Introduction to Tableau

1. Data Connection:

Tableau can connect to various data sources, including spreadsheets, databases, cloud-based data warehouses, and more.

It supports real-time data connections, allowing users to analyze live data.

2. Data Preparation:

Tableau offers tools for cleaning, transforming, and shaping data to ensure it's suitable for analysis.

Users can merge, pivot, and aggregate data to create a cohesive dataset.

3. Drag-and-Drop Interface:

One of Tableau's strengths is its intuitive drag-and-drop interface. Users can drag fields onto shelves to create visualizations without writing complex code.

4. Visualizations:

Tableau supports a wide range of visualizations, including bar charts, line graphs, scatter plots, heat maps, and more.

Users can create dashboards with multiple visualizations to present a comprehensive view of their data.

5. Interactivity:

Dashboards in Tableau are highly interactive. Users can create filters, highlight specific data points, and enable drill-down features to explore data at different levels of detail.

6. Calculations and Formulas:

Tableau allows users to create calculated fields and apply formulas to manipulate and analyze data.

Advanced calculations and statistical functions can be performed to derive insights.

7. Sharing and Collaboration:

Tableau provides options for sharing visualizations and dashboards with others. This includes exporting images, creating PDFs, and publishing interactive dashboards to Tableau Server or Tableau Online.

Collaborative features enable multiple users to work on the same project simultaneously.

8. Integration:

Tableau integrates seamlessly with various third-party tools and platforms. It can be integrated with data science tools, statistical software, and other business intelligence systems.

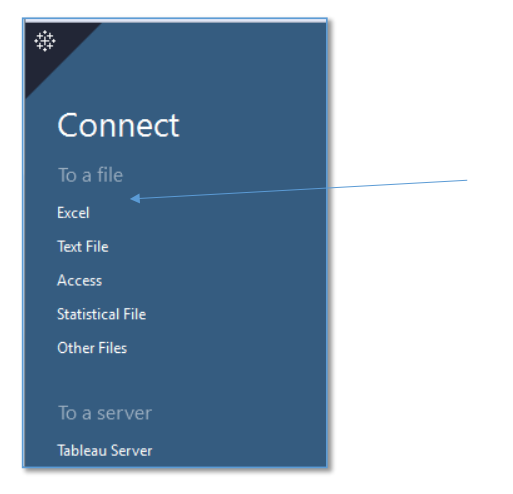
9. Tableau Server and Tableau Online:

Tableau Server and Tableau Online are platforms for sharing, collaborating, and distributing Tableau content within an organization.

These platforms enable users to access and interact with dashboards without requiring Tableau Desktop.

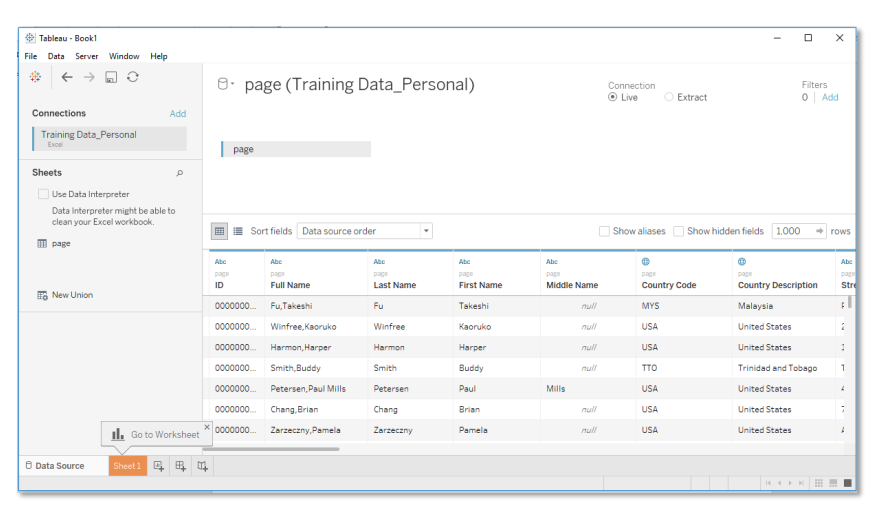
Connect to an Excel Data Source

Tableau works well with spreadsheets. In order to get familiar with the tool, you can download data from the Data Warehouse and save it locally in excel format. Once you have done this, you are ready to get started with the tool. In order to help you become familiar with the mechanics of the process you’ll need, our first example will create a connection to a local excel file. 1. Start by opening Tableau. From the start page, select the database type of Excel



2. Choose the file you’d like to use and click Open. For this example, use Training Data Personal in the Introduction to Tableau folder on your desktop.

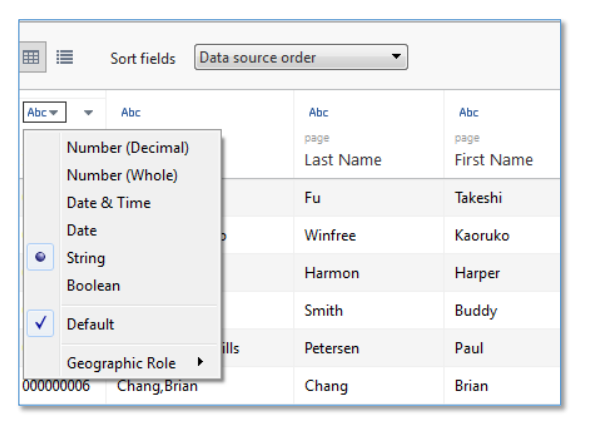
3. This will bring you to the Data Connection page. This page will appear any time you select a file, server, or saved data source on the start page. The left side of the page shows information about the data connection, including the data source type and tables within the data source. The white box in the upper area of the page shows the tables that are being used and includes options for adjusting the join type and fields used, if any Below this, in the area toward the bottom of the page is where you can preview the data. In the top area, under Connections, you can select whether to use a live connection or an extract. In the top right area, under Filters, you have the option to add filters to the data source to restrict the data included



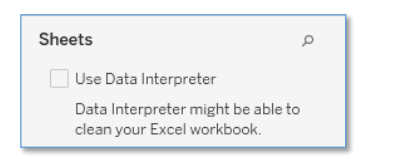
4. If you are opening a spreadsheet that has only one tab, you will see one sheet listed and the data connection will automatically be made. If there are multiple tabs, you will see a table listed for each tab. You’ll need to double-click the sheet name for the connection you’d like to make. Note that the data connection is automatically set as Live. If you want to create an extract, you should change the connection type.

5. Tableau has several options to ensure your data is properly imported. By default, it will import your data with the source format

If you notice that data doesn’t appear as expected on import, you can make adjustments to the definitions directly in the Metadata Grid. For example, an employee id might be created as a number or a date may have been imported as a string. You can change these to the appropriate definitions before going any further

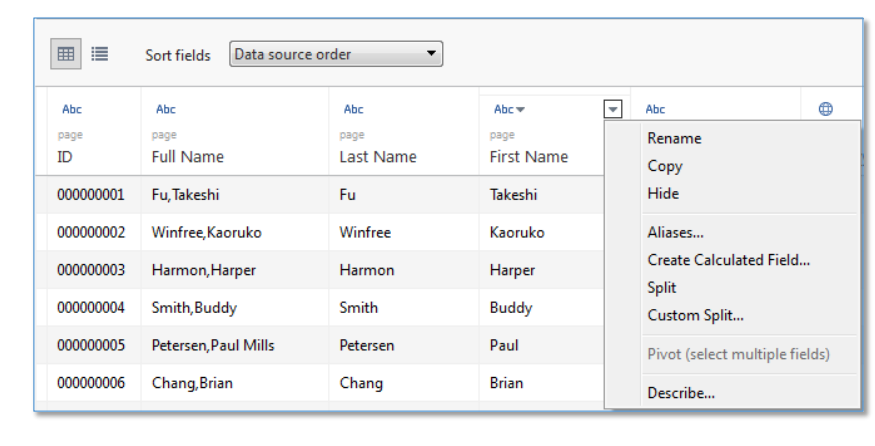


You can also turn on the Tableau Data Interpreter and it will attempt to interpret the data from your spreadsheet. When on, it will show warnings and a preview of its interpretations

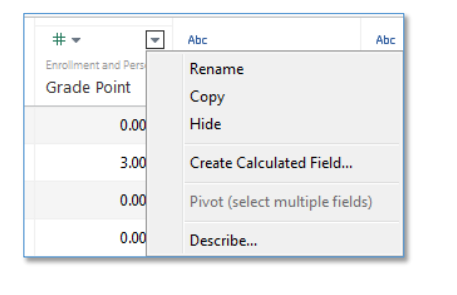


If you are connected to a Server data source, the data should already be properly defined. As a result, only the Geographical Role will be available for the field definition and the Tableau Data Interpreter will be turned off. 6. You can edit metadata information directly on the Data Connection page using the Metadata Grid. Alternatively, you can modify data attributes in the Data window or your worksheet. Using the Metadata Grid, you can rename, copy, hide, manage aliases, and create calculated fields from the view. This view makes it easy to see all of the fields in a simple format and to make quick changes to the data source before going to the worksheet

Additional options from this view are Split, Custom Split, and Pivot. The menu options will vary depending upon the selected field type and your data source type



The options for measures are limited to relevant actions including rename, copy, hide, and create calculated field



Please note that if you are using a server data source, you will not be able to manage aliases and that option is removed. 7. Once you have set your connection as desired, you are ready to start working with your data. At the bottom of the page, click the sheet under Go to Worksheet

Working with CSV Text Files in tableau

Here's a sample CSV file named "employee\_sales.csv":

EmployeeID,FirstName,LastName,Department,Sales

1,John,Doe,Sales,5000

2,Jane,Smith,Marketing,3000

3,Bob,Johnson,Operations,2000

4,Alice,Williams,Sales,6000

5,Charlie,Brown,Marketing,3500

Now, let's walk through the steps to visualize this data using Tableau:

* Open Tableau Desktop:
* Launch Tableau Desktop.
* Connect to Data:

Click on "Connect" and select "Text File." Locate and select the "employee\_sales.csv" file.

Review and Edit Data:

Tableau will display a preview of the data. Ensure that the data types are correct, and make any necessary adjustments.

Create a Simple Bar Chart:

Drag the "Department" field to the Columns shelf.

Drag the "Sales" field to the Rows shelf.

Tableau will automatically create a bar chart showing sales by department.

Customize the Chart:

Customize the chart by adding labels, titles, and adjusting colors as needed.

Create a Dashboard:

Click on the "New Dashboard" button to create a new dashboard.

Drag the bar chart onto the dashboard.

Save and Share:

Save the Tableau workbook, and you can share it as a Tableau file (.twb) or publish it to Tableau Server or Tableau Online.

Tableau Product Overview

Tableau Desktop:

Description: Tableau Desktop is the authoring and publishing tool that allows users to connect to various data sources, create interactive dashboards, and design visualizations.

Key Features:

Intuitive drag-and-drop interface.

Support for various data connectors.

Powerful data exploration and visualization capabilities.

Tableau Server:

Description: Tableau Server is an enterprise-level platform that facilitates the sharing, collaboration, and distribution of Tableau content within an organization.

Key Features:

Centralized repository for Tableau workbooks and data sources.

Web-based access to Tableau content.

Collaboration features, such as commenting and subscriptions.

Tableau Online:

Description: Similar to Tableau Server, Tableau Online is a cloud-based platform for sharing and collaborating on Tableau content. It eliminates the need for on-premises infrastructure.

Key Features:

Cloud-based deployment.

Scalability and flexibility.

Accessibility from anywhere with an internet connection.

Tableau Mobile:

Description: Tableau Mobile enables users to access and interact with Tableau dashboards on mobile devices, ensuring a responsive and user-friendly experience on tablets and smartphones.

Key Features:

Native apps for iOS and Android.

Offline access for selected content.

Touch-optimized interface.

Tableau Prep:

Description: Tableau Prep is a data preparation tool that allows users to clean, shape, and combine data from various sources before analyzing it in Tableau.

Key Features:

Visual and direct manipulation of data.

Cleaning and shaping operations.

Integration with Tableau Desktop and Tableau Server.

Tableau Public:

Description: Tableau Public is a free version of Tableau that allows users to create and share public Tableau visualizations and dashboards. It's often used for educational and personal projects.

Key Features:

Cloud-based hosting of public visualizations.

Community sharing and collaboration.

Limited to non-sensitive, public data.

Tableau Reader:

Description: Tableau Reader is a free desktop application that allows users to view and interact with Tableau packaged workbooks (.twbx) created by others. It's primarily for consuming content, not authoring.

Key Features:

Ability to view and explore Tableau visualizations.

No editing capabilities.

Tableau Extension API:

Description: The Tableau Extension API allows developers to create custom extensions that can be integrated into Tableau Desktop and Tableau Server dashboards, expanding functionality beyond built-in features.

Connecting to Databases in tableau

Connecting to data in Tableau is a fundamental step in the process of creating visualizations and dashboards. Tableau supports a wide range of data sources, including databases, spreadsheets, cloud services, and more. Here's a general guide on how to connect to data in Tableau:

Open Tableau Desktop:

Launch Tableau Desktop on your computer.

Connect to Data:

On the start page, click on the "Connect" pane or go to the "Data" menu and select "Connect to Data."

Choose a Data Source:

Tableau provides a variety of connectors for different data sources. Common options include Excel, SQL databases, cloud services like Google Sheets or Salesforce, and more.

Select the appropriate data source type for your data.

Select and Connect to Data:

Depending on the data source you choose, you may need to provide connection details such as server names, file paths, credentials, etc.

After entering the required information, click on the "Connect" button.

Navigate and Preview Data:

Tableau will display a data connection window where you can navigate through tables, sheets, or objects in your chosen data source.

Preview the data to ensure that it's loaded correctly.

Data Source Tab:

Once connected, Tableau will open the Data Source tab. Here, you can see the fields and tables available in your data source.

Join or Union Tables (if needed):

If your data is spread across multiple tables, you may need to join or union them in the Data Source tab to create a unified dataset.

Data Preparation:

Tableau provides options for data cleaning and shaping. You can rename fields, change data types, create calculated fields, and perform other data preparation tasks.

Drag and Drop to Create Visualizations:

Move to the "Worksheet" tab to start building visualizations. Drag fields from the left pane onto the Columns and Rows shelves to create charts and graphs.

Save the Workbook:

Save your Tableau workbook (.twb or .twbx) to preserve the connection and configuration settings.

Explore and Analyze:

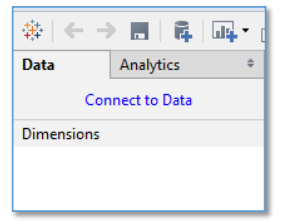
Explore your data, create calculated fields, add filters, and customize your visualizations to derive meaningful insights.

Save and Share:

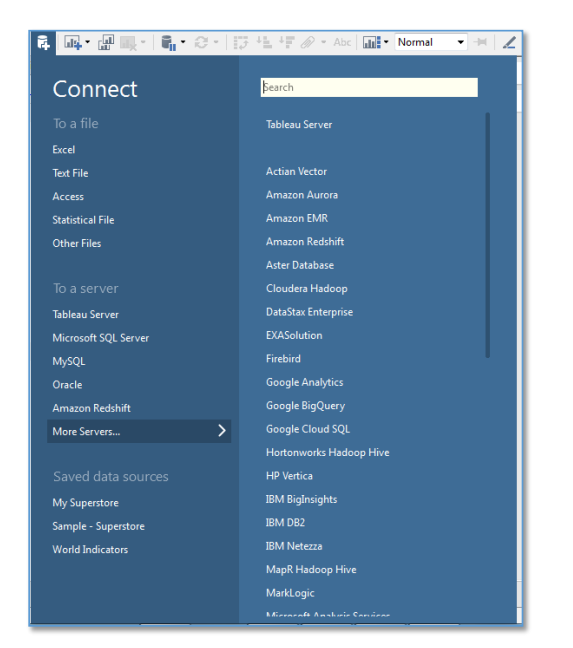
Once you've created your visualizations and dashboards, save your work, and consider publishing to Tableau Server or Tableau Online for wider sharing and collaboration.

Connecting to databases

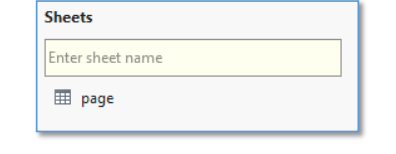
If you have files from more than one source you can still include these in the same visualization. You just need to take separate steps to join them in Tableau. When using a Data Blend, the secondary data is aggregated when joined to the first. You can practice this in our third example. 1. In Tableau, on the File menu click New to open a new session. This time we’ll connect to our data sources directly in our Data window. Click Connect to Data



The Data Connection selection from the Start Page will open.

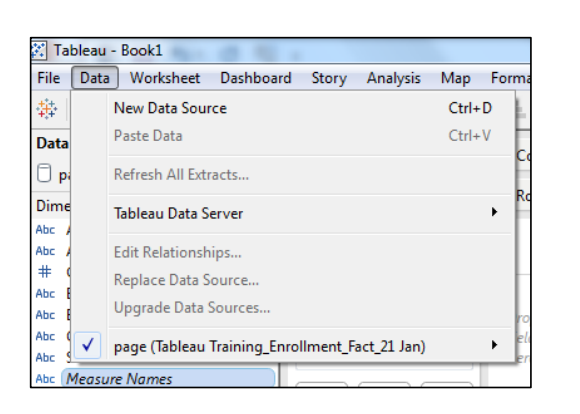


2. Click Excel and select Enrollment Fact.xlsx. This will bring you to the Data Connection page. Notice that as with our first example there is only one table available for selection

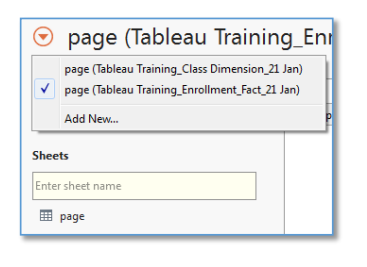


3. Click Sheet 1 to navigate to your worksheet.

4. In order to connect to a second table, on the Data menu select New Data Source

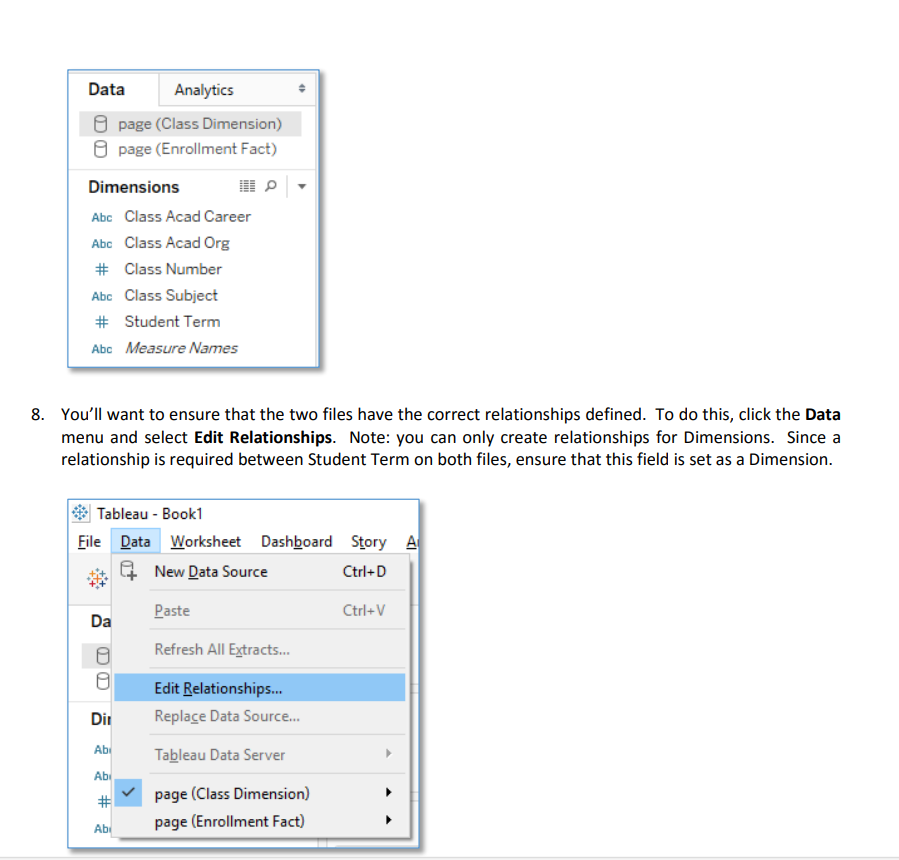


5. Select Excel and then choose Class Dimension.xlsx. This will bring you again to the Data Connection window. This time you’ll notice that again there is only one table displayed but this time the preview shows the data from the Class Dimension file. You do have the option to switch between data sources by clicking the orange arrow to the left of the data source name. However, you do not have the option to define the connections between the tables here

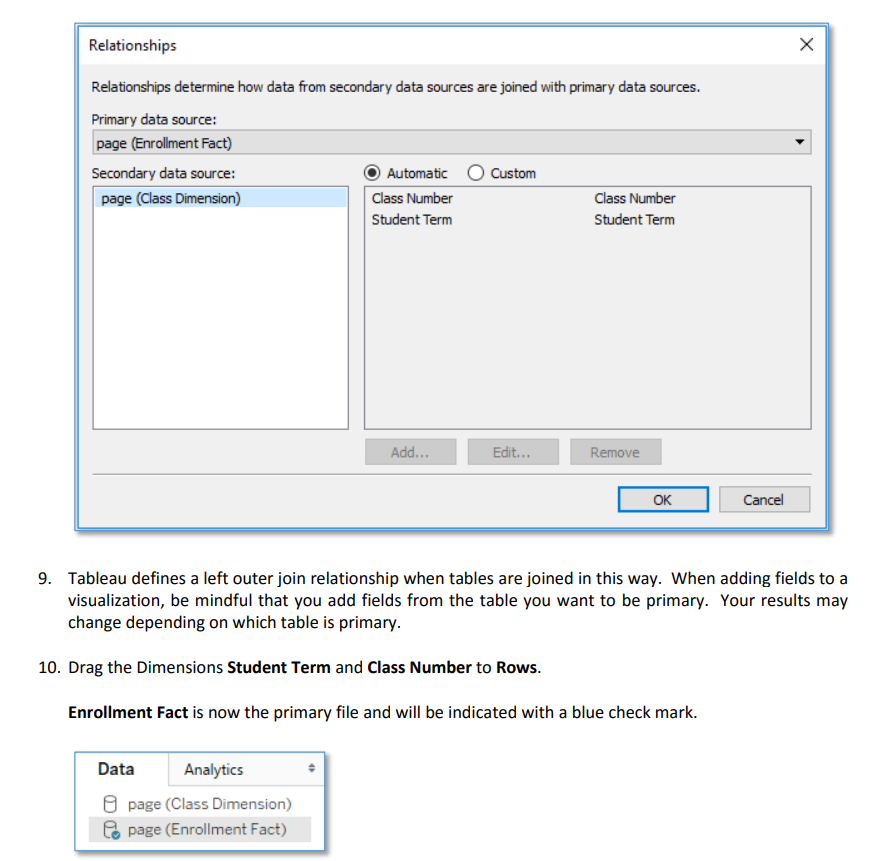


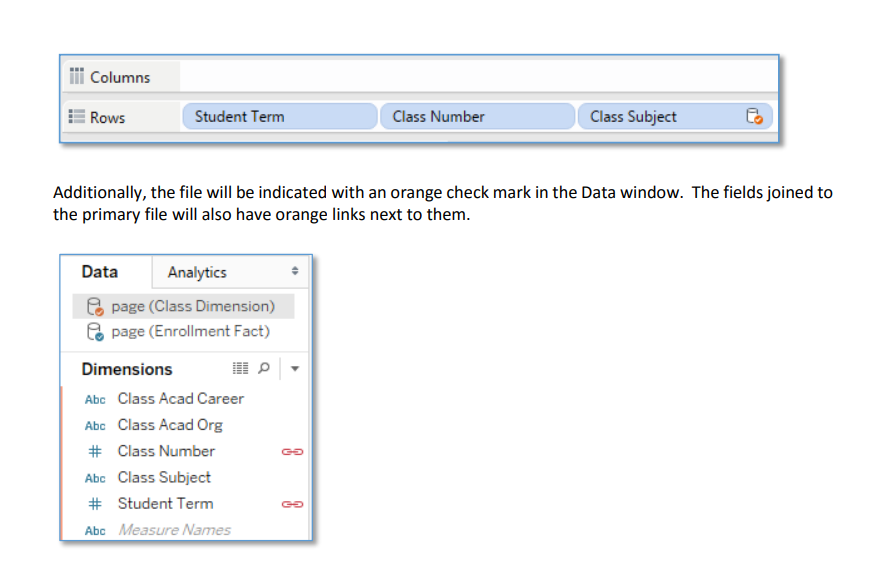
6. Click Sheet 1 to return to your worksheet.

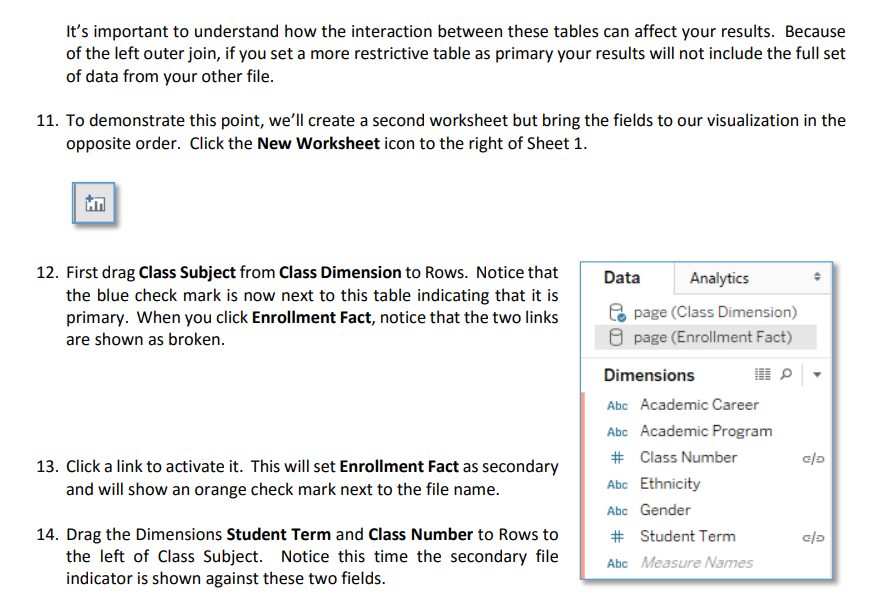
7. In the Data window notice that you now have two data sources shown, both Enrollment Fact and Class Dimension. The Dimensions and Measures will reflect the specific Data source you have selected

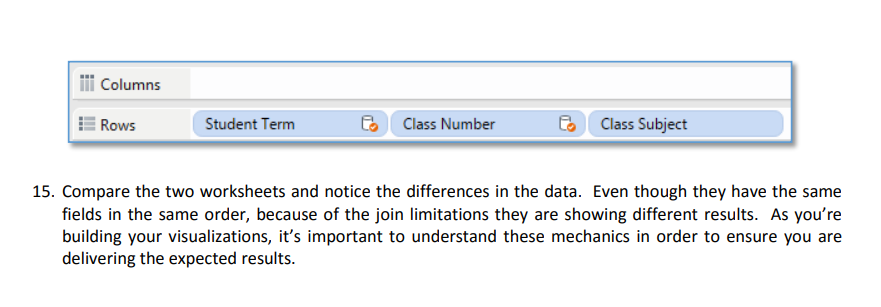


The Relationships window will open. You can select the table you want to be primary in your join here. Tableau will automatically attempt to map the two tables based on matching column descriptions. You can choose to leave the mapping or make changes by adding, editing, or removing these selections by selecting the Custom radio button. For our example, we will accept the mapping that Tableau has suggested. Click OK to close the window

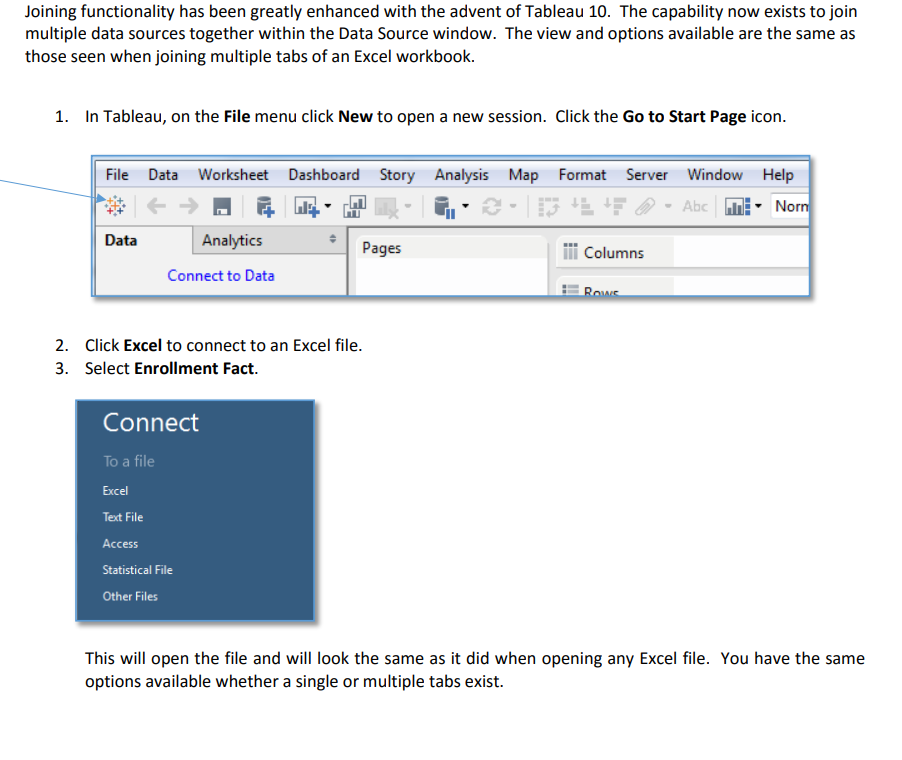


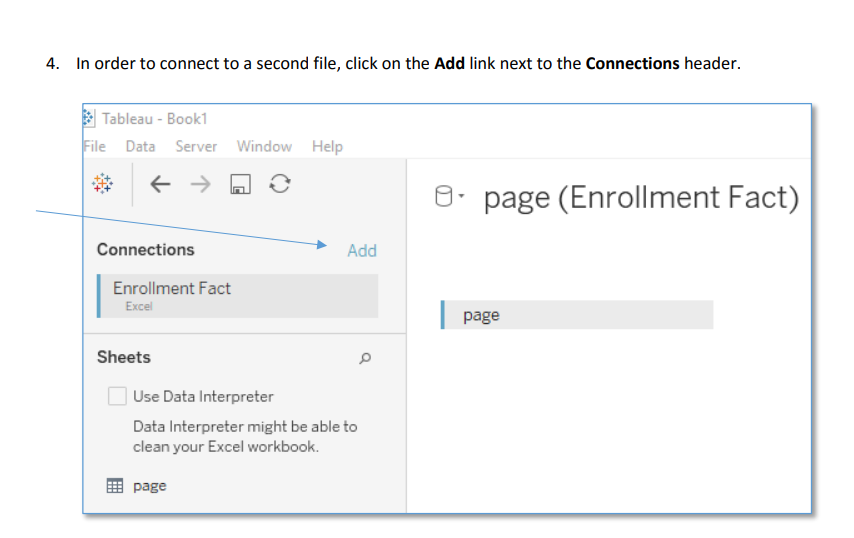


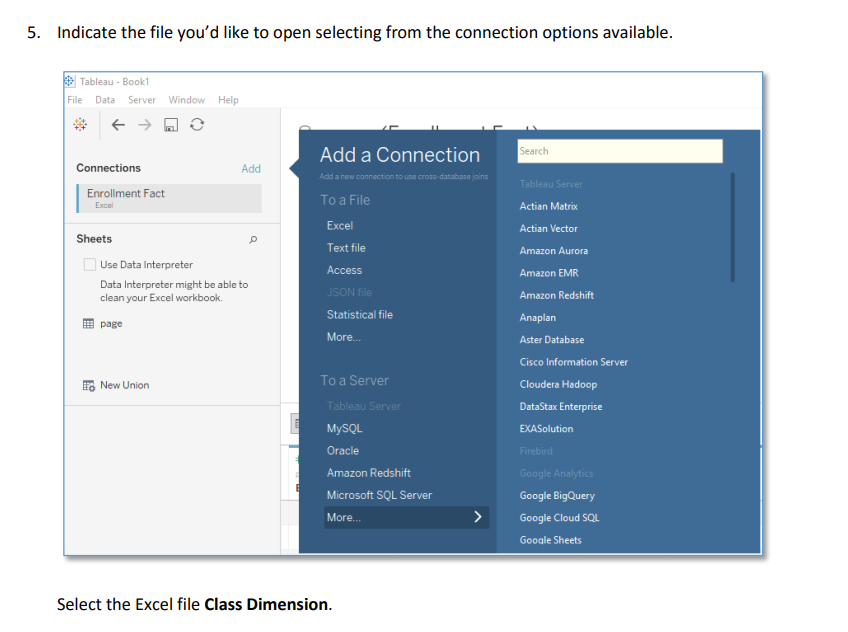




Working with data







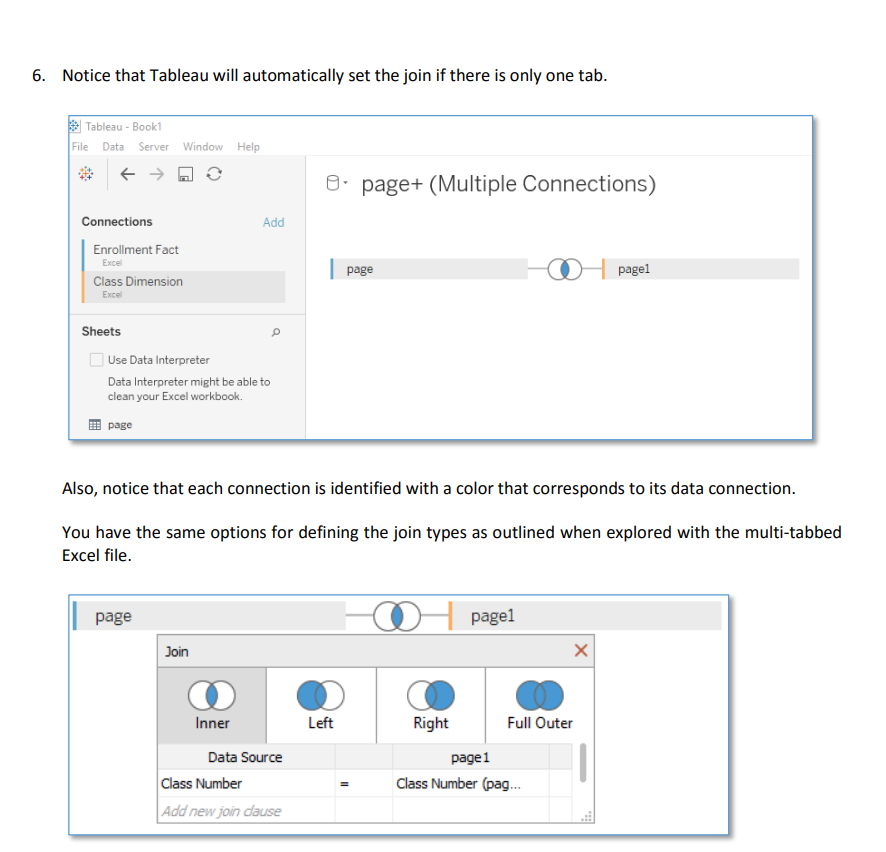
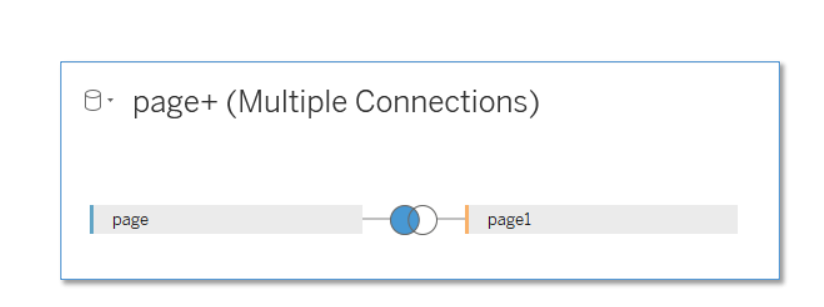
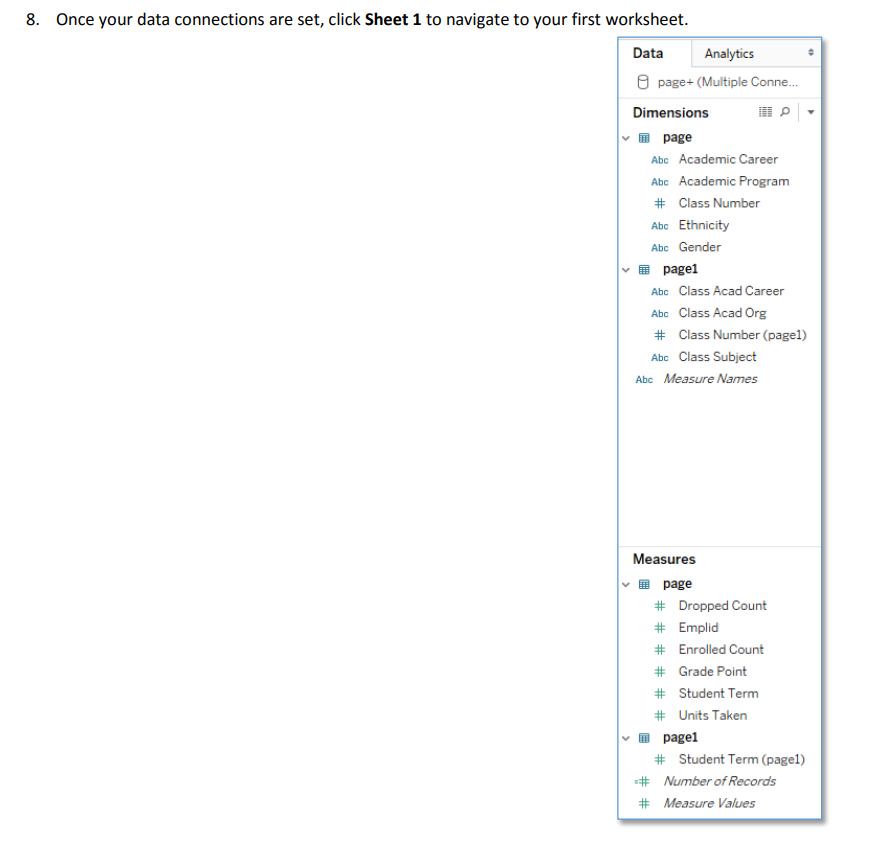


Tableau will automatically try to join on matching field names. If the initial join is not correct, fields may be added or removed and the join type changed. For our exercise, leave Class Number as the only join field and change the join type to a left join by clicking Left and then click the red X to close the Join window. 7. When complete, your connections should look like this





When to Blend and When to Join

Both methods of integrating your data will result in making additional details available within Tableau.

However, they are not completely interchangeable. Joins are generally preferable for supplementing a data

source. However, there are exceptions when blending may be the better choice. For example, the granularity

of two sources may differ. In this case, blending is be best choice for combining the two.

Specific instances where blending is either useful or the preferred method are outlined below.

• If there is a need to combine data from different data sources where joins are not supported, a data

blend should be used to integrate the data and make it available for use within a single sheet.

• If data in one data set is at a different level of granularity from data in a second set. For example, you

might want to join departmental transactions with the department quarterly budgets. There might be

two different data sets, one including detailed purchases for each department and a second including

the quarterly budget figures. Due to the different level of detail, a blend would be the preferred

solution.

• If you have tables that do not match up correctly after a join, blending allows for the integration of the data.

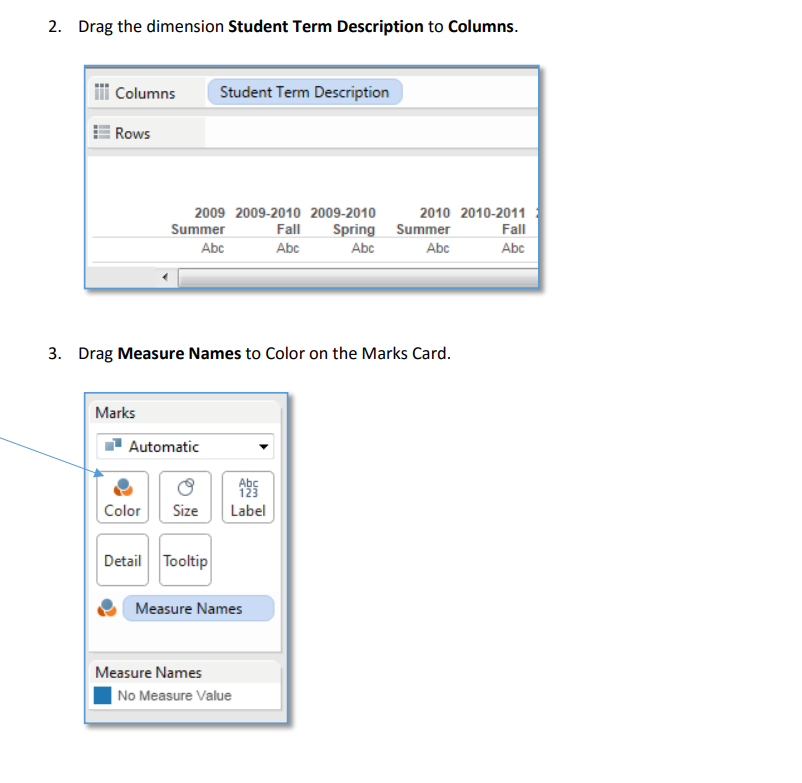
• When joining tables results in duplicate data, this is indicative of different levels of detail. In this case, it is best to choose a blend.

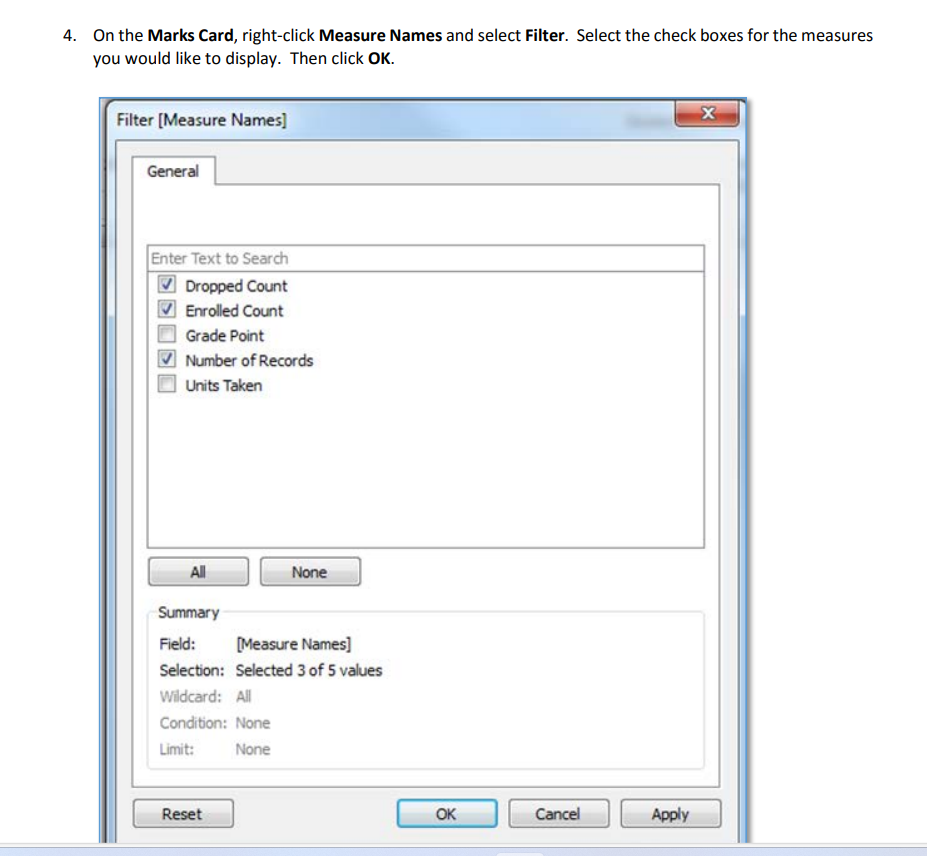
• If you’re working with very large sets of data, joins can put a strain on the database and significantly affect performance. Because Tableau handles combining the data after it’s been aggregated, there is

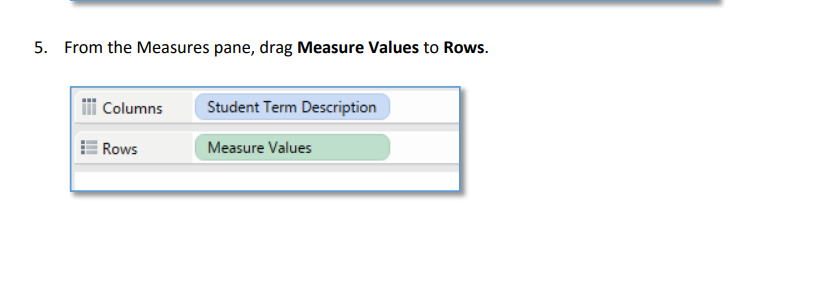
less data to combine within a blend. As a result, this may be a preferred method

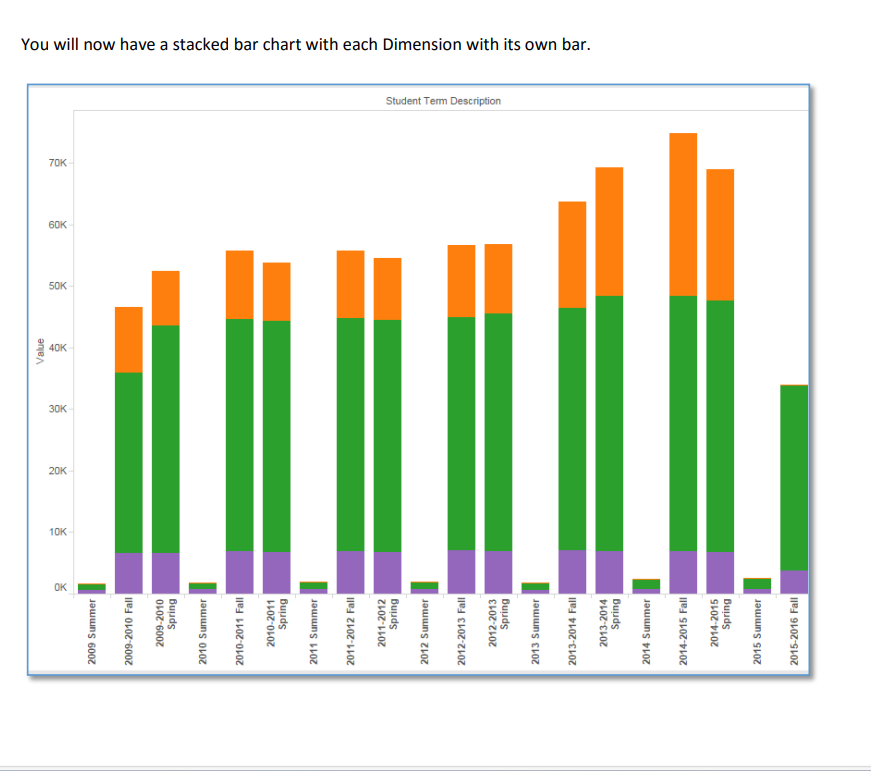
Analyzing and generating reports

Bar Charts A bar chart is a good choice of visualization when you are comparing different groups or to track changes over time. It’s good to note that when tracking changes over time, the bar chart is best when the change is large. If not another type of visualization might be better. There are several simple steps that can be followed to create different types of bar charts in Tableau. We’ll step through a few in our exercises. Exercise: Create a Stacked Bar Chart with a Separate Bar for Each Dimension 1. First connect to the server data source Enrollment and Person Fact +. Create a blank worksheet. To do this you can either create a new worksheet using the New Worksheet icon or by right-clicking on your current sheet and selecting New Worksheet. 2. Drag the dimension Student Term Description to Columns









TabPy, or Tableau Python Integration, allows you to execute Python scripts and code from within Tableau. This integration provides a way to leverage Python's advanced analytics and machine learning capabilities while still benefiting from Tableau's data visualization and dashboard creation features. Here's an overview of how TabPy combines Python and Tableau:

Setting Up TabPy:

Install TabPy Server:

You need to set up TabPy server on your machine or a server. You can download it from the Tableau website.

Configure Tableau Desktop:

Open Tableau Desktop.

Navigate to Help > Settings and Performance > Manage External Service Connection.

Enter the TabPy server information.

Using TabPy in Tableau:

Connecting to TabPy:

In Tableau Desktop, connect to your data source as usual.

Create a calculated field where you want to use Python code.

Writing Python Code:

In the calculated field, you can use the SCRIPT\_REAL, SCRIPT\_INT, or SCRIPT\_STR functions to embed Python code.

For example:

SCRIPT\_REAL("

import pandas as pd

# Your Python code here

result = your\_python\_function()

return result

", [Your Tableau Fields])

Passing Data between Tableau and Python:

You can pass data between Tableau and Python using parameters or input fields.

The fields specified in the last argument of the SCRIPT function are the input fields that are sent to Python.

Executing Python Code:

When you use a Tableau worksheet, the calculated field with Python code will execute the Python script on the TabPy server.

The result of the Python script is then returned to Tableau and used in the visualization.

Example Use Case:

Let's consider a simple example where you want to calculate the average of a column using Python:

Create a Calculated Field:

In Tableau Desktop, create a calculated field with the following Python script:

SCRIPT\_REAL("

return pd.Series(data).mean()

", [Your Numeric Field])

Use the Calculated Field in Your Visualization:

Drag the calculated field to your Tableau worksheet.

Tableau will execute the Python script, and you'll see the result in your visualization.