Data Source tab).

   If necessary, sign in to a server. For Tableau Online, enter https://online.tableau.com. For more information, see Sign in to Tableau Server or Online on page 310.

3. In the Publish Workbook dialog box, select the project, enter a name for the workbook, and add search tags.

   Learn more

   Projects are containers for workbooks and data sources. Permissions set on projects determine who can access your workbook, so knowing which project to publish to is important.

   For the workbook name, enter a unique name unless revision history is enabled, and you want to create a new revision of an existing workbook you own.

   A good practice is to create a workbook naming convention that helps others find workbooks easily and know what each contains. You won’t be able to rename the workbook on the server.

   Note: If you’re publishing to Tableau Server, and your administrator has turned off revision history, using an existing name overwrites the workbook. Revision history is enabled for all Tableau Online sites.

   Add metadata keyword tags

   Enter keywords that describe the workbook. Tags help users find related workbooks when they browse the server.

   Separate tags using a comma or space. To add a tag that contains a space, put the tag in quotation marks.
4. For **Permissions**, accept the default project settings. Generally a site administrator manages permissions on the server. If you think your workbook is an exception, work with your administrator to determine the best course of action, and see Set Permissions as You Publish a Data Source or Workbook on page 2526.

5. For **Data Sources**, select **Edit** if you want to change the method for how people to access the underlying data your workbook connects to, or to change how the data is published (embedded in the workbook or published separately).
   - If you're publishing an extract, and you want to set up a refresh schedule, you must select **Embed password** or **Allow refresh access**, depending on whether your underlying data is on-premises or in the cloud.
   - If your workbook connects to a Tableau data source, we recommend embedding the password. This is the default option if you are publishing one of the workbook's connections separately. Selecting Prompt users requires additional permissions on the Tableau data source.

   For more information, see Set Credentials for Accessing Your Published Data on page 2530.

6. Configure other settings that are available for this workbook.

   For more information, see Variable publishing options on the next page.

7. Click **Publish**.

8. (Optional) Set up a refresh schedule for each extract you published.

   The publishing workflow guides you through these steps. For some data types you publish to Tableau Online, the publishing process starts Tableau Bridge on your computer.
Variable publishing options

The following sections describe publishing options that appear when they’re appropriate for the workbook.

Specify how to generate thumbnails for workbooks with user filters

On the server where you publish, one of the options for browsing content is by thumbnail view. In this view, thumbnails are generated based on the workbook and its sheets. If your workbook contains user filter, you can specify which user’s filter to use for creating the thumbnails.

For example, if you want the thumbnail image to show all regions of a sales forecast, you can generate thumbnails per the user who is allowed to see all regions.

In the following scenarios, a generic image appears in place of the view thumbnail.

- The user you select does not have permission to see the data.
- The data is from a Tableau Server data source that utilizes data source filters, user calculations, impersonation, or other user references.

To learn more about user filters, see Restrict Access at the Data Row Level on page 2541.

Show or hide sheets when you publish

By default, Tableau Desktop publishes all sheets in a multiple-sheet workbook. In the Sheets section of the Publish Workbook dialog box, you can specify which sheets to include. Hiding
sheets is useful when you want to publish a dashboard or story without showing the worksheets that were used to create it.

Important: Hiding sheets is not a security measure. Anyone who has the Download/Web Save As capability can access the hidden sheets. Other editing permissions can also allow access to hidden sheets. For more information, see Content Access and Ownership in the Tableau Server Help.

Show sheets as tabs

If you select multiple sheets to show, you can specify how users navigate them.

- Select the Show Sheets as Tabs check box to provide tab-based navigation.

- Clear the check box to allow people to open only one view at a time.

Show selections

You can select this if you want a particular portion of the view to be highlighted when others open the workbook. Make your selections in Tableau Desktop before you start the publishing process.
Include external files

If your views contain any information that isn’t available to the server, or to other users who access the workbook on the server, the Publish Workbook dialog box will include a setting you can select to include that information when you publish.

For example, you might use an Excel, CSV, or other data source local to your computer; image files; and so on.

If you are publishing to Tableau Server, and the workbook references data sources or images on a mapped drive, you can include external files when you publish, or you can change the connection information so that the workbook references the UNC path to the data source. For example, you could change D:\datasource.xls to \filesrv\datasource.xls.

If you are publishing to Tableau Online, and the workbook connects to a data source that Tableau Online cannot connect to directly, such as one that is stored on your local network, select the Include External Files check box.

Set Permissions as You Publish a Data Source or Workbook

As the publisher of a workbook or data source, you can set permissions as part of the publishing process. Permissions allow or deny other users access to your published content on Tableau Server or Tableau Online. For example, who can interact with views in a workbook, edit them, or save changes to them, who can download a copy of a data source, and so on.

Permissions are separate from the type of access you set on the connection to the data. Accessing some data types requires signing in using a database name and password or embedding database credentials into the connection. For information about that, see Set Credentials for Accessing Your Published Data on page 2530.

Back to permissions, with few exceptions, your best choice will be to accept the default settings in the publishing dialog box. If you agree, you can return to the Comprehensive Steps for Publishing a Workbook on page 2521, and save some time not reading the rest of this topic.

The remaining sections give context and tips for when you think your workbook or data source might be an exception to the default settings, and shows you how to change the permissions.
About setting permissions during publishing

When you start the publishing process, the dialog box shows the permissions that will be applied. By default, the content you publish takes the capabilities that are already set on the server, typically as they’re set on the project you are publishing to.

When you change permissions in the publishing dialog box, you are setting capabilities explicitly for the content you’re publishing. The effect is that the workbook or data source you publish no longer inherits changes made at its parent level, such as the project. Depending on your environment, this might be as you intend, or it might conflict with the guidelines your administrator has set and have unintended consequences.
Tips for deciding whether to set permissions explicitly

- **Learn your organization’s practices**
  
  Consult with your Tableau administrator to learn the guidelines for your organization. It’s common (and recommended) practice for an administrator to manage permissions on your Tableau Server or Online site. If you work in such an environment, even if you set permissions during publishing, the person who manages permissions on the server might change these settings afterward.

- **Know the consequences of setting explicit permissions**
  
  In addition to the potential conflicts described earlier, explicit permissions on some content requires extra maintenance to keep track of which content has exceptions, and which exceptions are applied.

- **Publish quickly by accepting the default permissions settings**
  
  If necessary, you or your administrator can update permissions on the server afterward, where you have a more comprehensive view into the effects of your changes.

How to set explicit permissions during publishing

1. In the publishing dialog box, next to the summary that indicates the current settings, click **Edit**.

![Permissions dialog box](image)
2. In the popup that appears, do one of the following:

   - To set custom capabilities or assign a role explicitly, select an existing user or group and click **Edit**, or click **Add**.

   In the **Add/Edit Permissions** dialog box, make your changes.

   Click **Apply** to save changes and keep the dialog box open to configure another user or group. Click **OK** to close the dialog box.

   ![Add/Edit Permissions dialog box](image)

   - To remove roles or capabilities that are set explicitly, select the user or group, and then click **Remove**.

**Predefined roles you can assign**

When you publish a workbook, you can assign any of the following predefined roles to a selected user or group:

   - **Viewer**: Allows the user or group to view the workbook on the server, as well as add and view comments.

   - **Interactor**: Allows the user or group to view the workbook on the server, edit workbook views, apply filters, view underlying data, export images, and export data. All other
capabilities are inherited from the user’s group and project permissions.

- **Editor**: Allows all capabilities to the user or group.

When you publish a data source, you can select from these roles:

- **Data Source Connector**: Allows the user or group to connect to the data source on the server.

- **Data Source Editor**: Allows the user or group to connect to the data source on the server and to publish, edit, download, delete, set permissions, and schedule refreshes for the data source.

  **Note**: If you are not the content owner or an administrator, you cannot schedule refreshes directly on the server.

**Set Credentials for Accessing Your Published Data**

When you publish a workbook to Tableau Online or Tableau Server, you can publish the data source it connects to as part of the workbook (*embedded* into the workbook), or as a separate, standalone data source. In addition, if the data source you’re publishing requires authentication, you can customize how credentials are obtained.

The type of authentication to your data source is independent of how people sign in to your Tableau Online or Tableau Server site. For example, to give people direct access to the data in a workbook, you would embed a database user’s credentials into the data source’s connection. But anyone viewing the workbook would still need to be able to sign in to the site on Tableau Online or Tableau Server to open your workbook.

This topic describes how to set authentication on data connections as part of the publishing process.

In this article

- **Set the authentication type** on the next page

- **Dropbox, OneDrive connections** on page 2532

- **Workbook connections to Tableau data sources** on page 2532
Set the authentication type

For many types of connection you can embed a database user’s name and password, or use single sign on (SSO). Specific exceptions are described later in this topic.

The following steps describe how to set authentication as part of publishing a data source or workbook. You can do this for each connection in the data source.

1. In the Publish Workbook dialog box, go to the **Data Sources** area, which lists the workbook’s connections, and select **Edit**.

2. In the **Manage Data Sources** popup, after you decide whether to publish the data source separately or as part of the workbook, select an authentication type for each connection in the data source.

The available authentication types depend on the connection type, and they can include one or more of the following:

- **Prompt user**: Users must enter their own database credentials to access the published data when the view or workbook loads.

- **Embedded password**: The credentials you used to connect to the data will be saved with the connection and used by everyone who accesses the data source or workbook you publish.

- **Server run as account**: A single Kerberos service account will be used to authenticate the user. On Windows this is the account that Tableau Server runs as. On Linux it can be any Kerberos account.

- **Viewer credentials**: The viewer’s credentials are passed through to the database using SSO (usually Kerberos).

- **Impersonate with embedded account** or **Impersonate with server Run As service account**: Impersonation using embedded credentials connects with the embedded credentials and then switches to the viewer’s identity (only for databases that support this). Impersonation using the Run As service account is similar but first, connects with the Kerberos service account before switching to the viewer’s identity.

**Note**: This topic does not apply to connections to that do not require authentication, such as text files or Excel files.
- Refresh not enabled or Allow refresh access: These options appear when you publish an extract of cloud data such as from Salesforce, and database credentials are needed to access the underlying data. Allow refresh access embeds the credentials in the connection, so that you can set up refreshes of that extract on a regular schedule. Setting Refresh not enabled prompts users when they open the workbook.

Important: How you want to keep extracted data fresh is also a factor. If you want to set up an automatic refresh schedule, you must embed the password in the connection. In addition, if you’re publishing a cloud data connection to Tableau Online, the publishing steps will alert you if you need to add Tableau Online to the data provider’s authorized list.

Dropbox, OneDrive connections

For Dropbox and OneDrive, when you publish a data source or workbook and select Embedded password, Tableau creates a saved credential and embeds it in the data source or workbook.

Workbook connections to Tableau data sources

When you publish a workbook that connects to a Tableau Online or Tableau Server data source, rather than setting the credentials to access the underlying data, you set whether the workbook can access the published data source it connects to. Regardless of the original data type, the choice for server data sources is always Embedded password or Prompt users.

If you select to prompt users, a user who opens the workbook must have View and Connect permissions on the data source to see the data. If you select embed password, users can see the information in the workbook even if they don’t have View or Connect permissions.

See also

- If you publish to Tableau Server, see Edit Connections in the Tableau Server Help.
- If you publish to Tableau Online and the workbook connects to Salesforce, Google Analytics, Google Sheets, Google BigQuery, OneDrive, Dropbox, and QuickBooks Online data, see Refresh Data Using Saved Credentials in the Tableau Online Help.
- If you are a Tableau Server administrator looking for more information about
authentication, see the Tableau Server help topics, "Authentication" (Windows | Linux) and "Data Connection Authentication" (Windows | Linux).

Authorize Access to Cloud Data Published to Tableau Online

The information in this topic applies to you if you publish workbooks or data sources to Tableau Online that contain live connections to cloud data—for example, Amazon, Google, Salesforce.com.

As a security measure, cloud data providers might require you to supply a list of authorized IP addresses from which external applications request access to your data. A request from an IP address that is not explicitly approved could be rejected. To make sure live connections you publish to Tableau Online remain uninterrupted, add Tableau Online to your data provider’s authorized list.

The table lists IP address ranges Tableau Online uses, depending on your site location. You can see its location in the URL that appears after you sign in to Tableau Online.

<table>
<thead>
<tr>
<th>Host Name (Instance)</th>
<th>Site Location</th>
<th>IP Address or Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>10ax.online.tableau.com</td>
<td>N. America</td>
<td>34.208.207.197</td>
</tr>
<tr>
<td></td>
<td></td>
<td>52.39.159.250</td>
</tr>
<tr>
<td>10ay.online.tableau.com</td>
<td>N. America</td>
<td>34.218.129.202</td>
</tr>
<tr>
<td></td>
<td></td>
<td>52.40.235.24</td>
</tr>
<tr>
<td>10az.online.tableau.com</td>
<td>N. America</td>
<td>34.218.83.207</td>
</tr>
<tr>
<td></td>
<td></td>
<td>52.37.252.60</td>
</tr>
<tr>
<td>us-east-1.online.tableau.com</td>
<td>N. America</td>
<td>50.17.26.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>52.206.162.101</td>
</tr>
<tr>
<td>us-west-2b.online.tableau.com</td>
<td>N. America</td>
<td>34.214.85.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.214.85.244</td>
</tr>
<tr>
<td>dub01.online.tableau.com</td>
<td>EU</td>
<td>34.246.74.86</td>
</tr>
<tr>
<td>Host Name (Instance)</td>
<td>Site Location</td>
<td>IP Address or Range</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>52.215.158.213</td>
</tr>
<tr>
<td></td>
<td></td>
<td>185.92.123.0/29</td>
</tr>
<tr>
<td>eu-west-1a.online.tableau.com</td>
<td>EU</td>
<td>34.246.62.141</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.246.62.203</td>
</tr>
</tbody>
</table>

These addresses are dedicated to and controlled by Tableau.

**Note:** In addition to enabling communication over the Tableau Online IP range, you might need to enable access over the appropriate database port.

## Find authorization steps for your data provider

The following links take you to the steps on common data providers’ websites for authorizing external applications on their platforms.

Amazon:
- Redshift
- RDS
- EC2

Microsoft Azure

Google Cloud Platform

**Disclaimer:** The links in the list above take you outside of Tableau.com. Although we make every effort to ensure links to external websites are accurate, up to date, and relevant, Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the external site for answers to questions regarding its content.

## See also

[Keep Data Fresh](Tableau Online Help)
Publishers: Use Tableau Bridge to Keep Tableau Online Data Fresh

For published data sources you cannot refresh directly on Tableau Online, you can use Tableau Bridge to keep the data fresh. For example, use Tableau Bridge when the published data source connects to data you maintain behind a firewall on your network (on-premises).

You can use Tableau Bridge in the following ways:

- Maintain live connections to on-premises data
- Maintain published extract data sources that Tableau Online cannot reach directly

In this article

How Tableau Bridge relates to the Online sync client
below
Maintain live connections to on-premises data on the next page
Schedule an extract refresh from Tableau Online for Tableau Bridge on the next page
1. Publish the data source on page 2537
2. Set up the refresh schedule on page 2537
3. Embed database credentials into the connection information on page 2538
4. Consider some next steps on page 2539

How Tableau Bridge relates to the Online sync client

Tableau Bridge is the Online sync client plus live query functionality. Bridge was released with Tableau version 10.3, at which point it replaced the sync client.

Live query functionality enables you to maintain live connections between data sources published to Tableau Online and on-premises relational data.

Some feature notes:
If you use the sync client to refresh extracts, scheduled refreshes will continue to work as they always have.

Your site administrator must enable the option to maintain live connections.

As the sync client did, Tableau Bridge refreshes data sources that you publish separately, not data that's embedded in a published workbook.

If you’re a site administrator, see Use Tableau Bridge to Expand Data Freshness Options to learn more about the ways you can allow publishers to use Tableau Bridge.

Maintain live connections to on-premises data

After your administrator enables support for maintaining live connections, you will see this option during the publishing process. It is available when your workbook or data source contains a live connection to a relational database on your network, such as Microsoft SQL Server or MySQL.

After you publish, an available client handles the live queries. That’s all there is to it. To get started, publish a data source to Tableau Online, and select the option to maintain a live connection. Or, publish a workbook, and select the option to publish the data source separately, and then specify a live connection.

Schedule an extract refresh from Tableau Online for Tableau Bridge

For extract refresh schedules, Tableau Bridge works exactly as the sync client did. To schedule a refresh, you need to publish your extract data source separately, whether you go through the publish data source process or the publish workbook process.

Notes:

- Alternatively, you can also ask your site administrator to reassign a Tableau Bridge data source to a different Tableau Bridge client or to manage the data source using a centrally-managed client. For more information, see Manage data sources using a centrally-managed Tableau Bridge client.
- When an extract refresh is performed on extracts created in Tableau 10.4 and earlier (that is, a .tde extract), the extract is upgraded to .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, you won't be able open the
extract with earlier versions of Tableau Desktop. For more information, see Extract Upgrade to .hyper Format on page 788.

The following steps describe how to schedule the extract refresh during the publishing process from Tableau Online. To schedule an extract refresh for a data source that has already been published to Tableau Online, see Create a Schedule for a Tableau Bridge Data Source.

1. Publish the data source

   1. In Tableau Desktop, open the workbook that contains the data source you want to publish.
   2. Select Server > Publish Data Source.

      If you need to sign in, enter https://online.tableau.com, and then enter your Tableau Desktop credentials.

      Note: If you don't see the computer associated with your Tableau Bridge client during the publishing process, see

2. Set up the refresh schedule

   1. In the confirmation message that appears at the end of the publishing process, select Schedule using Tableau Bridge.

      ![Publishing Complete dialog]

      2. For Start Tableau Bridge on, select your computer, configure the schedule settings, and select Set schedule.
This starts a new Tableau Bridge client on your computer. Alternatively, you can select an existing Tableau Bridge client from the drop-down list. To assign an extract refresh to a Tableau Bridge client, you must be signed in to that client, and you must be the data source owner or a Tableau Online site administrator. You can’t assign an extract refresh to another user’s Tableau Bridge client.

A confirmation alert for the data source appears in the lower-right area of your screen, and the Tableau Bridge system tray icon appears in the Windows task bar.

**Note:** When you click **Set schedule**, you might see an alert from your web browser, letting you know it has received a request from Tableau Online to start an external application. The external application refers to Tableau Bridge. Allow the browser to start the application.

3. **Embed database credentials into the connection information**

If you need to provide a user name and password to access the underlying data in the published data source, you must embed those database credentials in the Tableau Bridge client to complete the refresh schedule settings.

1. From the Windows system tray on your computer, select the Bridge icon. It shows a red exclamation point to indicate that refreshes cannot run until you resolve an issue.

2. In the Edit Database Credentials dialog, enter your database credentials.
Note: You can click Test Connection to validate that the client can reach the database with the credentials you provided. This option is not available when connecting to file-based data.

4. Consider some next steps

- If you published a file-based data source, you might need to change the file path that the client references. For more information, see Change the file path for a linked data source.

- If you have existing workbooks that use the same data, and you want them to be refreshed on the same schedule as the published extract, you need to edit the workbooks to use the published extract instead of their own static (embedded) data sources. For information, see Replacing Data Sources.

- If you want Tableau Bridge to refresh extracts when you’re logged off Windows, you can run the client as a service (Service mode). For more information, see Run Tableau Bridge as a Windows service.

To go a step further, you can take your computer out of the equation completely, and run Tableau Bridge as a service on a centralized computer that is always on. Ask your administrator whether you have a shared Tableau Bridge client for running extract refreshes. For more information, see Manage extract data sources using a central Tableau Bridge client.
Schedule Extract Refreshes as You Publish a Workbook

When you publish workbooks that connect to extracts, you can set up a schedule for updates (refreshes) for those extracts, so the views in those workbooks stay current.

The ways you can set up and manage extract refresh schedules depends on where you publish and on the original data type.

- When you publish to **Tableau Server**, the schedule runs and is managed on the server.
- When you publish to **Tableau Online**:
  - If extracts are from cloud data (for example, Google Analytics or Salesforce), schedules run and are managed on Tableau Online.
  - If extracts are from on-premises or web data connector (WDC) data, you set up and manage refresh schedules using Tableau Bridge.

**Note:** In many organizations a server content manager or administrator manages all refresh schedules. We recommend that, before you publish, you find out from your administrator whether they have guidelines for these schedules.

Set up a schedule as you publish content

During the publishing process, after you click the **Publish** button, the scheduling options appropriate for your data types and publishing destination appear.

If you are publishing a multi-connection data source, you need to set a refresh schedule for each extract connection in the data source.

To use refresh schedules with connections to a database that requires you to sign in to it, you must store (embed) a database user’s credentials with the connection. For more information, see **Set Credentials for Accessing Your Published Data** on page 2530.

**Publish and refresh on Tableau Online**

For the subtleties around refreshing each type of data you publish to Tableau Online, see **Keep Data Fresh**. Here are examples of things you need to know:
- If your workbook connects to cloud data, add Tableau Online to your cloud data provider’s authorized list (whitelist). The IP address range for your site location is listed in Authorize Access to Cloud Data Published to Tableau Online on page 2533.

- Tableau Bridge starts during the publishing process if your data source or workbook connects to on-premises data.

  Tableau Bridge supports standard database authorization and cannot refresh data you connect to through OAuth. To learn more, see Use Tableau Bridge to Expand Data Freshness Options.

- When you publish a multi-connection data source to Tableau Online, if any one connection requires using Tableau Bridge, you must use Bridge to refresh all connections in the data source.

  For example, say you publish a data source with an extract connection to a MySQL data hosted in the cloud, and a live connection to an on-premises SQL Server database. Although Tableau Online supports refreshing hosted MySQL data, in this case, you would need to use the Bridge client to keep both the SQL Server and MySQL connections fresh.

  For information about using both live and extract connections in a data source, see About working with multi-connection data sources.

Refresh web data connector extracts

When you publish a workbook with a web data connector (WDC) data source, you must import the web data connector to Tableau Server before you can set up a refresh schedule. You can do this only on Tableau Server. For information, see Web Data Connectors in Tableau Server in the Tableau Server Help.

For information about refreshing web data connector data sources published to Tableau Online, see Use Tableau Bridge to Expand Data Freshness Options in the Tableau Online Help.

Restrict Access at the Data Row Level

When you share workbooks with others by publishing them to Tableau Server or Online, by default, all users who have access to the workbooks can see all of the data shown in the views. You can override this behavior by applying a type of filter that allows you to specify which data “rows” any given person signed in to the server can see in the view.
How user-based filtering works

Suppose you created a quarterly sales report for a set of products over several years, in different geographic regions.

When you publish this report, you want to allow each regional manager to see only the data relevant to his or her region. Rather than creating a separate view for each manager, you can apply a user filter that restricts access to the data based on users’ characteristics, such as their role.

Restricting access to data in this way is referred to as row-level security (RLS). Tableau offers the following approaches to row-level security:

- **Create a user filter and map users to values manually** on page 2544.
  
  This method is convenient but high maintenance, and security can be tentative. It must be done per-workbook, and you must update the filter and republish as your user base changes.

- **Create a dynamic filter using a security field in the data** on page 2547.
  
  Using this method, you create a calculated field that automates the process of mapping users to data values. This method requires that the underlying data include the security information you want to use for filtering.
The most common way to do this is to use a reference (or "look-up") table that contains this information. For example, if you want to filter a view so that only managers can see it, the underlying data must include user names and specify each user’s role.

Because filtering is defined at the data level and automated by the calculated field, this method is more secure than mapping users to data values manually.

Adding user filters to data sources

The two approaches in the previous section describe ways to add filters to data embedded in workbooks. If multiple workbooks connect to the same data, instead of wrangling filters on each workbook, you can filter the data source, and then connect the workbooks to the data source after you publish it.

Workbooks that connect to your filtered data source expose only the data the user signed in to the server is allowed to see. In addition, all connected workbooks show data refreshes as they occur.

Extracts vs. live connections with user filters

The related topic describes how to publish a live connection with user filters. Publishing extracts with user filters has its own complications around row duplication and performance. These are not addressed here.

However, we can direct you a comprehensive discussion about row level security with extracts, on a blog maintained by a Tableau Sales Consultant who has extensive experience with this area. See the following two posts:

- Defusing Row Level Security...Part 1
- Defusing Row Level Security...Part 2

Part 2 contains what our Sales Consultant refers to as the “scalable” approach. This is described in the Advanced Solutions section of Part 2.

Disclaimer: Clicking these links will take you away from Tableau.com. Although we make every effort to ensure links to external websites are accurate and relevant, Tableau cannot take responsibility or provide support for the external content.

For further help with publishing extracts with user filters, we recommend that you work with the Tableau Professional Services team.
Create a User Filter and Secure it for Publishing

Using the example shown in the overview topic *Restrict Access at the Data Row Level* on page 2541, this topic provides the steps for implementing each of the approaches Tableau takes toward row-level security. It includes additional steps for securing the filters for publishing with live connections, whether on data embedded in a published workbook, or in the data source you want to publish separately.

If you haven’t seen the example or read about the Tableau approach to row-level security, take a look at the overview topic before you complete the steps below.

In this article

- **Create a user filter and map users to values manually** below
- **Create a dynamic filter using a security field in the data** on page 2547
- **Filter a data source** on page 2550
- **Secure user filters on published content** on page 2551
- **Copy selected field values from one user to another** on page 2552

Create a user filter and map users to values manually

The steps in this section describe the simplest way to incorporate user-based filtering. This procedure can be sufficient if you have a small and fairly static set of users or groups to manage, and only a small number of workbooks that need user filters. Completing the steps is also a low-cost way to get more familiar with user filtering.

1. In Tableau Desktop, open the workbook, or create a new one, and set up the connection to the data you want to filter.

2. Navigate to the worksheet that you want to apply a filter to.

3. Select **Server > Create User Filter**. Then select the field you want to use for filtering the view. This example uses **Region**.

4. If prompted, sign in to your server or site. For information, see **Sign in to Tableau**
Server or Online on page 310.

5. In the User Filter dialog box, type a name for the set of rules you are creating.
   
   In this example, we’ll use Regional Managers.

6. In the list on the left, select a user or group. On the right, select the individual members of
   the field you selected earlier, that you want the selected users to be able to see.
   
   For this example, selected user Andrew Allen is the manager of the Eastern region, so in
   the field member list, you would select East.

   ![User Filter dialog box]

   Repeat this process for each user or group, and click OK when you’re done mapping
   users to values.

   After you create the user filter, it appears in the Sets area of the Data pane.

7. Drag the user filter to the Filters shelf.
The filter becomes a context filter, and the view adjusts to show data that you are allowed to see.

8. Do any of the following to test or fine-tune the filter:

- If the view appears as a blank canvas, you need to allow yourself or a group you are a member of to see a region.

  In the **Sets** area of the **Data** pane, open the drop-down menu on the user filter, and then select **Edit Set**.

- To preview how the filter works in the published view, in the lower-right corner of the workbook, open the **Filter as User** menu, and select the user or group from the list.
Note: Previewing is not available if the workbook connects to a Tableau Server data source.

- To return to viewing the workbook as yourself, in the top right corner of the Filter as User menu, select Reset.
- To copy the mapping selections you set on one user or group to another (rather than manually mapping the same settings), see Copy selected field values from one user to another on page 2552.

When you publish the workbook, you need to take additional steps to make sure users cannot edit the workbook and remove the filter. For information, see Secure user filters on published content on page 2551.

Create a dynamic filter using a security field in the data

The steps below are based on the following viz, which shows annual sales performance for a list of regional managers.
As described in How user-based filtering works on page 2542, if you want to take this approach, the database must include the field you want to use for filtering.

For this example, the data includes a reference table called People, which contains two columns: Region and Manager. Names in the Manager field match Tableau Server or Online user names, and we’ll use this field for filtering.

You can follow along using the Superstore data that comes with Tableau Desktop, although the fields and values do not match exactly.

Connect to the data and set up the user filter

1. In Tableau Desktop, open the workbook you want to add user filtering to, or create a new one, connecting to the data you want to filter.
   In this example, we use a table called Orders.

2. On the Data Source page, add the reference table, creating a left join. Here we add the People table and create a left join on the Region field.

3. Go to the worksheet, select Analysis > Create Calculated Field, and create the following field:
   - Name: User is a manager
   - Formula: USERNAME() = [Manager]
This new true/false field appears in the Dimensions pane. The formula returns TRUE if the user name of the person signed in to the server exists in the manager column.

4. Add the **User is a manager** field to the **Filters** shelf.

5. In the **Filter** dialog box, select **True**, and then click **OK**.

   This sets the filter so that only people who are managers can see the data in the view.

   If you are not listed in the Manager field, your view might appear as a blank canvas.

6. See how the view looks to a particular person: in the lower-right, open the **Filter as User** menu, and select someone you know is a manager.

   The following image shows how the view shown earlier would look if Andrew Allen were signed in.

   ![Filter as User example](image)

As with a manual user filter, you need to take steps to **Secure user filters on published content** on page 2551.
Filter a data source

Rather than maintain user filters and special permissions on each published workbook, you can filter a data source, and then publish the data source to make it available as a shared, one-to-many resource for anyone who uses that data.

This procedure builds from the dynamic-filter approach described in this topic.

1. Complete the steps in Create a dynamic filter using a security field in the data on page 2547.

2. In the lower-left area of Tableau Desktop, select the Data Source tab.

3. In the upper-right area of the Data Source page, under Filter, click Add.

4. In the Edit Data Source Filters dialog box, click Add, add the calculated field you created for the dynamic filter, and set the filter to True.

   In our examples, this is the User is a manager field.

   Click OK until you get back to the Data Source page.

Global filters and data source filters

When you create a data source filter, any global filters that use that data source are displayed automatically in the Edit Data Source Filters dialog box to make it easy for you to promote a global filter to be a data source filter. To promote the global filter to be a data source filter, click OK.
If you promote a global filter to be a data source filter, that global filter will no longer be visible in worksheets of the workbook (because it becomes a data source filter).

**Important:** Be aware that you do not need to select a global filter in the Edit Data Source Filters dialog box to promote it. When you click OK, all global filters in the list will be promoted.

To prevent a global filter from being promoted to a data source filter, select the global filter in the Edit Data Source Filters dialog box, and then click Remove.

Secure user filters on published content

When you publish a workbook or data source with user filters, you need to set permissions to make sure that users opening your workbook or connecting to your Tableau Server data source cannot remove the filter, thereby gaining access to all of the data.
Notes

- Before publishing, we recommend that you consult with your Tableau administrator regarding any existing policies in your organization, particularly for setting permissions. See also Set Permissions as You Publish a Data Source or Workbook on page 2526.

- This information applies publishing with live connections. Publishing extracts with user filters is not covered here. For a link to a comprehensive resource, see Extracts vs. live connections with user filters on page 2543 in the overview.

Permissions settings

To secure user filters, the following capabilities must be set to Deny, either during publishing, or afterward on the server.

<table>
<thead>
<tr>
<th>Workbooks</th>
<th>Data sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Edit</td>
<td>Save</td>
</tr>
<tr>
<td>Download Workbook/Save As</td>
<td>Download/Save As</td>
</tr>
<tr>
<td>Set Permissions</td>
<td>Set Permissions</td>
</tr>
</tbody>
</table>

Copy selected field values from one user to another

When you’re creating a user filter manually, after you map a user or group to data values (members), you might want to map another user or group in the same way. You can do this by copying and pasting the settings.

1. In the Data pane, under Sets, select a user filter. Click the drop-down arrow and select Edit Set.

2. In the User Filter dialog box, select the user or group on which you want to paste settings from another user or group.
3. Click **Copy From**, and select the user or group whose settings you want to copy.

See also

*Data Security with User Filters*, from the Tableau video learning library.

To watch the video, you need to sign in to Tableau.com (or create a free account).
Use Tableau on the Web

Get started

What can I do with a Tableau site? on the next page
Tour your Tableau Site on page 2557
Creators: Get Started with Web Authoring on page 2566
Viewers: What Can I Do with a Tableau Web View? on page 2603

Explore and manage content

Explore and Analyze Data in a View on page 2595
Use Custom Views on page 2649
Tag Content on page 2620
Manage Web Content on page 2633
Refresh Data or Pause Automatic Updates on page 2649
Work with Content Revisions on page 2642

Author web views

Web Authoring and Tableau Desktop Feature Comparison on page 2733
Creators: Connect to data on the web on page 2580
Creators: Prepare Data on the Web on page 2583
Edit Views on the Web
Connect to published data sources while web editing on page 376
Create a Dashboard on page 2248
Create a Story on page 2325
Save your work

Share and collaborate

Share Web Views on page 2664
Subscribe to Views on page 2665
Send Data-Driven Alerts from Tableau Online or Tableau Server on page 2669
Embed Views and Dashboards on page 2675
Comment on Views on page 2673
If this is the first time you’ve ever seen Tableau Server or Tableau Online and you’re not sure what you can do with it, watch this five-minute video.

To view more training and introductory videos, go to Free Training Videos on the Tableau website.

See all Help

What can I do with a Tableau site?

Your Tableau site is a private workspace on the web where you can share data insights and collaborate with your colleagues. With certain access levels, you can connect to data and create workbooks right on the web, or connect to data sources and workbooks built in Tableau Desktop. Your license level sets what you’re able to do in your Tableau site.

In this article

What is a site? below
How does my site work with Tableau Desktop? on the next page
What can I do on the web? on the next page

What is a site?

A Tableau site is a place for your team to publish data sources and workbooks to share with each other.

In Tableau-speak, we use site to mean a collection of users, groups, and content (workbooks, data sources) that’s walled off from other sites (collections of users, groups, and content).

As a site user, you might be able to sign in to one site, or you might be able to sign into multiple sites, depending on the access your administrator gives you. After you sign in, you can see projects and content that you can access.
How does my site work with Tableau Desktop?

Tableau Desktop and your Tableau site are built to work together. In Tableau Desktop, you can connect to data and build workbooks, stories, and dashboards. When you want to share insights with others, you can publish workbooks and data sources to your Tableau site.

In your Tableau site, you can view, interact with, and explore with those published views and data sources.

What can I do on the web?

Your user level and content permissions determine what you can see and do with Tableau on the web. Starting in 2018.1, your Tableau user level can be Creator, Explorer, or Viewer. For in-depth details on each user level, see the capability grid on Tableau Pricing.

---

Not sure what your user level is? To check, select the icon in the top right corner of the site with your initials or profile picture and select My Account Settings. Site Role tells you if you are a a Creator, Explorer, or Viewer.

---

In general:

**Creators can:**

- Make new connections to data in the browser
- Build and publish data sources and workbooks from the browser or Tableau Desktop
- Use Dashboard Starters
- Everything else that Explorers and Viewers can do

**Explorers\* can:**

- Connect to published data sources on the site to create new workbooks
- Edit and analyze data in published workbooks (Save and Save As options vary based on permissions)
- Create data alerts and custom views, download content (options vary based on permissions)
- Everything that Viewers can do
*The user level Explorer (can publish) is an Explorer who can publish new content from Desktop to a site and create and publish new content from an existing published data source, but cannot connect to external data or create new data sources.

**Viewers can:**

- See published and custom views others have created
- Explore the data in a view using filters and legends, sorting, and tooltips
- Share, comment on, and download content (options vary based on permissions)

For more details, see [What Can I Do with a Tableau Web View?](#)

**Tour your Tableau Site**

**In this article**

- An overview of the web workspace below
- Identify your site on the next page
- Navigate to views on page 2559
- Navigate projects on page 2560
- View and sort content on page 2561
- Interact with views on page 2564

**An overview of the web workspace**

When you sign in to Tableau Server or Tableau Online, the first page you see will look something like the following example. From this page, you can explore the content that is available to you by searching, filtering, and sorting the view.

The pages you can see and the options available to you are based on your site role and content permissions.

The main page includes the **Projects, Workbooks, Views**, or **Data Sources** menus. As you click these options, each page shows related content you can access.
If you have access to multiple sites, you select one when you sign in. You can then navigate to content on other sites by selecting the site from the site menu.

Identify your site

Each site on Tableau Server or Tableau Online has a name and an ID.

On Tableau Server, if only one site exists, that site is named **Default**. When you’re signed in to the Default site, the browser URL looks something like this:
https://server-name/#/projects

After the server name, it includes a hash character #, and the name of the page or workbook you navigated to. In the URL, if you don’t see a /site parameter, you’re signed in to the Default site.

On Tableau Online or a Tableau Server running multiple sites, the browser URL includes #/site/ followed by the site ID. The following URL shows an on-premises Tableau Server site whose site ID is finance:

https://localhost/#/site/finance/views

Navigate to views

To get to a view, do one of the following:

- Click **Projects**, click a workbook, and then click a view thumbnail or name.
- Click **Workbooks**, and then click a view thumbnail or name.
- Click **Views**, and then click a view thumbnail or name.
Navigate projects

Your administrators or project leaders might have created a project hierarchy to organize content you work with. In project hierarchies, projects can contain other (nested) projects.

The way you navigate to sub-projects depends whether you’re using thumbnail or list view. (These two views are defined in View and sort content on the next page below.)

<table>
<thead>
<tr>
<th>Navigate projects in thumbnail view</th>
<th>Navigate projects in list view</th>
</tr>
</thead>
<tbody>
<tr>
<td>On the Projects page, the project’s thumbnail image includes a folder icon that indicates the number of child projects it has. To display the child projects, select the icon.</td>
<td>In the list view, the Projects column tells you how many child projects a parent project has.</td>
</tr>
<tr>
<td><img src="image1.png" alt="Thumbnail view example" /></td>
<td><img src="image2.png" alt="List view example" /></td>
</tr>
<tr>
<td><img src="image1.png" alt="Thumbnail view example" /></td>
<td><img src="image2.png" alt="List view example" /></td>
</tr>
</tbody>
</table>

To open child projects in list view, select the parent project, and then select the Projects tab (the Workbooks page shows by default).

Use the breadcrumbs at the top of the page to navigate back to a parent project.
View and sort content

In the Projects, Workbooks, Views, or Data Sources pages, you can search, view, and sort content. When you select items of interest, you can change permissions, download the content, and perform a variety of other actions. For details, see Perform Actions on Web Content.

View content as lists or thumbnails

Icons at the top of the Projects, Workbooks, and Views pages let you display items as lists or thumbnails.

Here’s list view.

<table>
<thead>
<tr>
<th>Name</th>
<th>Views: All</th>
<th>Sheets</th>
<th>Size</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finance</td>
<td>349</td>
<td>6</td>
<td>3.7 MB</td>
<td>Default</td>
</tr>
<tr>
<td>Local Variety-US</td>
<td>113</td>
<td>5</td>
<td>1.4 MB</td>
<td>Default</td>
</tr>
<tr>
<td>Sales</td>
<td>65</td>
<td>5</td>
<td>1.9 MB</td>
<td>Default</td>
</tr>
<tr>
<td>Superstore</td>
<td>73</td>
<td>7</td>
<td>853.2 KB</td>
<td>Finance</td>
</tr>
</tbody>
</table>

And here’s an example of thumbnail view.
In thumbnail view, hover over workbook and view thumbnails to see sparklines showing popularity over the past 12 weeks.
Sort content

Depending on the type of content, you can sort by characteristics such as name, owner, number of views, creation date, and so on.

To sort items, click the **Sort By** drop-down arrow. This example shows the sorting options for views. (The Least-Most and Most-Least options sort based on relevance.)

You can also sort a list by clicking the column heading. An arrow indicates the order: ascending (up arrow) or descending (down arrow). To sort using multiple options, click the sort type in inverse order. For example, to sort on project first, workbook second, and name last, click those sort types in inverse order (name, workbook, and then project).
Sort data sources by the number of related pageviews or workbooks, or by last extract refresh or modified dates. (Modified dates show when a user last published edits to the data source.) You can also sort based on server alerts, highlighting data sources with extract refresh or latency issues.

To see data source type and authentication details, from the **View** menu, choose **Connections**.

**Interact with views**

When you open a view, the options available to you in the toolbar vary depending on the site configuration and your permissions.
Here's an example view in the browser:

![Example View](image)

And here are some typical view options:

![View Options](image)

When a workbook is published from Tableau Desktop using the **Show Sheets as Tabs** option, you can see each sheet by clicking the tabs at the top of the page:

![View Tabs](image)

After you finish looking at a view, you can navigate back to a **Projects, Workbooks, or Views** page by clicking the breadcrumb links:

![Breadcrumb Links](image)
Creators: Get Started with Web Authoring

There's a lot you can do with Tableau on the web. As a Creator, you can connect to data directly to power your analytics, create stunning data visualizations, and share those insights with others. In this guide, we’ll cover the Tableau Online environment, connecting to and preparing your data for analysis, creating a workbook and sharing those insights with others.

Not sure what your user level is? To check, select the icon in the top right corner of the site with your initials or profile picture and select My Account Settings. Site Role tells you if you are a Creator, Explorer, or Viewer.

In this article

Step 1: Sign in below
Step 2: Create a new workbook and connect to data on the next page
Step 3: Prepare your data on page 2568
Step 4: Author a view on page 2569
Step 5: Present your work on page 2576
Step 6: Save your work on page 2578
Step 7: Collaborate and share your findings on page 2579

Step 1: Sign in

To get started, sign in to your Tableau site and enter your user name and password.

When you log in, all of the content that you have permission to see is in the Content tab, organized by Projects, Workbooks, Views, and Data Sources.
Workbooks are filed into projects in the **Project** tab. Much like folders on your desktop, projects are a way to sort and manage content in your site. If you've been invited to an existing site, you may see projects created by your teammates.

Your site comes with a default folder, and you can use this folder for your explorations if additional projects are not available.

**Step 2: Create a new workbook and connect to data**

In some way, you'll need to connect to data in order to analyze it and build a workbook. As a Creator, Tableau lets you connect to data sources directly on the web, or you can connect to data sources published to Tableau Online through Tableau Desktop.

In the **Project** tab, select a project where you'd like to put your work.

Select **New Workbook**.

The **Connect to Data** window appears.

You can connect to the data that will power your workbook in several ways:

- Upload Excel or text-based data sources from the **Files** tab
- Connect to server or cloud data source from the **Connectors** tab
• Connect to published data sources from the **On This Site** tab

• On Tableau Online, quickly author and analyze data from Oracle Eloqua, Salesforce, ServiceNow ITSM, and Marketo from the **Dashboard Starters** tab

To learn more about connecting to data on the web, see **Creators: Connect to Data on the Web**.

**Step 3: Prepare your data**

Once you’ve connect to data, use the data source page to prepare your data for analysis.

**Parts of the data source page**

There are four areas of the data source page:

A. **Left pane**: Displays details about your data. For file-based data, the left pane might display the file name and the worksheets in the file. For relational-based data, the left pane might display the server, the database or schema, and the tables in the database.

B. **Canvas**: When connected to most relational and file-based data, you can drag one or more tables to the canvas area to set up your Tableau data source for analysis.

C. **Data grid**: You can use the data grid to review the fields and the first 1,000 rows of the data contained in the Tableau data source.
D. **Metadata grid**: The metadata grid button takes you to the metadata grid, which displays the fields in your data source as rows so you can look at its structure.

You can prepare your data for analysis in the data source page by:

- Renaming your data source
- Cleaning your data using the Data Interpreter
- Setting text file options
- Joining your data
- Editing data columns

To learn more about preparing your data for analysis on the web, see [Creators: Prepare your Data on the Web](#).

Once your data is ready for analysis, click on **Sheet 1** at the bottom of the screen to be taken to the Tableau workspace to build a view.

Note: You can save your work at any time. To save a workbook, select **File > Save As**.

**Step 4: Author a view**

You’ve connected to data, you’ve set it up for analysis, and now you’re ready to dive in and create a visual analysis of your data. Tableau calls this a **view**. If you’ve used Tableau Desktop before, the workspace area is very similar.

**Workspace area**

This is the layout of the Tableau workspace:
A. **Workbook name.** A workbook contains sheets. A sheet can be a worksheet, a dashboard, or a story. For more information, see [Workbooks and Sheets](#).

B. **Pages shelf, Filters shelf, Marks card** - Drag fields to the cards and shelves in the workspace to add data to your view. For more information, see [Shelves and Cards](#).

C. **Columns and Rows shelf** - Drag fields to the cards and shelves in the workspace to add data to your view. For more information, see [Shelves and Cards](#).

D. **Toolbar** - Use the toolbar to access commands and analysis and navigation tools. See the [Tableau Toolbar Button reference](#).

E. **View** - This is the workspace where you create your data visualizations.

F. **Side Bar** - In a worksheet, the side bar area contains the Data pane and the Analytics pane.

G. Go to the data source page. For more information, see [Creators: Preparing Data on the Web](#).

H. **Sheet tabs** - Tabs represent each sheet in your workbook. This can include worksheets, dashboards, and stories. For more information, see [Workbooks and Sheets](#).
Different ways to start building a view

Every time you drag a field into the view or onto a shelf, you are asking a question about that data. The question will vary depending on where you drag various fields, the types of field, and the order in which you drag fields into the view.

For every question you ask, the view changes to represent the answer visually - with marks (shapes, text, hierarchies, table structures, axes, color).

In the worksheet, the columns from your data source are shown as fields on the left side in the Data pane. The Data pane contains a variety of different kinds of fields; for now the most important ones are dimensions and measures.

Dimensions typically hold categorical data such as product types and dates, while measures hold numeric data such as sales and profit. (To learn more about see Field Types: Dimensions and Measures, Blue and Green.)

When you build a view, you add fields from the Data pane. You can do this in different ways. For example:
• Drag fields from the **Data** pane and drop them onto the cards and shelves that are part of every Tableau worksheet.

• Double-click one or more fields in the **Data** pane.

• Select one or more fields in the **Data** pane and then choose a chart type from **Show Me**, which identifies the chart types that are appropriate for the fields you selected. For
Filtering data

You might want to filter your data to only display a certain amount in your view. You can filter your data in a number of ways, including:

When you add fields to the filter Shelf, an interactive filter appears in the view. You can then select the items you want to include or exclude in the view.

- To create a filter in Tableau on the web, drag a dimension, measure, or date field to the filter shelf.

When you add a field to the filters shelf, an interactive filter appears in the view. You can select items you want to include or exclude from the view, or select a range of values.
You can also interact with a published view by selecting a single mark (data point) or click and drag in the view to select several marks. On the tooltip that appears, you can select Keep Only to keep only the selected marks in the view, or select Exclude to remove the selected marks from the view.

For a better understanding of filtering in Tableau, including Tableau's order of operations, see Filter Data from Your Views.

Using the Marks card

You can use the Marks card to add context and details to the marks in your view. You use the Marks card to set the mark type, and to encode your data with color, size, shape, detail, and text.
After you add a field to the Marks card, you can click the icon next to the field to change the property it is using. You can also click the property buttons in the Marks card to change those settings.

Many properties can have multiple fields. For example, you can add multiple fields to Label, Detail, Tooltip, and Color. Size and Shape can only have one field at a time. For more details, see Mark Properties Reference.
Undo your work

Tableau is extremely flexible and very forgiving. As you build a view, if you ever take a path that isn't answering your question, you can always undo to a previous point in your exploration.

- To undo or redo, click undo ← or redo → on the toolbar.

You can undo all the way back to the last time you opened the workbook. You can undo or redo an unlimited number of times.

Build and edit views on the web

There are many ways to explore data in Tableau. For instructions on building specific chart types on the web, see Edit Tableau Views on the Web on page 2586, or for more features available to help you build a visualization in Tableau, see Build and Explore Data Views.

For a 6-minute walkthrough on building data views, see the Getting Started with Visual Analytics free training video. Use your tableau.com account to sign in.

Step 5: Present your work

There are several ways to tell a story or persuade others with the data insights you've found in your views.

Format your work

You can format almost everything you see on a workbook, including parts of a view, your workbook’s fonts, and graph lines.

- Learn about visual best practices
- Format parts of a view
- Customize colors and fonts
- Format at the workbook level

Dashboards

A dashboard is a collection of several worksheets and supporting information shown in a single space so you can compare and monitor different data simultaneously.
To open a new dashboard sheet and start creating a dashboard, click the New Dashboard icon at the bottom of the workbook:

The Dashboard area appears on the left and lists the sheets in your workbook. You can add one or more views to a dashboard, add objects like web pages, blank space, and layout containers, or interactivity to associate different views on your sheet.

- Learn more about dashboards
- Watch a 6-minute walkthrough on building a dashboard
- See stunning dashboard examples in Tableau Public
- Learn about tips and tricks from dashboard experts

Stories

A *story* is a sheet than contains a sequence of worksheets and dashboards that work together to convey information. You can use stories to make a compelling case by showing how facts are connected, and how decisions create stories to tell a data narrative, provide context, demonstrate how decisions relate to outcomes, or make a compelling case with your data.
Create a story or learn more about stories.

Step 6: Save your work

You can save your workbook at any time by selecting **File > Save As**. Your workbook will save into the folder it was created.

You can also save your data source to create another workbook later, or enable members of your team to use that data for their own analysis.

To save a data source, hover over the data source name in the Data tab in the workspace until an arrow appears. Select the arrow and select **Save**.
When you save a data source on the web, it will appear as a published data source in the **Connect to Data** window as well as in the **Data Sources** section.

When you're finished, select **File > Close** to exit the workspace.

**Step 7: Collaborate and share your findings**

There are a number of ways to share your work and stay up-to-date.

**Collaborate**

Other Creators or Explorers in your organization can edit or update your workbooks and views if they have been given access to your projects. Access is controlled by your site's administrator, who can set project permissions, move content between projects, and change a user's role. If you've saved your data source, those users can connect to your data and use it to create new workbooks.

**Share Views and Insights**

Every view or workbook saved to your site can be shared via links in email and other applications, or can be embedded directly into web pages, wikis, and web applications.

To see a view, users must have permission to access it in Tableau Server or Tableau Online.

Learn how to:

- **Share web views** (link to a view or embed a view)
- **Subscribe to Views** to receive a snapshot of it by email at regular intervals
- Create **Data-Driven Alerts** to receive email notifications when data reaches important
thresholds.

- **Comment on views** to share a conversation with other users.
- **Embed views and dashboards** into webpages
- **Download views or workbooks**

**Creators: Connect to data on the web**

Before you can create a new workbook and build a view on the web to analyze your data on the web, you need to connect to your data. Tableau supports connecting to data sources on the web published through Tableau Desktop, or, connecting to data directly through Tableau Online or Tableau Server.

**Note:** Data connections created in Tableau Online or Tableau Server are **live connections** only. If you need to use an extract for web authoring, you can publish your data source through Tableau Desktop. To publish through Tableau Desktop, see **Publish Data Sources and Workbooks**.

To create a new workbook, sign into Tableau Online or Tableau Server.

In the Content tab, do either of the following:
Go to the Projects section, select a project, and select New Workbook.

Go to the Workbooks section and select New Workbook.

A Connect to Data window opens, which contains several tabs: Files, Connectors, On this Site, and, if you’re connected to Tableau Online, Dashboard Starters.

As a Creator, you can create a new workbook or add a new data source to an existing workbook in several ways:

- Upload Excel or text-based data from the Files tab
- Connect to server or cloud data with Connectors
- Connect to published data sources with On This Site
- On Tableau Online, quickly author and analyze data from Oracle Eloqua, Salesforce, ServiceNow ITSM, and QuickBooks Online from the Dashboard Starters tab

**Connect to Files**

Tableau supports uploading Excel or text-based data sources (.xlx, .tsv, .csv) directly in your browser. In the Files tab of the Connect to Data window, connect to an Excel or text file by dragging and dropping it into the field or clicking "Upload from computer."

When Tableau successfully connects to your data, the Data Source page opens so that you can prepare the data for analysis and begin building your view. To learn more, see Creators: Prepare Data on the Web.

**Use connectors**

From the Connectors tab, you can connect to data housed in a cloud database or on a server in your enterprise. You need to supply connection information for each data connection that you make. For example, for most data connections, you need to supply a server name and your sign-in information.

**Connect to Your Data** has information on how to connect Tableau to each of these connector types to set up your data source. If your connector doesn't appear in the Connectors tab, you can connect to data through Tableau Desktop and publish your data source to Tableau Online or Tableau Server for web authoring. Learn more about how to Publish a Data Source in Tableau Desktop.

Tableau web authoring supports connecting to your data through Tableau Online and Tableau Server using the following connectors:
If your site is in Tableau Server, you may see more connectors than are on this list.

When Tableau successfully connects to your data, the Data Source page opens so that you can prepare the data for analysis and begin building your view. To learn more, see Creators: Prepare Data on the Web.

**Note:** If you’re unable to connect to your data from Tableau Online, check to see if the database is publicly accessible. Tableau Online can only connect to data that’s accessible from the public internet. If your data is behind a private network, you can connect using Tableau Bridge. To learn more, see Publishers: Use Tableau Bridge to Keep Tableau Online Data Fresh.

**Connect to published data sources "On this Site"**

Tableau supports connecting to published data saved to your site with the **On this Site** tab. If you or another user has saved a data source or published to the web through Tableau Desktop, it will appear in Tableau Online or Tableau Server as a published data source.

When Tableau successfully connects to your data, the Data Source page opens so that you can prepare the data for analysis and begin building your view. To learn more, see Creators: Prepare Data on the Web.

**Keeping data fresh in web authoring**

**Update uploaded files in Tableau Online or Tableau Server:** If you manually upload a file (Excel or text) for web authoring, Tableau can’t refresh the file automatically. To update your data, select “Edit Connection” to upload a new version of the file.

**Update file-based published data sources in Tableau Online:** If you have a published data source in Tableau Online (published through Tableau Desktop) that uses file-based data,
you can keep it fresh using Tableau Bridge. For more information, see Expand Data Freshness Options by Using Tableau Bridge.

Dashboard Starters

Dashboard Starters help you quickly author and analyze data from cloud-based systems like Salesforce, ServiceNow ITSM, Oracle Eloqua, and Marketo. Simply create a new workbook and choose from several beautiful, informative designs that are tailor-made for key business metrics. For more information, see Dashboard Starters for Cloud-based Data Sources.

Creators: Prepare Data on the Web

Once you’ve connect to data in Tableau Online or Tableau Server, you can use the Data Source page to set up your data source and prepare your data for analysis. Having data that is formatted in a way Tableau likes is crucial to building a view or doing analysis in Tableau, and communicating the right information to the right people.

This article describes preparing your data in the browser with Tableau Online or Tableau Server. Although Tableau supports many of the same functionality for preparing your data for analysis in the browser as Tableau Desktop, for specific differences between the two, see Web Authoring and Tableau Desktop Feature Comparison on page 2733.

Note on keeping your data fresh: If you connect to a published flat file (Excel or text), that data will not be refreshed even if it’s modified. If your data is in an on-premises server and is published to the web through Tableau Desktop, it will be rendered as an extract and won’t be refreshed. If you need to keep data published through Tableau Desktop fresh on the web, you can use Tableau Bridge. To learn more about these requirements, see Keep Data Fresh and Publishers: Use Tableau Bridge to Keep Tableau Online Data Fresh on page 2535.

In this article

Tour the Data Source page on the next page
Set up data sources on page 2585
Creators: Prepare Data on the Web above
Tour the Data Source page

There are four parts of the data source page:

A. Left pane

Displays details about your data connection. For example, for Excel-based data, the left pane shows your Excel workbook name, and sheets contained in that workbook. For databases, the left pane displays the server, the database or schema, and tables in the database.

B. Canvas

When connected to most relational and file-based data, you can drag one or more tables to the canvas area to join tables for analysis.

In the canvas, you can also join tables from the same or multiple data sources
C. Data grid

You can use the data grid to review or rename fields, create calculations, preview your changes, and more.

D. Metadata grid

Select the metadata grid button to see the fields in your data source as rows. This view is helpful in understanding the structure of your data source.

Set up data sources

After you connect to your data, use the Data Source page to set up the data source and prepare your data for analysis. There are many optional configurations that you can make before you begin your analysis. The configurations that you make on the Data Source page creates the data source that Tableau uses to interpret and interact with your data.

Join Your Data on page 657
Union Your Data on page 708
Convert a Field to a Date Field on page 740
Pivot Data from Columns to Rows on page 747
Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter on page 752

Edit and sort fields in the data grid

You can make changes to your fields in the data grid. You can rename your fields, group them, create aliases, create calculated fields, change data type, and convert to a measure.

For more information:

- Understand Field Type Detection and Naming Improvements
- Get Started with Calculations in Tableau
- Convert a Field to a Date Field
- Data Types
- Field types: Dimensions and Measures, Blue and Green
Manage data sources

Edit Data Sources on page 817
Refresh Data Sources on page 829
Replace Data Sources on page 830
Close Data Sources on page 841
Save Data Sources on page 832

Edit Tableau Views on the Web

If you can see the Edit button when you are looking at a Tableau view, that means you can make changes to it. Depending on your access level and permissions, you can:

- Edit an existing published workbook and add worksheets for views, dashboards, and stories.
- Create and edit a new workbook based on a published data source.
- Edit an existing published workbook and add worksheets, and also connect to different published data sources while editing. For details, see Connect to Published Data Sources while Web Editing.

In this article

Create a new workbook on the web below
Edit a published view on the next page
Build common chart types on the web on page 2588

Create a new workbook on the web

1. Sign into a site on Tableau Online or Tableau Server.
2. Select Data Sources on the Content menu.
3. In the data sources list, select the check box next to the data source you want to use, and
then click **Actions > New Workbook**.

A new, blank workbook opens in the Tableau Server web editing environment.

**Edit a published view**

1. Sign into your site, and then select **Views** on the Content menu.
2. Open the view.

![View](image)

**Product**

4 views  
🌟 0

3. Click **Edit** in the view toolbar.

![View toolbar](image)

When you click **Edit**, the view opens in web authoring mode.
Now you can edit the view.

To get started, see **Build common chart types on the web** below in this article. Also see Step 4 in Creators: Get Started with Web Authoring and Explore and Analyze Data in a View.

4. The **Save** options available to you will vary depending on your permissions set by your Tableau site administrator.

   - To close the view without saving your changes, select **File > Close**.
   - To save your work in the current workbook, select **File > Save**.
   - To save your work in a new workbook, select **File > Save As**.

**Build common chart types on the web**

This section describes how to build three different types of charts in views. These examples use sales data from the **Sample-Superstore** data source available in Tableau Desktop.

- Build a bar chart on the next page
- Build a highlight table on page 2591
- Build a map view on page 2592
Build a bar chart

This procedure shows how to build a bar chart view that displays a list of technology products and how much profit each has generated.

1. Under **Dimensions** on the **Data** pane, drag **Category** then **Sub-Category** to the **Columns** shelf. From **Measures**, drag **Profit** to the **Rows** shelf.

   ![Diagram of Tableau interface](image)

   Click the image to replay it.

   Tableau now has enough to convert the data into a visualization (view), in this case, a horizontal bar chart.

2. To display only the items under Technology, click the Technology header and select **Keep Only**.
3. To display each bar as a separate color, drag **Sub-Category** to the **Color** card. As a final touch, click the **Color** card to edit the color palette and assign the **Blue** palette.
4. Right-click the sheet tab, click **Rename Sheet**, type **Bar Chart**, and then click **OK**.

**Build a highlight table**

This procedure shows how to create a table that lists product profits by category and uses color to highlight the range of sales sizes.

1. Create a new sheet by clicking the New worksheet icon.

2. From **Dimensions**, drag **Segment** to the **Columns** shelf and **Category** and **Sub-Category** to the **Rows** shelf, then drag **Profit** to the **Text** card.

3. Add color to your table by dragging **Profit** to the **Color** card and changing the **Marks** type to square.
Your view now displays data about profits in a table that uses color to highlight the different sales amounts.

4. Give the view a name by right-clicking the sheet tab, clicking Rename Sheet, and typing Highlight Table. Click OK.

Build a map view

This procedure shows how to create a map view from the Sample -Superstore data source. This view will show profits by U.S. state.

1. Create a new sheet by clicking the New worksheet icon.

2. On the Data pane, double-click the State dimension, then under Measures, double-click Profit.

3. On the Show Me menu, click Maps.
Tableau looks for geographic fields in any data source. For more information on how Tableau interprets geographic data in data sources, see Assign Geographic Roles in the Tableau Help.

4. Give the view a name by right-clicking the sheet tab, clicking Rename Sheet, typing Map. Click OK.
Explore and Manage Web Content
Explore and Analyze Data in a View

When you are ready to explore and analyze your view, Tableau offers a selection of dynamic data inspection tools that help you isolate the data of interest. For example, if you have a dense data view, you can focus on a particular region, select a group of outliers, and view the underlying data source rows for each mark.

Watch a Video: To see related concepts demonstrated in Tableau, watch Getting Started with Visual Analytics, a 6-minute free training video. Use your tableau.com account to sign in.

In this article

Select marks to highlight data points in the view
Analyze selected marks using tooltips
Highlight data by category in tooltips
Explore data in maps
Add analytics objects to the view
Compare marks data with recalculated lines

Select marks to highlight data points in the view

The simplest way to add highlighting to a view is to select the marks you want to highlight. When you manually select a mark in the view, all other marks are dimmed to draw attention to the selection. Your selection is saved with the workbook.

For more information on how to use selection tools, see Select Marks on page 2612. Also see Legend Highlighting on page 1797, Highlight Data Points in Context on page 1800, Highlight Toolbar Button on page 1803, and Highlight Actions on page 1794.

You can select multiple marks by holding down the Ctrl key on your keyboard (Command key on Mac) while you select each mark. You can also use one of the advanced selection tools to select marks within a specific shape.
You can also select marks and headers to filter what is being displayed in the view. For more details, see Select to keep or exclude data points in your view on page 1163 and Select headers to filter data on page 1164.

Analyze selected marks using tooltips

When you select one or multiple marks in a view, and then continue to hover in the same spot, several options for inspecting your data are available in the tooltip.

**Tableau Desktop version**

**Web version**
Tooltips provide additional information about the marks in your view, and provide the following data analysis options:

- **Keep only the selected marks in the view.** Keep Only
  
  For more information, see Select to keep or exclude data points in your view on page 1163.

- **Exclude the selected marks from the view.** Exclude
  
  For more information, see Select to keep or exclude data points in your view on page 1163.

- **Create a group based on the selected marks.** ▪ For more information, see Correct Data Errors or Combine Dimension Members by Grouping Your Data on page 1002.

- **Create a set that contains the selected marks (Tableau Desktop only).** • For more information, see Create Sets on page 1004.

- **View the underlying data of the marks selected.** □ For more information, see View Underlying Data on page 2616.

For related information on Tooltips, see **Tooltips** on page 214.

For details on related analysis tools, see **View Toolbar** on page 2610, **Select Marks** on page 2612, **Pan and Zoom** on page 2615, **Undo and Redo** on page 2616, and **Drop Lines** on page 1678.

---

**Highlight data by category in tooltips**

If you include discrete (categorical) dimensions or measures in your tooltip, you can use them to select marks in the view that have the same values. If Tooltip selection is turned on for your worksheet, then these fields show as underlined text when you mouse over them. Clicking on the tooltip selects all the marks in the view that have the same value.
In the example below, when you click on Bill Shonely (the value for Customer Name) in the tooltip, any mark in the view that also includes Bill Shonely as a customer is selected and all other marks are dimmed.

To turn this functionality off or on, select the **Allow selection by category** check box in the Edit Tooltips dialog box. For more information about setting tooltip properties, see [Add tooltips to marks](#) on page 1133.

---

**Explore data in maps**

Tableau maps can help you quickly find locations and analyze data worldwide. There are many ways you can explore and interact with map views. You can zoom in and out, pan, and select marks with the view toolbar, and even search for locations worldwide with map search.
For more details, see **Explore Data in Maps** on page 2086, **Search for Locations in Your Map** on page 2088, and **Measure Distances Between Data Points and Locations in a Map** on page 2089.

---

Add analytics objects to the view (trend lines, forecasts, reference lines and bands)

Drag reference lines, box plots, trend lines forecasts, and other items into your view from the **Analytics** pane, which appears on the left side of the workspace. Toggle between the **Data** pane and the **Analytics** pane by clicking the tabs at the top of the side bar. For more details, see **Apply Advanced Analysis to a View (Analytics Pane)** on page 174.
### Tableau Desktop Analytics pane

In Tableau Desktop, options for adding Analytics objects to the view are available in the **Analytics** pane or menu, or in context in the view. For example, reference lines and bands are available when you edit an axis, and trend lines and forecasts are available from the Analysis menu.

The **Analytics** pane provides drag-and-drop access for the various options.

On the web, most Analytics objects are available from the Analytics pane.

---

**Compare marks data with recalculated lines**

If there is an analytics object in your view, such as an average line, constant line, trend line, reference line or distribution line or band, selecting one or multiple marks in the view lets you instantly compare the analytical data for the selected marks to all data in the view.
For example, selecting marks in a view that contains a trend line creates a second, recalculated trend line, the value of which is determined by the selected marks only, so you can compare that trend to the overall trend.

Recalculated lines are displayed by default when you select marks in a view that already contains analytics objects. In Tableau Desktop, if you don't want recalculated lines to be created you can turn them off.

**Turn off recalculated lines (Tableau Desktop only):**

1. Select an analytics object in the view, such as a trend line, and click **Edit**.
2. In the **Edit** dialog box, clear **Show recalculated line for highlighted or selected data points**.
Alternatively, you can right-click (control-click on Mac) an analytics object in the view and clear **Show Recalculated Line**.

When you return to the view and select or highlight marks, recalculated lines will not appear. For more information about how to highlight marks, see **Select Marks to Highlight** on page 1796.

**Recalculated lines and highlight actions**

Recalculated lines also work with highlighting actions, both in worksheets and dashboards.

For example, if you create a highlight action in a dashboard, selecting marks in one sheet will highlight marks in the other sheets in the dashboard. If those other sheets have trend lines, reference lines, or other analytics objects, recalculated lines will appear as the views update.
What Can I Do with a Tableau Web View?

You’ve seen hundreds of presentations, emails, and documents with data charts. Those things are static and sad. First off, the data is frozen in time. Second, if you have any questions about it, you’ve got to ask whoever created the chart . . . if you can find them.

Tableau is different. When someone sends you a link to a web view, you’re often seeing the freshest data available, and more importantly, you can interact with it. Want to dig deeper and answer questions the minute they occur to you? Simply click the view right in your web browser; it’s that simple.

What can you do with a Tableau web view? A lot!

In this article

Will I hurt the data? on the next page
Filter and sort data on the next page
See details about specific data points on page 2605
Tip: We’re covering the most popular ways you can interact with views. Sometimes, view creators enable a subset of these features.

Will I hurt the data?

Not at all. When you interact with a view, you’re simply changing how it looks for a moment. Next time you or your colleagues open the view, it will look exactly as it originally appeared. The underlying data always remains safe and sound, so feel free to explore!

Filter and sort data

Want to trim or rearrange the visible data so you can focus on stuff you’re interested in? You’ve got a couple easy options:

Filter data

Most view creators add filters that let you limit visible data to specific date ranges, regions, and categories. When you see these, feel free to click or drag them as needed.

Click the image above to replay the animation.
Sort data

If you’re looking at a table of data and want to sort it alphabetically or numerically, just hover over a column header and click the sort icon.

See details about specific data points

As you move the mouse across a view, you’ll often see tooltips that reveal details about each data point, or *mark*. Marks can take many forms; here’s how they look in maps, bar charts, and tables:
View underlying data

If you’re the inquisitive type, you might want to see summarized source data to get a better sense of the numbers behind the visuals. If the view creator has given you permission, you can click any mark in the view, and then click the View Data icon.
Collaborate on data

Data becomes more meaningful when you explore it with other people.

Add comments

To quickly ask questions or share insights, click **Comment** in the toolbar. You can pair your remarks with a snapshot of the view that reflects any filters or other changes you made.

Share links

Want to share the excitement of directly engaging with data? Click **Share** in the toolbar, copy the link to the view, and then email or instant-message it to anyone with a Tableau account at your organization.
Get views emailed to you on a schedule

To automatically receive emails of a view on a regular schedule (before weekly meetings, for example), click **Subscribe** in the toolbar.

**Tip:** You might also receive emails to views that other people have arranged for you, including data-driven alerts that notify users when data crosses key thresholds for your business.

Save favorite views

As the Tableau content in your organization grows, you'll want to easily find data important to you. To save a favorite view, click the star icon above it.
To quickly return to a favorite, click the star icon in the upper-right corner of the browser. Favorites you choose in Tableau Server or Tableau Online also appear in the Tableau Mobile app, and vice versa, giving you instant access to key data wherever you go.

Download a view

You’ve seen the power of interactive Tableau web views. But there are times when you want to show off that data outside the browser. If the view creator has given you permission, click **Download** in the toolbar to create a static .png or .pdf file that reflects the view as it looks right now.

Congratulations, data rockstar!

See how easy that was? Now that you’re familiar with web views, start exploring all the Tableau content available at your company. Answer questions. Share insights. You’ve got this!
See also

Tour your Tableau Site on page 2557
Share and Collaborate on the Web on page 2664
Select Marks on page 2612

View Toolbar

The view toolbar appears in the upper-left corner of the view and lets you select marks, zoom in and out, and pan in the view. By default, the view toolbar appears when you hover over a map view, but you can also show the view toolbar in other types of views, such as scatter plots or box plots.

For more information about showing and hiding the view toolbar in map views, see Customize How People Interact with your Map on page 2069.

Tableau Desktop version

Web version

Show the view toolbar

To show the View Toolbar use one of the following options:

Tableau Desktop

- In a worksheet or dashboard, right-click (control-click on Mac) anywhere in the view, and then select Show View Toolbar.

Tableau Server or Tableau Online

- From a worksheet, from the top menu, select Worksheet > Show View Toolbar.
- From a dashboard, select the zone where you want to show the View Toolbar and do one of the following:
From the top menu, select Worksheet > Show View Toolbar.

Click the drop-down arrow for the selected zone and select View Toolbar from the context menu.

Set when the view toolbar appears in the view

- Select Worksheet > Show View Toolbar (in a dashboard, select a view, then select Worksheet > Show View Toolbar), and then select one of the following options:
  - Automatic – The toolbar appears only when you hover over a map view.
  - Show on hover – The toolbar appears when you hover over the selected view. You can select this option for any type of view.
  - Hide – The toolbar does not appear in the selected view.

Note: To improve usability on mobile devices, the view toolbar always appears when interacting with scrollable views or multiple selections.

Hide the view toolbar

To hide the View Toolbar use one of the following options:

**Tableau Desktop**

- In a worksheet or dashboard, right-click (control-click on Mac) anywhere in the view, and then select Hide View Toolbar.

**Tableau Server or Tableau Online**

- For a worksheet or dashboard, from the top menu, select Worksheet > Show View Toolbar > Hide.

- From a dashboard, select the zone where you want to show the View Toolbar, and then right-click (control-click on Mac), click the drop-down arrow and then select View Toolbar > Hide from the context menu.

Note: If the view toolbar is hidden, you can still use keyboard shortcuts to select marks, zoom, and pan. For more information, see Keyboard Shortcuts on page 2761.
Select Marks

You can select marks in the view to inspect your data. When you select a mark or a subset of marks in the view, you can see information about the marks in the tooltip that appears. You can also quickly filter the marks you select from the view, as well as view their underlying data. For more information, see **Tooltips on page 214**.

Select marks

Click an individual mark to select it. After you select a mark, hold down the Ctrl key (Command key on Mac) to add more marks to a selection.

To select multiple marks using the default selection tool, click and drag across the view. In most views, the Rectangular selection tool is the default tool.

You can also use the Radial, Rectangular, and Lasso tools on the view toolbar to select multiple marks. For more information about the view toolbar, see **View Toolbar on page 2610**.

**Note:** In Tableau Server and Tableau Online, the View Toolbar is only available for maps.

Radial selection tool

The Radial tool selects marks within a circular area. To use the Radial tool, hover over the arrow on the view toolbar, click the Radial tool button, and then click and drag across the view.
You can also measure distance in a map view with the Radial tool. For more information, see Measure Distances Between Data Points and Locations in a Map on page 2089.

Rectangular selection tool

The Rectangular tool selects marks within a rectangular shape. To use the Rectangular tool, hover over the arrow on the view toolbar, click the Rectangular tool button, and then click and drag across the view.
Note: The Rectangular tool is the default tool in most views, and may not appear in the view toolbar. In this case, you can drag across the view to use the rectangular tool.

Lasso selection tool

With the Lasso tool, you can select multiple marks by drawing a freehand shape around them. This tool is useful when you want to include only certain marks, and exclude others around them.

To use the Lasso tool, hover over the arrow on the view toolbar, click the Lasso tool button , and then draw a freehand shape around the marks you want to select.
Pan and Zoom

The pan tool and zoom controls help you interact with the view and inspect your data. They are located in the upper-left corner of the view, on the view toolbar. For more information, see View Toolbar on page 2610.

You can use the zoom controls to zoom in and out, zoom to a specific area, and fix or reset the axes in the view. Use the pan tool to move quickly around the view.

Zoom in and out

On the view toolbar, click the Zoom In button \[ + \] to zoom in and the Zoom Out button \[ - \] to zoom out. If the view toolbar is hidden, double-click the view to zoom in; to zoom out, hold down Shift, and then double-click the view.

Zoom to a specific area

To zoom in to a specific area of the view, click the Zoom Area tool button \[ \] on the view toolbar, and then drag to create the zoom area. If the view toolbar is hidden, hold down Ctrl +
Shift (Command-Shift on Mac) to use the Zoom Area tool.

**Reset the view**

After you zoom in or out, the axes in the view are fixed to a specific range. To quickly reset the axes so they automatically zoom the view to all of your data, do one of the following:

- In Tableau Desktop, click the Reset Axes button on the view toolbar.

- In Tableau Server or Tableau Online, click the Zoom Home button on the view toolbar.

**Pan**

To pan, do one of the following:

- Hold down Shift, and then drag across the view.

- On the view toolbar, hover over the arrow, select the Pan tool, and then click and drag across the view.

**Undo and Redo**

You can perform unlimited undo and redo of your actions. You can undo almost all actions in Tableau by pressing the **Undo** button on the toolbar. Likewise, you can redo almost all actions by pressing the **Redo** button on the toolbar.

In this regard, every workbook behaves like a web browser. You can quickly return to a previous view. Or you can browse all the views of a data source that you have created. Tableau saves the undo/redo history across all worksheets until you exit. The history is not saved between sessions.

**View Underlying Data**

The View Data command lets you display the values for all rows in the data source that underlie a set of marks in the view. It also shows you the summary data based on the aggregations in
the view. You can view data to verify the aggregated value associated with a mark, or to isolate and export the individual rows associated with data of interest, such as outliers.

You can view data for a selection of marks, for the fields in the Data pane, and when you’re connecting to data.

The View Data command works with all relational and multi-dimensional databases except Oracle Essbase and SAP® Business Information Warehouse databases. While you can view data with the Microsoft Analysis Services and the Teradata OLAP connector multi-dimensional databases, the database must be drill-through enabled; in addition there are some restrictions to the data you can see. Multi-dimensional data sources are supported only in Tableau Desktop on Windows.

In the view below, sales for two product dimensions (Category and Sub-Category) are displayed as a bar chart. Suppose you wanted to view data for the largest marks in each pane. To do this, you would do one of the following:

- In Tableau Desktop, select the marks, right-click (control-click on Mac) in the view, and select View Data on the context menu. Alternatively, you could select the Analysis > View Data menu item.

- In Tableau Server or Tableau Online, select the marks and click View Data on the Tooltip menu.

**Tableau Desktop version**

**Web version**
Viewing data may not return any records if you are using a field that contains floating point values as a dimension. This is due to the precision of the data source and mainly occurs when you are connected to Microsoft Excel, Microsoft Access, or text files.

Summary Data

Summarized data is shown on the **Summary** tab. Summarized data is a text table of the aggregated data for the fields shown in the view.

**Tableau Desktop version**

![Summary view in Tableau Desktop version](image1.png)

**Web version**

![Summary view in Web version](image2.png)

Full Data

All data for the selected marks are displayed on the **Full Data** tab. In the lower right of the dialog box you can see the number of rows in the underlying data.

If you’re using data blending, the **Full Data** tab shows only the data from the primary data source.

**Tableau Desktop version**

![Full data view in Tableau Desktop version](image3.png)

**Web version**

![Full data view in Web version](image4.png)
In Tableau Desktop

Sort the data by clicking one or more column headers. To restore the original sort order, click the header repeatedly until it is no longer highlighted with a sort arrow.

By default, Show all fields is selected. Clear this option to only show the columns used on shelves (or fields referenced by a calculation used on a shelf) in the current worksheet.

If you want to export one or more data source rows, select the data points of interest by selecting the row and then clicking Copy to copy the selected data.

In Tableau Server or Tableau Online

Click the Download all rows as a text file link to download all rows of data and open them in .csv file. If you want to see all fields related to the selected marks, select the Show all columns check box and then click the link to download the rows of data.
View Data (Microsoft Analysis Services)

In Tableau Desktop, view Data with a Microsoft Analysis Services database works almost the same way it does with relational data sources. The difference is that a Microsoft Analysis Services cube is generally set up and configured by an administrator who decides whether it is enabled for drill-through and the fields that a user is allowed to see. That means that when you try to view data using a database that is not enabled, you may get an error message alerting you that the cube is not enabled for drill-through.

In addition, Microsoft Analysis Services databases limit viewing data to a single mark at a time. More precisely, viewing the data (which uses MDX drill-through) is not an option when the selection of mark(s) is defined by more than one value of a dimension.

When you are viewing underlying data for a field, the Show all fields option is checked and disabled by default. With a Microsoft Analysis Services database, only the fields specified by the administrator are shown, so you cannot choose to include all data source fields in the dialog box.

Tag Content

Tags are keywords you can create for workbooks, views, and data sources to help you find, filter, and categorize content in Tableau Server and Tableau Online. Authors can add tags to content when they publish content to a Tableau server. But you can also add tags to any workbook, view, or data source that you are allowed to access, and you can delete any tags you have added.

View tags

In a published workbook or view, its tags appear in the tooltip when you hover over the thumbnail. You can also view tags in the Tags area located below a view.

Click a tag link to filter the view to any content that uses that tag.
Find tagged content

Use tags to find content with quick search and filtered search.
Add tags to a workbook, view, or data source

You can add tags to any content on the server that you have permission to access.

1. Navigate to a list of workbooks, views, or data sources.
2. Select one or more items you want to tag, and then select **Actions > Tag**.

Enter one or more words in the text box, and then click **Save Tags**.
Delete tags
You can delete any tags that you have created.

1. From the pop-up menu on specific content, select Tag.

2. Click the 'x' next to the tag you want to remove.

Search Content
Tableau Server and Tableau Online let you search content in a variety of ways. The quick search field at the top of the page searches the entire site for matching resources of any kind. Filtered search finds matches using search criteria that are specific to each resource type, such as modified date for workbooks, and connection type for data sources. You can also find workbooks or views that you've marked as favorites.
Both quick search and filtered search support attributes and operators that help you set the scope of the search. For example, you can limit a search to just titles of views, or use operators like and and not.

**Tip:** Search is not case sensitive.

In this article

- [Quick search](#)
- [Filtered search](#)
- [Search attributes](#)
- [Search operators](#)
- [Spaces and search](#)

**Quick search**

Use the quick search field at the top of the page to find resources anywhere in the site.

Quick search looks for matching text in resource names, owners, tags, captions, comments, and other information across the site. As you type, the top items that match your search text appear, ordered by relevance. Relevance is based on number of page views, recent activity, and your favorites. For data sources, the number of connected workbooks and views is also considered.

In the search results, you’ll see the total number of page views and favorites for a resource. Hover over an item to see a tooltip that tells you more about it.
Note: Quick search restricts searches to published data sources only. To search on embedded data sources (workbooks that use a data connection only), use a filtered search.

Filtered search

With filtered search, you can use a combination of search text and filters to find your content. Each content page has different filters available that are relevant to the resource type. When you enter a search term and select General Filters, those search options apply to every content type. For example, you could type "sales" and you could get relevant results for that term in the Projects, Workbooks, Views, and Data Sources pages.

1. Go the page for the resource type you want to search and filter. For example, if you want to search for data connection information, click the Data Sources menu, and then use
the Data Source Filters section of the Filters pane.

2. Enter text in the main search field, and then press the Enter key.

3. Click a filter, and enter a search term for filter value. Select a filter value, or select from other available filter options.

![Filters pane with search fields and filters]

To remove a filter, click Clear at the top of the Filters pane.

![Clear button in Filters pane]

To remove all filters, click Clear All Filters at the top of the Filters pane.
Click the Filters button to collapse or open the Filters pane and reset the filters to their defaults.

The toggle is a dark icon (▲) when the Filters pane is open and a light icon (▼) when the Filters pane is closed.

Search attributes

In addition to a general search, you can limit search on Tableau Server and Tableau Online to a specific attribute such as name, workbook, data source, and so on.

To include a search attribute, use the following syntax in the search box:

<attribute-name>:search-text (with no spaces on either side of the colon).

For example, to return only items whose names contain the words sales or projections, type the following:

name: sales projections.

You can include multiple attributes to further limit a search. For example, to find all dashboards that are owned by Smith, type the following:

sheettype: dashboard owner: smith

A complete list of attributes is shown below.

<table>
<thead>
<tr>
<th>This attribute...</th>
<th>Followed by...</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>name:</td>
<td>search term</td>
<td>Items with names that match the search term</td>
</tr>
<tr>
<td>title:</td>
<td>search term</td>
<td>Views with titles that match the search term</td>
</tr>
<tr>
<td>caption:</td>
<td>search term</td>
<td>Applies to views with captions.</td>
</tr>
<tr>
<td>This attribute...</td>
<td>Followed by...</td>
<td>Returns</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
<td>---------</td>
</tr>
<tr>
<td>owner:</td>
<td>user name</td>
<td>Items that are owned (published) by the specified users. <strong>Note:</strong> Prior to 8.2, owners were listed as publishers in Tableau Server. The <strong>publisher</strong> search attribute is still supported and returns the same results as the <strong>owner</strong> attribute.</td>
</tr>
<tr>
<td>publisher:</td>
<td>user name</td>
<td>(See owner above)</td>
</tr>
<tr>
<td>project:</td>
<td>search term</td>
<td>Items that are part of a project whose name matches the search term</td>
</tr>
<tr>
<td>comment:</td>
<td>search term</td>
<td>Views whose comments match the search term</td>
</tr>
<tr>
<td>tag:</td>
<td>search term</td>
<td>Items whose tags match the search term</td>
</tr>
<tr>
<td>field:</td>
<td>search term</td>
<td>Views with matching fields on the rows, columns, level of detail, pages, or encoding shelves</td>
</tr>
<tr>
<td>sheettype:</td>
<td>view, dash-</td>
<td>Views that are of the matching sheet type</td>
</tr>
<tr>
<td></td>
<td>board, or story</td>
<td></td>
</tr>
<tr>
<td>class:</td>
<td>type of data</td>
<td>Views and data sources that are associated with the matching type of data source</td>
</tr>
<tr>
<td></td>
<td>source (e.g., mysql)</td>
<td></td>
</tr>
<tr>
<td>dbname:</td>
<td>name of data-</td>
<td>Published data sources that are associated with the matching data source</td>
</tr>
<tr>
<td></td>
<td>base</td>
<td></td>
</tr>
<tr>
<td>nviews:</td>
<td>number</td>
<td>Workbooks that contain the specified number of views</td>
</tr>
</tbody>
</table>

**Note:** Quick search restricts the searches of **dbname** to published data sources only. To search on embedded data sources (workbooks that use a data connection only), use filtered search.

**Search operators**

You can use **and**, **or**, **not**, and **+** with search phrases, including combining them with attributes, to build search expressions. For example, if you want to search for all items that do
not match a specific phrase, or to match one phrase or another but not necessarily both.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>and</td>
<td>Returns items that match both search terms.</td>
<td>sales and marketing; pens and paper</td>
</tr>
<tr>
<td>or</td>
<td>Returns items that match either search term.</td>
<td>west or east; soccer and football</td>
</tr>
<tr>
<td>not</td>
<td>Excludes items that match the search term following this operator</td>
<td>not sheettype:dashboard</td>
</tr>
<tr>
<td>*</td>
<td>Acts as a substitute for any character or word following or as part of the search term. This operator can be used by itself or at the beginning or end of the search term. This operator is useful when you don’t know the exact term you are searching for.</td>
<td>dev* sales*</td>
</tr>
</tbody>
</table>

**Spaces and search**

If your search phrase includes spaces, punctuation, or reserved terms such as and, or, or not, enclose your search phrase in double quotation marks.

**Mark Favorites**

To make it easy to find your most-used workbooks, views, or published data sources, you can mark them as favorites.

**Note:** You cannot mark an embedded data source as a favorite.
Mark a workbook, views, or published data source as a favorite

- Select the favorites star next to the content.

Alternatively, from the list view, create a favorite by clicking the star next to the workbook, view, or published data source.
The content is added to your Favorites menu.

- Click the star that you used to mark the favorite.

Remove an item from Favorites

- Click the Favorites menu in the upper-right corner of the page. Type a search term to filter the list.
On the Favorites menu, ☑️ indicates a view, 📚 indicates a workbook, and 📁 indicates a published data source. Results are listed in alphabetical order.
If you have a large number of favorites, type a search term to filter the list or scroll to see all of them.

Manage Web Content

In Tableau Server and Tableau Online, content pages show available projects, workbooks, views, and data sources. On your private content page, you can also view all the content you own. The content and options you see are based on your site role and content permissions.

In this article

Use shared content pages below
Use your private content page on page 2638
Select content and perform actions on it on page 2640

Use shared content pages

Projects, workbooks, views, and data sources each have their own content pages with options that are specific to that content type. For example, a workbook content page includes menus for the views in the workbooks, data sources connected to the workbook, subscriptions, permissions settings, and details.
Each content page provides information such as the owner, and number of page views and favorites.

You can access actions that you're allowed to take on the content.
A breadcrumb trail at the top each content page shows your location in the site.

Each content item has a Details page. If you own the content and have appropriate permissions, you can edit the description, move the workbook to another project, change the owner, and more.
Projects

A project page includes the workbooks, views, and data sources in that project, as well as the default project permissions and project details.
Workbooks

A workbook page includes the views and data sources used by the workbook, and the workbook permissions and details.

Views

In a workbook, click a view to show the view and display options for interacting with it.
Data sources

In a site or a project, click **Data Sources**, and then click a data source name. The content page shows connections, refresh schedules, connected workbooks, permissions, and details.

Use your private content page

To see the content you own, click your profile image or initials at the top of the page, and then click **My Content**.

Your content page includes your user information, the workbooks and data sources that you own, your subscriptions, and your account settings.
Click **Settings** to go to your account settings from your content page.
Select content and perform actions on it

On the Projects, Workbooks, Views, and Data Sources pages, click check boxes to select content. Then click the Actions menu (…) to access commands available for that content.

Notes When you select multiple items, some options in the Actions menu may apply only to a subset of your selection. Tableau Online or Tableau Server will notify you of success or failure for individual items.

To select all content on the page, click the drop-down arrow next to n selected, and click Select All. To clear a content selection, click the drop-down arrow, and click Clear.
In thumbnail view, you access the **Actions** menu from the upper-right corner of a thumbnail.

In list view, you access the **Actions** menu to the right of content names.
Work with Content Revisions

When you publish a workbook or data source, a version is saved in the revision history for Tableau Server and Tableau Online. You can revert to a previous version at any time.

In this article

- **Required permissions for revision history** on the next page
- **Publish your content** on the next page
- **View revision history** on page 2644
- **Manage revisions** on page 2645
- **Potential revision history issues** on page 2648
Required permissions for revision history

To access revision history, you must have a site role of Creator or Explorer (Can Publish), plus the following permissions, depending on the content type:

- Project: View and Save
- Workbooks in the project: View, Save, and Download Workbook/Save As
- Data sources in the project: View, Save, and Download Data Source

Publish your content

In Tableau Desktop, click Server > Publish Workbook or Server > Publish Data Source. Make changes to the workbook or data source, and then publish it again to the same project, with the same name. (You’ll need to confirm that you want to overwrite the existing content.)

You can also save workbook revisions by editing and saving a workbook in the web-authoring interface of Tableau Server and Tableau Online.
View revision history

- Select a workbook or data source, and from its actions menu (…), select Revision History.

The following image shows a workbook's revision history.
Manage revisions

Preview a workbook revision

- From a revision’s actions menu (ellipsis), select Preview.

![Revision History](image1.png)

If a preview is available, it opens in a new tab in the browser. For workbooks that you can't preview on the server, such as those with OAuth data connections, you can download the revision and open it in Tableau Desktop.

Download a workbook or data source revision

1. From a revision’s actions menu (ellipsis), select Download.

![Revision History](image2.png)

2. Open the downloaded file in Tableau Desktop.
Restore a workbook revision

- In the revision history, select a revision, and then click **Restore**.

![Revision History](image)

The restored version becomes the current version.

**Note:** If a revision can't be restored directly on the server, download the revision, open it in Tableau Desktop, and republish it.

**Restoring workbooks that require database credentials**

If a workbook uses a live data connection that prompts for a user name and password, you have the option to embed credentials for the connection. If the workbook uses a data source with multiple connections, you might need to provide credentials for each one.

If a workbook uses data extracts with scheduled refreshes and embedded credentials, you’ll need to edit the data connection to provide the credentials.

**Restore a data source revision**

For a variety of reasons—for example, extracts commonly are on refresh schedules—a previous revision of a data source will not include the extract as it was published at the time. To restore the extract, you can download the previous revision of the data source (in .tdsx format), open it in Tableau Desktop, and re-publish the extract.

For more information, see **Data changes and deletions** on page 2648 under **Potential revision history issues** on page 2648.
1. Sign in to the Tableau Server or Tableau Online web editing environment, navigate to the data source, and open its revision history.

2. On a revision’s actions menu (…), select Download.

![Revision History](image)

3. Open the downloaded file in Tableau Desktop, and then republish it with the same name, to the same location on your Tableau Server or Tableau Online site (confirming that you want to overwrite the existing version).

   The uploaded version becomes the most current version.

Delete a workbook or data source revision from history

1. From a revision’s actions menu (…), select Delete.

   The revision history list updates to indicate a revision has been deleted.
Potential revision history issues

Overwriting content with the same name

If a different author publishes over a workbook or data source with the same name, the most recent author becomes the owner of the content and can see its entire revision history.

Data changes and deletions

- Workbooks and data sources are downloaded with the latest configuration of their extract or data connection. If the data model or data connection has changed between revisions, you might need to update the downloaded workbook or data source.
- Revisions of workbooks and data sources that use .xls or .csv files are saved with an extract of that data.
- Revisions are saved for extract files, unless they’re on a refresh schedule.
- If a workbook or data source is deleted from a site, all revisions are also deleted.

Turning revision history on and off

On Tableau Server, server administrators can disable revision history for specific sites.

If revision history is turned on and then off, saved revisions are retained, and new versions overwrite the latest version. If revision history is then turned on again, version numbering starts from the last saved revision.

If you’re a Tableau Server administrator, see Enable Users to Save Revision History for more about revision history settings.
Refresh Data or Pause Automatic Updates

Tableau Server and Tableau Online let you control how views interact with data sources so you can keep data fresh and improve performance.

Refresh Data

If a data source is changed (for example, with new fields, field names, or data values), the view reflects those changes the next time you load the page. To manually update data while you interact with a view, click Refresh on the toolbar.

When you refresh data, Tableau Server and Tableau Online clear any cache and retrieve the latest information from the data source. If you’re working with a complex view or a very large data source, refreshing data can take a long time.

Pause Automatic Updates

As you interact with a view, the server sometimes has to query the data source to update the view. To temporarily stop updates so you can more quickly interact with a view, click Pause on the toolbar.

Click Resume to again automatically query the data source as needed.

Use Custom Views

If you find yourself changing a Tableau Server or Tableau Online view every time you open it, consider saving your changes as a custom view. For example, if you apply a filter to include only data relevant to you, or you sort data differently, you can save separate custom views for each scenario.
Tip: A custom view doesn't change the original view, but it remains related to it. If the original view is updated or republished (with different filters, for example), your custom view is updated too. Likewise, if the original view is deleted from the server, your custom view is deleted.

In this article

Save a custom view below
Choose a different custom view on the next page
Change your default view on page 2652
Make a custom view public or private on page 2653
Delete a custom view on page 2654

Save a custom view

1. Open the individual view that you want to customize.

2. Filter the data, change sorting, select specific marks, zoom in or out, or make any other modifications.
   
   In the toolbar, the View option now includes an asterisk, indicating that the view has changed.

3. Click View in the toolbar.

4. Enter a name for the custom view. Then select whether you want it to be the default view you see, and if it should be public so other users can access it.
5. Click **Save**.

Choose a different custom view

1. Click **View** in the toolbar.
   
   The currently selected view appears with a gray background.

2. Click another view to display it.

Custom views that you have saved appear under **My Views**. The original view and views created by others appear under **Other Views**.
Change your default view

1. Click View in the toolbar, and then click Manage in the lower right corner of the Custom Views dialog box.

![Custom Views dialog box]

2. Click ▪ next to the custom view name to make the view the default view. The icon changes to ▪ to indicate that the view is now the default.

![Manage dialog box]

3. Click Done.
Make a custom view public or private

By default, a custom view is private so only you see it in your list. But if your site role is Interactor or Publisher, you can make a view public so other users can see it too. Anyone who has access to the original view will have access to your public, custom version.

1. Click View in the toolbar, and then click Manage in the lower right corner of the Custom Views dialog box.
2. Click the public or private icon to change the view’s status.

- The public icon 🈯️ indicates that the view is shared:
The private icon indicates that the view can be seen only by you:

3. Click **Done**.

**Delete a custom view**

You can delete a custom view you created at any time. Removing your custom view does not affect the original view.

1. Click **View** in the toolbar, and then click **Manage** in the lower right corner of the **Custom Views** dialog box.

2. Click the delete icon next to the custom view name.
3. Click **Done**.

**Manage Your Account Settings**

In Tableau Server and Tableau Online, your Account Settings page lets you manage your credentials, email settings, and user interface options in one central place.

In this article

- **Go to your Account Settings page** on the next page
- **Manage your credentials and passwords** on the next page
- **Remove connected clients** on page 2657
- **Change subscription settings** on page 2657
- **Change your start page** on page 2658
- **Change language and locale** on page 2658
- **Enable or disable extract refresh failure emails** on page 2658
- **Change your display name, password, or email address for Tableau Server** on page 2659
Go to your Account Settings page

- At the top of the browser window, click your profile image or initials, and then select My Account Settings.

To change your profile image, click the current image or initials on your account settings page.

Manage your credentials and passwords

When you access a workbook or data source that has a live connection to data and requires you to sign in, Tableau offers to save your password for you. If you accept, it stores your credentials in a cookie or an access token, depending on the data type. You can remove these credentials if you no longer use the data, or you have exceeded the maximum number of saved credentials and want to make room for a new one.

Under Saved Credentials, do either of the following:
Select the **Delete** link next to an individual access token.

Select **Clear All Saved Credentials**.

When you clear all credentials, the following items are removed from your user account:

- Passwords you used to access published data sources or workbooks that connect to them.
- Access tokens for OAuth data connections, such as to Google or Salesforce.com data.

**Caution:** Removing an access token is effectively like “changing the locks.” If the token is stored with workbooks or data sources you published, deleting the token also removes access to the data from those workbooks and data sources. If the token is embedded in an extract connection, and the extract is refreshed on a schedule, the refreshes will not be able to complete until you embed the new credentials or access token into the connection.

**Remove connected clients**

The first time you sign in to Tableau Server or Tableau Online from a Tableau **client** like Tableau Desktop or Tableau Mobile, a secure token is created is stored in your account. This token allows you to access the site directly, without having to sign in.

You can delete a connected client (token) if you no longer use it, or if you want to add a new client but get an error that says you’re already using the maximum number of connected clients. After you remove a connected client from your account, you will need to provide your credentials the next time you access Tableau Server or Tableau Online from that client.

- In the **Connected clients** section, next to the client you want to remove, select **Delete**.

**Change subscription settings**

1. Under **Subscription time zone**, select the time zone for schedules you create.

2. To modify subscriptions, click **Subscriptions** at the top of the page, and then select a workbook or view.

3. From the **Actions** drop-down menu, select **Change Schedule, Change Subject, Change Empty View Mode**, or **Unsubscribe**.
(The empty-view option sends subscription emails only when data exists in a view. It's a good choice for high-priority alerts.)

**Change your start page**

To change the start page that appears when you sign in, navigate to the page you want, click your name in the upper right area of the page, and then click **Make This My Start Page**. Updates to your start page take effect the next time the page loads completely, or after you sign out, and then sign in again.

To return to the default start page, click your user name, and then click **My Account Settings**. In the start page section, click **Reset to Default**. (The URL for your current start page is displayed here too; click the link to go to the page.)

![Start page](image)

**Change language and locale**

The **Language** setting determines the language you see for user interface options. **Locale** affects views, such as how numbers are formatted, or which currency is used.

Change the **Language** and **Locale**, and then click **Save Changes**. The language and locale update immediately.

![Language and Locale](image)

If you use Tableau Server and want to select a language that isn't currently supported, contact your administrator.

**Enable or disable extract refresh failure emails**

If you own published data sources that contain extracts on a refresh schedule, you can receive email notification when the scheduled refresh does not complete successfully.

- Under **Refresh Failure Notification**, select the **Send email when extract refreshes fail** check box to opt in, or clear it to opt out.
The email contains the following information:

- Extract or workbook name.
- The last successful refresh time.
- The number of consecutive times the refresh has failed.
  After five consecutive failures, refreshes are suspended until you take an action to resolve the cause of the failure.
- A suggested action to take to address the cause of the failure—such as updating embedded credentials or a path to the original data file—and a link to the server to take the action.

Change your display name, password, or email address for Tableau Server

Change your display name

If the server is configured to use the internal user management system (Local Authentication) instead of Active Directory, you can change your display name. Select the display name text and enter the new display name, and then click **Save Changes**.

![Display Name Change](image1)

Change your password

If the server is configured to use the internal user management system (Local Authentication) instead of Active Directory, you can change your password by clicking **Change Password**. Click **Save Password** to save your changes.

![Password Change](image2)

Change your email address

If you subscribe to views or receive data-driven alerts, the related email account appears on the Account Settings page. Enter the new email address in the **Email** text box, and then click **Save**.
Changes.

Change your display name or password for Tableau Online

If your site is not set up for single sign-on, your Tableau Online display name and password are based on your TableauID account. Your TableauID allows access to Tableau Online, the Tableau website, the Customer Portal, and other resources.

If you forgot your password

To reset your password, go to https://online.tableau.com, click Forgot password, and enter the email address you use to sign in to Tableau Online. Then follow the instructions in the email you receive.

If you’re signed in to Tableau Online

1. Open your Account Settings page, and click Change Password.
   This redirects you to the Tableau website.

2. If prompted, sign in using your Tableau Online credentials, and then select the Change Password link at the top.
   - In the fields provided, type your current and new passwords, confirm the new password, and click Change Password.
To change your display name, click **Manage Account**.

On the User Profile page, change your first or last name, update any other information that might be out of date, and click **Update**.

**Note:** When you change your display name or password, you're directed outside of Tableau Online to your TableauID account profile. You can also access your TableauID profile by navigating to the Tableau website and clicking **Sign In**.
Manage Saved Credentials for Data Connections

On Tableau Online, you can manage saved credentials on your Account Settings page. Saved credentials enable you to connect to a data source without being prompted for your credentials. The credentials saved for your connection can be OAuth access tokens, or other credentials, such as user name and password.

On Tableau Server, if your server administrator has allowed you to save credentials, you can find and manage them in the Saved Credentials section on your Account Settings page.

Note: If you do not see the Saved Credentials section, consult with your Tableau Server administrator about allowing saving credentials.

Remove saved credentials

To remove Tableau access to data, delete the associated saved credentials for that data from your account.

After you delete the credentials, you will need to sign in to the data the next time you access it. This will create new saved credentials.

Your administrator might choose for all users to use the same shared credentials for connecting to a data source. If this is the case, the saved credential is associated with the data connection for all users, and it does not appear under Saved Credentials on your Account Settings page.

Note: If you’re a Tableau Server user and can’t delete saved credentials, ask your administrator if they’ve cleared the Allow users to save data source access tokens option in the server settings.

Test connections using saved credentials

If the connector supports test functionality, you can test the connection using saved credentials.

1. While you’re signed in to Tableau Server or Tableau Online, display your Account Settings page.

2. In the Saved Credentials section, click the Test link next to the stored connection that you want to test.
This test confirms that Tableau Online or Tableau Server can access your account using this corresponding saved credential. If the test succeeds, but you cannot access your data through this managed connection, confirm that the credentials you provided for this connection can access your data.

For example, if you accidentally created the connection using your personal Gmail account, but you use a different account to access a Google Analytics database, you will need to delete the saved credentials and sign in to the data using the appropriate Gmail account.

Clear all saved credentials

When you select **Clear All Saved Credentials**, the following items are removed from your user account:

- All saved credentials for connections that are stored in your account.

  **Caution:** If any of these saved credentials are stored with published workbooks or data sources, deleting them also removes access to the data source from those locations. Effectively, this is like "changing the locks" anywhere the affected saved credentials are used.

- Passwords you have used to access published data extracts or workbooks that connect to them.
Share and Collaborate on the Web

Share Web Views

Every published view and workbook can be shared via links in email and other applications, or embedded directly into web pages, wikis, and web applications. To see a shared view, users must have permission to access it in Tableau Server or Tableau Online.

Link to a view

1. Click **Share** in the upper-right corner of the view.

2. Copy the provided link into another application. Click the **Email** button to automatically create an email message with the link.

   ![Share dialog](image)

   **Note:** If you use this link in your own embed code, add the HTML attribute `allowfullscreen="true"` to the `iframe` element to enable the Full Screen button.
Embed a view

1. Click **Share** in the upper-right corner of the view.

2. Copy the provided embed code, and then paste it into the source code of the page in which you want to embed the view.

![Share Icon](image)

**Note**: The embed code generated by Tableau automatically refers to the current view. For information about how custom views are displayed, see Embed Code for Custom Views.

Subscribe to Views

Subscriptions email you a snapshot of a view at regular intervals—without requiring you to sign in to Tableau Server or Tableau Online.

**Note**: If you use Tableau Server, administrators determine whether subscriptions are enabled for a site.

In this article

*Subscribe yourself or others to a view* on the next page

*Unsubscribe yourself from a view* on page 2668
Subscribe yourself or others to a view

When you open a view in Tableau Server or Tableau Online, and you see a subscription icon (Subscribe) in the toolbar, you can subscribe to that view or the entire workbook. If you own a workbook or are a project leader with an appropriate site role or an administrator, you can also subscribe other users who have permission to view the content.

1. In a site, click Views or Workbooks.

![Image of Tableau interface showing Views and Workbooks]

2. Open a view either directly, or after opening the containing workbook.

3. In the toolbar above the view, click Subscribe.

4. Add the Tableau users or groups you want to receive the subscription. To receive a subscription, users must have permissions to view the content. If they use Tableau Server, their accounts must also have email addresses.

**Note:** When you subscribe a group, each user is added individually at the time the subscription is created. If more users are added to the group later, you must re-
subscribe the group for those new users to receive the subscription. Likewise, users later removed from the group will not have their subscriptions removed automatically unless their permissions to the subscribed view are removed.

5. Choose whether subscription emails include the current view or the entire workbook.

6. Pick a schedule:
   - For Tableau Server, choose from subscription schedules established by your administrator.
   - For Tableau Online and Tableau Server version 2018.2 with custom schedules enabled, click the drop-down arrow to the right of the current settings.

   ![Schedule Options](image)

   Then specify a custom schedule that sends subscription emails whenever you wish. (The precise delivery time may vary if server load is high.)

   ![Custom Schedule](image)

   To change the time zone, click the Time Zone link it to go to your account settings page.

7. To clarify subscription emails, customize the subject line, and add a message.

8. If the view contains data only when high-priority information exists, select Don't send if view is empty.

9. If you own the workbook, select Subscribe me.
10. Click **Subscribe**.

When you receive the email on the subscription schedule, click the image to open the view in Tableau Server or Tableau Online.

**Unsubscribe yourself from a view**

1. Access your Tableau Server or Tableau Online account settings by doing one of the following:
   - Click **Manage my subscriptions** at the bottom of a subscription email.
   - Sign in to Tableau Server or Tableau Online. At the top of the page, select your
name, and then select **My Content**.

2. Click **Subscriptions**.

3. Select the check box next to the view you want to unsubscribe from, click **Actions**, and then click **Unsubscribe**.

See also

- Change Subscription Settings in the Tableau Desktop and Web Authoring Help.
- Project-level administration in the Tableau Online Help, to learn which site roles allow full Project Leader capabilities.

## Send Data-Driven Alerts from Tableau Online or Tableau Server

When data reaches important thresholds for your business, data-driven alerts automatically send email notifications to key people you specify. You can set data-driven alerts on dashboards and views, but not story points.

For time-based charts, use relative date filters so people automatically receive alerts as new data appears. If you don’t own the content, ask the author to make this change.

### Create a data-driven alert

1. Select a continuous numeric axis of any chart other than a Gantt chart or map. (Numeric bins and discrete numeric axes aren’t supported.)
2. In the toolbar, click **Alert**.

3. In the Create Alert pop-up window, set the data condition and threshold value that triggers alert emails.

   In the view, a red line shows where the threshold falls relative to your current data.
4. Specify the email subject line, schedule, and recipients. Then click **Create Alert**.

**Note:** For live data sources, alerts are checked every 60 minutes for Tableau Online, or on a frequency set by administrators for Tableau Server. Extracts are checked each time they’re refreshed.

**Manage your data-driven alerts**

You can manage alerts from the **My Content** area of Tableau web pages, but doing so directly from alert emails can be quicker. Click the links below the image to manage all your alerts, or add or remove yourself from the emailed alert. If you’re the alert owner, click **Edit this alert** to change alert settings such as threshold, schedule, and recipients.

**Tip:** To add recipients to an alert you don’t own, forward the email and ask people to click **Add me to this alert**.
Fix failing alerts

Alerts can sometimes fail, usually for these reasons:

- Temporary connectivity issues. In this case, the alert will fix itself.
- A data source has been removed
- Credentials to data have expired
- The workbook or sheet that the alert was created on has been removed.

If an alert fails, you'll receive a notification email that tells you which alert failed and when, with a link to your site to fix the problem.

There are a few ways to fix a failing alert if you're the alert owner:
• Open the workbook in Tableau Online or Tableau Server. A warning will appear if credentials to your data have expired or if the data for your workbook has been removed.

• Click **Edit this alert** to change alert settings such as threshold, schedule, and recipients. A warning will appear to tell you if the workbook or sheet that your alert was created on has been removed.

You'll receive an email notification when the alert is working again.

**Comment on Views**

Comments let you share a conversation about data discoveries with other Tableau Server or Tableau Online users. Any comments you add in a desktop browser also appear in Tableau Mobile, and vice versa, so you can easily communicate with colleagues on the go.

**Tip:** To keep comments when you republish a workbook, be sure to use the same workbook and view names.

**Add comments**

1. In the toolbar above a view, click **Comments**.

2. In the Comments pane at right, enter your remarks, and @mention colleagues to notify them via email.

3. If you've filtered the view, click the snapshot icon to share an interactive image that highlights the data you're describing.

   **Note:** The snapshot icon doesn't appear if a view is **user-filtered**, protecting secure data.

4. Click **Post**.
5. To filter a view and see a data snapshot in detail, hover over the thumbnail image, and click View.
Delete comments

If a comment is unnecessary or inaccurate, you can quickly delete it. Just click the X in the upper-right corner.

You can delete a comment if you created it, are the content owner, are a project leader with an appropriate site role, or are an administrator. To learn which site roles are required for full project leader access, see Project-level administration in the Tableau Online Help.

Embed Views and Dashboards

Embed Views into Webpages

You can embed interactive Tableau views and dashboards into web pages, blogs, wiki pages, web applications, and intranet portals. Embedded views update as the underlying data
changes, or as their workbooks are updated on Tableau Server or Tableau Online. Embedded views follow the same licensing and permission restrictions used on Tableau Server and Tableau Online. That is, to see a Tableau view that's embedded in a web page, the person accessing the view must also have an account on Tableau Server or Tableau Online.

Alternatively, if your organization uses a core-based license on Tableau Server, a Guest account is available. This allows people in your organization to view and interact with Tableau views embedded in web pages without having to sign in to the server. Contact your server or site administrator to find out if the Guest user is enabled for the site you publish to.

You can do the following to embed views and adjust their default appearance:

- **Get the embed code provided with a view:** The Share button at the top of each view includes embed code that you can copy and paste into your webpage. (The Share button doesn’t appear in embedded views if you change the showShareOptions parameter to false in the code.)

- **Customize the embed code:** You can customize the embed code using parameters that control the toolbar, tabs, and more. For more information, see Parameters for Embed Code on page 2679.

- **Use the Tableau JavaScript API:** Web developers can use Tableau JavaScript objects in web applications. To get access to the API, documentation, code examples, and the Tableau developer community, see the Tableau Developer Portal.

**Note:** For users to successfully authenticate when they click an embedded view, their browsers must be configured to allow third-party cookies.

**Writing Embed Code**

If you’re writing your own embed code, you can take one of two approaches:

- **Use Tableau JavaScript:** This is the preferred approach. Use the embed code that Tableau generates as the starting point for your own code, adding or editing object parameters that control the toolbar, tabs, and more. The default embed code, which relies on a Tableau JavaScript file, is also the only way to control the load order of multiple embedded views.

- **Specify the View URL:** Embed a view using an iframe or image tag, where the source is the URL from the Link box of the Share dialog box. You may want to do this if you can’t
use JavaScript on your website. There may also be situations when all you can specify is a URL.

When you embed a view, you should define a width and height that the view will be displayed in. Otherwise, the client browser will arbitrarily pick a width and height.

**Note:** When you need to specify the server name for Tableau Online in embed code (for example, to point to the location of the JavaScript API), use the URL **https://online.tableau.com**.

**Use Tableau JavaScript**

The following code shows an example of embed code that is generated when you click `Share` on a published view. Special characters in the `host_url` parameter are URL encoded, and those in the `site_root` and `name` parameters are notated as HTML numeric character references.

**Tableau Server example:**

```html
<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder' style='width:800; height:600;'>
<object class='tableauViz' width='800' height='600' style='display:none;'>
   <param name='host_url' value='http%3A%2F%2Fmyserver%2F' />
   <param name='site_root' value='&amp;#47;t&amp;#47;Sales' />
   <param name='name' value='MyCoSales&amp;#47;SalesScoreCard&amp;#47;' />
   <param name='tabs' value='yes' />
   <param name='toolbar' value='yes' /></object>
</div>
```

**Tableau Online example:**

```html
<script type='text/javascript' src='https://online.tableau.com/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder' style='width:800; height:600;'>
<object class='tableauViz' width='800' height='600' style='display:none;'>
   </object>
</div>
```
The source for the `<script>` tag is the URL for the Tableau Server and Tableau Online JavaScript file, `viz_v1.js`. The JavaScript file handles assembling the full URL of the view that’s displayed for your users. The `name` and `site_root` object parameters are the only required parameters; all other parameters are optional.

**Specify the View URL**

Here’s an example of embedding the same view using an IFrame, where the source is the URL from the Link box of the Share dialog box:

**Tableau Server example:**

```html
<iframe src=
  c="http://myserver/t/Sales/views/MyCoSales/SalesScoreCard
?:embed=yes&:tabs=yes&:toolbar=yes" width="800"
height="600"></iframe>
```

**Tableau Online example:**

```html
<iframe src="h-
https://online.tableau.com/t/Sales/views/MyCoSales/SalesScoreCard
?:embed=yes&:tabs=yes&:toolbar=yes" width="800"
height="600"></iframe>
```

The `embed` URL parameter is required, and you can optionally include parameters that control the toolbar and revert options, among others. You can also add filters to the URL that control the specific data that shows when a view is loaded.
Parameters for Embed Code

You can configure an embedded view using either of these methods:

- **Object parameters for JavaScript tags** below
- **URL parameters for iframe tags** on page 2694

Object parameters for JavaScript tags

<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert-s</td>
<td>no</td>
<td>Hides the Alerts button in the toolbar, which lets users create data-driven alerts.</td>
<td><code>&lt;param name='alerts' value='no'/&gt;</code></td>
</tr>
<tr>
<td>customViews</td>
<td>no</td>
<td>Hides the View button in the toolbar, which lets users save</td>
<td><code>&lt;param name='customViews' value='no'/&gt;</code></td>
</tr>
<tr>
<td>Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td>custom views.</td>
<td>If a dashboard has layouts for mobile devices, displays a specific layout, regardless of screen size. If this parameter isn't set, Tableau Server or Tableau Server.</td>
<td><code>&lt;param name='device' value='phone'/&gt;</code></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>au Online detects screen size and loads a corresponding layout. See <a href="#">Embed Dashboards on page 2714</a> for examples.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filter string</td>
<td>Filters the data displayed when the view</td>
<td>&lt;param name='filter' value='Team=Blue'/&gt;</td>
<td></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>open-s. You can also filter using URL parameters.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| host_url | string | Required. The server name as it appears in the URL. | `<param name='host_url' value='http://myserver.exampleco.com'/>`
`<param name="host_url" value="http://localhost/">` |
<p>| link-target | string | The target window name for external hyperlinks. | <code>&lt;param name=&quot;linktarget&quot; value=&quot;_blank&quot;/&gt;</code> |</p>
<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>load-order</td>
<td>number</td>
<td>When multiple views are embedded, determines the order in which they load on the page. Negative numbers are allowed.</td>
<td><code>&lt;param name=&quot;load-order&quot; value=&quot;2&quot;/&gt;</code></td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>Required object parameter, with this</td>
<td><code>&lt;param name='name' value='ExampleCoSales/Sales'/&gt;</code>&lt;br&gt;<code>&lt;param name='name' value=&quot;ExampleCoSales/Sales/jsmith@example.com/EastRegionSales'/&gt;</code></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td>structure: [work-book name]/ [sheet name]/ [optional custom view name]</td>
<td>Forma t optional custom view names like this: username@domain/ [custo</td>
<td></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>m view name ]</td>
<td></td>
<td>If you refer to the Table au Server or Table au Online URL to confirm the value of name, exclude the session ID ( :iid =&lt;n&gt; ) at the end of the URL.</td>
<td></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| :original_view  | yes    | If the name parameter refers to a work-book or sheet URL (and does not explicitly refer to a custom view) including this parameter displays the view as the ori-
<p>|                  |        |             | &lt;param name='filter' value=':original_view-w=yes'/&gt; |</p>
<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>path</td>
<td>string</td>
<td>For trusted authentication only, cannot be used with the ticket parameter. Overrides value of the name parameter and is used as the URL. For more</td>
<td><code>&lt;param name='path' value='trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4'/&gt;</code>&lt;br&gt;<a href="http://tableauserver/trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4?:embed=yes&amp;:tabs=yes">http://tableauserver/trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4?:embed=yes&amp;:tabs=yes</a></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>information, see Display the View with the Ticket in the Tableau Server Adminstrator help.</td>
<td></td>
</tr>
<tr>
<td>showShareOptions</td>
<td>true; false</td>
<td>Controls whether the Share options are displayed in an embedded view.</td>
<td>&lt;param name='showShareOptions' value='true' /&gt;</td>
</tr>
<tr>
<td>Object Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>site_root</td>
<td>string</td>
<td>Required. The site name. The Default site value is null (value=''). If your server is multi-site and you want to use trusted authentication, see Display the</td>
<td>&lt;param name='site_root' value='/#/Sales'/&gt; &lt;param name='site_root' value=''/&gt;</td>
</tr>
<tr>
<td>Object Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>View with the Ticket in the Tableau Server help.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>subscriptions</td>
<td>no</td>
<td>Hides the Subscriptions button in the toolbar, which lets users set up email subscriptions to receive snapshots of</td>
<td>&lt;param name='subscriptions' value='no'/&gt;</td>
</tr>
<tr>
<td>Object Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>views</td>
<td>at regular intervals.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tabs</td>
<td>yes; no</td>
<td>Displays or hides tabs.</td>
<td><code>&lt;param name='tabs' value='yes'/&gt;</code></td>
</tr>
<tr>
<td>ticket</td>
<td>string</td>
<td>For trusted authentication only, cannot be used with the path object parameter. Must be used with</td>
<td><code>&lt;param name='ticket' value='9D1ObyqDQmSI0yQpKdy4Sw==:dg62gCsSE0QRArXNT0p6mlJ5'/&gt;</code>&lt;br&gt;<a href="http://tableauserver/trusted/9D1ObyqDQmSI0yQpKdy4Sw==:dg62gCsSE0QRArXNT0p6mlJ5/views/workbookQ4/SalesQ4?:embed=yes&amp;tabs=yes">http://tableauserver/trusted/9D1ObyqDQmSI0yQpKdy4Sw==:dg62gCsSE0QRArXNT0p6mlJ5/views/workbookQ4/SalesQ4?:embed=yes&amp;tabs=yes</a></td>
</tr>
<tr>
<td><strong>Object Parameter</strong></td>
<td><strong>Values</strong></td>
<td><strong>Description</strong></td>
<td><strong>Examples</strong></td>
</tr>
<tr>
<td>----------------------</td>
<td>------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>name object to construct the trusted ticket redemption URL. For more information, see Display the View with the Ticket in the Tableau Server help.</td>
<td>name object to construct the trusted ticket redemption URL. For more information, see Display the View with the Ticket in the Tableau Server help.</td>
<td>name object to construct the trusted ticket redemption URL. For more information, see Display the View with the Ticket in the Tableau Server help.</td>
<td>name object to construct the trusted ticket redemption URL. For more information, see Display the View with the Ticket in the Tableau Server help.</td>
</tr>
<tr>
<td><strong>toolbar</strong></td>
<td>yes; n-</td>
<td>The toolbar</td>
<td>&lt;param name='toolbar' value=top'/&gt;</td>
</tr>
<tr>
<td><strong>Object Parameter</strong></td>
<td><strong>Values</strong></td>
<td><strong>Description</strong></td>
<td><strong>Examples</strong></td>
</tr>
<tr>
<td>----------------------</td>
<td>------------</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>o; to; p</td>
<td>bar is displayed on the bottom by default. The toolbar is placed above the view when you set this parameter to <strong>top</strong> and excluded from the embedded view when you</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>tooltip</td>
<td>yes; no</td>
<td>Tool-tips are displayed by default If you set this parameter to no, however, tool-tips are excluded from the embedded view.</td>
<td><code>&lt;param name='tooltip' value='no'/&gt;</code></td>
</tr>
</tbody>
</table>

URL parameters for iframe tags

Note: Before you add URL parameters, remove `:iid=[#]` at the end of the URL. This is a temporary view counter for your current browser session.
<table>
<thead>
<tr>
<th>URL Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>:alerts</td>
<td>no</td>
<td>Hides the Alerts button in an embedded view.</td>
<td><a href="http://tabserver/views/Date-Time/DateCalcs?:embed=yes&amp;:alerts=no">http://tabserver/views/Date-Time/DateCalcs?:embed=yes&amp;:alerts=no</a></td>
</tr>
<tr>
<td>:customViews</td>
<td>no</td>
<td>Hides the Remember my changes option.</td>
<td><a href="http://tabserver/views/Date-Time/DateCalcs?:embed=yes&amp;:customViews=no">http://tabserver/views/Date-Time/DateCalcs?:embed=yes&amp;:customViews=no</a></td>
</tr>
<tr>
<td>:device</td>
<td>desktop; tablet; phone</td>
<td>If a dashboard has layouts for mobile devices, displays a specific layout, regardless of screen size. If this parameter isn't set, Tableau Server or Tableau Online detects screen size and loads a corresponding layout. See</td>
<td><a href="http://tabserver/views/sales/sales_dashboard?:device=tablet">http://tabserver/views/sales/sales_dashboard?:device=tablet</a></td>
</tr>
<tr>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>:embed</td>
<td>yes</td>
<td>Required URL parameter. Hides the top navigation area, better blending the view into your web page.</td>
<td><a href="http://tabserver/views/Date-Time/DateCalcs?:embed=yes">http://tabserver/views/Date-Time/DateCalcs?:embed=yes</a></td>
</tr>
<tr>
<td>:format</td>
<td>pdf; png</td>
<td>Displays a view as a PDF or .png file.</td>
<td><a href="http://tabserver/views/Sales/Q2?:format=pdf">http://tabserver/views/Sales/Q2?:format=pdf</a></td>
</tr>
<tr>
<td>:highdpi</td>
<td>false</td>
<td>Renders a view using standard DPI (dots per inch) for high-resolution displays and devices.</td>
<td><a href="http://tabserver/views/Sales/Q2?:highdpi=false">http://tabserver/views/Sales/Q2?:highdpi=false</a></td>
</tr>
<tr>
<td>:link-string</td>
<td>string</td>
<td>The target</td>
<td><a href="http://tabserver/views/Date-">http://tabserver/views/Date-</a></td>
</tr>
<tr>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>target</td>
<td>window name for external hyperlinks.</td>
<td>Time/DateCalc-s?:embed=yes&amp;:linktarget=_blank</td>
<td></td>
</tr>
<tr>
<td>:record_performance (Tableau Server only. Not applicable to Tableau Online.)</td>
<td>yes</td>
<td>Starts a performance recording for a view. Add this at the end of the URL, immediately before the session ID (:iid= &lt;n&gt;). For more information, see Create a Performance Recording in Tableau Server help.</td>
<td><a href="http://tabserver/views/Sales2013/Regions?:recordperformance=yes">http://tabserver/views/Sales2013/Regions?:recordperformance=yes</a></td>
</tr>
<tr>
<td>:refresh</td>
<td>yes</td>
<td>Renders the view using the latest data from</td>
<td><a href="http://tabserver/views/DateTime/DateCalc-s?:embed=yes&amp;:refresh=yes">http://tabserver/views/DateTime/DateCalc-s?:embed=yes&amp;:refresh=yes</a></td>
</tr>
<tr>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tableau Server or Tableau Online.</td>
<td></td>
</tr>
</tbody>
</table>

**Tip:**
To continually refresh a view, in the `<head>` section of the webpage, add `<meta http-equiv="refresh">`.
<table>
<thead>
<tr>
<th>URL Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>:render</td>
<td>true; false; number</td>
<td>If client-side rendering is enabled (the default), setting this to false forces server-side rendering for the session. If client-side rendering is enabled, set :render=true. Not applicable to Tableau Online.</td>
<td><a href="http://tabserver/views/Date-Time/DateCalcs?:render=false">http://tabserver/views/Date-Time/DateCalcs?:render=false</a></td>
</tr>
<tr>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rendering is disabled, setting this to <code>true</code> enables it for the session. A number from 1 to 100 can be entered to set the complexity threshold above which views are rendered by the server. For more information, see About Client-Side Rendering in Tableau Server help.</td>
<td></td>
</tr>
<tr>
<td>:revert</td>
<td><code>all; filters;</code></td>
<td>Returns the item to its original state.</td>
<td><a href="http://tabserver/views/DateTime/DateCalcs?:embed=yes&amp;:revert=all">http://tabserver/views/DateTime/DateCalcs?:embed=yes&amp;:revert=all</a></td>
</tr>
<tr>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sort;</td>
<td></td>
<td>Zettel [42]</td>
<td></td>
</tr>
<tr>
<td>axe;</td>
<td></td>
<td>Zettel [42]</td>
<td></td>
</tr>
<tr>
<td>shelves</td>
<td></td>
<td>Zettel [42]</td>
<td></td>
</tr>
<tr>
<td>:subscriptions</td>
<td>no</td>
<td>Hides the Subscribe button in an embedded view.</td>
<td><a href="http://tabserver/views/DateTime/DateCalcs?embed=yes&amp;:subscriptions=no">http://tabserver/views/DateTime/DateCalcs?embed=yes&amp;:subscriptions=no</a></td>
</tr>
<tr>
<td>:tabs</td>
<td>yes; no</td>
<td>Displays or hides tabs.</td>
<td><a href="http://tabserver/views/DateTime/DateCalcs?embed=yes&amp;:tabs=no">http://tabserver/views/DateTime/DateCalcs?embed=yes&amp;:tabs=no</a></td>
</tr>
<tr>
<td>:showShareOptions</td>
<td>true; false</td>
<td>Controls whether the Share options are displayed in an embedded view.</td>
<td><a href="http://tabserver/views/DateTime/DateCalcs?embed=yes&amp;:showShareOptions=true">http://tabserver/views/DateTime/DateCalcs?embed=yes&amp;:showShareOptions=true</a></td>
</tr>
<tr>
<td>:toolbar</td>
<td>yes; no; top</td>
<td>The toolbar is displayed by default on the bottom when this parameter is not set. When no the toolbar is</td>
<td><a href="http://tabserver/views/DateTime/DateCalcs?embed=yes&amp;:toolbar=no">http://tabserver/views/DateTime/DateCalcs?embed=yes&amp;:toolbar=no</a></td>
</tr>
<tr>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>excluded from the embedded view. When top, the toolbar is placed above the view.</td>
<td></td>
</tr>
<tr>
<td>:tooltip</td>
<td>yes; no</td>
<td>Tooltips are displayed by default in a view when this parameter is not set. If set to no, tooltips are excluded from the embedded view.</td>
<td><a href="http://t-abserver-/views/workbookQ4/SalesQ4?embed=yes&amp;:tooltip=no">http://t-abserver-/views/workbookQ4/SalesQ4?embed=yes&amp;:tooltip=no</a></td>
</tr>
</tbody>
</table>

**Add Filters to Embed Code**

You can include filter values in embedded views so they highlight just the data you want. For example, you may want to include a hyperlink from another part of your web application to an embedded sales performance view that shows only a specific region.

**Note:** Stories don't support embedded filters.
In this article

Filter on one field below
Filter on multiple fields on the next page
Filter dates and times on page 2705
Filter measures on page 2706

Filter on one field

Passing a filter on one field is a quick way to focus an embedded view on specific data.

Script Tag Example

```html
<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'></script>
<object class='tableauViz' width='800' height='600' style='display:none;'

    <param name='host_url' value='http://myserver/' />
    <param name='site_root' value='' />
    <param name='name' value='Superstore/Product' />
    <param name='filter' value='Region=East' />
</object>
```
To pass through multiple filter values, just separate each with a comma. For example:

```html
<param name='filter' value='Region=East,West' />
```

**Iframe Tag Examples**

```html
<iframe src=
="http://myserver/views/Superstore/Product?:embed=y&Region=East,West"
width="800" height="600"></iframe>
```

**Filter on multiple fields**

You can pass filters on as many fields as you want, including fields that are not in visible the original view.

**Script Tag Example**

```html
<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'></script>
<object class='tableauViz' width='800' height='600' style='display:none;'
  >
  <param name='host_url' value='http://myserver/' />
  <param name='site_root' value=''/>
  <param name='name' value='Superstore/Product'/>
  <param name='filter' value='Region=Central,South&Customer Segment=Consumer,Home Office'/>
</object>
```

**Iframe Tag Example**

```html
<iframe src="http://myserver/views/Superstore/Product?:embed=y&Region=Central,South&Segment=Consumer,Home Office"
width="800" height="600"></iframe>
```

The first image below shows an example of the URL you might get when you click **Share** on a view and copy the link in the **Link** field.
The second image shows how you might modify the URL and add it to an iframe by deleting the `showShareOptions` and `display_count` parameters, adding filter parameters for Region and Segment, and adding width and height parameters, to create an embed link that displays only Consumer and Home Office products from the Central and South regions.

```
<iframe src="http://myserver/views/Superstore/Product?embed=yes&showShareOptions=true&display_count=0&Region=Central, South&Segment=Consumer, Home Office" width="460" height="400"></iframe>
```

**Note:** If a filter value contains a special character, such as a comma, replace the character with the URL encoding sequence for \ (backslash, %5c) followed by the URL encoding sequence for the special character. The backslash is needed to escape the special character. For example, the URL encoding sequence for \, (backslash, comma) is %5c%2c.

Filter dates and times

If you want to filter on a Date/Time field, include the value using the default Tableau format shown below:

`yyyy-mm-dd hh:mm:ss`
The time part uses a 24-hour clock. Many databases store all date values as Datetime fields, so you may need to pass a time value along with your date.

**Script Tag Example**

```html
<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'></script>
<object class='tableauViz' width='800' height='600' style='display:none;'>
  <param name='host_url' value='http://myserver/' />
  <param name='site_root' value=''/>
  <param name='name' value='Sales/Sales-Performance'/>
  <param name='filter' value='Date=2012-12-01'/>
</object>
```

This example filters on both a date field and a datetime field:

```html
<param name='filter' value='2012-12-01%2022:18:00'/>
```

**Iframe Tag Example**

```html
<iframe src="http://myserver/Sales/Sales-Performance?:embed=yes&Date=2008-12-01%2022:18:00" width="800" height="600"></iframe>
```

To filter multiple dates, separate each date with a comma.

**Filter measures**

You can filter measures by including one or more values. There is no support for greater than, less than, or ranges. The example below filters to show only $100 and $200 sales.

**Script Tag Example**

```html
<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'>
</script>
<object class='tableauViz' width='800' height='600' style='display:none;'>
  <param name='host_url' value='http://myserver/' />
</object>
```
Iframe Tag Example

<i)frame src="http://myserver/views/Sales/Sales-Performance?embed=yes&Profit=100,200" width="800" height="600"></i)frame

How View URLs Are Structured

To understand how Tableau structures view URLs, let's take a close look at an example in the browser address bar.

In this article

The structure of the base URL below
Adding parameters to the base URL below
Select an example view to work with on the next page
Filter the view on page 2709
Exclusive filtering on page 2711
Special considerations for date filtering on page 2712

The structure of the base URL

- In Tableau Server or Tableau Online, the basic URL pattern for a Tableau view is:
  
  http://<servername>/#/views/<workbook>/<sheet>

- In a multi-site environment, for views saved to sites other than Default, the site ID is also included in the URL:

  http://<servername>/#/site/<sitename>/views/<workbook>/<sheet>

Adding parameters to the base URL

To create a filtering URL, you append a query string to the base URL.
- For URLs pointing to Tableau Online or Tableau Server, start the query string with a question mark (?).

- For URLs pointing to Tableau Public, because default URLs already include a query string, begin your parameter code with the ampersand (&) character.

Within the query string, parameters (filters) are separated by ampersands (&), and multiple values per parameter are separated by commas. For example:

```
http://<servername>/#/views/<workbook>/sheet
?param1=value1,value2&param2=value
```

In addition, any characters in your field or sheet names that cannot appear in URLs are URL encoded. For example, a space is converted to %20.

**Note:** Before you add URL parameters, remove :iid=[#] at the end of the URL. This is a temporary view counter for your current browser session.

Select an example view to work with

The examples in this article use the Performance view, in the Superstore sample data that comes with Tableau Server:

![Superstore Performance view](image)

The example URL reflects the following attributes:
• The sheet name is **Performance**
• The workbook name is **Superstore**
• The server name is indicated using a placeholder IP address, where your on-premise server name, or online.tableausoftware.com or public.tableausoftware.com would appear.

To follow along with the examples, you can look at a similar view from Superstore, or you can adjust the example URLs to use the field, sheet, and other names from your own environment. If you are using your own view, make sure you publish it to the server.

**Filter the view**

For the example view used in this article, the base URL is:

http:<servername>/#/views/Superstore/Performance

**Furniture Only**

To display only sales of furniture, at the end of the URL, add the following query string:

?Category=Furniture

For example:

http:<servername>/#/views/Superstore/Performance

?Category=Furniture

**Consumer Only**

To limit the view to Consumer sales, change the URL to:
Home Office and Consumer

To show both home office and consumer sales, change the URL to:

http:<servername>/#/views/Superstore/Performance?Segment=Home%20Office,Consumer

Notice that in the field name "Home Office," %20 represents the URL-encoded space character, as described in Adding parameters to the base URL on page 2707.

Here multiple values are separated with a comma, but not a space.
Furniture in the Central

To show multiple field/value pairs, separate them with an ampersand.

http:<servername>/#/views/Superstore/Performance
?Region=Central&Category=Furniture

Exclusive filtering

So far the examples shown in this article display all values for the fields in the resulting views. For more flexibility, you can specify multiple values for a specific filter, and include an additional
parameter that displays only particular intersections of those values.

The following URL describes the base, unfiltered view used in this example, showing department sales by region.

http://<servername>/#/views/Superstore/Performance

Next, to show sales for only the Furniture and Technology departments in the Central and West regions, you would add the following query string to the base URL:

?Region=Central,West&Category=Furniture,Technology

To show only Furniture sales in the Central region and Technology sales in the West region, add the ~s0 parameter to the query string used in the previous step.

?Region~s0=Central,West&Category~s0=Furniture,Technology

The final URL looks like this:

http://<servername>/#/views/Superstore/Performance?Region~s0=Central,West&Category~s0=Furniture,Technology

Special considerations for date filtering

When you want to filter date fields, take into account how dates are formatted and behave in a database environment:

- Date (and time) values passed via URL parameter need to match the following Tableau default format:
  
  yyyy-mm-dd hh:mm:ss

- Many databases store dates as datetime values, so you may need to include a time part in the value you provide in the parameter.

  The time part is based on a 24 hour clock, so 10:18 pm is specified as 22:18:00.

Example Date parameters

The following example query strings use a date field called **Order Date**. As in the previous examples of this article, you would add these to the base URL of your view.

- If the Order Date field type includes only the date (with no time of day), and you want to show data only for July 8, 2015, the query string would look something like this:

  ?Order%20Date=2015-07-08

- If Order Date includes the time part, to filter on July 8, 2015 at 10:18 pm, the query string
might look like this:

?Order%20Date=2015-07-08%2022:18:00

- If Order Date is only the date, and you want to filter on multiple dates, you would use commas, as described earlier in this article. For example:


Parameters as DATEPART filters

To filter by date part, use the same nomenclature as in the default Tableau Desktop date hierarchy. For more information, see Date Functions in the Tableau Help.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>year(Order%20Date)</td>
<td>Integer</td>
</tr>
<tr>
<td>quarter(Order%20Date)</td>
<td>Integer between 1 and 4</td>
</tr>
<tr>
<td>month(Order%20Date)</td>
<td>Integer between 1 and 12</td>
</tr>
<tr>
<td>day(Order%20Date)</td>
<td>Integer between 1 and 31</td>
</tr>
<tr>
<td>hour(Order%20Date)</td>
<td>Integer 0–23</td>
</tr>
<tr>
<td>minute(Order%20Date)</td>
<td>Integer 0–59</td>
</tr>
<tr>
<td>second(Order%20Date)</td>
<td>Integer 0–59</td>
</tr>
<tr>
<td>week(Order%20Date)</td>
<td>Integer 1–53</td>
</tr>
<tr>
<td>my(Order%20Date)</td>
<td>Six-digit integer: YYYYMM</td>
</tr>
<tr>
<td>mdy(Order%20Date)</td>
<td>Eight-digit integer: YYYYMMDD</td>
</tr>
</tbody>
</table>

Control Load Order for Multiple Embedded Views

You can control the order in which multiple views load for the people working with your views. This feature can be accessed only using embed code that relies on the Tableau JavaScript file.
In the following example, two views are embedded. The second view loads first, followed by the top view. If you embed multiple views and give them all the same load order value, or if you don't specify load order parameters, they are loaded in the order in which they appear on the page.

**Script Tag Example**

```html
<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'></script>
<object class='tableauViz' width='600' height='400' style='display:none;'><param name='host_url' value='http://myserver/' />
<param name='site_root' value=''/>
<param name='name' value='MyCoSales/TopPerformers'/>
<param name='tabs' value='yes'/>
<param name='toolbar' value='yes'/>
<param name='filter' value='Salesperson=Top 5'/>
<param name='load-order' value='0'/>
</object>
</script>

```html
<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'></script>
<object class='tableauViz' width='600' height='400' style='display:none;'><param name='host_url' value='http://myserver/' />
<param name='site_root' value=''/>
<param name='name' value='MyCoSales/SalesScoreCard'/>
<param name='tabs' value='yes'/>
<param name='toolbar' value='yes'/>
<param name='load-order' value='-1'/>
</object>
</script>

**Embed Dashboards**

You can embed a Tableau dashboard in your own web site or wiki page and you can have it display in different layouts based on the size of the IFrame.
To automatically make the right layout appear for your users, regardless of which device they are using, create device-specific layouts for your dashboard in Tableau Desktop. When you do, Tableau Server and Tableau Online automatically display the correct layout based on IFrame size, provided you use 100% for the IFrame's width and height instead of exact pixel values (see below).

To always make a certain layout appear, regardless of IFrame size, use the `device` parameter in your embed code.

In the following example, the embed code displays a dashboard that's 800 pixels wide by 600 pixels high. Exact width and height values are part of the default embed code you get when you click the Share button at the top of a view or dashboard:

```html
<script type='text/javascript' src='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder' style='width: 800px; height: 600px;'></div>
<object class='tableauViz' width='800' height='600' style='display:none;'><param name='host_url' value='http://mysite.myserver.com'/><param name='site_root' value=''/><param name='name' value='ProfitAnalysis/Sales_Dashboard'/><param name='tabs' value='yes'/><param name='toolbar' value='yes'/><param name='filter' value=':original_view=yes'/></object>
```

In this example, where the dashboard uses device-specific layouts, the style attributes for the `div` class have been removed and the object class values for width and height have been replaced with 100%. In most cases, the correct layout displays; that is, if width and height aren't also being controlled elsewhere in your CSS:

```html
<script type='text/javascript' src='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder'>
<object class='tableauViz' width='100%' height='100%' style='display:none;'><param name='host_url' value='http://mysite.myserver.com'/>
```

-2715 -
<param name='site_root' value=''/>
<param name='name' value='ProfitAnalysis/Sales_Dashboard'/>
<param name='tabs' value='yes'/>
<param name='toolbar' value='yes'/>
<param name='filter' value=':original_view=yes'/></object>

If the correct layout doesn't display, it may be because the HTML page you're using for embedding has a `<!DOCTYPE html>` tag, and that tag is preventing items in the body of the page from resizing to 100% (see details on Stack Overflow). A workaround is to add the following lines to the body of your HTML page:

```html
<style>
    html, body { height: 100% }
</style>
```

The following example assumes that the embedded dashboard has device-specific layouts. The `device` parameter is set to `phone`. This means that, no matter which device displays the dashboard, the layout created for phones will be the one that's displayed.

```html
<script type='text/javascript' src='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder'>
<object class='tableauViz' width='100%' height='100%' style='display:none;'>
    <param name='host_url' value='http://mysite.myserver.com'/>
    <param name='site_root' value=''/>
    <param name='name' value='ProfitAnalysis/Sales_Dashboard'/>
    <param name='device' value='phone'/>
    <param name='tabs' value='yes'/>
    <param name='toolbar' value='yes'/>
    <param name='filter' value=':original_view=yes'/></object></div>
```

**IFrame dimensions and device layouts**

The dashboard layout a device displays is based on the smallest dimension (height or width) of the iframe in which the Tableau view appears. Sometimes Desktop, Tablet, or Phone layouts
may appear on other types of devices. For example, a Tablet layout may appear on a desktop computer if the display or browser window is small.

<table>
<thead>
<tr>
<th>If the smallest iframe dimension is ...</th>
<th>This device layout appears ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 pixels or less</td>
<td>Phone</td>
</tr>
<tr>
<td>Between 501 and 800 pixels</td>
<td>Tablet</td>
</tr>
<tr>
<td>Greater than 800 pixels</td>
<td>Desktop</td>
</tr>
</tbody>
</table>

For details on how to create a dashboard that uses device-specific layouts, see Create Dashboard Layouts for Different Device Types on page 2302.

Embed Code for Custom Views

When you embed a custom view of a workbook or sheet, the default view is determined by these factors:

- If the embed code URL specifically refers to a custom view, that view is displayed by default.
- If the embed code URL does not refer to a custom view, the Default custom view is displayed by default.
- If no Default custom view has been defined, the original view is displayed by default.

**Note:** To ensure the original view will be displayed by default in an embedded view, make sure the embed code URL for the name parameter does not explicitly refer to a custom view, and include the following filter parameter in the embed code:<param name='filter' value=':original_view=yes'/>

In the following example, the embed code will always display the original view of the Profit Analysis sheet in the Profit Analysis workbook, because the filter parameter is set to :original_yes, and the name parameter does not refer to a specific custom view in the URL for the sheet.

```html
<script type='text/javascript' src='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder' style='width: 1496px; height: 749px; '>
<object class='tableauViz' width='1496' height='749'
```
In this example, the setting for the name parameter in this example specifically refers to the URL for a custom view named Furniture (in the Profit Analysis sheet in the Profit Analysis workbook).

In this example, the name parameter does not refer to a specific custom view in the URL for the sheet, and the original_view parameter has not been specified. The embed code here will display the custom view that has been set to Default in the Profit Analysis sheet in the Profit Analysis workbook. However, if the original view is still the Default (no other custom view has been set to Default), then the original view will be displayed as the default view.
Embed Views into Wikis

You can easily embed a view into a wiki or other web page simply by putting the view inside an `<iframe>` tag.

1. Navigate to the wiki page you want to embed a view into.

2. Edit the page and add an `<iframe>` where the source is the URL from the Email box of the Share View dialog box. For example:

   `<iframe src="http://myserver/views/Date-Time/DateCalc-s?:embed=yes&:toolbar=no" width="800" height="600"></iframe>`

3. Save your changes.

   If you use Tableau Server and both it and the wiki are configured to use Active Directory to automatically authenticate users, they will immediately see the view. Otherwise, users will be asked to sign in before they can see the view.
Embed Images of Tableau Server Views

In addition to embedding a view into a `<script>` or `<iframe>` tag you can also embed the view as an image. When you embed an image the view is not interactive; however, it is updated every time the page fully reloads, showing the latest data.

**Note:** This approach works only if users accessing the embedded image have an active web browser session with Tableau Server and are automatically signed in using Active Directory.

1. Navigate to the page where you want to embed the image.

2. Edit the page and add an `<img>` tag where the source is the URL from the Email box of the Share View dialog box for the view, plus the .png file extension. For example:

   ```html
   <img src="http://tableauserver/views/Date-Time/DateCalcs.png" width="900" height="700">
   ```

Embed Tableau Server Views into SharePoint (Active Directory Authentication)

If your Tableau Server is configured to use Active Directory and automatically authenticate users, they can see views you embed in SharePoint pages with the Tableau web part.

If your Tableau Server instead uses Local Authentication to authenticate users, see Embed Tableau Server Views into SharePoint (Local Authentication) on page 2723. If you don't know which type of authentication your Tableau Server uses, ask your administrator.

In this article

Requirements below
Embedding a View into SharePoint on the next page

Requirements

**Licensed users:** Anyone who accesses an embedded view must be a licensed user on Tableau Server.
**SharePoint version:** Starting with Tableau Server 8.1, you must use SharePoint 2013 to embed Tableau Server views in SharePoint pages. SharePoint 2013 uses Microsoft .NET Framework version 4.5, which meets Tableau Server’s security requirements.

**TableauEmbeddedView web part:** You must have a TableauEmbeddedView web part deployed to your SharePoint server before you can embed Tableau views in a SharePoint page. For sample SharePoint code and instructions for how to create a web part and deploy it to your SharePoint server, see `C:\Program Files\Tableau\Tableau Server\<version>\extras\embedding\sharepoint`. **Note:** The sample SharePoint code is provided as an example, and may require modification to work in your SharePoint deployment.

**Embedding a View into SharePoint**

You can embed the Tableau web part in a new or existing SharePoint page.

1. Open the page where you want to embed a view and switch to edit mode.
2. In the section of the page where you want to embed the view, on the **Insert** tab, click **Web Part**.
3. Under Categories, in the **Custom** (or **Miscellaneous**) folder, select **TableauEmbeddedView**, and then click **Add** in the lower-right corner.
4. Select the TableauEmbeddedView web part, click the drop-down arrow, and then select **Edit Web Part**.
5. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part.

- In **Tableau Server Name**, enter the name of your Tableau Server. You do not need to enter "http://" before the Tableau Server name.
- In **View Path**, enter the path to the view you want to embed.
- Specify whether you want to show the toolbar, use Trusted Authentication, use SSL, or if you want to embed the view as an image instead of as an interactive view.
- In the **Appearance** section you can specify a **Title** for the web part, the **Height**, **Width**, **Chrome State**, and **Chrome Type**. In general you should specify a fixed
height (for example, 700 Pixels) and adjust the width to fit the zone.

6. Click **OK** to apply the changes and exit edit mode.

The view will be embedded into the web part that you just created. To see the view, your users will be automatically authenticated using Active Directory.

**Embed Tableau Server Views into SharePoint (Local Authentication)**

If your Tableau Server uses Local Authentication to authenticate users, there are some extra steps you need to take so they can see views embedded in SharePoint pages.
If your Tableau Server instead uses Active Directory to authenticate users, see Embed Tableau Server Views into SharePoint (Active Directory Authentication) on page 2720. If you don’t know which type of authentication your Tableau Server uses, ask your administrator.

In this article

Requirements below
Edit Security Permissions for TableauEmbeddedView.dll below
Install and Deploy TableauEmbeddedView.wsp on page 2726
Verify the Web Part’s Deployment on page 2727
Embed a View Using the Tableau Web Part on page 2728

Requirements

Users: To access an embedded view, users must be licensed Tableau Server users and their user name on SharePoint must be the same as their user name on Tableau Server.


Edit Security Permissions for TableauEmbeddedView.dll

Edit the security permissions for TableauEmbeddedView.dll so that all users of the operating system can use it.

1. Locate the TableauEmbeddedView.dll and TableauEmbeddedView.wsp files that install with Tableau Server. If Tableau Server is installed on drive C, the files will be in the following directory:

   C:\Program Files\Tableau\Tableau Server\2018.2\extras\embedding\sharepoint\n
2. Copy the files to the root directory of your SharePoint server. The root directory is usually located at C:\Inetpub\wwwroot\wss\VirtualDirectories\<port>\bin, for example:

   C:\Inetpub\wwwroot\wss\VirtualDirectories\80\bin
3. To edit the security permissions on TableauEmbeddedView.dll, right-click **TableauEmbedded.dll** and then select **Properties > Security**.

4. Under **Group or user names**, select **Everyone**, and then click **Edit**.

5. Under **Permissions for Everyone**, for the **Full control** permission, select **Allow**.
6. Click OK.

Install and Deploy TableauEmbeddedView.wsp

The TableauEmbeddedView.wsp file gives SharePoint more information about what to do with the .dll file. You copied the TableauEmbeddedView.wsp file to the SharePoint root directory in the previous procedure. To install and deploy the .wsp file, follow these steps:

1. Open SharePoint 2013 Management Shell and enter the following command:

   Add-SPSolution -LiteralPath "C:\Inetpub\wwwroot\wss\VirtualDirectories\80\bin\TableauEmbeddedView.wsp"

2. On the SharePoint Central Administration home page, click System Settings.

3. In the Farm Management section, click Manage farm solutions.

4. On the Solution Management page, click the solution that you want to deploy.

5. On the Solution Properties page, click Deploy Solution.
6. On the Deploy Solution page, in the **Deploy When** section, select one of the following options:
   - **Now**
   - **At a specified time.** Specify a time by using the date and time boxes.

7. In the **Deploy To?** section, in the **A specific web application** list, click **All web applications** or select a specific Web application, and then click **OK**.

8. Open your SharePoint site. Click the settings icon, and then select **Site settings**.

9. Under Site Collection Administration, click **Site collection features**.

10. Scroll to the TableauEmbeddedView feature and then click **Activate** to activate the feature.

**Verify the Web Part's Deployment**

In the following procedure, you will verify that the Tableau web part is installed.

1. Open your SharePoint site in a web browser.
   It may take a few moments for the site to appear.

2. Click the settings icon, and then select **Site settings**.

3. Under **Web Designer Galleries**, click **Web parts**.
4. Confirm that TableauEmbeddedView.webpart is listed.

Embed a View Using the Tableau Web Part

You can embed the Tableau web part in a new or existing SharePoint page.

1. Open the page where you want to embed a view and switch to edit mode.

2. In the section of the page where you want to embed the view, on the Insert tab, click Web Part.

3. Under Categories, in the Custom (or Miscellaneous) folder, select TableauEmbeddedView, and then click Add in the lower-right corner.
4. Select the TableauEmbeddedView web part, click the drop-down arrow, and then select **Edit Web Part**.

5. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part.
   - In **Tableau Server Name**, enter the name of your Tableau Server. You do not need to enter "http://" before the Tableau Server name.
• In **View Path**, enter the path to the view you want to embed.

• Specify whether you want to show the toolbar, use Trusted Authentication, use SSL, or if you want to embed the view as an image instead of as an interactive view.

• In the **Appearance** section you can specify a **Title** for the web part, the **Height**, **Width**, **Chrome State**, and **Chrome Type**. In general you should specify a fixed height (for example, 700 Pixels) and adjust the width to fit the zone.

6. Click **OK** to apply the changes and exit edit mode.
Now the view is embedded in the page and users who access it will be automatically signed in based on their user name and password for SharePoint.

This is an example of embedding views into SharePoint using the provided .dll file. You can also embed views into other types of web applications. For more information, see the JavaScript API on the Tableau Developer Portal.

Link to a PNG, PDF, or CSV of a View

If you work with a website or application that doesn't support embedded, interactive Tableau views, you can link to PNG or PDF versions of them.

If you're familiar with scripting, you can also use links to automatically convert multiple views into PNGs, PDFs, or even CSVs. The resulting files can be shared with people who lack Tableau Online or Tableau Server accounts, incorporated into presentations, or archived for future reference.

When you link to any of these formats, they always load the latest data available on the server. But be aware that CSVs of dashboards link to only one sheet—the one whose title is first in alphabetical order.

1. With the help of a Tableau content owner or site administrator, ensure that your audience has permission to access the content. (If you're using a script to process multiple files, only you need access.)
   - For links to PNG and PDF files, users need the Download Image/PDF permission.
   - For links to CSVs, users need the Download Full Data permission.

2. Replace the end of a view's browser URL with the appropriate file extension.
   For example, change
   
   http://<servername>/#/views/<workbook>/sheet?iid=7 to
   http://<servername>/#/views/<workbook>/sheet.png

3. If you want to filter a view, add a question mark after the file extension, followed by URL parameters that reflect the view's data structure.
   For example, change
   
   http://<servername>/#/views/<workbook>/sheet.png to
Download Views or Workbooks

**Note:** The download formats available to you depend on permissions granted by Tableau content owners and site administrators.

1. At the top of a view, click **Download**. Or, click the download button wherever it appears on the page.

2. Select one of the following options:
   - **Image**: Downloads an image of the view in .png format.
   - **Data**: Opens a new tab in the browser window and displays the view's data in summary and detail. You can then download the data as a comma-separated value (.csv) file.
     When downloading from a dashboard, first click the specific sheet with data you want.
   - **Crosstab**: Downloads the view, or the selected sheet in a dashboard, as a .csv file you can open in Microsoft Excel.
   - **PDF**: Downloads a PDF of the view, specific sheets from a dashboard, or specific sheets from a workbook. Under **Include**, select the part of the workbook you want to download. Select specific sheets or select all. Select **Scaling** to control the image's appearance on the PDF. Select **Paper Size** and **Orientation**.
     If you’re downloading a dashboard to PDF format, web page objects aren’t included.
   - **Tableau Workbook**: Downloads a workbook you can open with Tableau Desktop.
     You have the option of downloading the workbook to different versions of Tableau or keeping the workbook in its current version. For example, if the workbook needs

http://<servername>/#/views/<workbook>/sheet.png
?Region=South&Department=Sales
to be opened in Tableau Desktop 10.5, under Version select **Tableau 10.4** > **Download**. For information about version compatibility, see **Make Workbooks Compatible Between Versions** on page 2509.

As an alternative, you can select **Server > Open Workbook** in Tableau Desktop.

**Note:** Downloading extremely large amounts of data can affect server performance and might not complete successfully. If you encounter these issues, try exporting the data directly from the underlying data source.

**Web Authoring and Tableau Desktop Feature Comparison**

For anyone familiar with Tableau Desktop and new to the web authoring environment in Tableau Server and Tableau Online, this topic provides a summary of the web features that you use similarly to the way you do in Tableau Desktop. It also lists some fundamental differences between the two environments.

**Note:** This topic does not cover every difference between the desktop and web environments, but summarizes core authoring functionality.

**In this article**

- **Web authoring capabilities** below: **Data management** on the next page, **Analytics** on page 2735, **Filtering and sorting** on page 2736, **Formatting** on page 2737
- **General differences in web authoring** on page 2738
- **Features listed by version** on page 2738
- **Related links** on page 2738

**Web authoring capabilities**

In the web environment, you can connect to data and create workbooks from those data sources, or data published through Tableau Desktop. You can edit views created on the web or
published from Tableau Desktop.

Administrators can set at the site level what web authoring abilities users can have. Explorers can edit workbooks, create new workbooks from published data sources, connect to published data sources, and create and edit views, dashboards, and stories. Creators have those same capabilities, but can also create new workbooks, connect to data on the web, and use Dashboard Starters to quickly dive into analysis.

Data management

- **Creators**: Connect to data sources, upload files (text and Excel), or use pre-built Dashboard Starter templates for Eloqua, Salesforce, ServiceNow, and Marketo data. For more information, see [Creators: Connect to data on the web](#).

- **Creators**: Prepare data on the web in the Data Source page. For more information, see [Creators: Prepare Data on the Web](#).

**Note**: There is a limitation on the number of rows that can be viewed in the Data Source page when authoring data on the web, determined by browser:

- Internet Explorer: 10,000 rows
- Other browsers: 100,000 rows

Independent of browser, the total number of records (rows by columns) that can be viewed in the Data Source page on the web is **3 million**.

- Join data from different tables in the same data source or from different databases using a multi-connection data source
- Union data
- Pivot data
- Copy values in a grid (Ctrl+C, or Command-C on a Mac)
- **Edit Data Sources**
- Clean data using the Data Interpreter

- **Explorers**: Connect to published data sources.
- Blend published data sources.
- Save a data source (embedded in a published workbook) as a separate, published data
source.

- Change aggregation of measures in the view. Change the default aggregation of measures in the Data pane.
- Search for fields in the schema.
- Duplicate, hide, or rename fields.
- Change the data type of fields.
- Convert measures to dimensions or vice versa.
- Convert a discrete field to continuous and vice versa. This option is available for measures and date dimensions.
- Assign a geographic role for a field.
- Create aliases for members of dimensions.
- Create and edit groups.
- Use sets (not create or edit).
- Use parameters (not create or edit).

Analytics

- Create, edit, rename, duplicate, and clear sheets (views, dashboards, and stories) in a workbook.
- Search for fields in the Data pane with schema search
- Drag fields to the view, Rows, Columns, and different mark types in the Marks card.
- Use Show Me to create views. Also, select and drag dimensions and measures of interest to the view area to automatically create a "Show Me" view.
- View underlying data (via tooltips).
- Viz in Tooltip works in web views, but must be configured in Tableau Desktop. Viz in Tooltip worksheets can be hidden, the same way you would hide worksheets used in stories or dashboards.
- Create and edit calculated fields.
• Create bins from continuous measures, and edit bins.
• Create and edit table calculations, and use quick table calculations.
• Use the **Analytics** pane to drag reference lines, trend lines, and other objects into the view. Edit reference lines, trend lines, and bands. Create and configure reference distributions on a continuous axis.
• Create groups by selecting marks in the view and then clicking Group Members (paperclip) in the tooltip for that selection. You can also edit existing groups in the Data pane.
• Create hierarchies by dragging one dimension onto another in the Data pane.
• Change options for interacting with maps, including enabling or disabling pan and zoom, or showing map search, the view toolbar, or map scale. Users can also map units.
• Drill up and down a continuous hierarchy in the view. In a view with a continuous hierarchy, hover near the headers on a continuous axis to display the + and - controls. Click to drill down or up.
• Show labels, totals, and subtotals.
• Show and hide titles and captions.
• Show and hide cards for filters and highlighters.
• Show, hide, and resize headers in the view.
• Swap X and Y axes. Resize axes in the view.
• Change the view size.
• Show and hide the View Toolbar for any view or dashboard.

Filtering and sorting

• Use data highlighting.
• Add, modify, and remove filters (shown as **Quick Filters**), and edit a quick filter layout. (**General**, **Wildcard**, **Condition**, and **Top** tabs are not available.)
• Filter across published data sources.
• Apply table calculation filters to totals in the view.
• Show hidden fields, and exclude or remove fields from the view.

• Sort fields in the view in ascending or descending order. Access the Sort dialog box by right-clicking a dimension on the Rows or Columns shelf. Nested sorting on dimension values within the context of each pane.

---

Formatting

• Resize the width of row headers and the height of column headers.

• Edit workbook formatting, including formatting lines.

• Edit worksheet and dashboard titles.

• Edit axes (double-click an axis in the view). Other options available: Synchronize dual axes, clearing the axis range (Reset), and tick mark settings. Enable or disable Dual axis in a field context menu (right-click a measure field on Rows or Columns shelf). Logarithmic scales can be positive or symmetric (includes 0 and negative values).

• Create, edit, and remove annotations. (Formatting and moving annotations not yet supported.)

• Add and edit text boxes in dashboards.

• Change the color palette. For categorical fields you can assign specific colors and custom colors (using a hex code) to data items. For continuous fields, you can set custom colors for start and end colors (using a hex code).

• Edit number formatting (decimal places, percentage, thousands separator, units, and currency).

• Edit and view device-specific dashboard layouts.

• Add objects, including horizontal and vertical layout containers, to a dashboard.

• Set a dashboard item's exact size, position, and spacing.

• Add padding, borders, and background colors around items in dashboards.

• Select a background map in map views.

• Legends per measure. If you create separate color legends for the measures in your view, Tableau assigns the default color palette to each new color legend. To change the
color legend for each measure, click the drop-down arrow on the color legend to open the **Edit Colors** dialog box and select the palette that you want to use. For more details, see **Legends per measure**.

---

**General differences in web authoring**

- Your authoring capabilities are determined by your license level. For an overview of what you can do with each license level, see [What can I do with a Tableau site?](#).

- You can access right-click menu actions on Measures and Dimensions in the view, but not on every item in the workspace.

- Keyboard shortcuts for web authoring and Tableau Desktop are not the same. For a list of web authoring keyboard shortcuts, see [Shortcuts for web authoring](#).

---

**Features listed by version**

For a list of the latest web editing features to be added to each release, see the web authoring sections in [What's New in Tableau](#) and [What's New in Tableau Online](#).

For a list of features compared by version, see the [Tableau Desktop vs Tableau Web Editor](#) viz on Tableau Public. This information is curated and maintained by Andrew Pick of the [Information Lab](#), a Tableau Gold partner in the UK.

**Disclaimer:** Clicking this link will take you away from the Tableau website. Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact The Information Lab if you have questions regarding its content.

---

**Related links**

[What can I do with a Tableau site?](#)

[Set a Site’s Web Authoring Access](#)
Getting Started with Web Authoring

Creators: Connect to Data on the Web

Creators: Prepare your Data on the Web

Build Views on the Web

Grant Web Edit, Save, and Download Permissions
Reference

Upgrade Tableau Desktop, manage your license, learn how to optimize workbook performance, find out how to connect to a particular data source, and more.

Upgrade Tableau Desktop

This section includes information on upgrading from a previous or a beta version, and on how to turn the product update feature off or on. For more information about how to upgrade Tableau Desktop, see the Upgrade Tableau Desktop article in the Tableau Desktop Deployment Guide.

Upgrade from a Previous or Beta Version

When you upgrade to Tableau 2018.2 from previous versions, your repository is upgraded. Any bookmarks, workbooks, and data sources that you had in your old repository will still be accessible by the application. In addition, the new sample data sources and workbooks will replace the old samples unless you have modified them and saved them as your own.

**Note:** When you download a new version of Tableau Desktop, the product update installers are downloaded to the Downloads/TableauAutoUpdate folder. If the Downloads folder doesn't exist, the installers are downloaded to the TEMP/TableauAutoUpdate folder.

If you have participated in the Tableau Software Beta program, you also have a beta repository. While this folder will still exist after you install Tableau 2018.2, the application will no longer access it. To make your beta workbooks accessible in Tableau 2018.2, copy the workbooks from the beta repository to your new 2018.2 repository.

Turn Product Updates Off or On

To ensure that you always have the most up-to-date features, security resolutions, and corrected issues, Tableau Desktop includes a product update feature. When you start Tableau Desktop, product updates prompts you to download an updated maintenance version of Tableau Desktop, if one exists. The update downloads immediately and then installs when you exit Tableau.
You can also choose to postpone or skip the update. If you do this, you can always check for product updates at any time by selecting Help > Check for Product Updates.

For more information about how to turn product updates off or on, see the Control Product Updates article in the Tableau Desktop Deployment Guide.

**Note the following:**

- The product update installers are downloaded to the Downloads/TableauAutoUpdate folder. If the Downloads folder doesn't exist, the installers are downloaded to the TEMP/TableauAutoUpdate folder.
- Updates are not downloaded and installed on your computer if your Product Maintenance has expired. For more information, see the Product Maintenance FAQ.
- You may not be prompted for product updates. There are a number of reasons why this may be so. For details, see the Troubleshoot maintenance updates section in the Troubleshoot Tableau Desktop Installation article in the Tableau Desktop Deployment Guide.

**Turn off product updates**

Product updates is on by default. You can turn off product updates from the Help menu, or on Windows, by running the installer.

**Use the Help menu**

Select Help > Settings and Performance > Enable Automatic Product Updates and clear the check box.

**Run the installer (Windows only)**

To turn off product updates on Windows, run the Tableau Desktop installer, and then click Customize. In the Custom setup dialog box,

1. Clear the Check for Tableau product updates check box.
2. Click Install.

To turn product updates on again, run the installer, click Customize, and select the check box.

**Administrators control product updates**

As an administrator, you can turn product updates off or on for your users. You can also determine the Tableau Desktop version that your users will update to. Rather than having users
update to the version of their choice (or choose not to update), you can make sure that your users update to the version you choose. For more information, see the Control Product Updates article in the Tableau Desktop Deployment Guide.

Maintain Licenses for Tableau Desktop and Tableau Prep

**Important:** The Tableau licensing service was moved to a new data center on October 6, 2018. This means that any environments that required special configuration (static IP safe listing for example) to access licensing.tableau.com or licensing.tableausoftware.com will need to be updated before you can activate, refresh, or deactivate a Tableau product key. For more information, see Tableau Community.

Tableau Desktop and Tableau Prep can be licensed under a term license model. Term licenses, also known as subscription licenses, allow you to use and update Tableau Desktop and Tableau Prep for a specified period of time.

Term licenses must be renewed to continue providing uninterrupted service. You can continuously renew the term license as each specified period expires. If you don’t renew your term license and the term expires, Tableau will stop working and you will no longer have access to the software.

**Note:** Trial licenses for Tableau Desktop or Tableau Prep expire after a set period of time, usually 14 days. After the trial period expires, you’ll need to purchase a license to continue using the product.

View data about your license

After you install Tableau Desktop or Tableau Prep open the application and then navigate to Help > Manage Product Keys from the top menu to see information about the type of license you have and when it expires.

You can also activate or deactivate a product license key or refresh a maintenance license key from this dialog.
Note: Tableau offers term licenses that provide a range of capabilities. The type of license that you have is displayed in the Product field. For more information about the different type of user-based licenses that are available, see User-based licenses in the Tableau Server help.

Existing Tableau Desktop users may have a perpetual (permanent) license. Perpetual licenses don't expire. However, to get access to product updates and technical support you must purchase Support and Maintenance services. These services must be renewed to continue receiving the service. Perpetual (permanent) licenses are no longer available for Tableau Desktop.

![Manage Product Keys](image)

Use the following buttons to take action on your license key:

- **Refresh**: Click the Refresh button to refresh a maintenance license that is expiring, then close and restart Tableau Desktop. If the Maintenance Expires date doesn't update, check with your license administrator as the key or maintenance agreement may have changed.

  To refresh a maintenance key from the command line see Refresh the product key in the Tableau Desktop and Tableau Prep Deployment guide.

- **Deactivate**: Select a license key in the list then click Deactivate to deactivate the license. Deactivate a license if you need to move the license to another computer or no longer need the license on this computer.

  For more information about deactivating a license, see Move or Deactivate License Keys in the Tableau Desktop and Tableau Prep Deployment guide.

- **Activate**: After Tableau Desktop or Tableau Prep is installed, click Activate to open the activation dialog and enter your product key. If you get an error and can't activate Tableau Desktop or Tableau Prep using your license key, contact Tableau Support.
For more information about activating a license key, see Activate Tableau Desktop or Activate and Register Tableau Prep in the Tableau Desktop and Tableau Prep Deployment guide.

Track Tableau Desktop license usage and expiration data

If you want to track and view license usage and expiration data for Tableau Desktop in Tableau Server you must configure Tableau Desktop to send license data to Tableau Server on a set interval, and then enable reporting on Tableau Server.

This enables server administrators to access two reports:

- **Desktop License Usage**: This report lets server administrators see usage data for Tableau Desktop licenses in your organization.

- **Desktop License Expiration**: This report gives server administrators information about which Tableau Desktop licenses in your organization have expired or need maintenance renewal.

If Tableau Desktop and Tableau Server are configured for license reporting, when signed in to Tableau Server as an Administrator, you will see these two reports listed in the Status tab in the Analysis section.
If you don’t see these reports listed, then Tableau Desktop and Tableau Server may not be configured for Tableau Desktop usage reporting.

For information about how to configure Tableau Desktop and Tableau Server for usage reporting, see Manage Tableau Desktop License Usage in the Tableau Desktop and Tableau Prep Deployment guide.

Additional resources

For more information about managing your license refer to the following topics:

- To find your product key and activate Tableau Desktop see Install Tableau Desktop.
- To find your product key and activate Tableau Prep, see Install Tableau Prep.
- To deactivate a product key or move it to another computer, see Move or Deactivate Tableau Desktop.
- To learn more about licensing for non-persistent virtual desktops or for computers that are regularly re-imaged, see Configure Virtual Desktop Support.
To learn more about license management for Tableau Server or Tableau Online, see Licensing Overview (Linux | Windows)

Glossary

A

action
An interaction that you can add to your views. There are three types of action: Filter, Highlight, and URL.

ad-hoc calculation
A calculation that you can create and update as you work with a field on a shelf in the view. Also known as type-in calculation or in-line calculation.

aggregation
A result of a mathematical operation applied to a measure. Predefined aggregations include summation and average. You can convert dimensions to measures by aggregating them as a count. For relational data sources, all measures must be either aggregated or disaggregated (unless they appear on the Filters shelf). Tableau aggregates measures, usually as a summation, when you place them on a shelf. For multidimensional (OLAP) data sources, aggregations are defined when the cube is created and cannot be modified in Tableau.

alias
An alternative name that you can assign to a field or to a dimension member.

Analytics pane
A pane on the left side of your workbook that provides quick and easy access to common analytic features in Tableau. From the Analytics pane,
you can drag reference lines, box plots, trend lines forecasts, and other items into your view. Toggle between the Data pane and the Analytics pane by clicking one of the tabs at the top of the side bar.

**bin**
A user-defined grouping of measures in the data source.

**blending data**
The process of combining data from different data source types in a view. The first data source that you use in the view becomes the primary data source. The remaining data sources become the secondary.

**bookmark**
A .tbm file in the Bookmarks folder in the Tableau repository that contains a single worksheet. Much like web browser bookmarks, .tbm files are a convenient way to quickly display different analyses.

**calculated field**
A new field that you create by using a formula to modify the existing fields in your data source.

**canvas**
An area at the top of the Data Source page where you can drag tables or select queries or cubes to set up your data source. See also: Data Source page.
**caption**
A description of the current view on the active worksheet. For example, “Sum of Sales for each Market”. You can automatically generate captions or create your own custom captions. Show and hide the caption by selecting Worksheet > Show Caption.

**cell**
A basic element of any table that you create in Tableau. You can control cells to enhance your data view, which is useful for text tables and heat maps.

**color legend**
An area of the view that displays the colors associated with a measure or dimension member. The default legend is modified when you place a dimension or a measure on the Color property.

**Color property**
A property on the Marks card that enables you to encode data by assigning different colors to the marks in a view. The property accepts measures and dimensions. When you place a dimension on the Color property, Tableau separates the marks according to the members in the dimension, and assigns a unique color to each member. When you place a measure on the Color property, Tableau draws each mark with a different color using a continuous range. In both cases, a legend describes the color encoding.

**Columns shelf**
A shelf at the top of the workbook that you use to create the columns of a data table. The shelf accepts any number of dimensions and measures. When you place a dimension on the Columns shelf, Tableau creates headers for the members of that dimension. When you place a measure on the
Columns shelf, Tableau creates quantitative axes for that measure. See also Rows shelf.

cross-database join
A join that contains tables from two or more connections to different data-bases.

crosstab
A text table view. Use text tables to display the numbers associated with dimension members.

cube
A data source that is connected to a multidimensional database. Also known as multidimensional data source. For example, data sources that connect to Microsoft Analysis Services or Oracle Essbase are called cubes.

custom geocoding
A process of adding your own location data to extend the built-in geocoding.

dashboard
A combination of several views arranged on a single page. Use dashboards to compare and monitor a variety of data simultaneously.

data grid
An area at the bottom of the Data Source page where you can review the fields and the first 1,000 rows of the data in the data source. You can also use the data grid to make general modifications to your data source, such as adding a calculation, hiding or renaming a field, or changing its data type. See also: Data Source page
**Data Interpreter**
A tool that parses your Excel or Google Sheets data source to help prepare your data for analysis.

**Data pane**
A pane on the left side of the workbook that displays the fields of the data sources to which Tableau is connected. The fields are divided into dimensions and measures. The Data pane also displays custom fields such as calculations, binned fields, and groups. You build views of your data by dragging fields from the Data pane onto the various shelves that are a part of every worksheet.

**data source**
The link between your data and Tableau. A Tableau data source contains information about how to connect to your data, table names, the relationships the tables have with each other, and any customizations that you make on top.

**Data Source page**
A page where you can set up your data source. The Data Source page generally consists of four main areas: left pane, canvas, data grid, and metadata grid.

**data view**
see: view

**Detail property**
A property on the Marks card that you can use to separate the marks in a view according to the level of detail (that is, members) of a dimension. The Detail property works only on aggregated data.
**dimension**
A field of categorical data. Dimensions typically hold discrete data such as hierarchies and members that cannot be aggregated. Examples of dimensions include dates, customer names, and customer segments. See also: measure.

**encoding**
A visual representation of your data. You can encode your data by color, shape, size, and path using the associated worksheet shelves.

**extract**
A saved subset of a data source that you can use to improve performance and analyze offline. You can create an extract by defining filters and limits that include the data you want in the extract.

**extract mode**
A connection state. In extract mode, a snapshot of your data is taken, which then becomes the Tableau data source. See also: live mode

**field**
A dimension or a measure in a database. For relational data sources, fields are the columns of a table. For cube (multidimensional) data sources, fields are the dimensions of a cube. Each dimension or column contains a unique attribute of the data, such as customer name, sales, or product type.
field label
A row or column heading that indicates the data field used to create the view. For example, a view that has rows for East, Central, and West might have a Region field label at the top of the column indicating that each row is a member of the Region field.

Filters shelf
A shelf on the left of the workbook that you can use to exclude data from a view by filtering it using measures and dimensions.

forecast
A calculation that predicts future trends based on current trends and data.

Format pane
A pane that contains formatting settings that control the entire worksheet, as well as individual fields in the view. When open, the Format pane appears on the left side of the workbook.

group
A field you can use to combine dimension members into higher level categories. For example, you could group a dimension that contains states into regions. Groups are marked with a paper clip icon in the Data pane.

header
A label for member names for each field that you place on Rows or Columns.
hexagonal binning
   A technique for clustering data in a two-dimensional plane.

in-line calculation
   see: ad-hoc calculation

integrated data source
   see: multi-connection data source

join condition
   A relationship between fields in a join. You can define the relationship on
   the canvas of the data source. See also: joining.

joining
   A way to combine data from multiple tables.

level of detail (LOD) expression
   A syntax that supports aggregation at dimensionalities other than the view
   level. With level of detail expressions, you can attach one or more dimen-
   sions to any aggregate expression.

live mode
   A connection state. In live mode, the Tableau data source fetches your data
   and saves the metadata associated with your data to the Tableau data
   source. See also: extract mode
**LOD**

see: level of detail (LOD) expression

---

**marks**

A part of the view that visually represents one or more rows in a data source. A mark can be, for example, a bar, line, or square. You can control the type, color, and size of marks.

**Marks card**

A card to the left of the view where you can drag fields to control mark properties such as type, color, size, shape, label, tooltip, and detail.

**measure**

A field of quantitative data. Measures are fields that are dependent variables. They are typically quantitative fields or calculated fields like sales, temperature, or frequency. You can also create discrete measures in Tableau. See also: dimension.

**metadata grid**

An area that can be accessed from the Data Source page by clicking the metadata button and that displays the fields in your data source as rows so that you can quickly examine the structure of your data source and perform routine management tasks, such as renaming fields or hiding multiple fields at once. When connected to cube data, the metadata displays by default. See also: Data Source page.

**multi-connection data source**

A Tableau data source that contains two or more connections to tables from different databases.
multidimensional data source
see: cube

P

packaged workbook
A single zip file with a .twbx extension that contains a workbook along with any supporting local file data sources and background images. Use this format to package your work for sharing with others who don’t have access to the data.

Pages shelf
A shelf to the left of the view that you can use to split a view into a sequence of pages based on the members and values in a discrete or continuous field. Adding a field to the Pages shelf is like adding a field to the Rows shelf, except that a new page is created for each new row.

pane
An area in the table that is created by the intersection of rows and columns. Tables consist of one or more panes. The number of panes in a view depends on the number and type of fields placed on the Rows and Columns shelves.

parameter
A dynamic value that can replace a constant value in calculations, filters, and reference lines.

pass-through function
A function that sends SQL expressions directly to the database to access custom database functions.
Path property
A property on the Marks card that you can use to encode data by connecting marks using a particular drawing order. The property accepts measures and dimensions. Dimensions connect the marks according to the members in the dimension. If the dimension is a date, the drawing order is given by the date order. If the dimension contains words, the line is drawn based on the order of the words in the data source. Measures connect the marks according to the values of the measure. The measure can be aggregated or dis-aggregated. See also: path.

pill
A field in the view.

pivot
A way to convert data from crosstab format into columnar format.

primary data source
The first data source that you use in a blended view. See also: blending data, secondary data source.

query
A set of formalized instructions that Tableau uses to communicate with databases. Common query languages include SQL and MDX. Every time you build a view of your data, Tableau translates your actions into queries and retrieves the requested information from the data source. If you are building a dense data view, you can turn queries off until all the fields you want are placed on shelves.
relational database
A database that presents information in tables with rows and columns. Examples of relational databases that Tableau supports are Excel workbooks, Access databases, comma-delimited text files, MySQL database, and Tableau Data Extract files.

Repository
A collection of workbooks, bookmarks, data sources, and logs. By default, the Tableau Repository is located on the drive where Tableau is installed, in the My Documents\My Tableau Repository folder.

Rows shelf
A shelf at the top of the workbook that you can use to create the rows of a data table. The shelf accepts any number of dimensions and measures. When you place a dimension on the Rows shelf, Tableau creates headers for the members of that dimension. When you place a measure on the Rows shelf, Tableau creates quantitative axes for that measure.

secondary data source
A second or subsequent data source that you use in a blended view. See also: blending data, primary data source.

set
A custom field that defines a subset of data based on some conditions. A set can be based on a computed condition or on a specific data point in the view. Sets appear at the bottom of the Data pane in the Sets area.
**Shape legend**
A legend that displays the shapes associated with dimension members. The legend appears on worksheets that have a dimension placed on the Shape shelf.

**Shape shelf**
A shelf to the left of the view that you can use to encode data by assigning different shapes to the marks in the view. The Shape shelf accepts dimensions only. When you place a dimension on the shelf, Tableau separates the marks according to the members of the dimension, and a legend describes the encoding. You cannot place a measure on the shelf because measures do not contain members.

**sheet**
A view (also known as worksheet), dashboard, or story. Sheets appear as tabs at the bottom of the workbook.

**shelves**
Named areas to the left and top of the view. You build views by placing fields onto the shelves. Some shelves are available only when you select certain mark types. For example, the Shape shelf is available only when you select the Shape mark type.

**Size shelf**
A shelf to the left of the view that allows you can use to encode data by assigning different sizes to the marks in the view. The Size shelf accepts measures and dimensions. When you place a dimension on the shelf, Tableau separates the marks according to the members in the dimension, and assigns a unique size to each member. When you place a measure on the shelf, Tableau assigns a different size to each mark using a continuous range.
small multiples
   A view that contains small multiple charts of the same type.

story
   A sheet that contains a sequence of views or dashboards that work together to convey information.

story point
   An individual view in a story.

T


table
   A visual presentation of a data view. Tables consist of panes, headers, and cells.

table calculation
   A computation that uses data from multiple rows in the database and that is applied to the values in the table.

Text shelf
   A shelf to the left of the view that you can use to view the numbers associated with a view, and to encode data by assigning text labels to the marks. The shelf accepts measures and dimensions. The most common view using the Text shelf is a text table.

tooltip
   Data details that appear when you hover over one or more marks in the view.

type-in calculation
   see: ad-hoc calculation
unioning
A way to combine data from multiple tables by appending values (rows).

view
A representation of your data in a Tableau worksheet or dashboard. You can create data views by placing fields on shelves.

workbook
A file with a .twb extension that contains one or more worksheets (and possibly also dashboards and stories).

worksheet
A sheet where you build views of your data by dragging fields onto shelves.
Keyboard Shortcuts

In this article

<table>
<thead>
<tr>
<th>Description</th>
<th>Windows shortcut</th>
<th>Mac shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>New workbook</td>
<td>Ctrl+N</td>
<td>Command+N</td>
</tr>
<tr>
<td>New worksheet</td>
<td>Ctrl+M</td>
<td>Command+T</td>
</tr>
<tr>
<td>Describe sheet</td>
<td>Ctrl+E</td>
<td>Command+E</td>
</tr>
<tr>
<td>Cycle forward through open worksheets</td>
<td>Ctrl+Tab, Ctrl+F6</td>
<td>Shift+Command+Right Bracket</td>
</tr>
<tr>
<td>Cycle backward through open worksheets</td>
<td>Ctrl+Shift+Tab, Ctrl+Shift+F6</td>
<td>Shift+Command+Left Bracket</td>
</tr>
<tr>
<td>Description</td>
<td>Windows shortcut</td>
<td>Mac shortcut</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Switch in and out of Presentation Mode</td>
<td>F7, Ctrl+H</td>
<td>Option+Return</td>
</tr>
<tr>
<td>Switch in and out of Full Screen mode</td>
<td></td>
<td>Control+Command+F</td>
</tr>
<tr>
<td>Open file</td>
<td>Ctrl+O</td>
<td>Command+O</td>
</tr>
<tr>
<td>Save file</td>
<td>Ctrl+S</td>
<td>Command+S</td>
</tr>
<tr>
<td>Revert workbook to last saved state</td>
<td>F12</td>
<td>Option+Command+E</td>
</tr>
<tr>
<td>Close the current workbook</td>
<td>Alt+F4</td>
<td>Command+W</td>
</tr>
<tr>
<td>Print</td>
<td>Ctrl+P</td>
<td>Command+P</td>
</tr>
<tr>
<td>Open Help</td>
<td>F1</td>
<td>Control+Command+Question Mark</td>
</tr>
</tbody>
</table>

**Shortcuts for data (Tableau Desktop)**

<table>
<thead>
<tr>
<th>Description</th>
<th>Windows shortcut</th>
<th>Mac shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect to data source</td>
<td>Ctrl+D</td>
<td>Command+D</td>
</tr>
<tr>
<td>Activate the find command in the Data pane</td>
<td>Ctrl+F</td>
<td>Command+F</td>
</tr>
<tr>
<td>Refresh the data source</td>
<td>F5</td>
<td>Command+R</td>
</tr>
<tr>
<td>Run data updates on a view</td>
<td>F9</td>
<td>Shift+Command+0</td>
</tr>
<tr>
<td>Toggle automatic data updates on and off</td>
<td>F10</td>
<td>Option+Command+0</td>
</tr>
</tbody>
</table>

**Shortcuts for selection tools (Tableau Desktop)**

<table>
<thead>
<tr>
<th>Description</th>
<th>Windows shortcut</th>
<th>Mac shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Windows shortcut</td>
<td>Mac shortcut</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Select all data</td>
<td>Ctrl+A</td>
<td>Command+A</td>
</tr>
<tr>
<td>Copy selected data</td>
<td>Ctrl+C</td>
<td>Command+C</td>
</tr>
<tr>
<td>Use Rectangular Selection tool</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Use Lasso Selection Tool</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>Use Radial Selection Tool</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>Clear the selection</td>
<td>Esc</td>
<td>Esc</td>
</tr>
</tbody>
</table>

**Shortcuts for authoring views (Tableau Desktop)**

<table>
<thead>
<tr>
<th>Description</th>
<th>Windows shortcut</th>
<th>Mac shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Me!</td>
<td>Ctrl+1, Ctrl+Shift+1</td>
<td>Command+1</td>
</tr>
<tr>
<td>Add the selected field to the sheet. (Only works with a single field.)</td>
<td>Enter or double-click</td>
<td>Return or double-click</td>
</tr>
<tr>
<td>Place selected field on Columns shelf</td>
<td>Alt+Shift+C</td>
<td>Option+Shift+C</td>
</tr>
<tr>
<td>Place selected field on Filters shelf</td>
<td>Alt+Shift+F</td>
<td>Option+Shift+F</td>
</tr>
<tr>
<td>Place selected field on Size</td>
<td>Alt+Shift+I</td>
<td>Option+Shift+I</td>
</tr>
<tr>
<td>Place selected field on Detail</td>
<td>Alt+Shift+L</td>
<td>Option+Shift+L</td>
</tr>
<tr>
<td>Place selected field on Color</td>
<td>Alt+Shift+O</td>
<td>Option+Shift+O</td>
</tr>
<tr>
<td>Place selected field on Pages shelf</td>
<td>Alt+Shift+P</td>
<td>Option+Shift+P</td>
</tr>
<tr>
<td>Place selected field on Rows shelf</td>
<td>Alt+Shift+R</td>
<td>Option+Shift+R</td>
</tr>
<tr>
<td>Action Description</td>
<td>Keyboard Shortcuts</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>----------------------------</td>
<td></td>
</tr>
<tr>
<td>Place selected field on Shape</td>
<td>Alt+Shift+S</td>
<td></td>
</tr>
<tr>
<td>Place selected field on Text/Label</td>
<td>Option+Shift+S</td>
<td></td>
</tr>
<tr>
<td>Place selected field on Rows shelf</td>
<td>Alt+Shift+T</td>
<td></td>
</tr>
<tr>
<td>Place selected field on Columns shelf</td>
<td>Option+Shift+T</td>
<td></td>
</tr>
<tr>
<td>Open the Drop Field menu</td>
<td>Right-click+Drag to shelf</td>
<td></td>
</tr>
<tr>
<td>Copy a field in the view and place it on another shelf or card</td>
<td>Ctrl+Drag</td>
<td></td>
</tr>
<tr>
<td>Swap rows and columns</td>
<td>Control+Command+W</td>
<td></td>
</tr>
<tr>
<td>Flip orientation of column labels at bottom of view</td>
<td>Ctrl+L</td>
<td></td>
</tr>
<tr>
<td>Toggle dashboard grid on and off</td>
<td>G</td>
<td></td>
</tr>
<tr>
<td>Toggle between Dashboard and Layout tabs</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Cut text selection (in captions, titles, formulas, etc.)</td>
<td>Ctrl+X</td>
<td></td>
</tr>
<tr>
<td>Paste clipboard</td>
<td>Command+V</td>
<td></td>
</tr>
<tr>
<td>Undo</td>
<td>Ctrl+Z</td>
<td></td>
</tr>
<tr>
<td>Redo</td>
<td>Command+Shift+Z</td>
<td></td>
</tr>
<tr>
<td>Clear the current worksheet</td>
<td>Alt+Shift+Backspace</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Option+Shift+Delete</td>
<td></td>
</tr>
</tbody>
</table>
Shortcuts for resizing rows and columns (Tableau Desktop)

<table>
<thead>
<tr>
<th>Description</th>
<th>Windows shortcut</th>
<th>Mac shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smaller cell size</td>
<td>Ctrl+B</td>
<td>Command+B</td>
</tr>
<tr>
<td>Bigger cell size</td>
<td>Ctrl+Shift+B</td>
<td>Command+Shift+B</td>
</tr>
<tr>
<td>Make rows narrower</td>
<td>Ctrl+Left Arrow</td>
<td>Control+Command+Left Arrow</td>
</tr>
<tr>
<td>Make rows wider</td>
<td>Ctrl+Right Arrow</td>
<td>Control+Command+Right Arrow</td>
</tr>
<tr>
<td>Make columns shorter</td>
<td>Ctrl+Down Arrow</td>
<td>Control+Command+Down Arrow</td>
</tr>
<tr>
<td>Make columns taller</td>
<td>Ctrl+Up Arrow</td>
<td>Control+Command+Up Arrow</td>
</tr>
</tbody>
</table>

Shortcuts for pages (Tableau Desktop)

<table>
<thead>
<tr>
<th>Description</th>
<th>Windows shortcut</th>
<th>Mac shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start or stop forward playback on the Pages shelf</td>
<td>F4</td>
<td>F4</td>
</tr>
<tr>
<td>Start or stop backward playback on the Pages shelf</td>
<td>Shift+F4</td>
<td>Shift+F4</td>
</tr>
<tr>
<td>Skip forward one page</td>
<td>Ctrl+Period</td>
<td>Command+Period</td>
</tr>
<tr>
<td>Skip backward one page</td>
<td>Ctrl+Comma</td>
<td>Command+Comma</td>
</tr>
</tbody>
</table>

Shortcuts for selecting and navigating marks

<table>
<thead>
<tr>
<th>Description</th>
<th>Windows shortcut</th>
<th>Mac shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a mark</td>
<td>Click</td>
<td>Click</td>
</tr>
<tr>
<td></td>
<td>Windows shortcut</td>
<td>Mac shortcut</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td><strong>Select a group of marks</strong></td>
<td>Drag</td>
<td>Drag</td>
</tr>
<tr>
<td><strong>Add individual marks to the selection</strong></td>
<td>Ctrl+Click</td>
<td>Command+Click</td>
</tr>
<tr>
<td><strong>Add a group of marks to the selection</strong></td>
<td>Ctrl+Drag</td>
<td>Command+Drag</td>
</tr>
<tr>
<td><strong>Pan around the view</strong></td>
<td>Shift+Drag</td>
<td>Shift+Drag</td>
</tr>
<tr>
<td><strong>Zoom in to a point in the view (if not map, requires zoom mode)</strong></td>
<td>Double-click, Ctrl+Shift+Click</td>
<td>Double-click, Shift+Command+Click</td>
</tr>
<tr>
<td><strong>Zoom out from a point on a map (if not map, requires zoom mode)</strong></td>
<td>Ctrl+Shift+Alt+Click</td>
<td>Shift+Option+Command+Click</td>
</tr>
<tr>
<td><strong>Zoom out</strong></td>
<td>Shift+Double-click</td>
<td>Shift+Double+click</td>
</tr>
<tr>
<td><strong>Zoom in to an area in the view (requires zoom mode if not map)</strong></td>
<td>Ctrl+Shift+Drag</td>
<td>Shift+Command+Drag</td>
</tr>
<tr>
<td><strong>Zoom in and out on a map</strong></td>
<td>Scroll</td>
<td>Scroll</td>
</tr>
<tr>
<td><strong>Drag a row and scroll through a long list simultaneously</strong></td>
<td>Click+Drag to bottom of pane+Hold</td>
<td>Click+Scroll, Command+Hold</td>
</tr>
</tbody>
</table>

## Shortcuts for web authoring (Tableau Online or Tableau Server)

<table>
<thead>
<tr>
<th>Description</th>
<th>Windows shortcut</th>
<th>Mac shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toggle full screen</td>
<td>F11</td>
<td>Control+Command+F</td>
</tr>
<tr>
<td>Rename data source</td>
<td>Alt+F2</td>
<td>Option+F2</td>
</tr>
<tr>
<td>New data source</td>
<td>Ctrl+Alt+D</td>
<td>Control+D</td>
</tr>
<tr>
<td>Action</td>
<td>Keyboard Shortcuts</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td>Edit data source</td>
<td>Ctrl+Alt+Shift+D</td>
<td></td>
</tr>
<tr>
<td>Create new worksheet</td>
<td>Ctrl+Alt+T</td>
<td></td>
</tr>
<tr>
<td>Add field to sheet</td>
<td>Double-click</td>
<td></td>
</tr>
<tr>
<td>Copy a field in the view and place it on another shelf or card</td>
<td>Ctrl+Drag</td>
<td></td>
</tr>
<tr>
<td>Rename column field (when field is selected in the data schema pane)</td>
<td>F2</td>
<td></td>
</tr>
<tr>
<td>Toggle dashboard grid on and off</td>
<td>G</td>
<td></td>
</tr>
<tr>
<td>Toggle between Dashboard and Layout tabs</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Move floating dashboard object</td>
<td>Arrow key moves 1 pixel, Shift+arrow moves 10 pixels</td>
<td></td>
</tr>
<tr>
<td>Resize floating dashboard object</td>
<td>Alt+arrow key resizes in 1-pixel increments, Shift+Alt+arrow resizes in 10-pixel increments</td>
<td></td>
</tr>
<tr>
<td>Undo</td>
<td>Ctrl+Z</td>
<td></td>
</tr>
<tr>
<td>Redo</td>
<td>Ctrl+Y, Ctrl+Shift+Z</td>
<td></td>
</tr>
<tr>
<td>Save workbook</td>
<td>Ctrl+S</td>
<td></td>
</tr>
<tr>
<td>Save workbook as</td>
<td>Ctrl+Shift+S</td>
<td></td>
</tr>
<tr>
<td>Close web authoring</td>
<td>Alt+Q</td>
<td></td>
</tr>
<tr>
<td>Open Help</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Control+Command+Question Mark</td>
<td></td>
</tr>
</tbody>
</table>
Copyright

© 2018 Tableau Software, Incorporated and its licensors. All rights reserved.

Additional information regarding Tableau’s copyrights, patents and trademarks can be found at www.tableau.com/ip.

For information about Third Party Code which may be distributed with the Software, please review the Tableau Software End User License Agreement.