SRM Institute of Science and Technology

Chennai Ramapuram

Faculty of Science & Humanities

(A Place for Transformation)

PG Department of Computer Applications

PRACTICAL RECORD

NAME :

REGISTER NUMBER :

PROGRAMME : **MCA**

SEMESTER/YEAR : **II/I**

SECTION :

COURSE CODE : **PCA25DO6J**

COURSE NAME : **DATA VISUALIZATION TECHNIQUES**

APRIL 2026

SRM Institute of Science and Technology

Chennai Ramapuram

Faculty of Science & Humanities

(A Place for Transformation)

PG Department of Computer Applications

REGISTER NUMBER :

BONAFIDE CERTIFICATE

This is to certify that the bonafide work done by _____

in the subject Data Visualization Techniques (PCA25D06J) at, SRM Institute of Science and Technology, Chennai Ramapuram in APRIL 2026.

STAFF IN-CHARGE **HEAD OF THE DEPARTMENT**

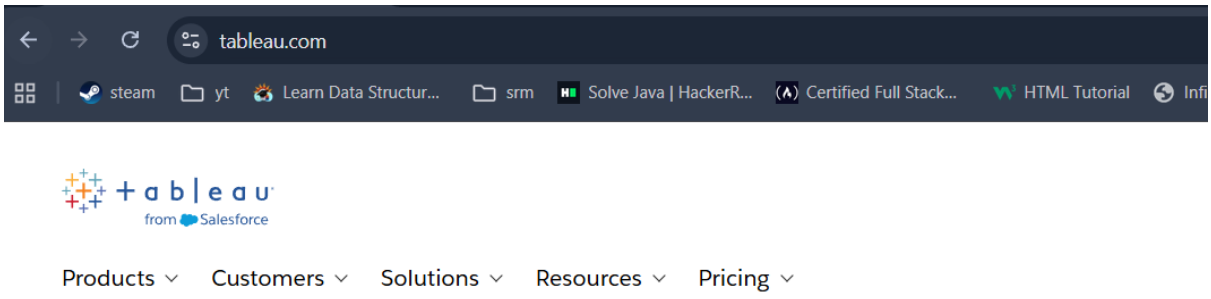
Submitted for the University Practical Examination help at SRM Institute of Science and Technology, Chennai Ramapuram on _____.

INTERNAL EXAMINER 1 **INTERNAL EXAMINER 2**

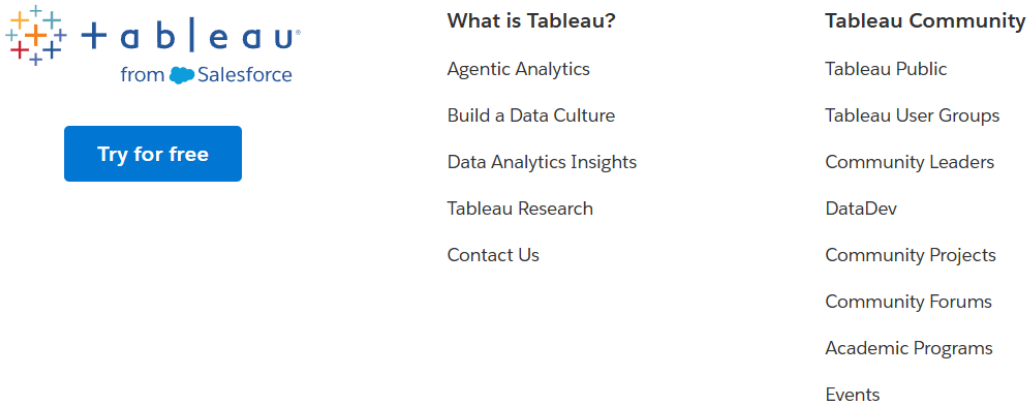
INDEX

Practice 1: Installing Tableau

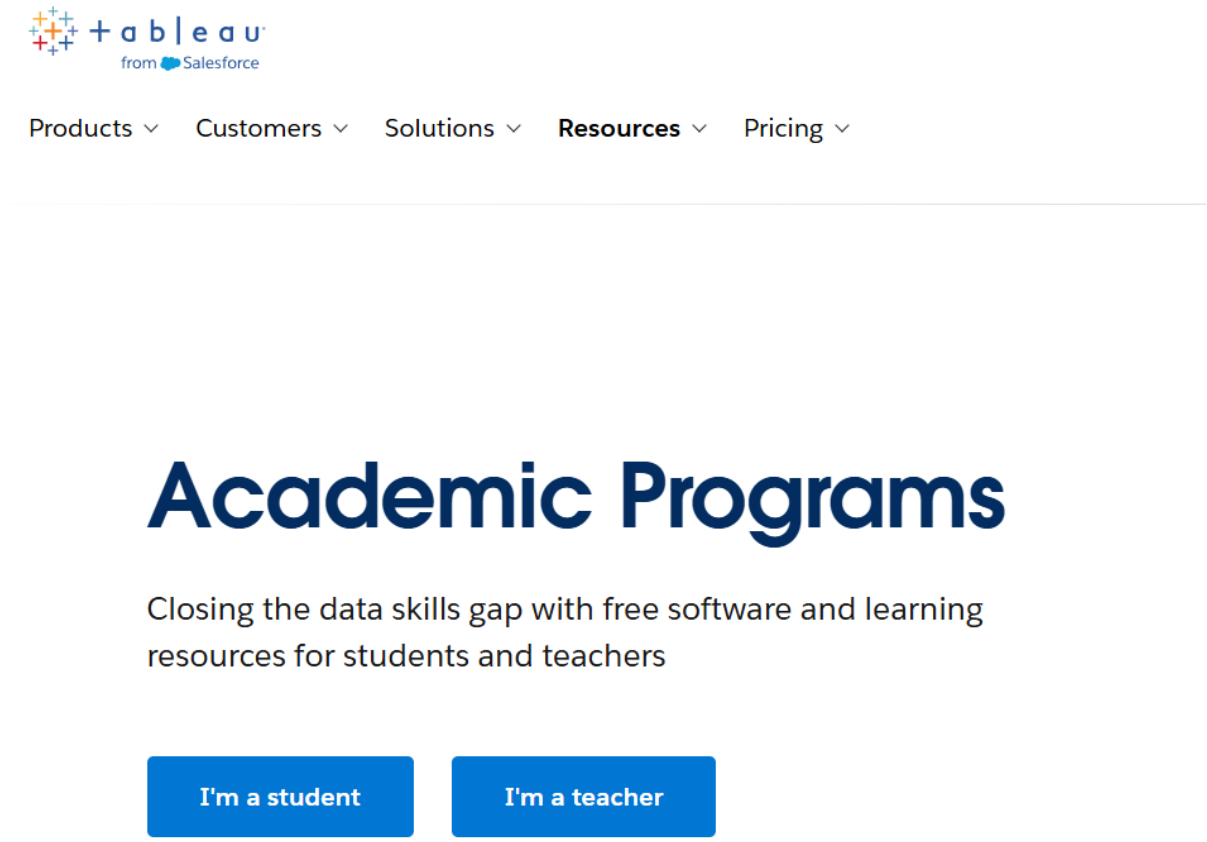
Step 1) Go to https://www.tableau.com/ on your web browser.



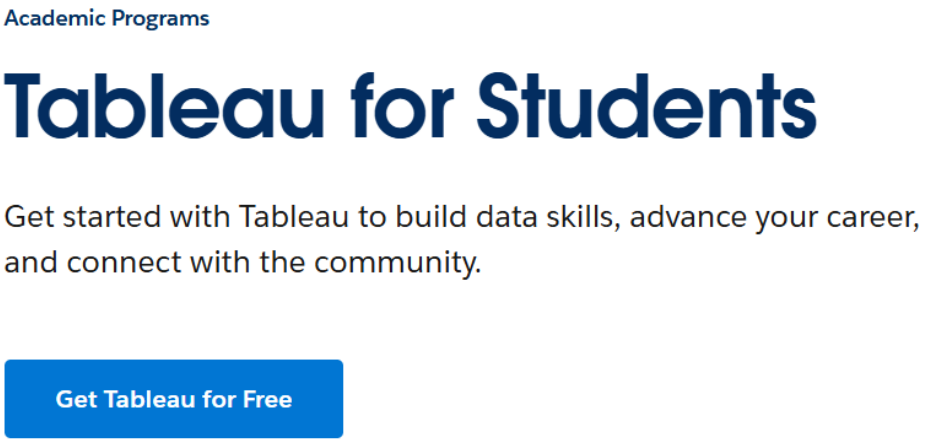
Step 2) Scroll down in the same page and select "**Academic Programs**" in **Tableau Community.**



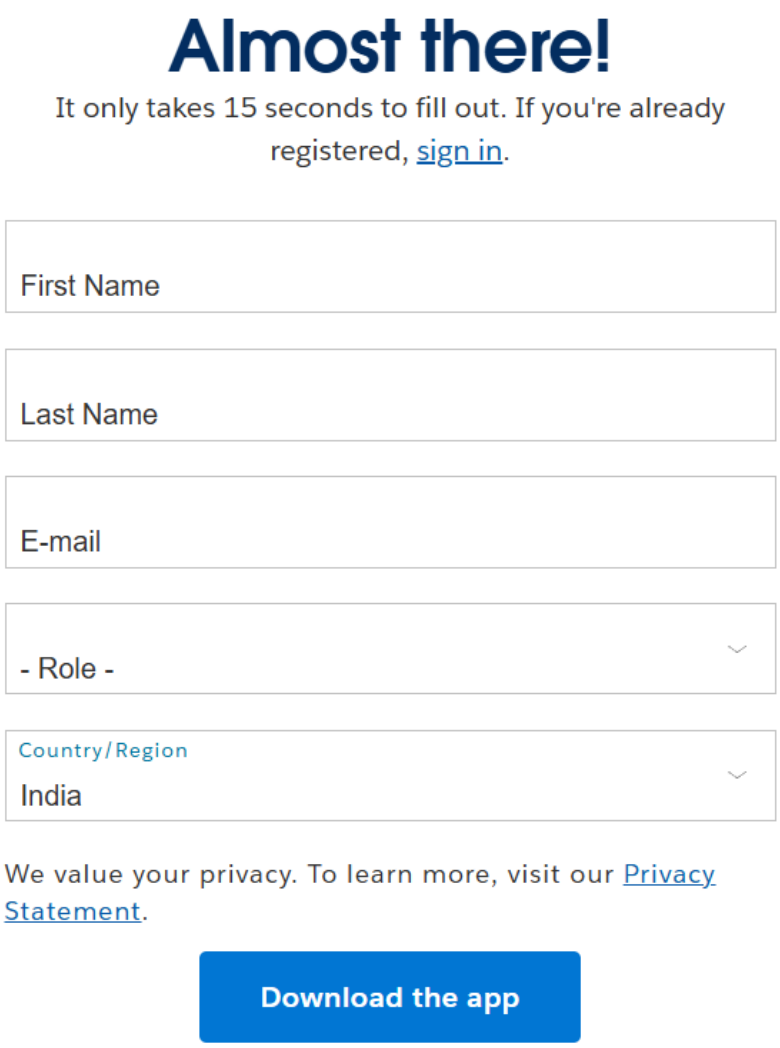
Step 3) It opens a page and we have to select "**I am Student**"



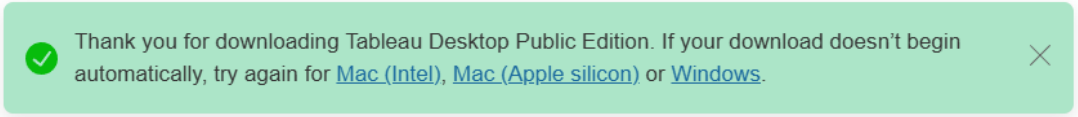
Step 4) We get a new page Tableau for Students and we have to click on **"Get Tableau for Free"**



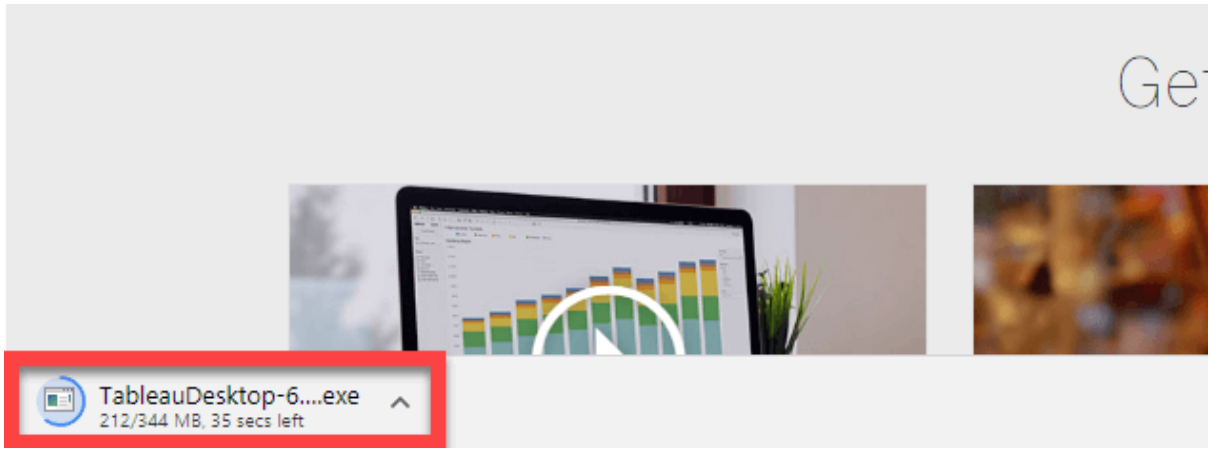
Step 5) It will ask for your information for downloading, after filling the information click on "**Download the app**"



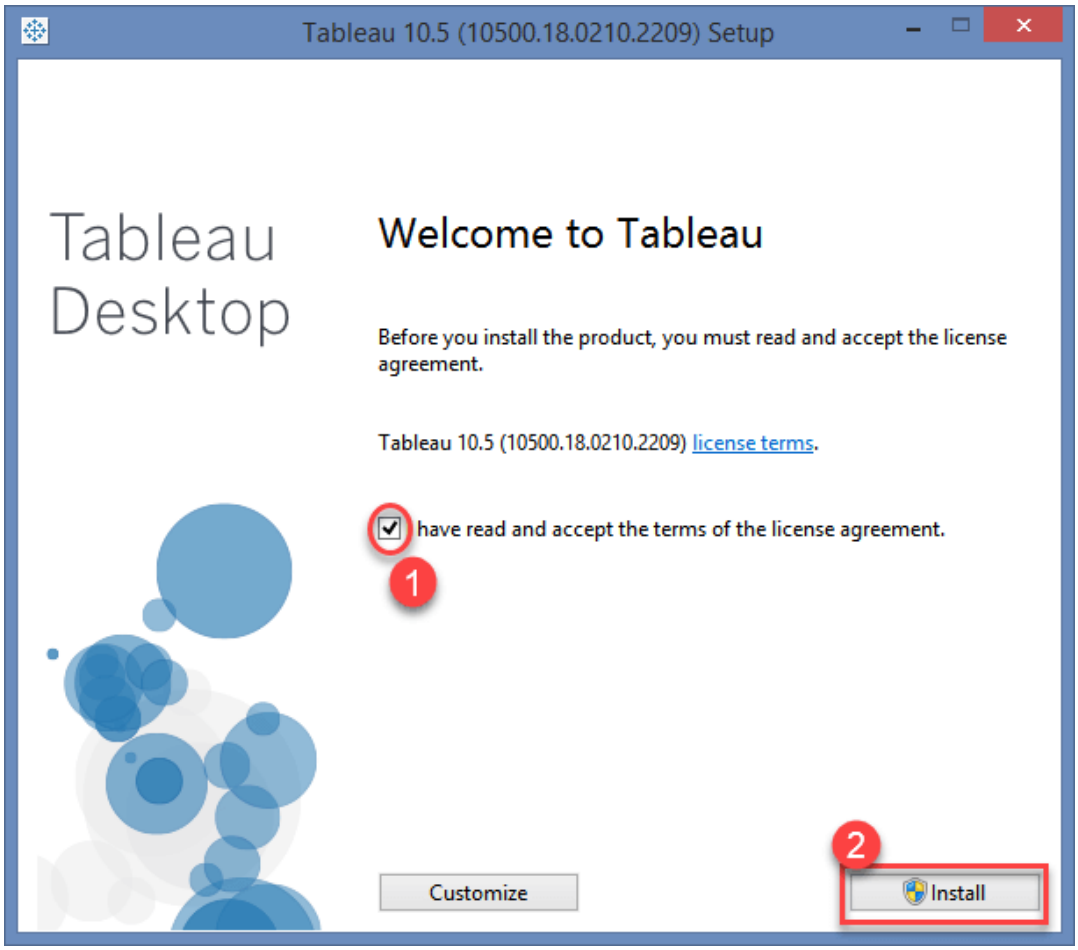
Step 6) A optional pop-up message will be shown. If download doesn't begin automatically click on "**Windows button"** on the message.



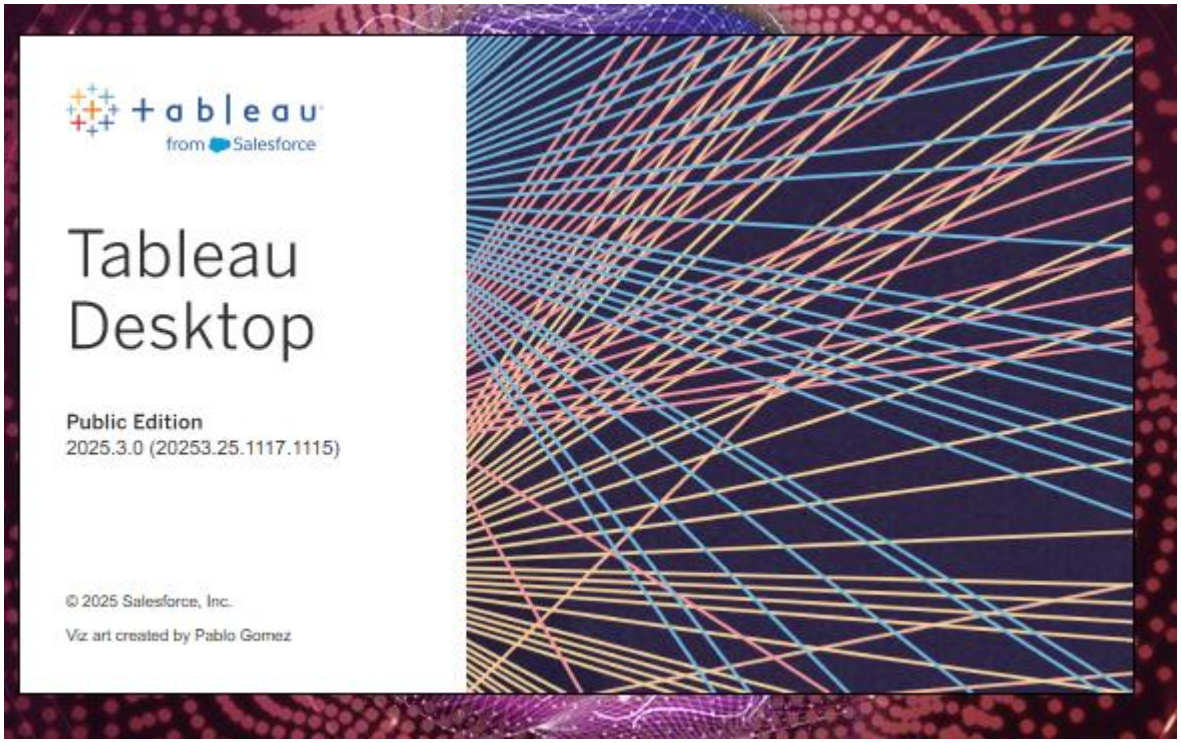
Step 7) This will start downloading tableau latest version. An .exe file for Windows is downloaded, and you can see the downloading process in the bottom left corner of the website.



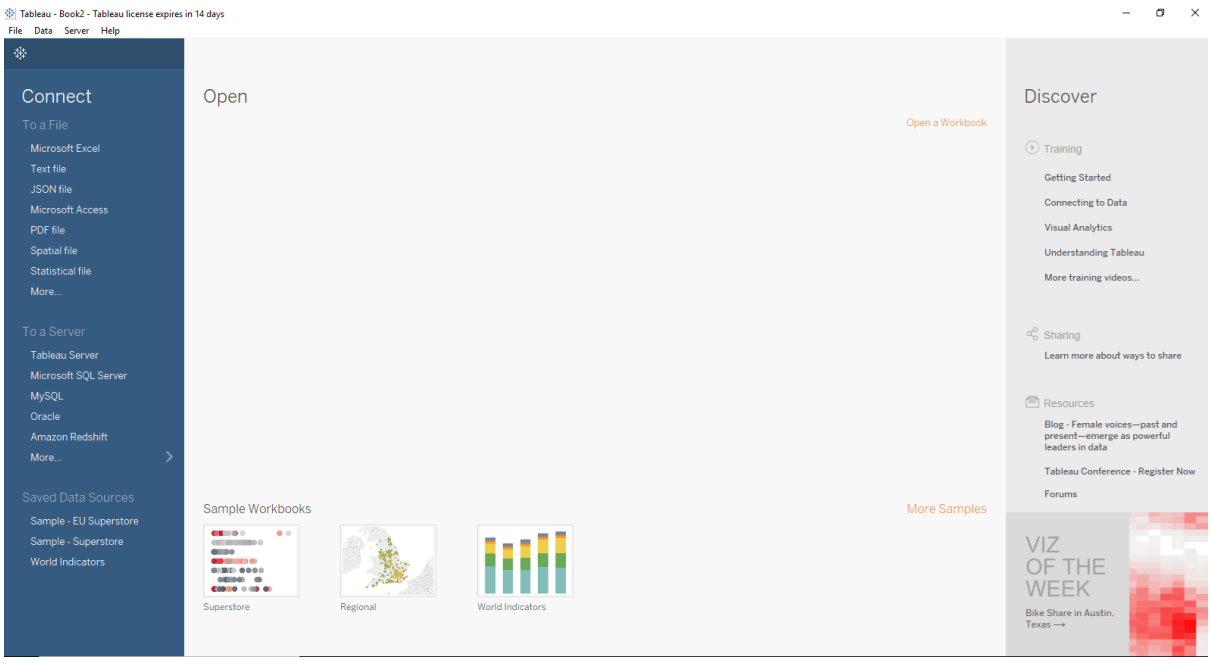
Step 8) Open the downloaded file. Check in to accept the terms and conditions and click on "Install" button.



Step 9) Once the tableau desktop download and installation is completed, open the Tableau Desktop software.



Step 10) Start Screen of Tableau is shown

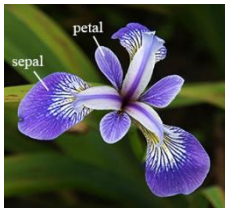


Now you are all set to use the Tableau Desktop in your windows system.

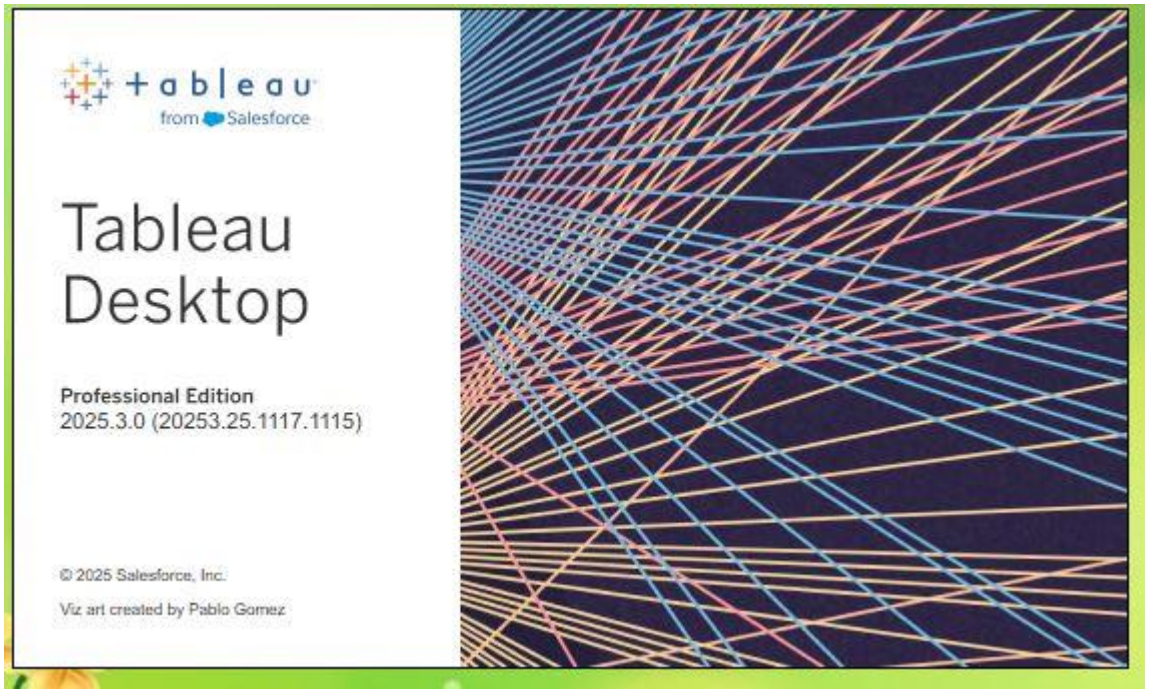
Practice 2: Working with sample dataset in a Tableau

Steps to Connect to a Data Source

Example: Iris Dataset



Step 1) Open Tableau



Launch Tableau Desktop. On the left-hand side of the start screen, we will see the Connect Panel.

Step 2: Select a Data Source

Under the File section in the Connect Panel, choose the type of data file we want to connect with. For example, to connect to an Excel file, click on Microsoft Excel

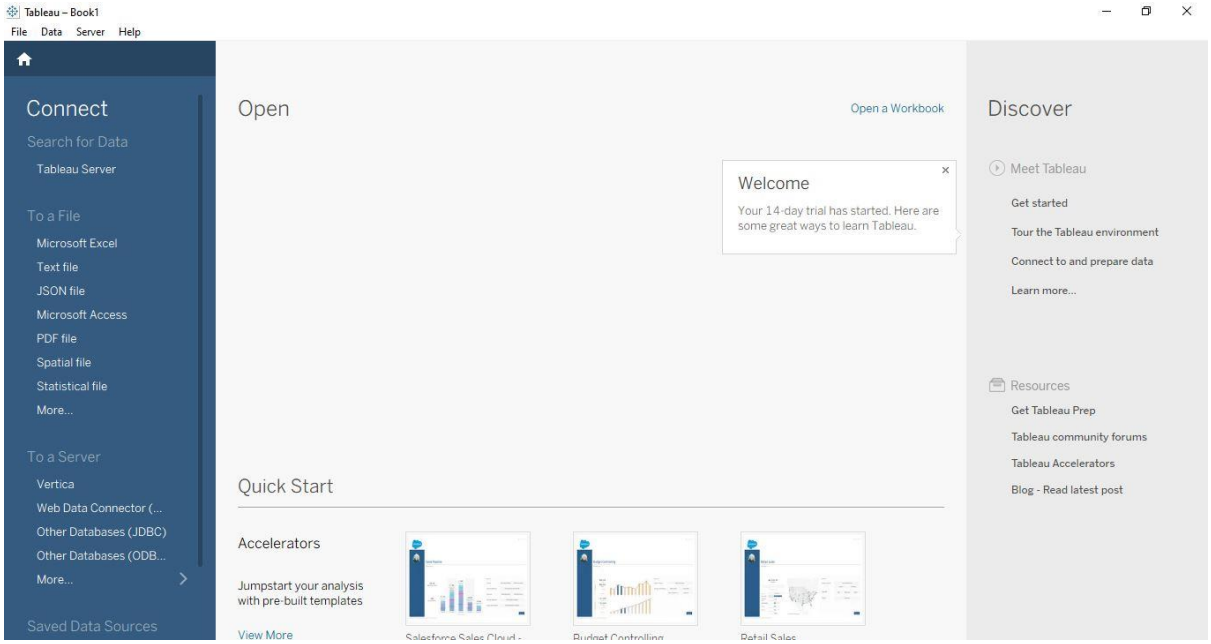
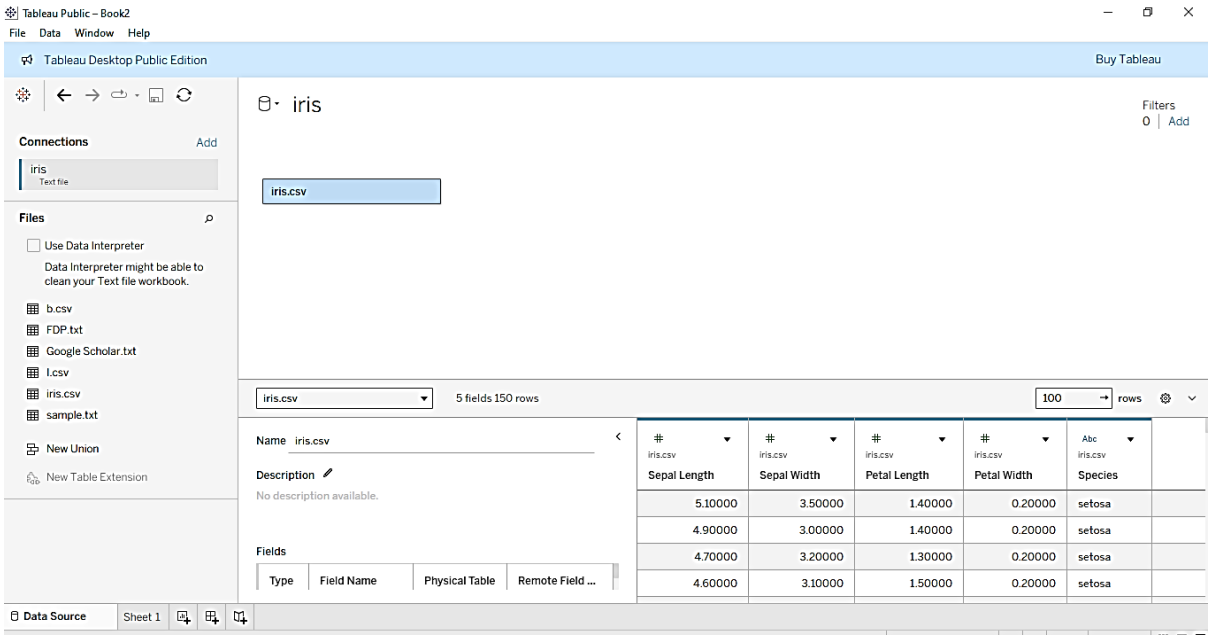


Tableau – Book1
File Data Server Help
Connect
Search for Data
Tableau Server
To a File
Microsoft Excel
Text file
JSON file
Microsoft Access
PDF file
Spatial file
Statistical file
More...
To a Server
Vertica
Web Data Connector (...
Other Databases (JDBC)
Other Databases (ODB...
More...
Saved Data Sources
Open
Open a Workbook
Welcome
Your 14-day trial has started. Here are some great ways to learn Tableau.
Quick Start
Accelerators
Jumpstart your analysis with pre-built templates
View More
Discover
Meet Tableau
Get started
Tour the Tableau environment
Connect to and prepare data
Learn more...
Resources
Get Tableau Prep
Tableau community forums
Tableau Accelerators
Blog - Read latest post

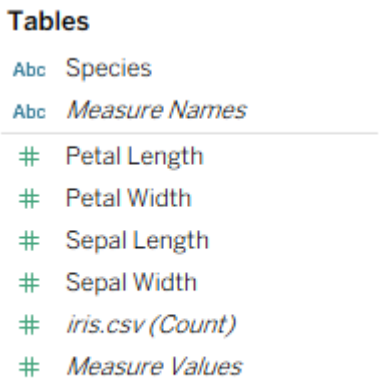
Step 3: Browse and Load the File

Once selecting MS Excel or CSV file, a file dialog box will appear and we can select the desired data file or Drag and drop the file.

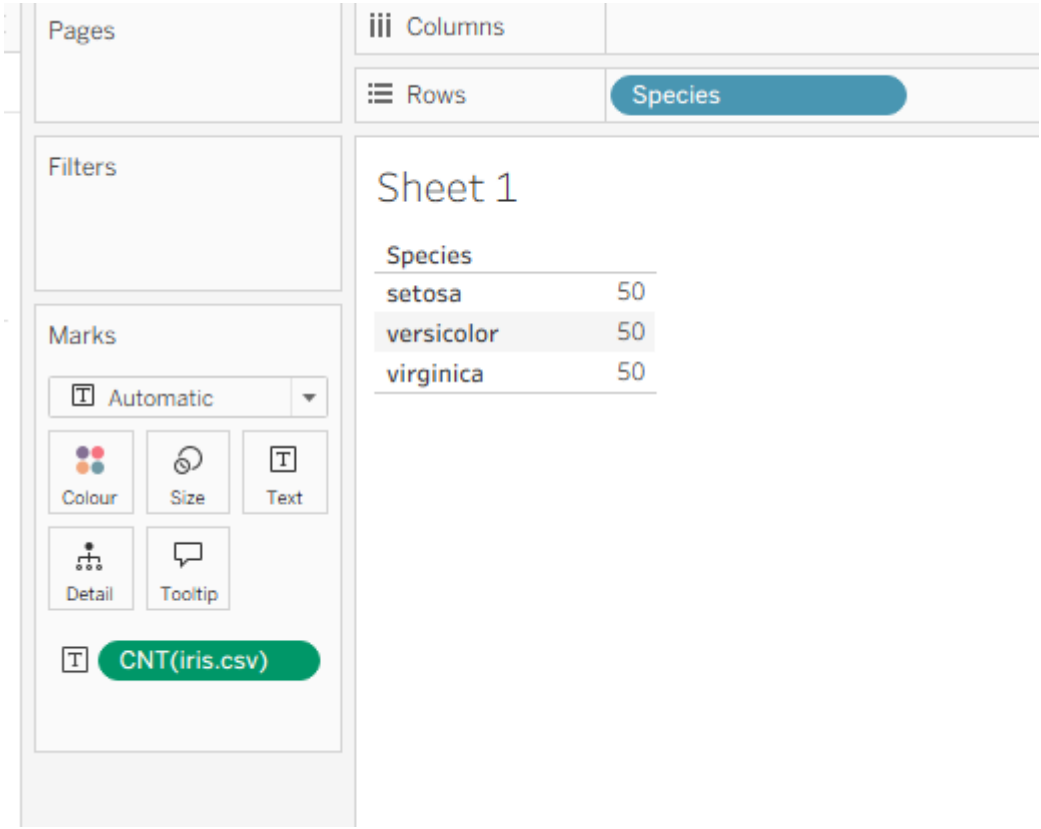
Step 4: The iris data file loaded and ready for visualization.



Step 5: Select Sheet 1. In tables, Species is present in Dimension, and Petal Length, Petal Width, Sepal Length, Sepal Width is present in Measures.

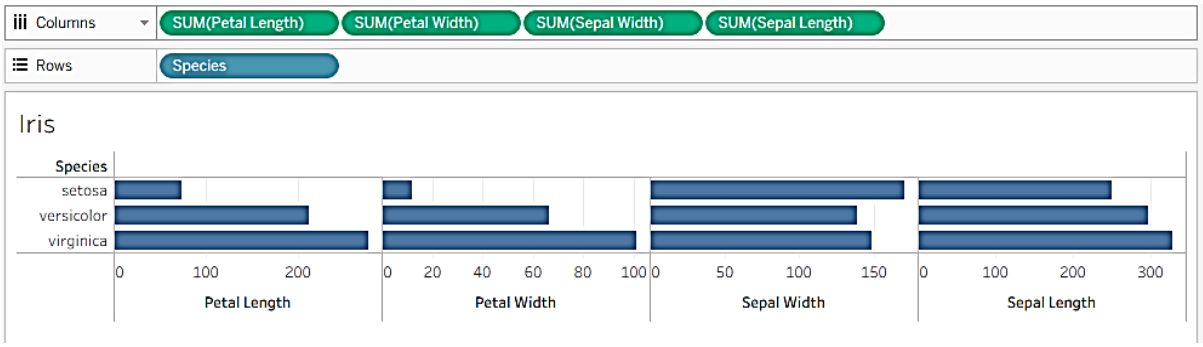


Step 6: For Data Visualization, Drag Species and drop in rows, iris.csv (count) in Text of Marks filed.

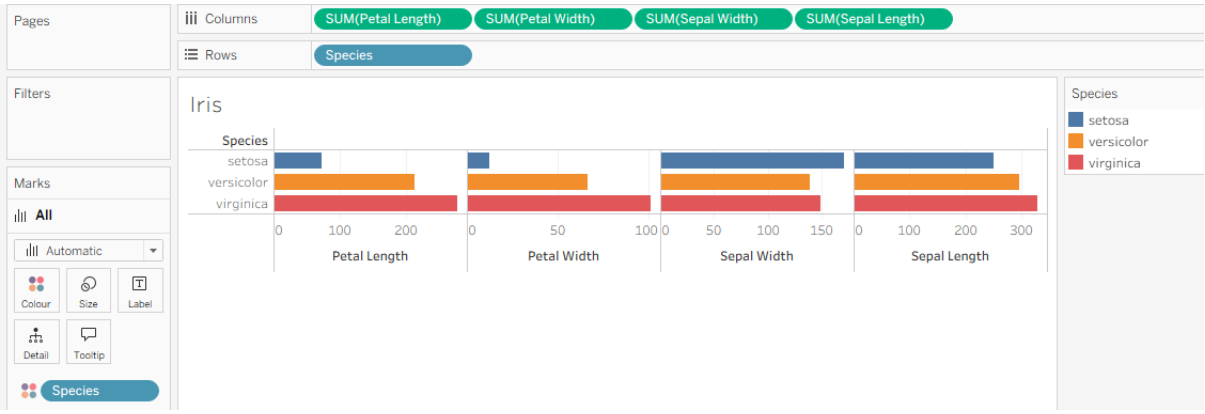


The result shows the count of each species individually.

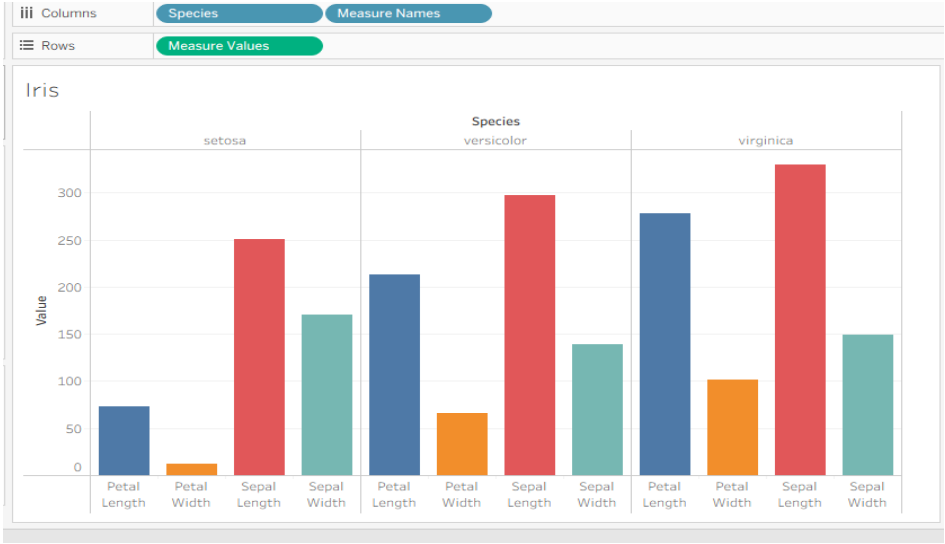
Step 7: Drag and drop Species in Rows, Petal Length, Petal Width, Sepal Width, Sepal Length in Columns. In Show Me, you can view in different visualizations.



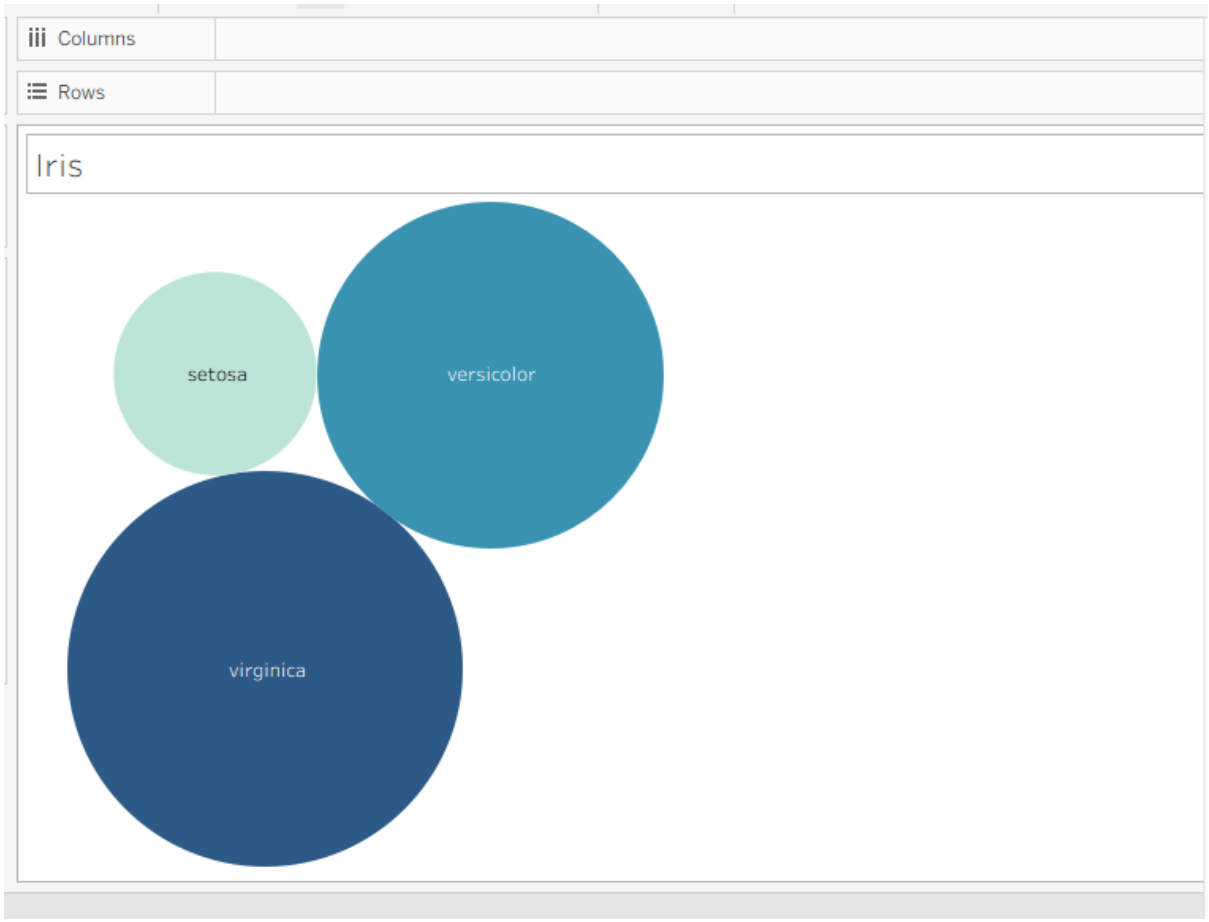
Step 8: Drag and Drop Species in Colour.



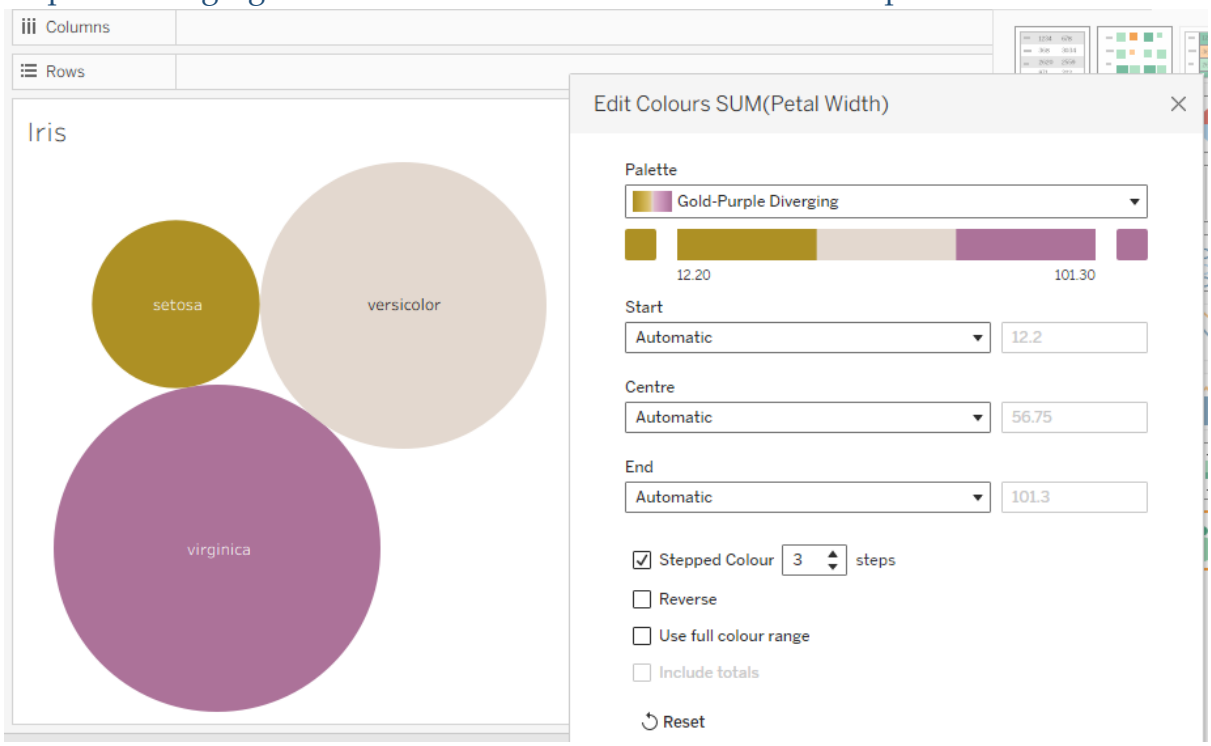
Step 9: Different types of visualizations. Select Side-by-Side bar in ShowMe.



Step 10: Choose Packed Bubbles in ShowMe



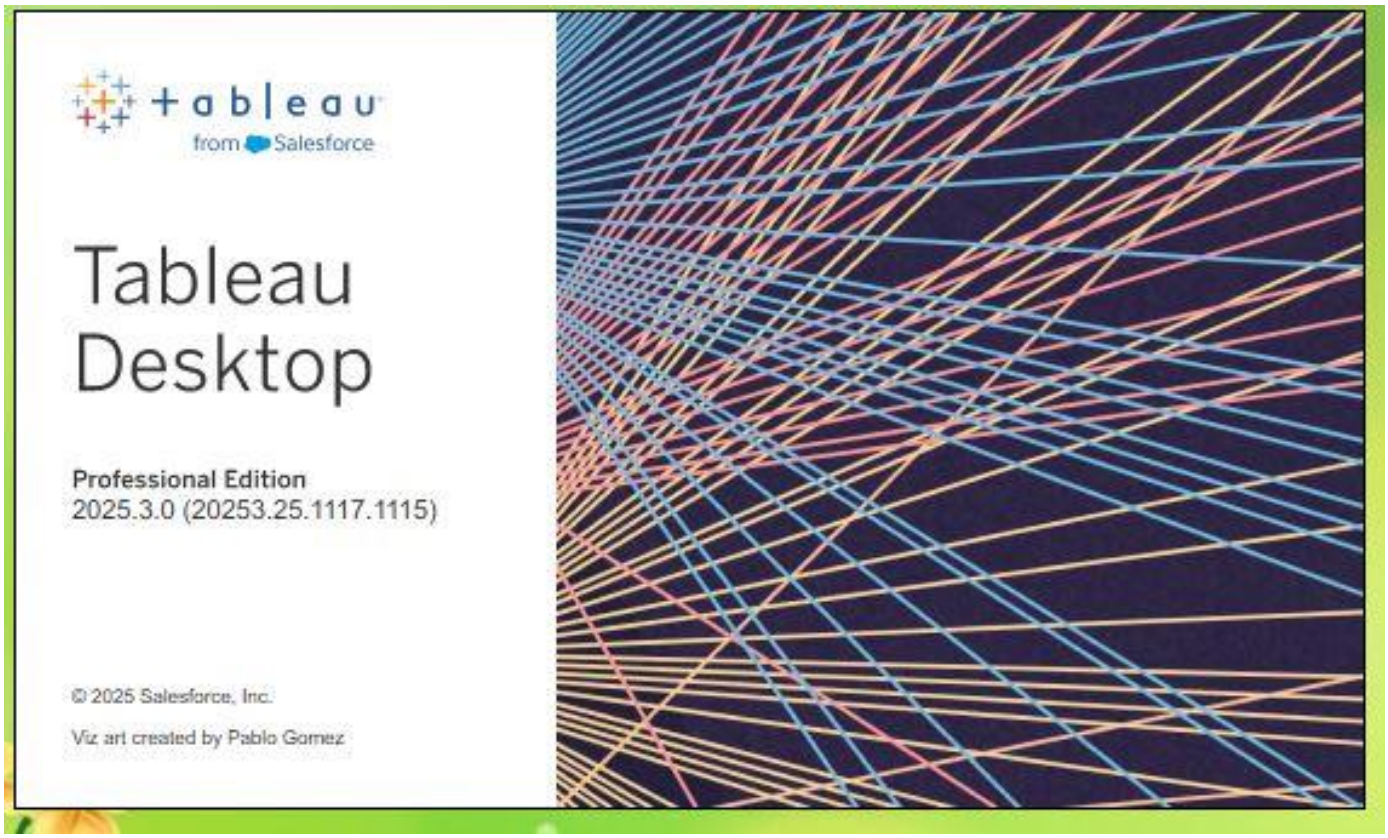
Step 11: Changing colours in Colour menu with Edit colour options.



Practice 3: Working with Data Tables

Step 1: Open Tableau

Launch Tableau Desktop. On the left-hand side of the start screen, we will see the Connect Panel.



Step 2: Select a Data Source

Under the File section in the Connect Panel, choose the type of data file we want to connect with. For example, to connect to a Text file, **click on Text File**.

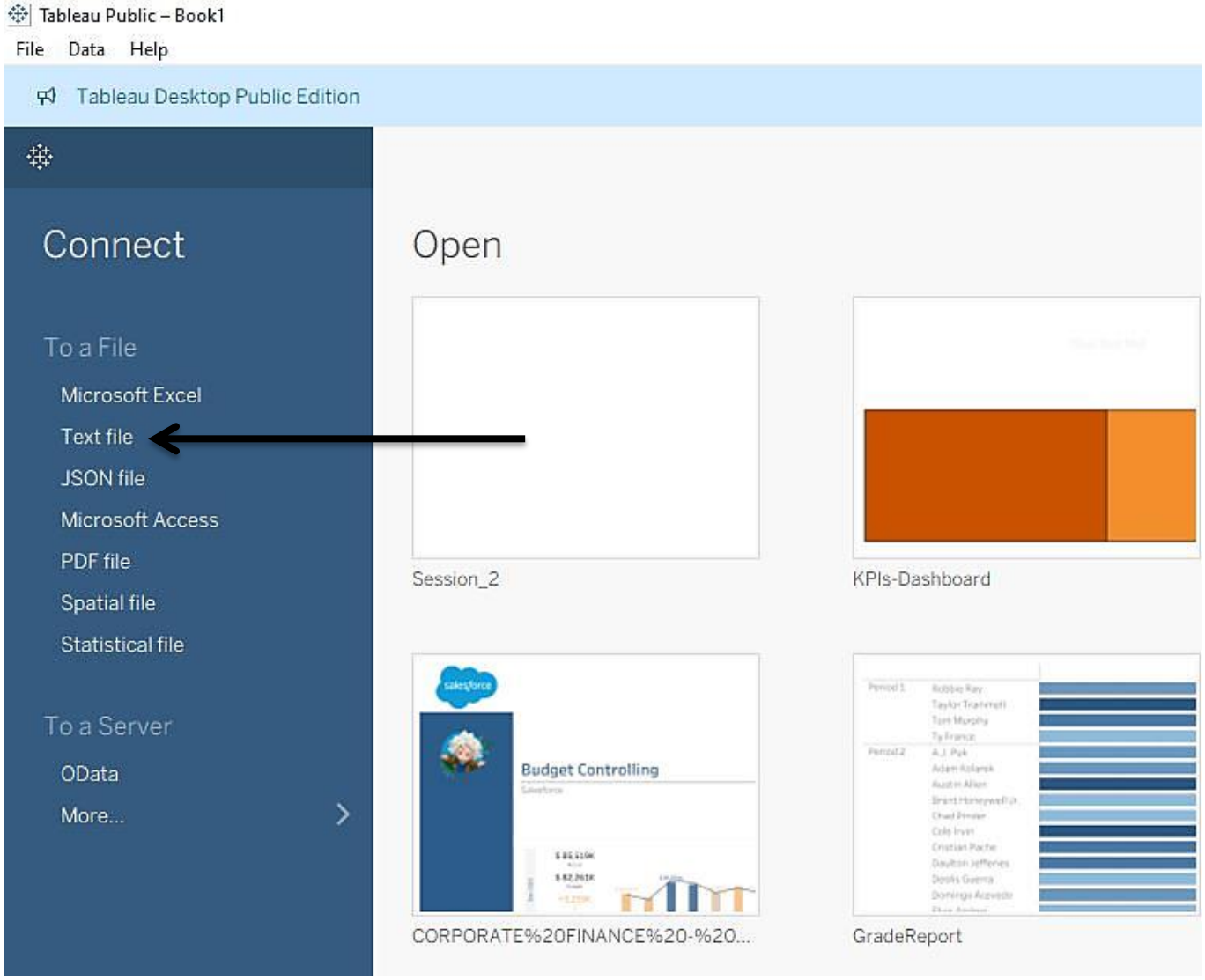
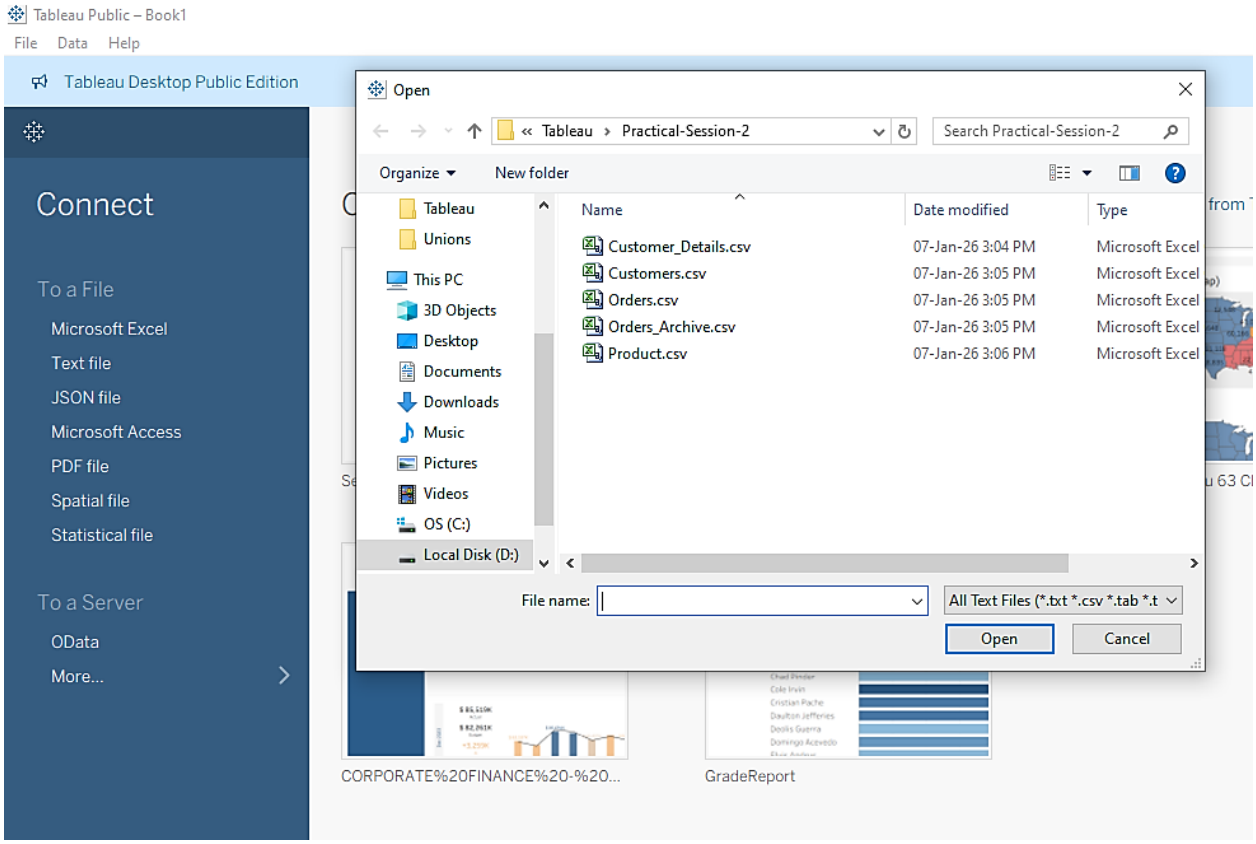


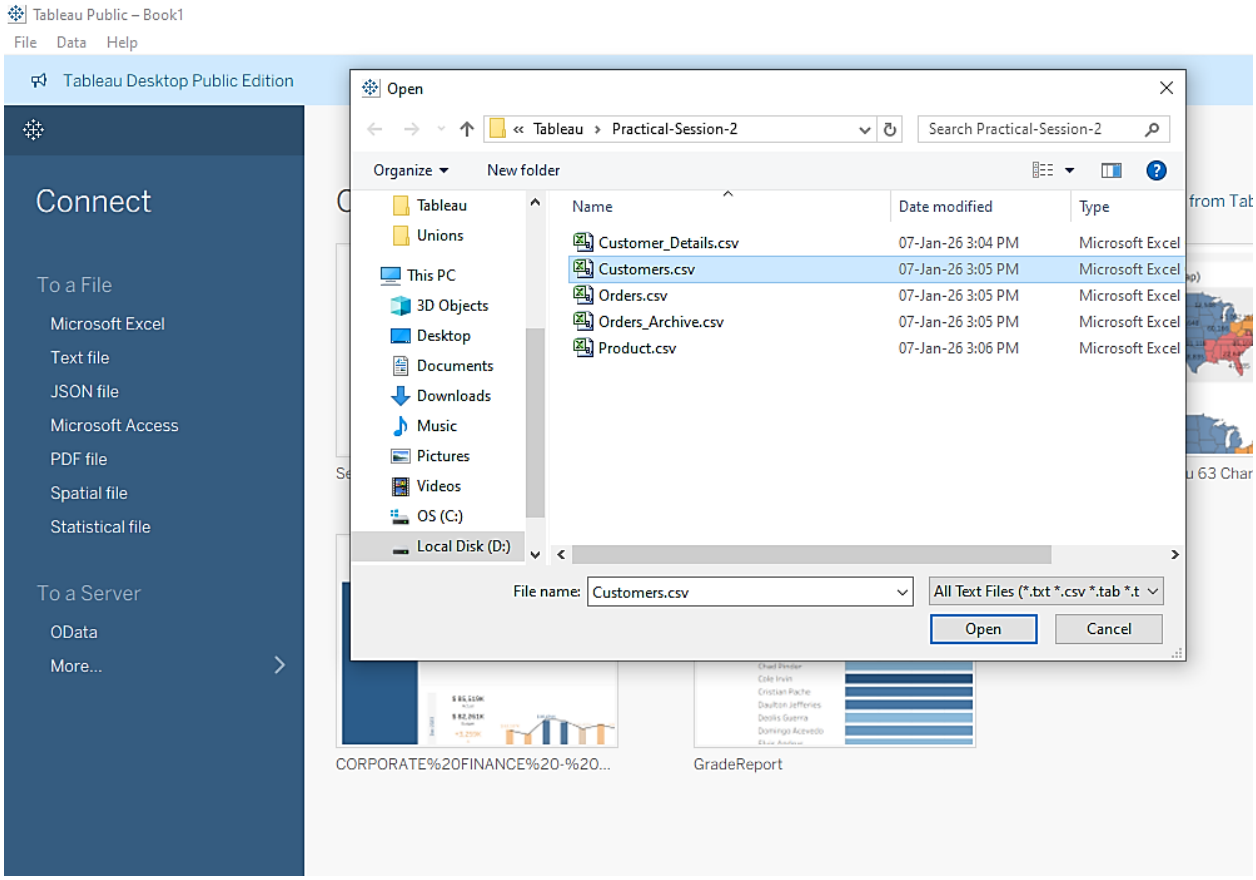
Tableau Public – Book1
File Data Help
Tableau Desktop Public Edition
Connect
To a File
Microsoft Excel
Text file
JSON file
Microsoft Access
PDF file
Spatial file
Statistical file
To a Server
OData
More...
Open
Session_2
KPIs-Dashboard
Budget Controlling
CORPORATE%20FINANCE%20-%20...
GradeReport

Step 3: Browse and Load the File

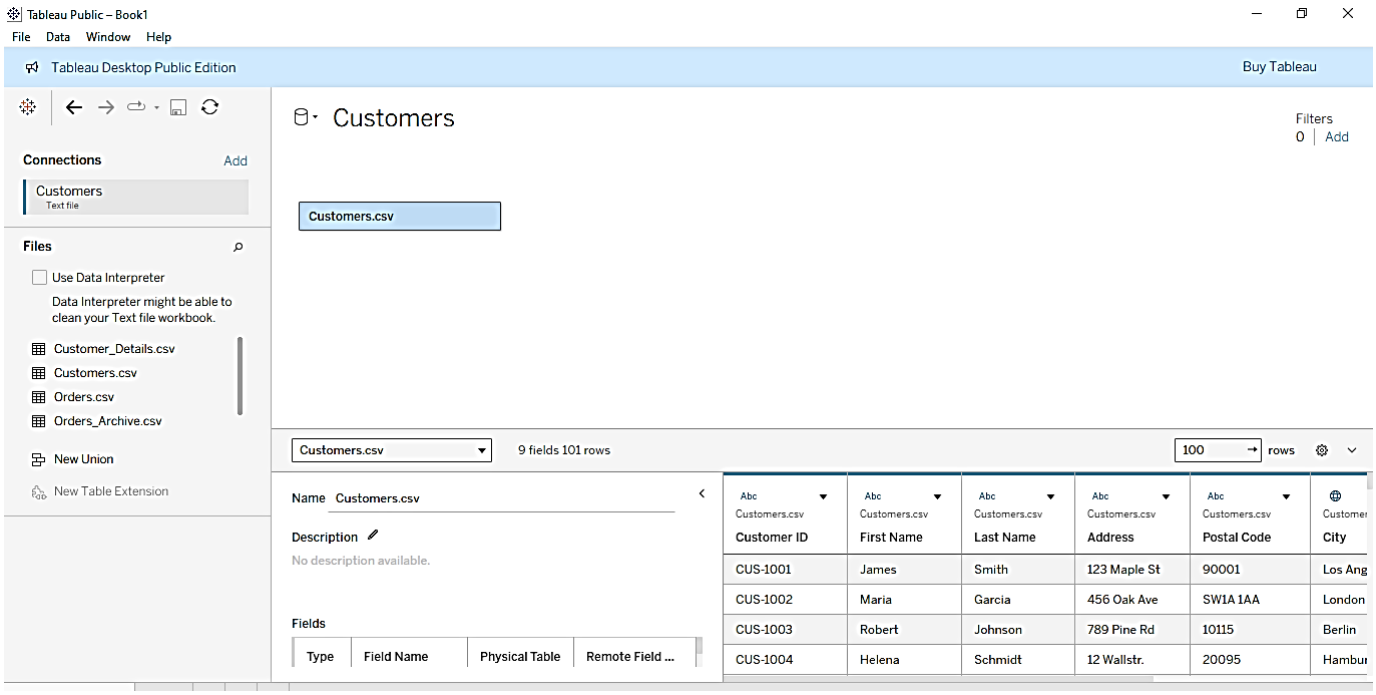
Once selecting Text file, a file dialog box will appear



Step 4: Select Customers.csv and click open.

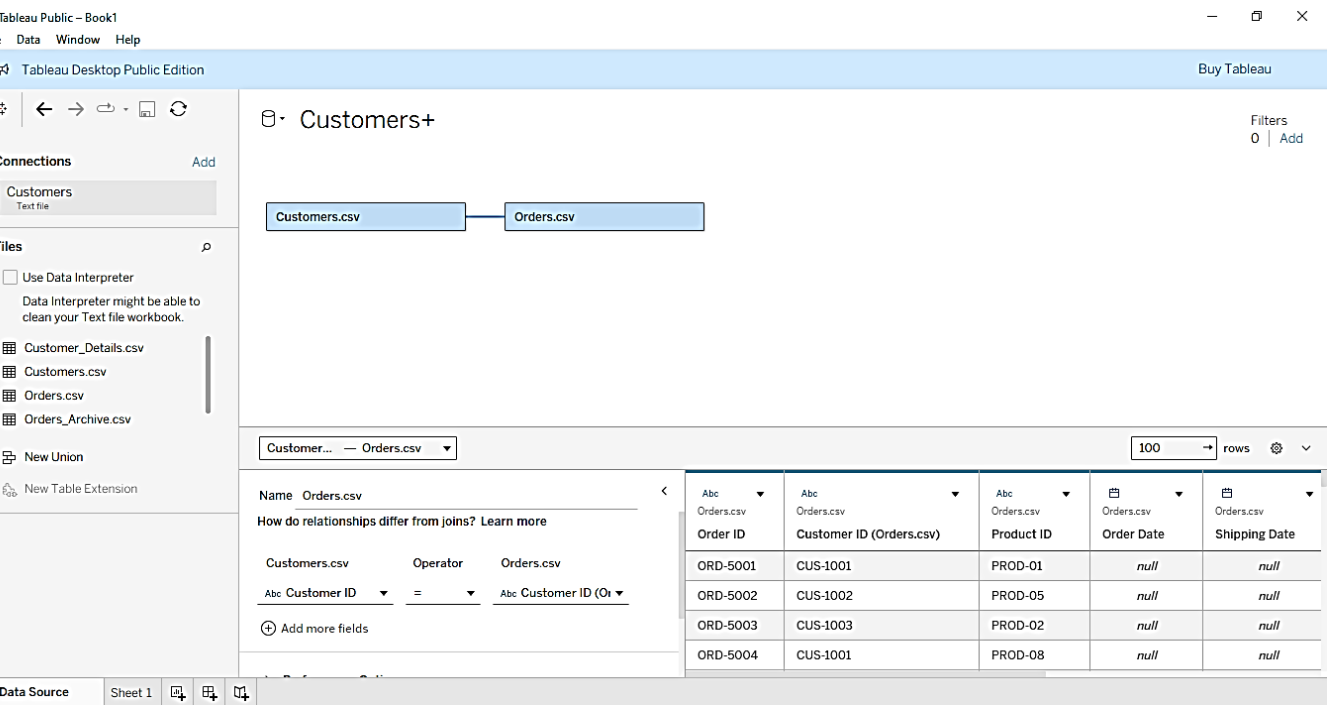


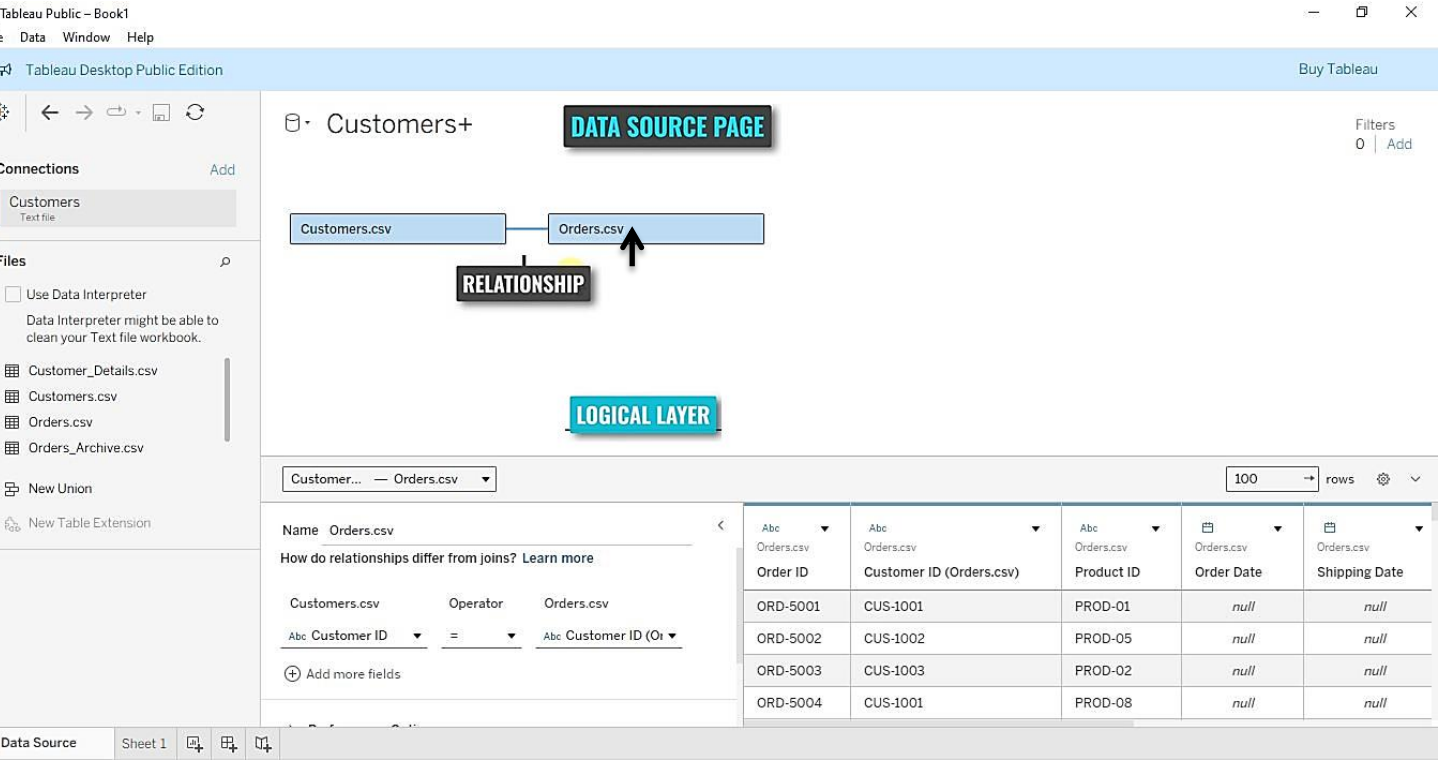
Step 5: In the Data Source Page, Customers.csv is loaded and left hand side of Files section, you can view all the four files Customer_Details.csv, Customers.csv, Orders.csv, Product.csv



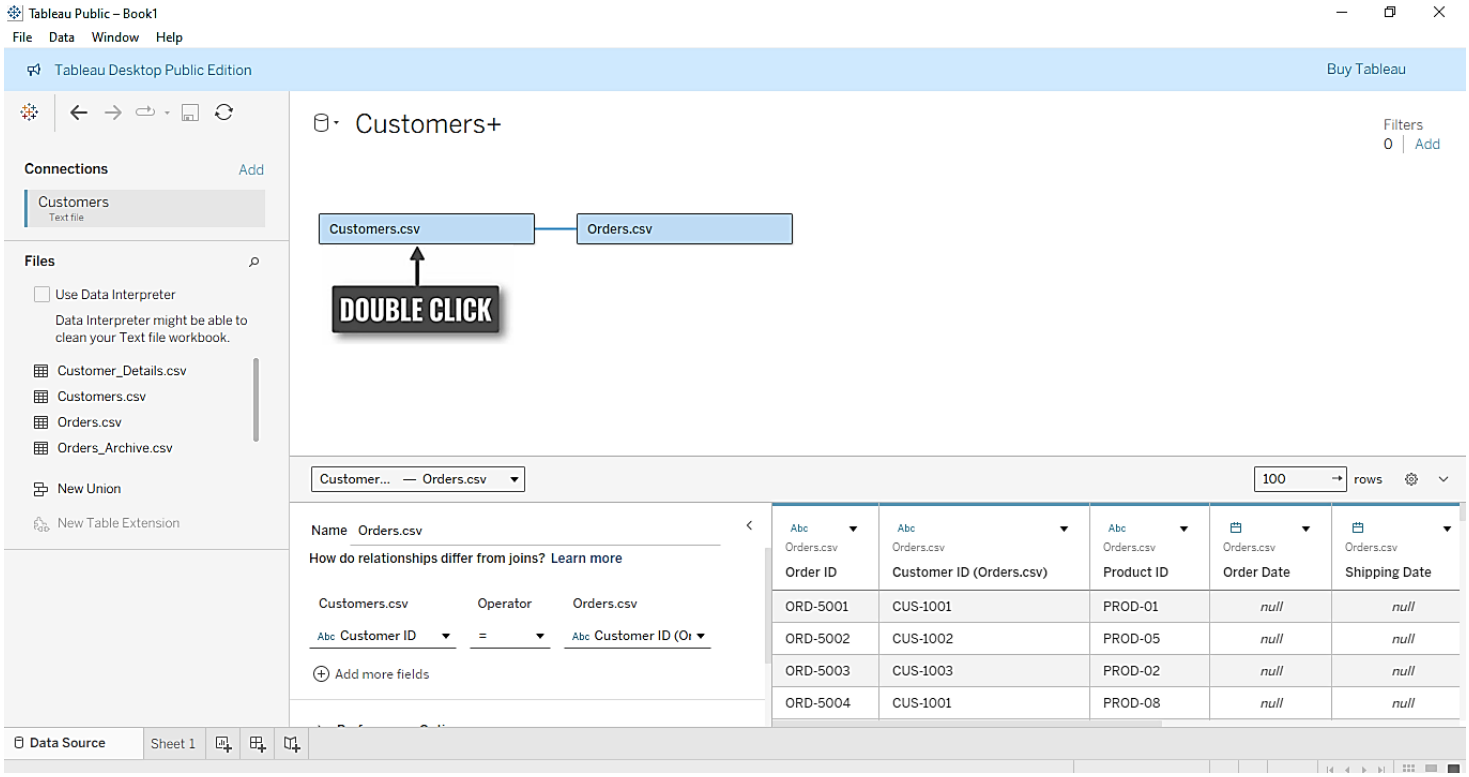
Join Operation

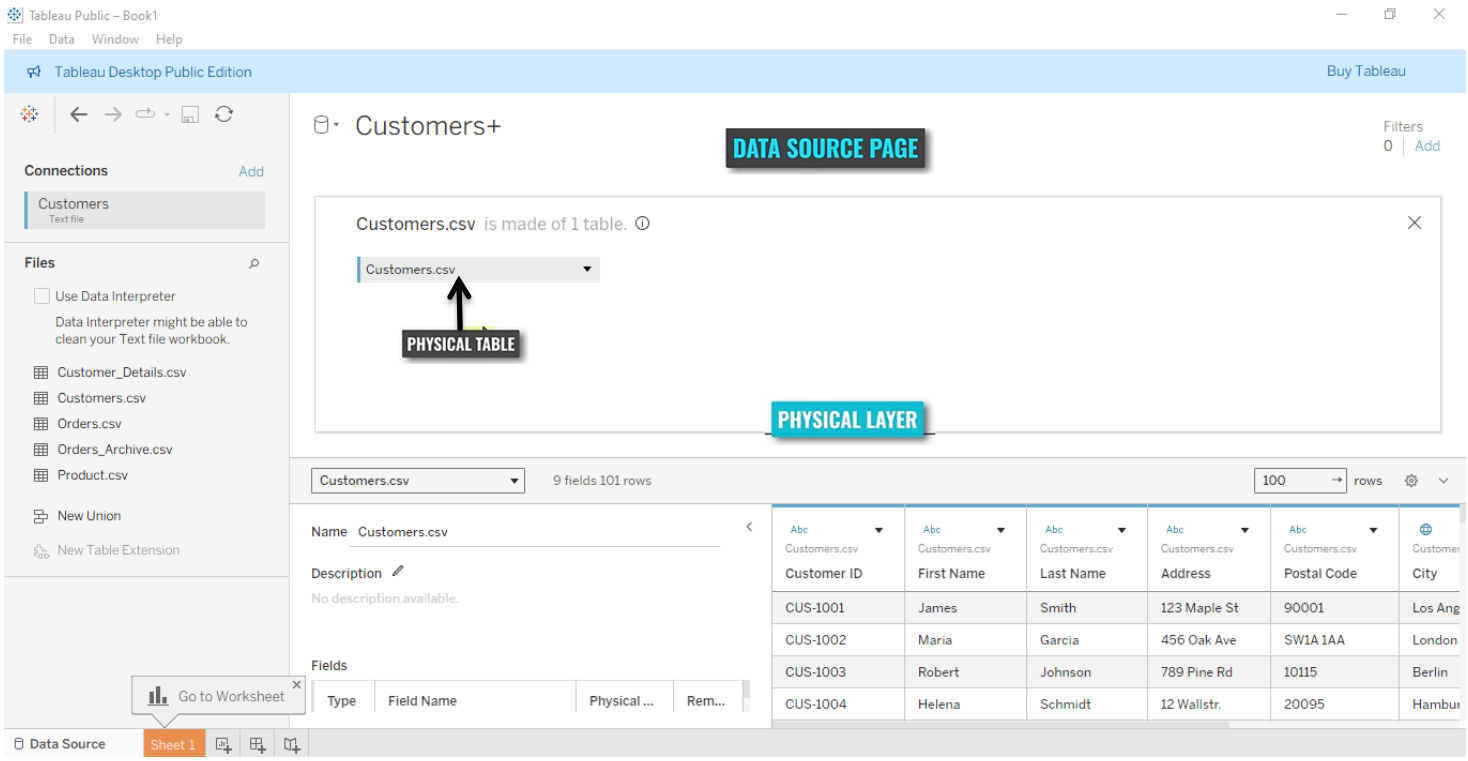
Step 6: Drag and Drop Orders.csv in the data source page with Customers.csv. Now You can find the relationship between Customers.csv and Orders.csv in Logical Layer.



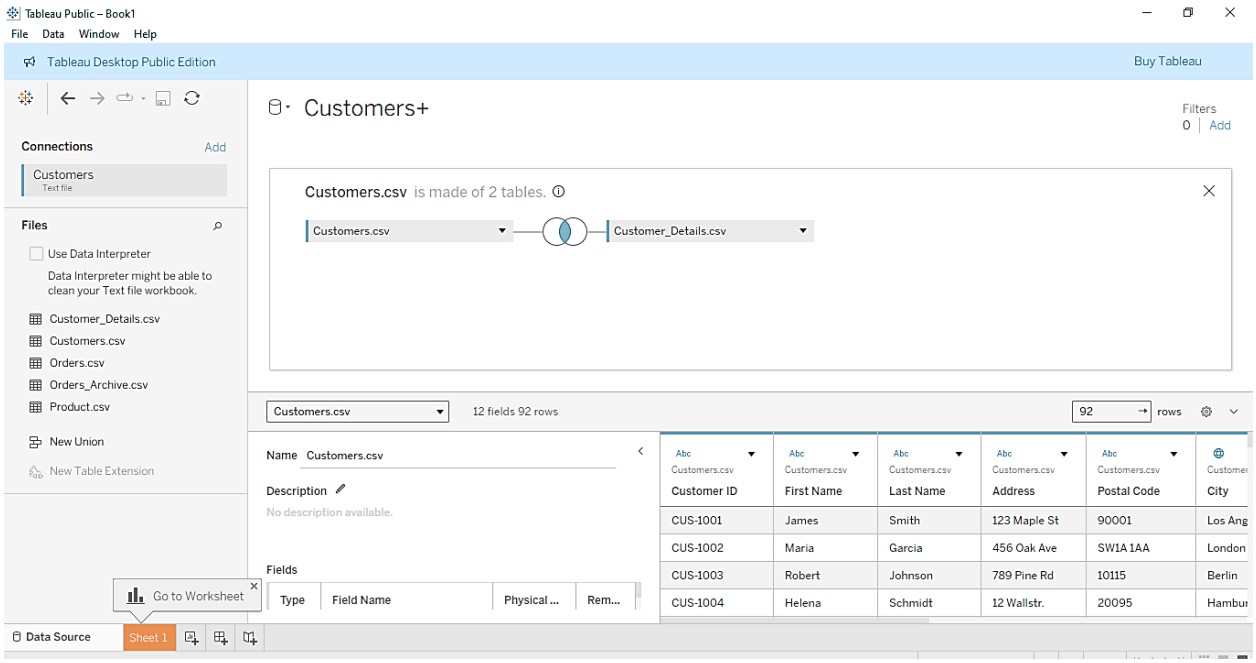


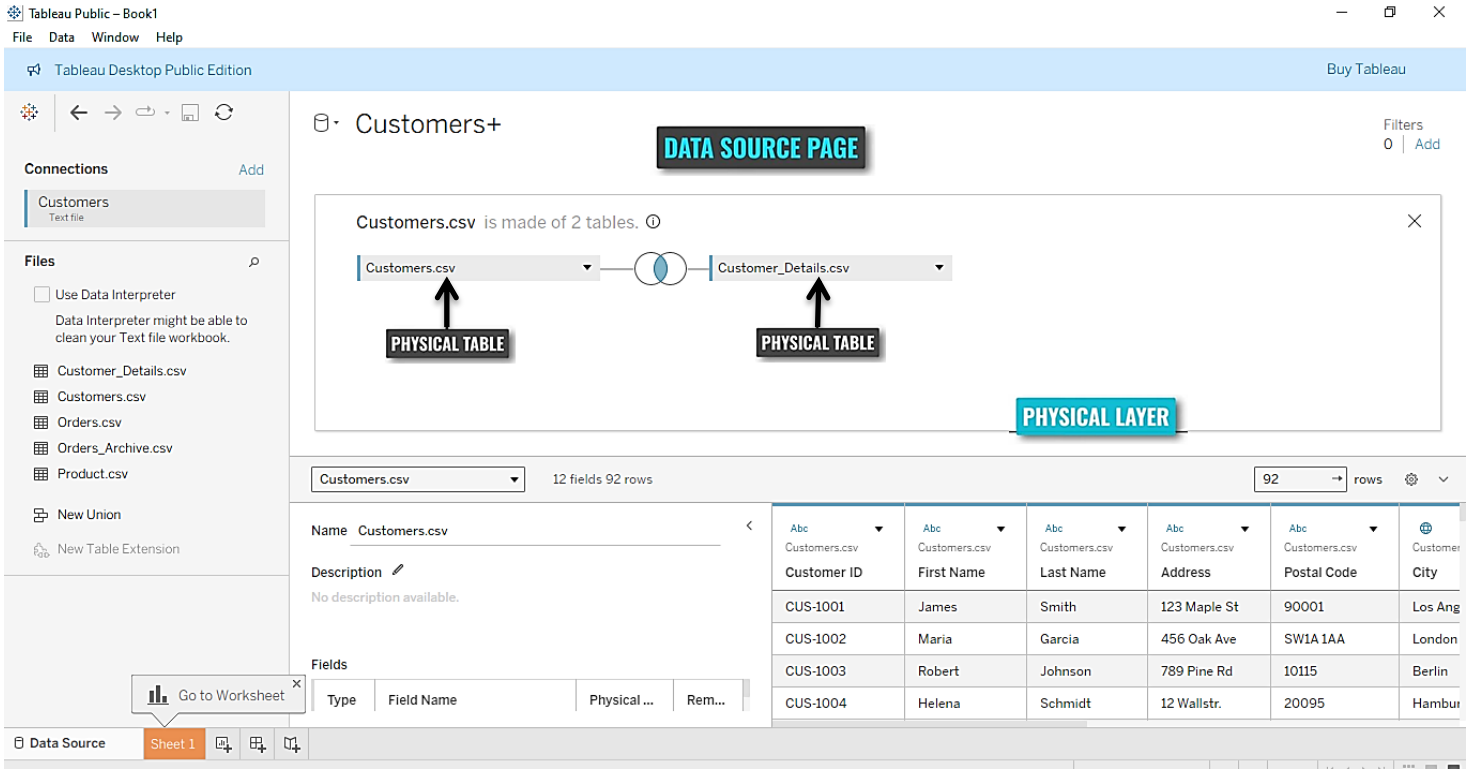
Step 7: Double Click on Customer.csv file in the Logical layer, in order to view Physical Layer.

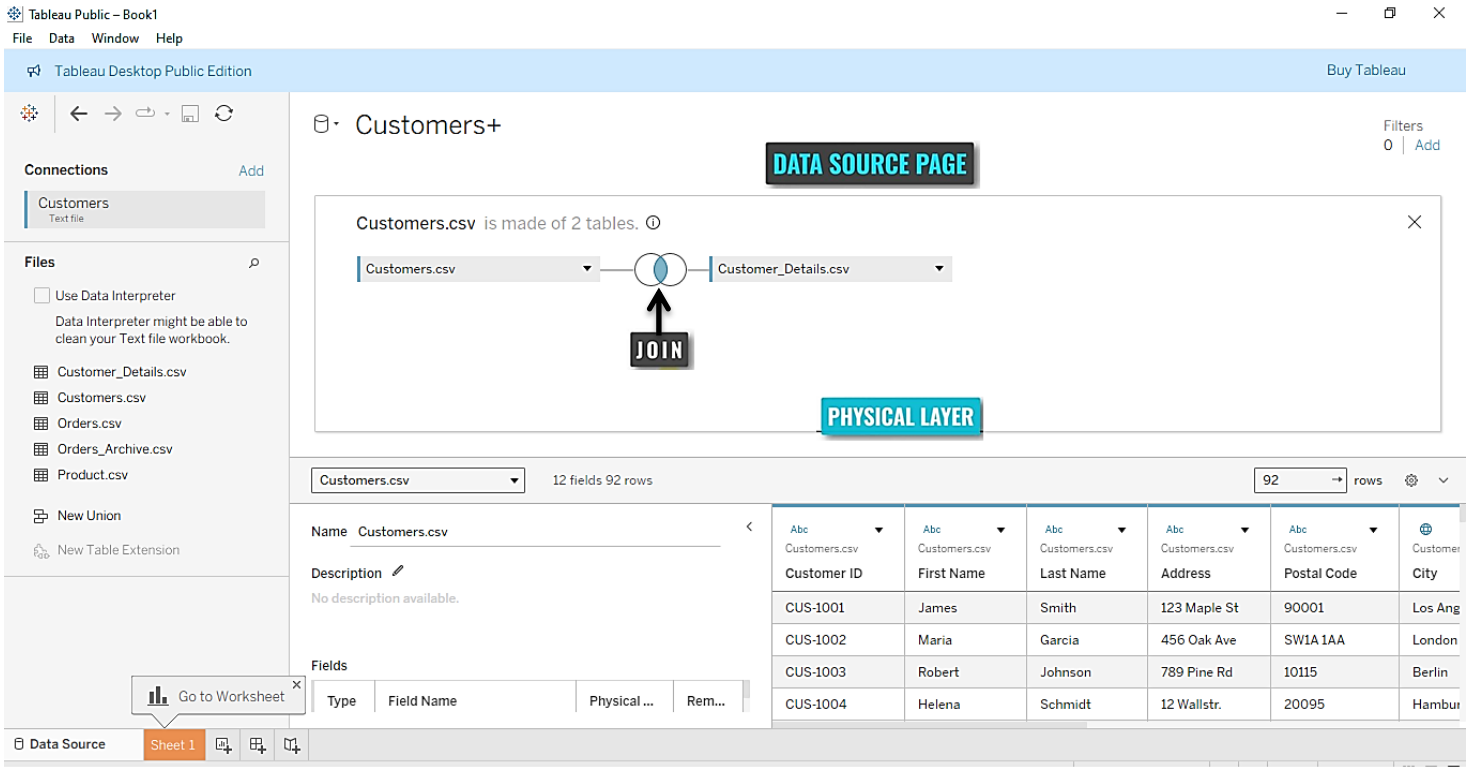




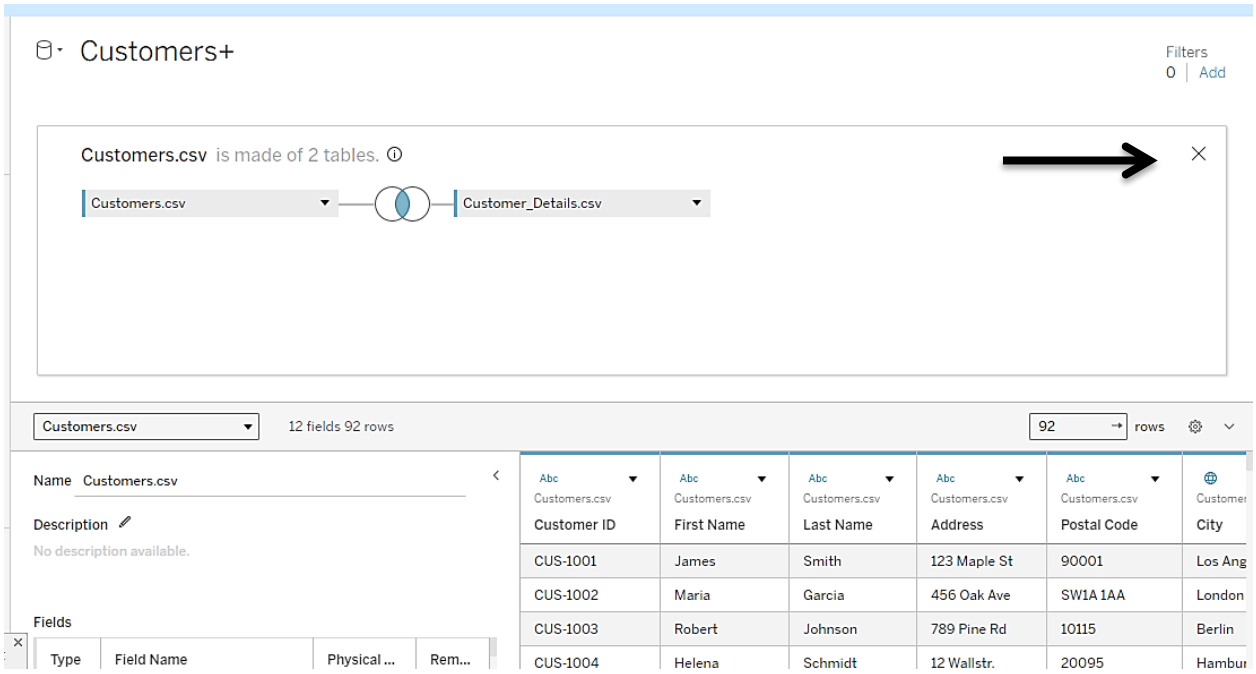
Step 8: Drag and Drop Customers_Details.csv in the physical layer to perform Join operation.



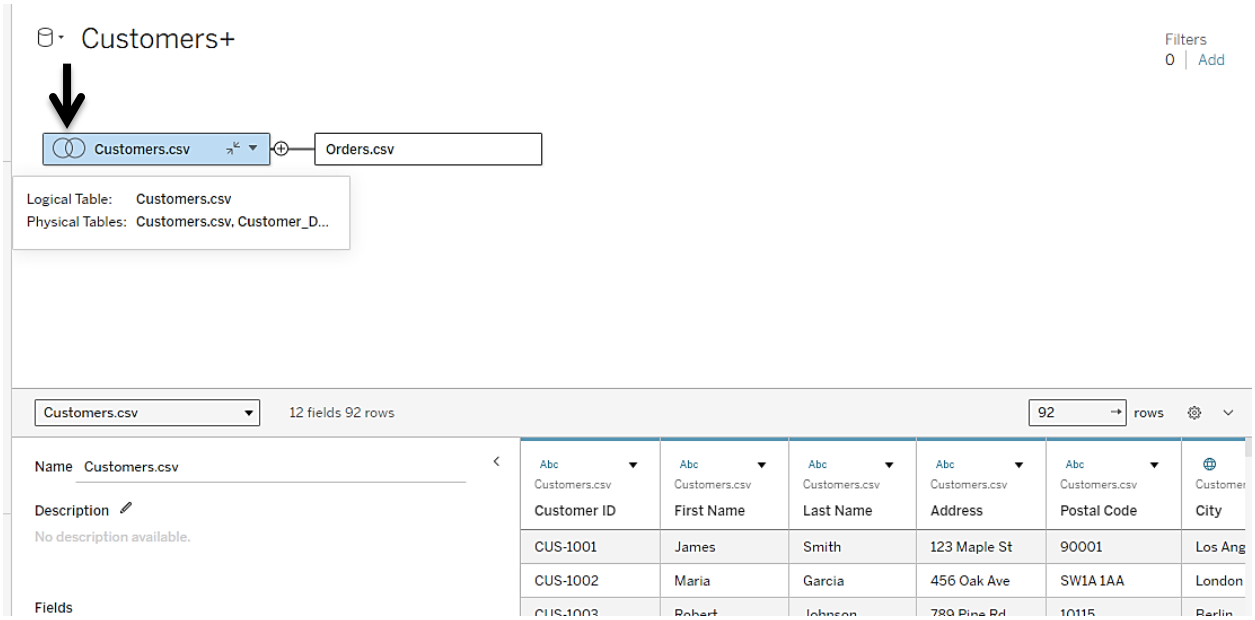




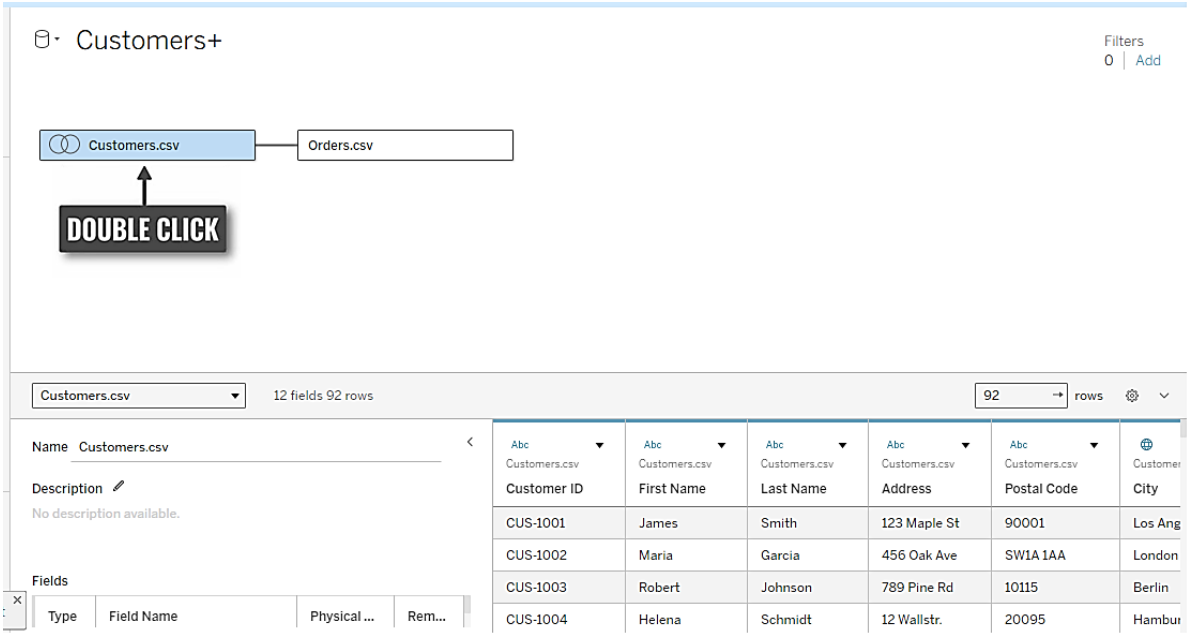
Step 9: Close the Physical Layer to get back to Logical Layer.

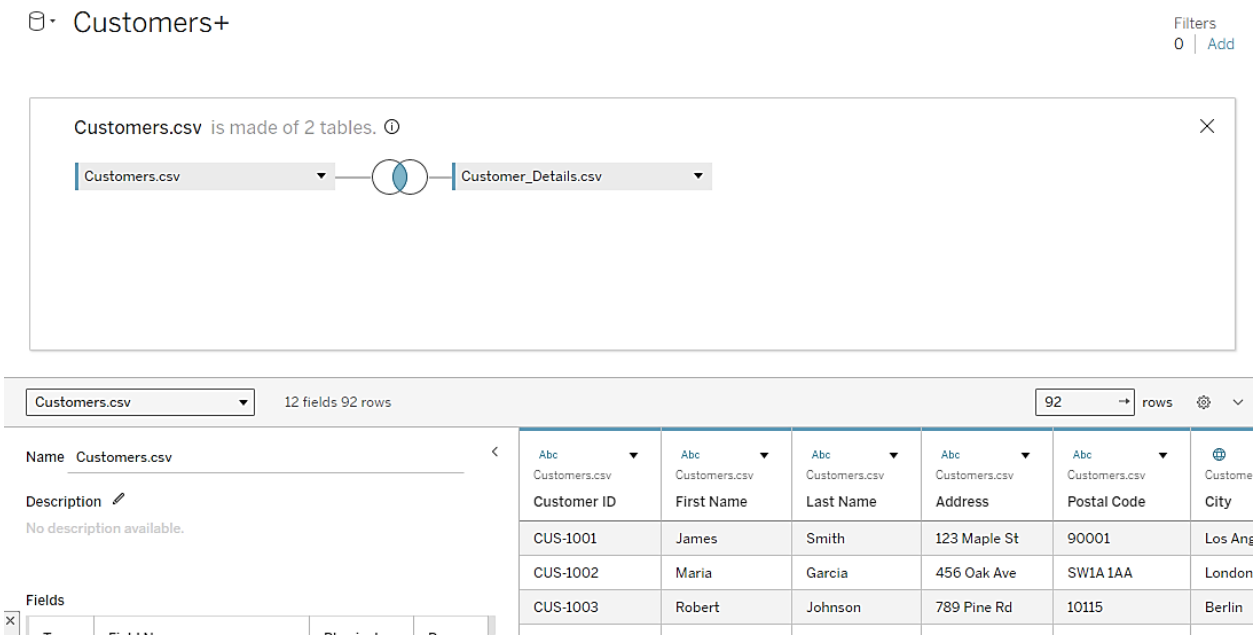


Step 10: Now in Logical Layer Customers.csv is displayed with Join icon. Customers.csv contains one Logical table Custermers.csv. Customers.csv contains two Physical table such as Customers.csv and Customers_Details.csv.

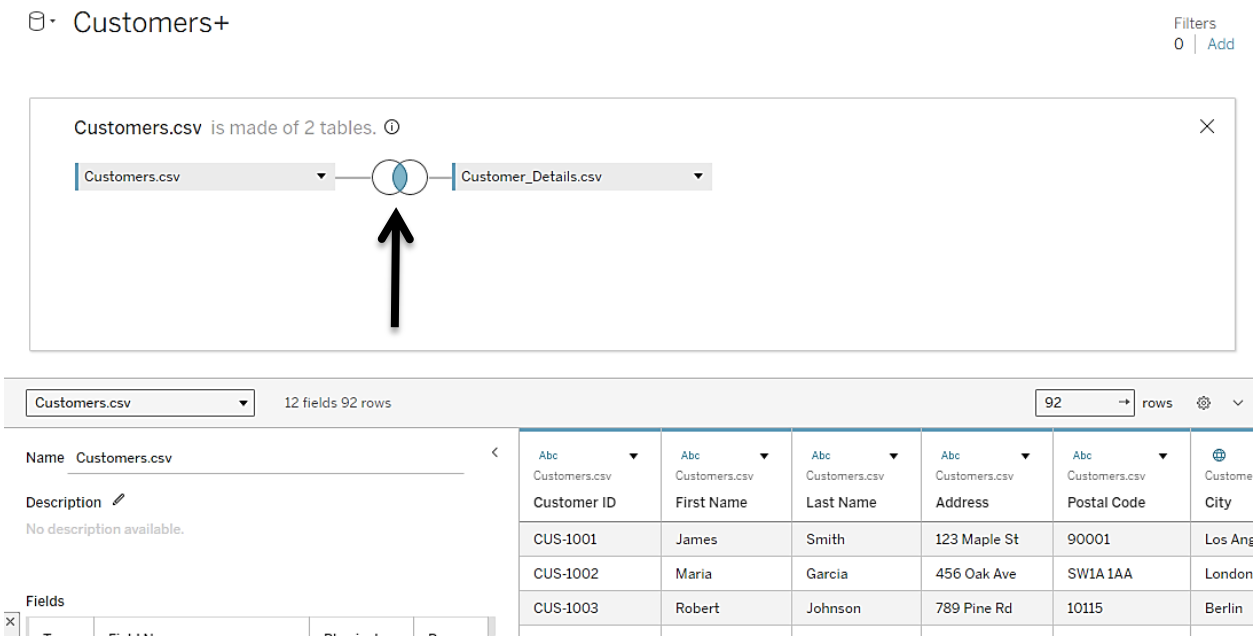


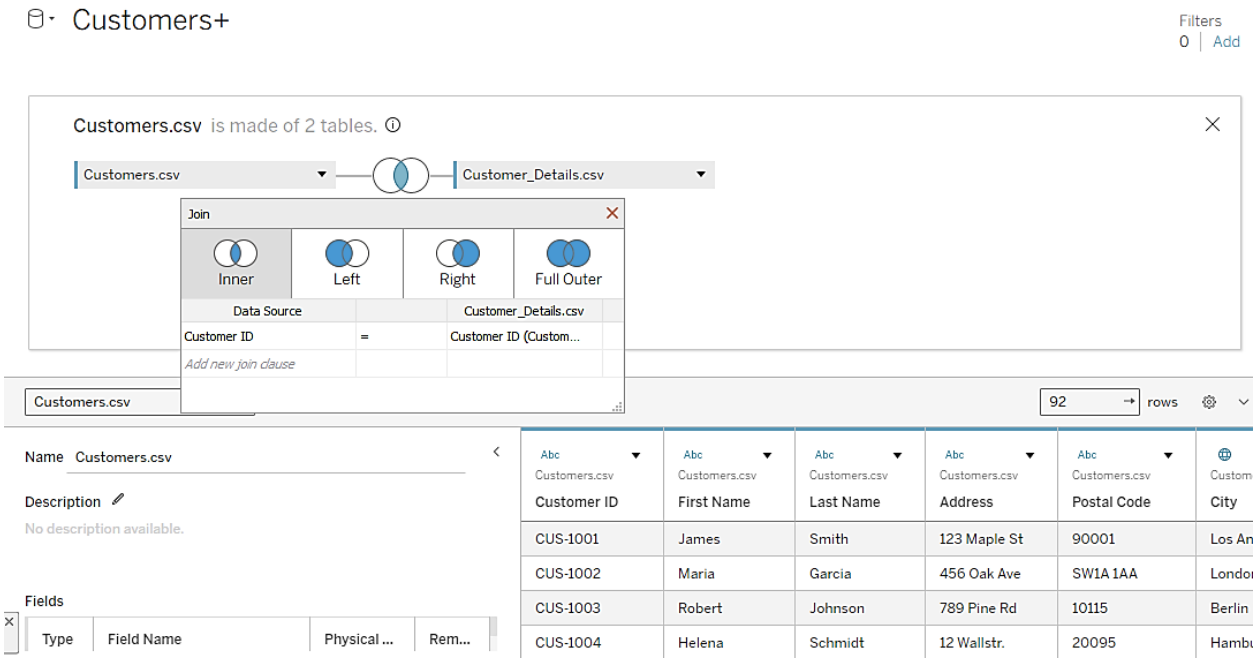
Step 11: Double Click on Customers.csv to apply different join operations in physical layer.



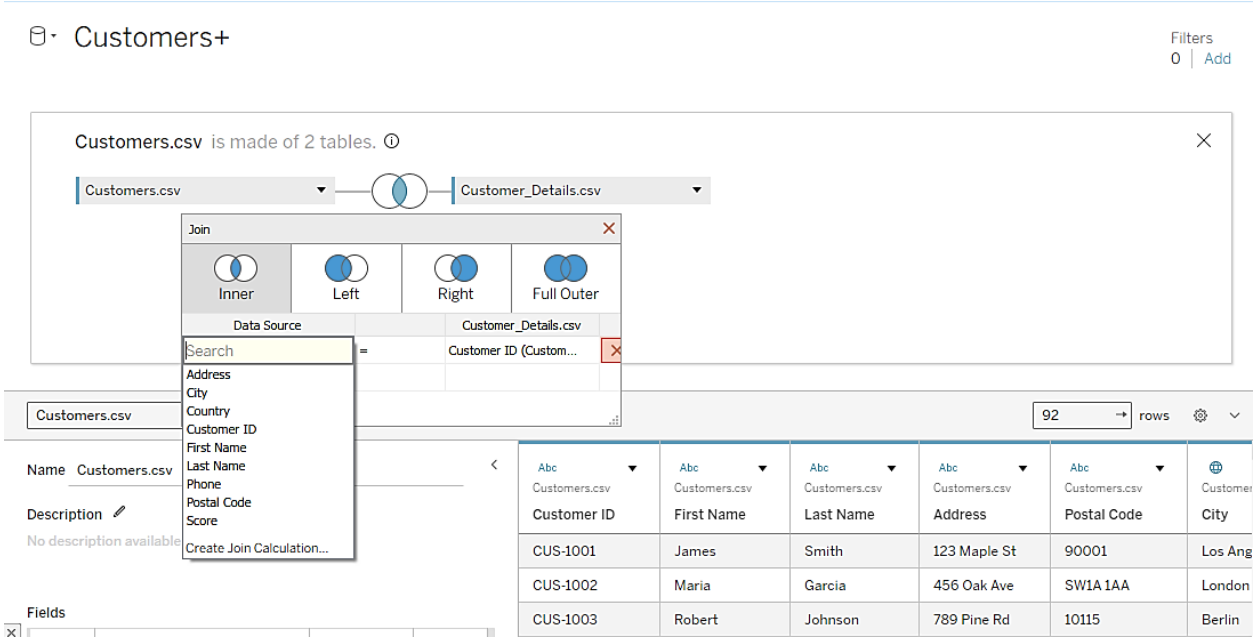


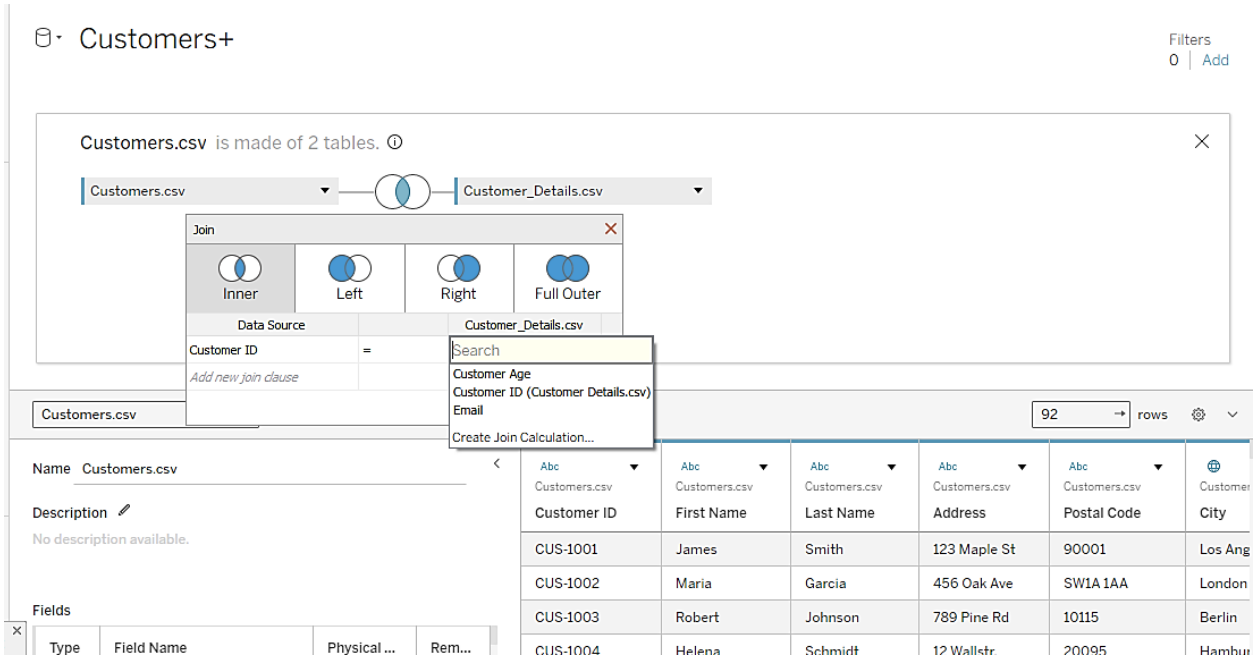
Step 12: Click on Join to select different join operations.





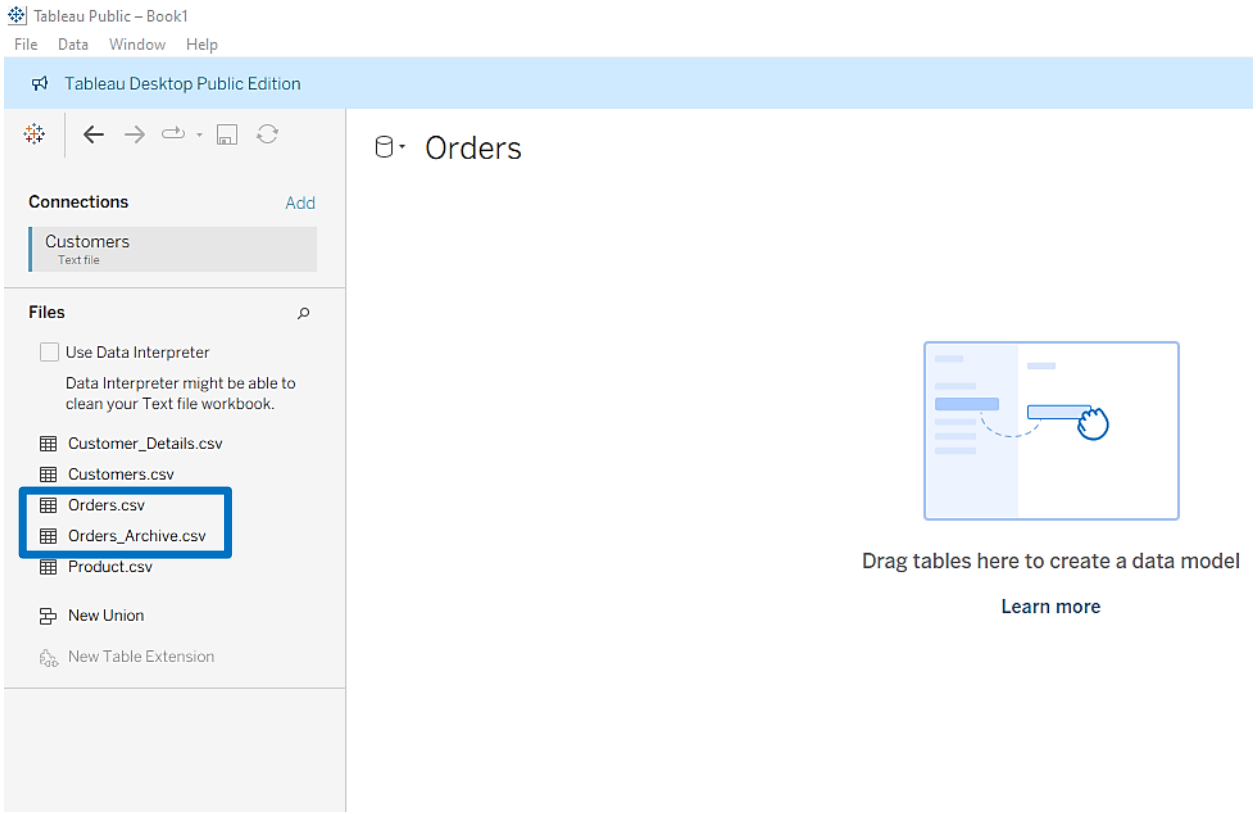
Step 13: You can select any operation on join, and you can change the key value to any value but only matching with two files. By default it is Customer ID, so leave it as it is.



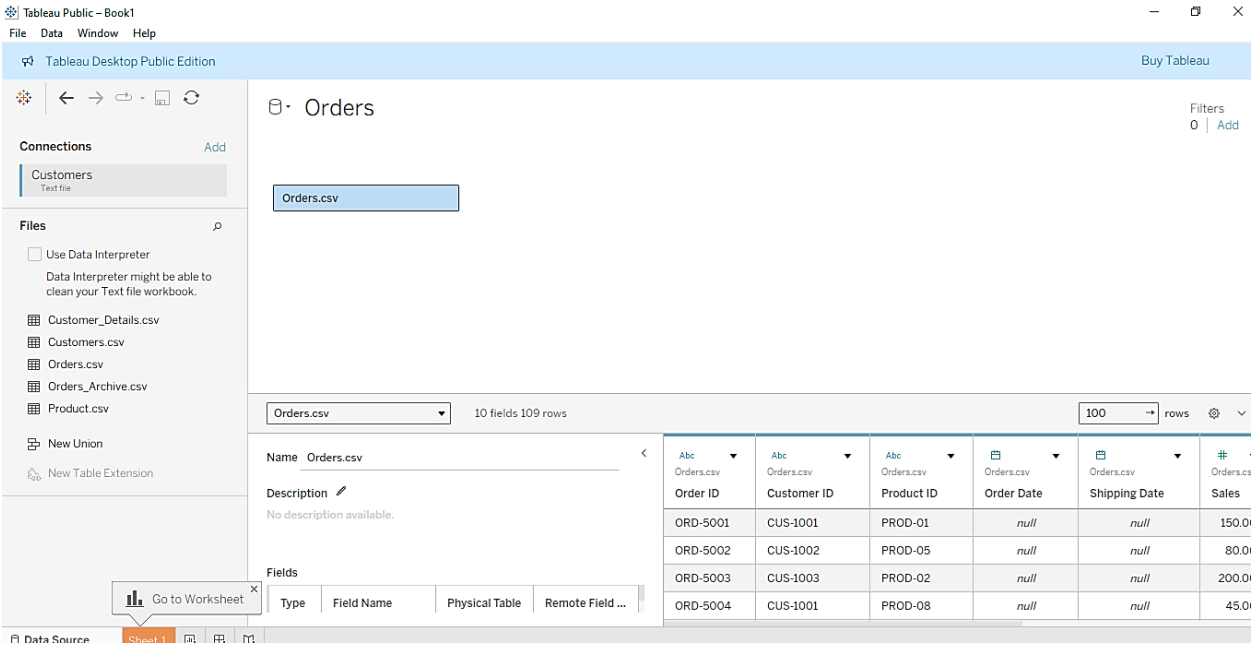


Union Operation

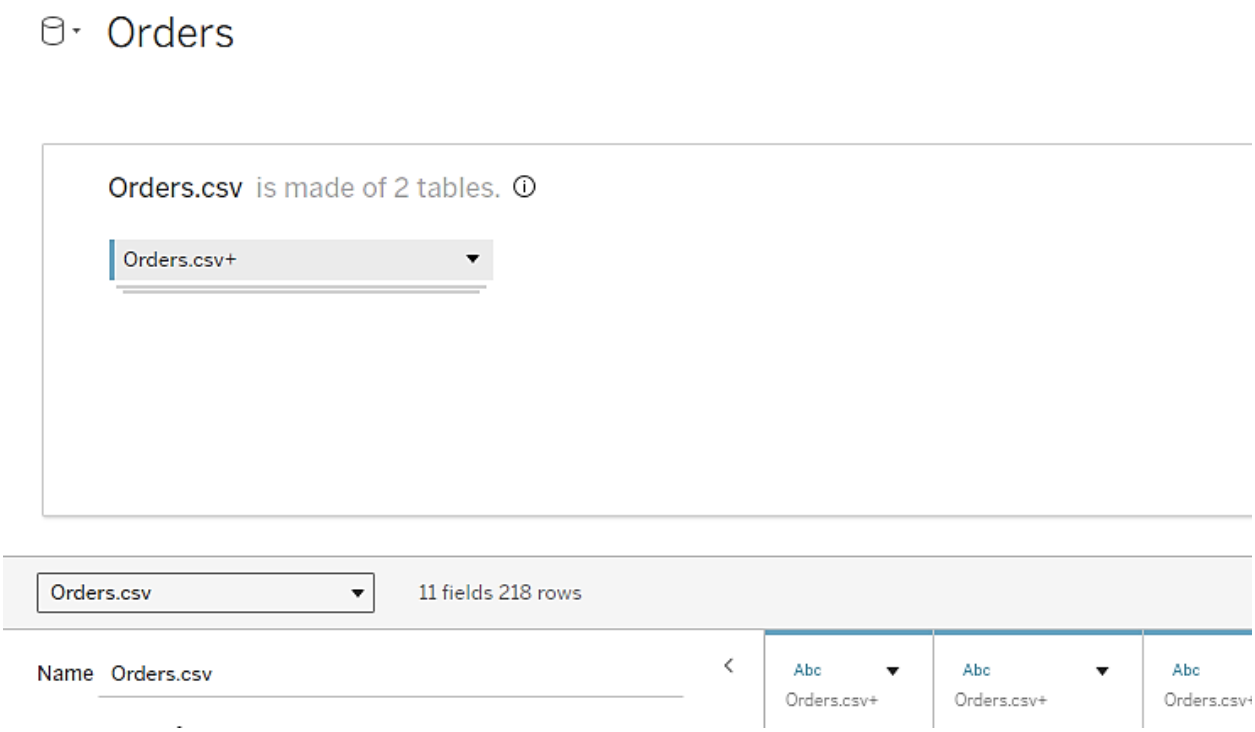
Step 1: Select the files in a way that both files have i) Same Number of Fields and ii) Same Data Types to perform Union Operation. So we are selecting Orders.csv and Orders_Archive.csv which have same number of fields and same data types.



Step 2: Drag and Drop Orders.csv file.



Step 3: Drag and Drop Orders_Archieve.csv file to the Orders.csv file.

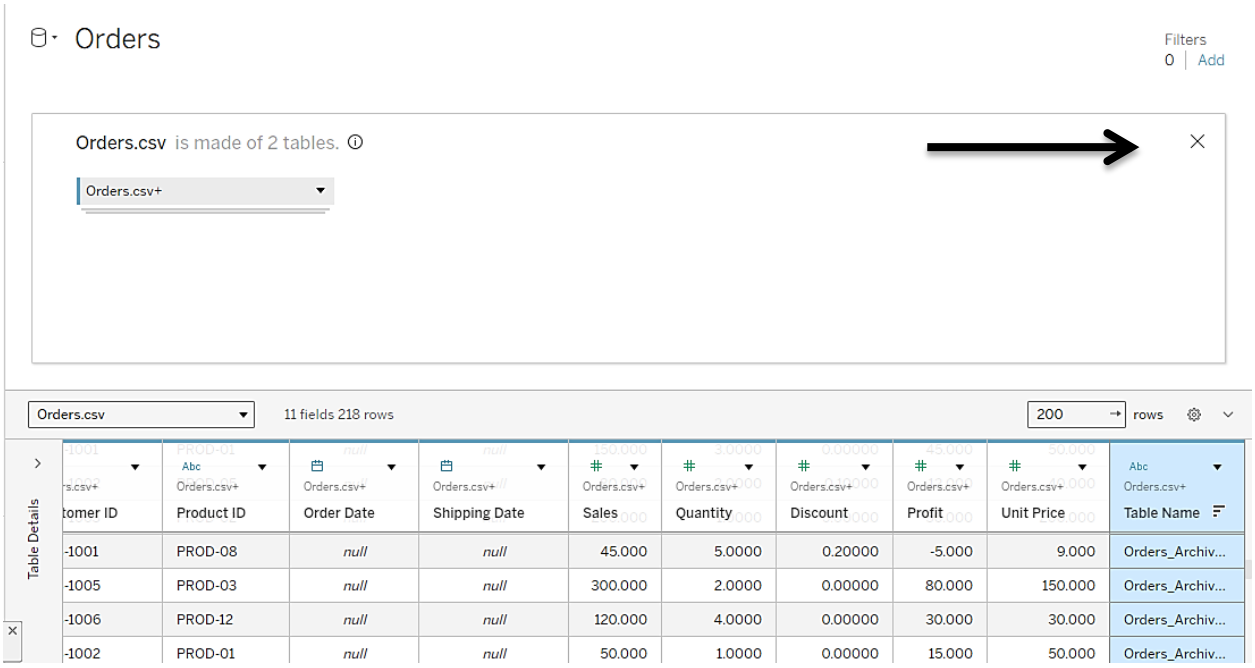


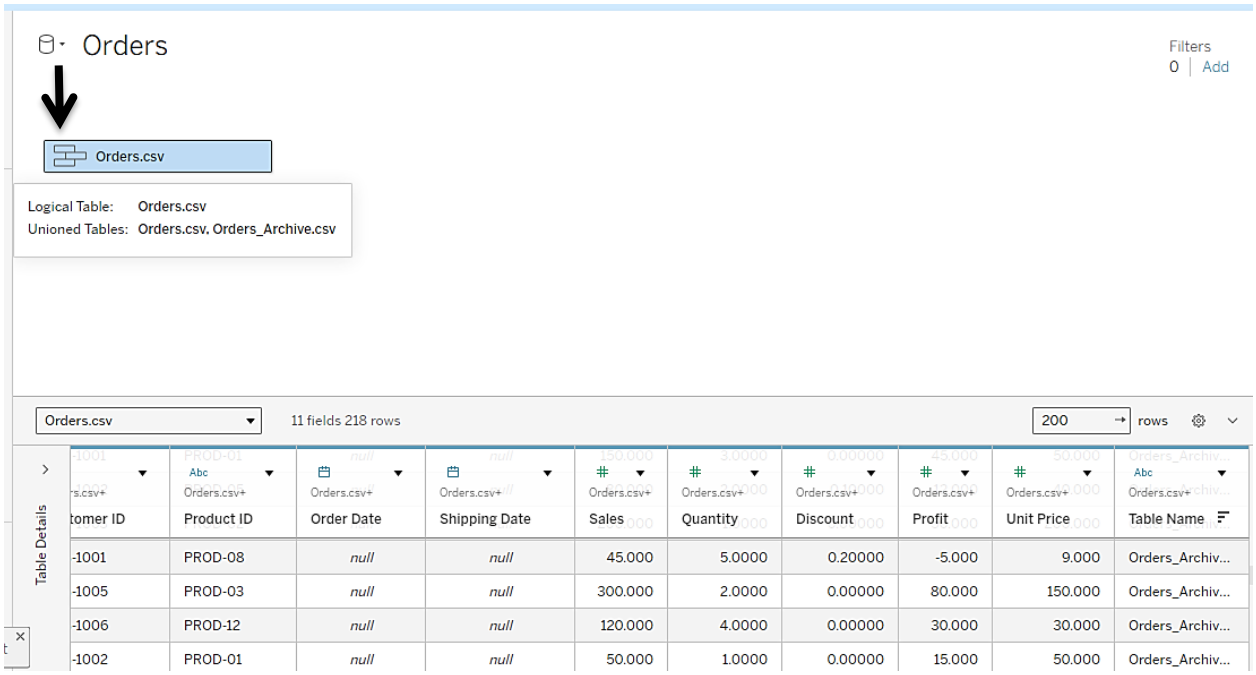
Step 4: We can view the Physical table at the end of the Table Name to check whether two tables are combined or not.

Orders.csv+ Shipping Date	Orders.csv+ Sales	Orders.csv+ Quantity	Orders.csv+ Discount	Orders.csv+ Profit	Orders.csv+ Unit Price	Orders.csv+ Table Name
06/01/2022	215,88	1	0,400000	35,7700	215,480	ders.csv
20/01/2022	1.690,89	10	0,600000	30,0000	168,480	Orders.csv
22/01/2022	39,97	2	*null*	60,6700	19,980	Orders.csv
30/01/2022	500,55	30	0,200000	35,2200	16,480	Orders.csv
10/02/2022	164,22	5	0,000000	-5,7400	32,840	Orders.csv
06/02/2022	430,05	2	0,000000	20,7500	215,020	Orders.csv
13/02/2022	90,99	6	0,300000	*null*	14,860	Orders.csv
28/02/2022	375,14	15	0,200000	20,4400	24,800	Orders.csv

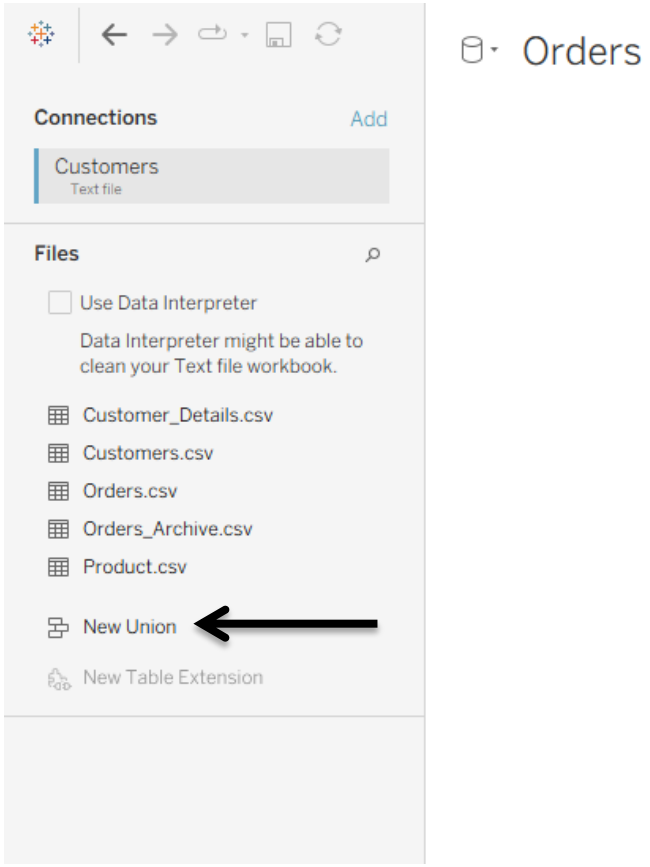
Orders.csv+ Shipping Date	Orders.csv+ Sales	Orders.csv+ Quantity	Orders.csv+ Discount	Orders.csv+ Profit	Orders.csv+ Unit Price	Abc Orders.csv+ Table Name
28/02/2022	375,14	15	0,200000	20,4400	24,800	Orders.csv
24/02/2022	525,07	35	0,690000	40,6600	14,310	Orders.csv
03/03/2022	56,11	12	0,000000	*null*	4,670	Orders.csv
03/11/2021	220,88	1	0,400000	35,7700	220,480	Orders_Archiv
18/11/2021	1.490,89	10	0,600000	30,0000	148,480	Orders_Archiv
20/12/2021	40,90	2	*null*	60,6700	20,450	Orders_Archiv
30/12/2021	450,55	30	0,200000	35,2200	14,810	Orders_Archiv
20/12/2022	214,22	5	0,000000	-5,7400	42,840	Orders_Archiv

Step 5: We can close the view at physical layer and view the new union button at Logical layer.

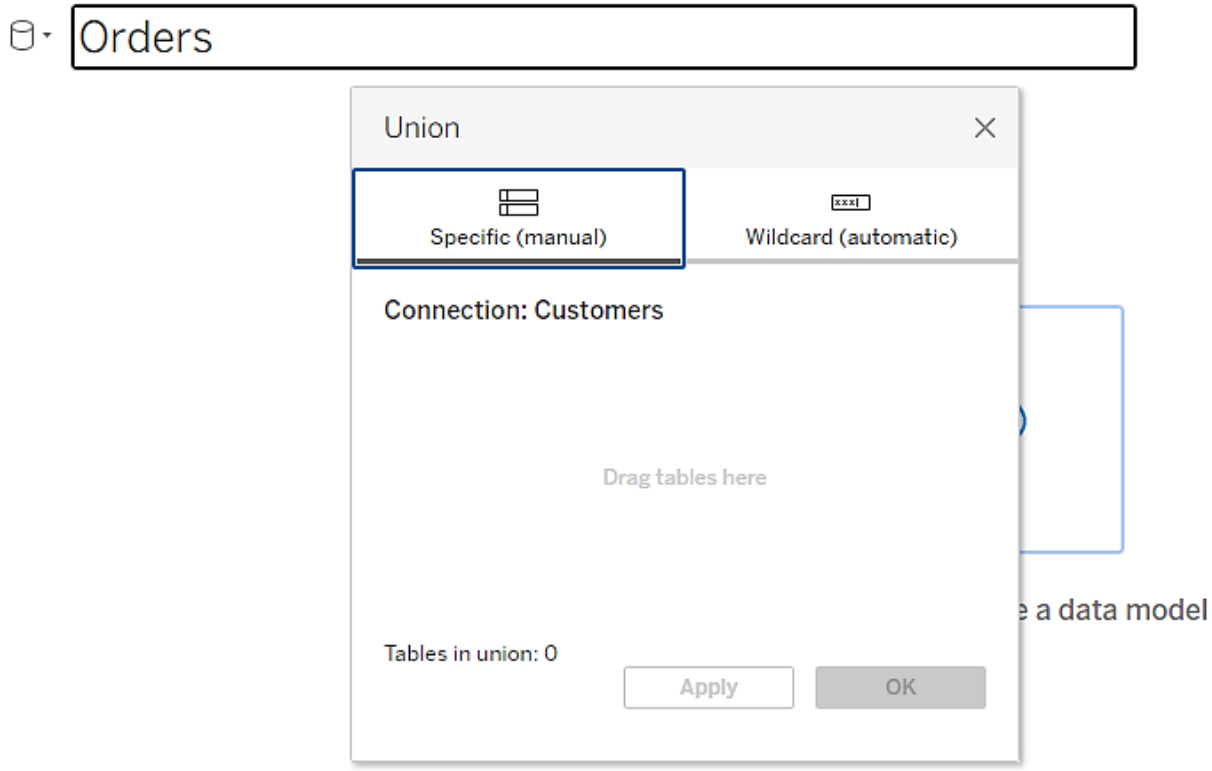




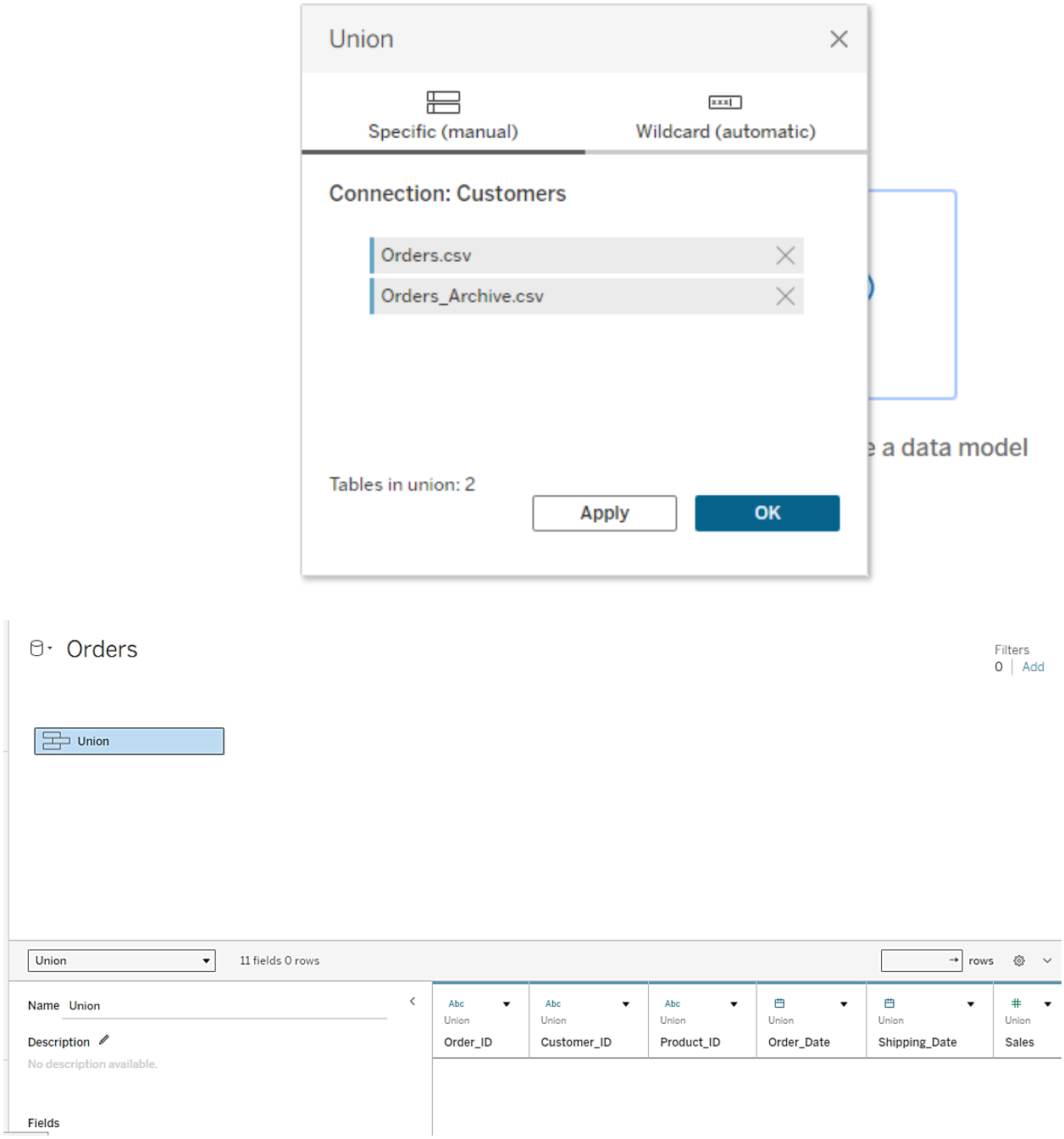
Step 6: Alternate Method for Union operation. Select New Union button at the Connection Pane.



Step 7: New window will be opened to ask whether to perform manually or automatic.

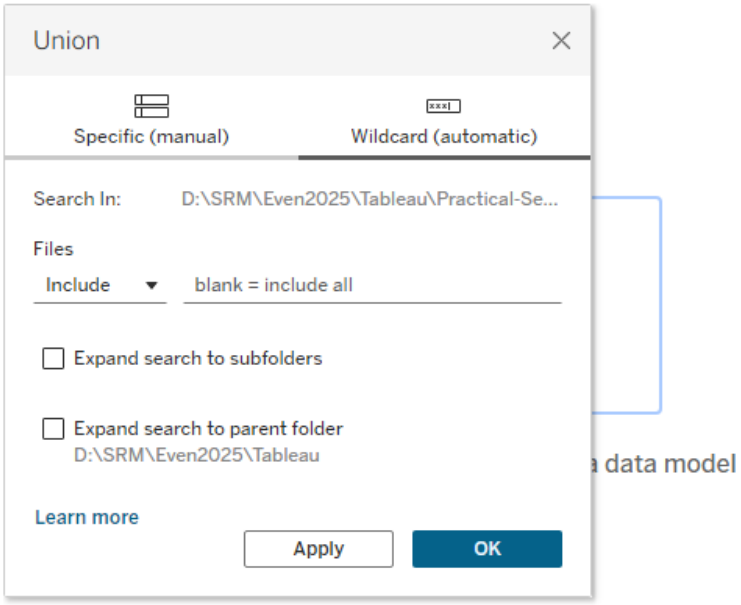


Step 8: For manual, drag and drop the files one by one and click ok for union.

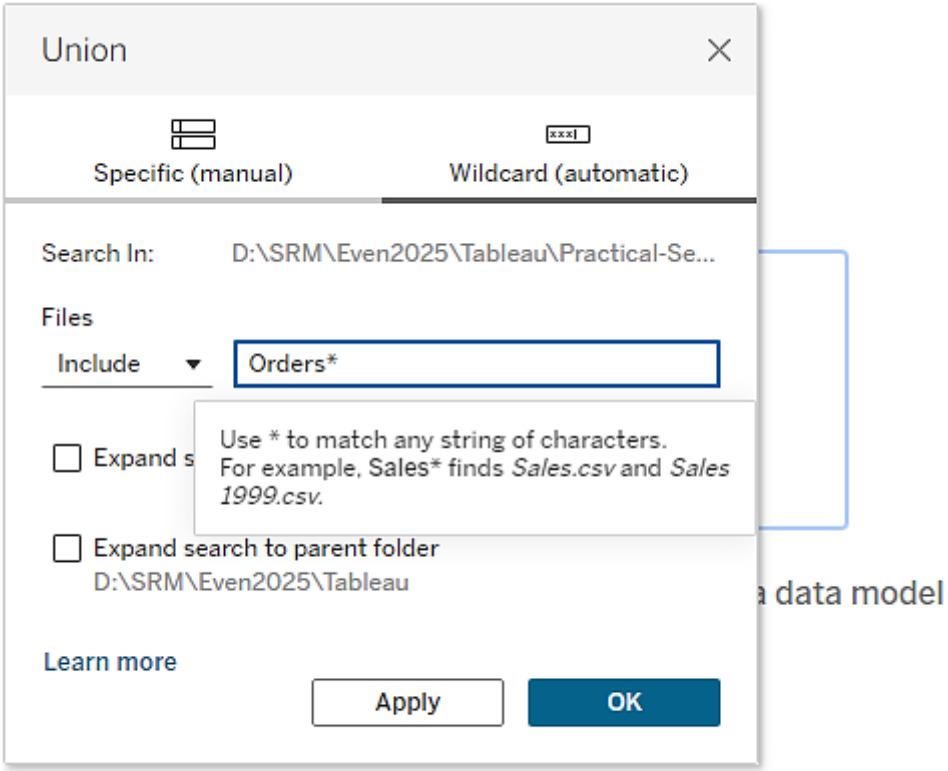


Step 9: For Automatic, Select include button and enter rule and select the option such as either i) expand search to subfolders or ii) Expand search to parent folder or both.

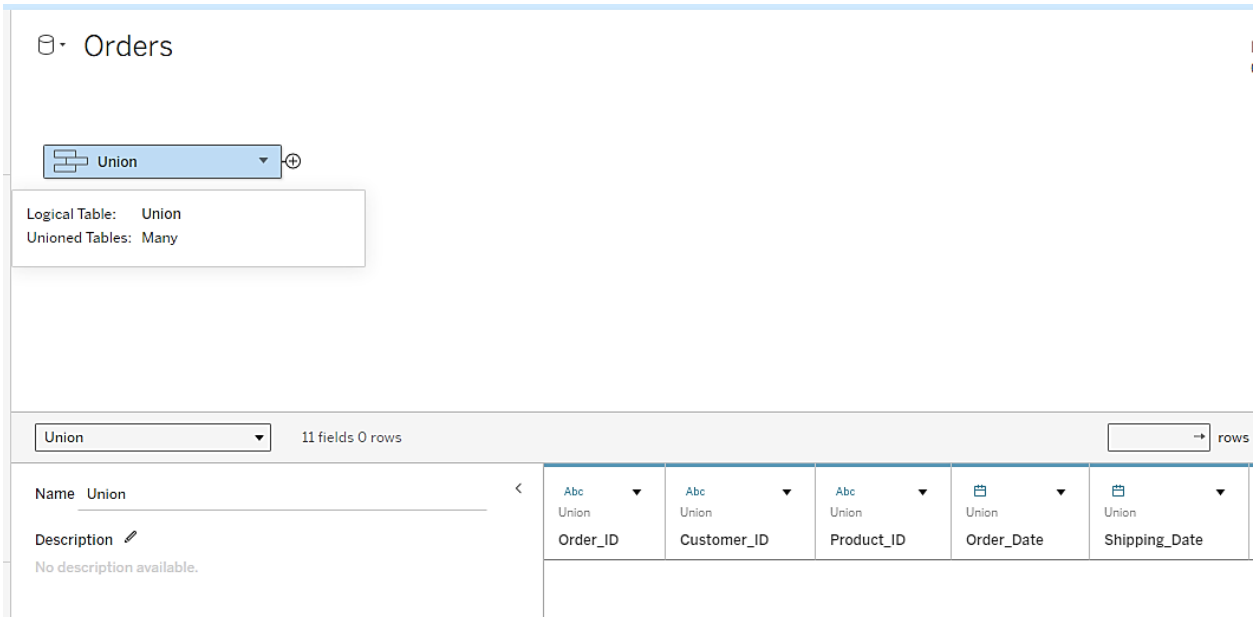
Orders



data model

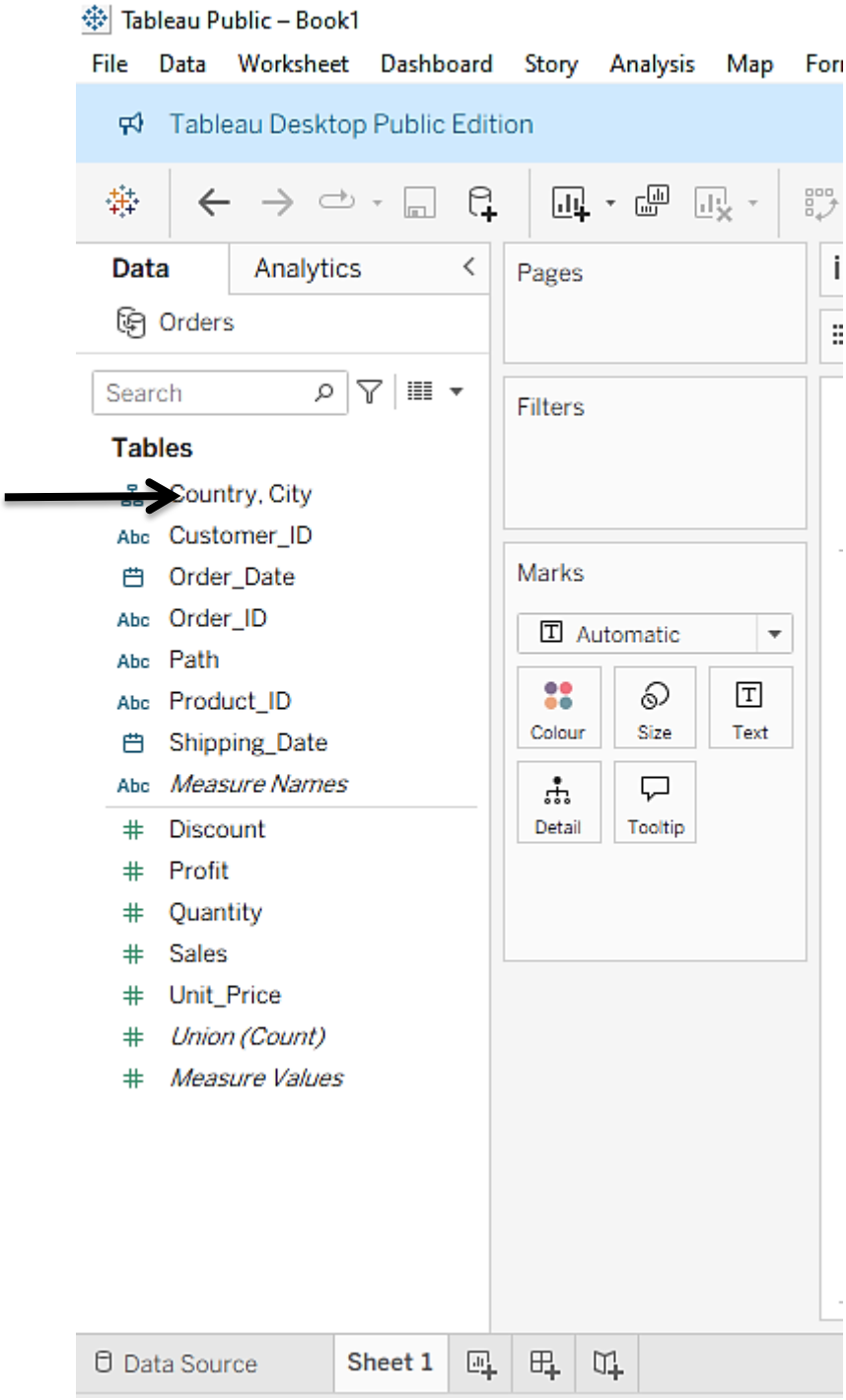


data model



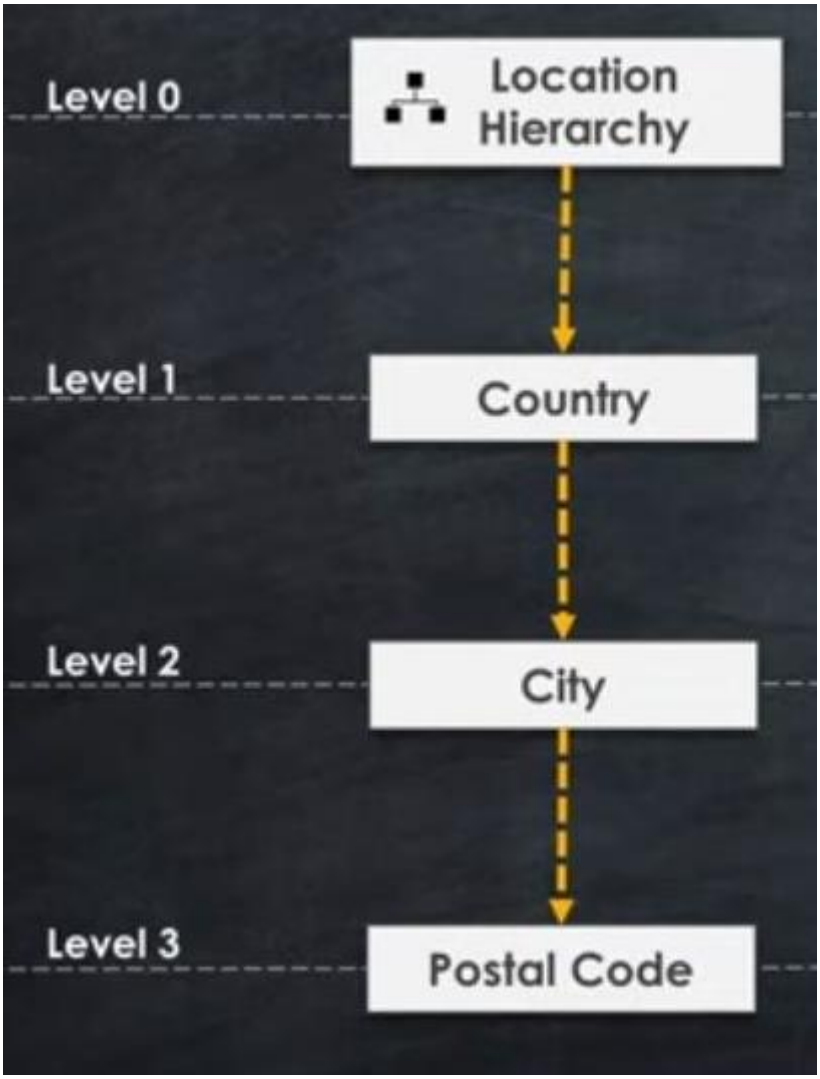
Hierarchies

Just click sheet 1, you will get the Table view, and you can check default hierarchy.



Create Hierarchy

Example: Create Location hierarchy, which includes Country, City and Postal Code.

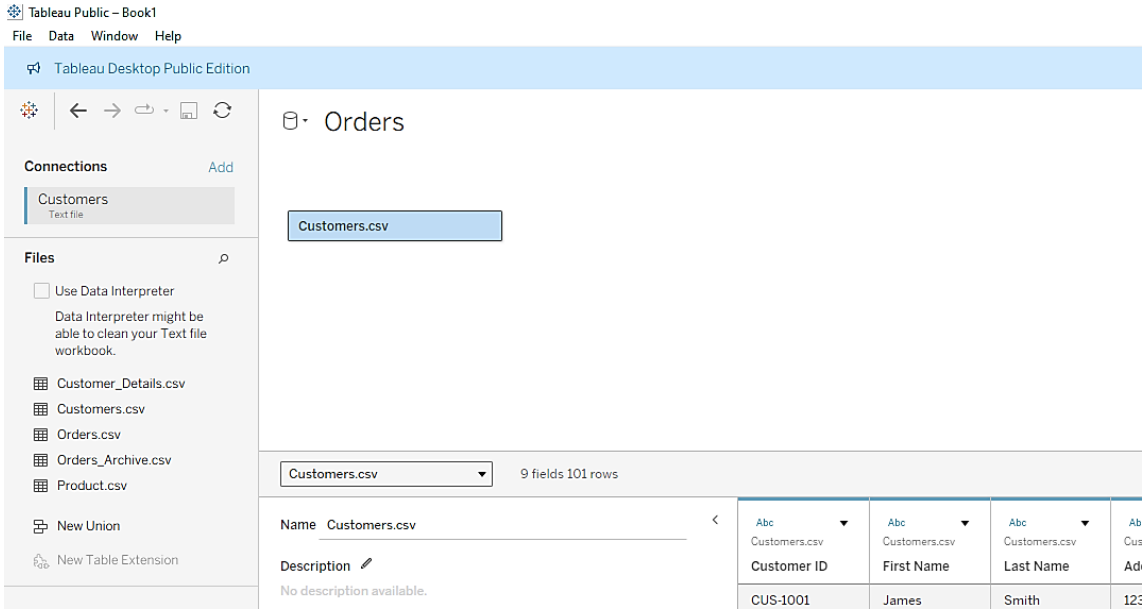


Two Methods:

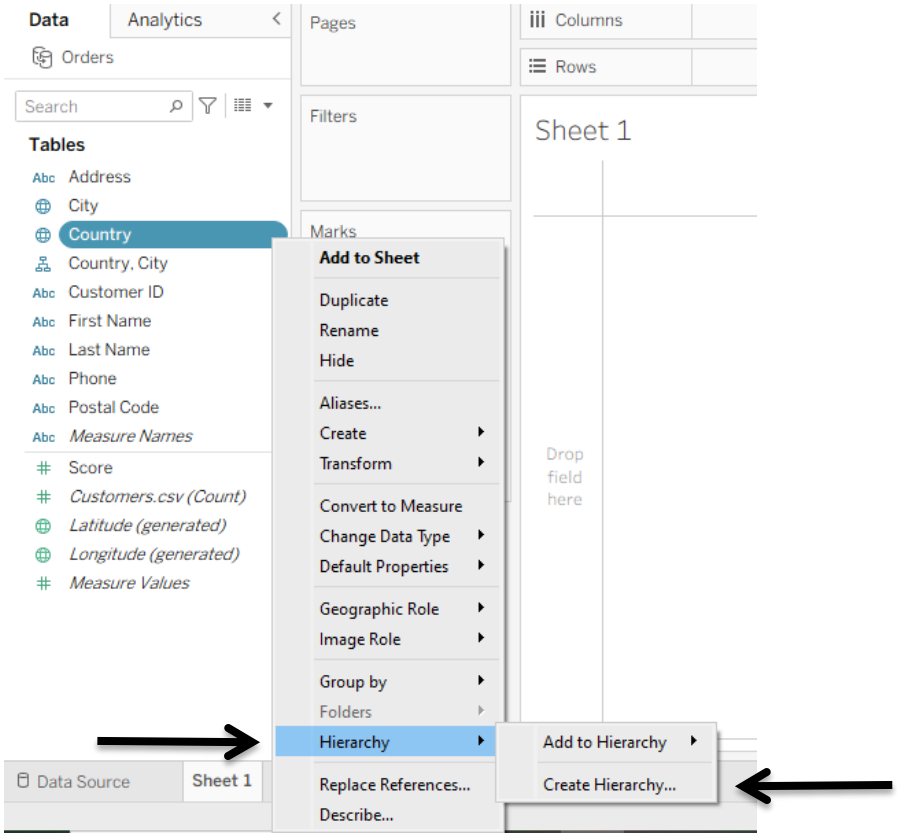
1. Drop Down Menu
2. Drag and Drop

1. Drop Down Menu

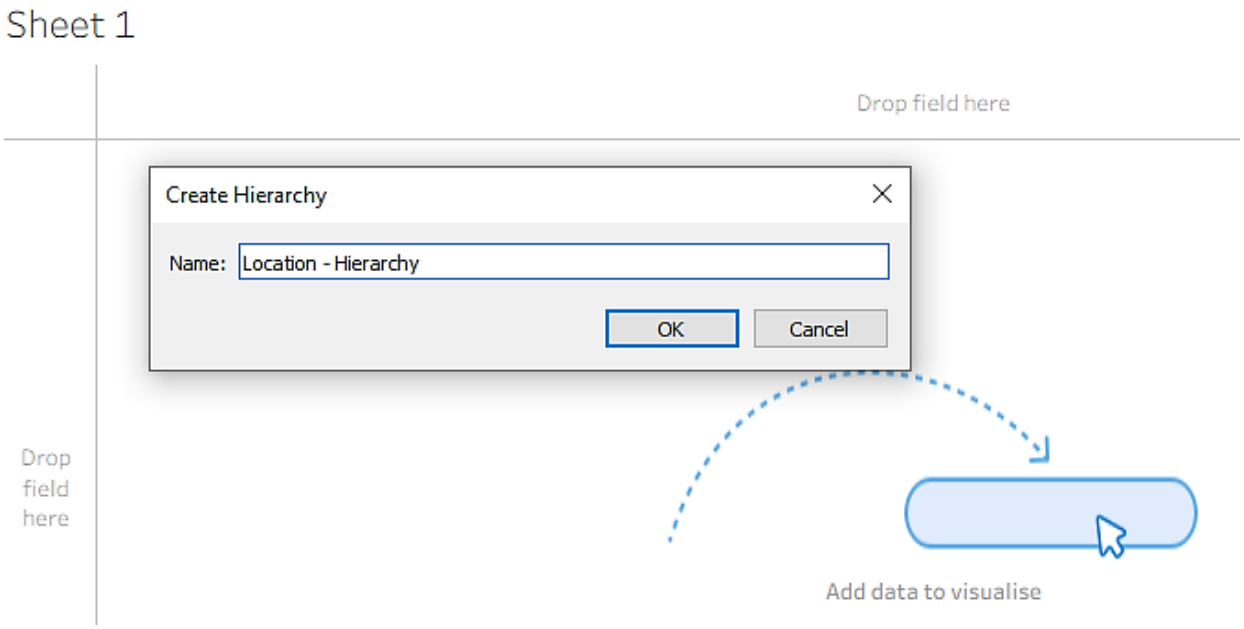
Step 1: First drag and drop Customers.csv in the Data source page.



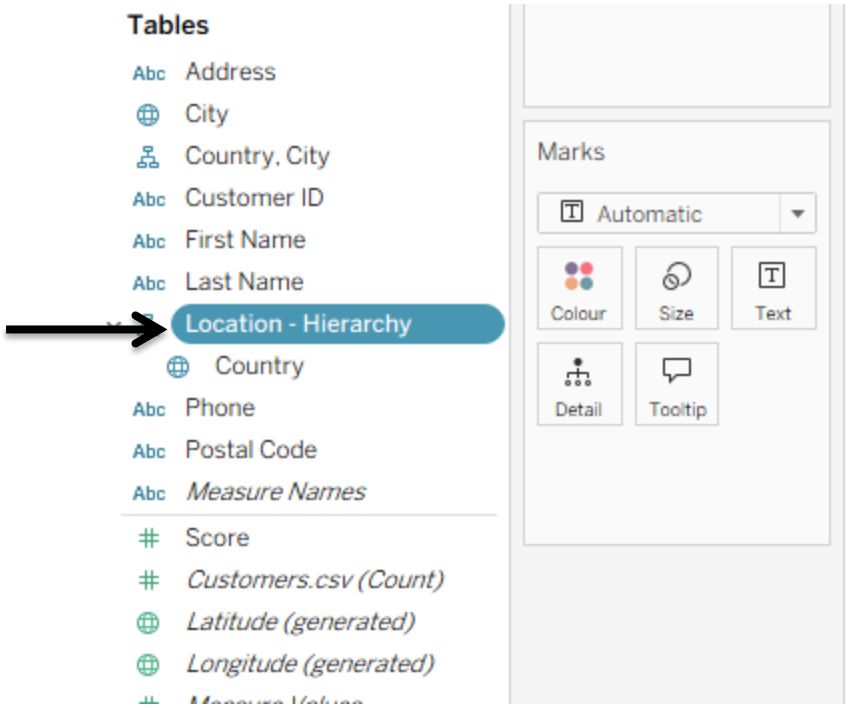
Step 2: Right click on Country, You will have Hierarchy field in the drop down menu, afterwards you will get Create Hierarchy field.



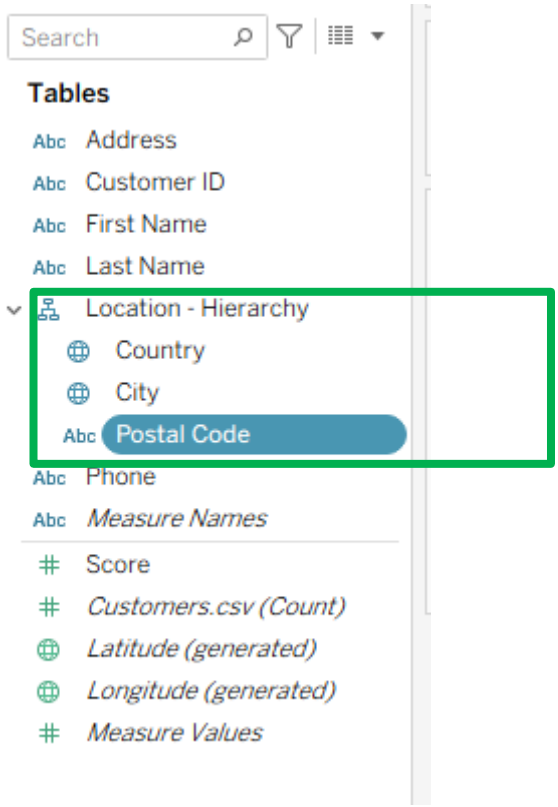
Step 3: A dialog box will open, enter the name of the hierarchy, for example “Location Hierarchy” and then click ok.



Step 4: Location Hierarchy is created with only one field called "Country".

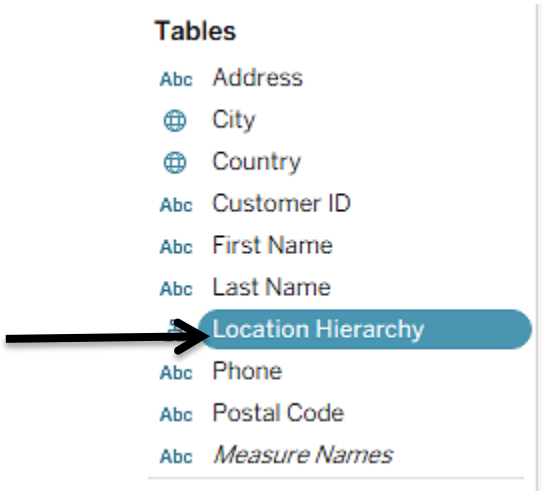


Step 5: To add city and postal code to the location hierarchy, just drag and drop both the fields.

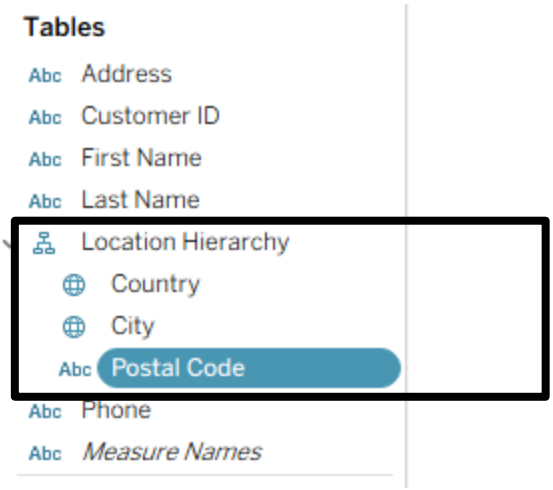


2. Drag & Drop Method

Step 1: By default, already Country, City Hierarchy is created. So change the name of the hierarchy to "Location Hierarchy".



Step 2: Just Drag and Drop, Country, City and Postal code in the Location Hierarchy.

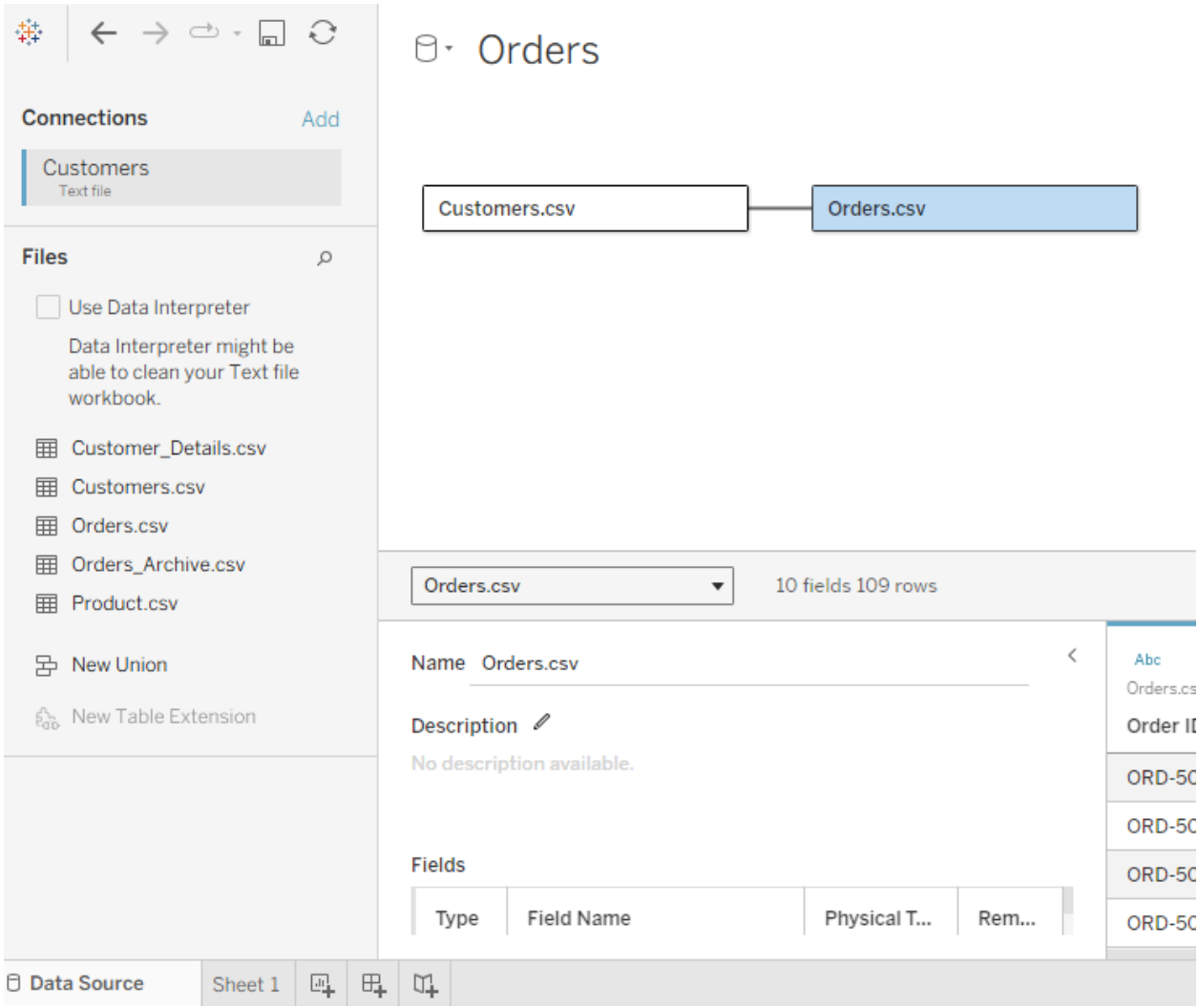


Two Operations

1. **Drill-Down**
2. **Drill-up**

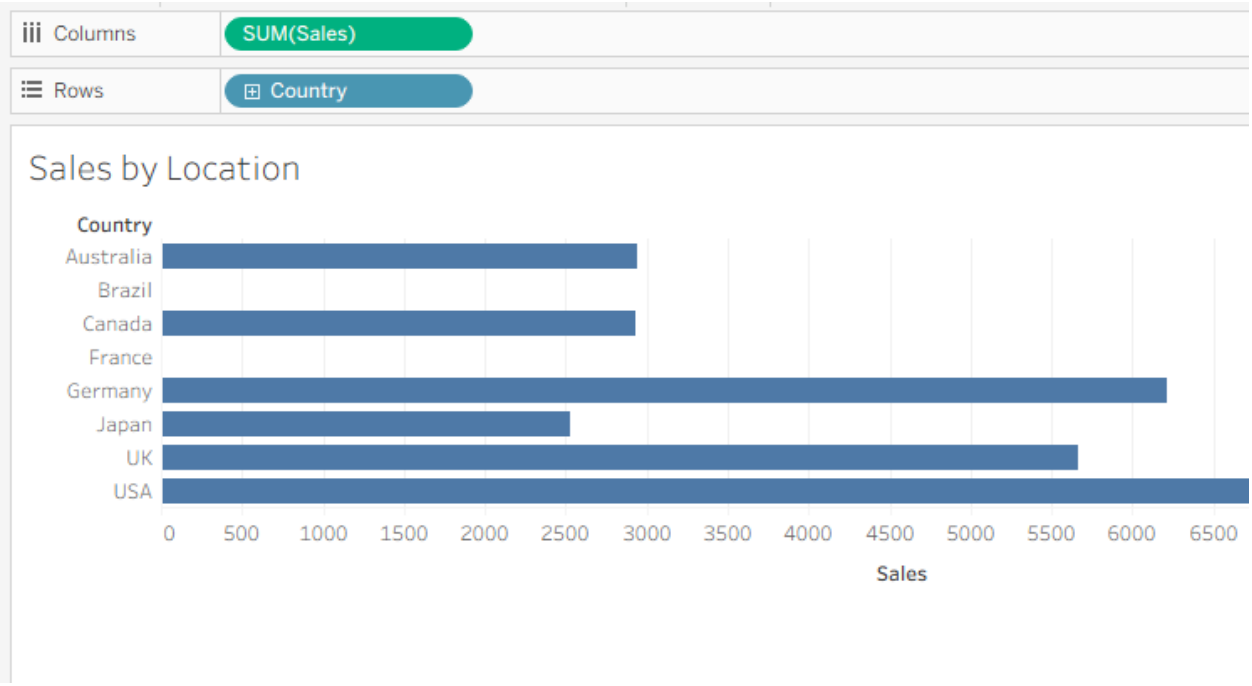
Drill-Down

Step 1: Initially, Drag and Drop Customers.csv with Orders.csv in order to perform hierarchies.

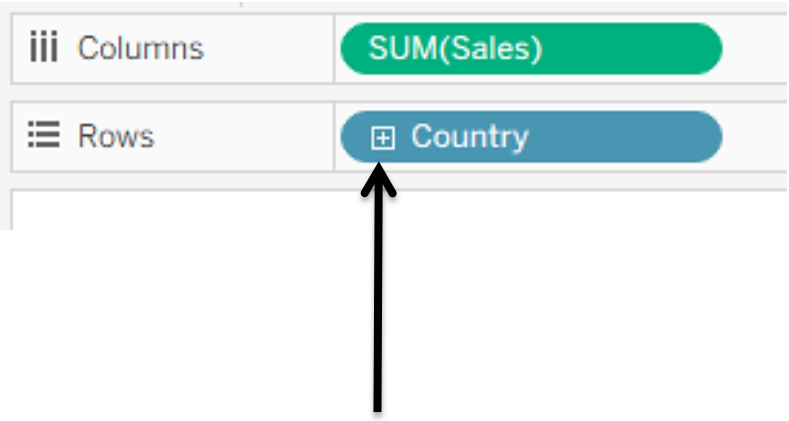


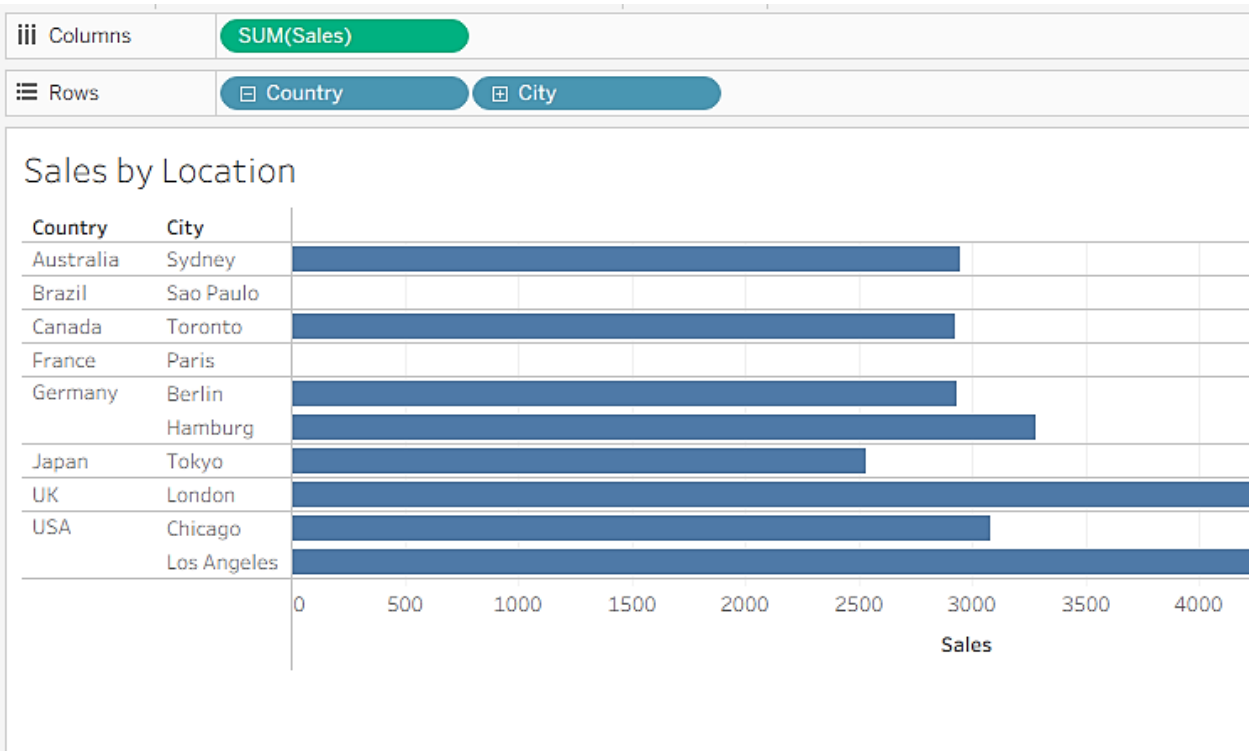
Step 2: Move to Sheet 1, drag Country and drop in rows, drag sales from order and drop in columns.

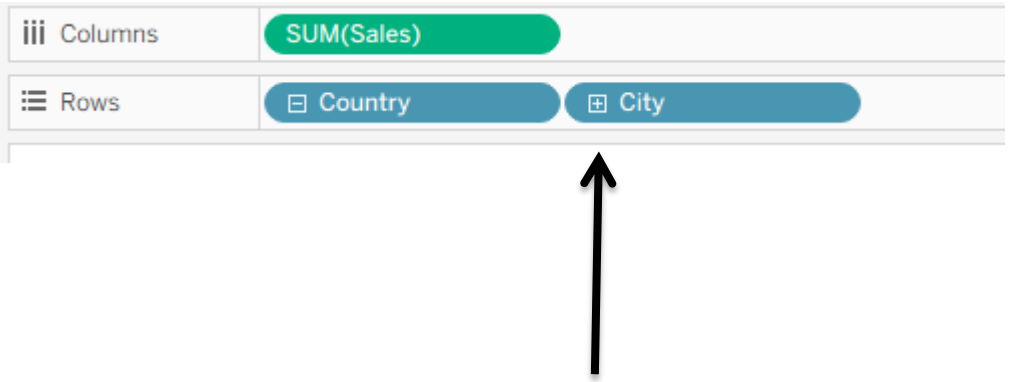
Change the Sheet 1 name to Sales by Location.

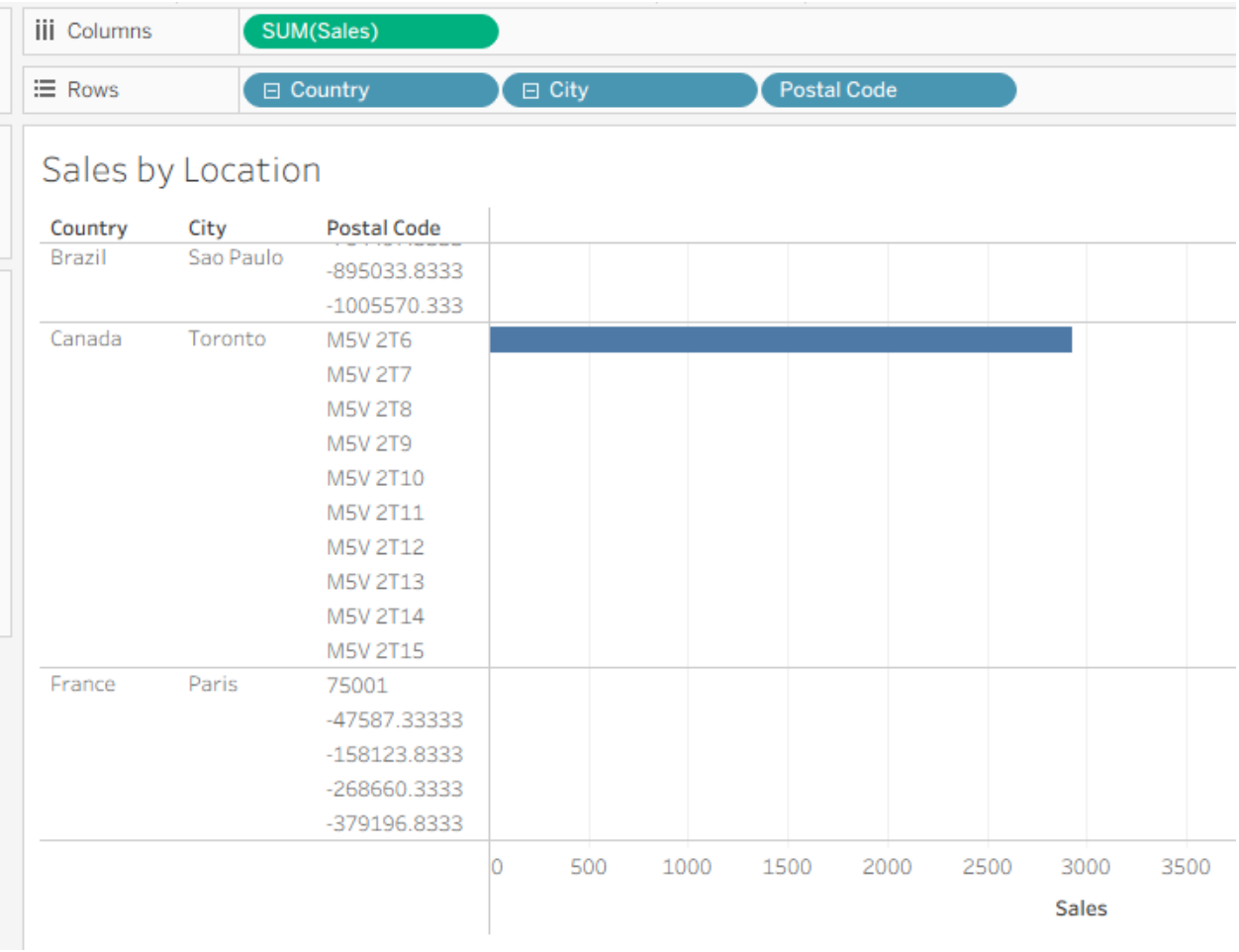


Step 3: Click the + sign in country, in order to view city, and + sign in city to view postal code which is the drill down operation in step by step.



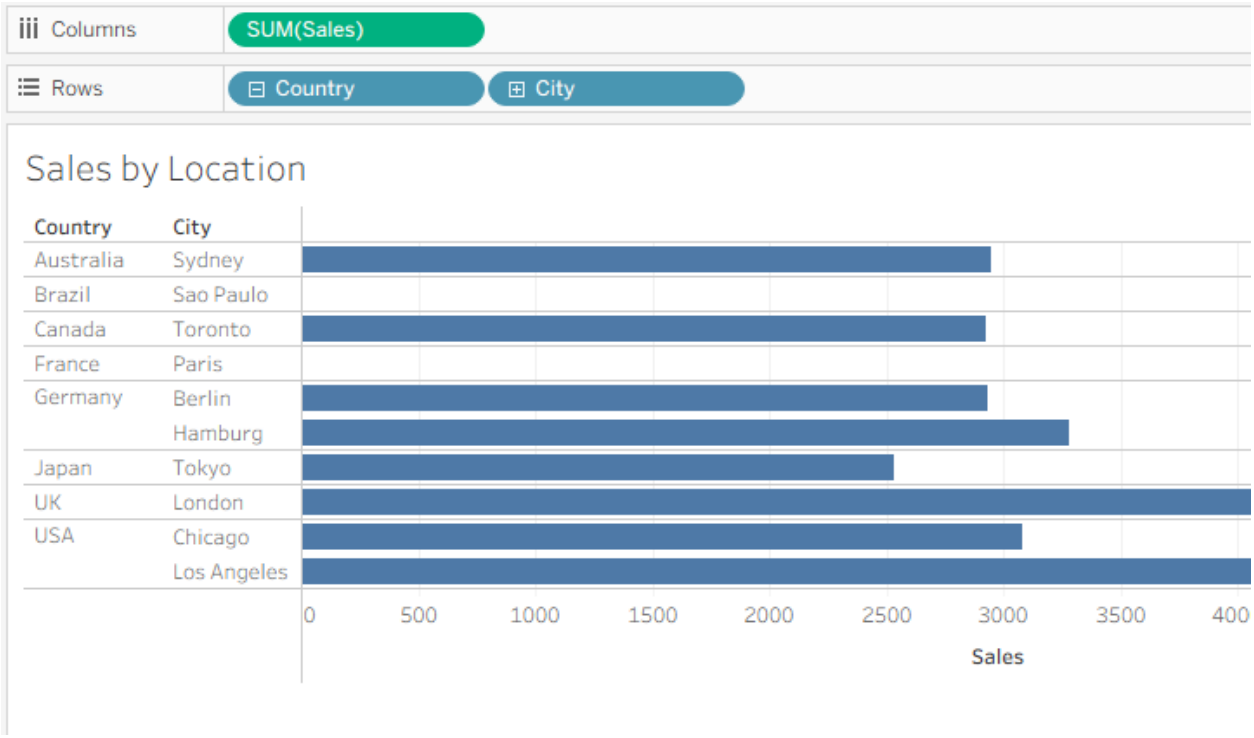

Columns
SUM(Sales)
Rows
Country
City
Sales by Location
Country
City
Australia
Sydney
Brazil
Sao Paulo
Canada
Toronto
France
Paris
Germany
Berlin
Hamburg
Japan
Tokyo
UK
London
USA
Chicago
Los Angeles
0
500
1000
1500
2000
2500
3000
3500
4000
Sales

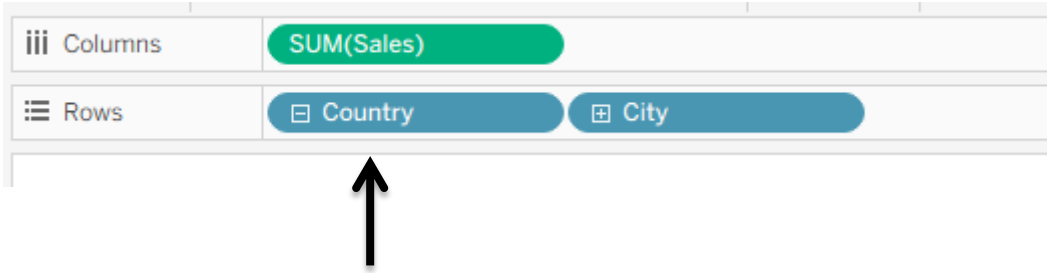

Columns
SUM(Sales)
Rows
Country
City

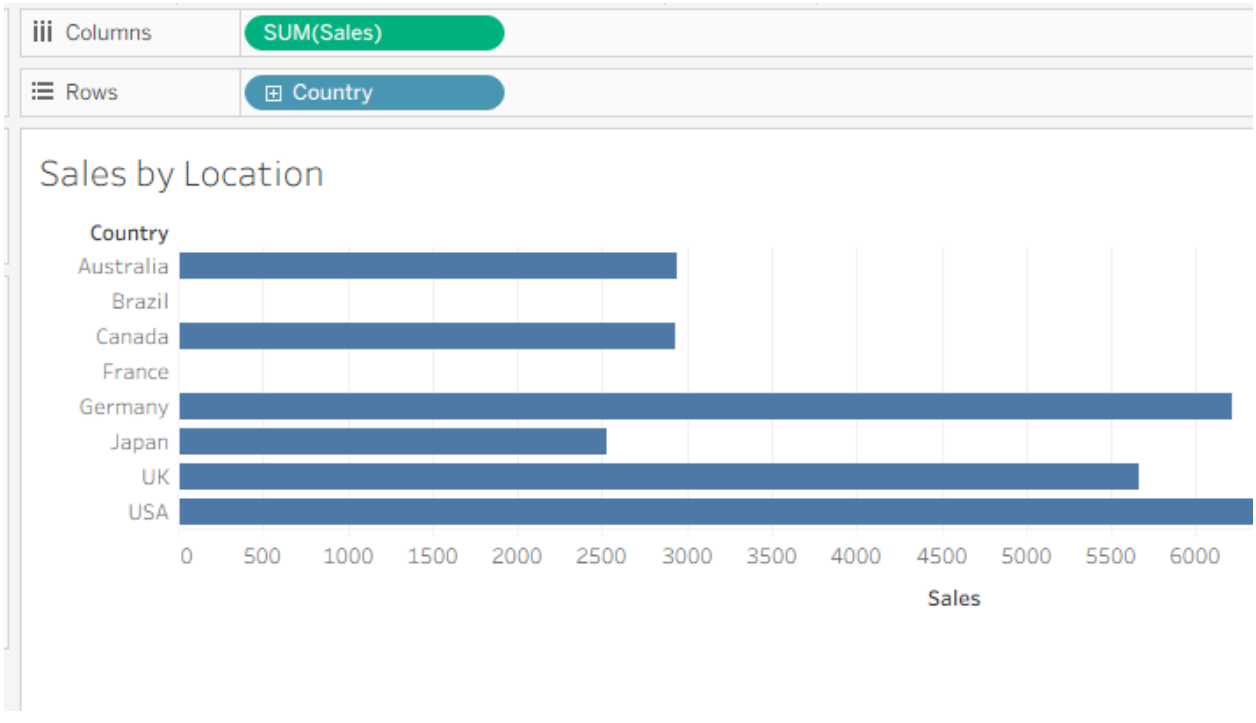


Drill Up Operation

Step 1: Click – sign in City, to remove postal code, and click – sign in Country to remove or drill up.


Columns
SUM(Sales)
Rows
Country
City
Sales by Location
Country
City
Australia
Sydney
Brazil
Sao Paulo
Canada
Toronto
France
Paris
Germany
Berlin
Hamburg
Japan
Tokyo
UK
London
USA
Chicago
Los Angeles
0
500
1000
1500
2000
2500
3000
3500
400
Sales


Columns
SUM(Sales)
Rows
Country
City


Columns
SUM(Sales)
Rows
Country
Sales by Location
Country
Australia
Brazil
Canada
France
Germany
Japan
UK
USA
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
6000
Sales

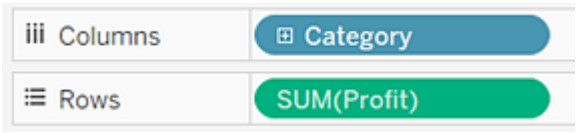
Practice 4: Graphical Tools for data elaboration – 1

1. Bar chart

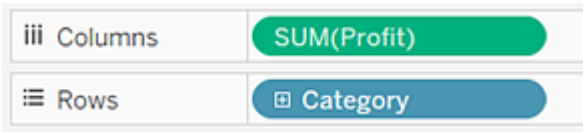
Bar charts are used to compare data across categories. You create a bar chart by placing a dimension on the **Rows** shelf and a measure on the **Columns** shelf, or vice versa.

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.

Creates Vertical Bars

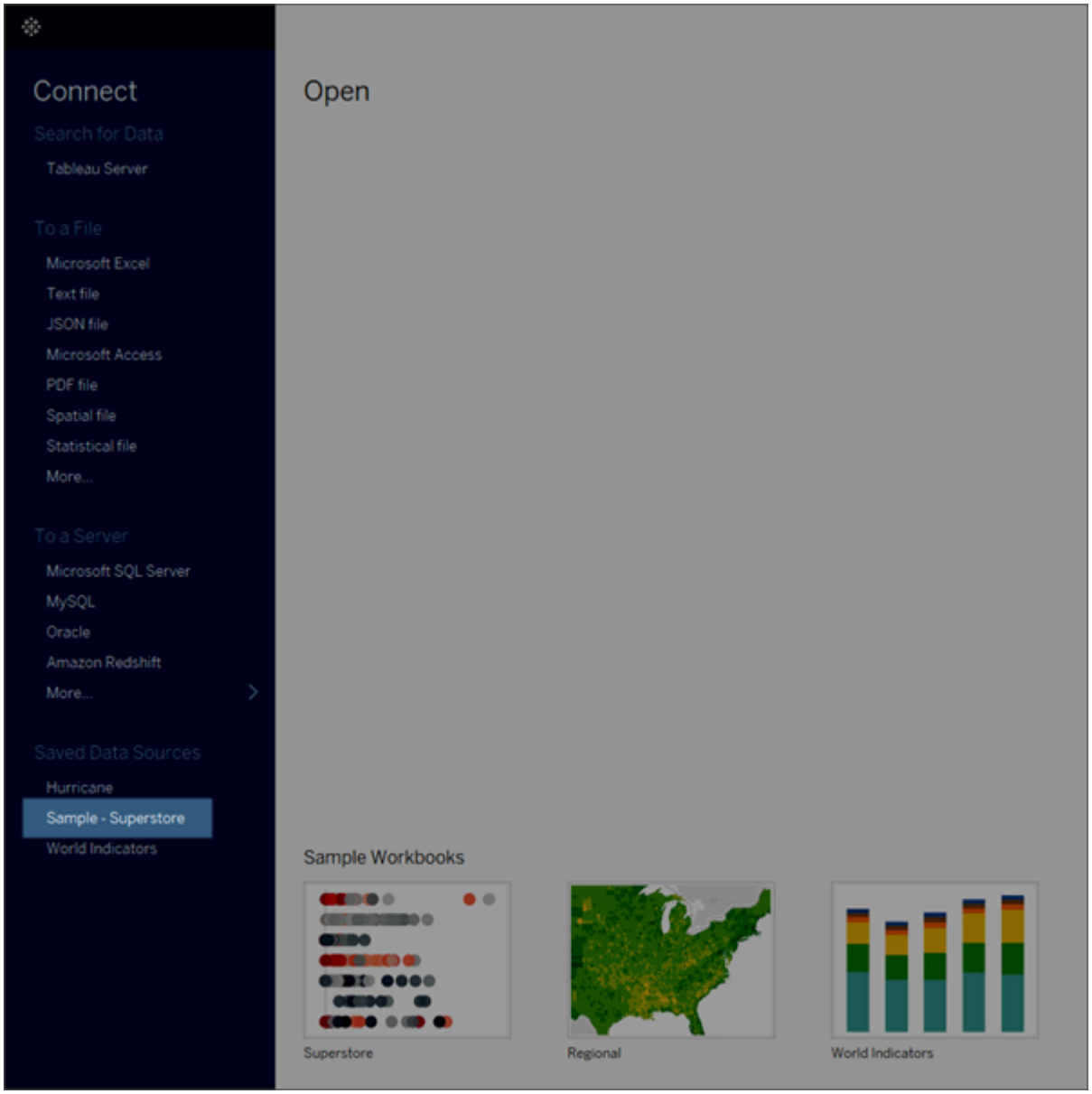


Creates Horizontal Bars



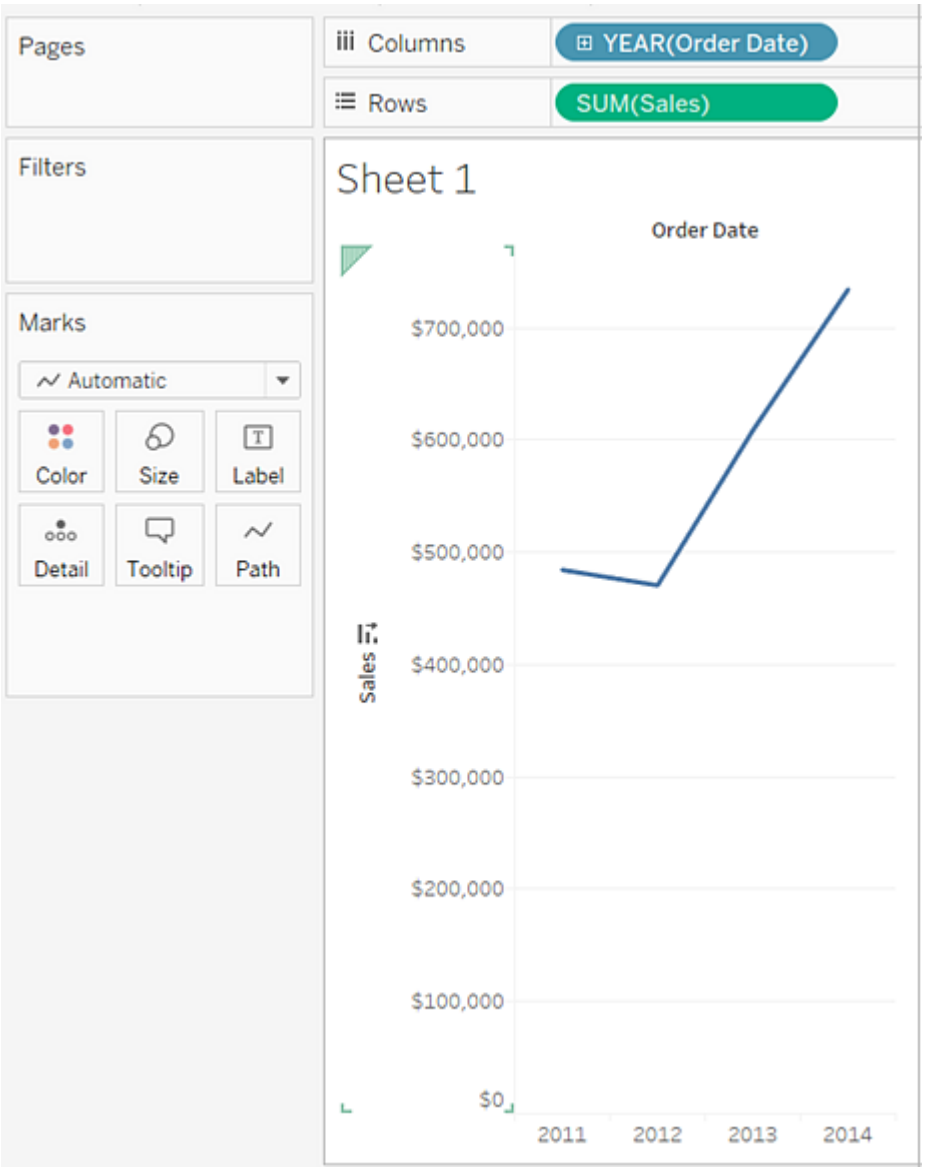
To create a bar chart that displays total sales over a four-year period, follow these steps:

1. Connect to the **Sample - Superstore** data source.

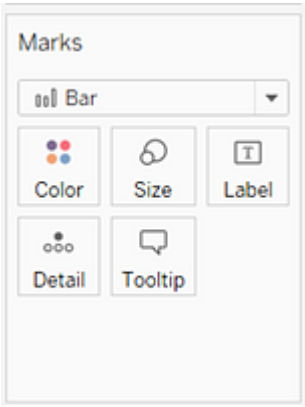


2. Drag the **Order Date** dimension to **Columns** and drag the **Sales** measure to **Rows**.

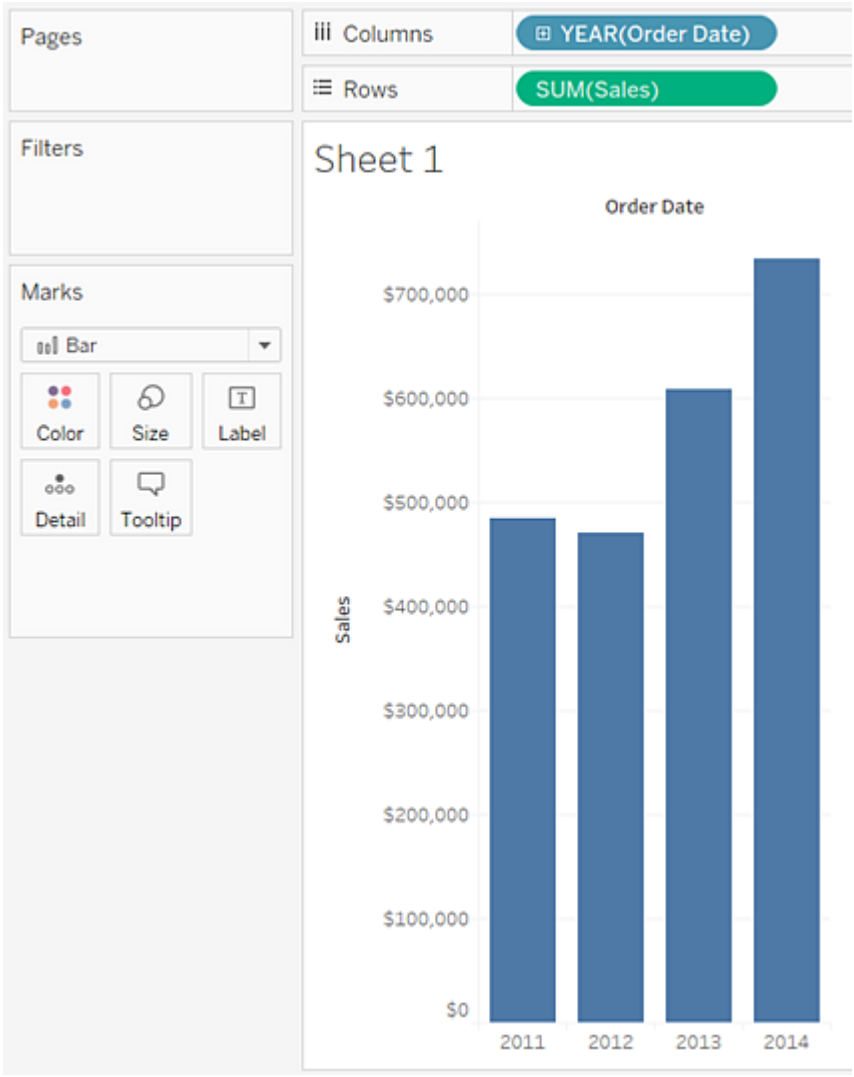
   Notice that the data is aggregated by year and column headers appear. The Sales measure is aggregated as a sum and an axis is created, while the column headers move to the bottom of the view. Tableau uses **Line** as the mark type because you added the date dimension.



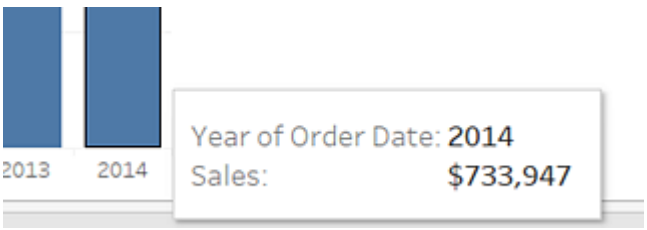
3. On the **Marks** card, select **Bar** from the drop-down list.



The view changes to a bar chart.

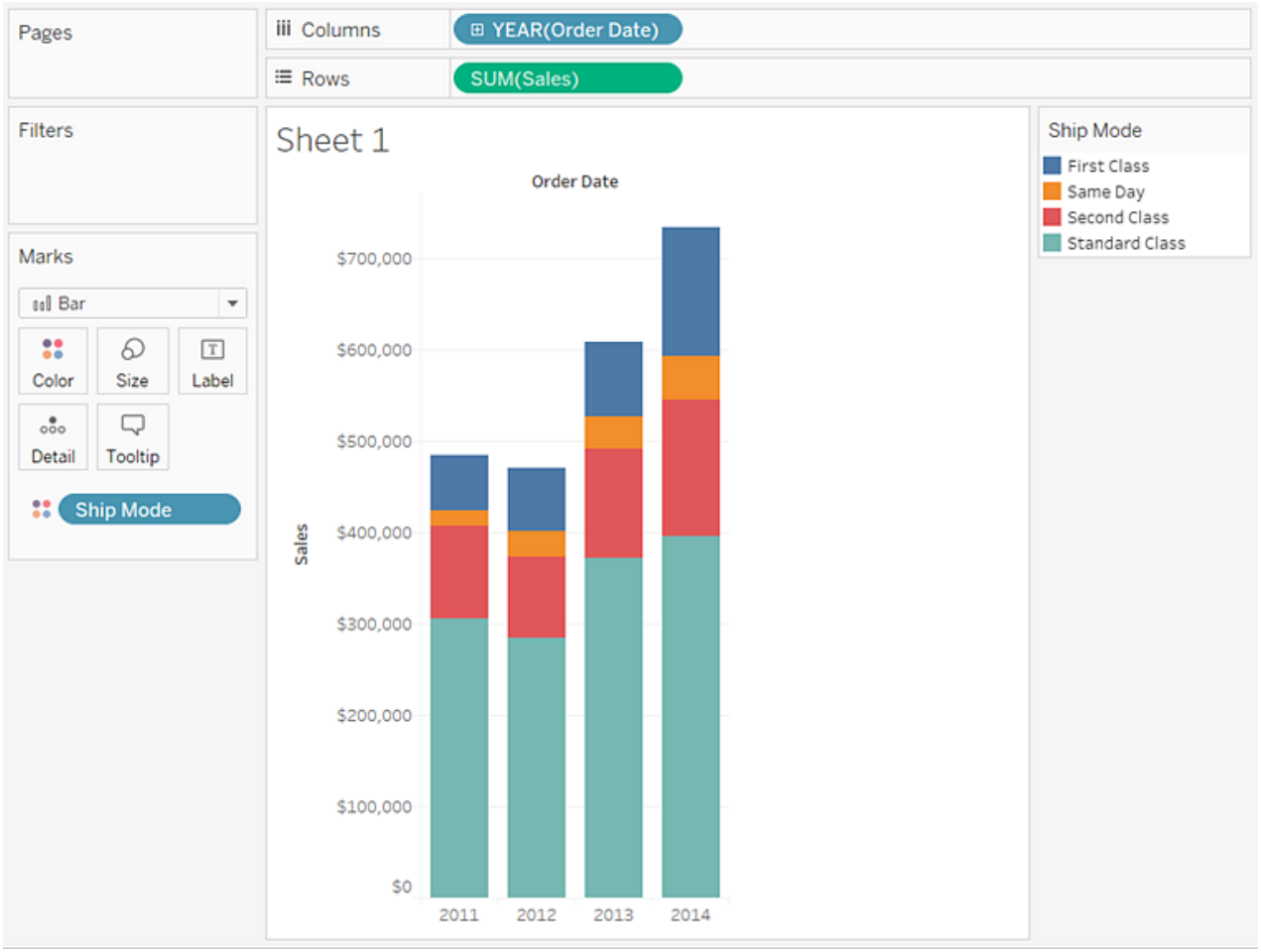


The marks (which are bars in this case) are vertical because the axis is vertical. The length of each mark represents the sum of the sales for that year. The actual numbers you see here might not match the numbers you see—the sample data changes from time to time.

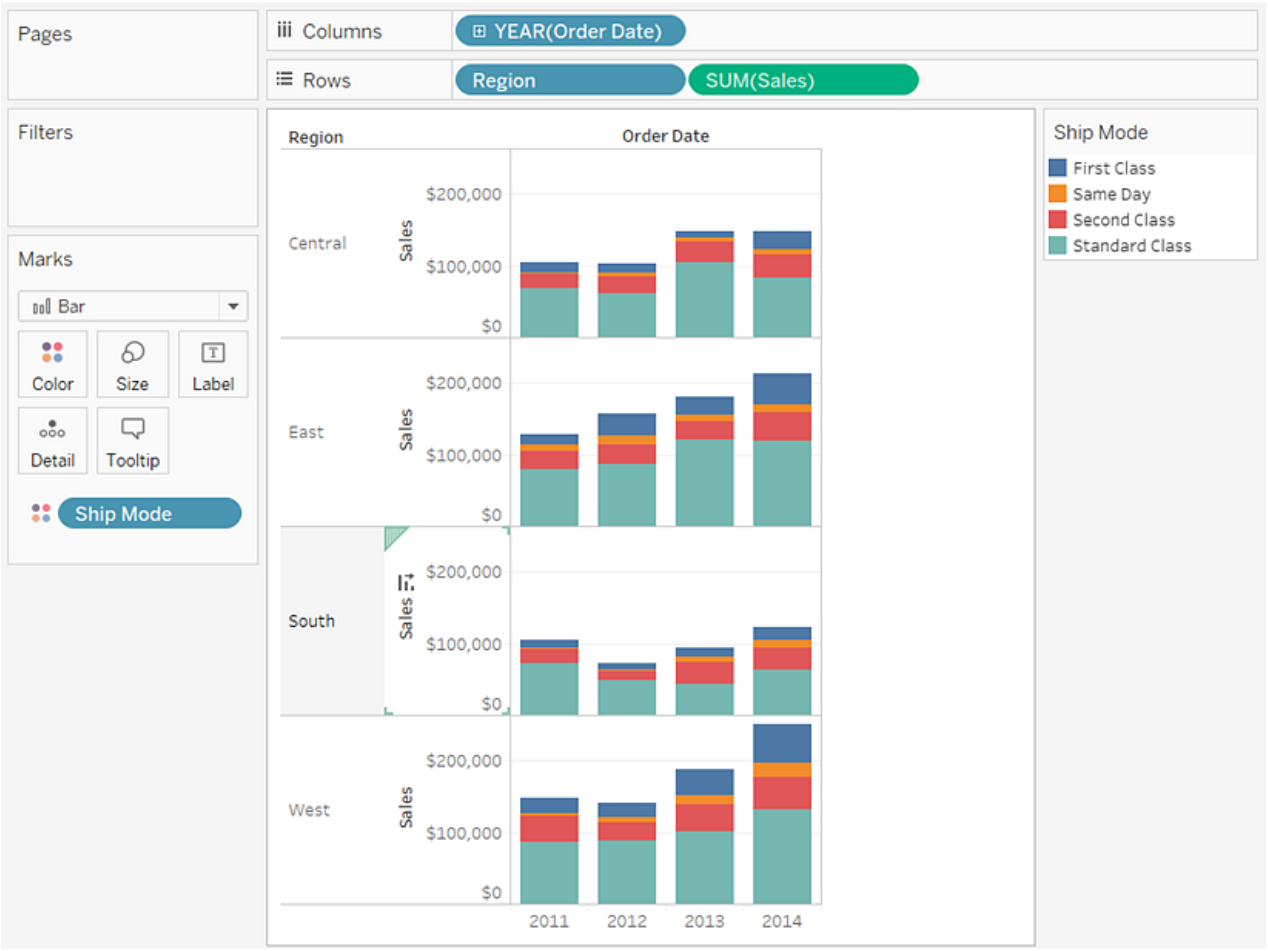


4. Drag the **Ship Mode** dimension to **Color** on the **Marks** card.

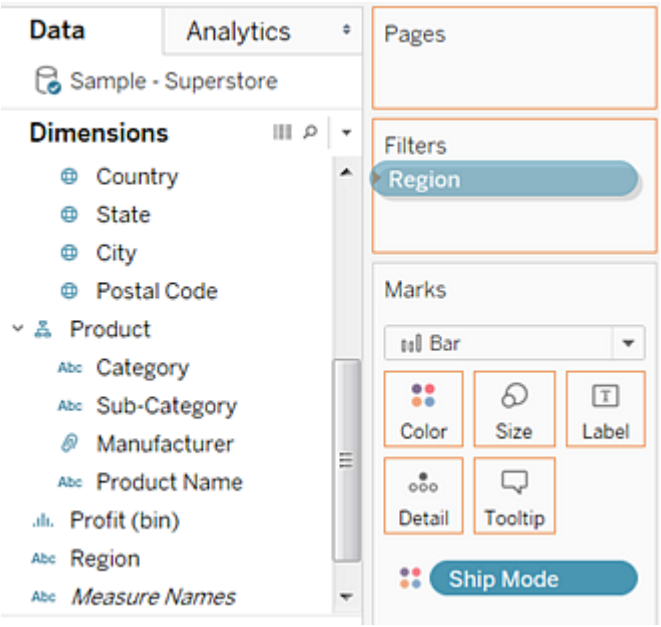
   The view shows how different shipping modes have contributed to total sales over time. The ratios look consistent from year to year.



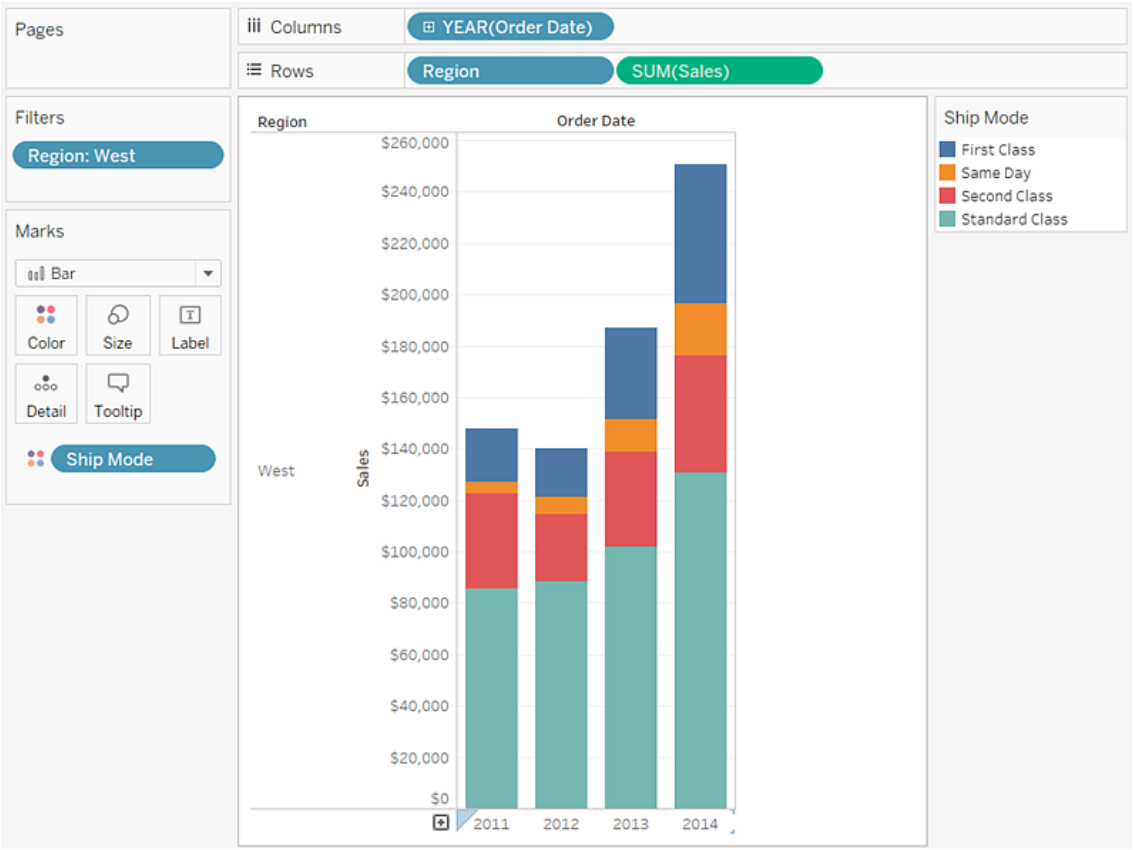
5. Drag the **Region** dimension to **Rows**, and drop it to the left of **Sales** to produce multiple axes for sales by region.



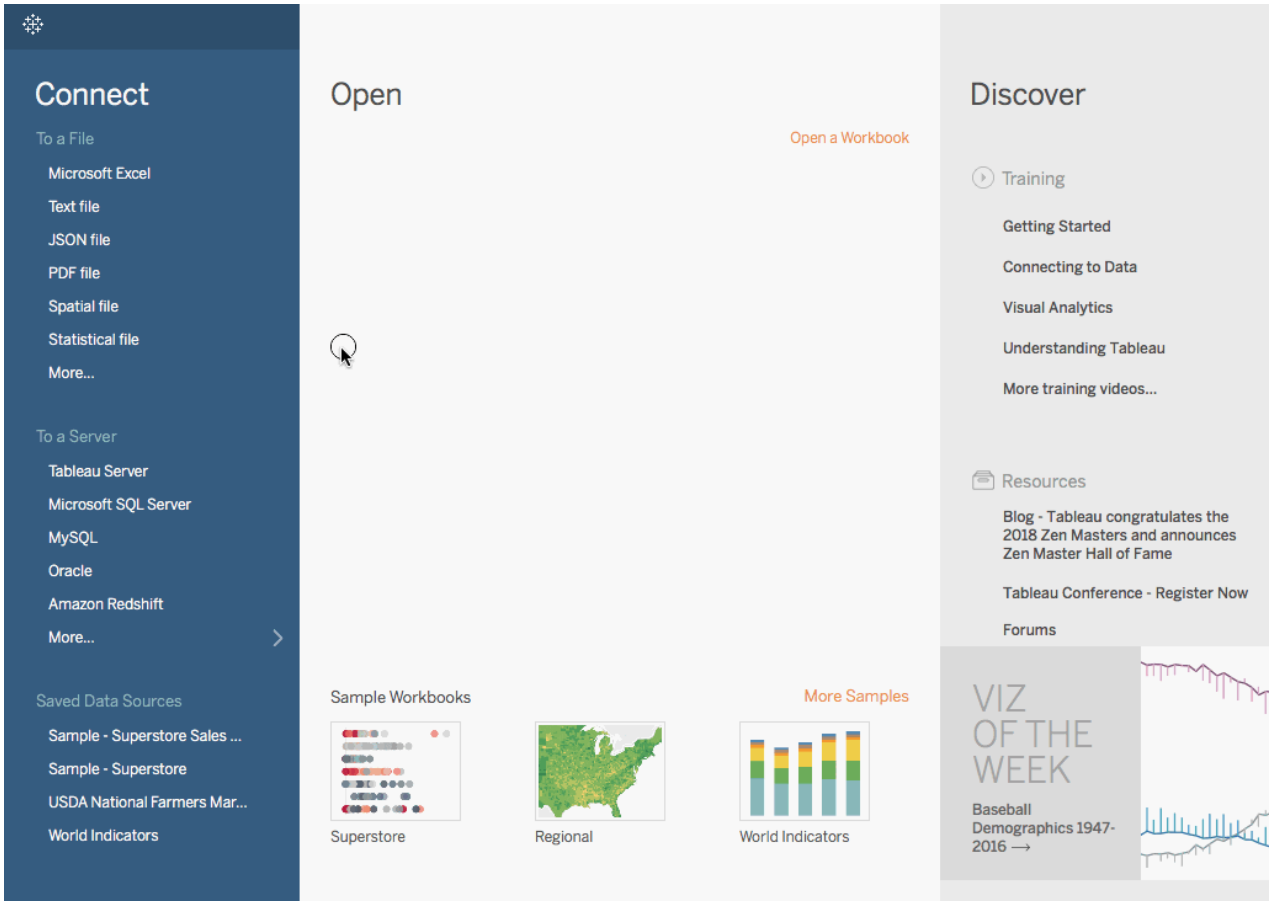
6. To view data in the West region only, you can filter out the other regions. To do this, drag the **Region** dimension again, this time from the **Data** pane to the **Filters** shelf.



7. In the Filter [Region] dialog box, clear the **Central**, **East**, and **South** check boxes, and then click **OK**.



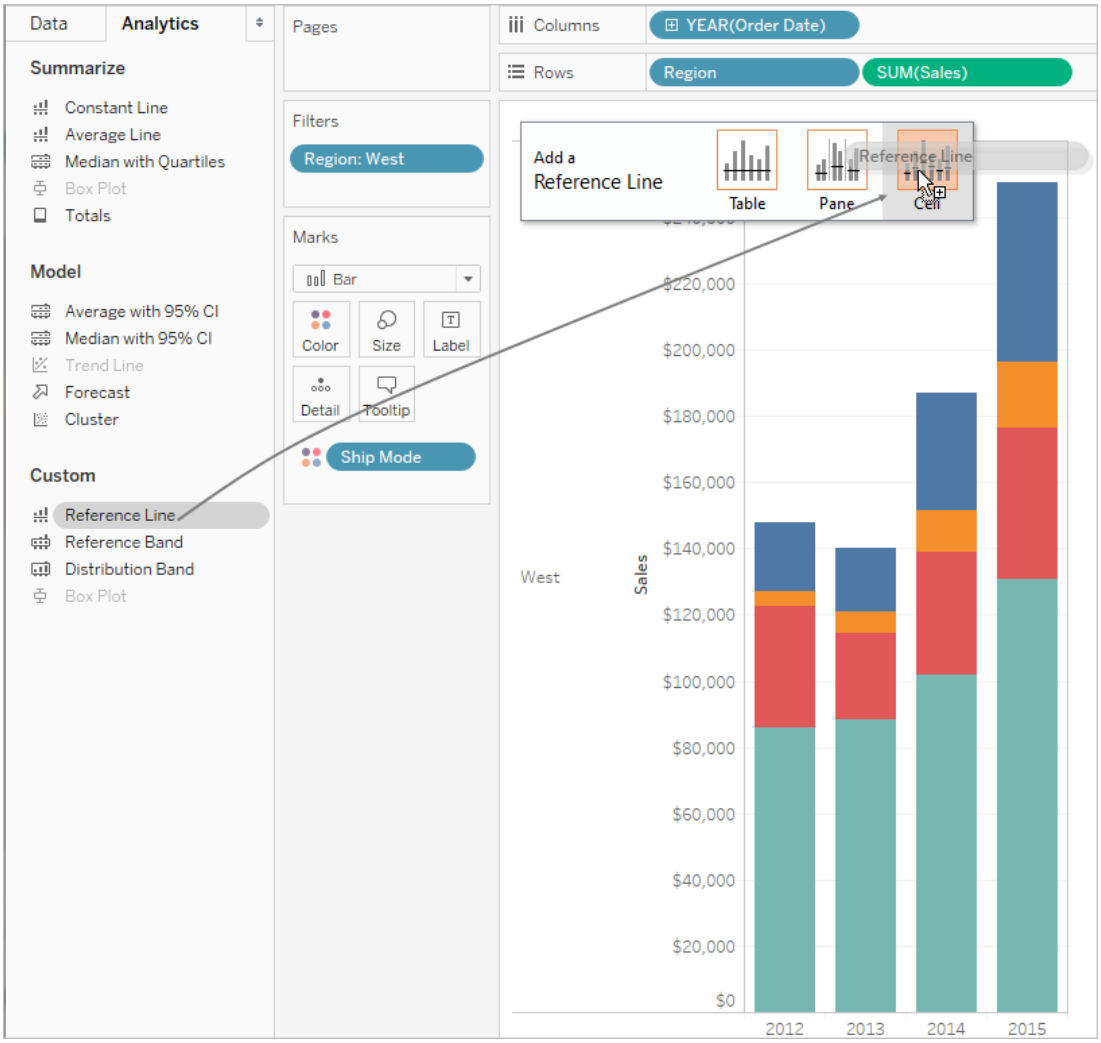
This view gives you insight into your data—for example, how the ship mode changed in the West over the four-year period.



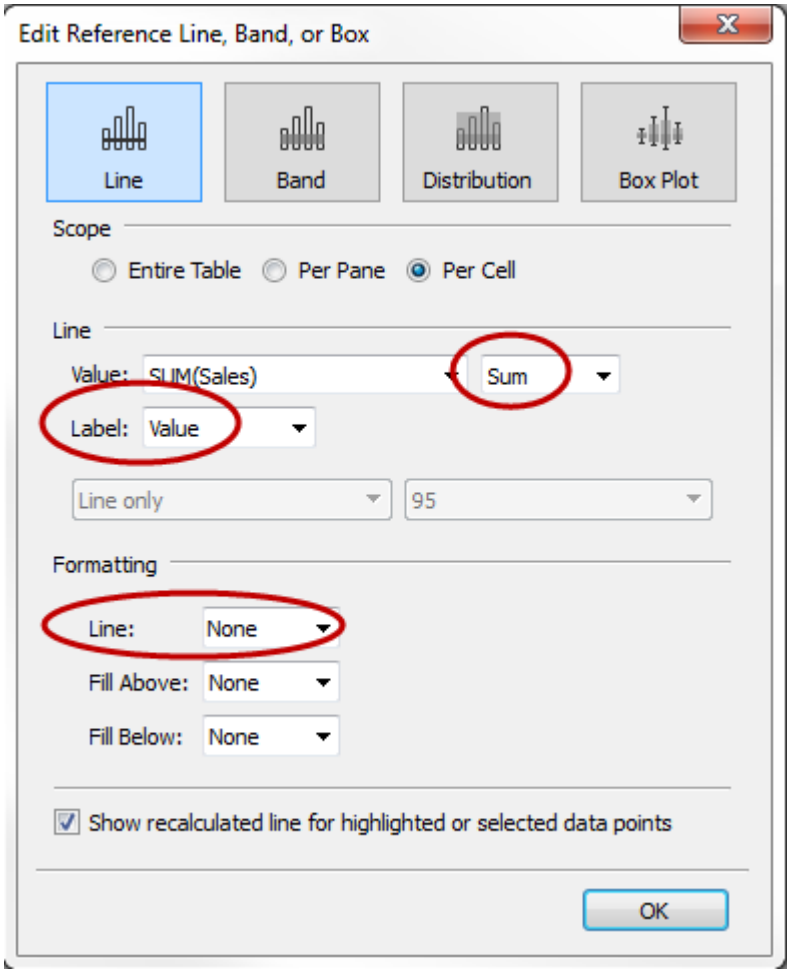
One Step Further: Add Totals To Stacked Bars

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**.

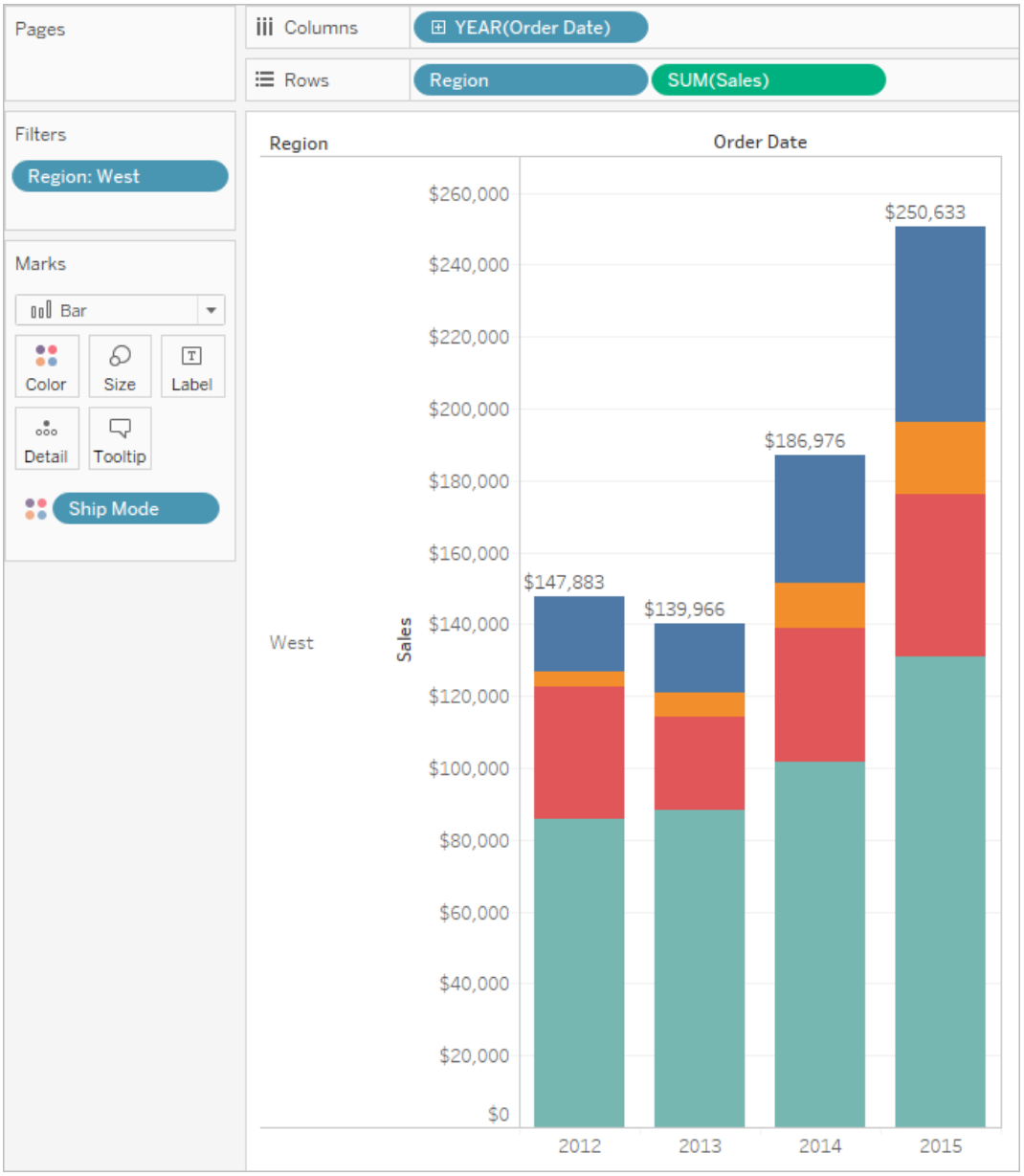


2. In the Edit Line, Band, or Box dialog box, set the aggregation for **SUM(Sales)** to **Sum**, set **Label** to **Value**, and set **Line** under Formatting to **None**:



Then click **OK** to close the Edit Reference Line, Band, or Box dialog box.

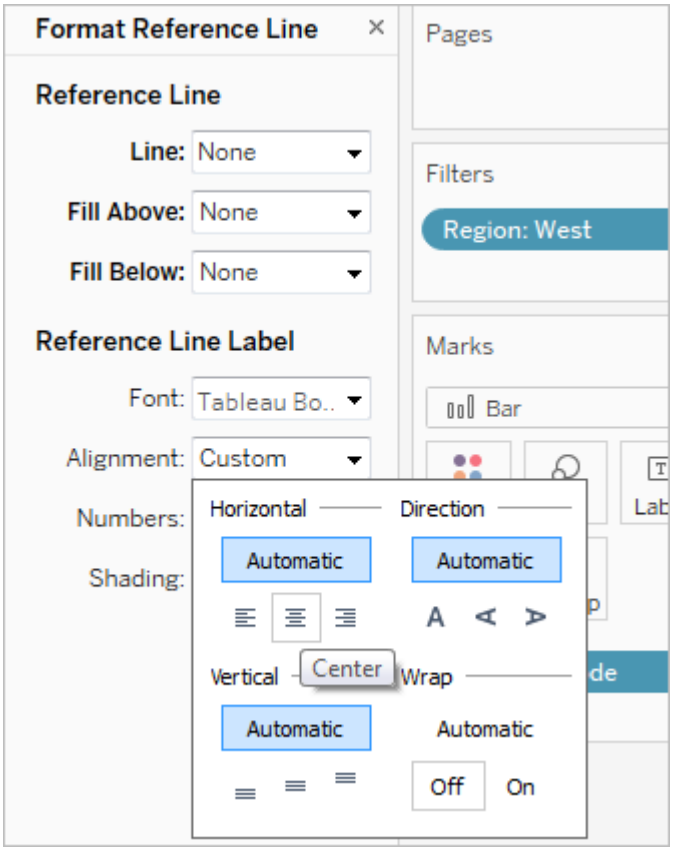
Your view now has currency totals at the top of each bar:



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:

3. Right-click any of the totals on the bar chart and select **Format**.

4. In the Format window, in the **Reference Line Label** area, open the **Alignment** control and select the Center option for Horizontal alignment:



2. Pie chart

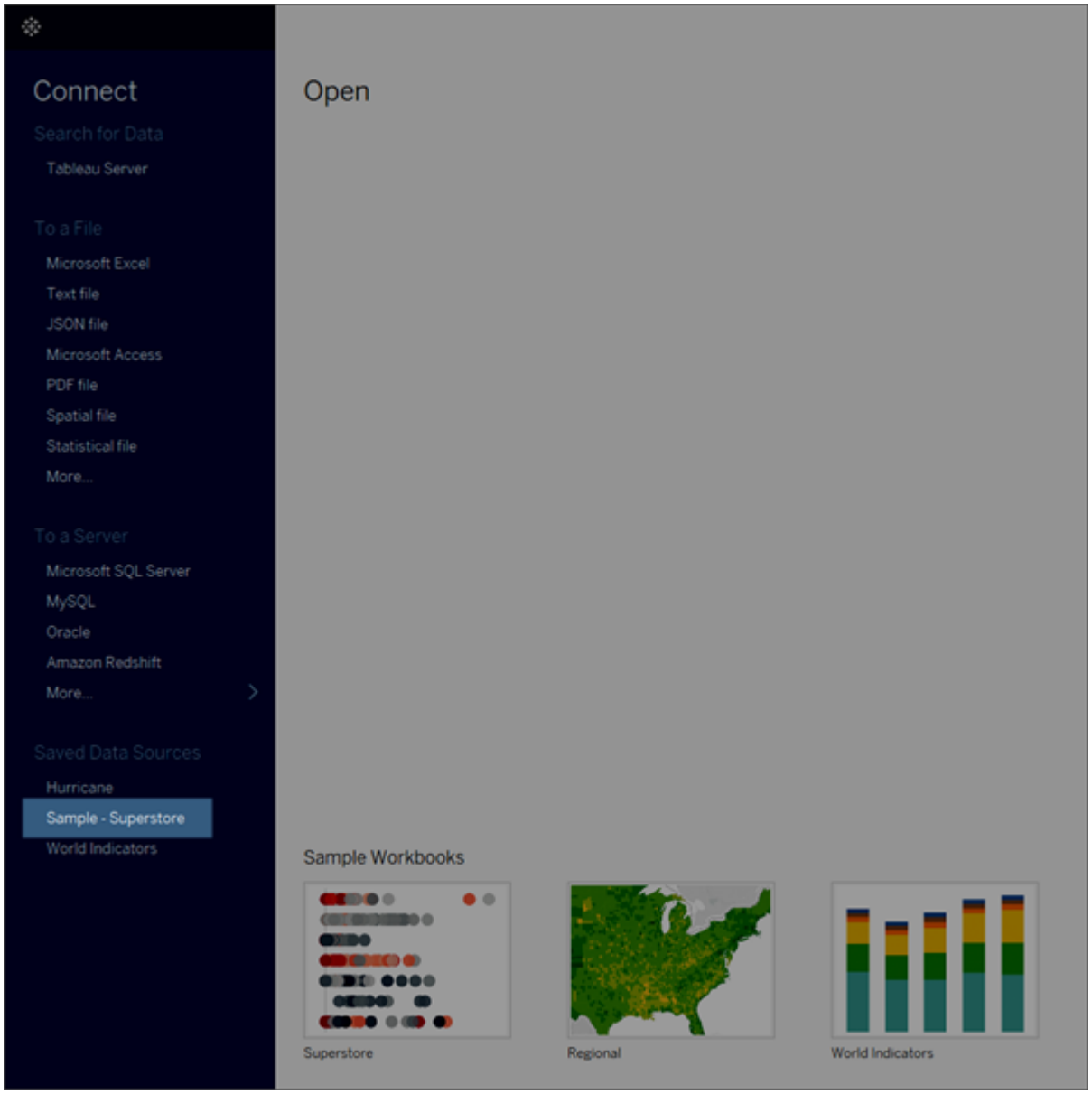
Pie Charts are used to show proportions of a whole.

The basic building blocks for a pie chart are as follows:

Mark type:	Pie
Color:	Dimension
Angle:	Measure

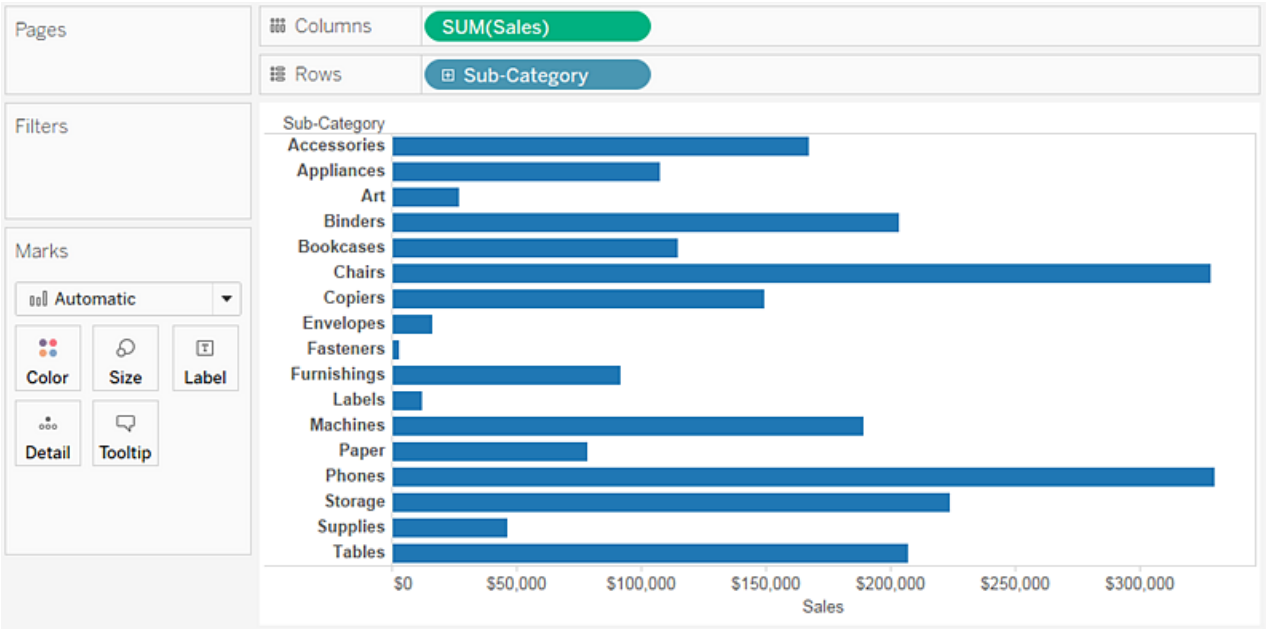
To create a pie chart view that shows how different product categories contribute to total sales, follow these steps:

1. Connect to the **Sample - Superstore** data source.



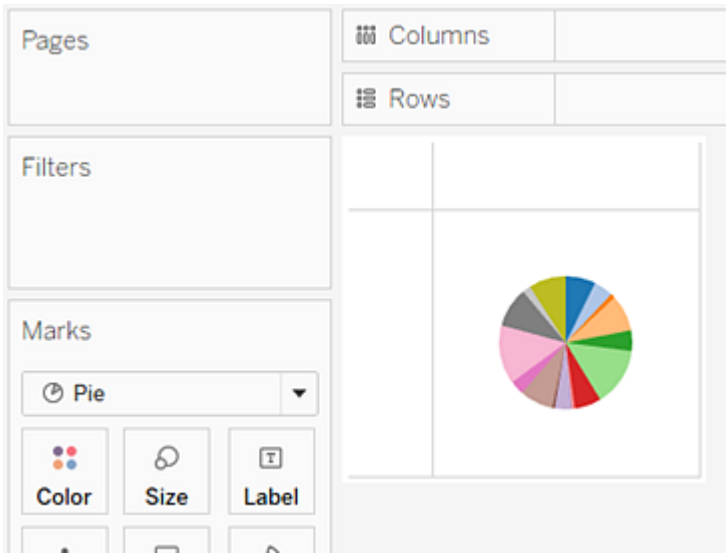
2. Drag the **Sales** measure to **Columns** and drag the **Sub-Category** dimension to **Rows**.

   Tableau aggregates the **Sales** measure as a sum. By default, Tableau displays a bar chart.

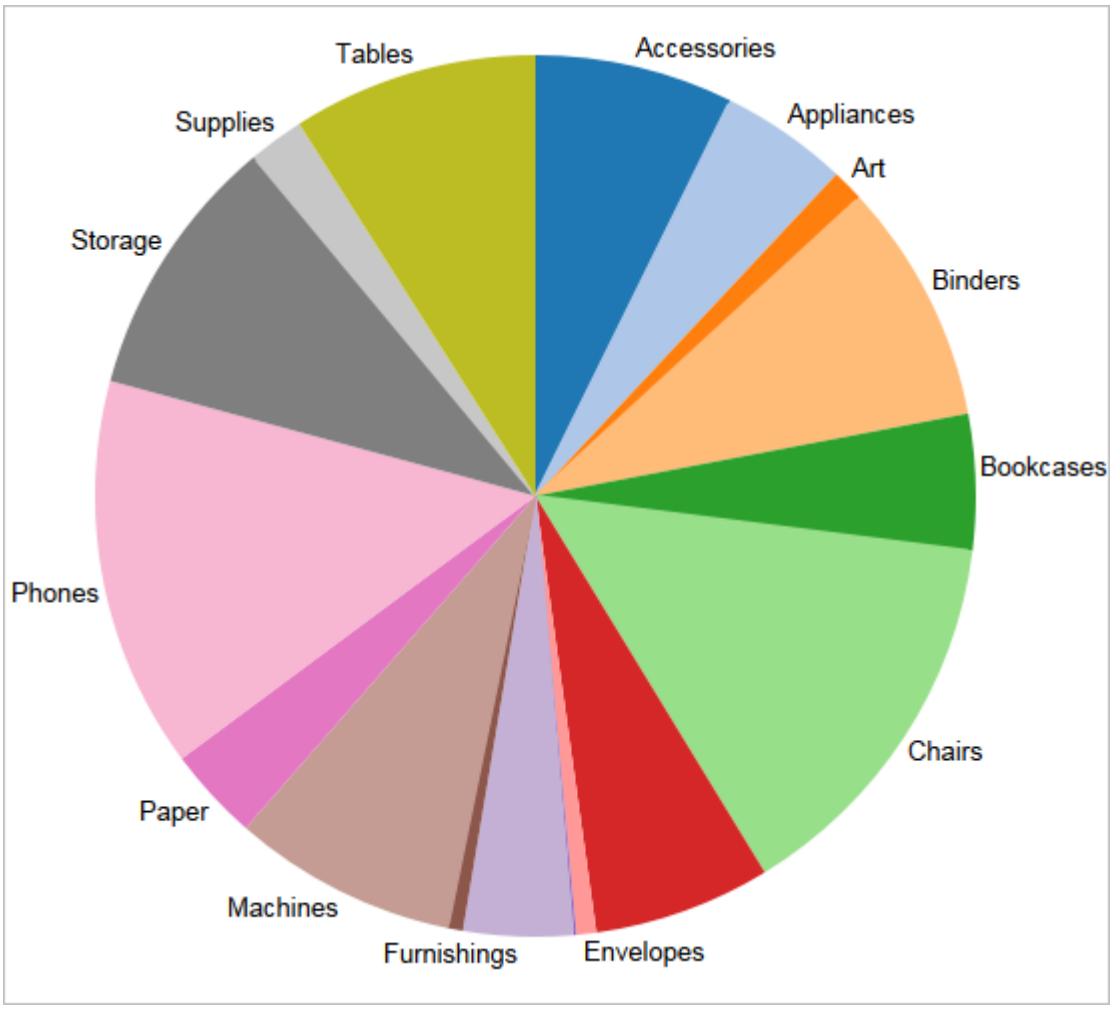


3. Click **Show Me** on the toolbar, then select the pie chart type. Pie charts require at least one or more dimensions and one or two measures. Aggregate fields, such as Profit Ratio, don't contribute to those requirements.

The result is a rather small pie. To make the chart bigger, hold down Ctrl + Shift (hold down ñ + z on a Mac) and press B several times.



4. Add labels by dragging the **Sub-Category** dimension from the **Data** pane to **Label** on the **Marks** card.



3. Line chart

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.

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**.

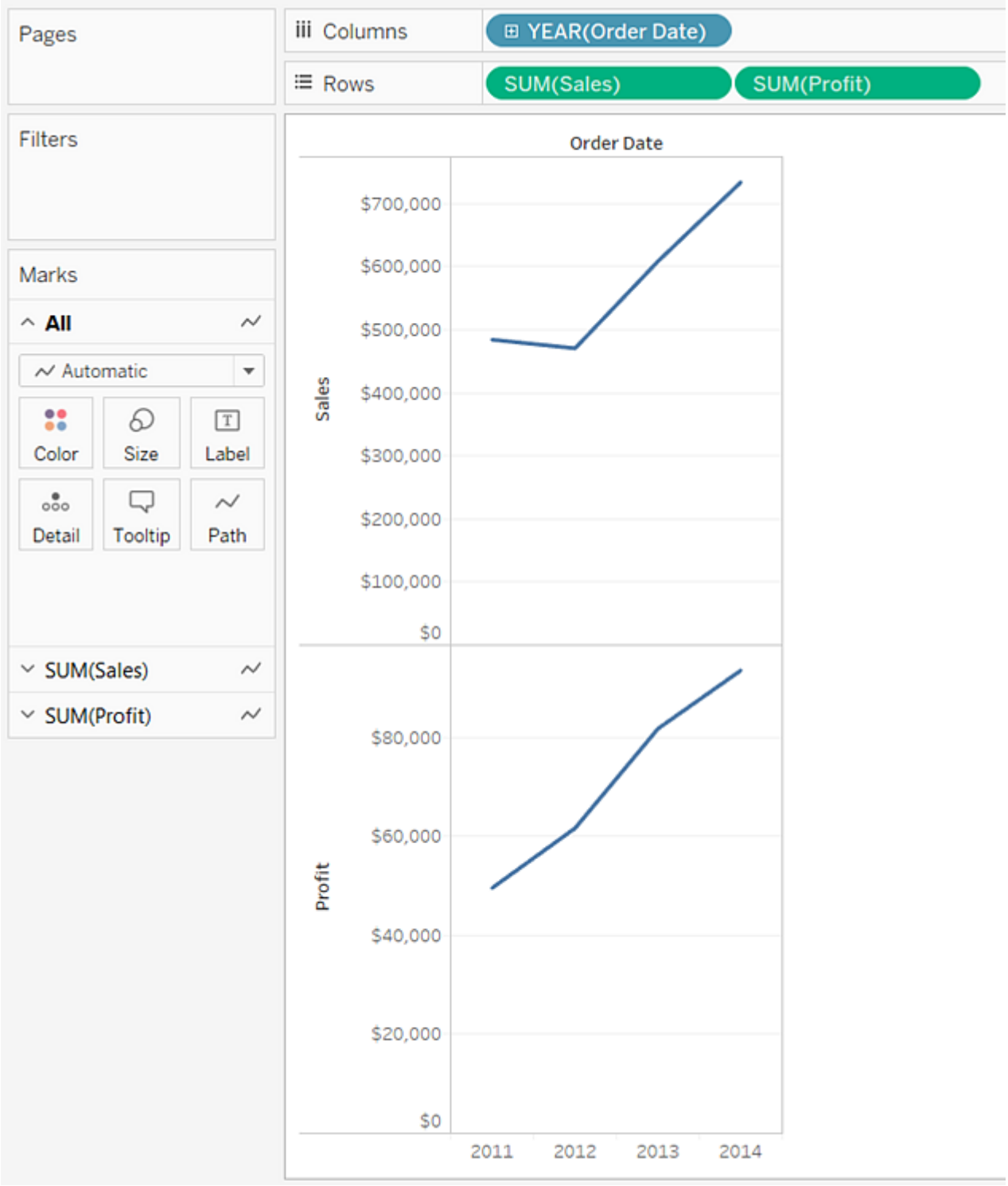
   Tableau aggregates the date by year, and creates column headers.

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.

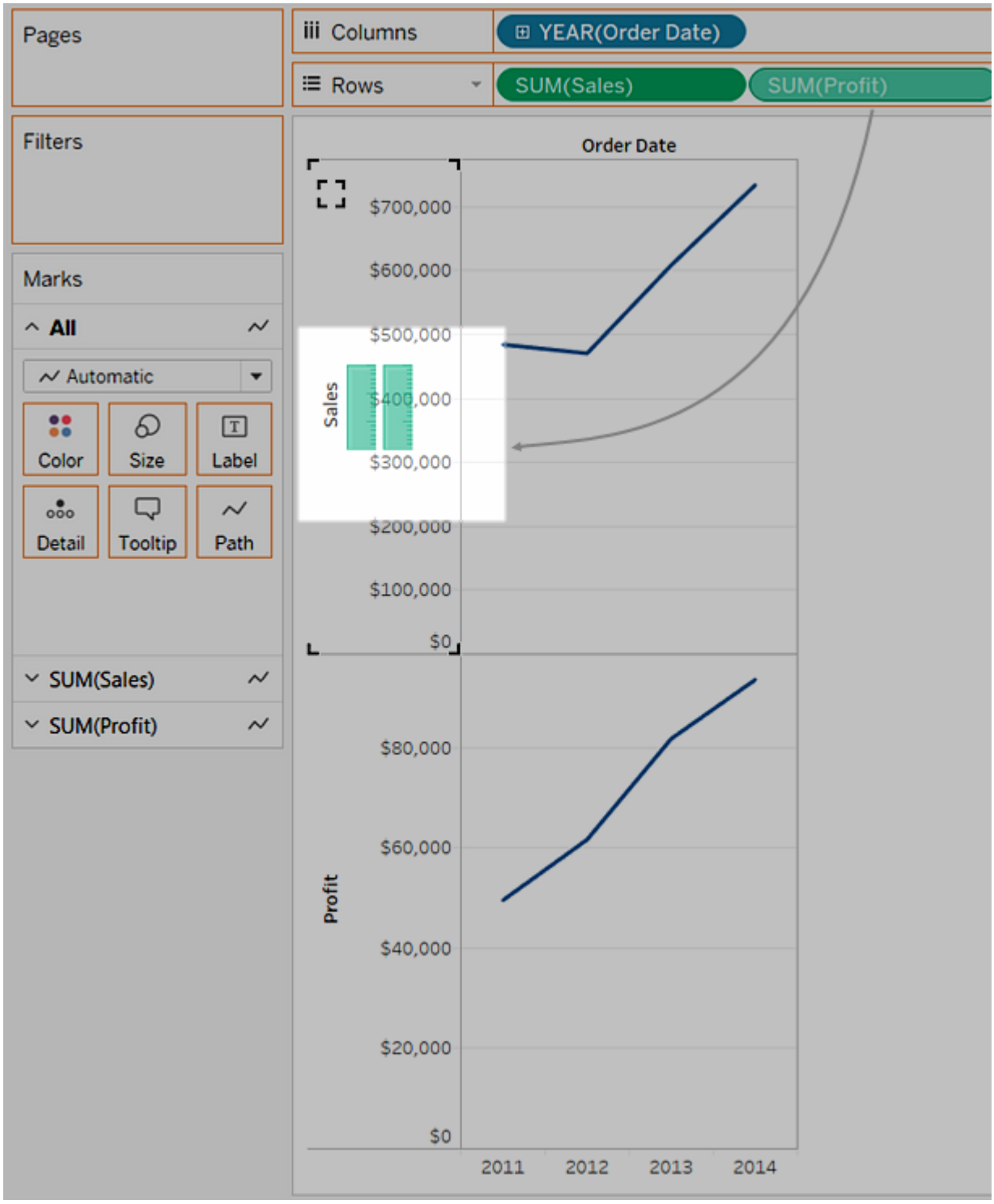
   Tableau creates separate axes along the left margin for **Sales** and **Profit**.



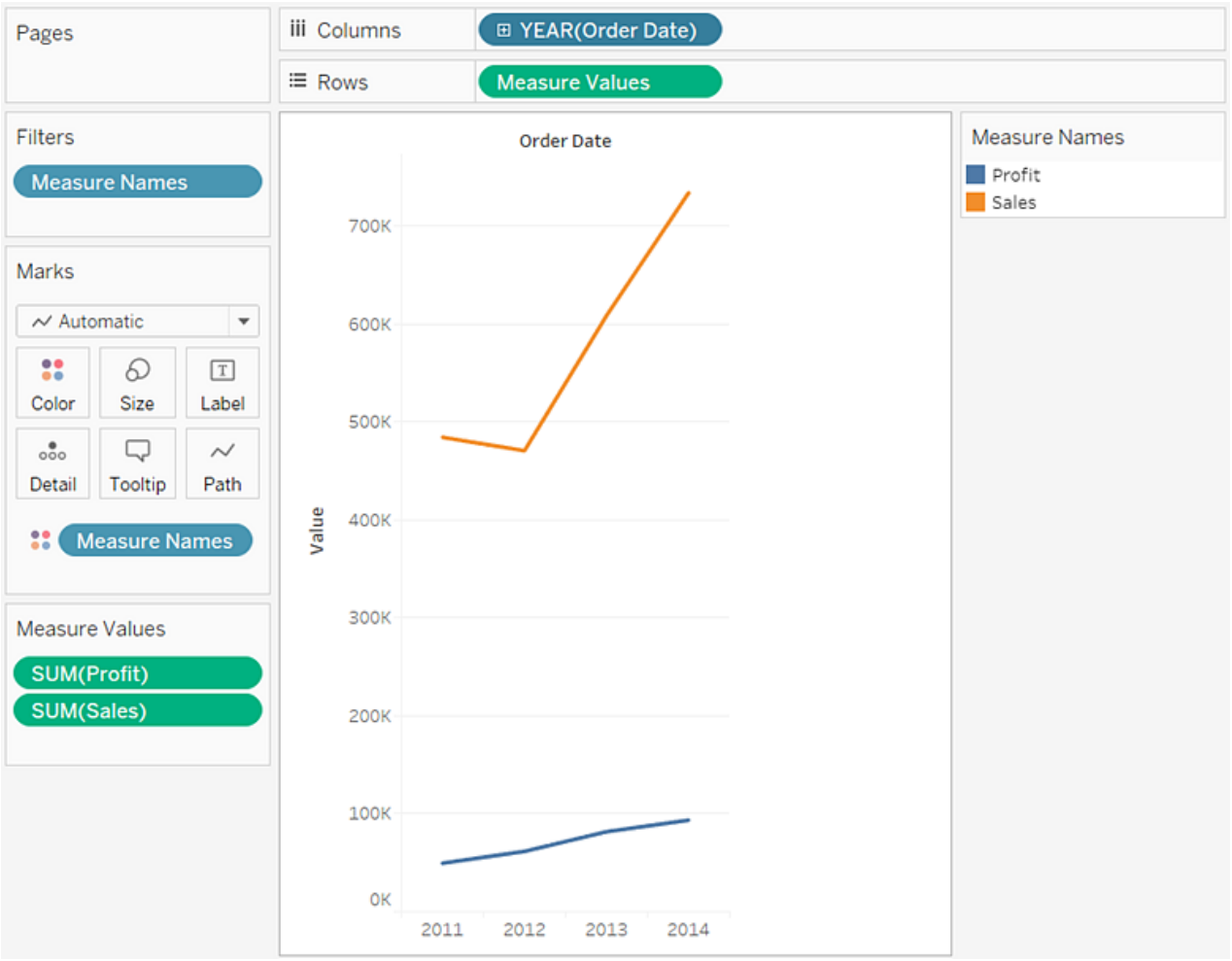
Notice that the scale of the two axes is different—the **Sales** axis scales from $0 to $700,000, whereas the **Profit** axis scales from $0 to $100,000. This can make it hard to see that sales values are much greater than profit values.

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.

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.

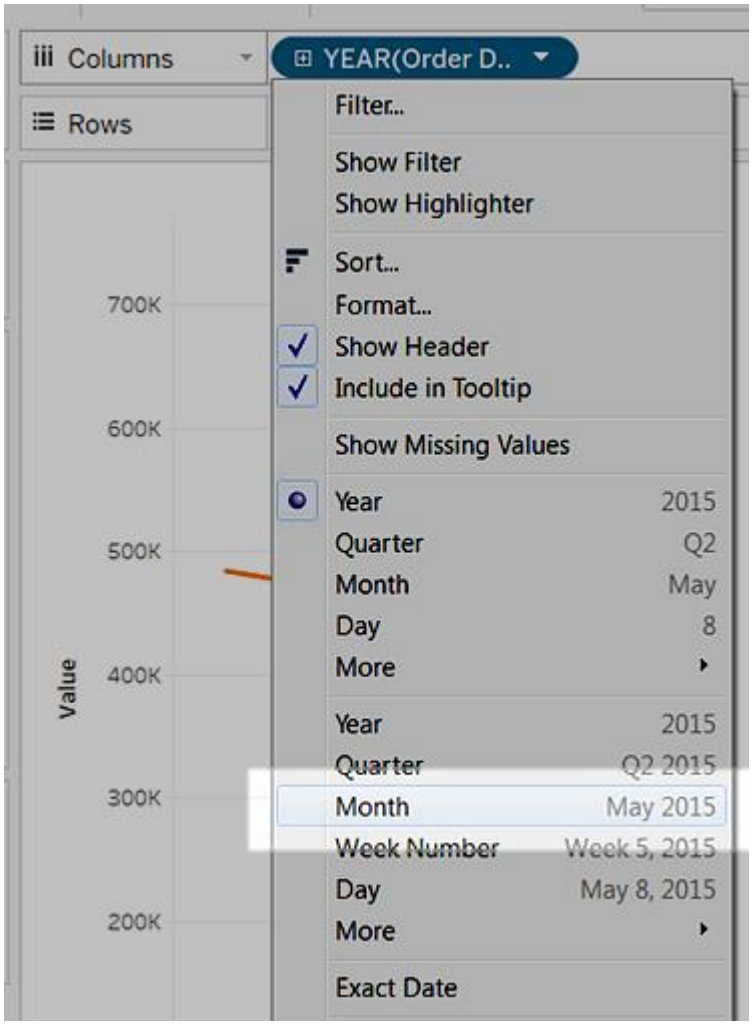

Pages
Columns
YEAR(Order Date)
Rows
SUM(Sales)
SUM(Profit)
Filters
Marks
All
Automatic
Color
Size
Label
Detail
Tooltip
Path
SUM(Sales)
SUM(Profit)
Order Date
$700,000
$600,000
$500,000
$400,000
$300,000
$200,000
$100,000
$0
Sales
$80,000
$60,000
$40,000
$20,000
$0
Profit
2011
2012
2013
2014

:

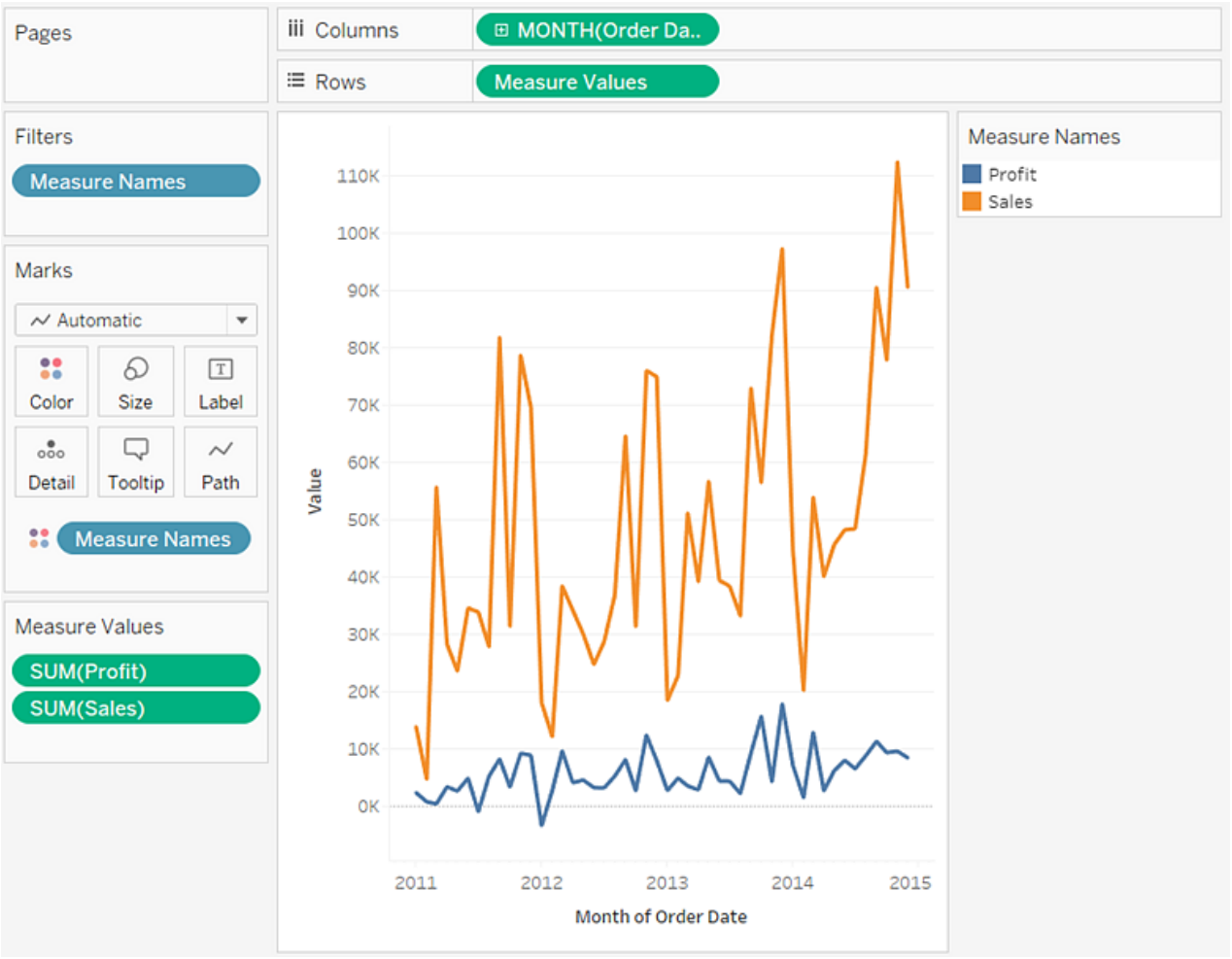


The view is rather sparse because we are looking at a summation of values on a per-year basis.

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.

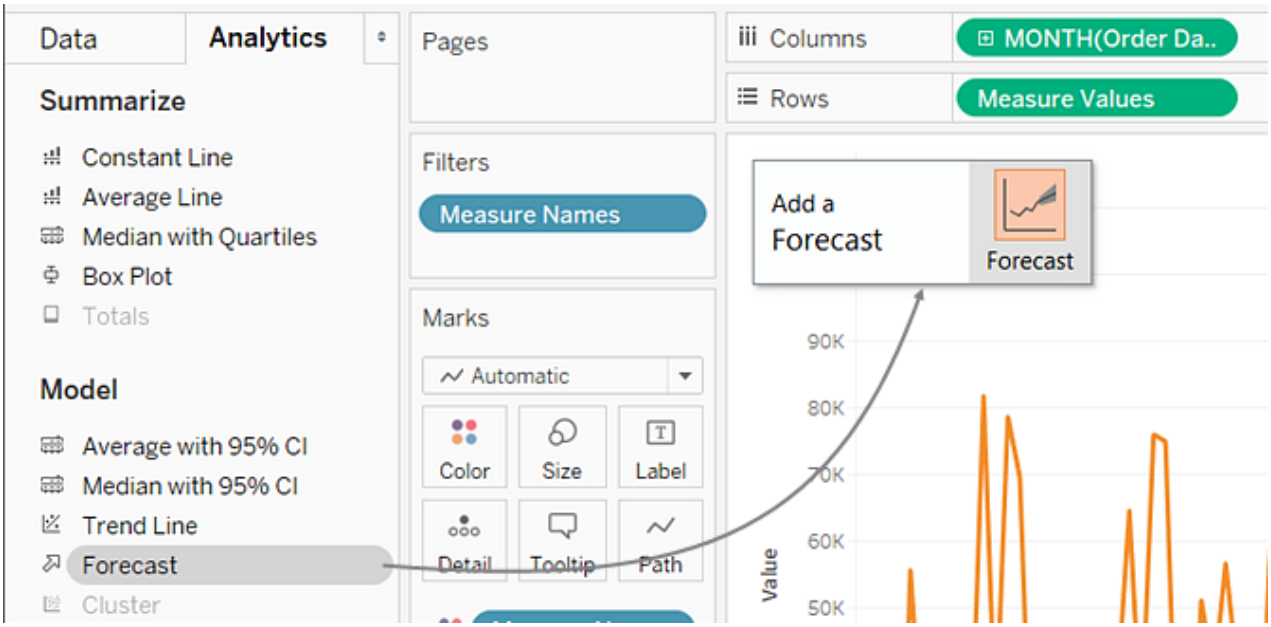


The resulting view is a lot more detailed than the original view:

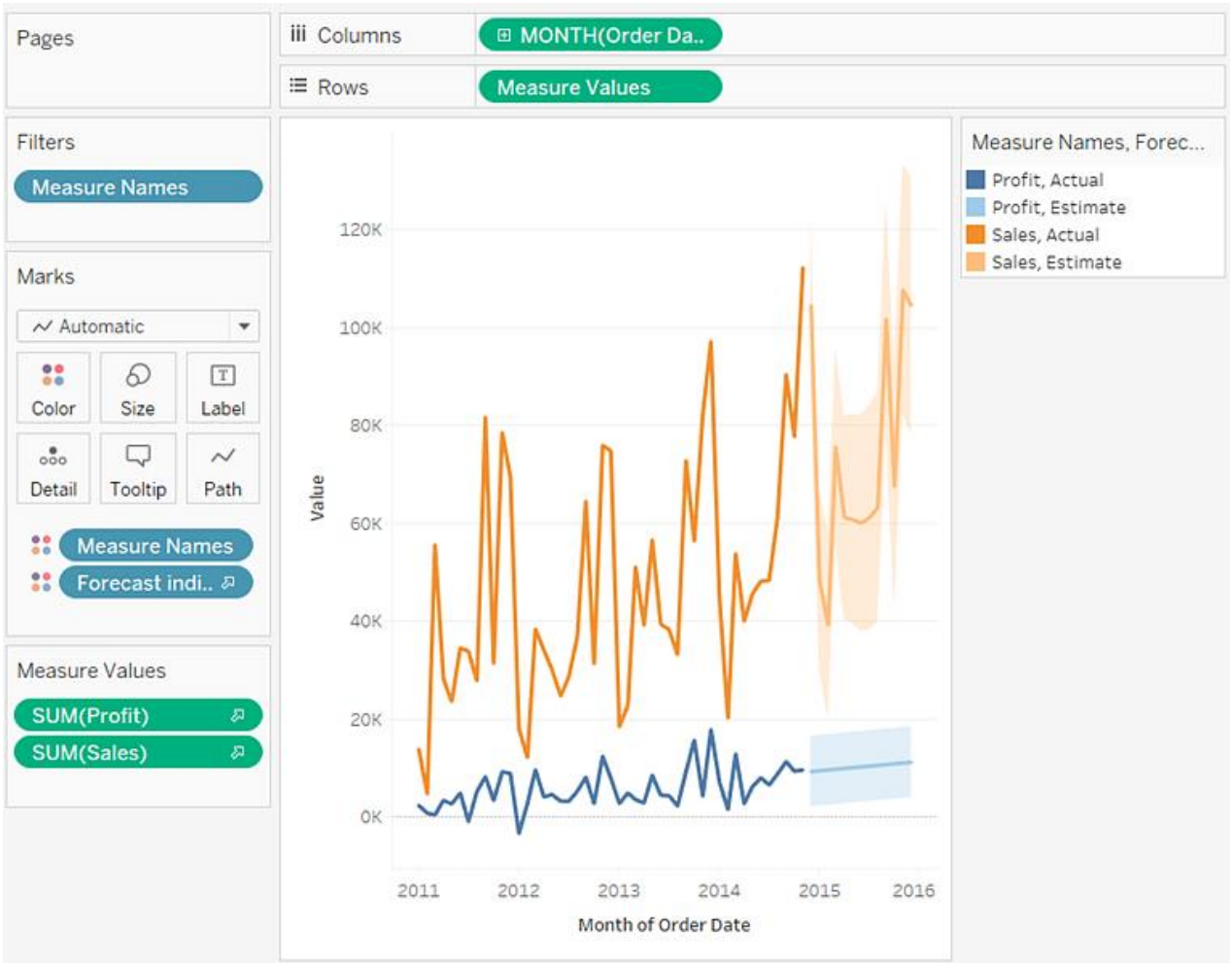


Notice that the values seem to go much higher just before the end of each year. A pattern like that is known as *seasonality*. If we turn on the forecasting feature in the view, we can see whether we should expect that the apparent seasonal trend will continue in the future.

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:



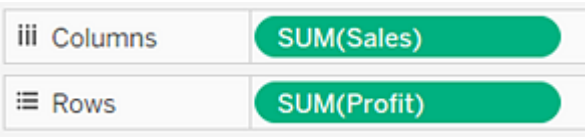
Practice 5: Graphical Tools for data elaboration – 2

1. 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.

Creates Simple Scatter Plot



Creates Matrix of Scatter Plots

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..

To use scatter plots and trend lines to compare sales to profit, follow these steps:

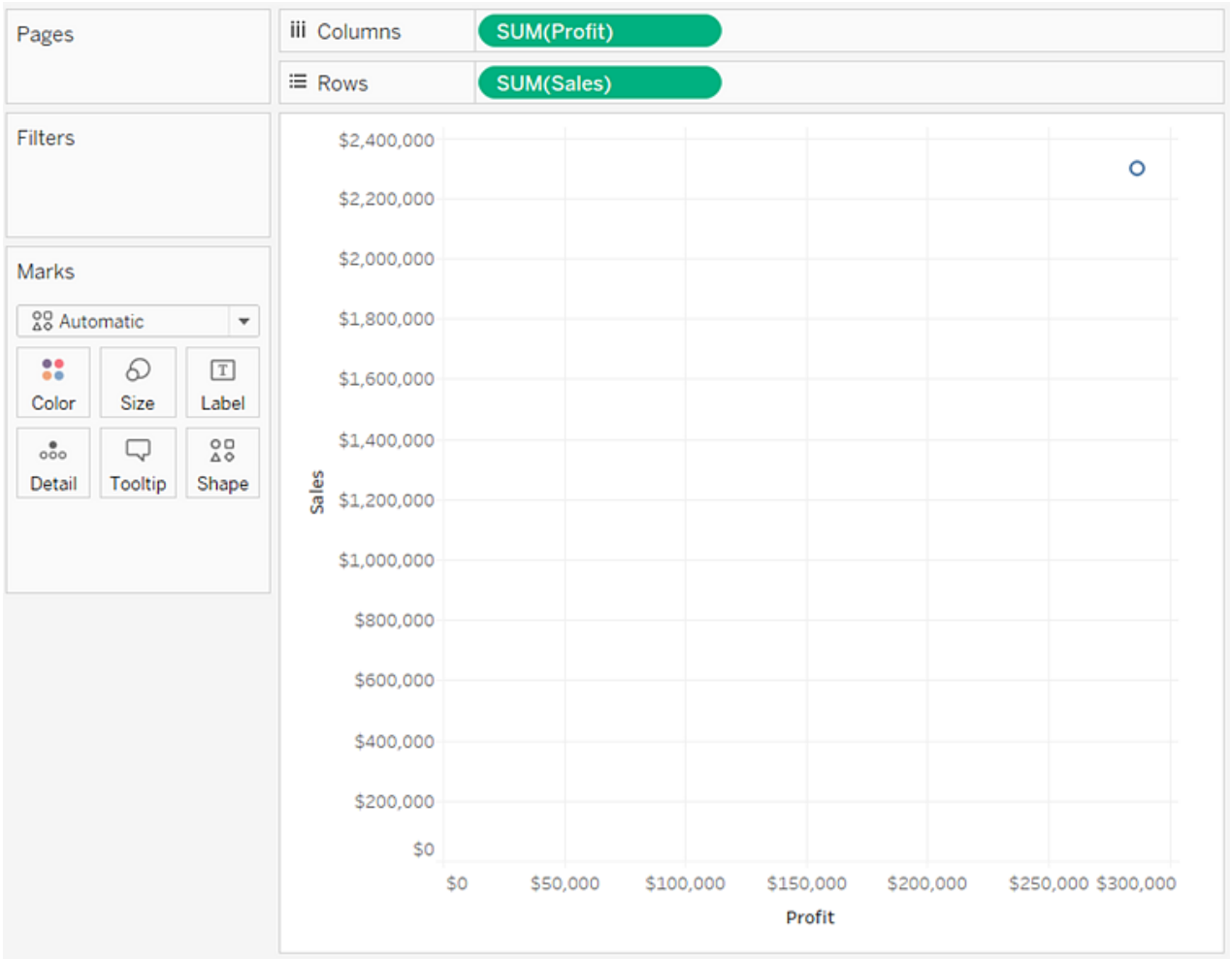
1. Open the **Sample - Superstore** data source.
2. Drag the **Profit** measure to **Columns**.

   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.

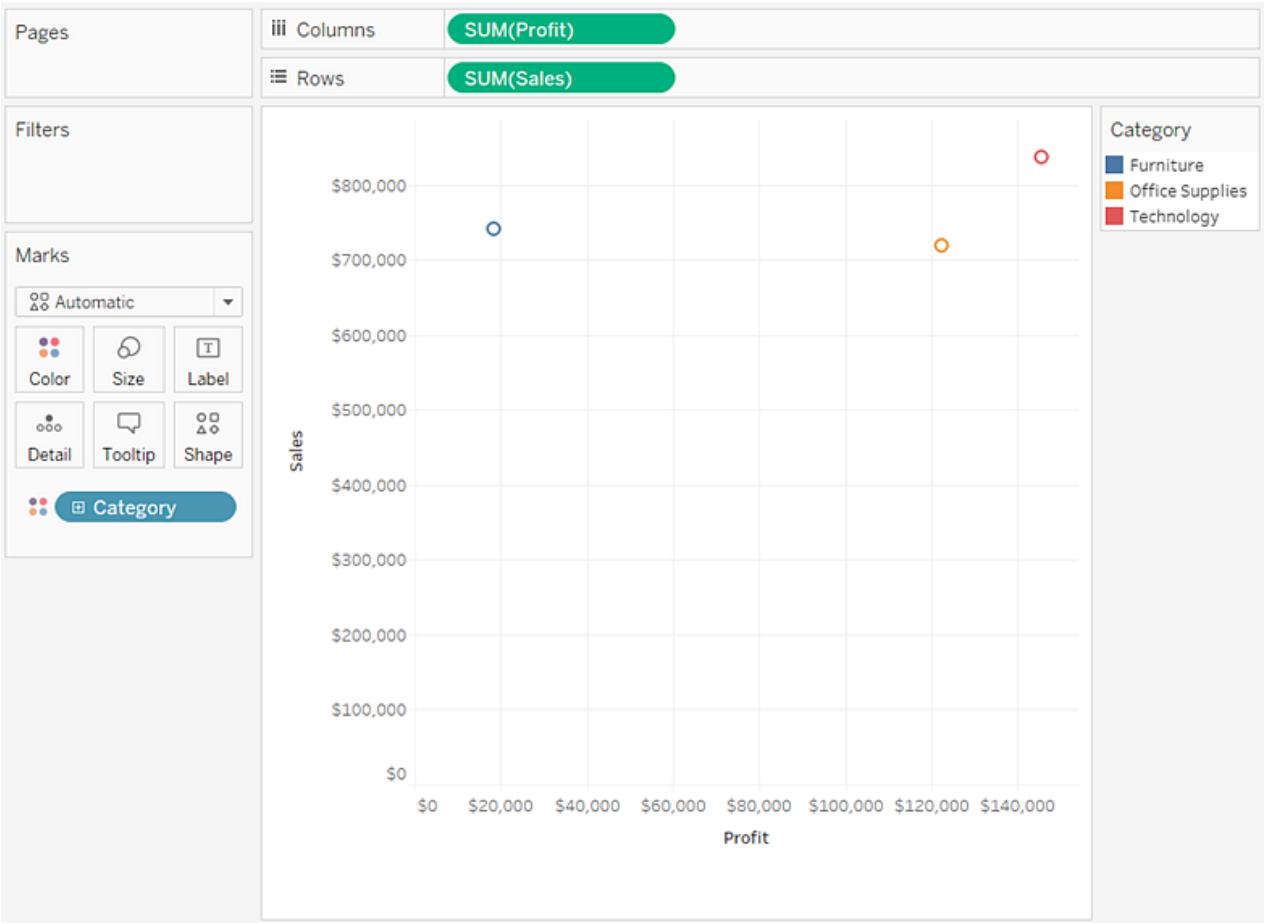
   Measures can consist of continuous numerical data. When you plot one number against another, you are comparing two numbers; the resulting chart is analogous to a Cartesian chart, with x and y coordinates.

   Now you have a one-mark scatter plot:



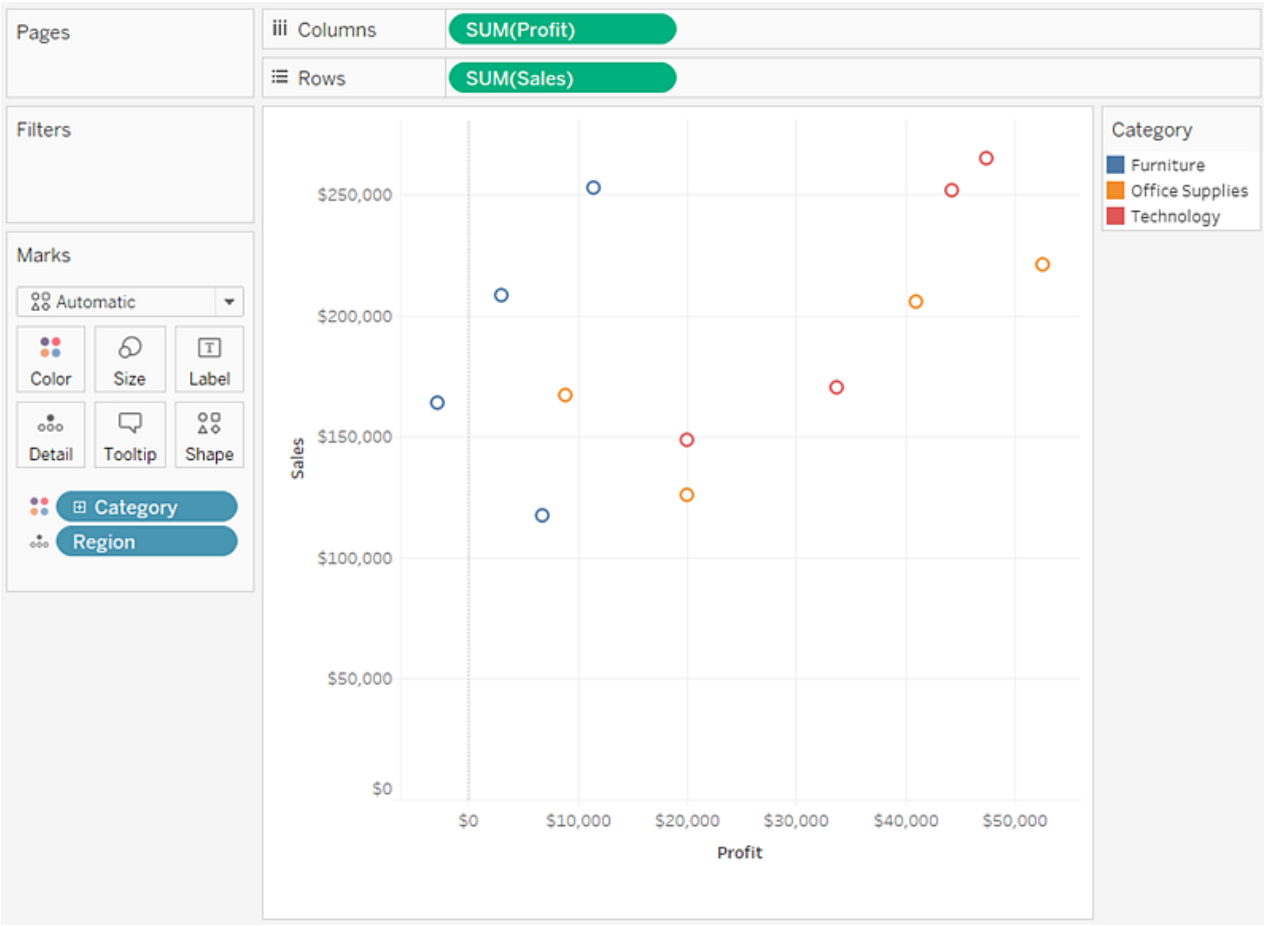
4. Drag the **Category** dimension to **Color** on the Marks card.

   This separates the data into three marks—one for each dimension member—and encodes the marks using color.

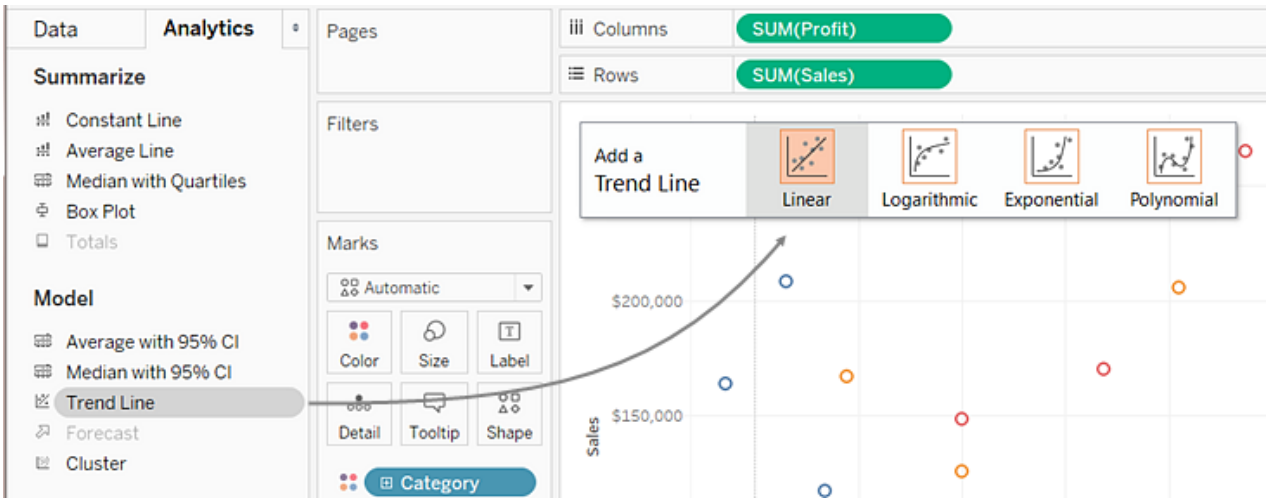


5. Drag the **Region** dimension to **Detail** on the **Marks** card.

   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**.)

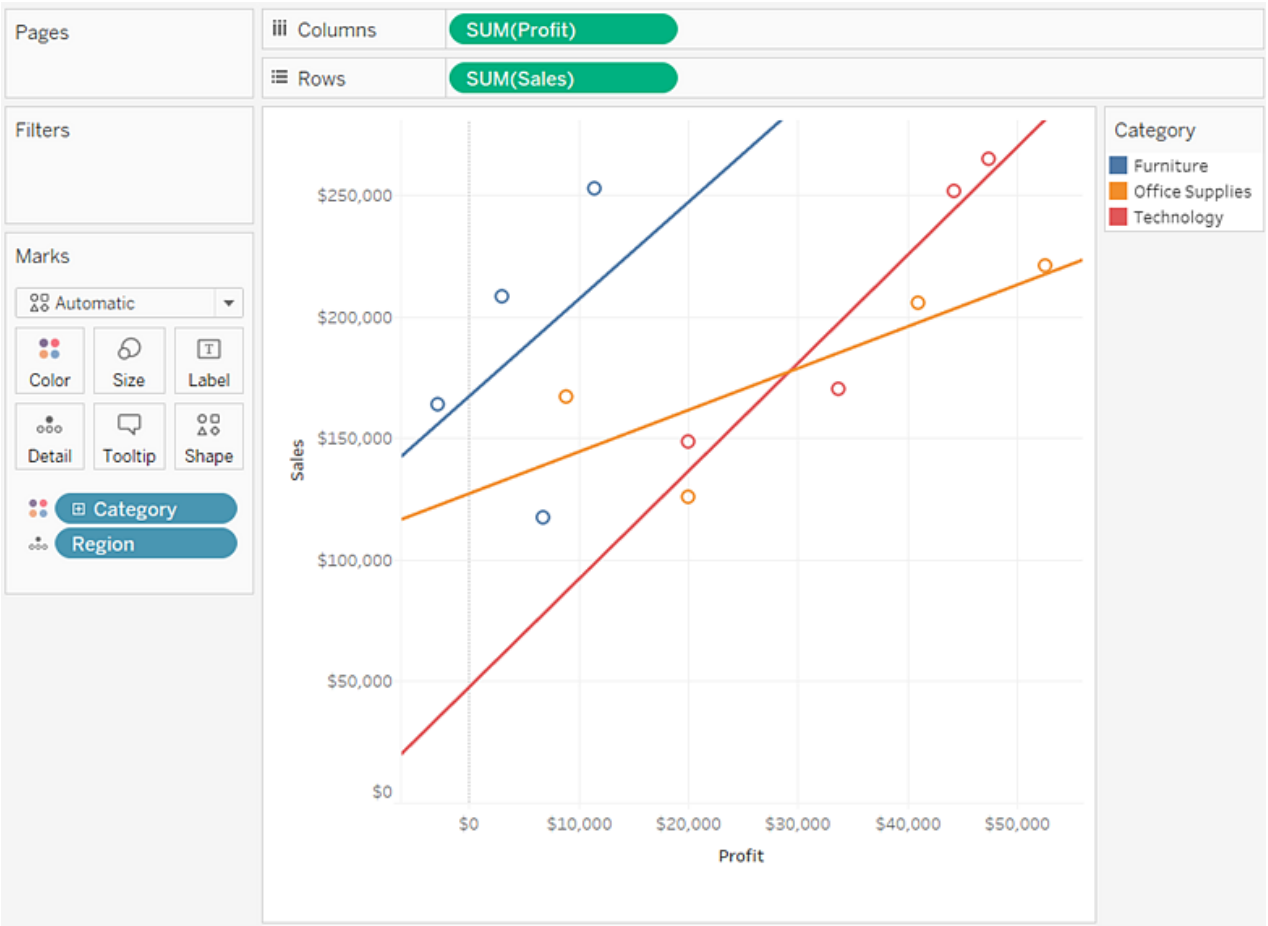


6. To add trend lines, from the **Analytics** pane, drag the **Trend Line** model to the view, and then drop it on the model type.

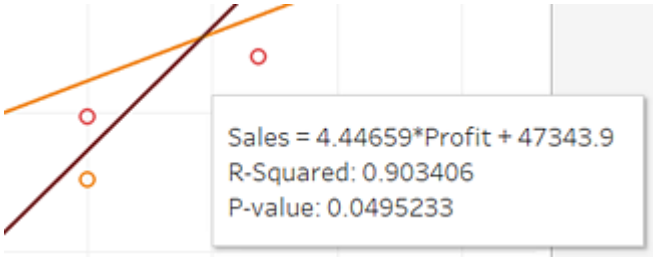


A trend line can provide a statistical definition of the relationship between two numerical values. To add trend lines to a view, both axes must contain a field that can be interpreted as a number—by definition, that is always the case with a scatter plot.

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:



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

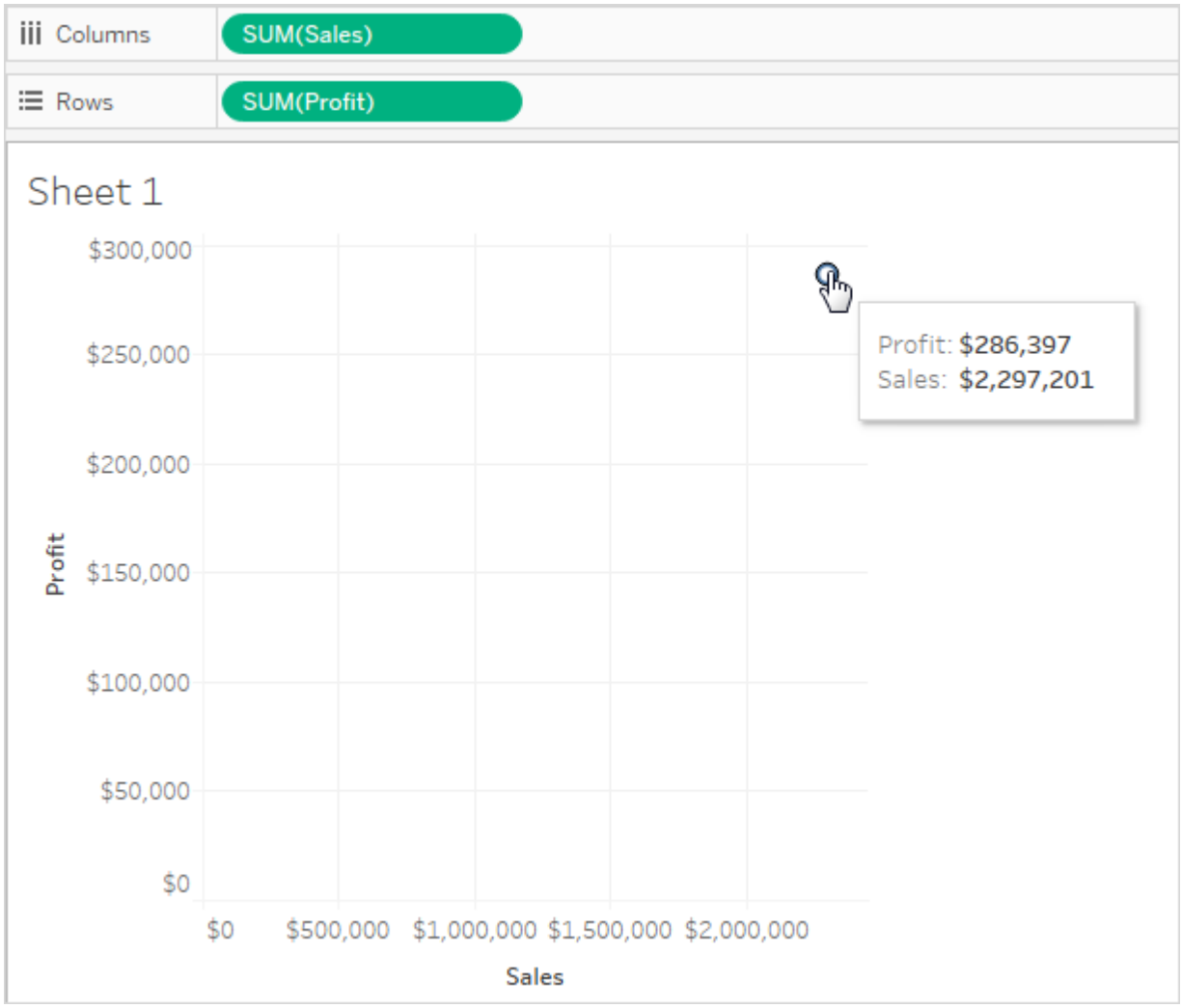
There are various ways to add detail to a basic scatter plot: **you can use dimensions to add detail**, you can add additional measures and/or dimensions to the Rows and Columns shelves to create multiple one-mark scatter plots in the view, or you can **disaggregate the data**. And, you can also use 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.



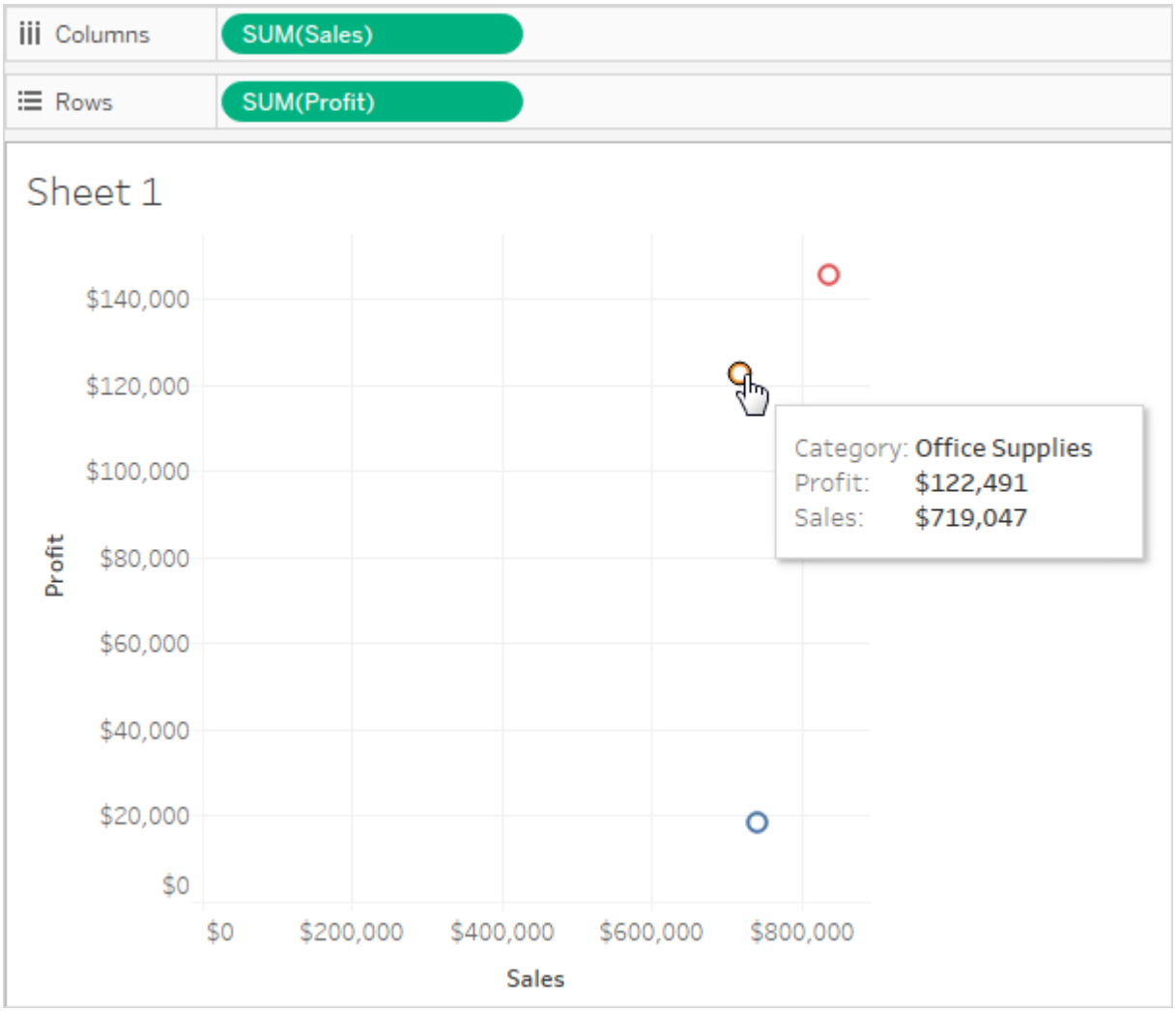
Follow the steps below to use dimensions to add detail to the view and to disaggregate data.

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.

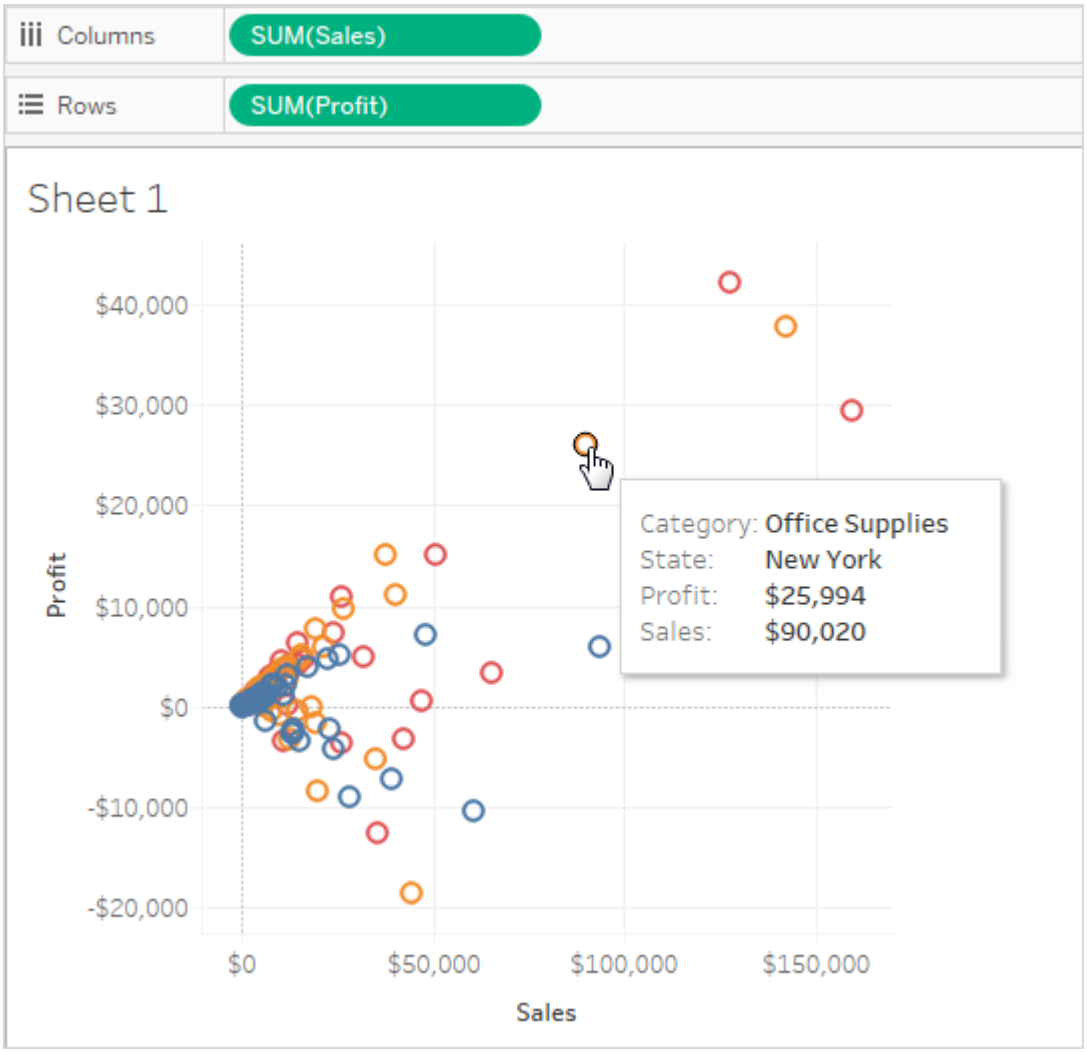
1. Drag the **Category** dimension to **Color** on the Marks card.

   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.

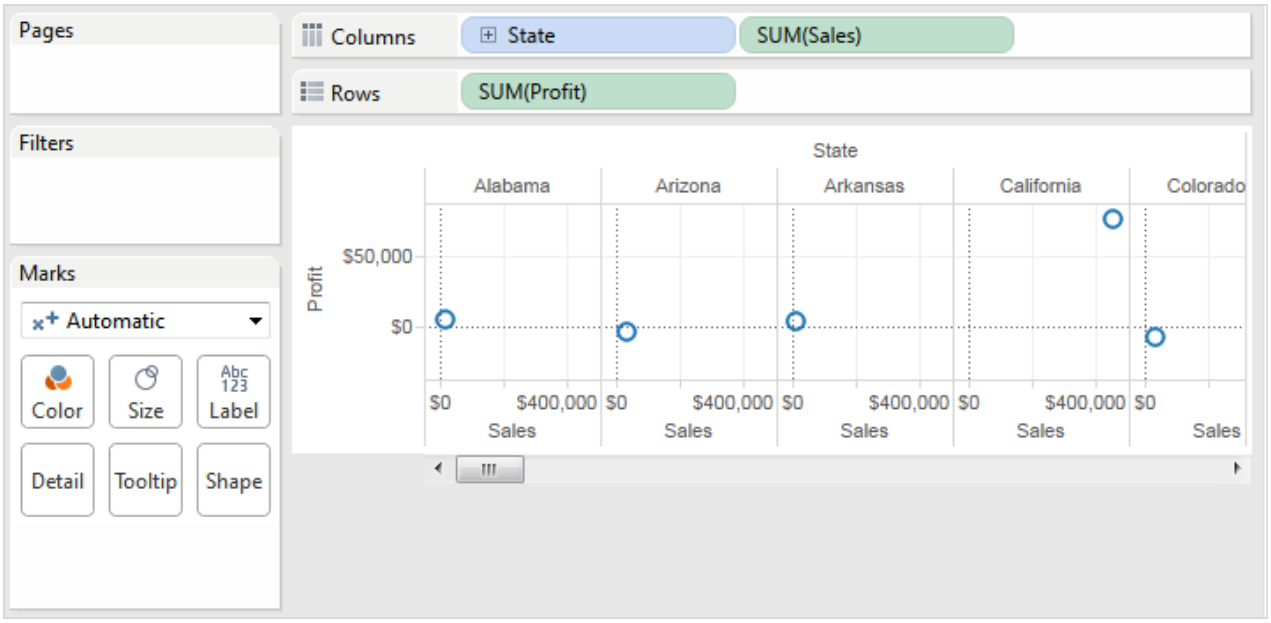
Maybe this process is developing the view in a direction you find useful, or maybe you prefer to go in a different direction—for example, by adding a time dimension to the view, or by introducing trend lines or forecasting. You decide what questions to ask.

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.

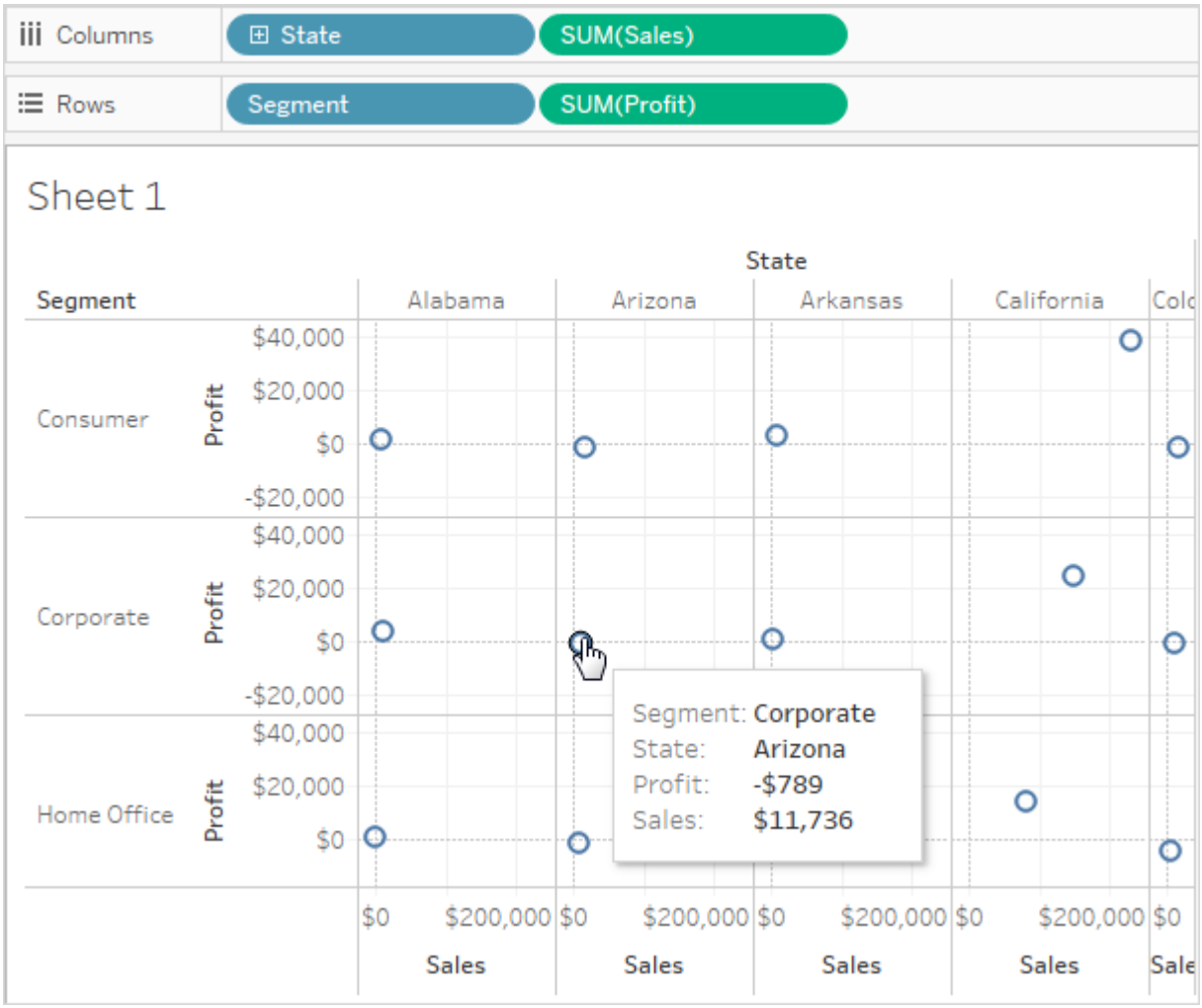
1. Drag the **State** dimension to the **Columns** shelf.

   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.

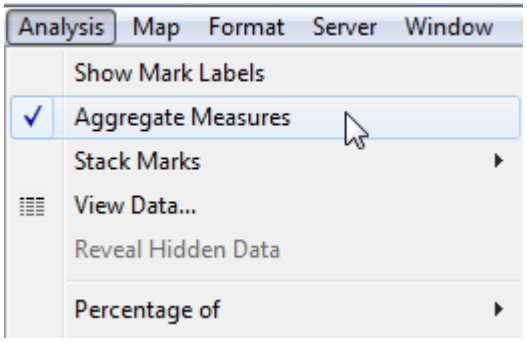
   You now have a view that provides an overview of Sales and Profit across states and customer segments. It can be interesting to hover over the marks in the view to see tooltip data for various segments:



Try disaggregating the data

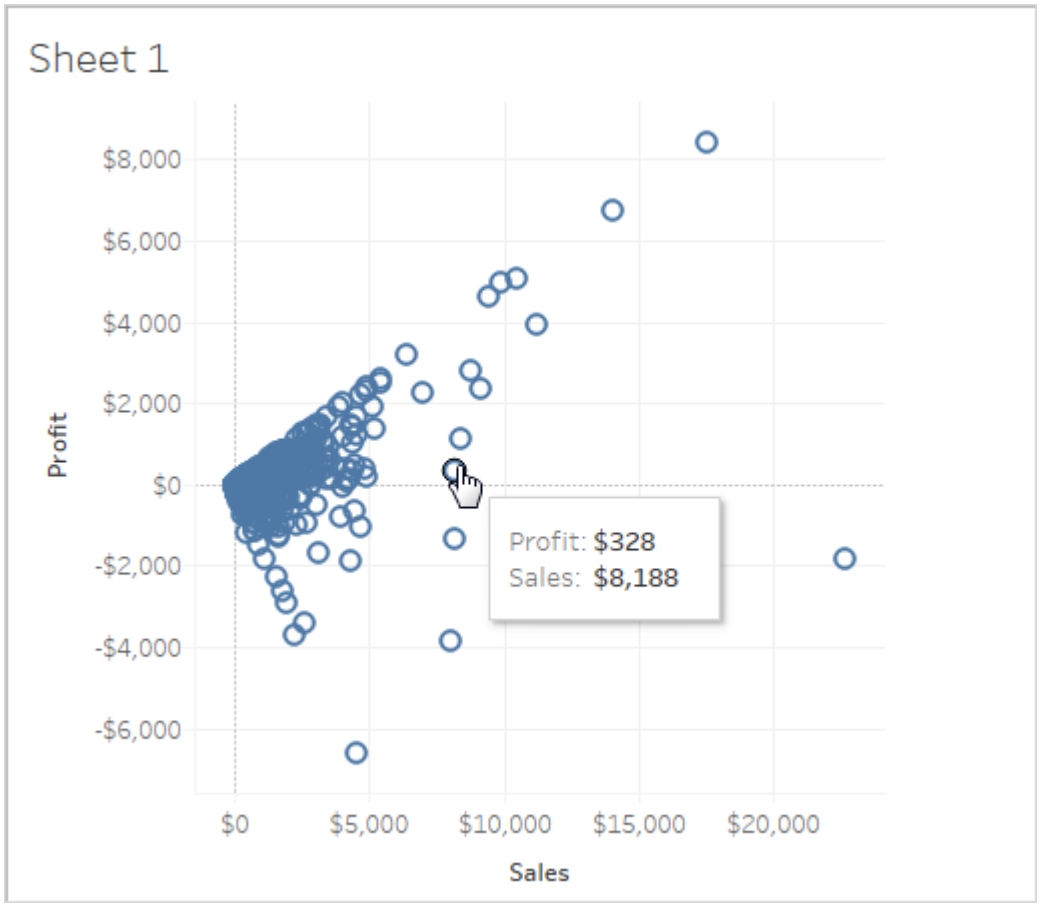
Another way to modify your original one-mark scatter plot to display more marks is by disaggregating the data.

Clear the **Analysis >Aggregate Measures** option. If it is already selected, click **Aggregate Measures** once to deselect it.



What you have actual 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:



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.

2. Histogram in Tableau

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:

Mark type:	Automatic
Rows shelf:	Continuous measure (aggregated by Count or Count Distinct)
Columns shelf:	Bin (continuous or discrete). ***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(Link opens in a new window).*

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.

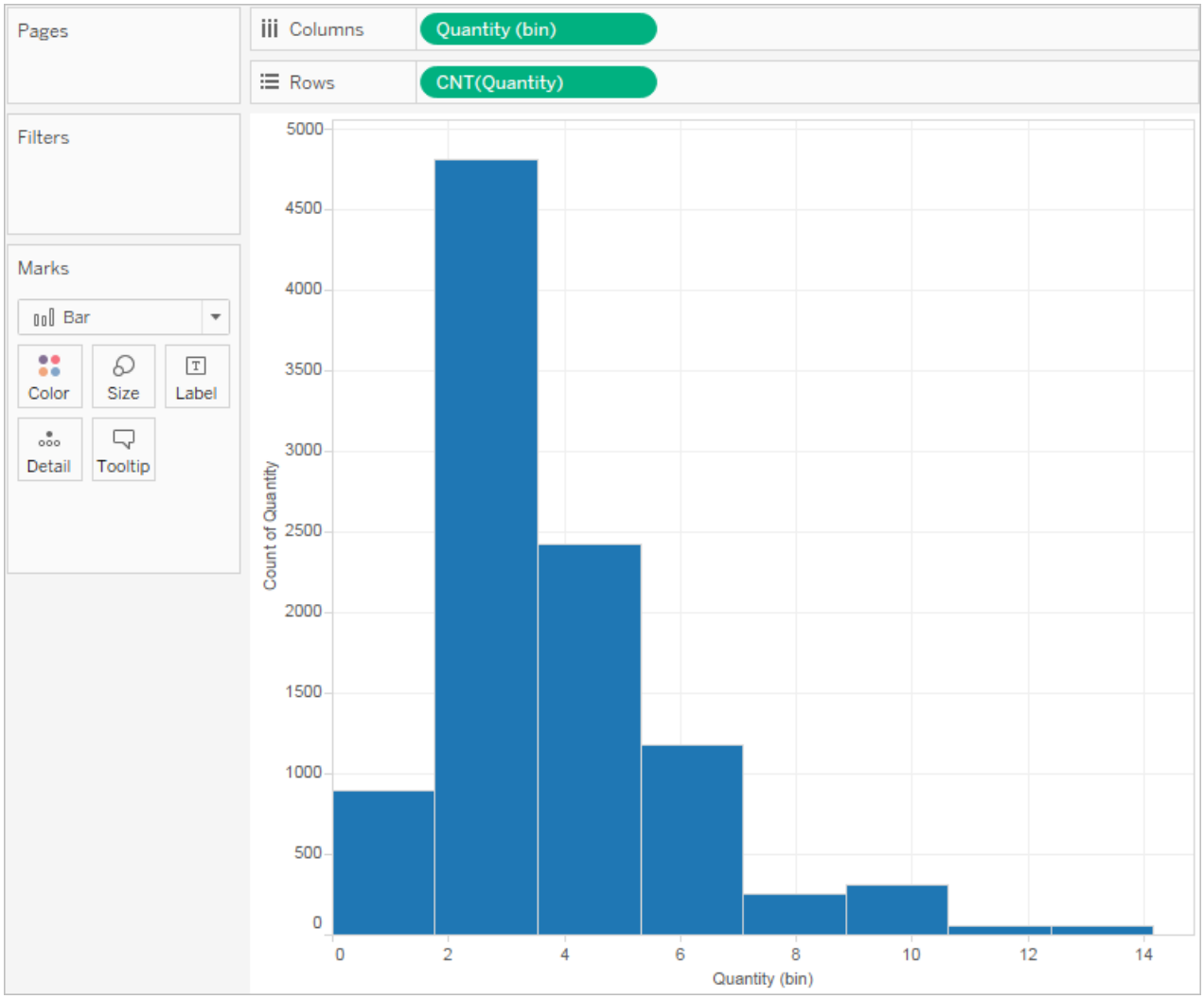
Three things happen after you click the histogram icon in **Show Me**:

- 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, right-click the bin and select **Edit**.

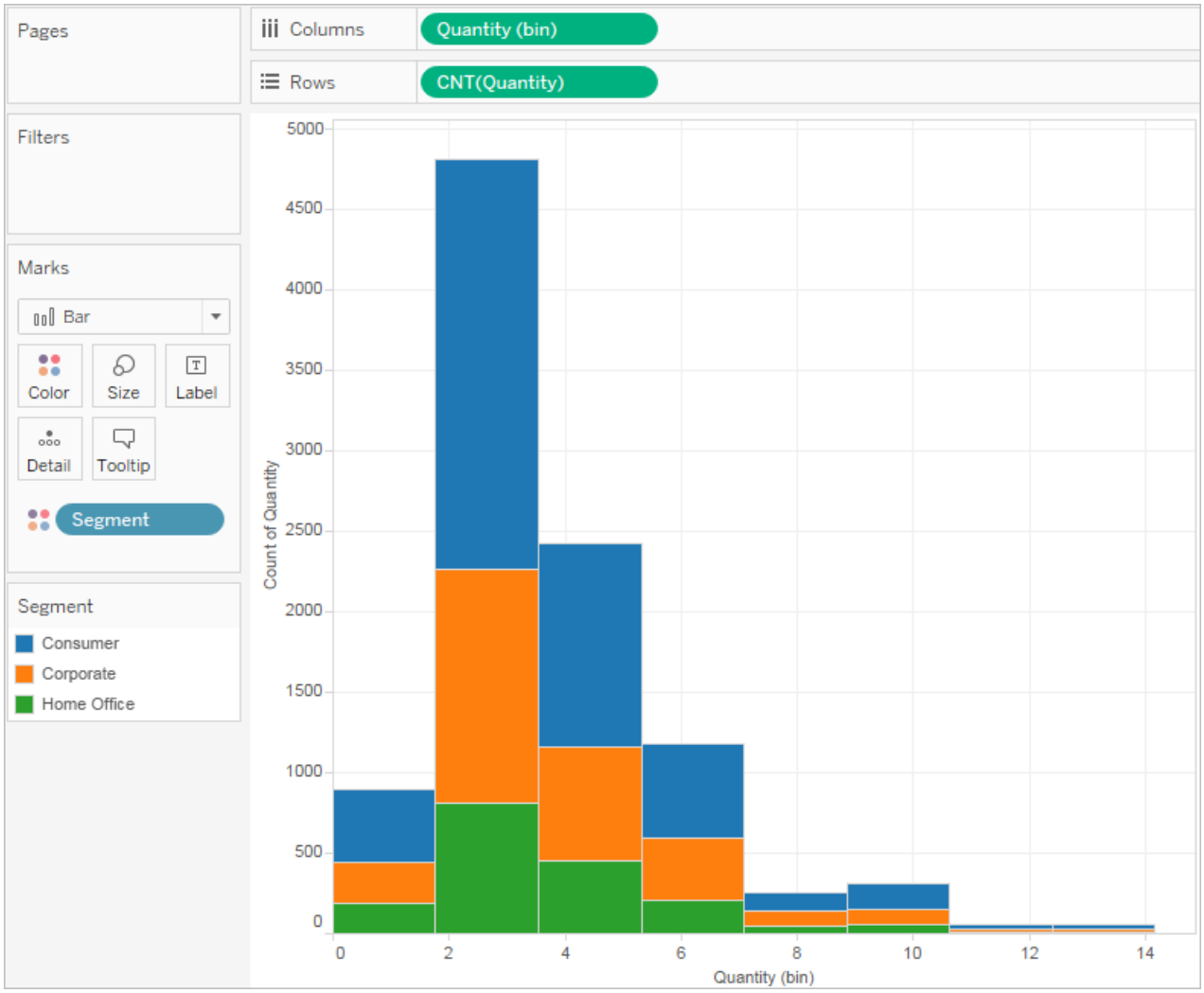
- The **Quantity** measure moves to the **Rows** shelf and the aggregation changes from SUM to CNT (Count).



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.

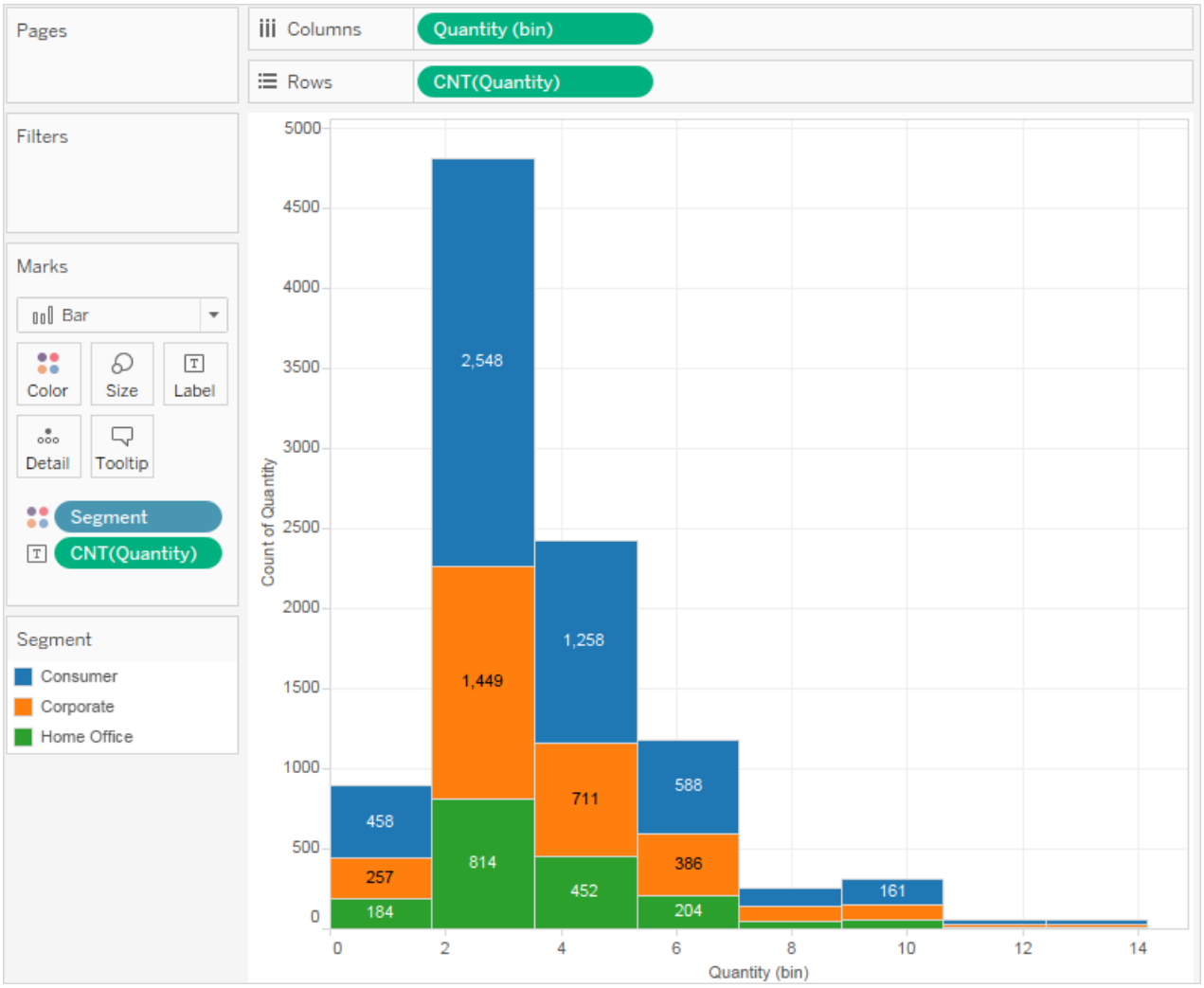
Let's take this view one step further and add **Segment** to **Color** to see if we can detect a relationship between the customer segment (consumer, corporate, or home office) and the quantity of items per order.

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.

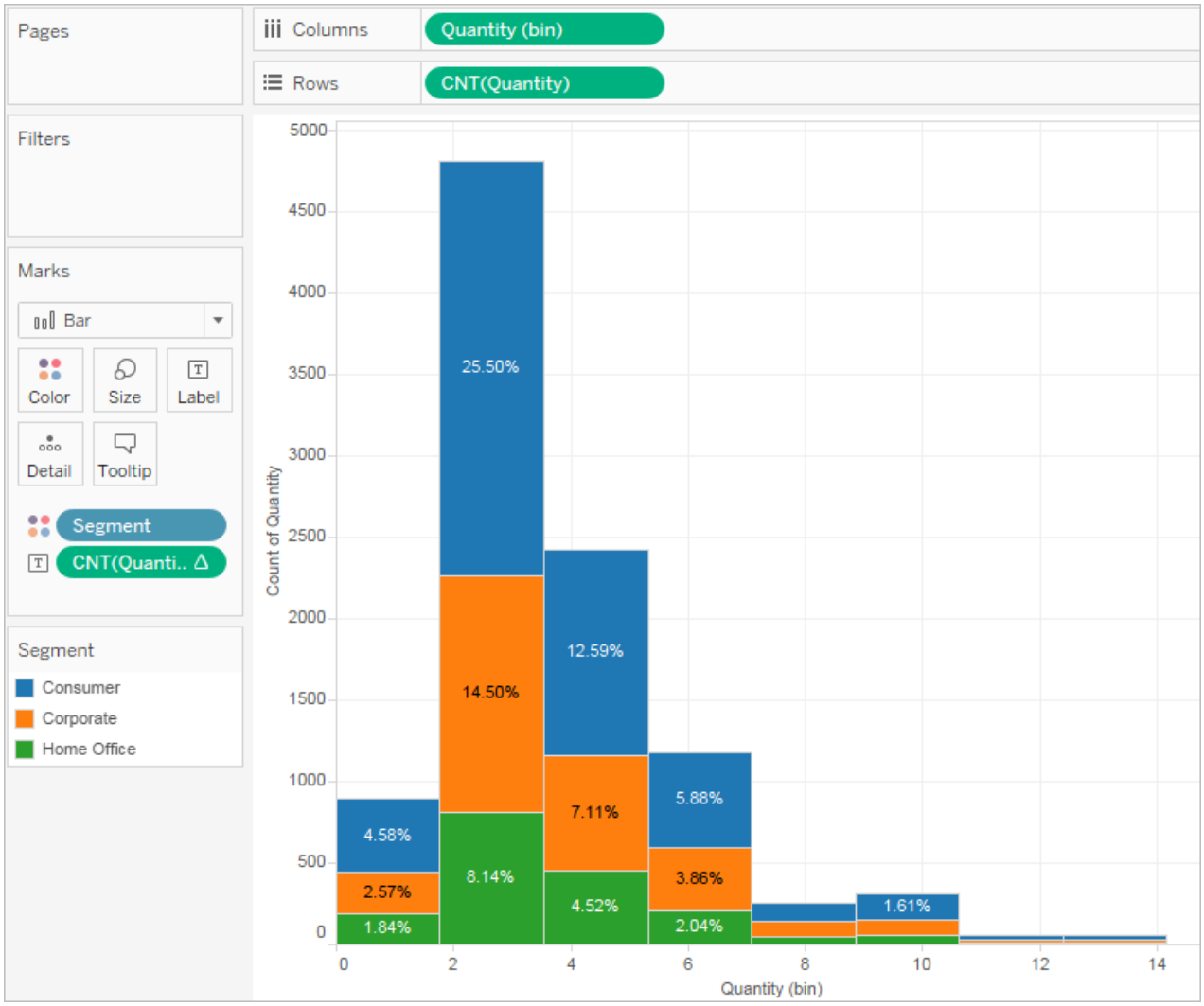
5. Hold down the Ctrl key and drag the **CNT(Quantity)** field from the **Rows** shelf to **Label**.



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**.

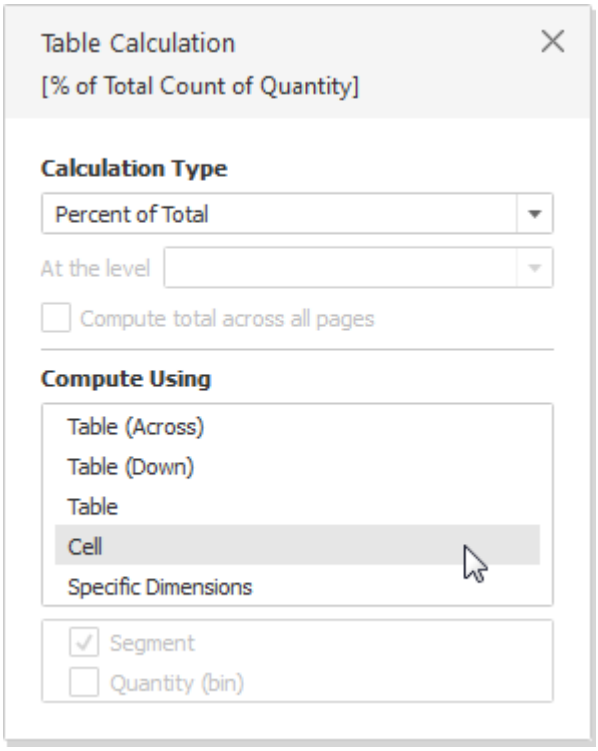
   Now each colored section of each bar shows its respective percentage of the total quantity:



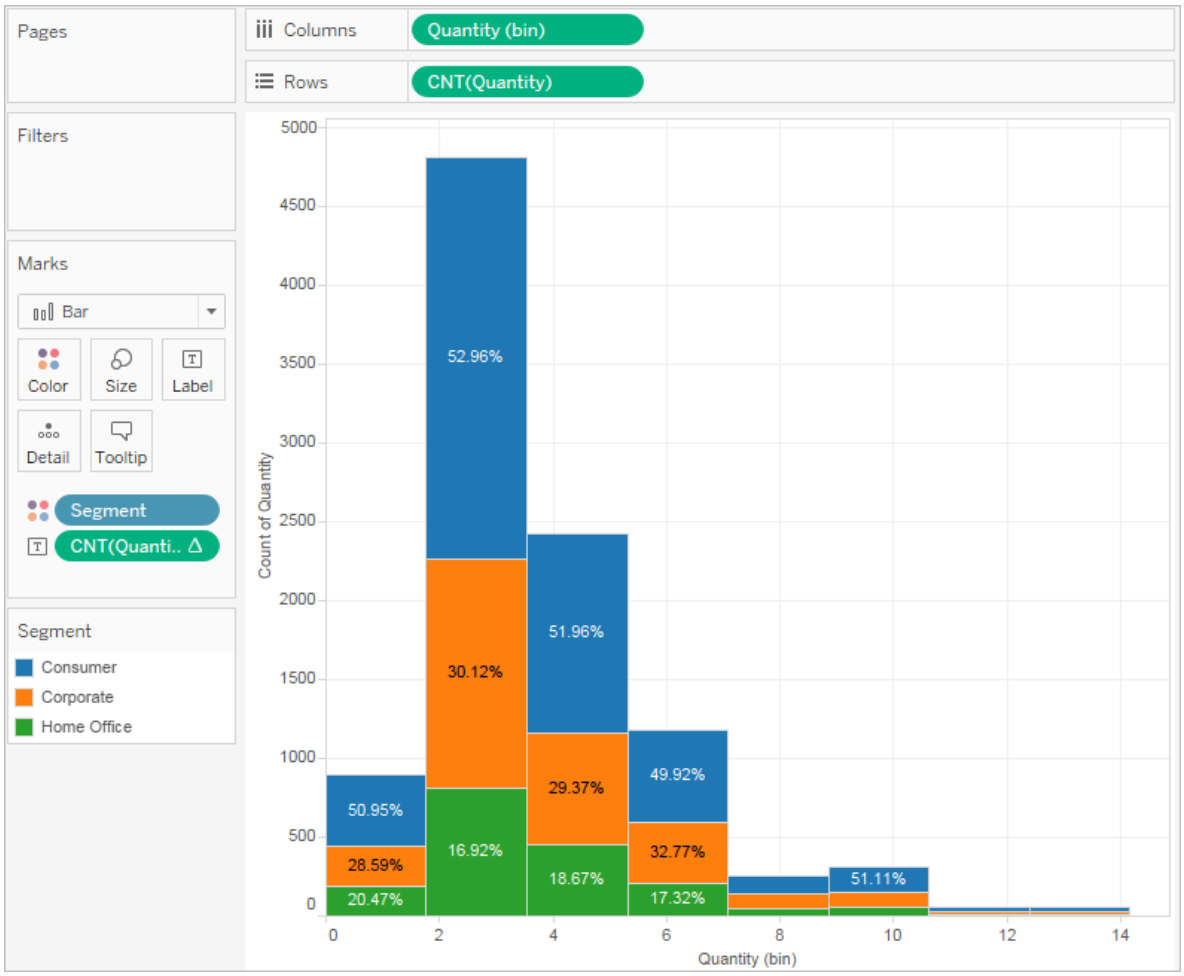
But we want the percentages to be on a per-bar basis.

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.

3. Heatmap

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 heat map.

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).

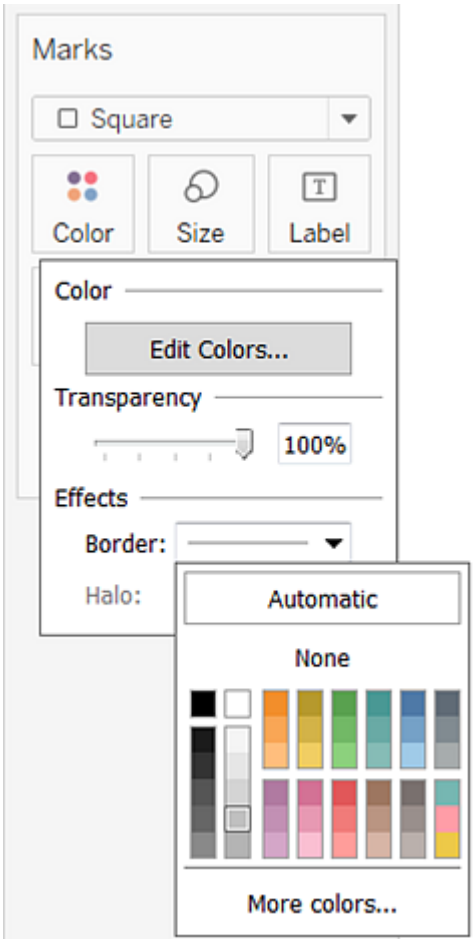
4. Drag the **Profit** measure to **Color** on the **Marks** card.

   Tableau aggregates the measure as a sum. The color legend reflects the continuous data range.

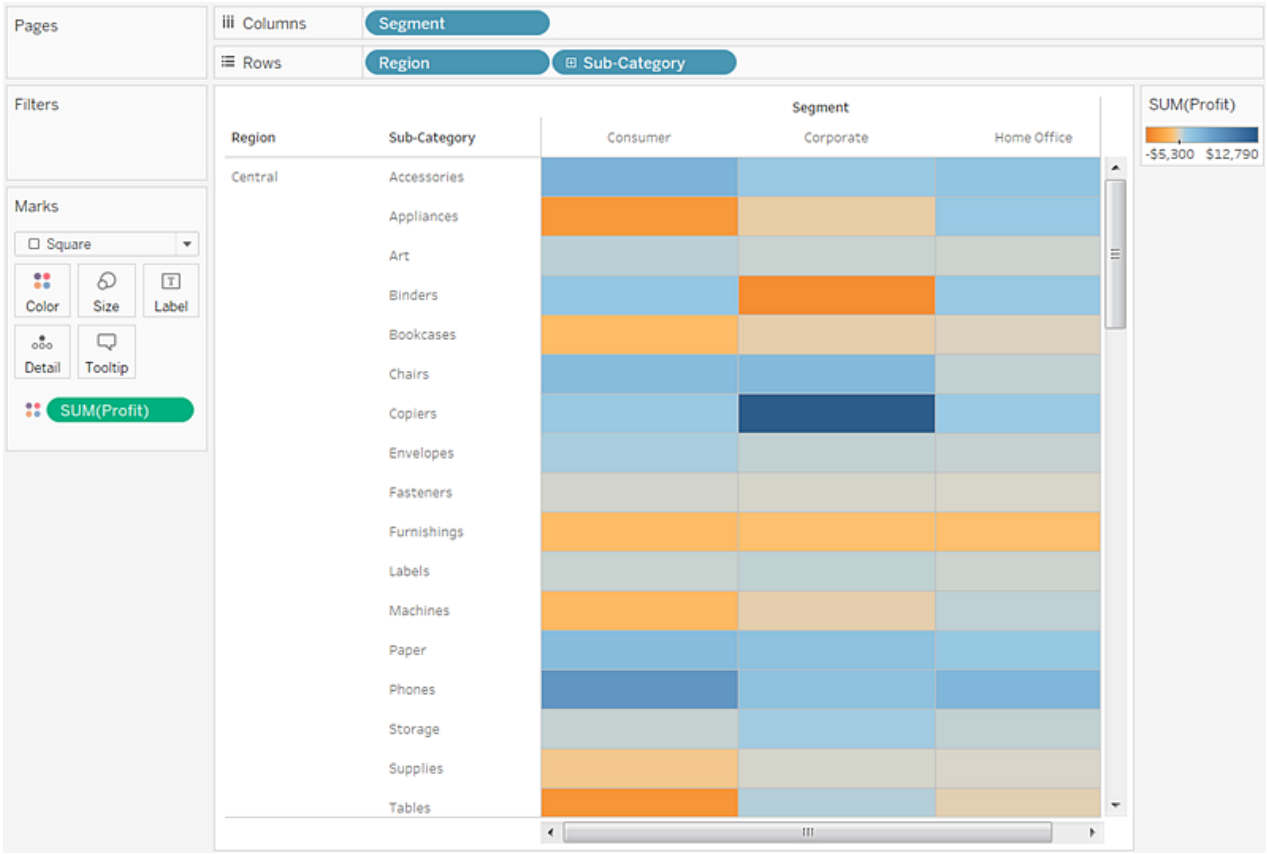
In this view, you can see data for only the Central region. Scroll down to see data for other regions.

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:

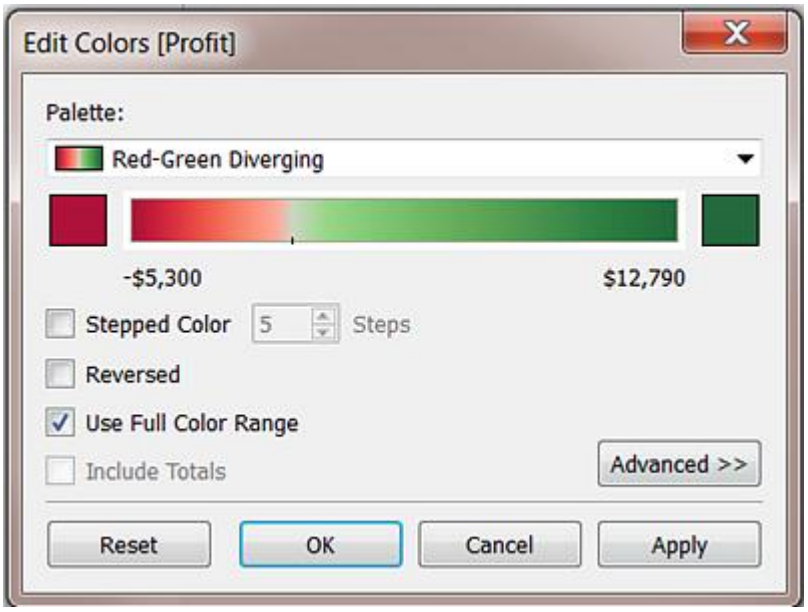


6. The default color palette is Orange-Blue Diverging. A Red-Green Diverging palette might be more appropriate for profit. To change the color palette and to make the colors more distinct, do the following:

    - Hover over the **SUM(Profit)** color legend, then click the drop-down arrow that appears and select **Edit Colors**.

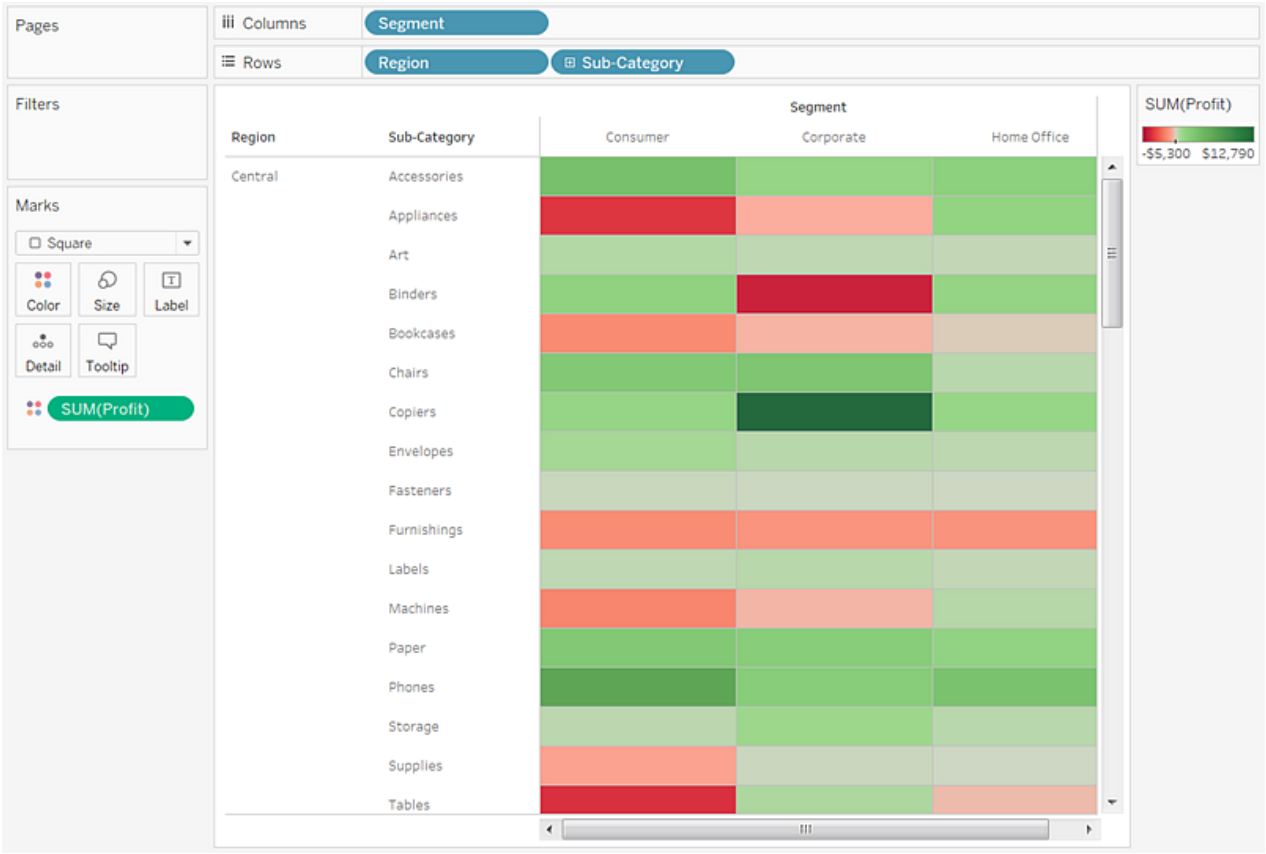
    - In the **Edit Colors** dialog box, in the **Palette** field, select **Red-Green Diverging** from the drop-down list.

    - Select the **Use Full Color Range** check box and click **Apply** and then click **OK**.



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.

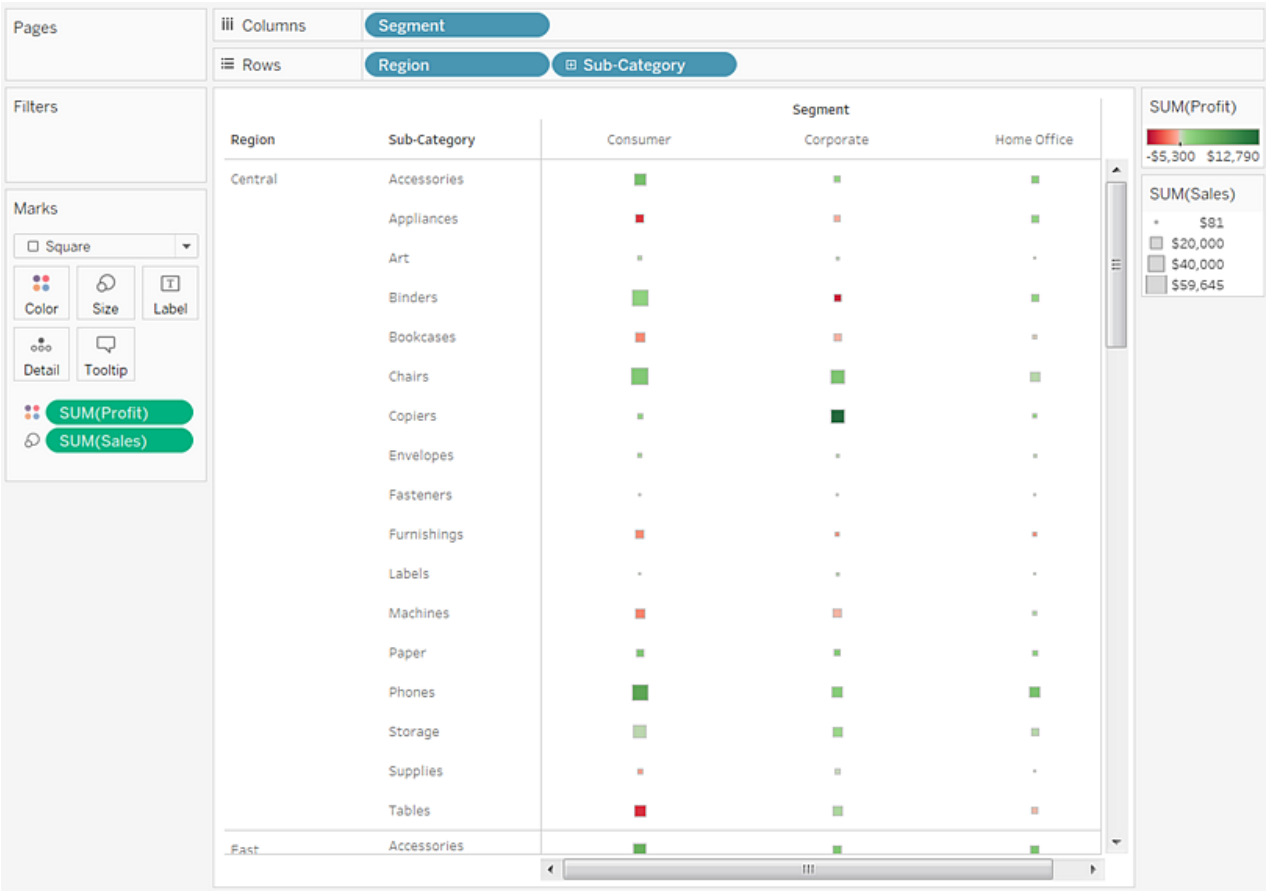
When you do not select **Use Full Color Range**, Tableau assigns the color intensity as if the range was from -100 to 100, so that the change in shade is the same on both sides of zero. The effect is to make the color contrasts in your view much more distinct.



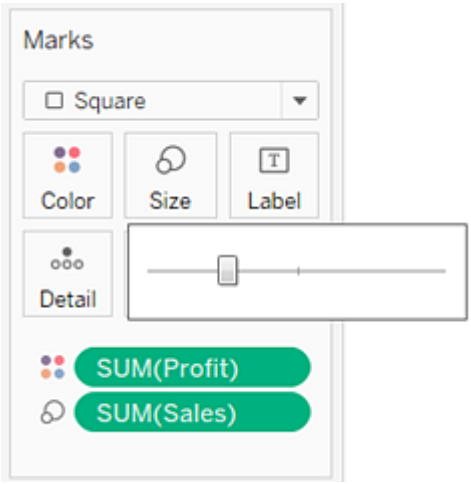
Modify the size to create a heat map

7. 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).

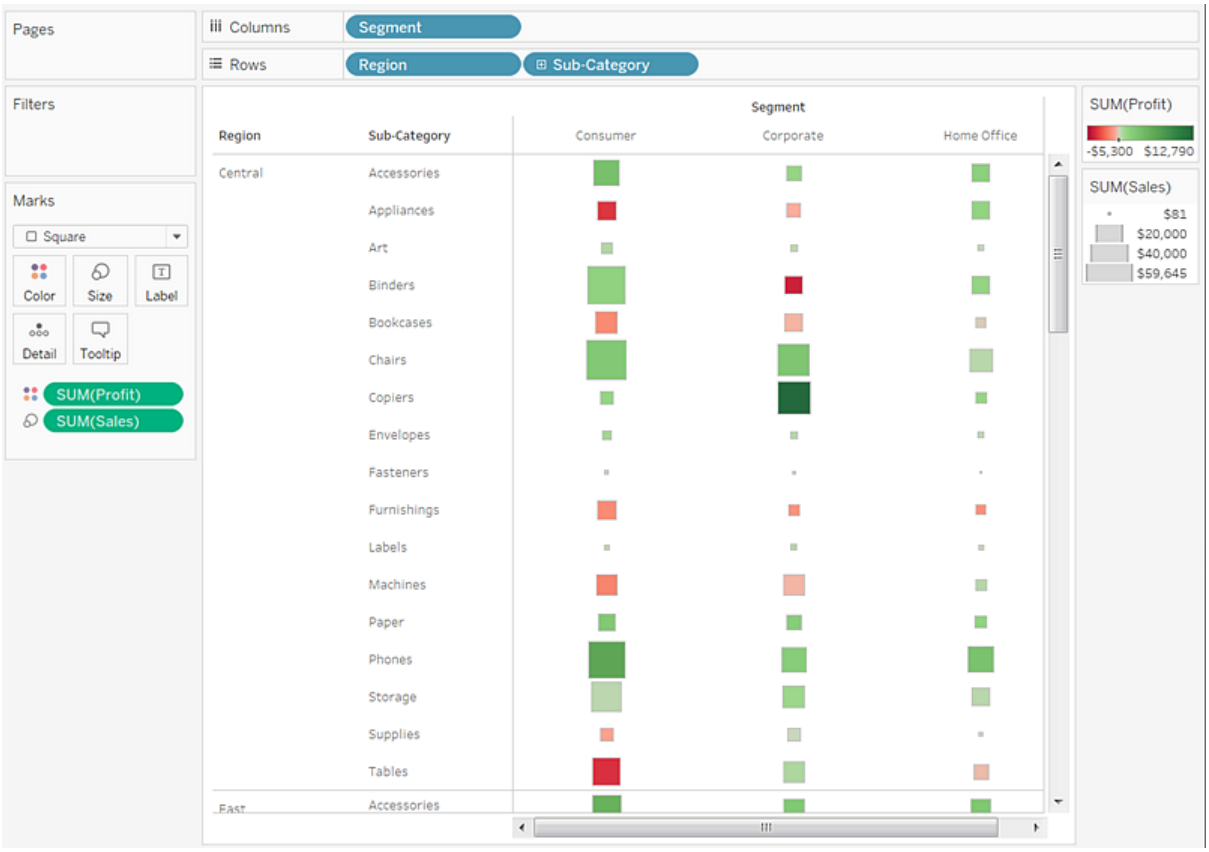
   Initially, the marks look like this:



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:


Pages
Columns
Segment
Rows
Region
Sub-Category
Filters
Marks
Square
Color
Size
Label
Detail
Tooltip
SUM(Profit)
SUM(Sales)
Segment
Region
Sub-Category
Consumer
Corporate
Home Office
Central
Accessories
Appliances
Art
Binders
Bookcases
Chairs
Copiers
Envelopes
Fasteners
Furnishings
Labels
Machines
Paper
Phones
Storage
Supplies
Tables
East
Accessories
SUM(Profit)
-$5,300
$12,790
SUM(Sales)
$81
$20,000
$40,000
$59,645

Practice 6: Create a story with Tableau

How to Create a Story in Tableau?

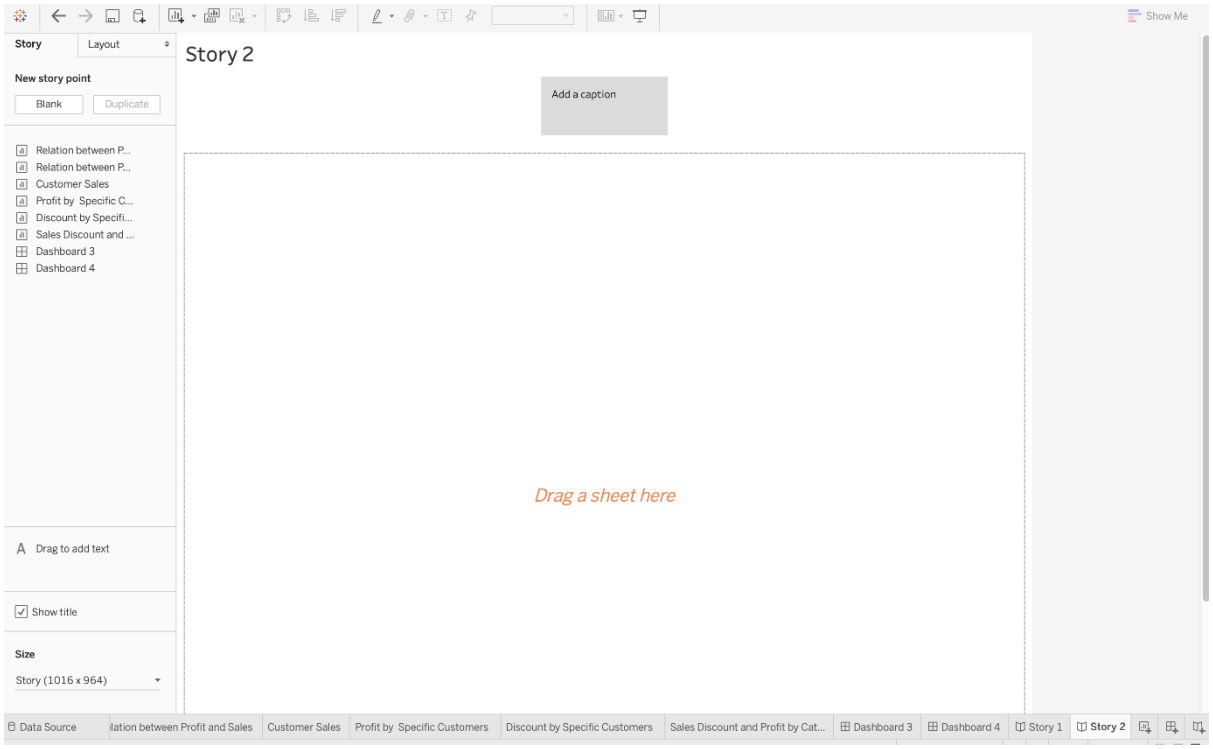
A story is a very powerful thing. It can influence its viewers and make them change their minds. It can take them through a journey that moves from various highs and lows and ends at a point that the storyteller wants. A story can also shock and surprise its viewers and provide them all the information they need to make the necessary decisions. That is why a story is an extremely important part of **Tableau**. **But what is a story in Tableau?**

Well, it is a sequence of different charts that combine to provide a cohesive plot to its viewers. In essence, all these charts tell a story about the data which allows the viewers to form their conclusion. The story in Tableau contains story points, where each story point is either a worksheet or a dashboard.

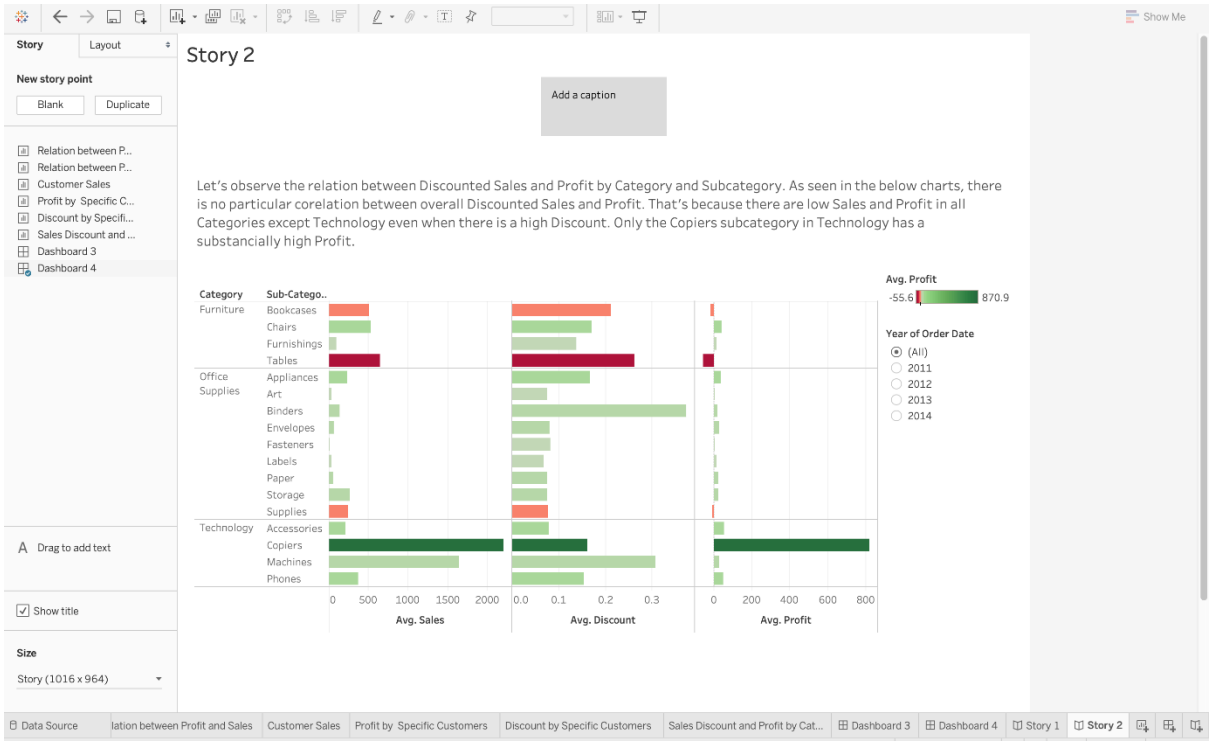
How to create a Story?

Let's see the various steps required to create a Story in Tableau. This story uses the Superstore data set that is available as a sample on Tableau Desktop.

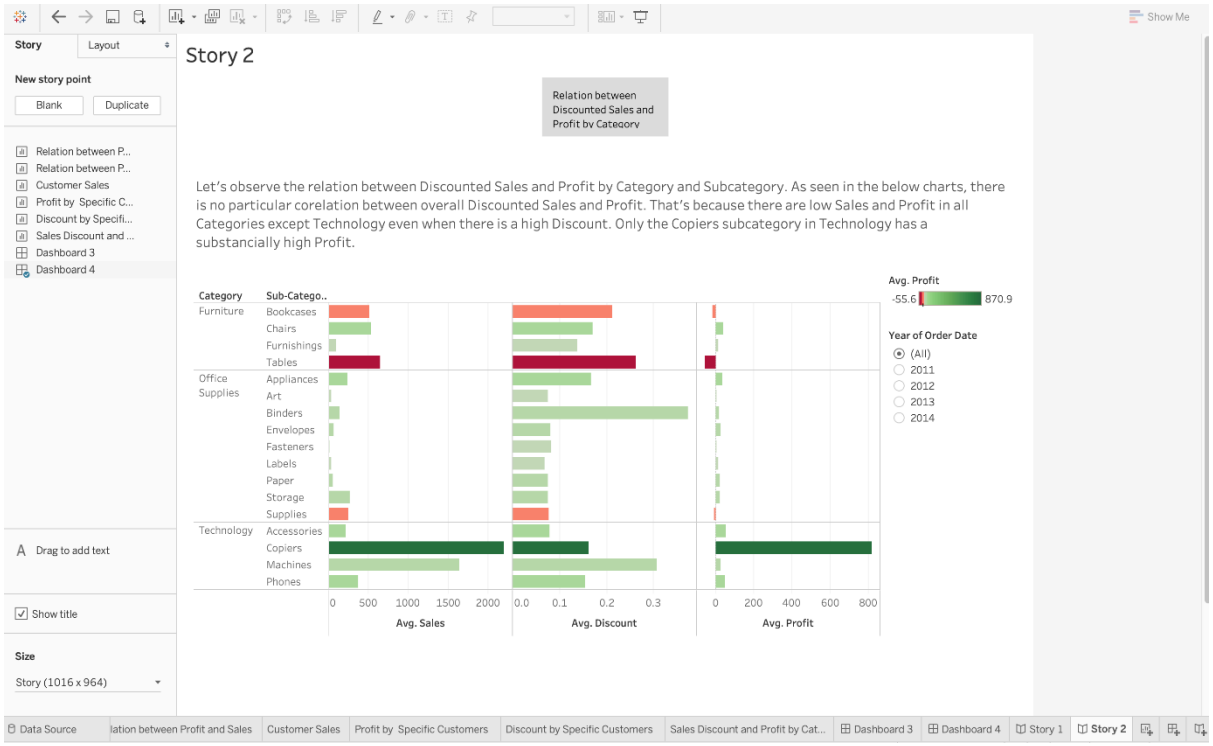
Step 1: Click on the new Story tab to create a new story. You can then add various sheets and dashboards to create a story point.



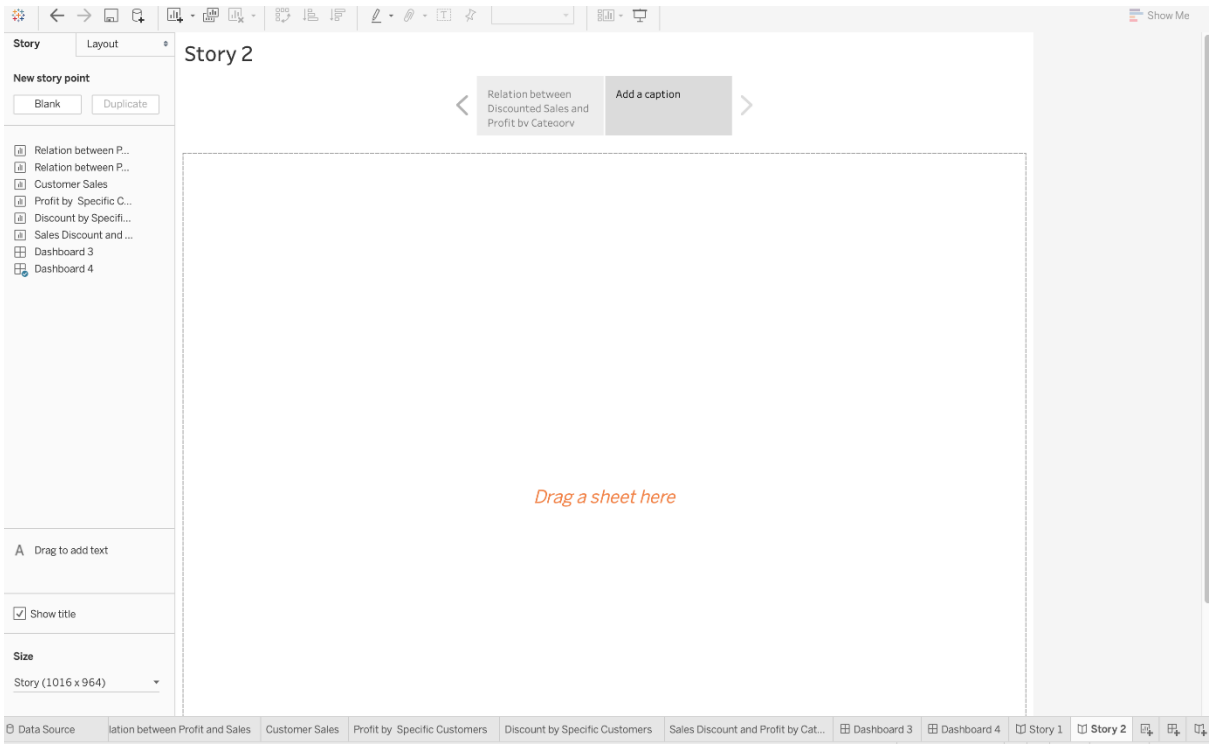
Step 2: You can double-click on the sheets and dashboards on the left to add them to a story point. You can also drag the sheets into your story point on the Tableau desktop. All the sheets and dashboards that are added to a story are connected to their original forms. So any changes made to the original sheets or dashboards are reflected in the story. For example, let's add a dashboard containing the relation between Discounted Sales and Profit by Category to the story.



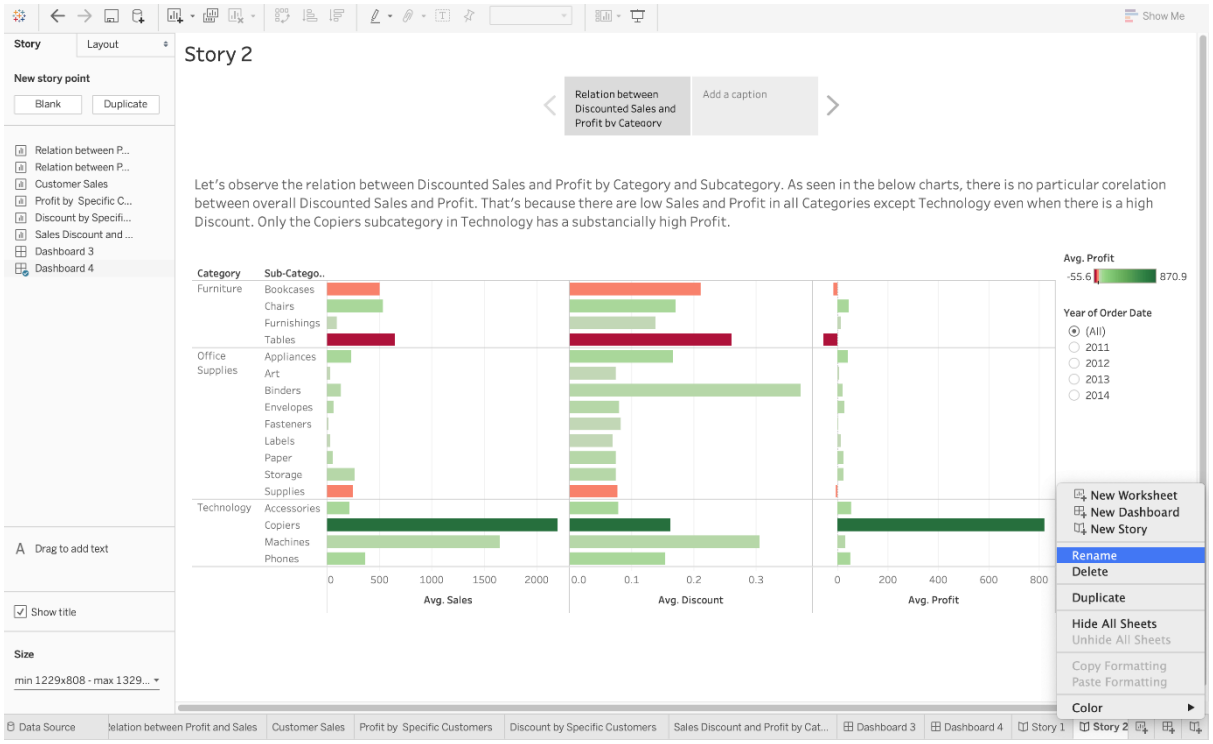
Step 3: We can also add a caption to summarize the story point by clicking on "Add a caption" and then writing it. Let's add the caption "Relation between Discounted Sales and Profit by Category and Subcategory" to our example.



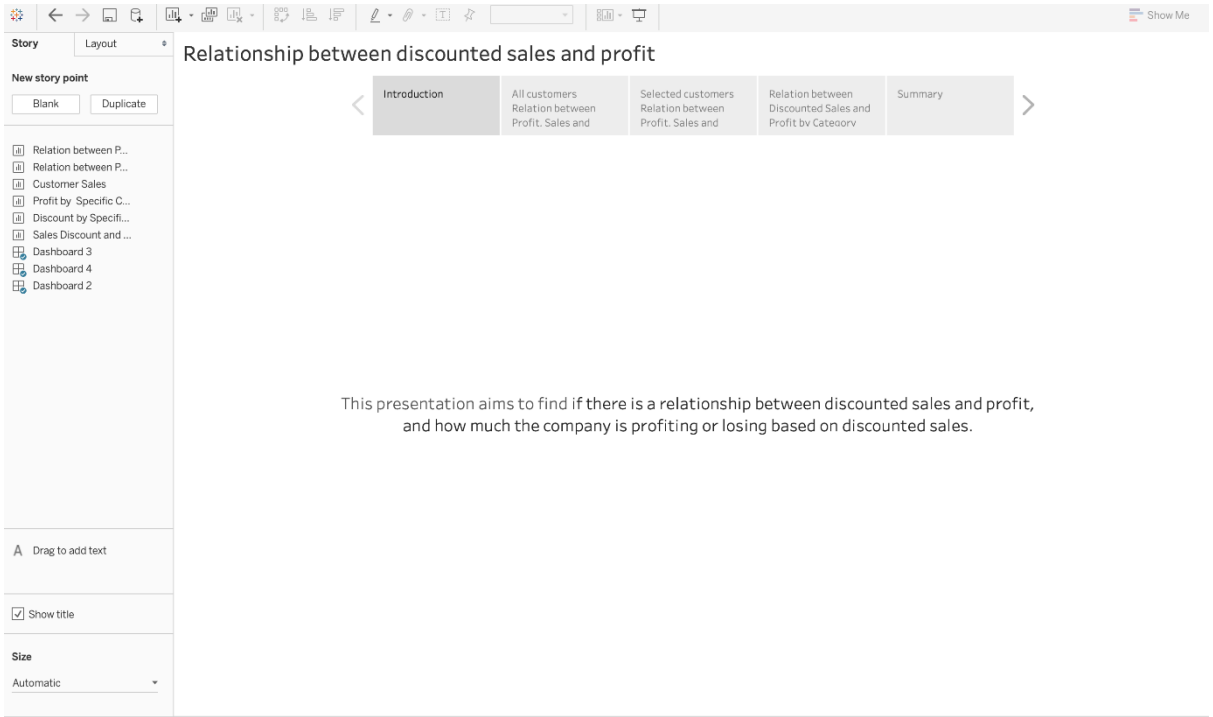
Step 4: It is possible to add another story point by 2 methods. You can either click on the Blank tab to use a blank sheet for the next story point or click on the Duplicate tab to obtain a duplicate sheet as the current story point. Let's click on the blank option.

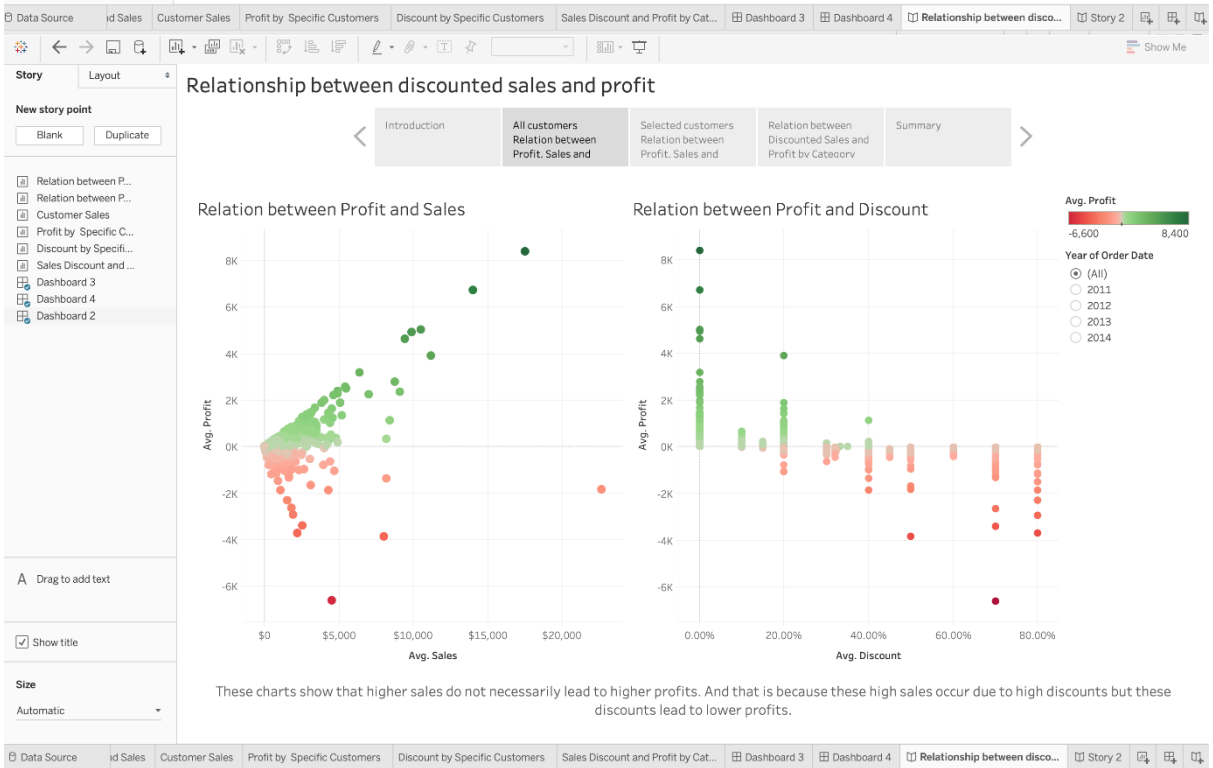


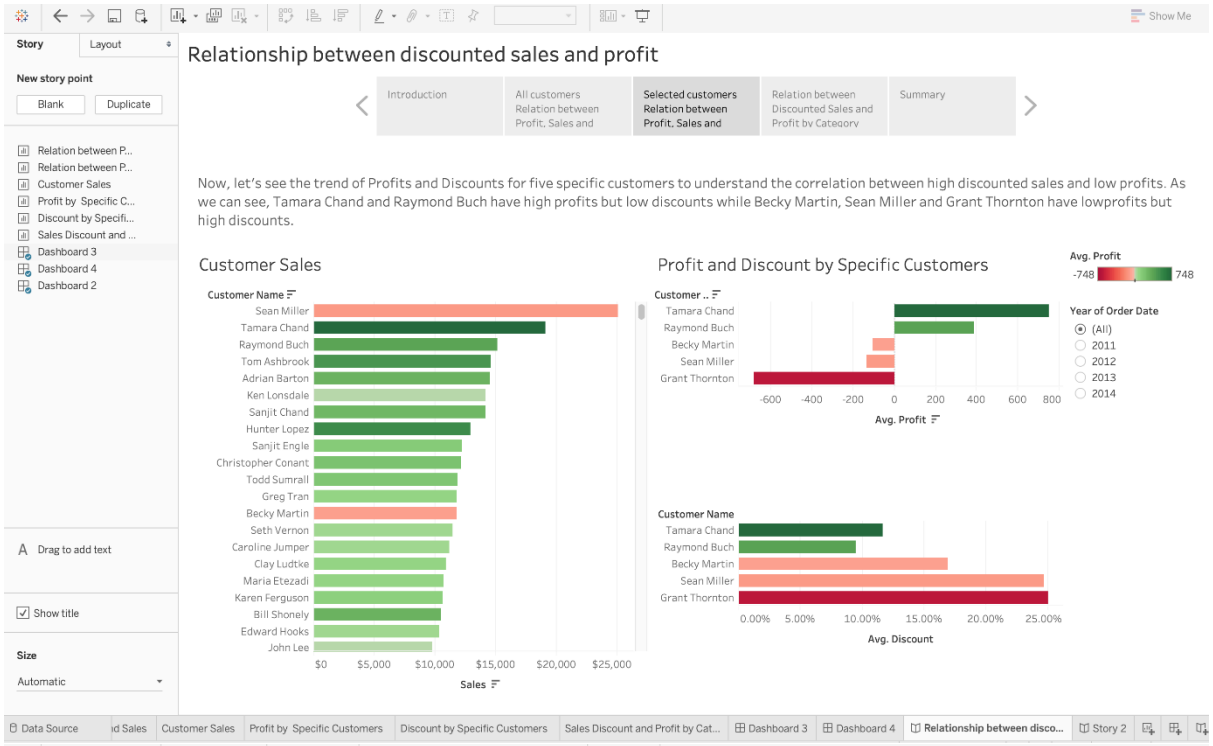
Step 5: You can change the size of your story by clicking on the Size option in the lower-left corner. You can choose from one of the predefined sizes or set your custom size in pixels. You can also change the name of your story by right-clicking on your Story tab and choosing rename.

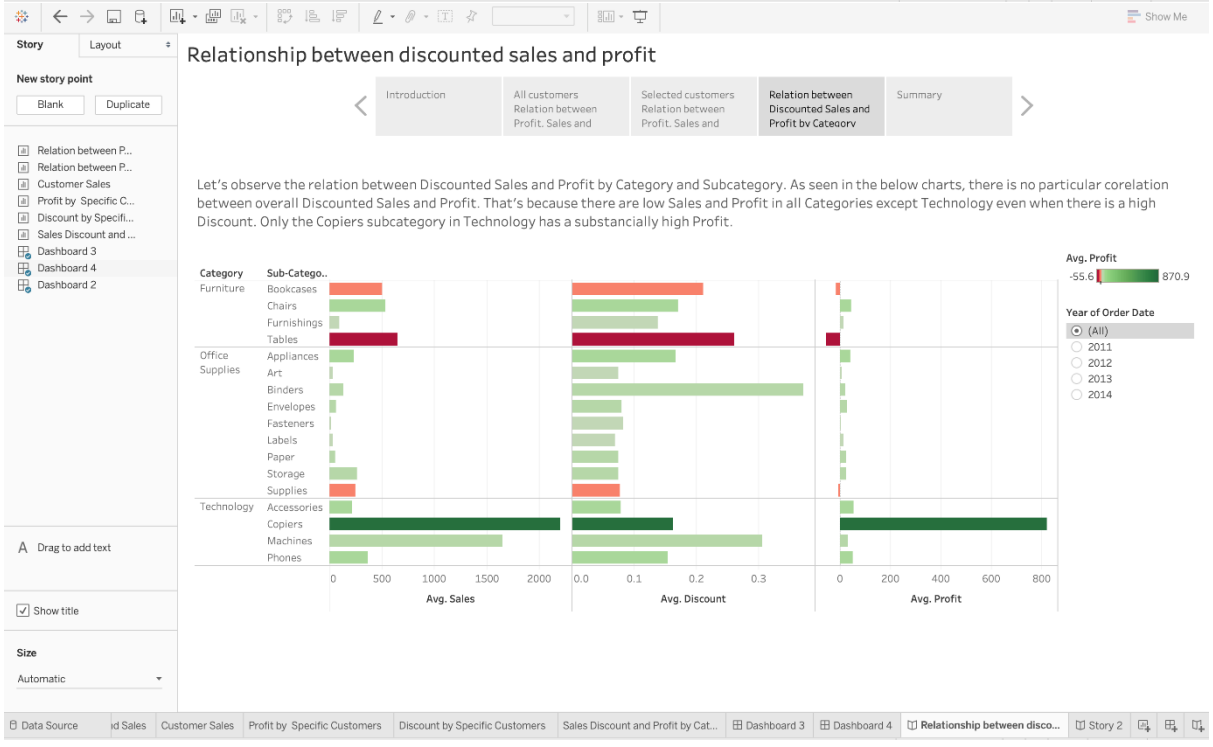


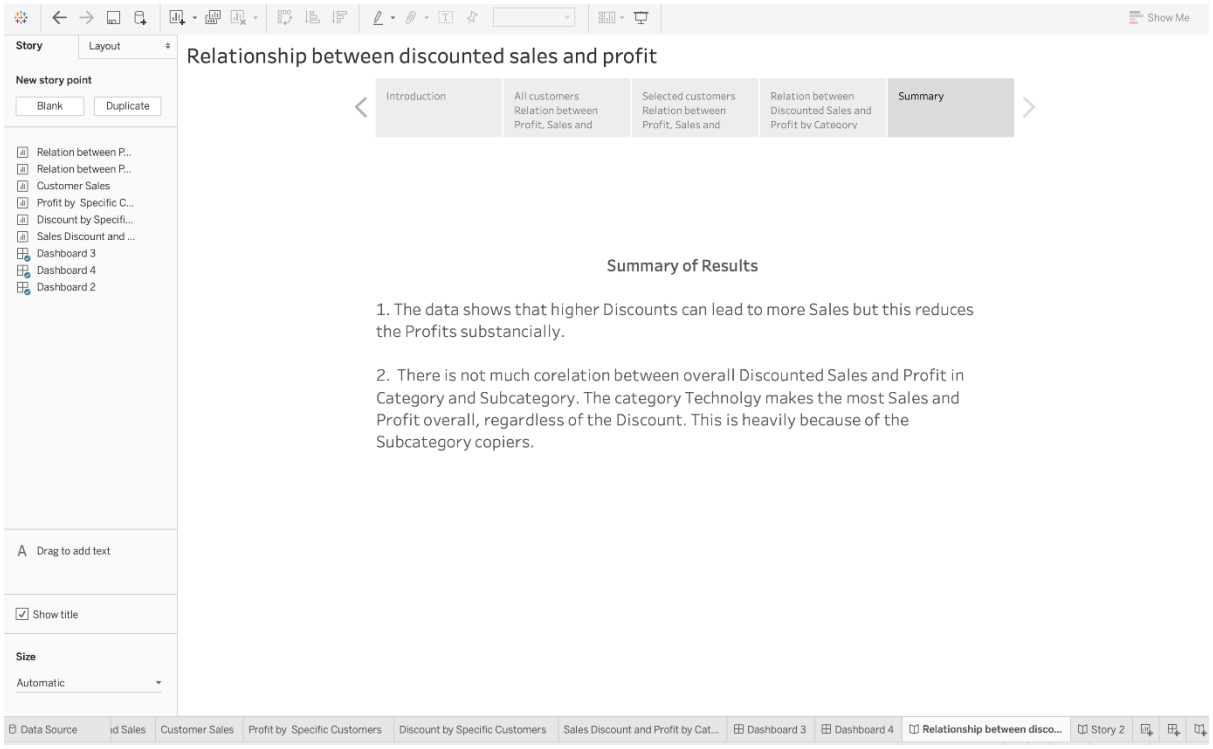
Step 6: Now, let's see a complete story on the relationship between the discounted sales and profit


Story
Layout
New story point
Blank
Duplicate
Relation between P...
Relation between P...
Customer Sales
Profit by Specific C...
Discount by Specifi...
Sales Discount and ...
Dashboard 3
Dashboard 4
Dashboard 2
Drag to add text
Show title
Size
Automatic
Show Me
Relationship between discounted sales and profit
Introduction
All customers Relation between Profit, Sales and
Selected customers Relation between Profit, Sales and
Relation between Discounted Sales and Profit by Category
Summary
This presentation aims to find if there is a relationship between discounted sales and profit, and how much the company is profiting or losing based on discounted sales.
Data Source
Sales
Customer Sales
Profit by Specific Customers
Discount by Specific Customers
Sales Discount and Profit by Cat...
Dashboard 3
Dashboard 4
Relationship between disco...
Story 2


Story
Layout
New story point
Blank
Duplicate
Relation between P...
Relation between P...
Customer Sales
Profit by Specific C...
Discount by Specifi...
Sales Discount and ...
Dashboard 3
Dashboard 4
Dashboard 2
Drag to add text
Show title
Size
Automatic
Show Me
Relationship between discounted sales and profit
Introduction
All customers Relation between Profit, Sales and
Selected customers Relation between Profit, Sales and
Relation between Discounted Sales and Profit by Category
Summary
Relation between Profit and Sales
Relation between Profit and Discount
Avg. Profit
-6,600
8,400
Year of Order Date
(All)
2011
2012
2013
2014
8K
6K
4K
2K
0K
-2K
-4K
-6K
Avg. Profit
$0
$5,000
$10,000
$15,000
$20,000
Avg. Sales
0.00%
20.00%
40.00%
60.00%
80.00%
Avg. Discount
These charts show that higher sales do not necessarily lead to higher profits. And that is because these high sales occur due to high discounts but these discounts lead to lower profits.
Data Source
Sales
Customer Sales
Profit by Specific Customers
Discount by Specific Customers
Sales Discount and Profit by Cat...
Dashboard 3
Dashboard 4
Relationship between disco...
Story 2


Story
Layout
New story point
Blank
Duplicate
Relation between P...
Relation between P...
Customer Sales
Profit by Specific C...
Discount by Specifi...
Sales Discount and ...
Dashboard 3
Dashboard 4
Dashboard 2
Drag to add text
Show title
Size
Automatic
Show Me
Relationship between discounted sales and profit
Introduction
All customers Relation between Profit, Sales and
Selected customers Relation between Profit, Sales and
Relation between Discounted Sales and Profit by Category
Summary
Now, let's see the trend of Profits and Discounts for five specific customers to understand the correlation between high discounted sales and low profits. As we can see, Tamara Chand and Raymond Buch have high profits but low discounts while Becky Martin, Sean Miller and Grant Thornton have lowprofits but high discounts.
Customer Sales
Customer Name
Sean Miller
Tamara Chand
Raymond Buch
Tom Ashbrook
Adrian Barton
Ken Lonsdale
Sanjit Chand
Hunter Lopez
Sanjit Engle
Christopher Conant
Todd Sumrall
Greg Tran
Becky Martin
Seth Vernon
Caroline Jumper
Clay Ludtke
Maria Etezadi
Karen Ferguson
Bill Shonely
Edward Hooks
John Lee
$0
$5,000
$10,000
$15,000
$20,000
$25,000
Sales
Profit and Discount by Specific Customers
Avg. Profit
-748
748
Customer ..
Tamara Chand
Raymond Buch
Becky Martin
Sean Miller
Grant Thornton
-600
-400
-200
0
200
400
600
800
Avg. Profit
Year of Order Date
(All)
2011
2012
2013
2014
Customer Name
Tamara Chand
Raymond Buch
Becky Martin
Sean Miller
Grant Thornton
0.00%
5.00%
10.00%
15.00%
20.00%
25.00%
Avg. Discount
Data Source
Customer Sales
Profit by Specific Customers
Discount by Specific Customers
Sales Discount and Profit by Cat...
Dashboard 3
Dashboard 4
Relationship between disco...
Story 2

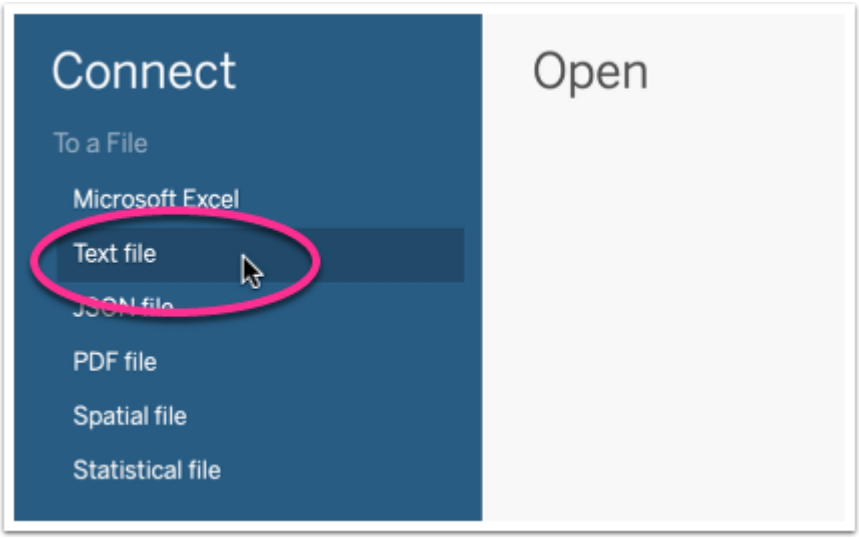

Story
Layout
New story point
Blank
Duplicate
Relation between P...
Relation between P...
Customer Sales
Profit by Specific C...
Discount by Specifi...
Sales Discount and ...
Dashboard 3
Dashboard 4
Dashboard 2
Drag to add text
Show title
Size
Automatic
Show Me
Relationship between discounted sales and profit
Introduction
All customers Relation between Profit, Sales and
Selected customers Relation between Profit, Sales and
Relation between Discounted Sales and Profit by Category
Summary
Let's observe the relation between Discounted Sales and Profit by Category and Subcategory. As seen in the below charts, there is no particular corelation between overall Discounted Sales and Profit. That's because there are low Sales and Profit in all Categories except Technology even when there is a high Discount. Only the Copiers subcategory in Technology has a substancially high Profit.
Category
Sub-Catego..
Furniture
Bookcases
Chairs
Furnishings
Tables
Office Supplies
Appliances
Art
Binders
Envelopes
Fasteners
Labels
Paper
Storage
Supplies
Technology
Accessories
Copiers
Machines
Phones
0
500
1000
1500
2000
Avg. Sales
0.0
0.1
0.2
0.3
Avg. Discount
0
200
400
600
800
Avg. Profit
Avg. Profit
-55.6
870.9
Year of Order Date
(All)
2011
2012
2013
2014
Data Source
Customer Sales
Profit by Specific Customers
Discount by Specific Customers
Sales Discount and Profit by Cat...
Dashboard 3
Dashboard 4
Relationship between disco...
Story 2


Show Me
Story
Layout
Relationship between discounted sales and profit
New story point
Blank
Duplicate
Relation between P...
Relation between P...
Customer Sales
Profit by Specific C...
Discount by Specifi...
Sales Discount and ...
Dashboard 3
Dashboard 4
Dashboard 2
Introduction
All customers Relation between Profit, Sales and
Selected customers Relation between Profit, Sales and
Relation between Discounted Sales and Profit by Category
Summary
Summary of Results
1. The data shows that higher Discounts can lead to more Sales but this reduces the Profits substancially.
2. There is not much corelation between overall Discounted Sales and Profit in Category and Subcategory. The category Technolgy makes the most Sales and Profit overall, regardless of the Discount. This is heavily because of the Subcategory copiers.
A Drag to add text
Show title
Size
Automatic
Data Source
Customer Sales
Profit by Specific Customers
Discount by Specific Customers
Sales Discount and Profit by Cat...
Dashboard 3
Dashboard 4
Relationship between disco...
Story 2

Practice 7: Create a new dashboard

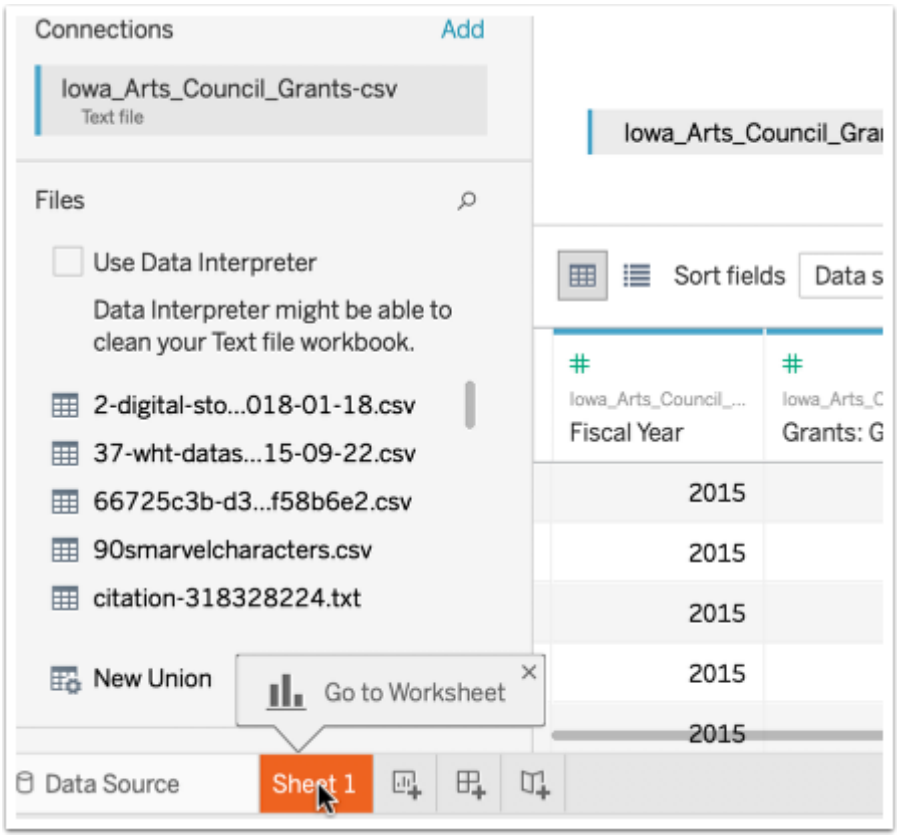
1. Choose your data source

After opening Tableau, you're presented with a list of file types you can choose to work with ("connect"). Even though our Iowa Arts Council Grants file *opens* in Excel, it's saved as a CSV. To Tableau, a CSV is a text file. So select **Text file.** Then navigate to the file you downloaded earlier and double-click to open it.



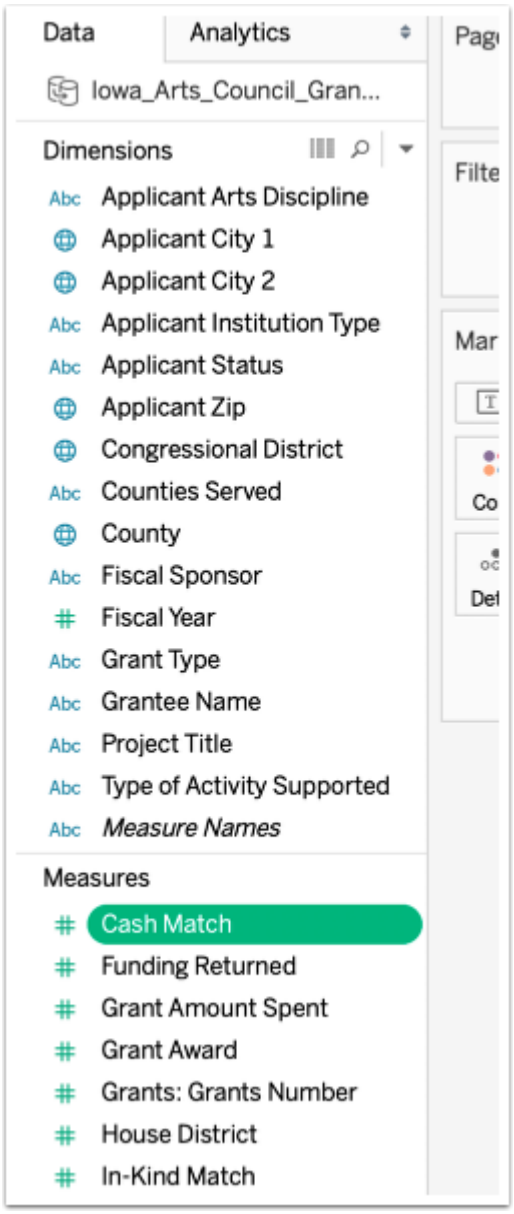
2. Create a sheet

Tableau will open your file. It should look pretty familiar! In Tableau, you begin visualizing data by creating a **Sheet**. Do that by clicking on the orange **Sheet** button in the lower left-hand corner.



3. Data types in Tableau

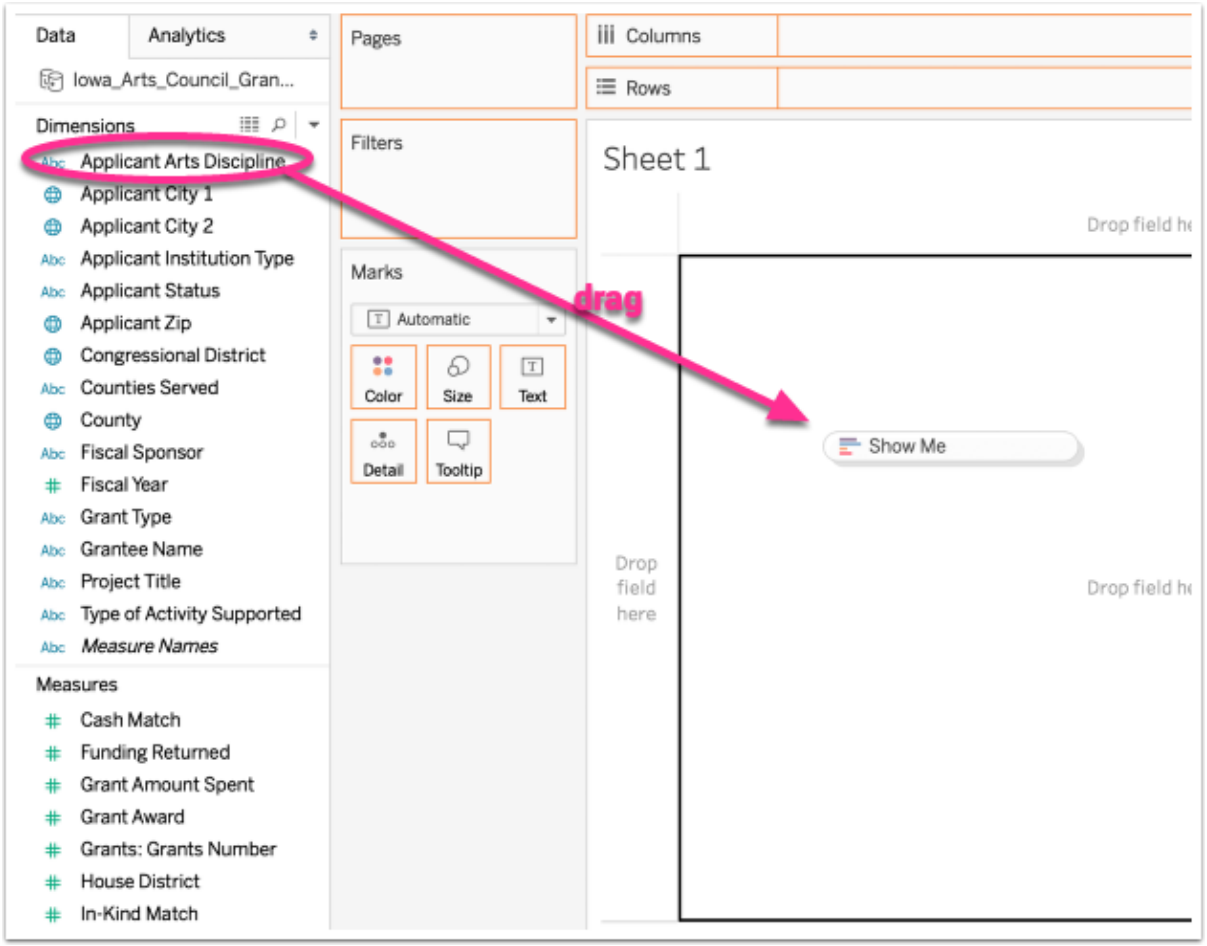
Tableau divides your content types (that is, your columns) into **dimensions** and **measures**. Measures consist of numeric information: values that can be added together. Everything else is a dimension. Tableau will often provide recommendations based on these data types.



4. Your first visualization

Get started by clicking on **Applicant Arts Discipline** and drag it into the main section of the sheet (the **canvas**). It's not hugely exciting; you just see a list of arts disciplines.

There's a reason for that: Tableau doesn't know what you want it to count.

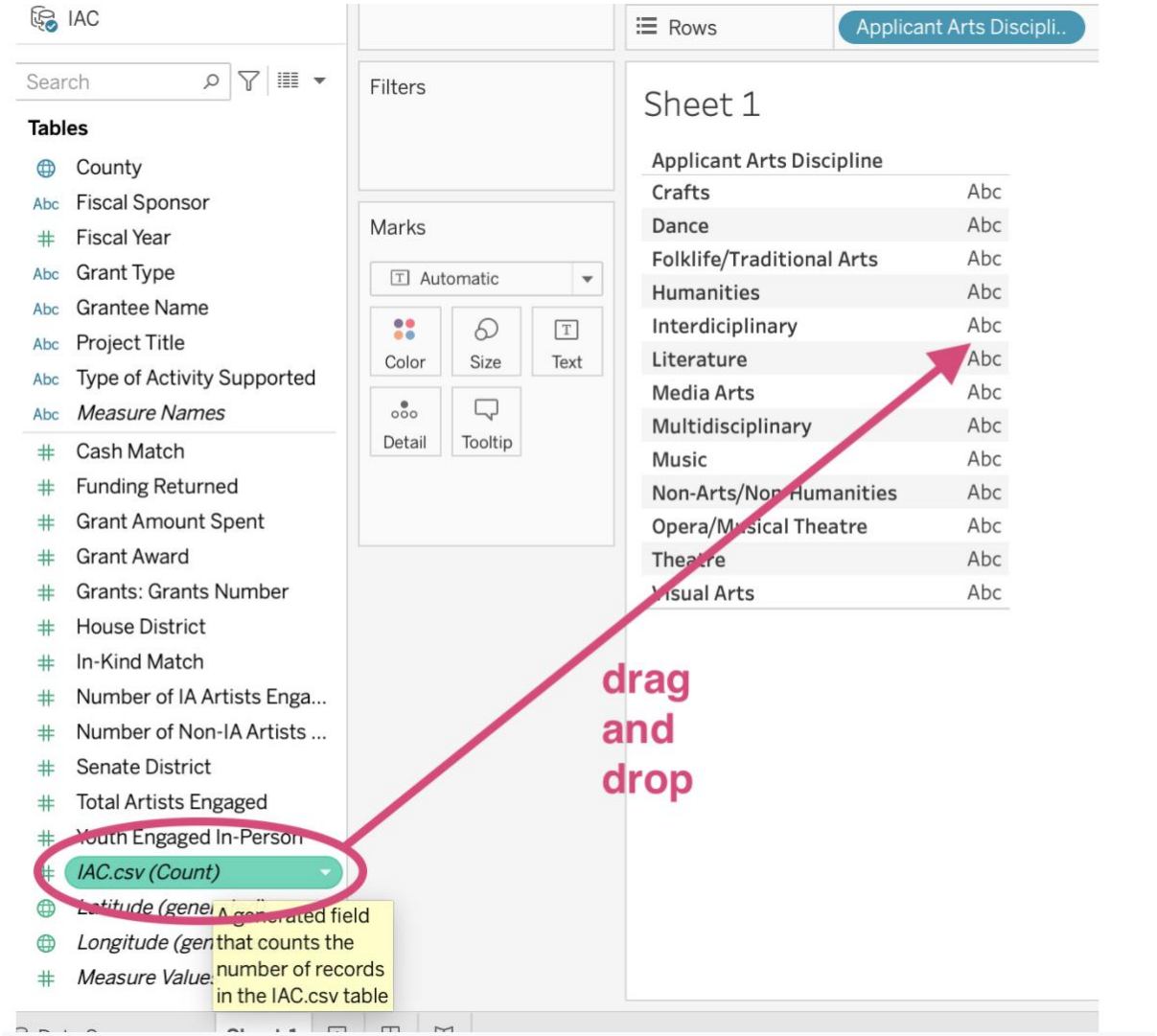


5. Tell Tableau which measure to use

We want Tableau to create a chart that visualizes the number of grants awarded per arts discipline. In order to do that, we need Tableau to count up the values for each category.

Scroll to the bottom of the **Data** column, and look at the measure types that are in italics. You'll see that they contain the word **generated** next to them in parentheses. This means that these are numbers that Tableau has calculated for you.

You'll notice a measure called **IAC.csv (Count)**. This measure provides a count of all grants. Click on this measure and drag it to the table on your canvas. Drop it in the second column of the table, where the values are currently represented as "Abc."



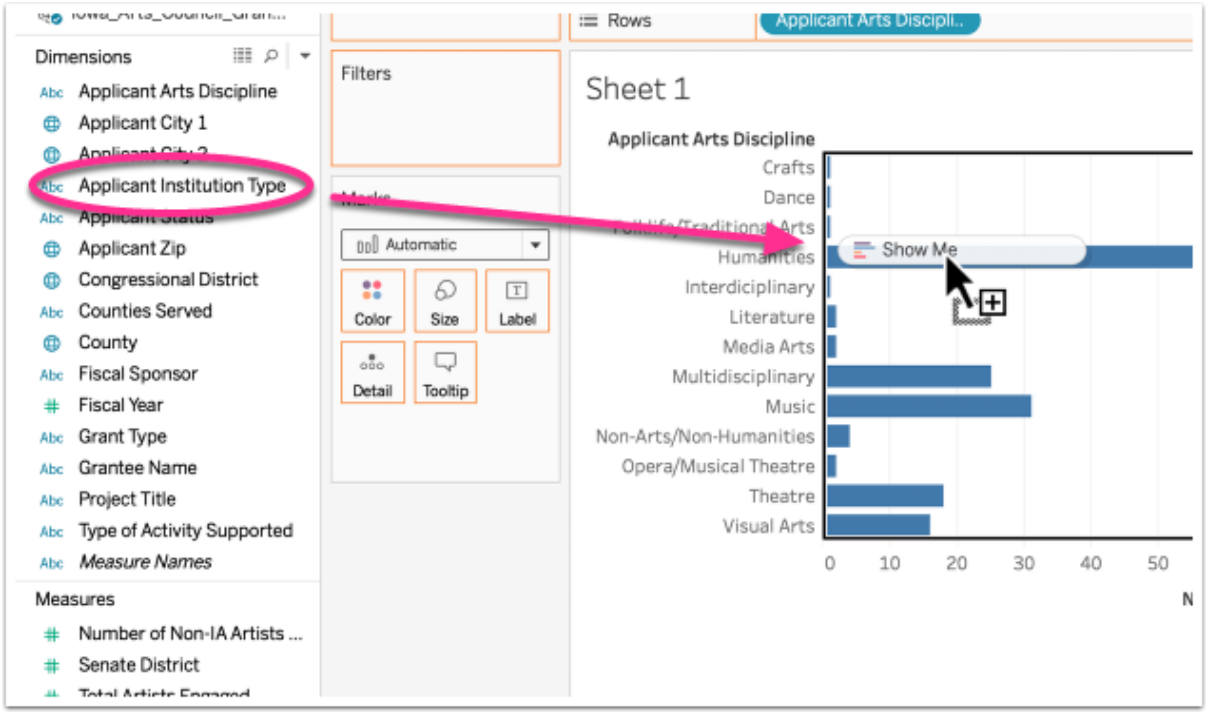
6. Choose the chart type you want

Once you've dropped the "Number of Records" measure, you'll see that they're nicely summarized for you in the table you created. You'll also notice that highlighted options appear in the palette of chart types on the right-hand side of your window. Now that you have measures, you have some chart options! Click on the bar chart.

7. Compare multiple values

You created a visualization! Now, let's see if we can create a stacked bar chart, the way we did with Excel. We'll show how **Application Institution Type** correlates with **Application Arts Discipline**.

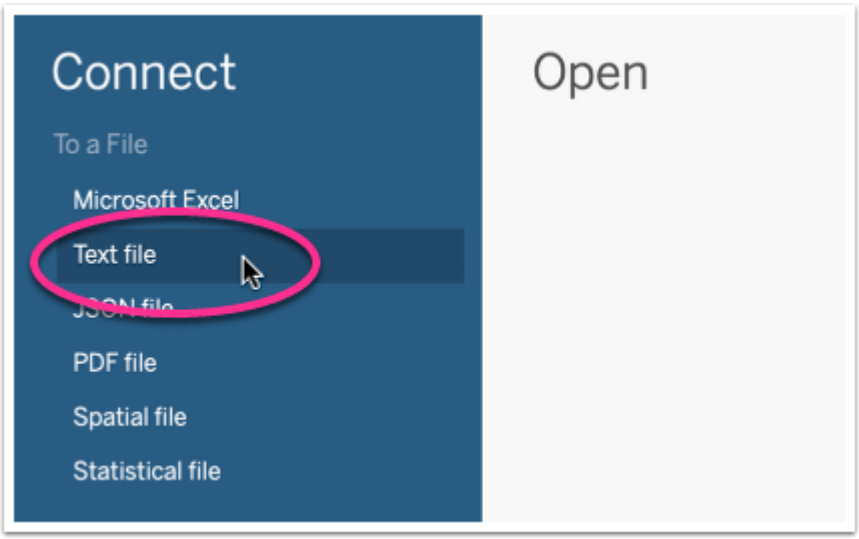
Luckily, this is easy. Just drag the Application Institution Type measure onto the bar chart you've already created.


Rows
Applicant Arts Discipli..
Dimensions
Applicant Arts Discipline
Applicant City 1
Applicant Institution Type
Applicant Zip
Congressional District
Counties Served
County
Fiscal Sponsor
Fiscal Year
Grant Type
Grantee Name
Project Title
Type of Activity Supported
Measure Names
Measures
Number of Non-IA Artists ...
Senate District
Filters
Automatic
Color
Size
Label
Detail
Tooltip
Sheet 1
Applicant Arts Discipline
Crafts
Dance
Humanities
Interdiciplinary
Literature
Media Arts
Multidisciplinary
Music
Non-Arts/Non-Humanities
Opera/Musical Theatre
Theatre
Visual Arts
Show Me
0
10
20
30
40
50
N

Practice 8: Working with dashboard

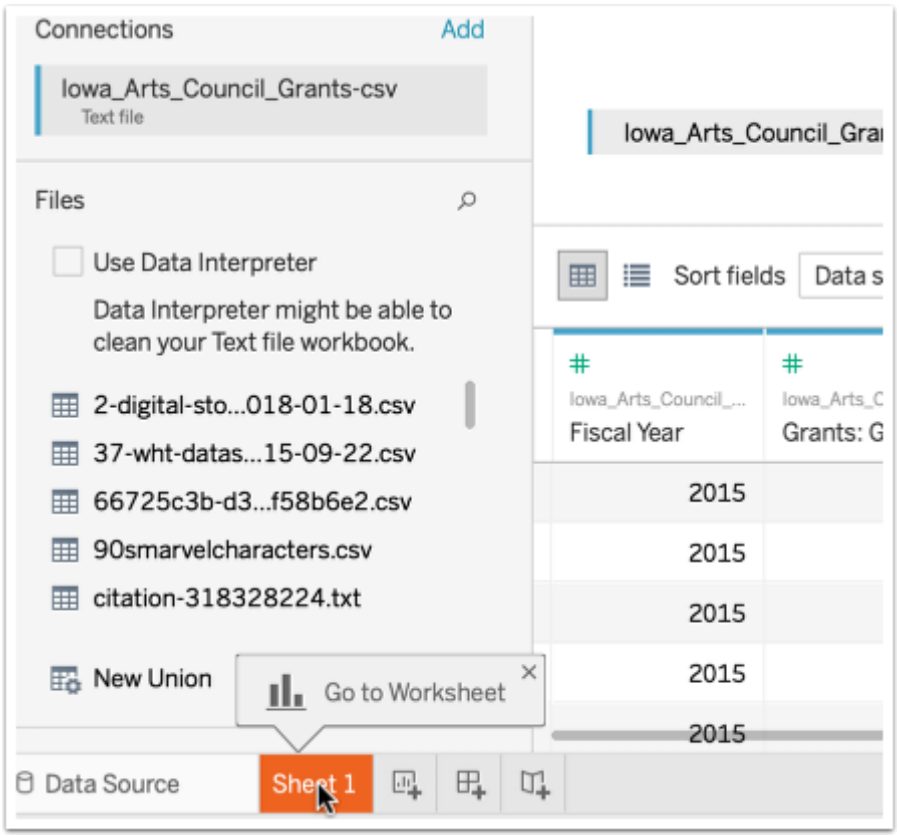
1. Choose your data source

After opening Tableau, you're presented with a list of file types you can choose to work with ("connect"). Even though our Iowa Arts Council Grants file *opens* in Excel, it's saved as a CSV. To Tableau, a CSV is a text file. So select **Text file.** Then navigate to the file you downloaded earlier and double-click to open it.



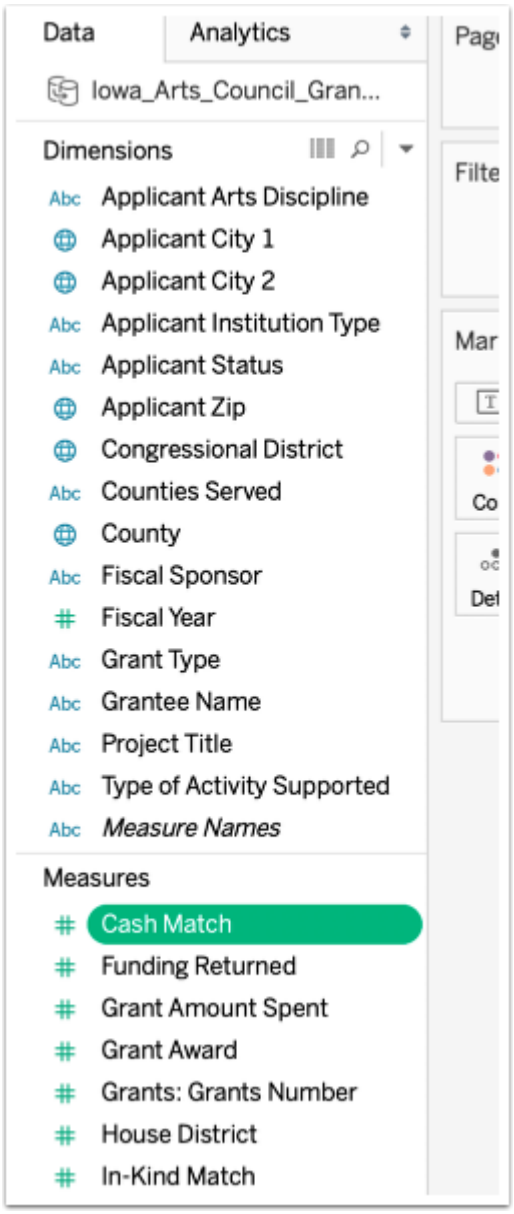
2. Create a sheet

Tableau will open your file. It should look pretty familiar! In Tableau, you begin visualizing data by creating a **Sheet**. Do that by clicking on the orange **Sheet** button in the lower left-hand corner.



3. Data types in Tableau

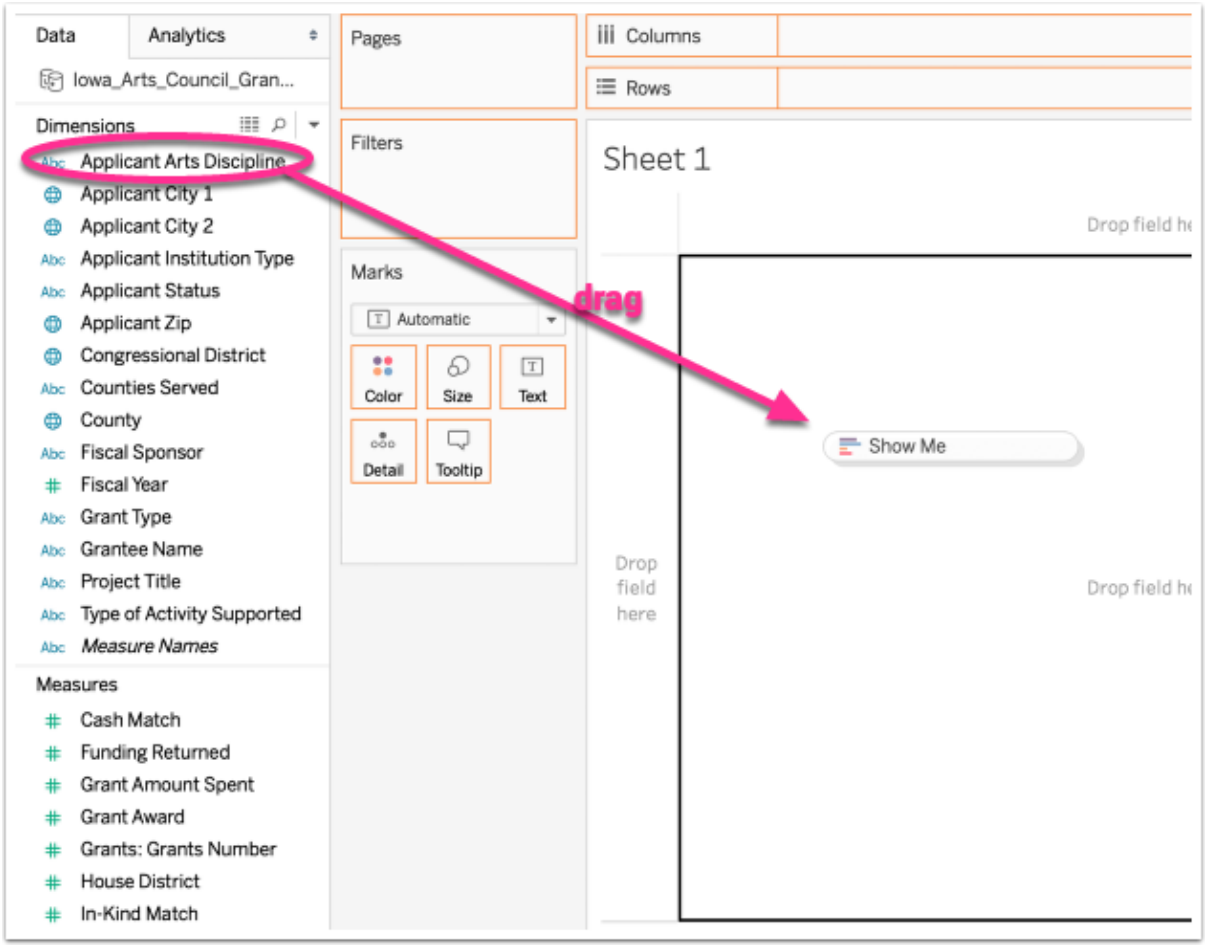
Tableau divides your content types (that is, your columns) into **dimensions** and **measures**. Measures consist of numeric information: values that can be added together. Everything else is a dimension. Tableau will often provide recommendations based on these data types.



4. Your first visualization

Get started by clicking on **Applicant Arts Discipline** and drag it into the main section of the sheet (the **canvas**). It's not hugely exciting; you just see a list of arts disciplines.

There's a reason for that: Tableau doesn't know what you want it to count.

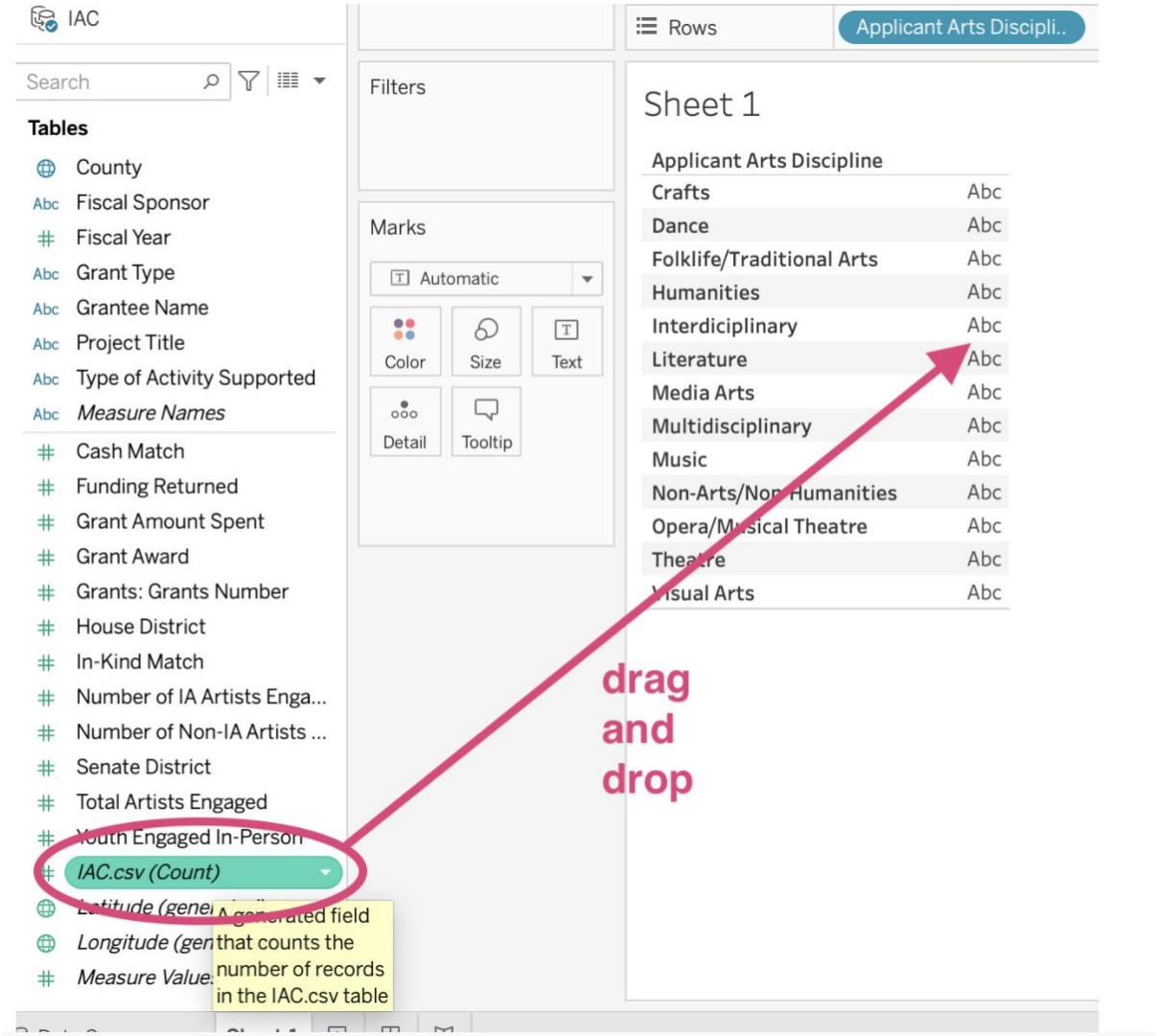


5. Tell Tableau which measure to use

We want Tableau to create a chart that visualizes the number of grants awarded per arts discipline. In order to do that, we need Tableau to count up the values for each category.

Scroll to the bottom of the **Data** column, and look at the measure types that are in italics. You'll see that they contain the word **generated** next to them in parentheses. This means that these are numbers that Tableau has calculated for you.

You'll notice a measure called **IAC.csv (Count)**. This measure provides a count of all grants. Click on this measure and drag it to the table on your canvas. Drop it in the second column of the table, where the values are currently represented as "Abc."



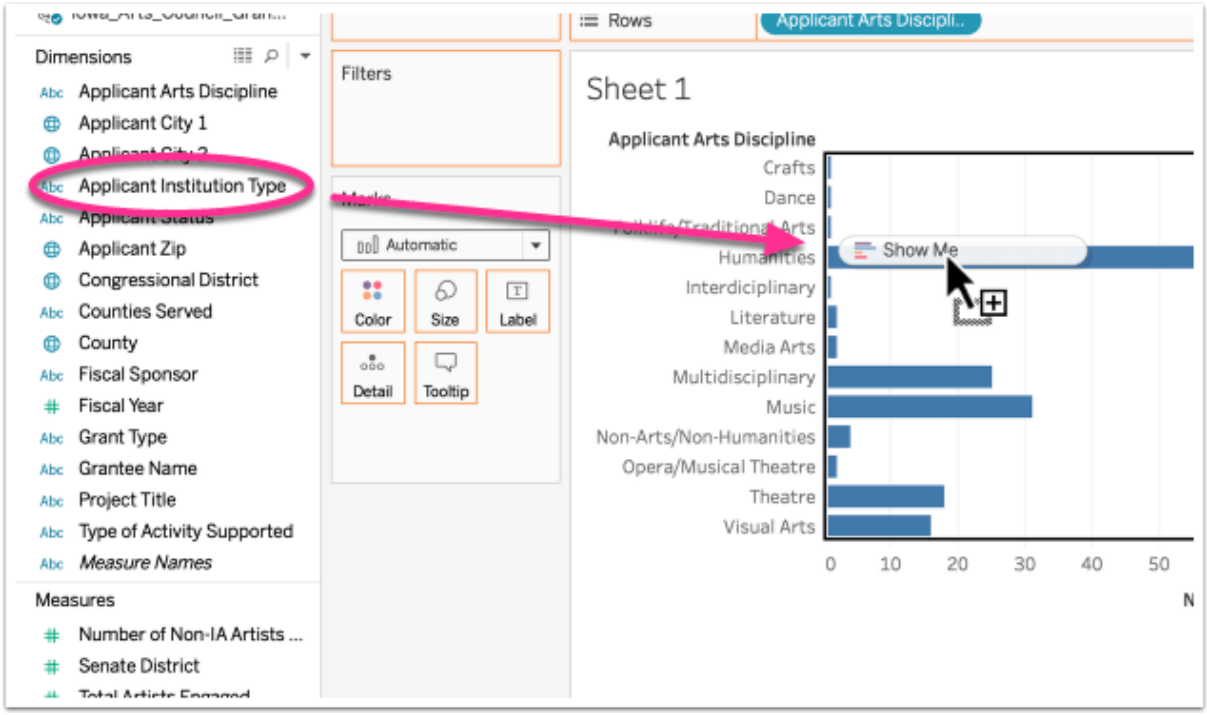
6. Choose the chart type you want

Once you've dropped the "Number of Records" measure, you'll see that they're nicely summarized for you in the table you created. You'll also notice that highlighted options appear in the palette of chart types on the right-hand side of your window. Now that you have measures, you have some chart options! Click on the bar chart.

7. Compare multiple values

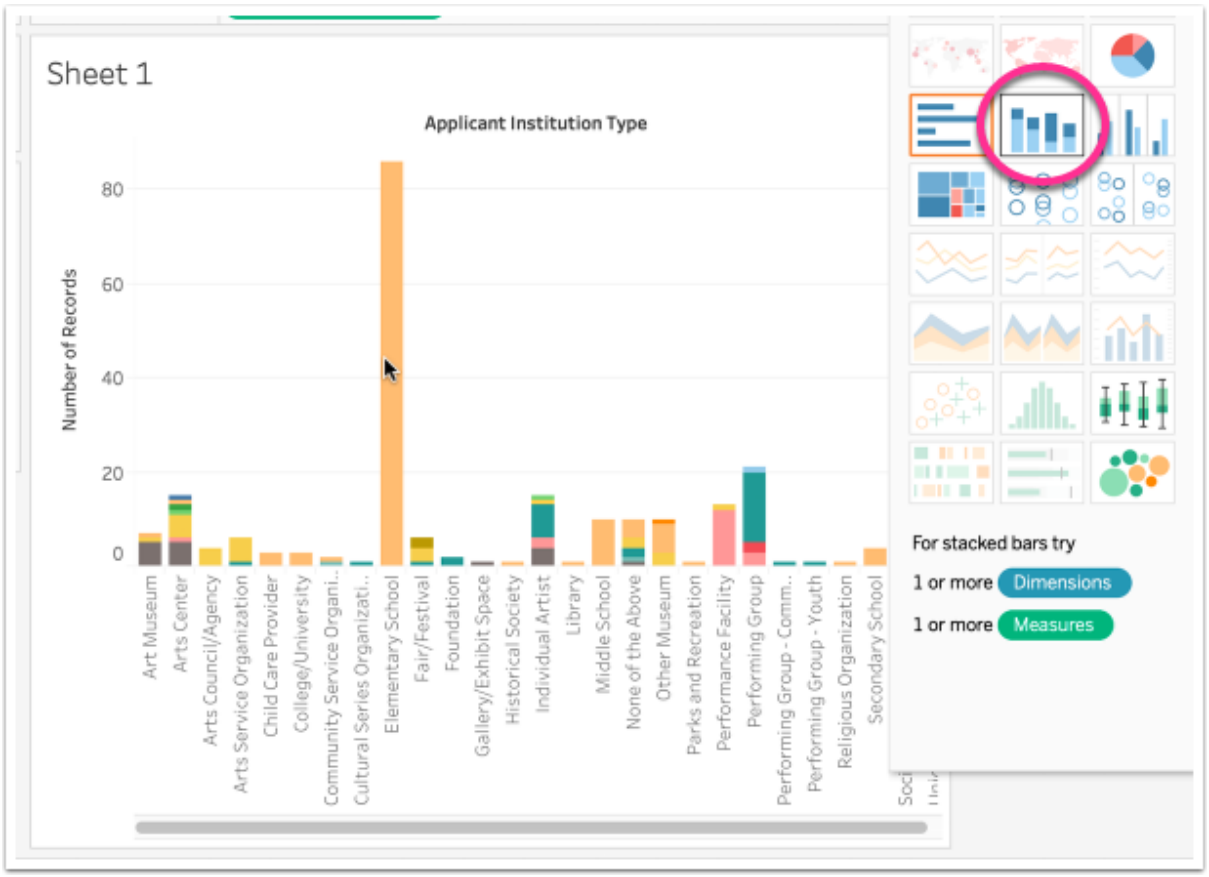
You created a visualization! Now, let's see if we can create a stacked bar chart, the way we did with Excel. We'll show how **Application Institution Type** correlates with **Application Arts Discipline**.

Luckily, this is easy. Just drag the Application Institution Type measure onto the bar chart you've already created.



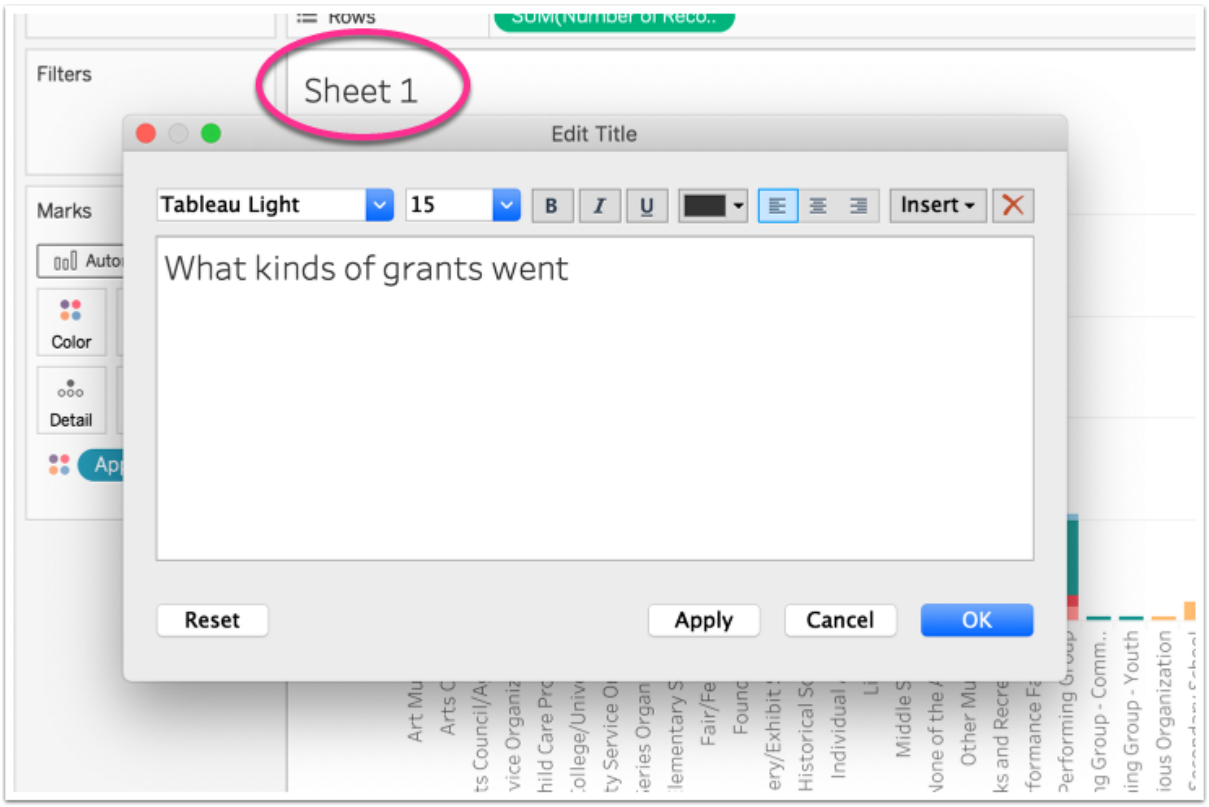
8. Create a stacked bar chart

Now, let's switch to a stacked bar chart, so we can see the distinctions among institution types more clearly. You'll notice that as you hover over each segment, a tooltip gives you more information.



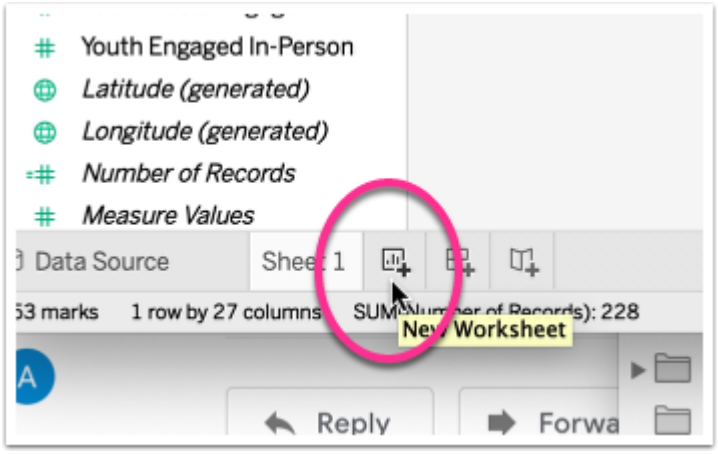
9. Give your chart a name

Click on your chart's title (it currently reads **Sheet 1**) to give it a more descriptive name.



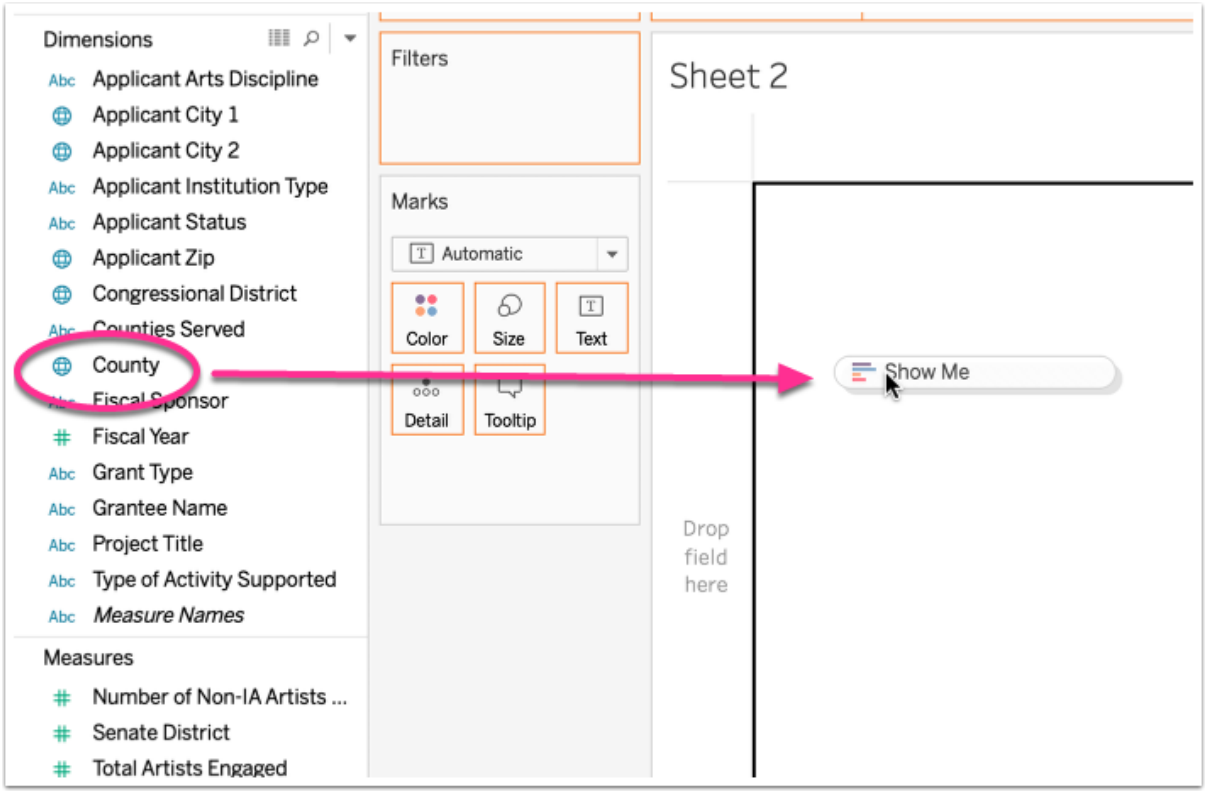
10. Start a new chart

Now let's make our second chart. Click on the **New worksheet** button (circled below) to begin our new visualization.



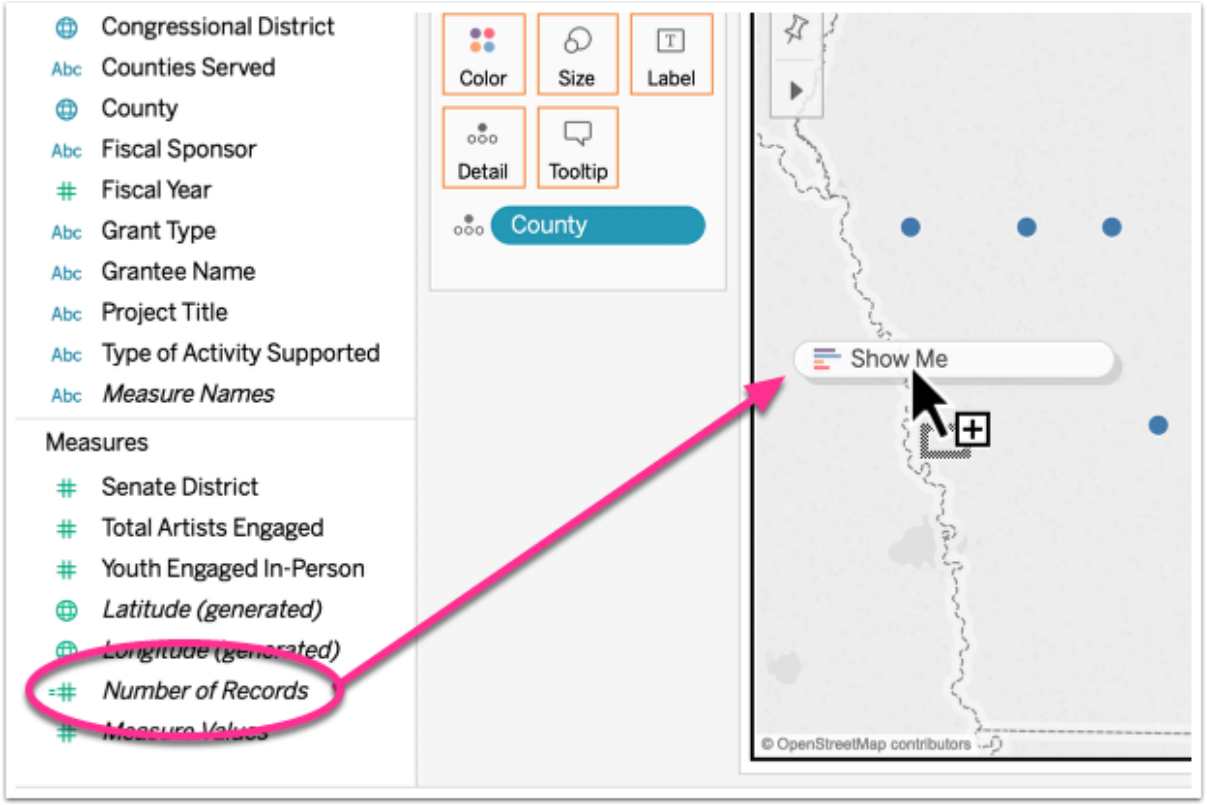
11. Make a map

Tableau has automatically geocoded our geographic dimensions. You can tell because a tiny globe appears next to them. Drag **County** into the main canvas area, and give Tableau a moment to work.



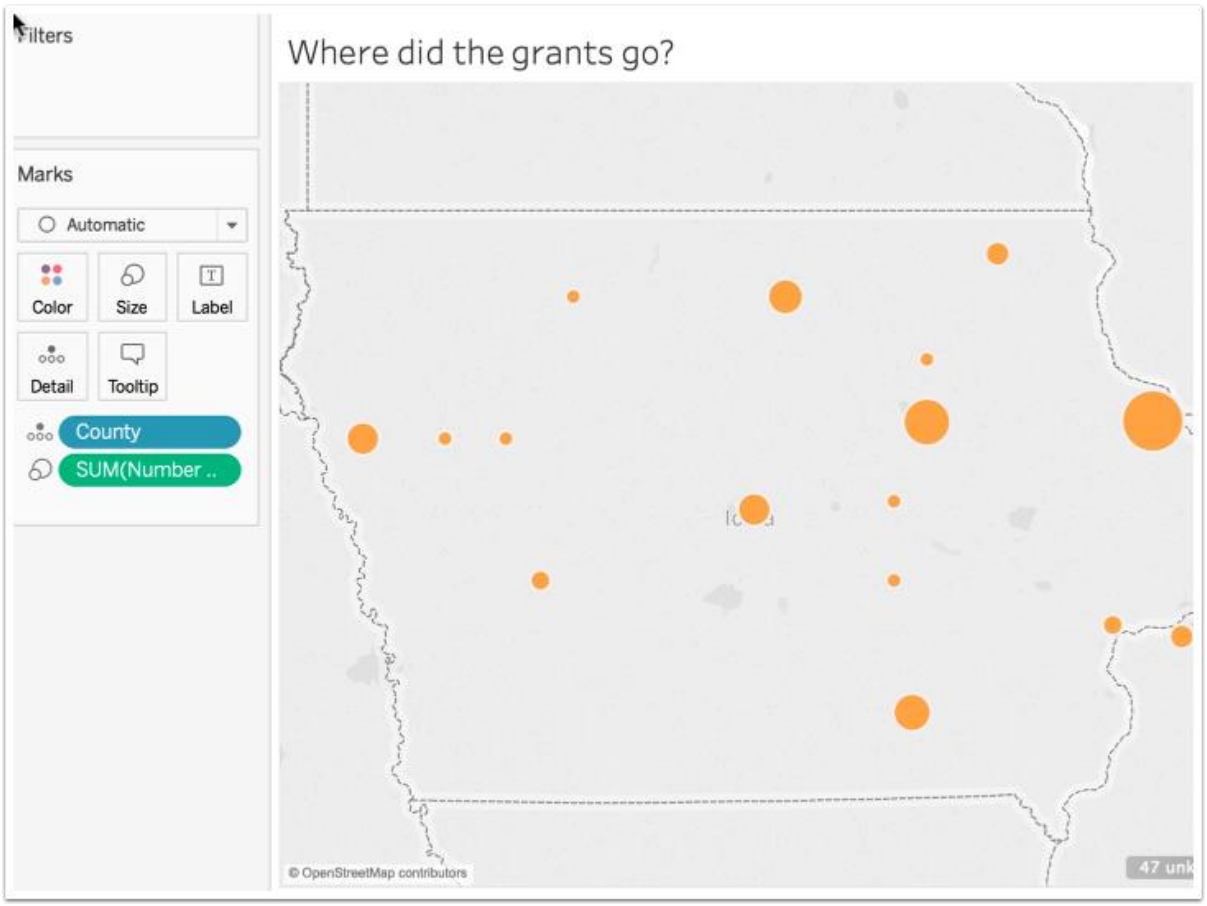
12. You have a map!

Now that you've done a map, let's add a measure to it. Drag **Number of Records** into the main canvas area, on top of your map.



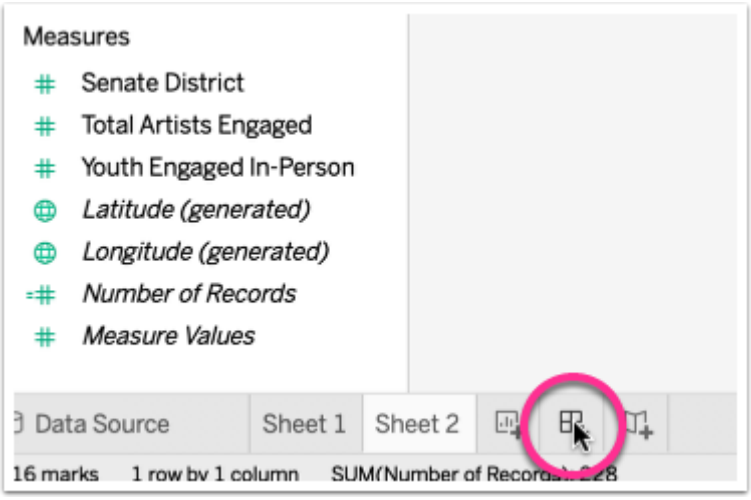
13. Finish your map

Now the circles on your map grow larger as the number of grants awarded to that county increases. You can fine-tune the look of your map by altering the options in the **Marks window**. Give your new map a title, as you did for your chart.



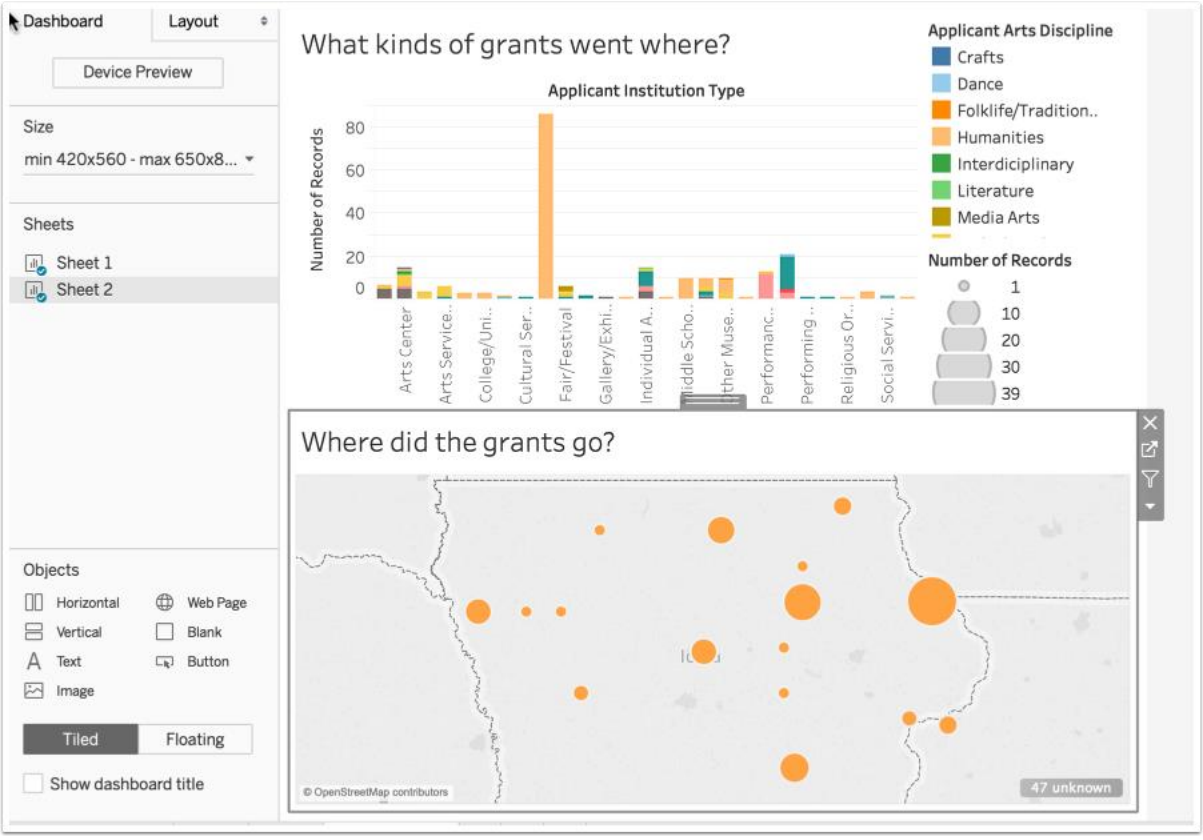
14. Create a dashboard

Now we'll combine our charts to create a **dashboard** -- a snapshot of multiple visualizations. Do that by clicking on the **Dashboard** button, circled below.



15. Drag your sheets onto your dashboard

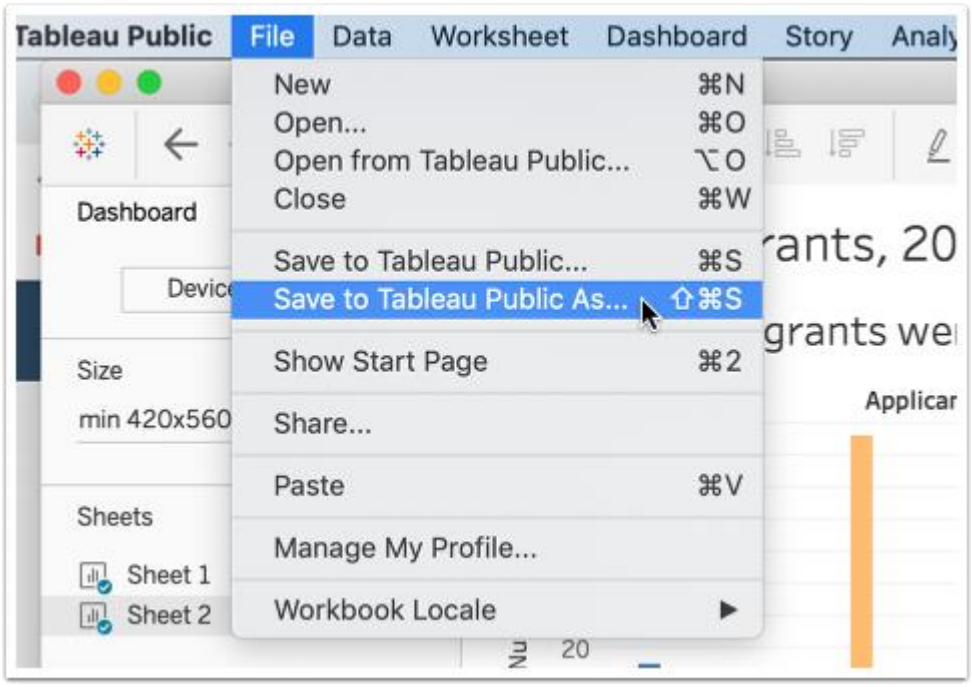
From the left-hand side of your dashboard's window, click on each of your sheets in turn and drag them into your main canvas. Very nice!



16. Options for exporting

As I've mentioned, in order to make your visualization web-accessible, you will need to create a Tableau account and publish to Tableau's site. From there, you can embed your visualizations in other web pages. To begin this process, select **Save to Tableau Public**

As... from the **File** menu. You will be prompted to begin the process of creating an account, logging in, and publishing to the web.



Practice 9: Build a real time dashboard

Section 1: Understanding Real-Time Concepts in Tableau

True "real-time" (sub-second updates) is often difficult to achieve without specialized streaming data connectors. For most business intelligence purposes, "near real-time" is sufficient, meaning data refreshes every few seconds, minutes, or on a scheduled basis.

Key enablers for near real-time dashboards in Tableau:

1. Live Connections: Connecting directly to your data source without extracting the data. Tableau will query the data source each time the viz is loaded or refreshed.
2. Automatic Refresh: Configuring Tableau Desktop or Tableau Server/Cloud to automatically refresh live connections at defined intervals.
3. Efficient Data Sources: Your underlying data source needs to be performant enough to handle frequent queries from Tableau.

Section 2: Preparing Your Data Source for Real-Time Updates

The efficiency of your real-time dashboard heavily depends on your data source.

- Indexed Tables: Ensure your database tables are properly indexed, especially on columns used for filtering, joining, or sorting in Tableau.
- Optimized Queries: Tableau will generate queries. Make sure these queries run quickly on your database. You might consider creating views in your database to pre-aggregate data if appropriate.
- Timestamp Columns: Include a timestamp column in your data that indicates when a record was last updated or created. This is crucial for incremental refreshing or for showing data "as of" a certain time.
- Direct Database Access: For best performance, ensure Tableau Desktop/Server has direct, low-latency access to your database.

Section 3: Connecting to Your Data in Tableau Desktop

Step 1: Open Tableau Desktop and Connect to Data

1. Open Tableau Desktop.

2. On the Connect pane, under "To a Server," select your data source type (e.g., PostgreSQL, Microsoft SQL Server, Google BigQuery). For this example, let's assume a generic SQL database.
    - (Self-reflection: At this point, I would instruct the user to capture a screenshot of the "Connect to Data" pane with various server options displayed.)

Step 2: Enter Connection Details

1. Input the necessary connection details: server name, port, database, authentication credentials (username and password).
2. Click Sign In or Connect.
    - (Self-reflection: A screenshot of the database connection dialog box would be helpful here, with input fields for server, port, user, etc.)

Step 3: Choose Your Database and Tables

1. Once connected, select the database you want to work with.
2. Drag the tables you need from the left pane to the canvas.
3. Set up any necessary joins between your tables.
    - (Self-reflection: A screenshot of the Data Source tab in Tableau, showing tables dragged to the canvas and a join configured.)

Step 4: Select a "Live" Connection

1. In the top-right corner of the Data Source tab, next to your connection name, ensure Live is selected. This is critical for achieving real-time updates.
    - (Self-reflection: A screenshot highlighting the "Live" radio button selected on the Data Source tab.)

Section 4: Building Your Dashboard

Now, let's create some visualizations and assemble them into a dashboard.

Step 1: Create Worksheets

1. Go to a new worksheet (click the "New Worksheet" icon).
2. Drag and drop dimensions and measures onto the shelves (Columns, Rows, Color, Size, etc.) to create your desired charts (e.g., line charts for trends, bar charts for comparisons, text tables for key metrics).
3. Name your worksheets clearly (e.g., "Sales Trend," "Current Stock Level").
    - (Self-reflection: Multiple screenshots of different types of charts being built on individual worksheets.)

Step 2: Create a New Dashboard

1. Click the "New Dashboard" icon.
2. Drag your created worksheets from the "Sheets" pane on the left to the dashboard canvas. Arrange them as desired.
    - (Self-reflection: A screenshot of an empty dashboard, then a screenshot of worksheets being dragged onto it and arranged.)

Step 3: Add Filters and Parameters (Optional but Recommended)

1. Drag relevant dimensions to the "Filters" pane on your worksheets and then select "Show Filter" from the context menu to make them available on the dashboard.
2. Parameters can also be used for dynamic input.
    - (Self-reflection: A screenshot showing how to add a quick filter to a dashboard.)

Section 5: Configuring Real-Time Refresh

Option 1: Automatic Refresh in Tableau Desktop (for individual users)

1. While on your dashboard, go to the top menu: Worksheet > Refresh Automatically.
2. This option is primarily for personal use in Tableau Desktop. It will refresh the data for the open workbook at fixed intervals when this menu option is checked.
    - (Self-reflection: A screenshot of the "Worksheet" menu with "Refresh Automatically" highlighted.)

Option 2: Automatic Refresh on Tableau Server/Cloud (for shared dashboards)

This is the most common and robust method for real-time dashboards in an organizational setting.

1. Publish Your Workbook:
    - With your dashboard active, go to Server > Publish Workbook.
    - Follow the prompts to select your Tableau Server/Cloud site, project, and name for the workbook.
    - In the "Publish Workbook to Tableau Server" dialog, under "Data Sources," ensure your live connection is marked as "Live."
    - (Self-reflection: Screenshots of the "Publish Workbook" dialog, especially highlighting the "Live" data source option.)
2. Set Up Refresh Schedule (for live connections on Server/Cloud):
    - Once published, when users view the dashboard on Tableau Server/Cloud, the live connection will query the database.

- Tableau Server/Cloud automatically manages live connection refreshes. The dashboard will show the most current data each time it's loaded by a user.
- Crucially, for views that are left open in a browser, you need to enable automatic refresh on the Server/Cloud view itself.
    - When viewing the dashboard on Tableau Server/Cloud, look for a refresh icon (often two arrows forming a circle) or an option under an "Edit" or "More Actions" menu on the dashboard view page.
    - Some organizations configure their Tableau Server/Cloud to refresh live dashboards in the browser automatically every X minutes. This is a server-side setting.
    - Alternatively, users can manually refresh their browser window or click the refresh button on the Tableau Server/Cloud interface.
- (Self-reflection: Screenshots of a published dashboard on Tableau Server/Cloud, pointing out the manual refresh button or any auto-refresh settings if visible.)

Option 3: Using a Web Data Connector (WDC) or Extensions (More Advanced)

- If your real-time data comes from an API that doesn't have a direct Tableau connector, you might use a Web Data Connector (WDC) to pull data. WDCs can be configured to refresh.
- Tableau Extensions can also provide custom real-time functionality by integrating external applications or scripts. These are more complex and require development skills.

Section 6: Performance Optimization

Real-time dashboards can be resource-intensive.

- Limit Data Displayed: Only show the most essential data points. Avoid displaying large, detailed tables that require heavy queries.
- Optimize Queries: Work with your database administrator to ensure the queries Tableau generates are efficient. Use database performance monitoring tools.
- Reduce Number of Worksheets: Each worksheet is a potential query to your live data source. Consolidate where possible.
- Context Filters: Use context filters to reduce the amount of data Tableau has to process in subsequent filters.
- Disable Unnecessary Auto-Updates: For complex dashboards in Desktop, you might temporarily pause automatic updates while building, then re-enable.

Section 7: Testing and Validation

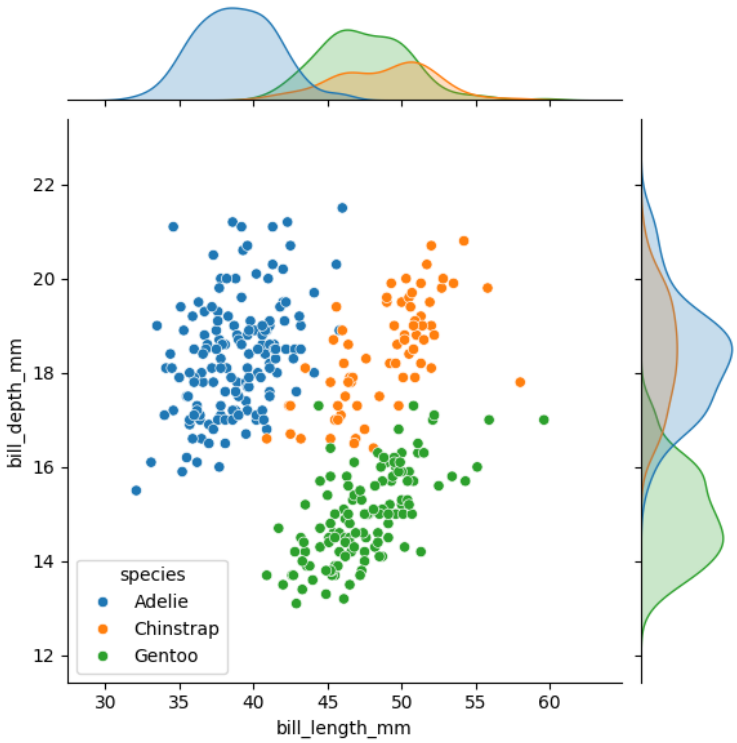
1. Test Refresh: Make a change in your underlying data source and then observe if and how quickly it reflects in your Tableau dashboard (both in Desktop and on Server/Cloud).
2. Monitor Performance: Keep an eye on your database's performance metrics to ensure frequent Tableau queries aren't overwhelming it.
3. User Acceptance Testing: Have end-users test the dashboard to ensure it meets their real-time data needs.

Practice 10: Explore Different types of Bivariate distributions

1. Scatter Plot (The Classic)

A scatter plot is the most direct way to see how two continuous variables interact. Adding a hue allows you to see how a third categorical variable affects that relationship.

```
import seaborn as sns

import matplotlib.pyplot as plt

# Load dataset

penguins = sns.load_dataset("penguins")

# Basic Scatter Plot with Marginal Histograms

sns.jointplot(data=penguins, x="bill_length_mm", y="bill_depth_mm", hue="species")

plt.show()
```



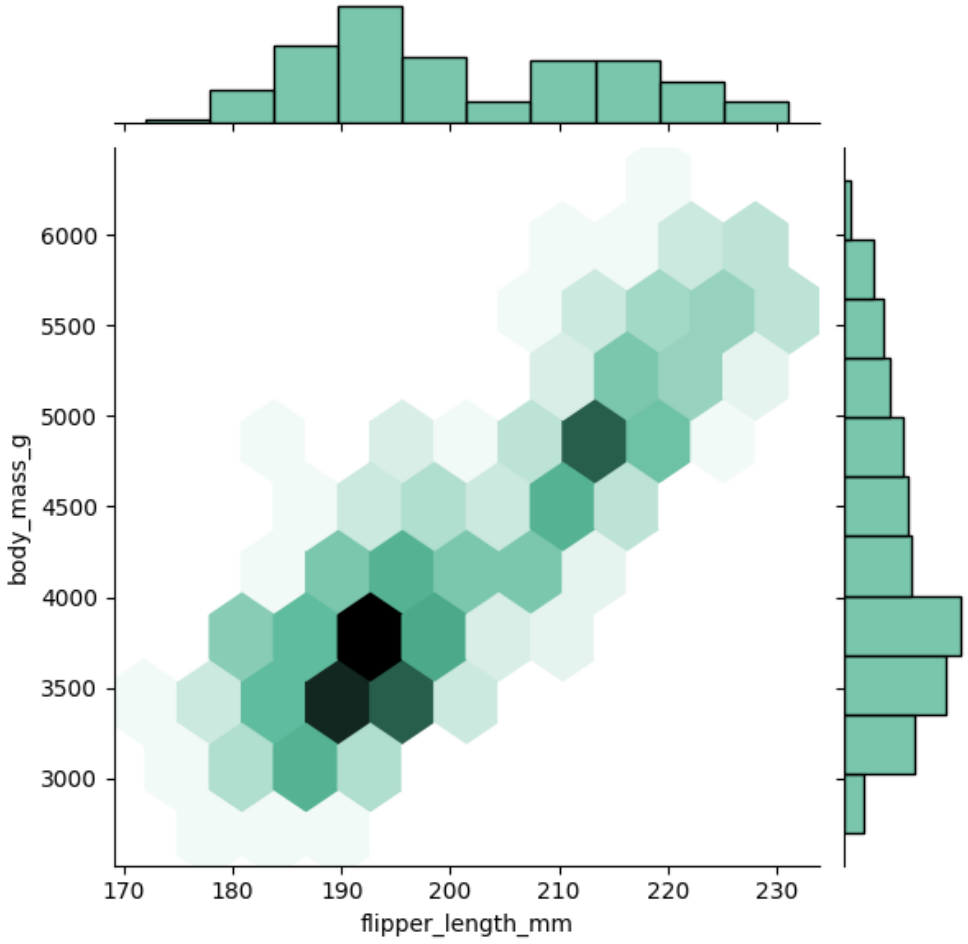
2. Hexbin Plot (For Dense Data)

When you have thousands of points, scatter plots become a "blob." Hexbins group points into hexagonal bins; darker colors indicate higher density.

Hexbin plot

sns.jointplot(data=penguins, x="flipper_length_mm", y="body_mass_g", kind="hex", color="#4CB391")

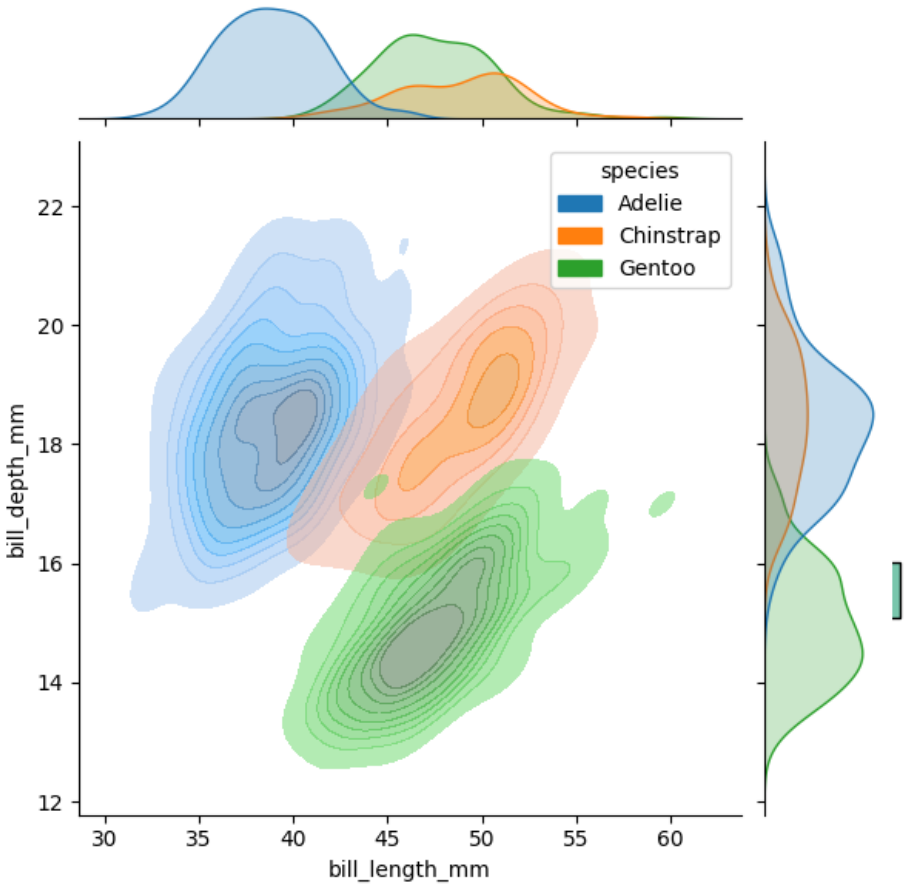
plt.show()



3. KDE Plot (Kernel Density Estimate)

This creates a "topographic map" of your data. It smooths out the noise to show the underlying probability distribution.

```
# 2D KDE Plot

sns.jointplot(

    data=penguins,

    x="bill_length_mm", y="bill_depth_mm", hue="species",

    kind="kde", fill=True, alpha=0.5

)

plt.show()
```



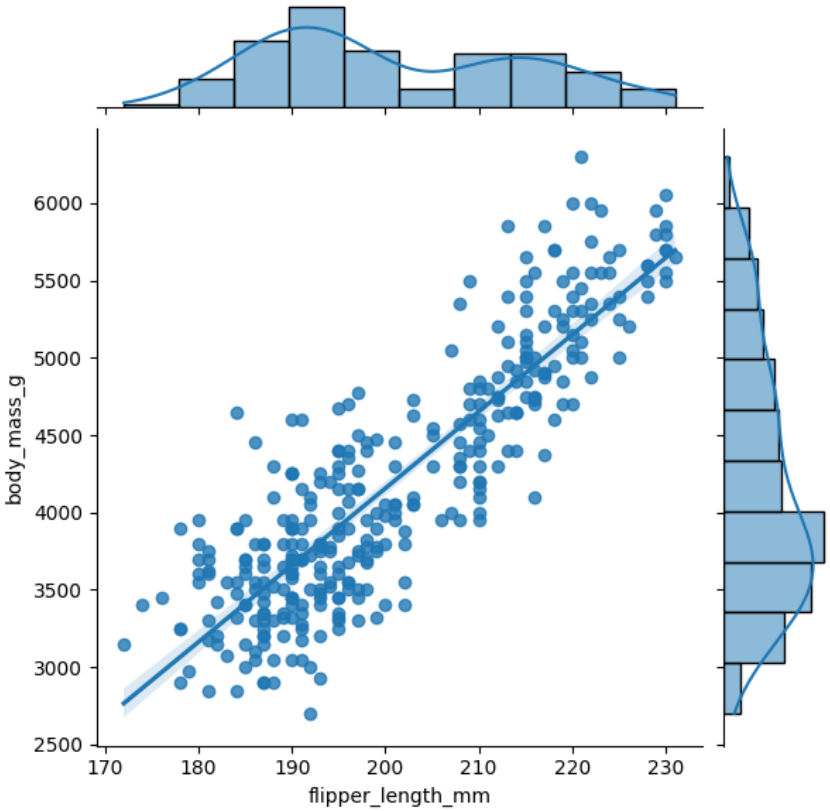
4. Reg Plot (Regression)

If you want to see the trend line (linear relationship) along with the bivariate distribution, use kind="reg".

Regression Plot

sns.jointplot(data=penguins, x="flipper_length_mm", y="body_mass_g", kind="reg")

plt.show()



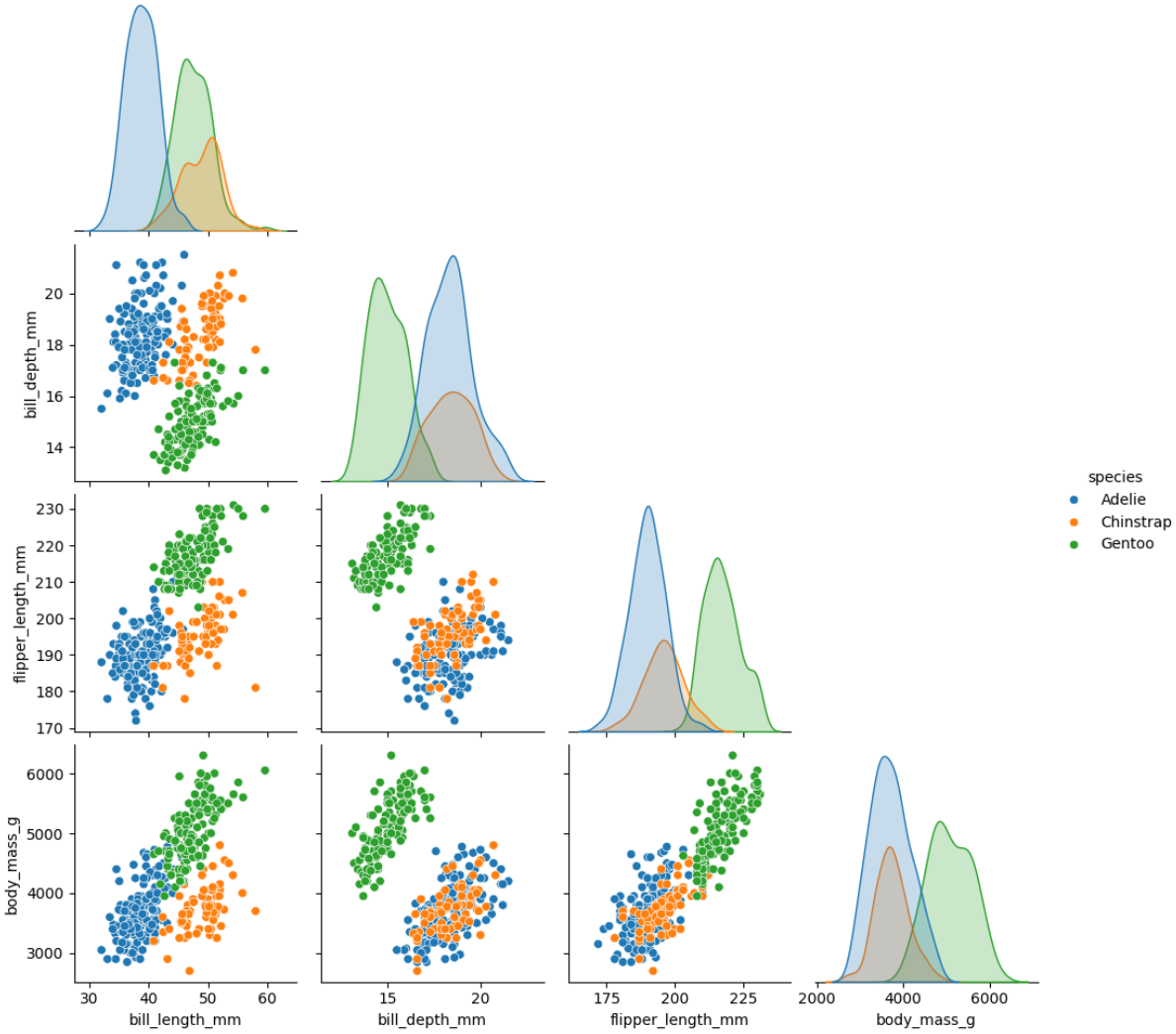
5. Pair Plots (The "Whole Picture" Approach)

If you have many variables and want to see the bivariate relationship between *every* possible pair, pairplot is your best friend.

Matrix of bivariate plots

sns.pairplot(penguins, hue="species", corner=True)

plt.show()

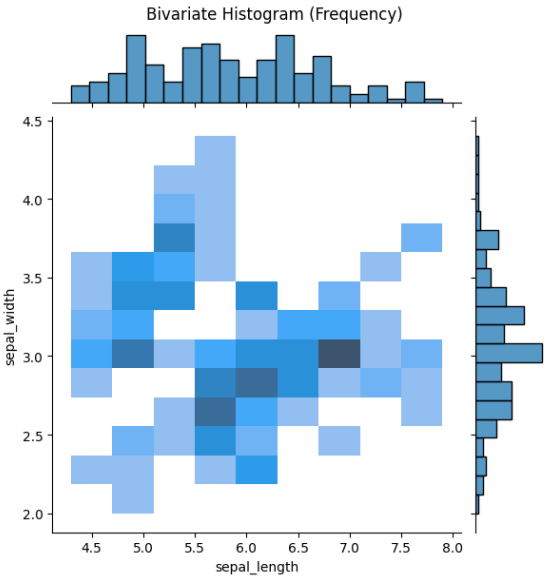


Practice 11: Analyze Bivariate distributions and multiple variable parts

1. Bivariate Analysis: Distribution Types

Bivariate analysis focuses on the relationship between two variables (X and Y). The "type" of distribution is often determined by the density and correlation of the data points.

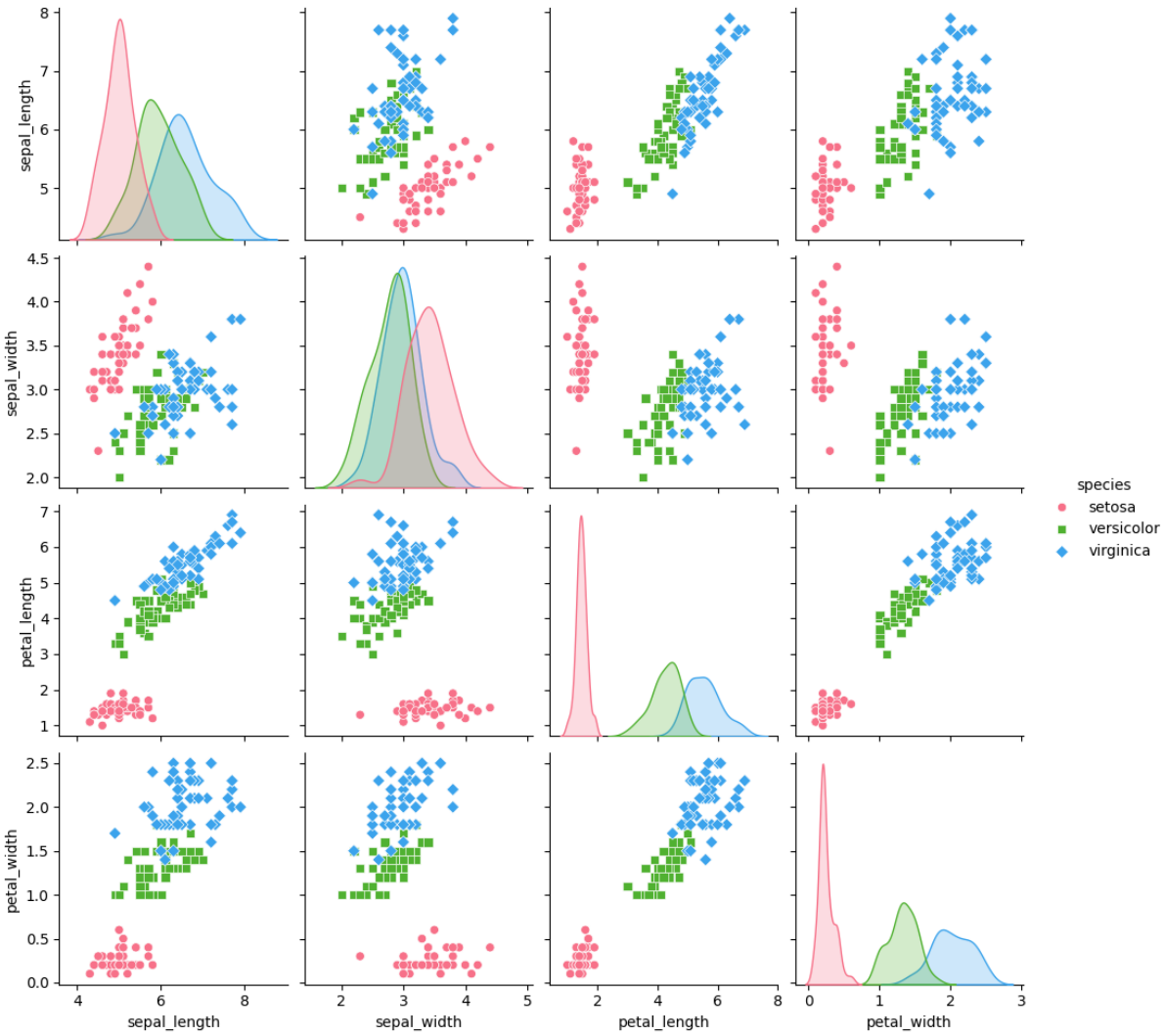
```
import seaborn as sns

import matplotlib.pyplot as plt

# Load a robust dataset

iris = sns.load_dataset("iris")

# 2D Histogram (Discretized Bivariate Distribution)

# This shows the joint frequency distribution in a grid format.

sns.jointplot(data=iris, x="sepal_length", y="sepal_width", kind="hist",
marginal_kws=dict(bins=20))

plt.suptitle("Bivariate Histogram (Frequency)", y=1.02)

plt.show()
```



2. Multivariate Analysis: The "Pair" Strategy

When you have more than two variables, you cannot plot them all on a single 2D plane easily. Instead, we use a **Pair Plot** to see every possible bivariate combination in a matrix.

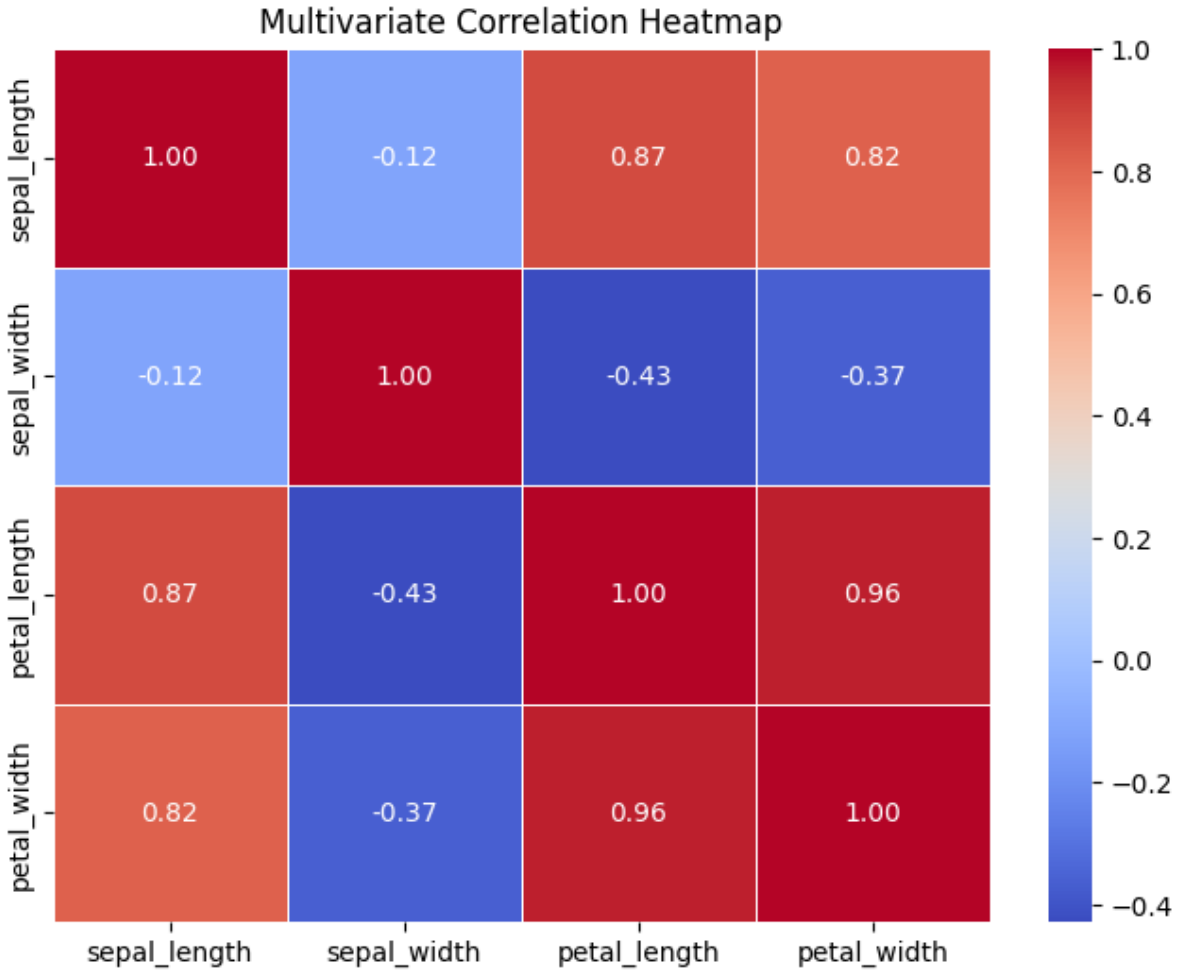
```
# Multivariate Matrix Plot

# 'hue' adds a third categorical dimension to the bivariate plots

sns.pairplot(iris, hue="species", palette="husl", markers=["o", "s", "D"])

plt.show()
```



3. Multivariate Analysis: Heatmaps (Correlation Matrix)

To analyze the "strength" of the relationships across many variables simultaneously, we calculate a correlation matrix. This represents the **Multivariate Normal** relationship across the entire feature set.

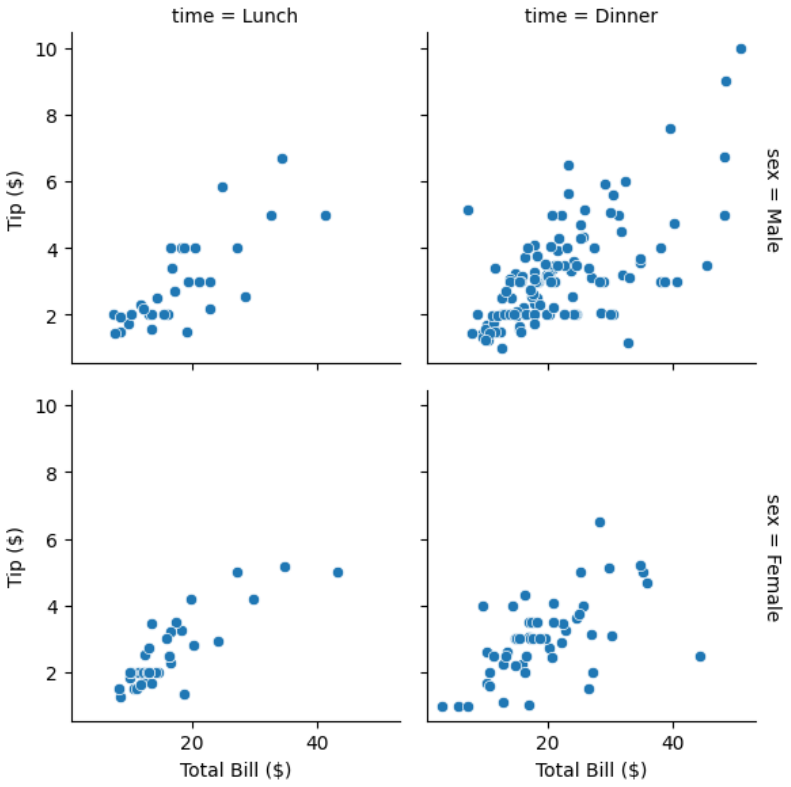
```
# Calculate correlation matrix

corr = iris.drop(columns='species').corr()

# Plot Heatmap

plt.figure(figsize=(8, 6))

sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)

plt.title("Multivariate Correlation Heatmap")

plt.show()
```



4. Complex Multivariate: Facet Grids

If you want to see a bivariate distribution (like X vs Y) filtered by two other categorical variables (like A and B), you use a **FacetGrid**. This is "True" multivariate exploration.

```
# Tips dataset is better for categorical multivariate analysis
tips = sns.load_dataset("tips")

# Bivariate (Total Bill vs Tip) conditioned by Time and Sex (Multivariate)
g = sns.FacetGrid(tips, col="time", row="sex", margin_titles=True)
g.map_dataframe(sns.scatterplot, x="total_bill", y="tip")
g.set_axis_labels("Total Bill ($)", "Tip ($)")
plt.show()
```

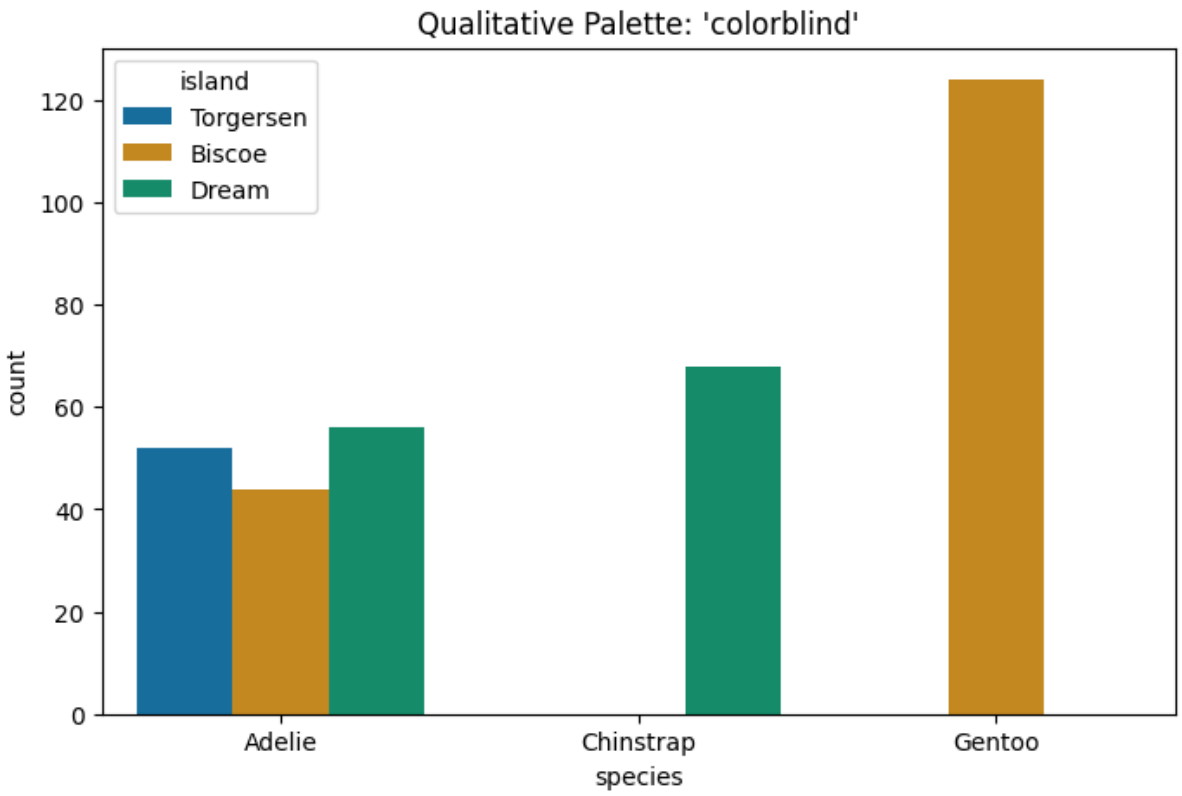


Practice 12: Program using color palettes

1. Qualitative Palettes (Categorical Data)

Used when your data has no inherent order (e.g., Species, Countries, or Departments).

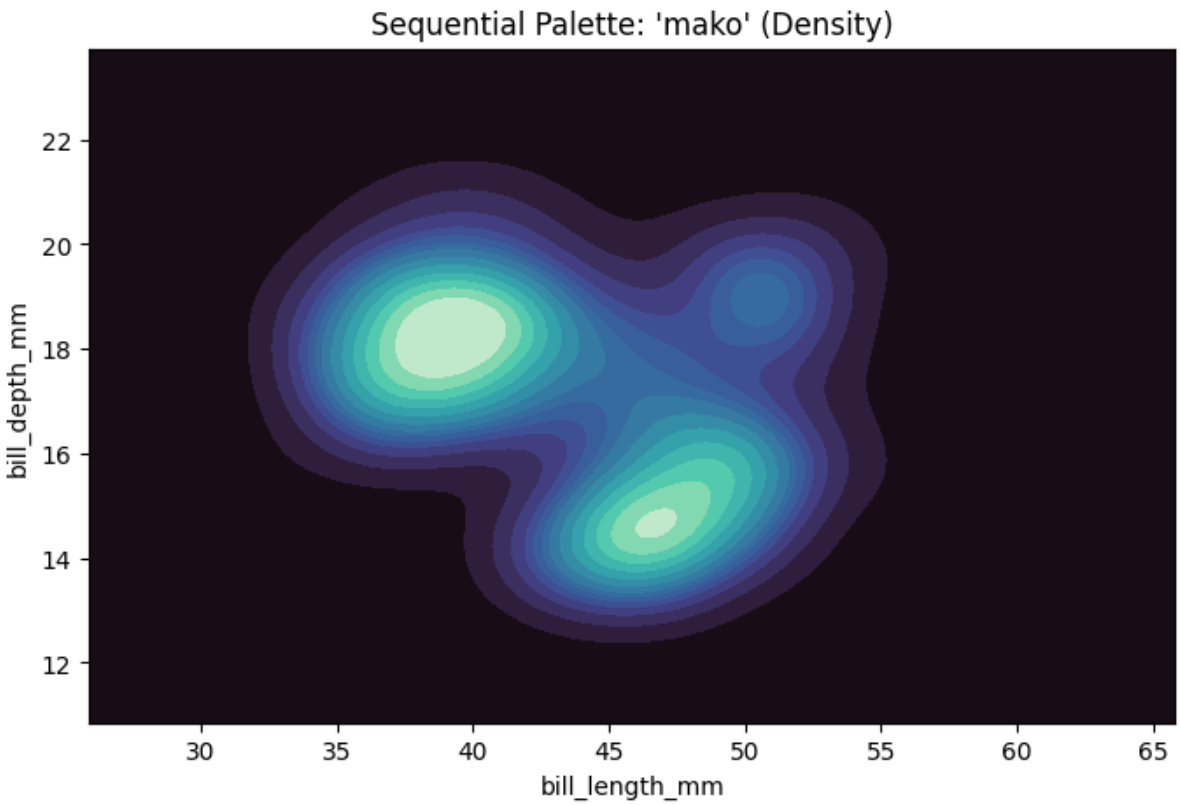
```
import seaborn as sns
import matplotlib.pyplot as plt

# Load sample data
penguins = sns.load_dataset("penguins")

# Set a qualitative palette: 'pastel', 'bright', 'dark', 'colorblind', 'husl'
sns.set_palette("colorblind")

plt.figure(figsize=(8, 5))
sns.countplot(data=penguins, x="species", hue="island")
plt.title("Qualitative Palette: 'colorblind'")
plt.show()
```



2. Sequential Palettes (Ordered Data)

Used when data has a clear range from low to high (e.g., Age, Temperature, or Count). Light colors usually represent low values, and dark colors represent high values.

```
# 'Blues', 'Greens', 'Reds', 'rocket', 'mako'

plt.figure(figsize=(8, 5))

sns.kdeplot(

    data=penguins, x="bill_length_mm", y="bill_depth_mm",

    fill=True, thresh=0, levels=15, cmap="mako"

)

plt.title("Sequential Palette: 'mako' (Density)")

plt.show()
```



3. Diverging Palettes (Data with a Mid-point)

Used when both high and low extremes are interesting, and there is a meaningful midpoint (e.g., Temperature deviations from 0, or Correlation coefficients).

'vlag', 'icefire', 'coolwarm', 'RdBu'

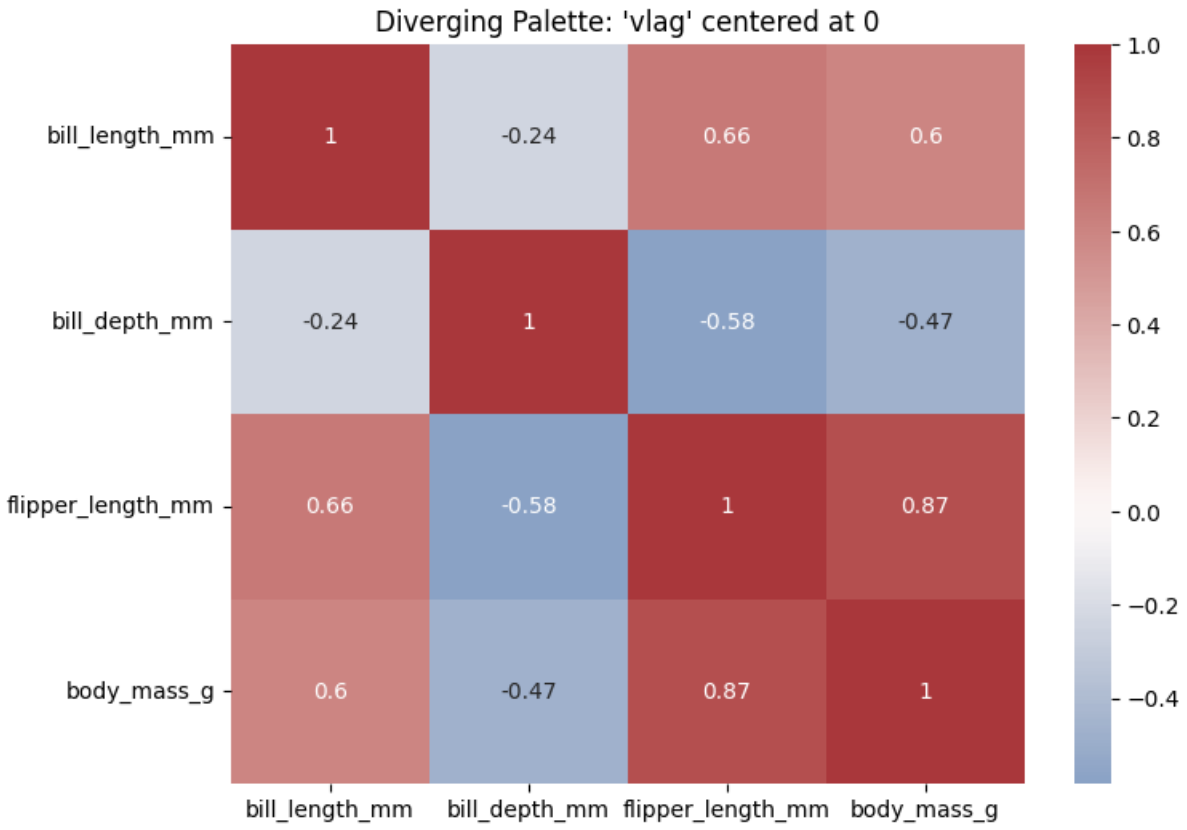
corr = penguins.corr(numeric_only=True)

plt.figure(figsize=(8, 6))

sns.heatmap(corr, annot=True, cmap="vlag", center=0)

plt.title("Diverging Palette: 'vlag' centered at 0")

plt.show()



4. Customizing and Displaying Palettes

If you want to create your own color scheme or see what a palette looks like before applying it, use color_palette().

Display a custom palette based on a specific hex color

custom_pal = sns.light_palette("#2ecc71", n_colors=10)

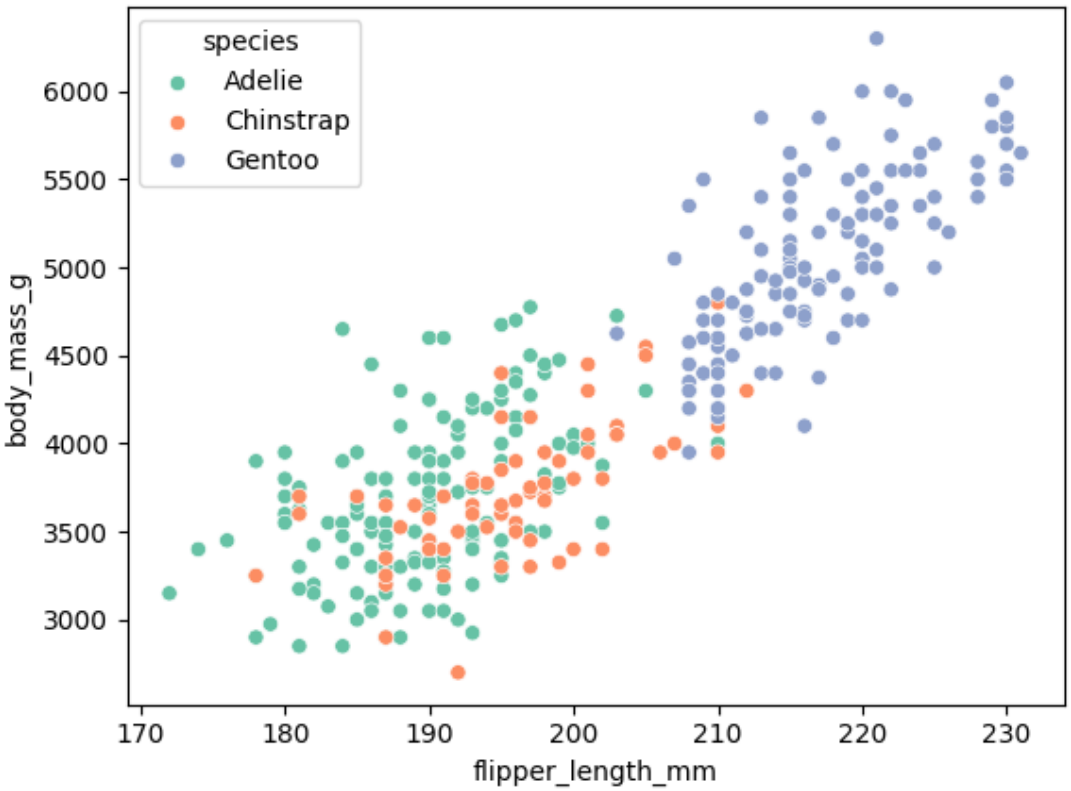
sns.palplot(custom_pal)

plt.show()

Applying a custom palette to a scatter plot

sns.scatterplot(data=penguins, x="flipper_length_mm", y="body_mass_g", hue="species", palette="Set2")

plt.show()


species
Adelie
Chinstrap
Gentoo
body_mass_g
6000
5500
5000
4500
4000
3500
3000
170
180
190
200
210
220
230
flipper_length_mm

Practice 13: Summary Statistics using native

```
import math


class NativeStats:
    def __init__(self, data):
        self.data = sorted(data)
        self.n = len(data)

    def mean(self):
        return sum(self.data) / self.n

    def median(self):
        mid = self.n // 2
        if self.n % 2 == 0:
            return (self.data[mid - 1] + self.data[mid]) / 2
        return self.data[mid]

    def variance(self):
        mu = self.mean()
        return sum((x - mu) ** 2 for x in self.data) / self.n

    def std_dev(self):
        return math.sqrt(self.variance())
```

Usage

dataset = [10, 2, 38, 23, 38, 23, 21]

stats = NativeStats(dataset)

print(f"Mean: {stats.mean():.2f}")

print(f"Median: {stats.median()}")

print(f"Std Dev: {stats.std_dev():.2f}")

Output:

```
Mean: 22.14
Median: 23
Std Dev: 12.30
```

Practice 14: Plot graphs using Matplotlib

Matplotlib is the foundation of data visualization in the Python ecosystem. It allows for the creation of static, animated, and interactive visualizations. Below is a comprehensive implementation showing how to create the most common types of statistical graphs.

```
import matplotlib.pyplot as plt

import numpy as np


# 1. Prepare Data

x = np.linspace(0, 10, 100)

y_line = np.sin(x)


categories = ['D', 'A', 'C', 'B']

values = [12, 10, 36, 24]


# Sorting data for the bar chart

sorted_indices = np.argsort(values)

sorted_cats = [categories[i] for i in sorted_indices]

sorted_vals = [values[i] for i in sorted_indices]


x_scatter = np.random.rand(50)

y_scatter = np.random.rand(50)

data_hist = np.random.randn(1000)


# 2. Create the Figure and Axes

# Using subplots(rows, cols) creates a layout grid
```

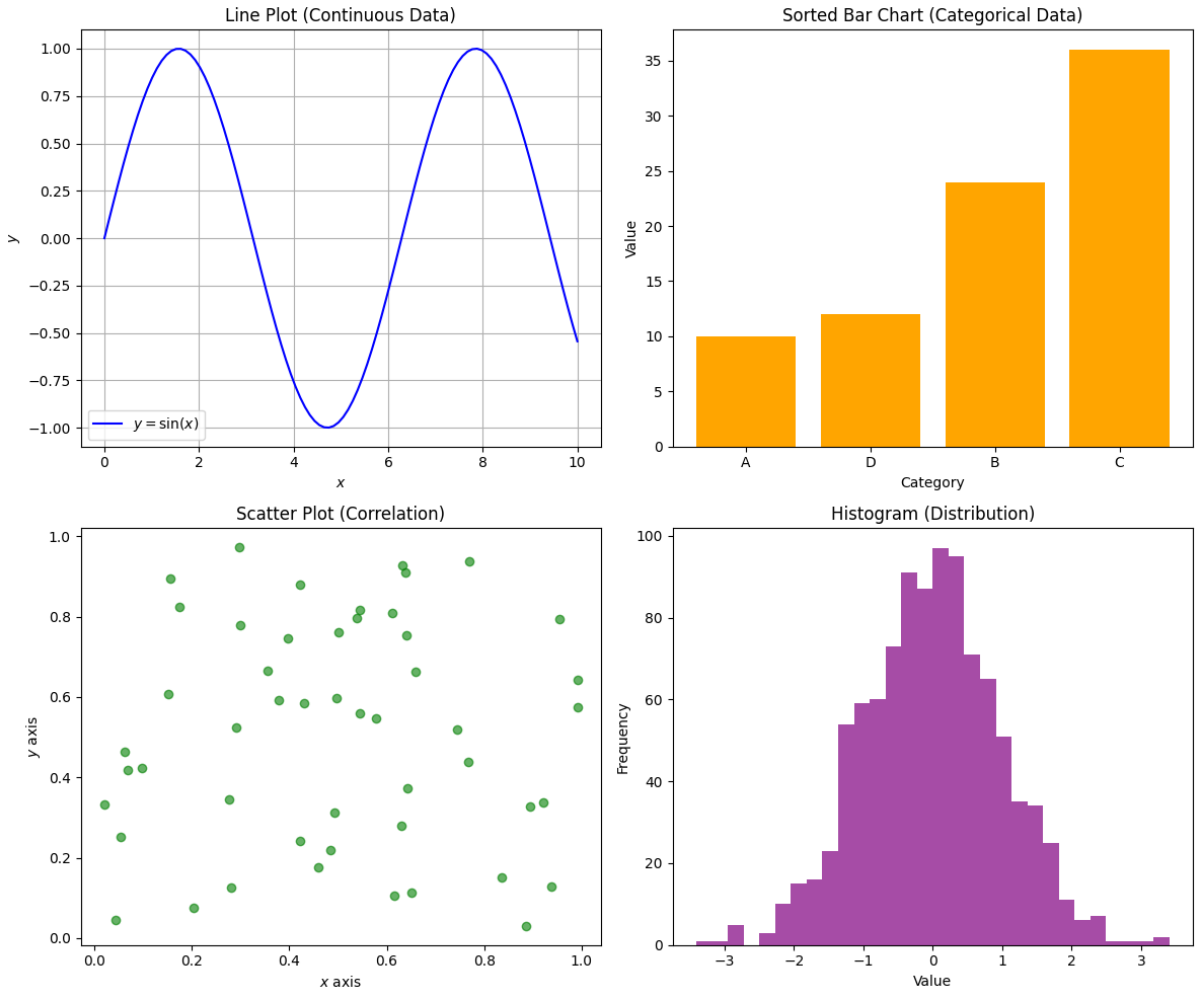
```
fig, axs = plt.subplots(2, 2, figsize=(12, 10))

# --- Line Plot ---
axs[0, 0].plot(x, y_line, color='blue', label='$y = \sin(x)$')
axs[0, 0].set_title('Line Plot (Continuous Data)')
axs[0, 0].set_xlabel('$x$')
axs[0, 0].set_ylabel('$y$')
axs[0, 0].legend()
axs[0, 0].grid(True)

# --- Bar Chart ---
axs[0, 1].bar(sorted_cats, sorted_vals, color='orange')
axs[0, 1].set_title('Sorted Bar Chart (Categorical Data)')
axs[0, 1].set_xlabel('Category')
axs[0, 1].set_ylabel('Value')

# --- Scatter Plot ---
axs[1, 0].scatter(x_scatter, y_scatter, alpha=0.6, color='green')
axs[1, 0].set_title('Scatter Plot (Correlation)')
axs[1, 0].set_xlabel('$x$ axis')
axs[1, 0].set_ylabel('$y$ axis')

# --- Histogram ---
axs[1, 1].hist(data_hist, bins=30, color='purple', alpha=0.7)
axs[1, 1].set_title('Histogram (Distribution)')
axs[1, 1].set_xlabel('Value')
```

axs[1, 1].set_ylabel('Frequency')

3. Final Adjustments

plt.tight_layout()

plt.savefig('matplotlib_summary.png')



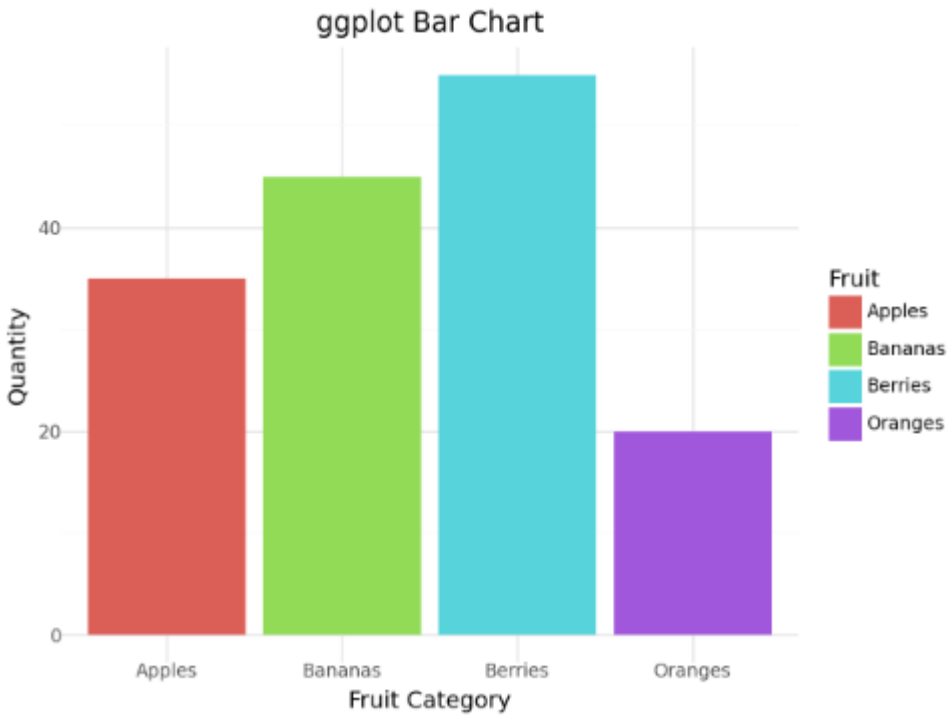
Practice 15: Bar chart using ggplot, bokeh and pygal

1. Plotnine (ggplot for Python)

plotnine is the most faithful implementation of R's ggplot2 in Python. It is excellent for static, publication-quality figures.

```
from plotnine import ggplot, aes, geom_bar, labs, theme_minimal
import pandas as pd

# Data must be in a DataFrame
df = pd.DataFrame({
    'Fruit': ['Apples', 'Bananas', 'Oranges', 'Berries'],
    'Count': [35, 45, 20, 55]
})

chart = (
    ggplot(df, aes(x='Fruit', y='Count', fill='Fruit'))
    + geom_bar(stat='identity')
    + labs(title='ggplot Bar Chart', x='Fruit Category', y='Quantity')
    + theme_minimal()
)

print(chart)
```

Output



2. Bokeh (Interactive Web Plots)

Bokeh generates HTML and JavaScript, making it perfect for dashboards where you want to hover over bars or zoom in.

from bokeh.plotting import figure, show

from bokeh.io import output_notebook # <--- Essential for Colab

1. Initialize the notebook output

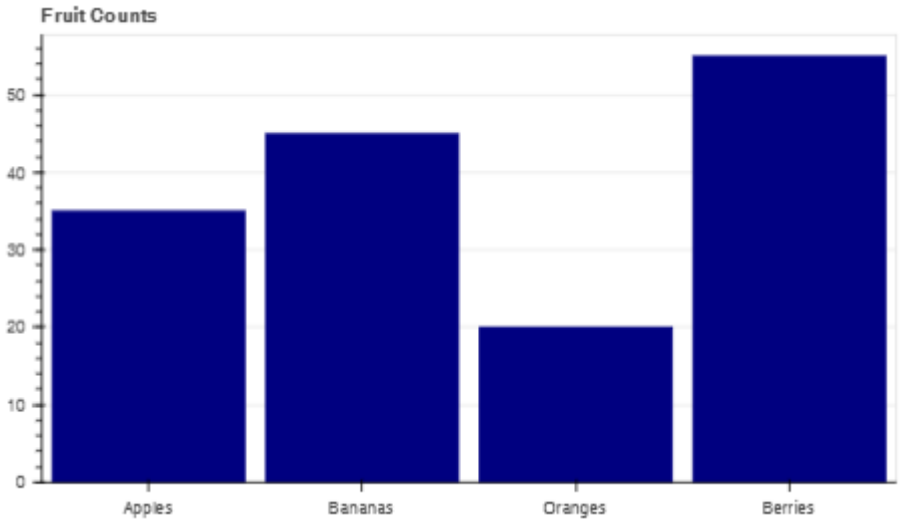
output_notebook()

fruits = ['Apples', 'Bananas', 'Oranges', 'Berries']

counts = [35, 45, 20, 55]

```
# 2. Create the figure
p = figure(x_range=fruits, height=350, title="Fruit Counts",
        toolbar_location=None, tools="hover") # Added hover for interactivity

# 3. Add bars
p.vbar(x=fruits, top=counts, width=0.9, color="navy")

# 4. Customization
p.xgrid.grid_line_color = None
p.y_range.start = 0

# 5. Show inside the cell
show(p)
```

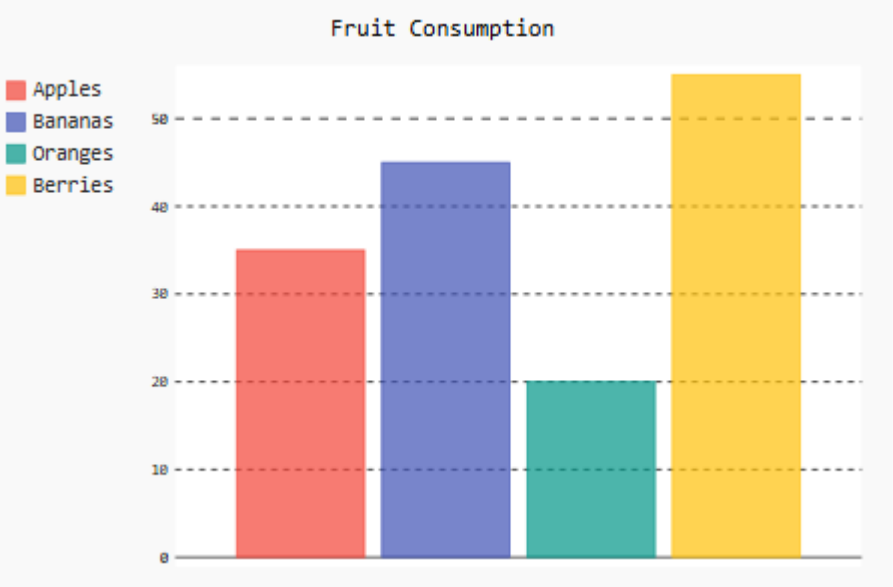
Output:



3. Pygal (SVG Charts)

Pygal is unique because it outputs .svg files. These are resolution-independent and have built-in tooltips without needing a complex server.

```
# 1. Install pygal (Required for Colab)

!pip install pygal -q


import pygal

from IPython.display import SVG, display


# 2. Create the chart

# We use 'explicit_size=True' to ensure it renders correctly in the cell

bar_chart = pygal.Bar(width=600, height=400, explicit_size=True)

bar_chart.title = 'Fruit Consumption'


# 3. Add data

bar_chart.add('Apples', 35)

bar_chart.add('Bananas', 45)

bar_chart.add('Oranges', 20)

bar_chart.add('Berries', 55)


# 4. Render and Display

# We use render() to get the SVG data and display it directly

chart_svg = bar_chart.render(is_unicode=True)

display(SVG(chart_svg))
```


Fruit Consumption
Apples
Bananas
Oranges
Berries
50
40
30
20
10
0