APPENDIX

CHAPTER 1: INTRODUCTION: VISUAL ANALYTICS WITH TABLEAU

ACTIVITY 1.01: IDENTIFY AND CREATE THE APPROPRIATE CHART TO FIND OUTLIERS IN YOUR DATA

Solution:

In this activity, you will identify and create the appropriate chart to find outliers in your data. The dataset being used has two measures—namely, **`Profit`** and **`Marketing`**. **`Marketing`** refers to the money being spent on marketing efforts, while **`Profit`** is the profit that you are making. You need to compare **`Marketing`** and **`Profit`** across different products and across different markets (so, two dimensions and two measures).

The outliers to be identified are as follows:

- High marketing and low profit
- Low marketing and high profit

You will use the *CoffeeChain Query* table from the *Sample-Coffee Chain.mdb* dataset.

Perform the following steps to complete this activity:

1. Connect to the Microsoft Access file named *Sample-Coffee Chain.mdb*. Refer to the following screenshot:

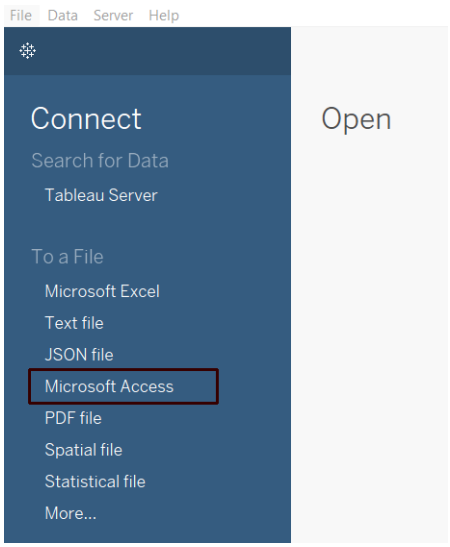


Figure 1.46: A screenshot showing the Microsoft Access connection option on the landing page of Tableau

2. Drag the **`CoffeeChain Query`** table into the **`Drag sheets here`** section. Refer to the following screenshot:

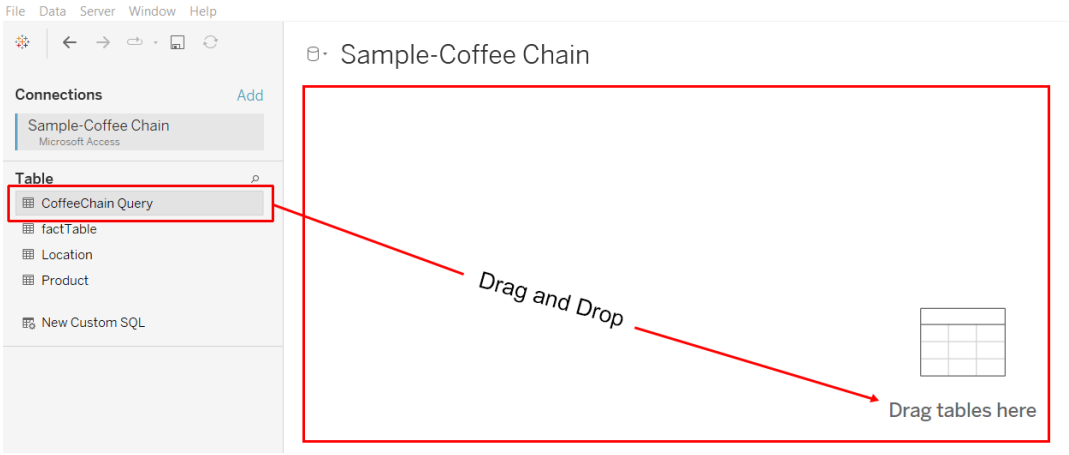


Figure 1.47: A screenshot showing how to connect to and read the data from the CoffeeChain Query table

3. Keeping the default selections, click on the **`Sheet1`** tab, which also has a popup saying **`Go to Worksheet`**. Refer to the following screenshot:

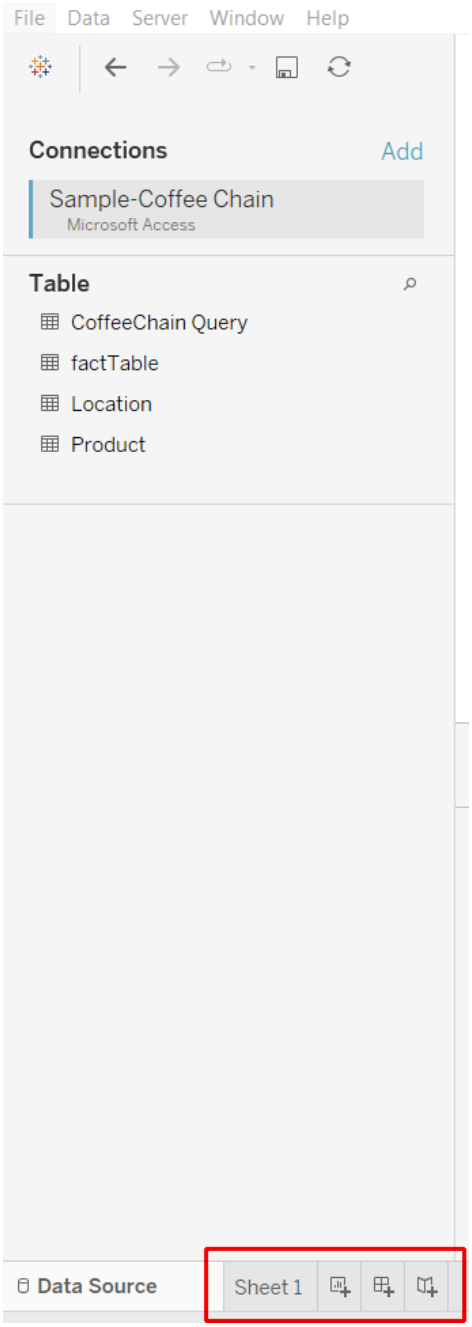


Figure 1.48: A screenshot showing how to use the Go to Worksheet option

4. Once you do that, you will see the Tableau workspace with the **`Dimensions`** and **`Measures`** sections along with the **`Rows`** shelf, **`Columns`** shelf, and so on. Determine which chart would help you identify the *outliers* in your data when looking at *two measures* (that is, **`Profit`** and **`Marketing`**) across two dimensions (that is, Product and Market). The outliers that you are looking for are *high marketing and low profit* and *low marketing and high profit* and, using the given hint, you will identify a scatter plot as the best chart to address your requirement.

5. Create a scatter plot by performing multi-selection in your **`Data`** pane using the *CTRL* key and the **`Show Me button`**. Refer to the following screenshot:

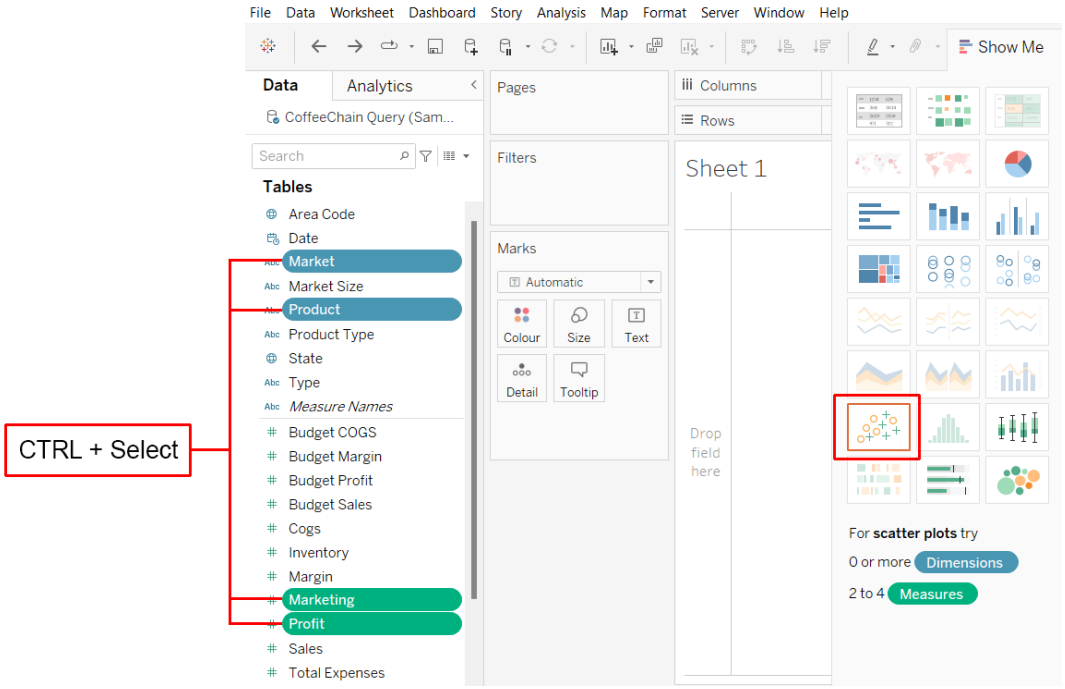


Figure 1.49: A screenshot showing Show Me with the scatter plot option highlighted

After selecting the scatter plot option, your view will update, as shown in the following screenshot:

Figure 1.50: A screenshot showing the final output of the scatter plot created using the Show Me button

6. As you can see in the preceding screenshot, the point highlighted in **black** is the product that is **`Low Marketing High Profit`**. Referring to the legends, identify that product is **`Colombian`**, which is sold in the **`East`** market, whereas the one highlighted in red is the product that is **`High Marketing Low Profit`**. Again, after referring to the legends, identify that the Product is **`Caffe Mocha`**, which is sold in the **`East`** market.

7. Export your view to PowerPoint by choosing the **`File > Export as PowerPoint`** option from the toolbar menu. Save this PowerPoint export on your desktop, as shown in the following screenshot:

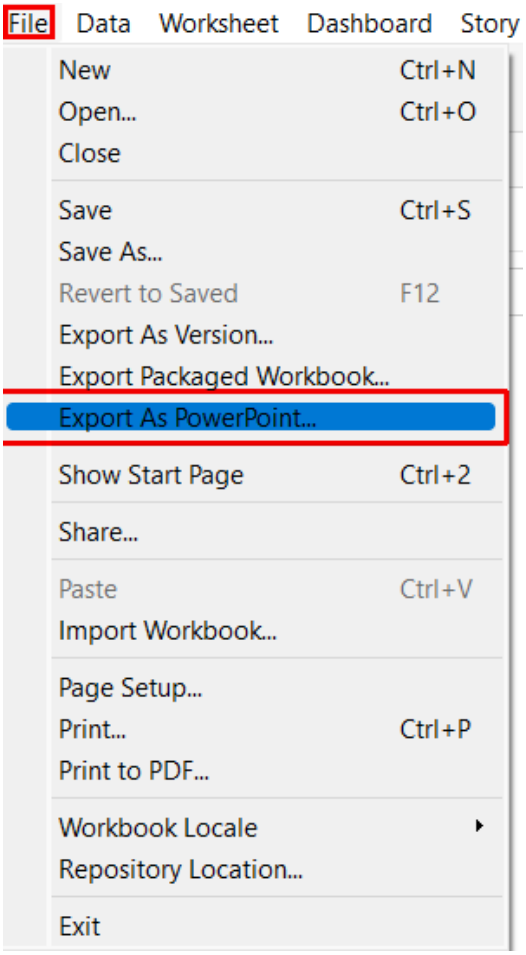


Figure 1.51: A screenshot showing the Export to PowerPoint option

8. Choose the **`File > Save As`** option from the toolbar menu, which will open a new window to save the workbook as a packaged workbook. Choose the **`Tableau Packaged Workbook (.twbx)`** option and type the filename as **`My first Tableau view.twbx`**. Refer to the following screenshot:

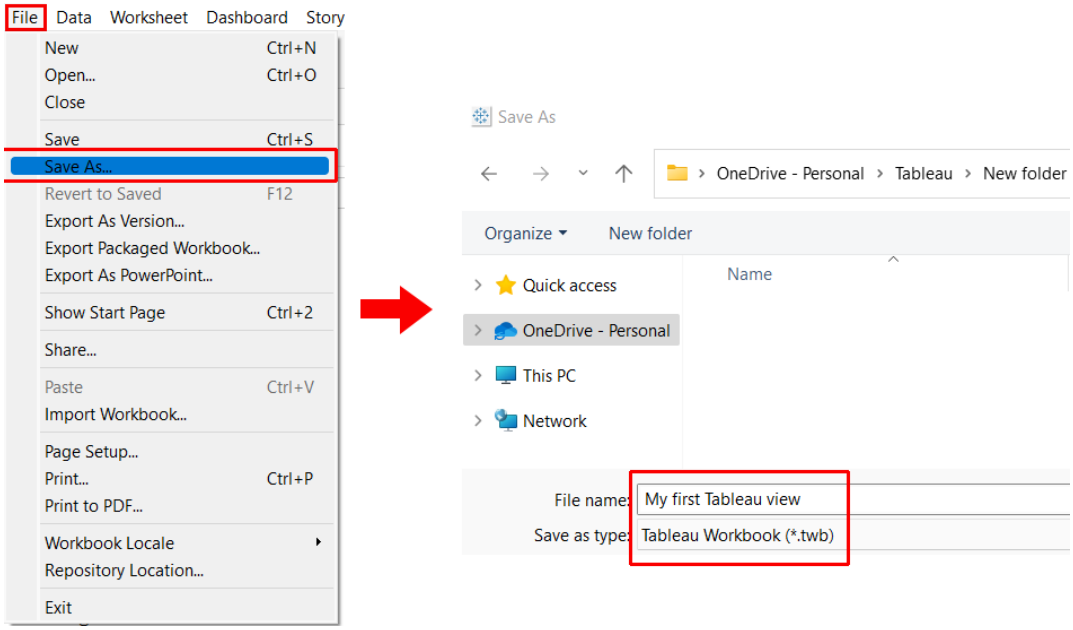


Figure 1.52: A screenshot showing how to save the file as a Tableau Packaged Workbook (.twbx)

In this activity, you used dimensions (**`Profit`** and **`Marketing`**) and measures (**`Market`** and **`Product`**) to find outliers in the given data.

CHAPTER 2: DATA PREPARATION: USING TABLEAU DESKTOP

ACTIVITY 2.01: IDENTIFYING THE RETURNED ORDERS

Solution:

1. Connect to the **`Sample - Superstore`** dataset in Tableau using the **`Connect`** pane.

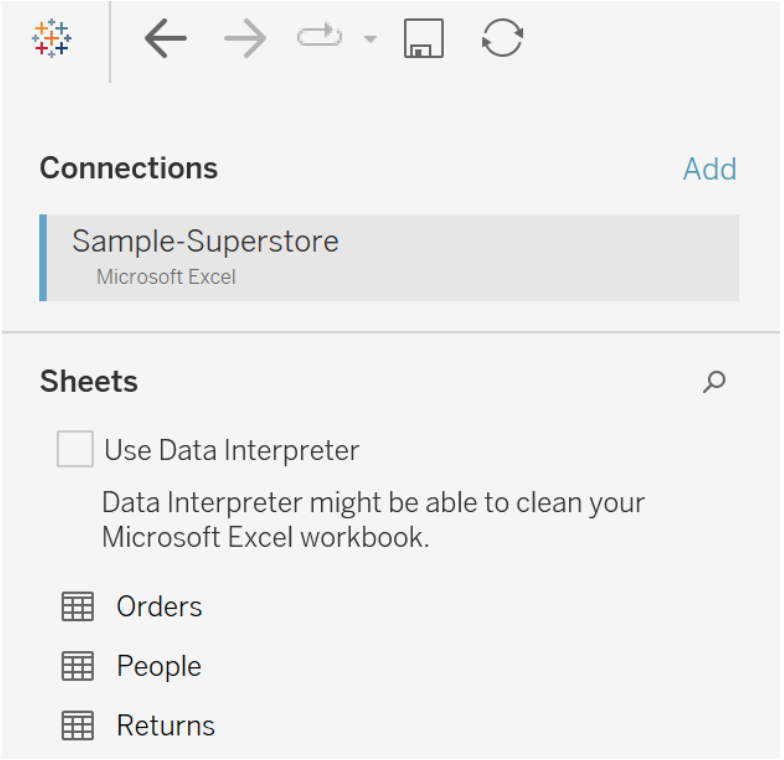


Figure 2.81: Connect pane displaying sheets within the input Excel

2. Rename the data source **`Activity 1`**.

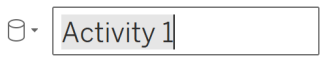


Figure 2.82: Renaming the data source

3. Drag the **`Orders`** table onto the canvas.

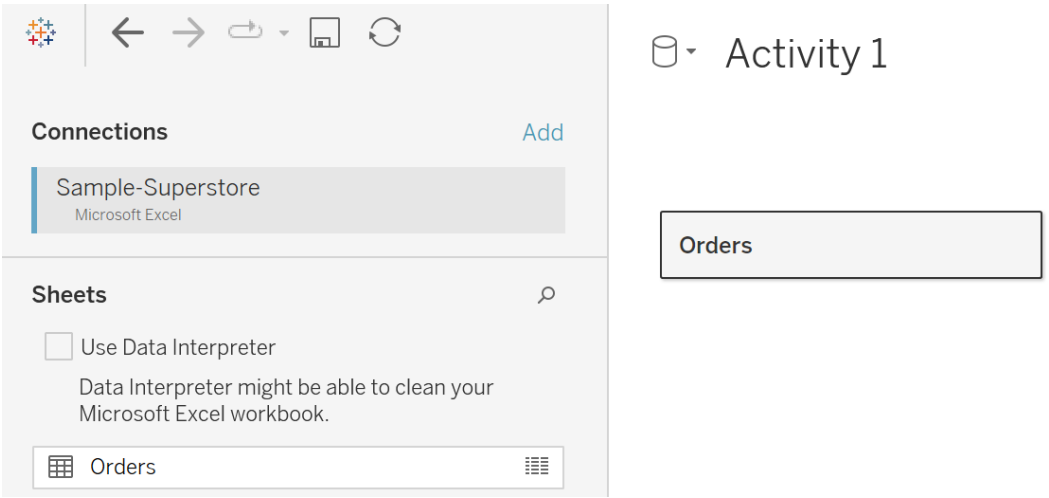


Figure 2.83: Dragging the Orders table onto the canvas

4. Drag the **`Returns`** table onto the canvas.

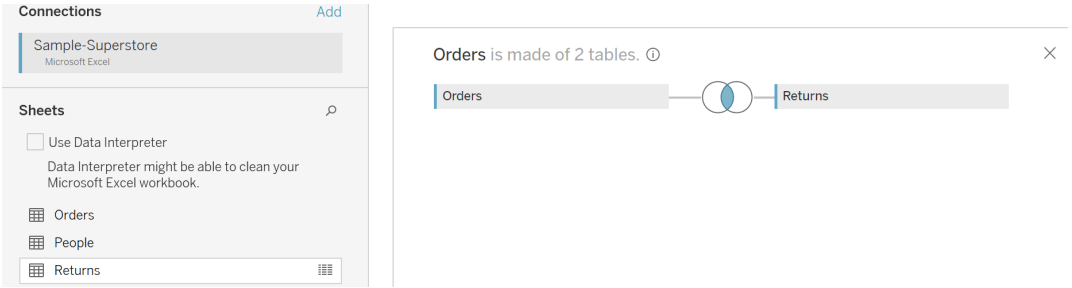


Figure 2.84: Adding the Returns table to the canvas

5. Check, for each join type, what the value of the row count is. Start with the inner join, where you get 3,226 records. Here, all matching rows from the **`Returns`** table and the **`Orders`** table are returned.

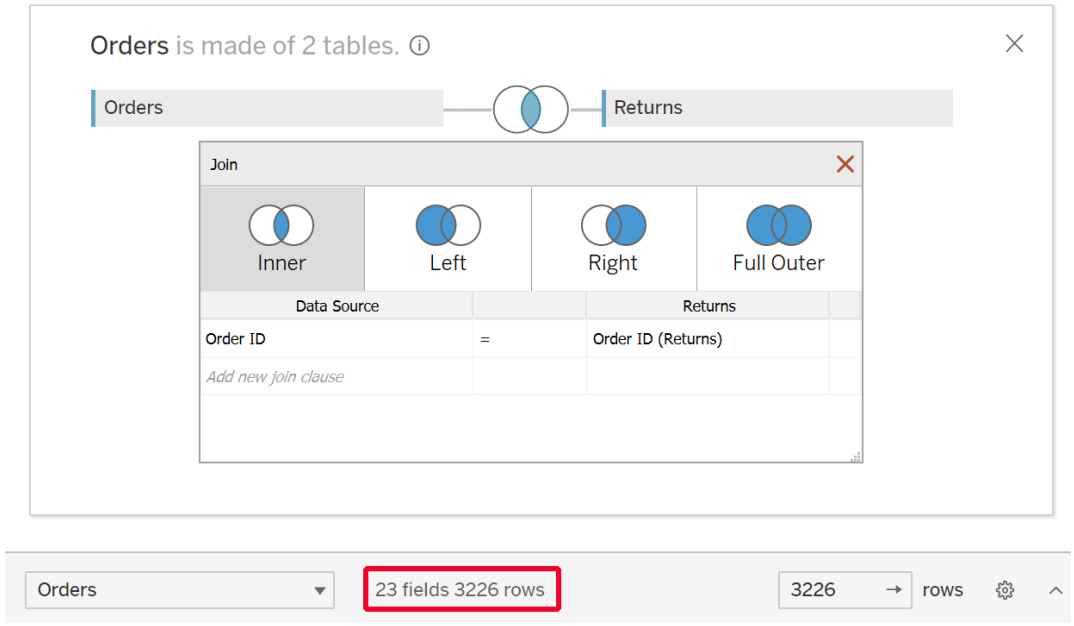


Figure 2.85: Inner join interpretation

6. Change the join to left. You get 12,420 records. Here, in addition to all rows from the **`Returns`** table, you also get all the orders that were not returned.

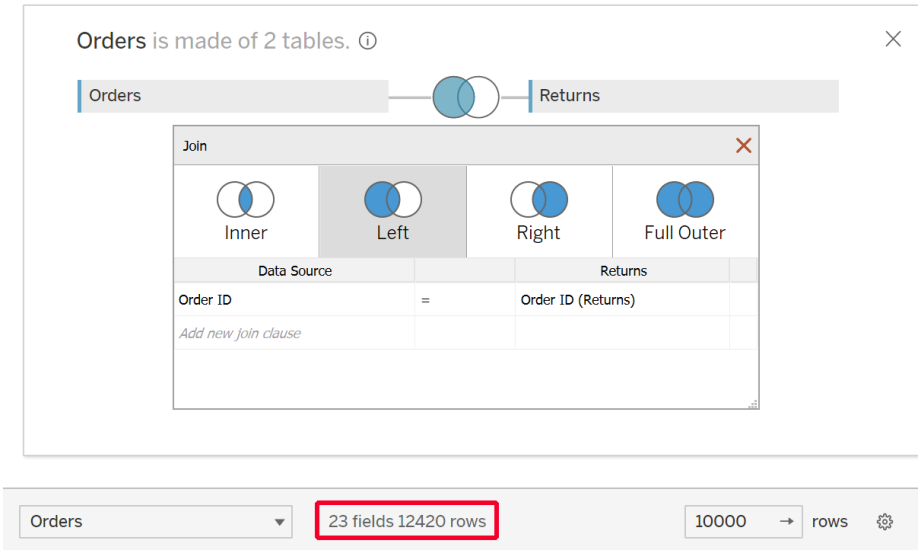


Figure 2.86: Left join interpretation

7. Change the join to right. You get 3,226 records. Here, all rows from the **`Returns`** table are returned along with only those matching the **`Orders`** table. Identify which join gives a similar result.

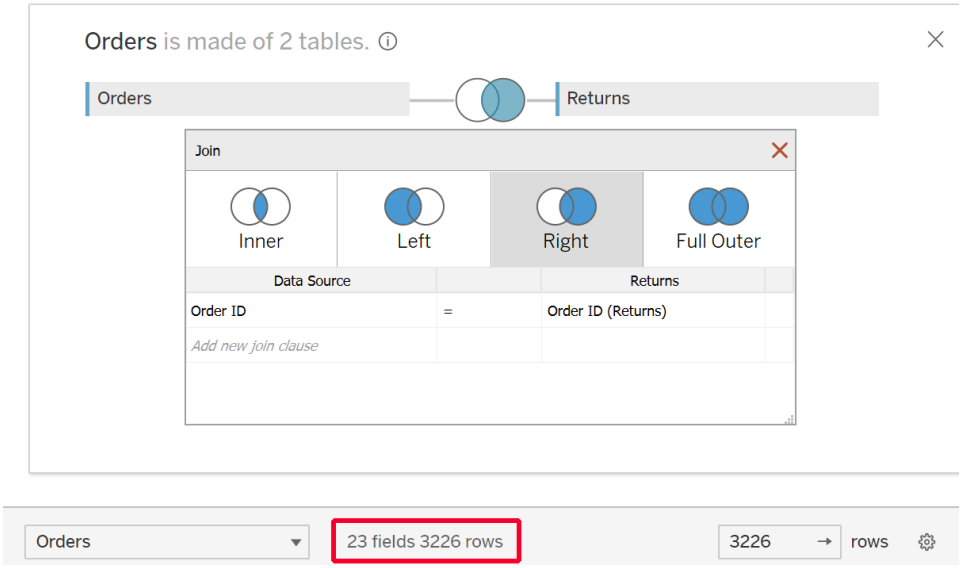


Figure 2.87: Right join interpretation

8. Change the join to full outer. You get 12,420 records. Here, all rows from the **`Returns`** table along with the **`Orders`** table are returned. These are all the orders that are present in your data.

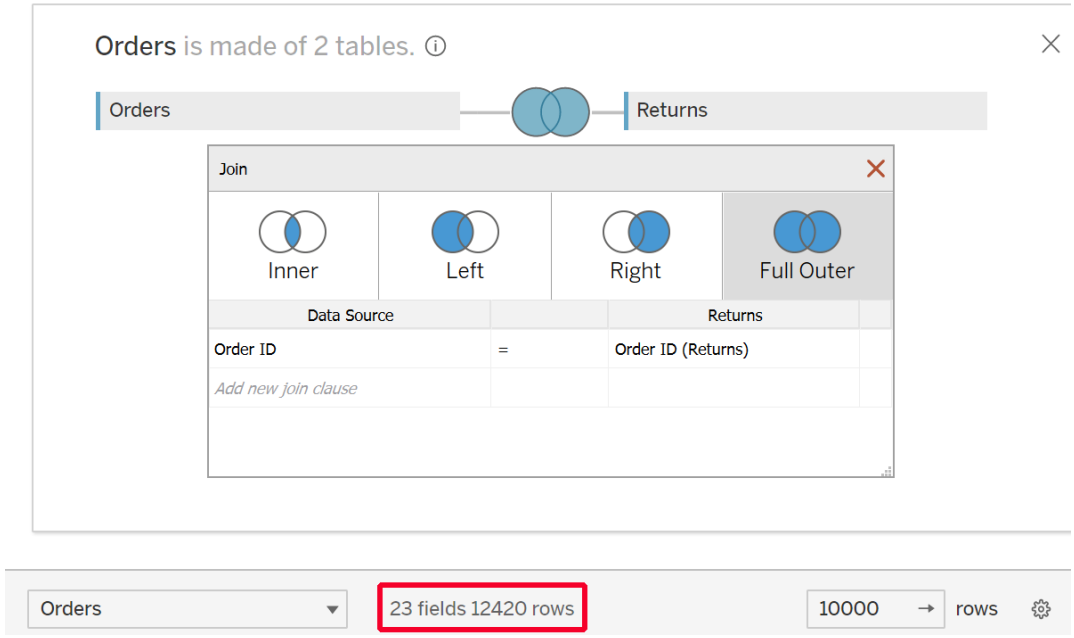


Figure 2.88: Full outer join interpretation

Based on the preceding join results, a left join is the join suitable for your requirement (i.e., understanding which orders were both fulfilled and returned) because, for an order to be returned, it first needs to exist in the **`Orders`** table. Thus, when you do a left join, you will get all the orders from the **`Orders`** table along with the **`Returns`** table.

Now, to identify how many orders you returned, you will look at the inner join output as, for an order to be returned, it needs to first exist. 3,226 orders are returned. This will not match the row count for the **`Returns`** table as an order ID can contain multiple products clubbed together. Try sorting the order IDs to validate this.

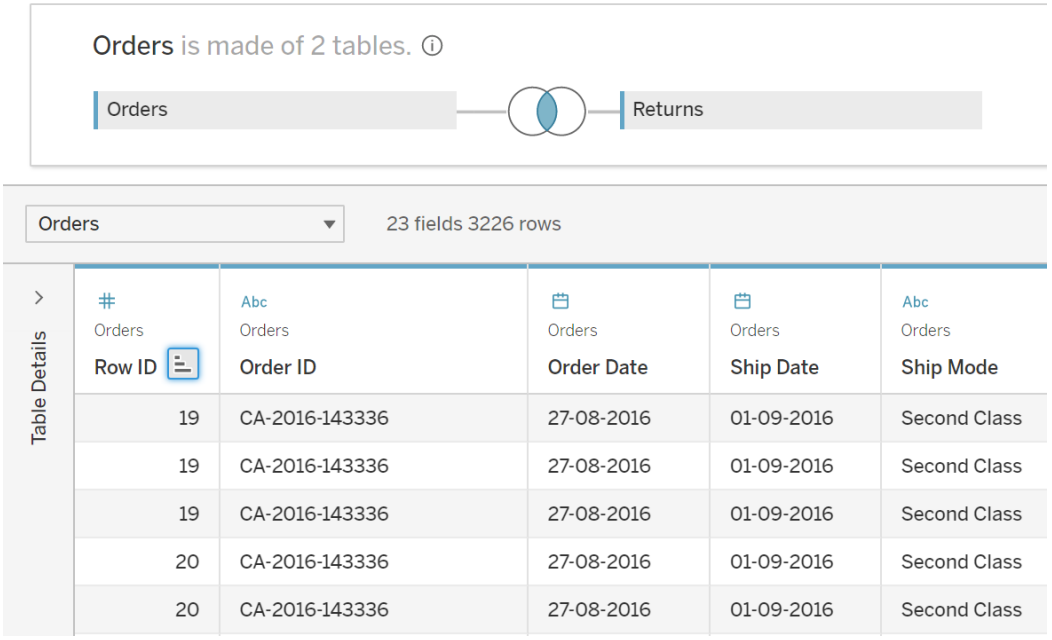


Figure 2.89: Validation of order IDs having multiple products

The final output should look like the following:

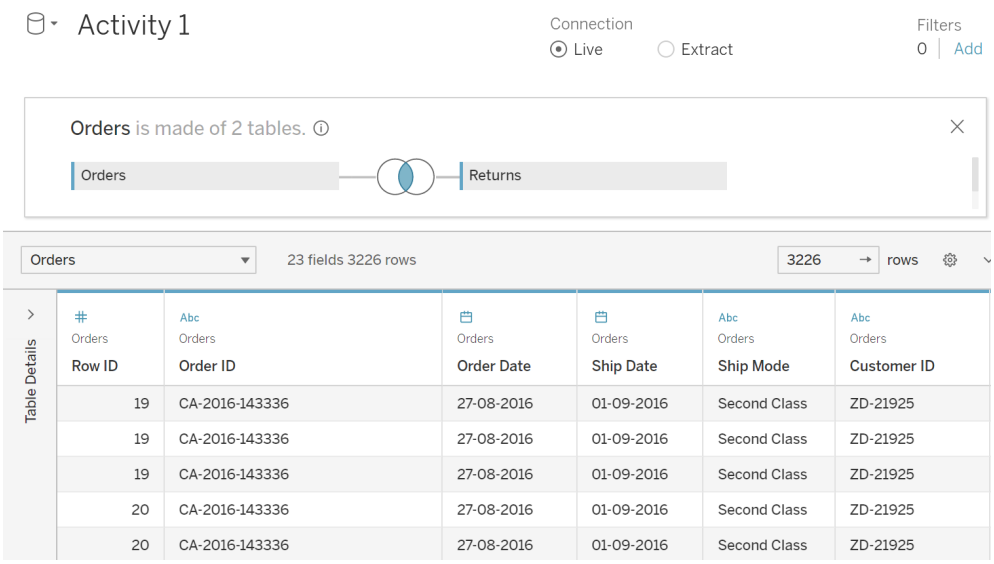


Figure 2.90: Activity output

In this activity, you strengthened your knowledge of various joins and their outputs. You also learned how to interpret the results by changing the join types.

ACTIVITY 2.02: PREPARING DATA FOR VISUALIZATION

Solution:

The following steps will help you complete this activity:

1. Create an extract from the data connection result of the last activity.

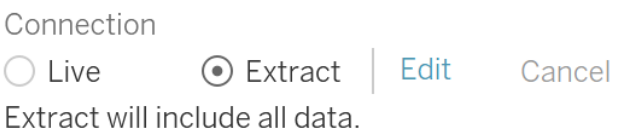


Figure 2.91: Creating an extract connection

2. Add a filter to the data to pull the **`Furniture`** and **`Office Supplies`** categories.

   For this, click on **`Filters`** | **`Add`** | **`Add`**.... In the **`Add Filter`** popup, select **`Furniture`** and **`Office Supplies`** for **`Category`**. Click on **`OK`** to add these filters to the extract.

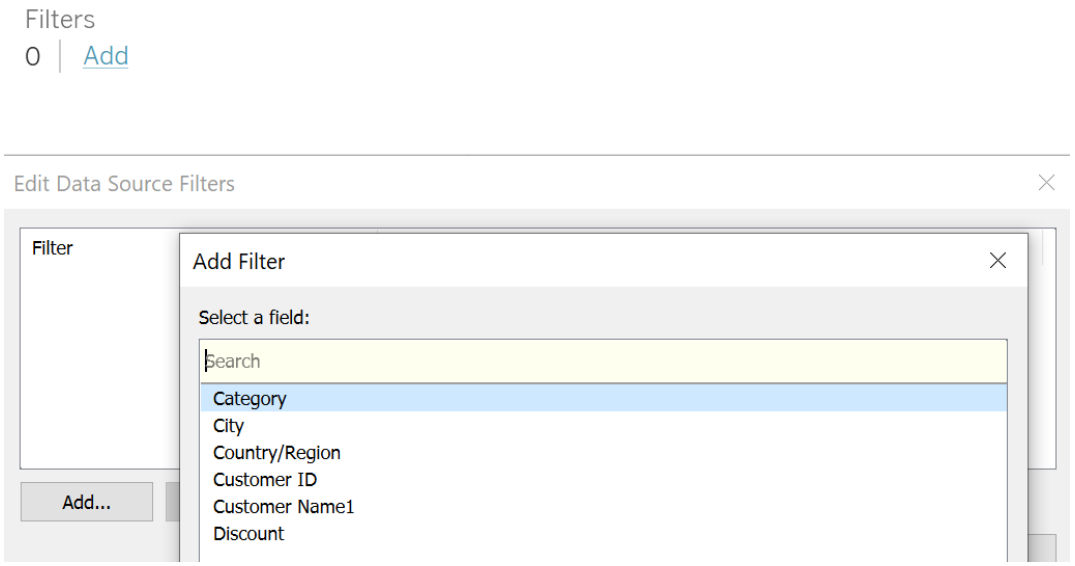


Figure 2.92: Adding the filtering condition

In the following figure, the required categories are being chosen:

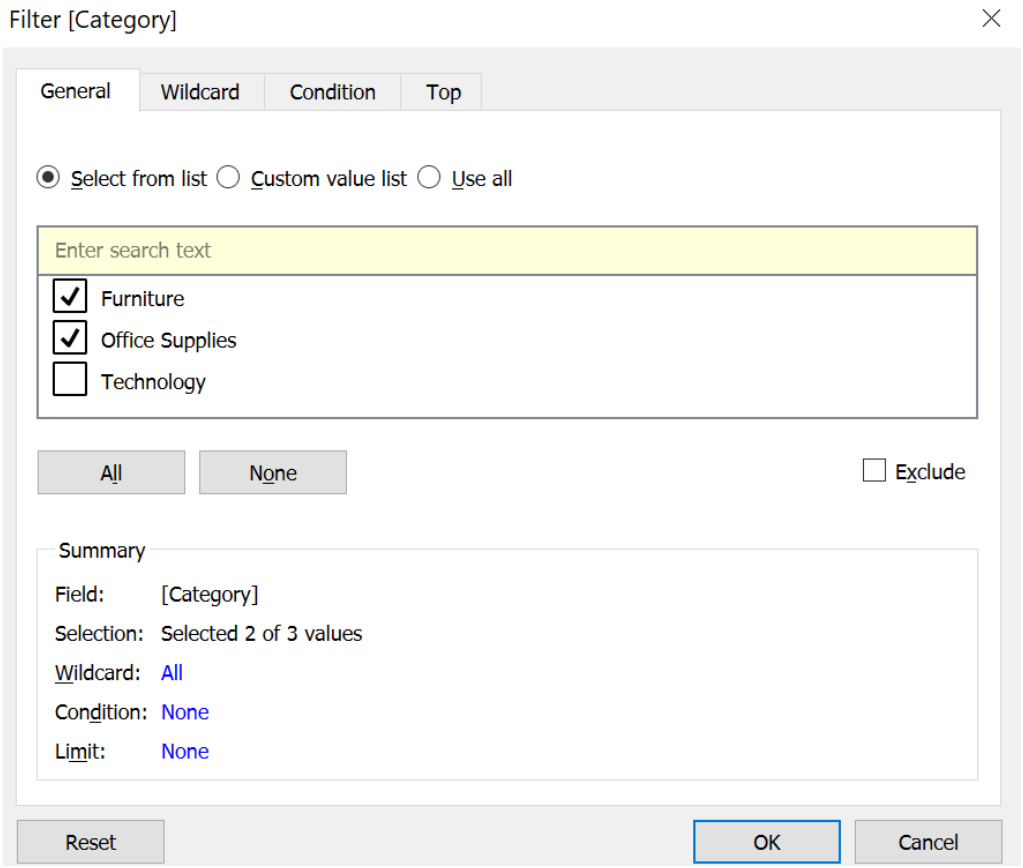


Figure 2.93: Choosing the required categories

The row count comes to 10,093, which means that technology products account for only 19% of your orders, which is a small portion.

3. Alias the **`Returns`** table columns to replace the nulls with blank. The columns for the Returns table are toward the end, indicated by their table names in the headers.

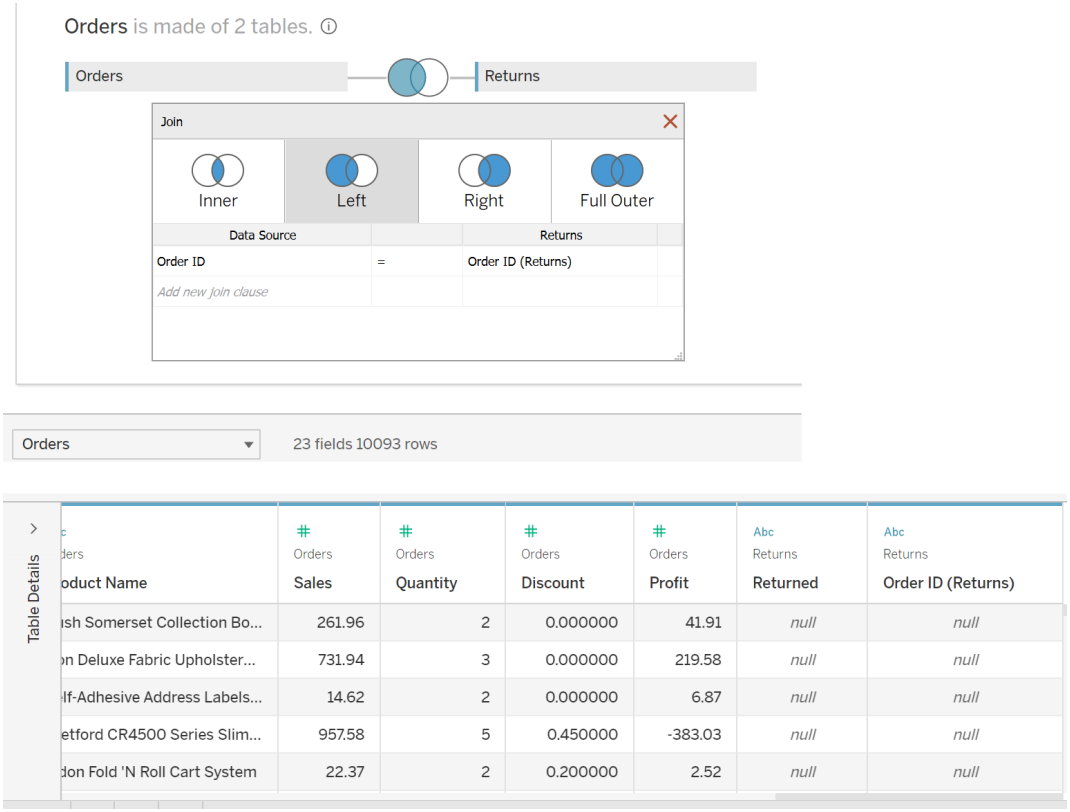


Figure 2.94: Alias transformation

4. To alias the columns, click on the dropdowns of the columns, which brings up the **`Edit Aliases`** popup.

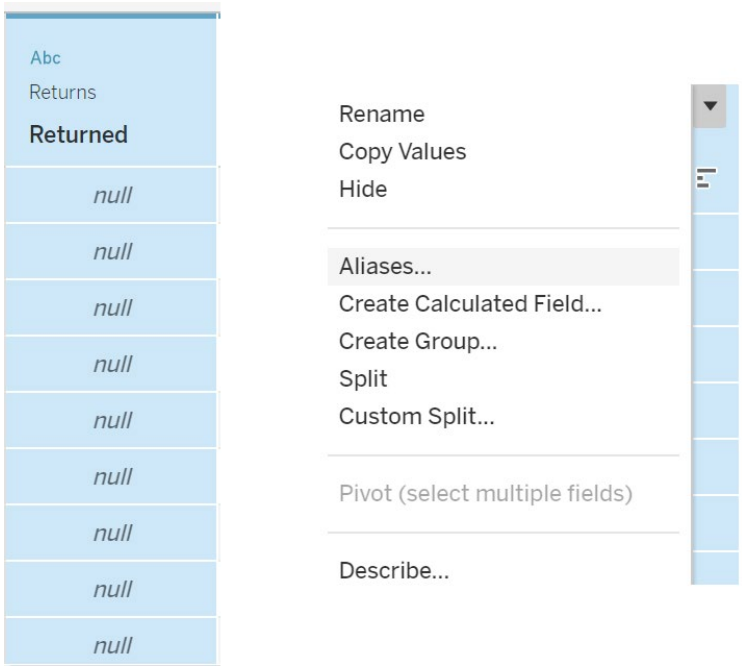


Figure 2.95: Setting the alias

5. Replace **`Null`** with a blank space and click **`OK`** to add it.

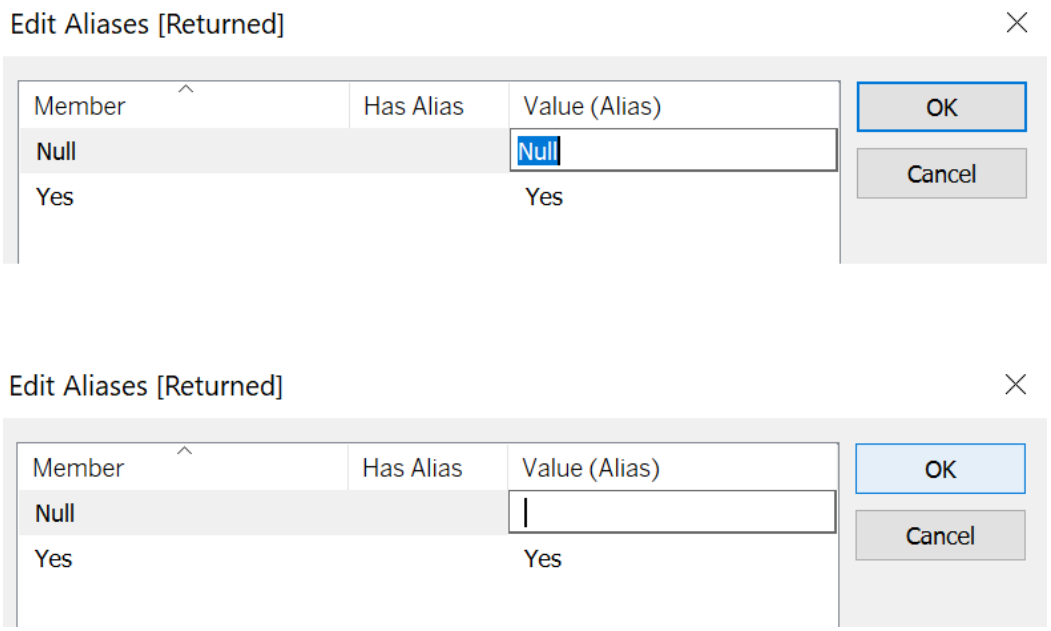


Figure 2.96: Replacing the Null value in the alias properties

6. Do the same as above for the **`Order ID`** column of the **`Returns`** table.

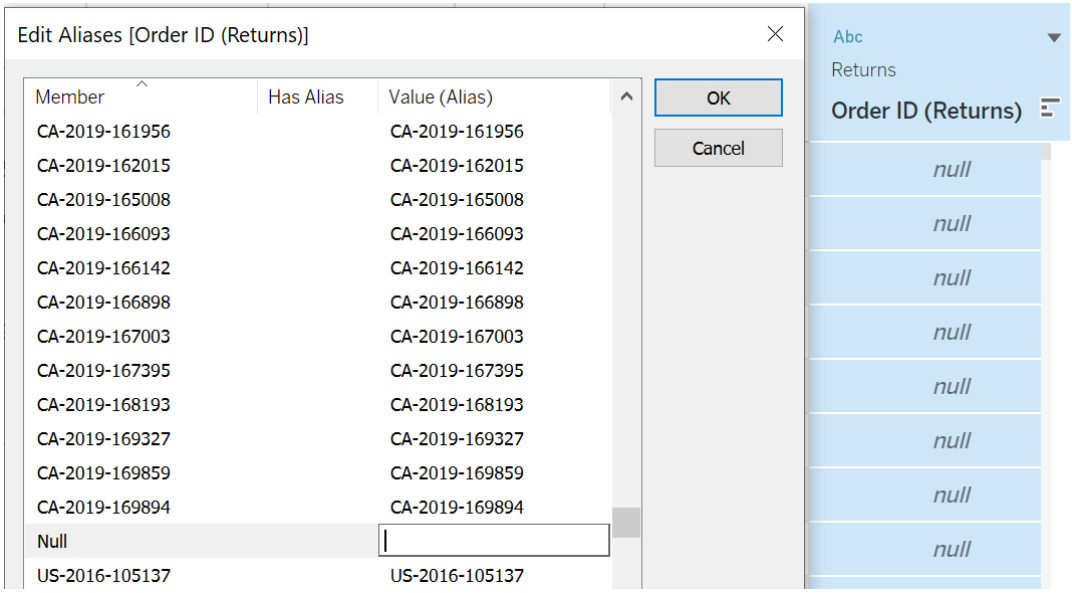


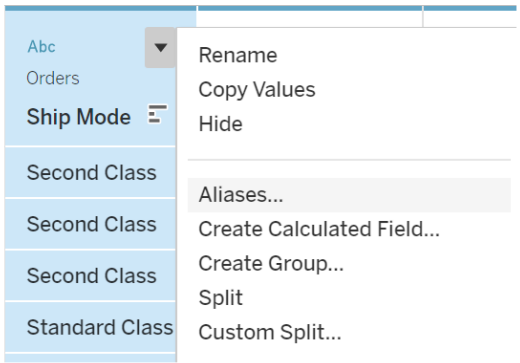
Figure 2.97: Adding the Order ID alias

Once completed, these two columns will look like this, which is much cleaner than showing nulls:

# Orders Sales	# Orders Quantity	# Orders Discount	# Orders Profit	Abc Returns Returned	Abc Returns Order ID (Returns)
261.96	2	0.000000	41.91		
731.94	3	0.000000	219.58		
14.62	2	0.000000	6.87		
957.58	5	0.450000	-383.03		
22.37	2	0.200000	2.52		
48.86	7	0.000000	14.17		

Figure 2.98: Nulls replaced with blanks in the alias

7. Alias the **`Ship Mode`** column to exclude the word Class.



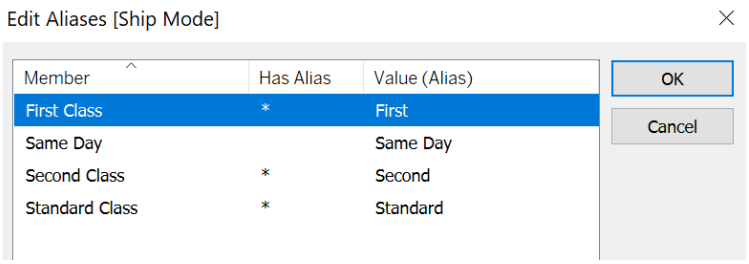


Figure 2.99: Alias for the Ship Mode column

8. Once done, the **`Ship Mode`** column will look like this:

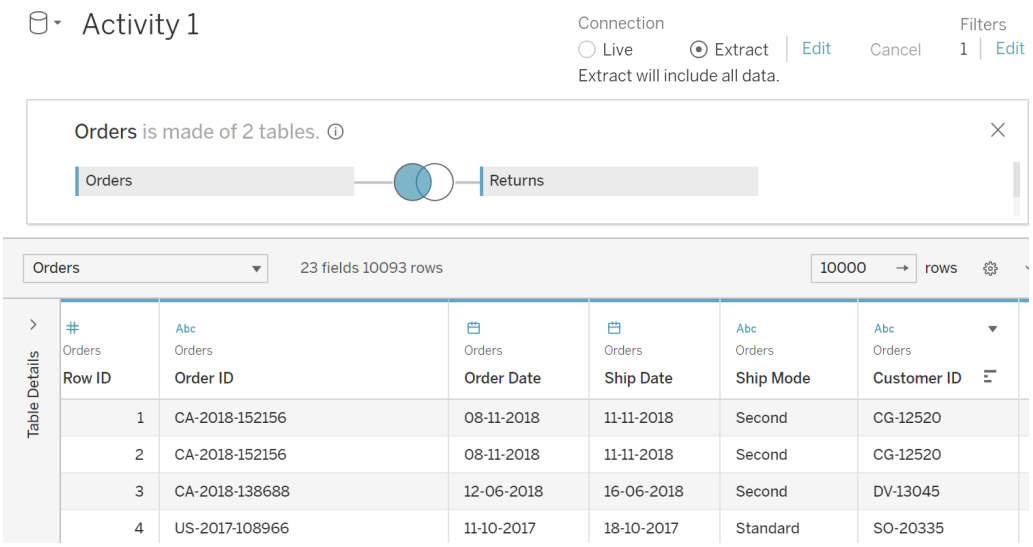


Figure 2.100: Activity output after completing the data transformation

In this activity, you learned how to extract the data. You also added filters for the **`Category`** column to just pull the selected categories. Many times, you will work on projects that require the data to be segregated at the beginning, such as regional data. These filters help you to achieve exactly this. You also transformed the data using aliases, making it much cleaner by removing repeated words and nulls.

CHAPTER 3: DATA PREPARATION: USING TABLEAU PREP

ACTIVITY 3.01: FINDING THE MONTH WITH THE HIGHEST ORDERS

Solution:

1. Open the dataset in Tableau Prep. Click on **`Connections`** and then **+**, and then select Microsoft Excel.

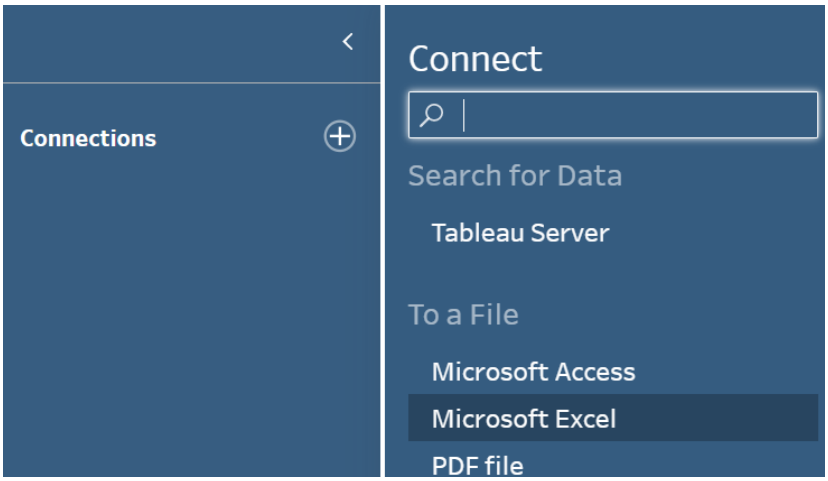


Figure 3.129: Connecting to Excel

2. Navigate to the location where you have downloaded the **`Activity File`** sheet and select it. You should then be able to see the various sheets present in this Excel sheet as follows:

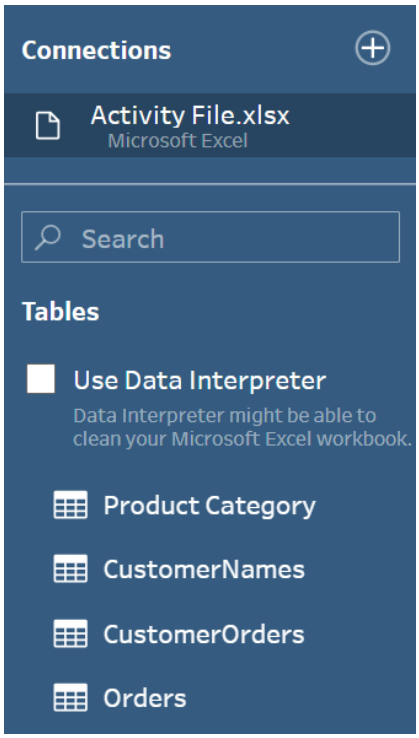


Figure 3.130: Excel sheets preview in Prep after file import

3. Now, add the **`Orders`** table to the flow. Drag it as follows to open the data input window.

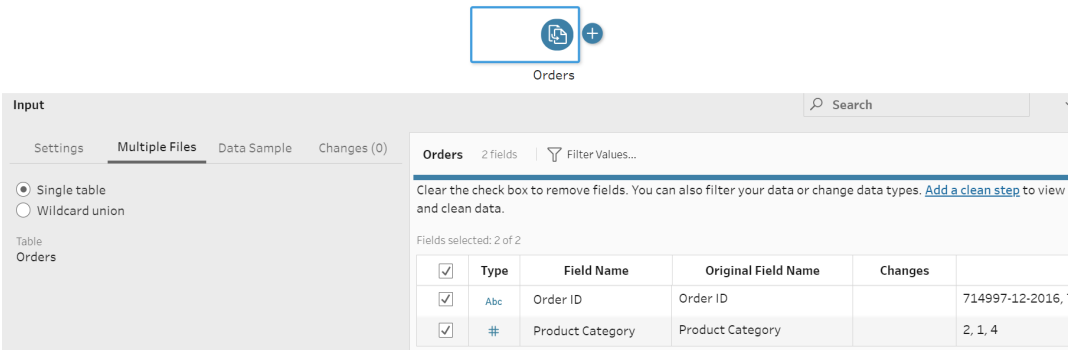


Figure 3.131: Adding Orders sheet in the workflow

You can see that it contains two fields.

4. Add a clean step for cleaning the data. You will observe that **`Order ID`** is a combination of an ID, month, and year:

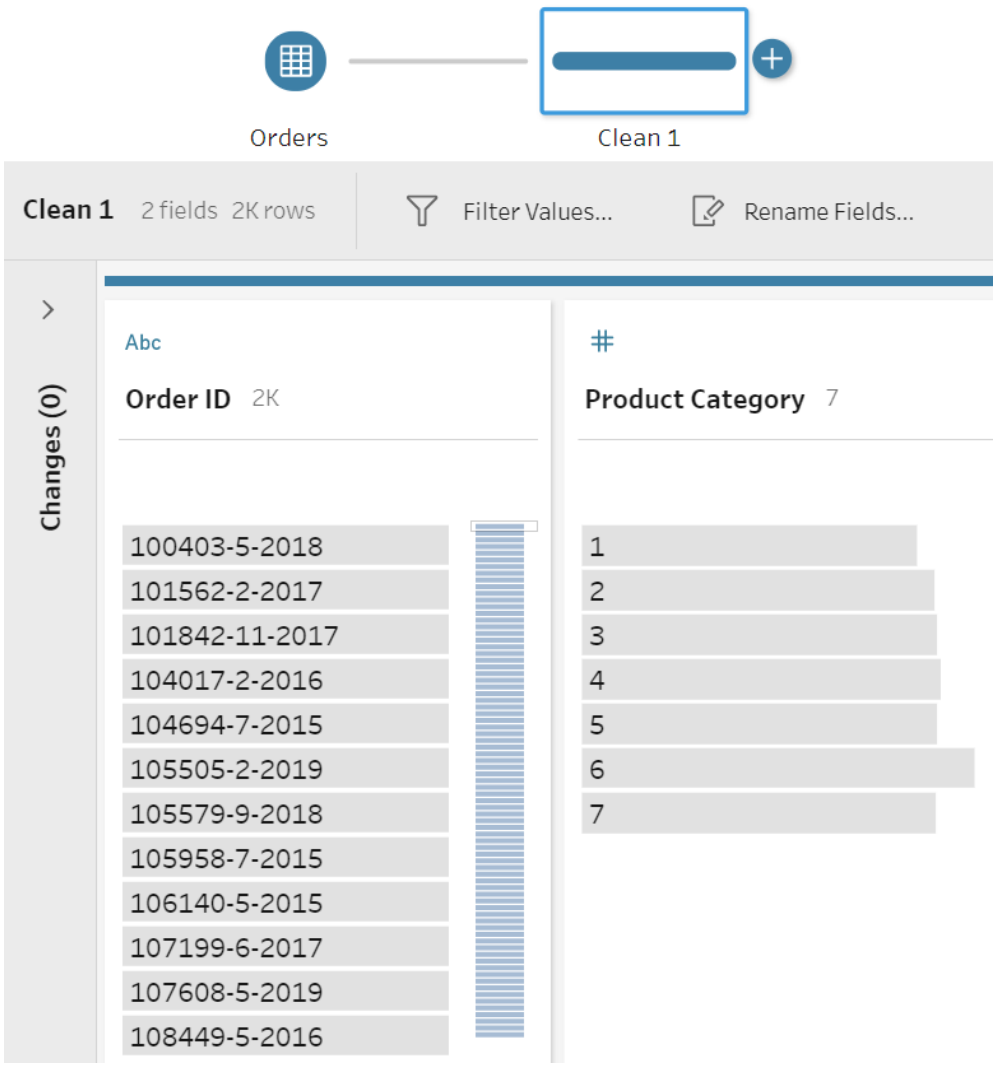


Figure 3.132: Order sheet preview

5. You need to find the month-year combination with the highest sales. To do that, you need to extract the order month and year values first – you can do that by applying **`Automatic Split`** on the **`Order ID`** column. Also rename the fields accordingly.

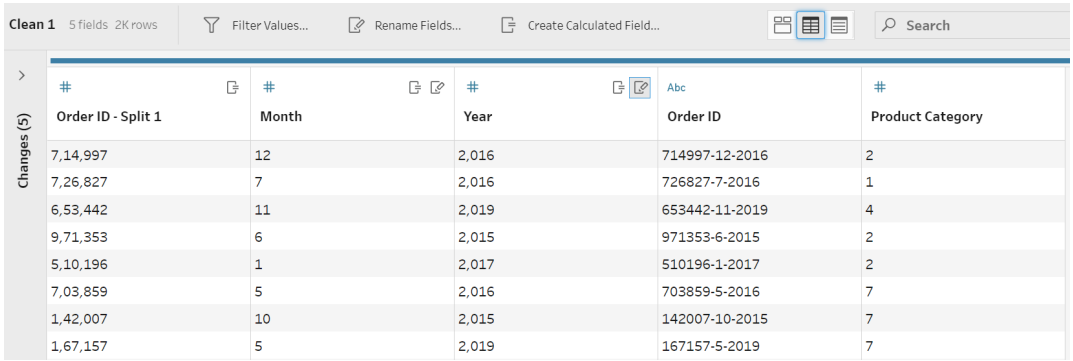


Figure 3.133: Splitting the Order ID

6. To find the month-year combination with the highest sales, you need to combine these columns. Create a calculated field and combine the month and the year as follows:

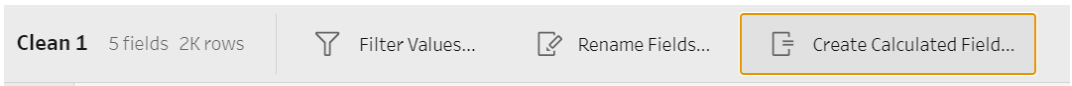


Figure 3.134: Creating a calculated field

7. Add the calculation and rename as follows:

```
STR([Month])+ "/" +STR([Year])
```

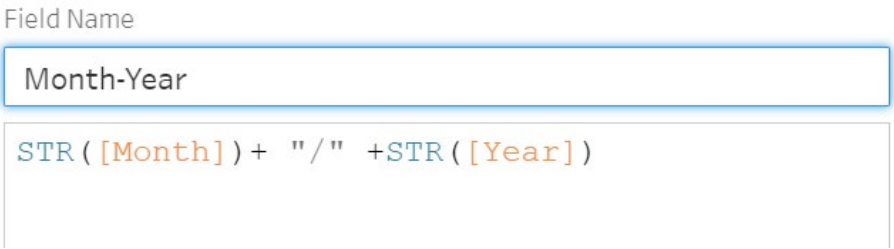


Figure 3.135: Syntax for month and year in the calculation editor

Note that you must wrap the month and year as strings to combine them. You will learn more about calculations later in the book. Click **Save** to add the calculation.

8. To see the highest orders, use the sort option (highlighted in Figure 3.136):

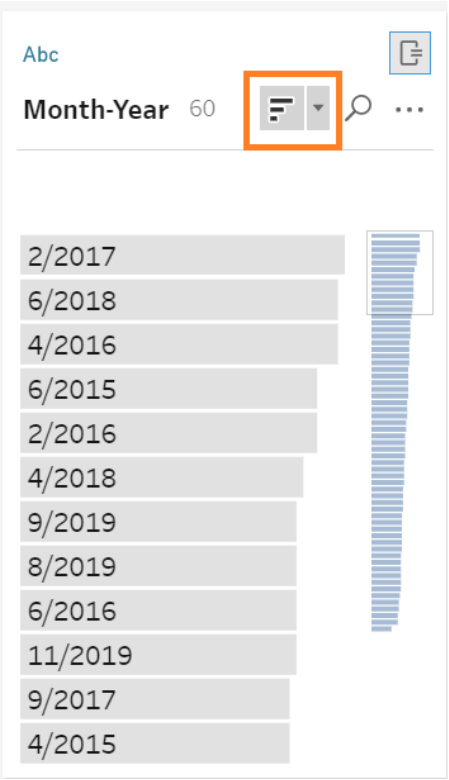


Figure 3.136: Identifying the month-year combination with highest orders

You can see that February 2017 has the highest number of orders.

9. Add the product category descriptions to the data. If you look at the **`Product Category`** column, you will see that it only contains values from 1 to 7. So, the type of this column should be a string and not a number. Let's change that:

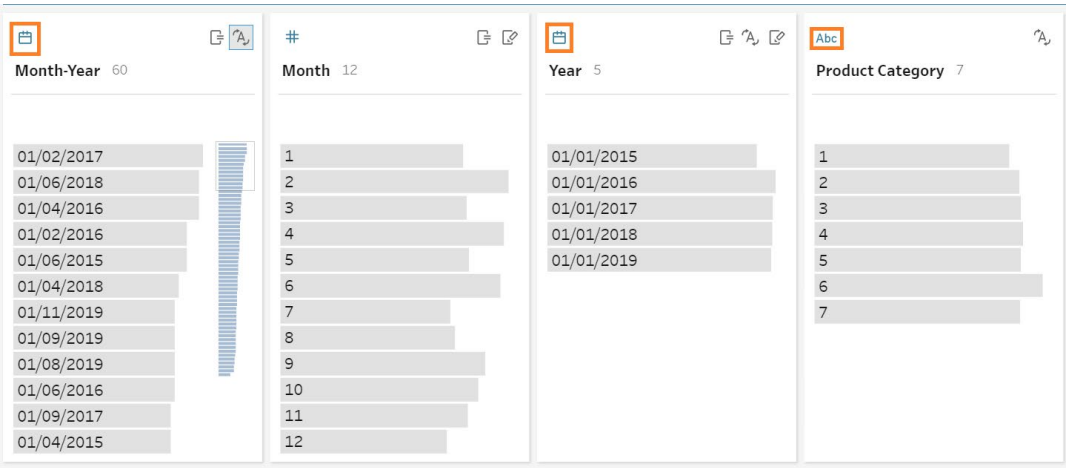


Figure 3.137: Changing the data type for Product Category

Also, change the data type for the Year and Month-Year columns to the Date type.

10. Now, add **`Product Category`** to the flow by dragging the Product Category sheet into the flow and adding the clean step as follows:

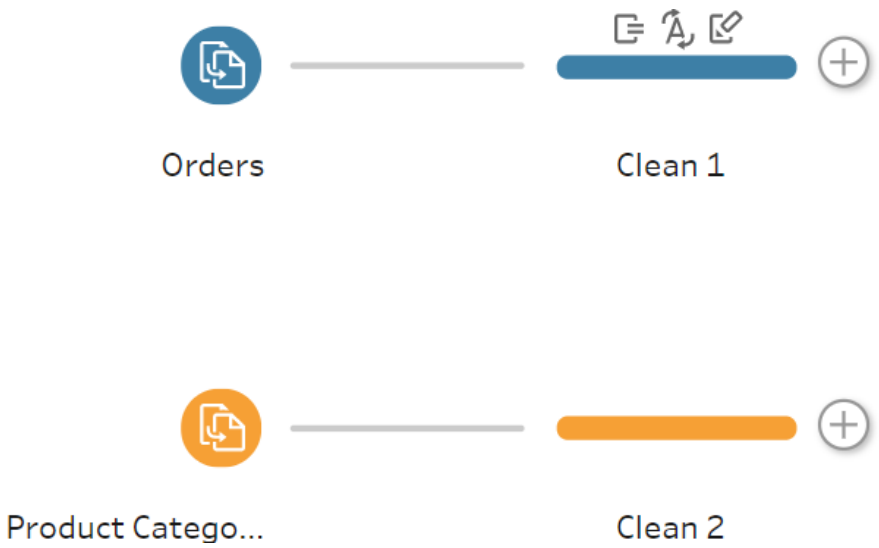


Figure 3.138: Adding the Product Category sheet to the workflow

11. If you look at the product category data, you will find that the **`ID`** field contains the same range from **`1`** to **`7`** and **`Type`** contains the category name, which needs to be added to the product data, as shown in the following screenshot:

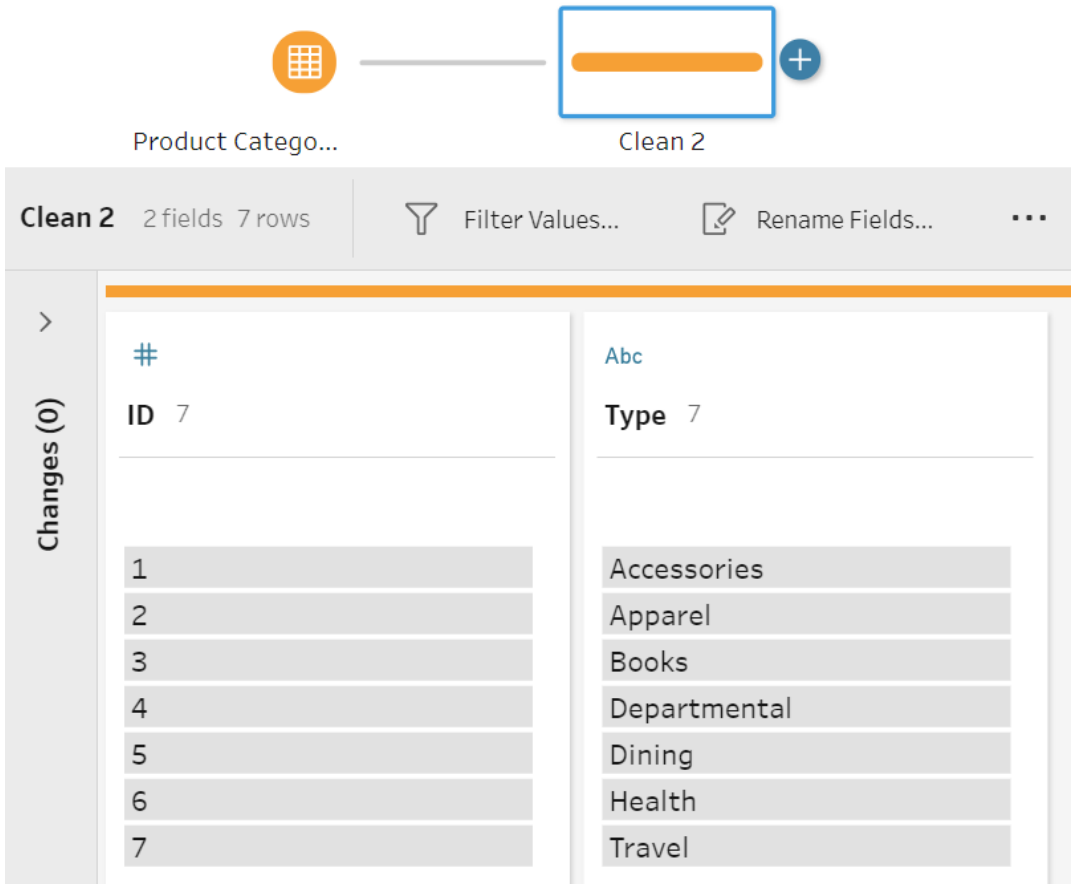


Figure 3.139: Product Category preview

12. Change the data type of the **`ID`** column to string, as it is not possible to perform a join on numbers.

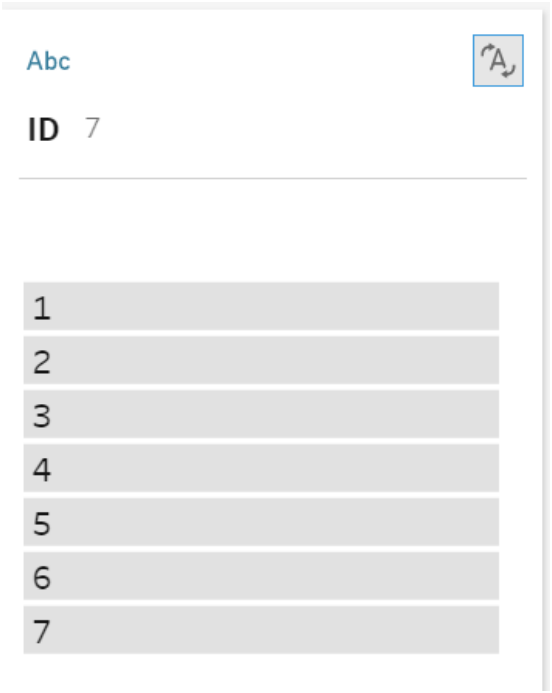


Figure 3.140: Changing the data type for the ID column

13. Now, drag **`Clean 2`** to join with **`Clean 1`**. An alert will pop up, as there are no automatically joined clauses, since the column names do not match.

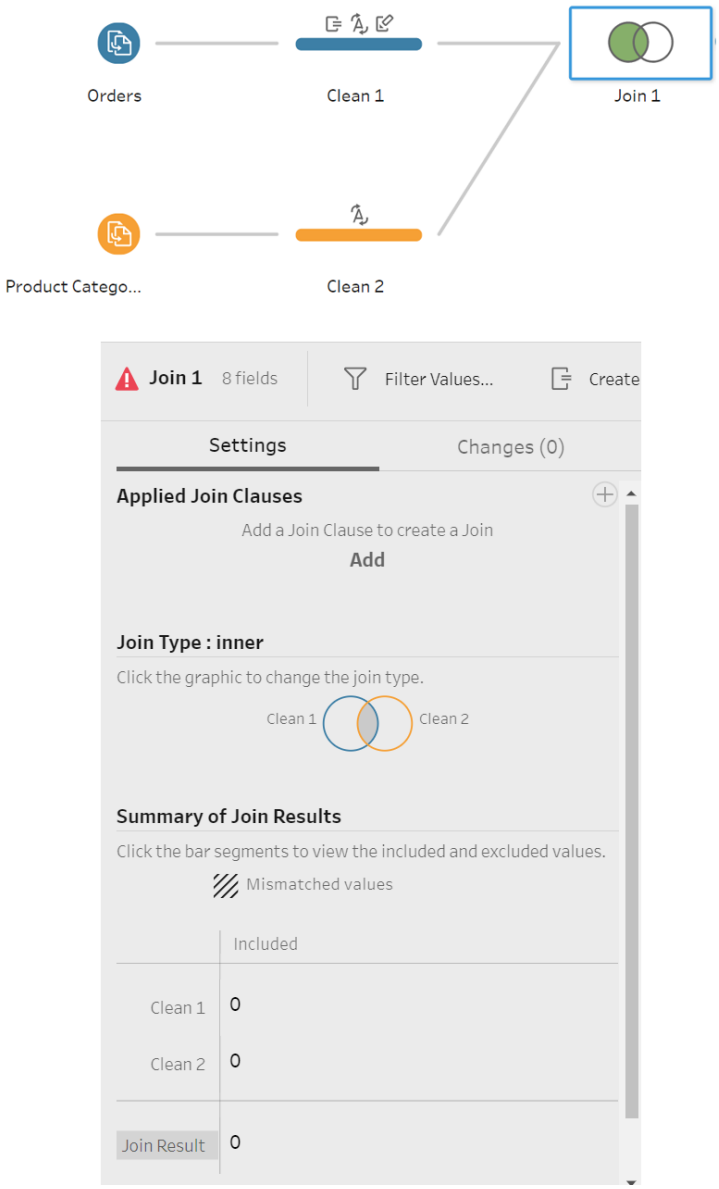


Figure 3.141: Joining the Orders and Product Category data in the workflow

14. Click on **Add** (which you can see in the preceding screenshot) to add a join clause for the **Product_Category** and **ID** columns.

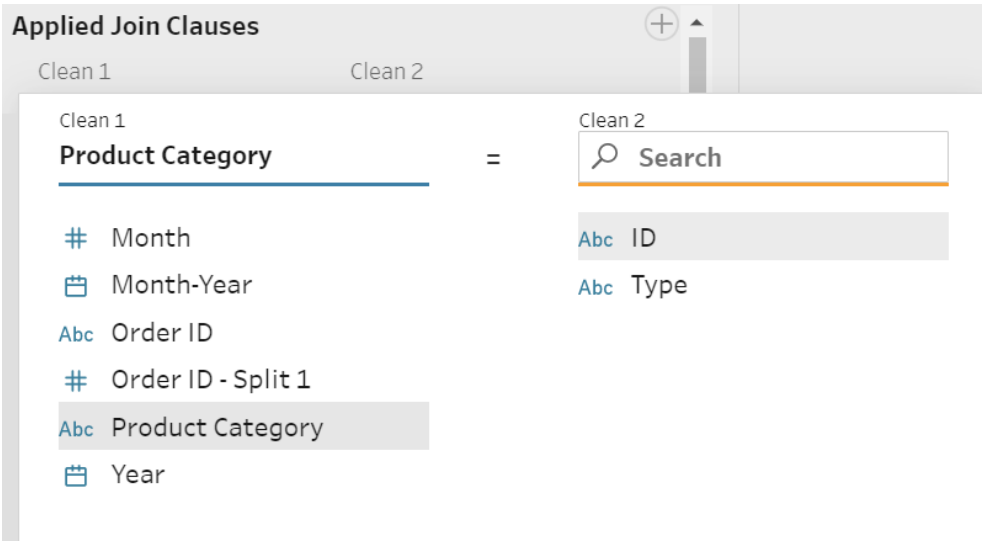


Figure 3.142: Modifying the join clause

15. Change the join to left join because you need all columns from the **Orders** table. Surprisingly, all the product categories match the **Orders**, so you will get around **2,002** records in the data with no mismatched values:

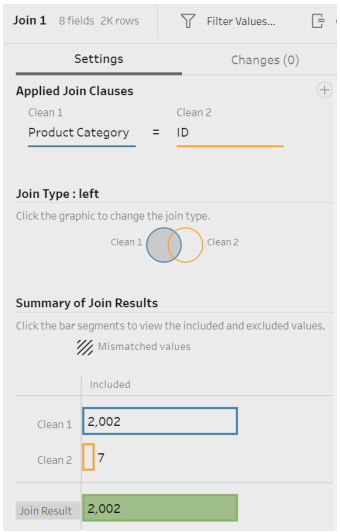
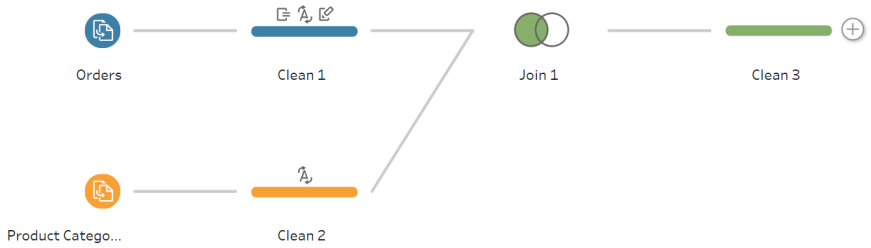


Figure 3.143: Join result preview

16. Finally, add a clean step to this join and preview the data.



Clean 3 8 fields 2K rows | Filter Values... | Rename Fields... | Create Calculated Field... | Search

Changes (0)

Month-Year	Month	Year	Product Category	Order ID - Split 1
01/12/2016	12	01/01/2016	2	7,14,997
01/07/2016	7	01/01/2016	1	7,26,827
01/11/2019	11	01/01/2019	4	6,53,442
01/06/2015	6	01/01/2015	2	9,71,353
01/01/2017	1	01/01/2017	2	5,10,196
01/05/2016	5	01/01/2016	7	7,03,859

Figure 3.144: Preview of the combined data

17. You can see that **`ID`** and **`Product Category`** columns are redundant and add no value. Remove these columns and rename the **`Type`** column **`Product Category`**:

Clean 3 8 fields | Filter Values... | Rename Fields... | Create Calculated Field...

Changes (0)

Type	Field Name	Changes
	Month-Year	
	Month	
	Year	
Abc	Product Category	
	Order ID - Split 1	
Abc	Order ID	
Abc	ID	
Abc	Type	

Figure 3.145: Excluding duplicate columns

After cleaning, you will have six columns in the data:

Type	Field Name
📅	Month-Year
#	Month
📅	Year
Abc	Order ID
Abc	Product Category

Figure 3.146: Final columns structure

18. By sorting on the new **`Product Category`** column, you will see that **`Dining`** has the highest orders for the **`Product`** category.

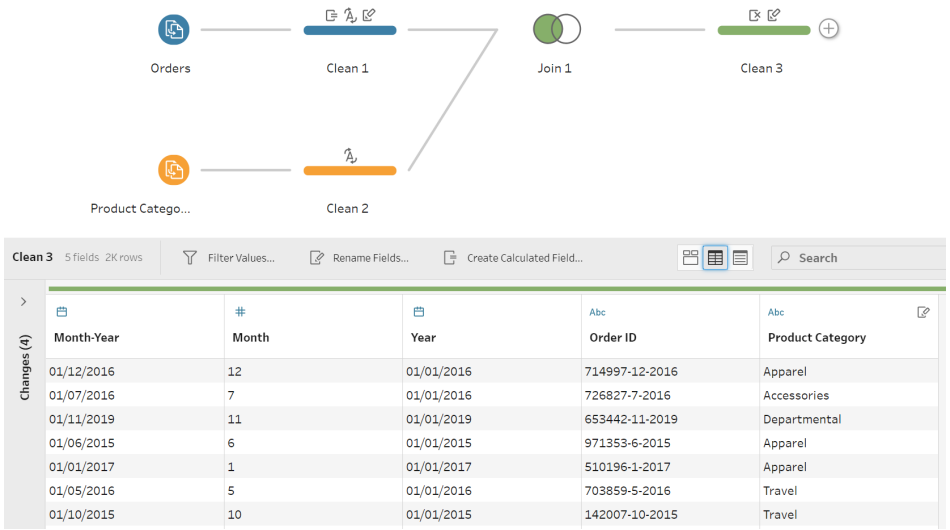


Figure 3.147: Activity 1 workflow output

In this activity, you analyzed data to draw certain insights by answering queries based on the data profile of the resulting dataset.

ACTIVITY 3.02: DATA TRANSFORMATION

Solution:

Continuing from the previous activity, add the **`CustomerNames`** table in the workflow and a clean step:

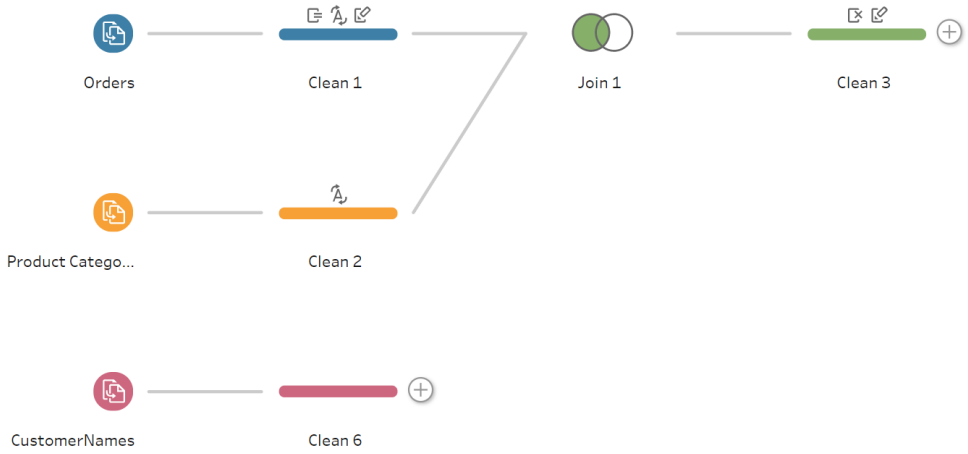


Figure 3.148: Adding the CustomerNames sheet to Activity 1 output workflow

19. Create a calculated field to combine **`First name`** and **`Last name`** into **`Customer Name`** and remove the original columns:

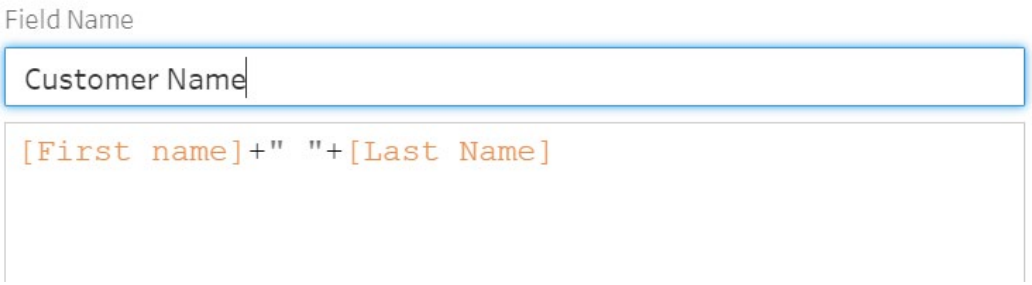
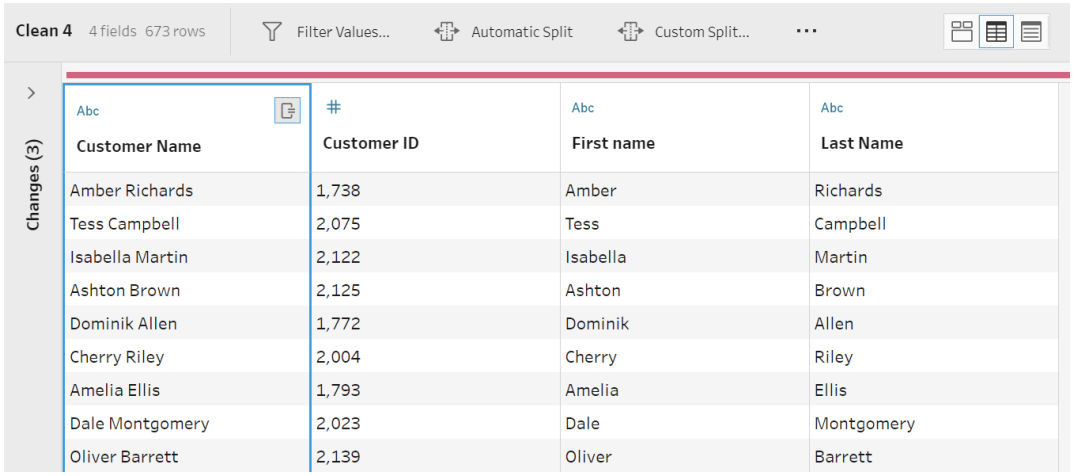


Figure 3.149: Merging the first and last names of customers calculation

This should result in a combined field:



Abc Customer Name	# Customer ID	Abc First name	Abc Last Name
Amber Richards	1,738	Amber	Richards
Tess Campbell	2,075	Tess	Campbell
Isabella Martin	2,122	Isabella	Martin
Ashton Brown	2,125	Ashton	Brown
Dominik Allen	1,772	Dominik	Allen
Cherry Riley	2,004	Cherry	Riley
Amelia Ellis	1,793	Amelia	Ellis
Dale Montgomery	2,023	Dale	Montgomery
Oliver Barrett	2,139	Oliver	Barrett

Figure 3.150: Customer name preview

20. Remove the **`First Name`** and **`Last Name`** columns. Also, change the data type of **`Customer ID`** to string.

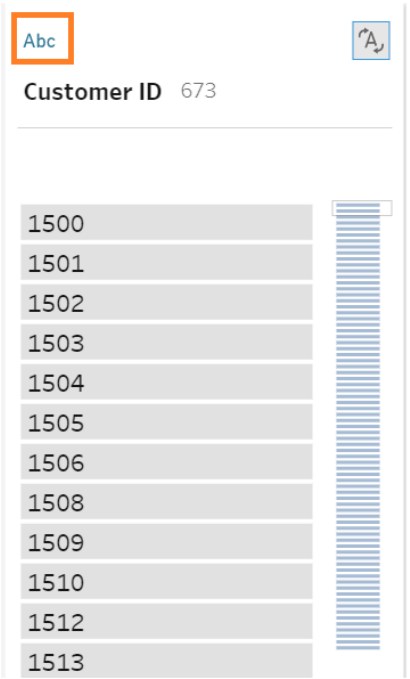


Figure 3.151: Changing the CustomerID column data type from numeric to string

21. Now there is an issue with the data as the customers' first names and last names have been interchanged and given different IDs. For example, Robert Cooper and Cooper Robert are the same person but have different customer IDs. To fix this, go to **`Group Values – Common characters`**, to identify and group these customer names:

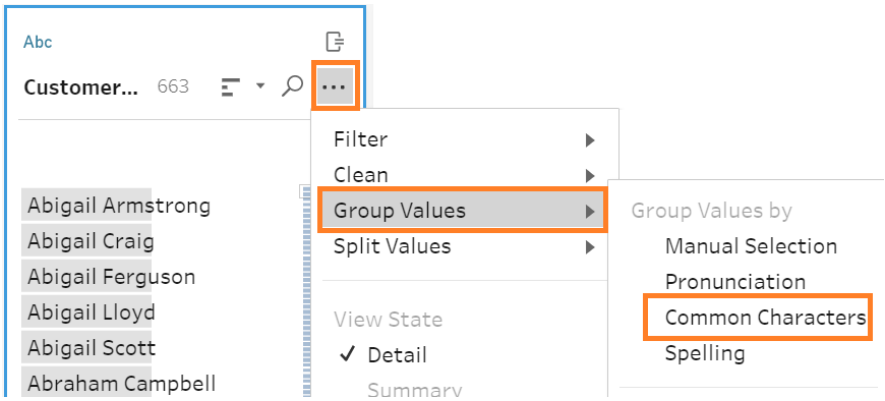


Figure 3.152: Cleaning the customer names to identify duplicate entries

Note that the Common Characters method is used here so that, if required, you can perform additional manual selection to improve the grouping.

22. Next, scroll through the created groups. As you can see, the data looks good. Click on **`Done`** to add the groups:

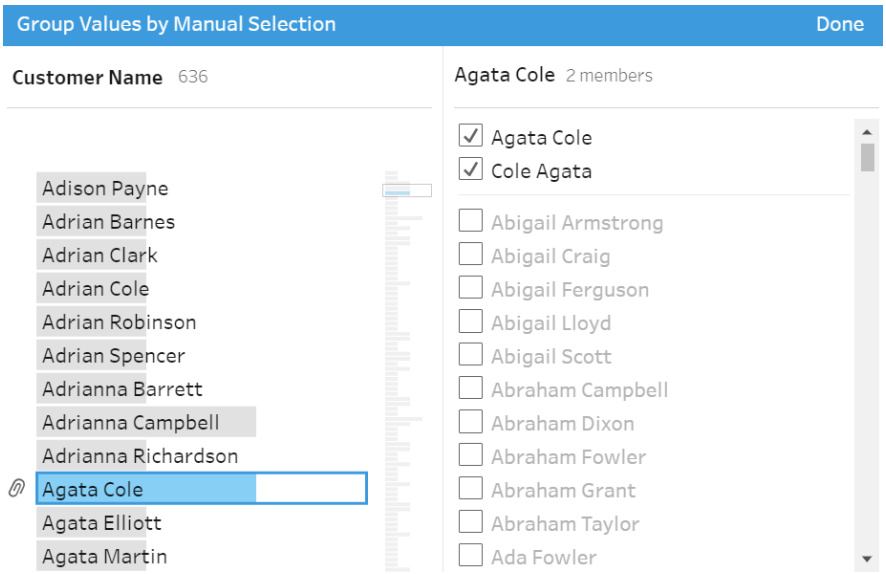
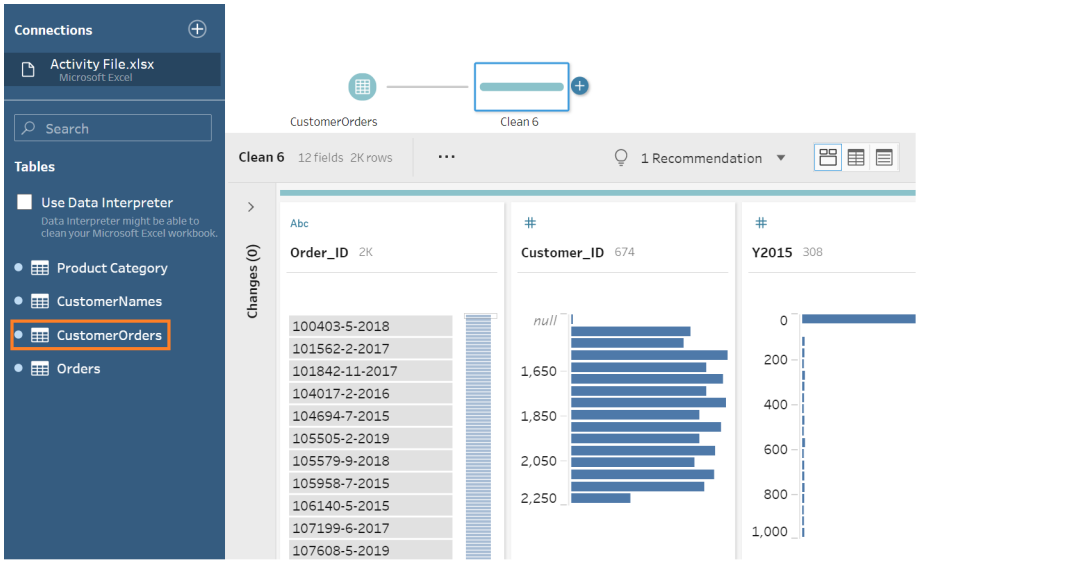


Figure 3.153: Manually selecting duplicate customer names

23. Add the **`CustomerOrders`** table in the flow along with a clean step. The data contains the order IDs along with the sales and quantities saved in horizontal format, as follows:



Order_ID	Customer_ID	Y2015	Y2016	Y2017	Y2018	Y2019	Q15	Q16	Q17	Q18	Q19
714997-12-2016	1,738	0	6,316	0	0	0	0	93	0	0	0
726827-7-2016	2,075	0	0	0	0	0	0	22	0	0	0
653442-11-2019	2,122	0	0	0	0	60,847	0	0	0	0	845

Figure 3.154: CustomerOrders data preview

24. Add a pivot to the clean step. Here, you need two pivots – one for **`Sales`** and the other for **`Quantity`**. *Ctrl* + click to multi-select and drag the columns. Add these columns, as shown in the following screenshot, to the **`Sales`** column, and rename the column:

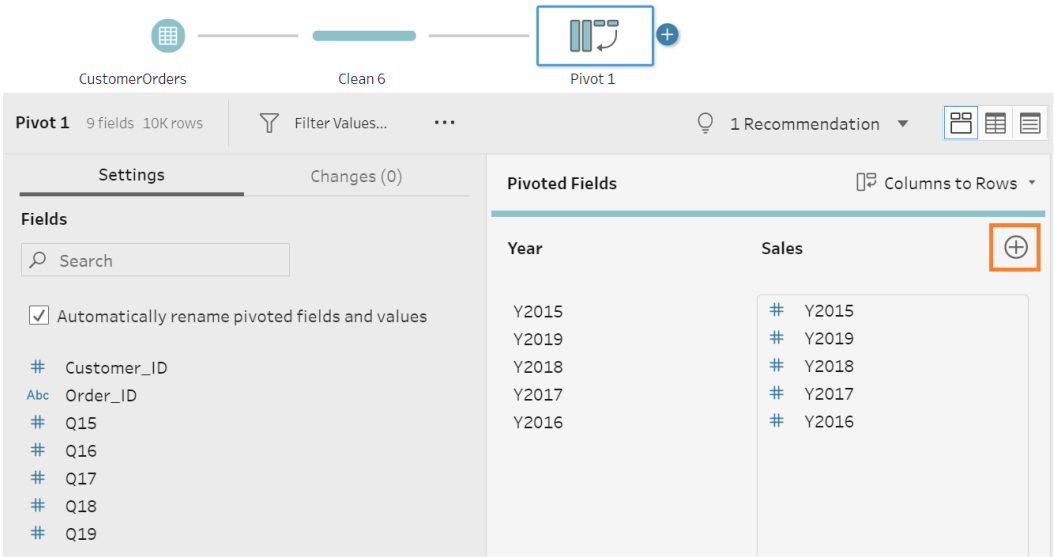


Figure 3.155: Adding pivot columns to transform year and sales values

25. Repeat the previous step for **`Quantity`** by adding another pivot using the **+** icon and then rename the column. The data preview should now show **`Sales`** and **`Quantity`** as two separate columns.

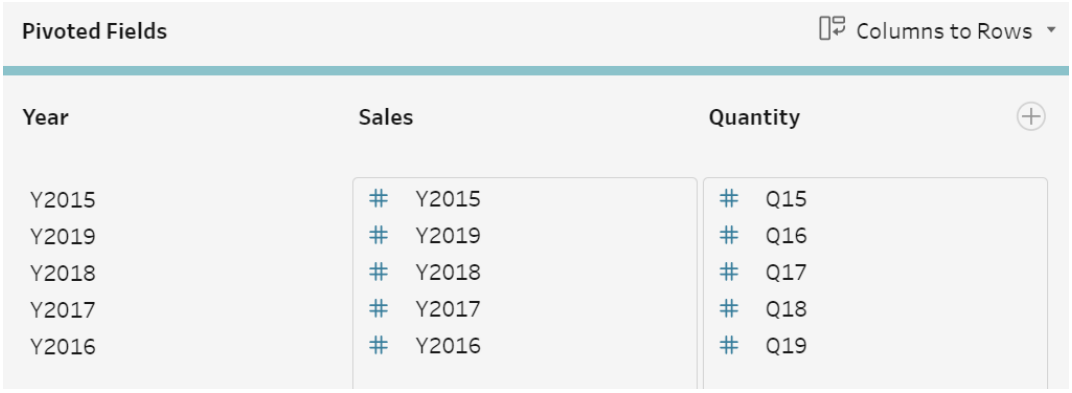


Figure 3.156: Adding a pivot column to transform Quantity values

26. There is no need for the **`Year`** column, and you need to change the data type of **`Customer_ID`** to string. Do so by switching to list view and making the changes, as shown in the following screenshot:

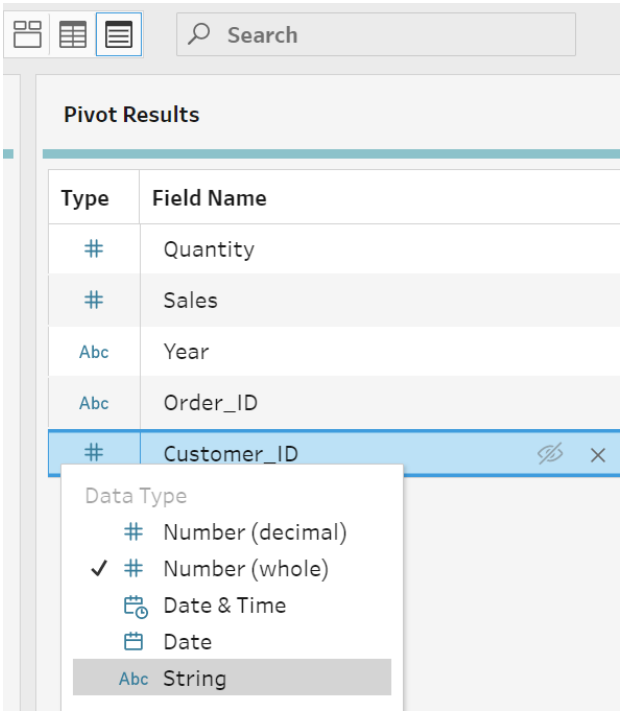


Figure 3.157: Switching to list view

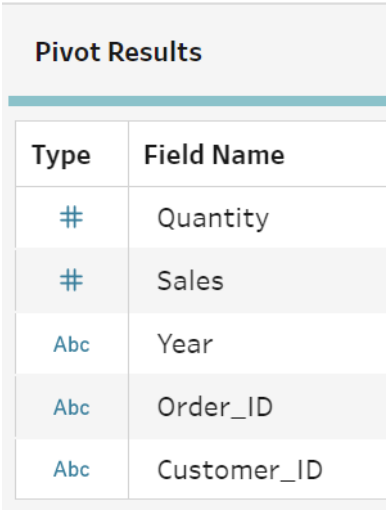


Figure 3.158: Preview of pivot results

27. The data is now ready. Join this with the **`CustomerNames`** table to get the customer names in the data. Drag the pivot on **`clean 4`** to join these two tables. Prep will join the data on the **`Customer ID`** column.

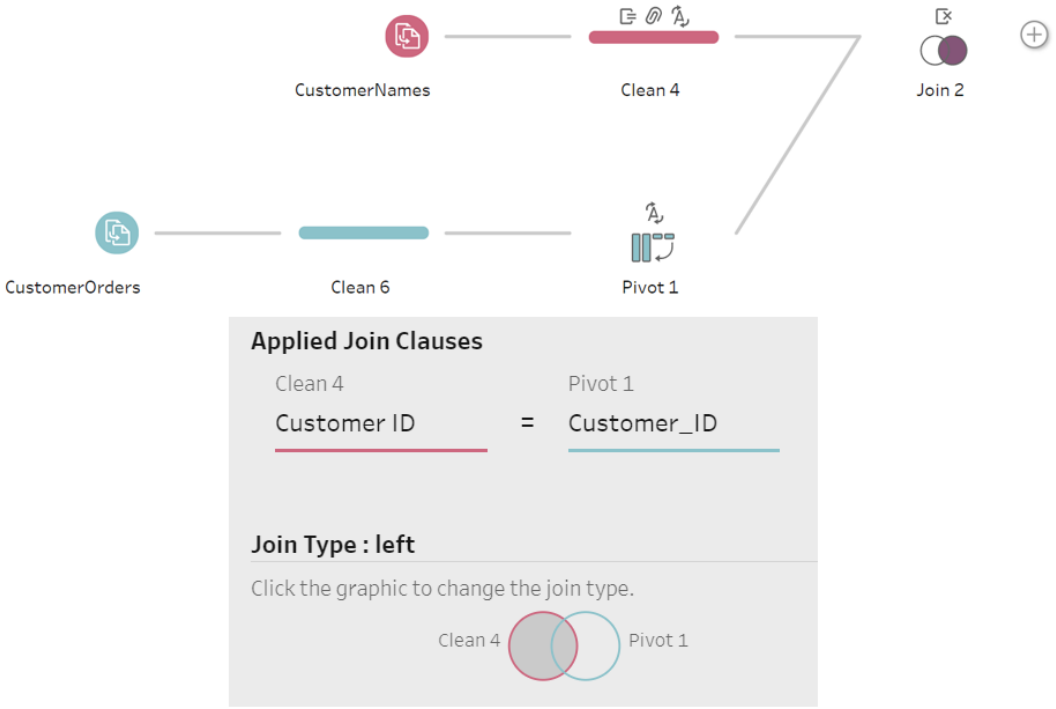


Figure 3.159: Joining customer names with customer orders

28. You will see that there are some null **`Customer ID`** values. This is because some orders do not have a customer ID. For these instances, you need all the data from the **`CustomerOrders`** table. So, change the join type to right, indicated by the color of the table:

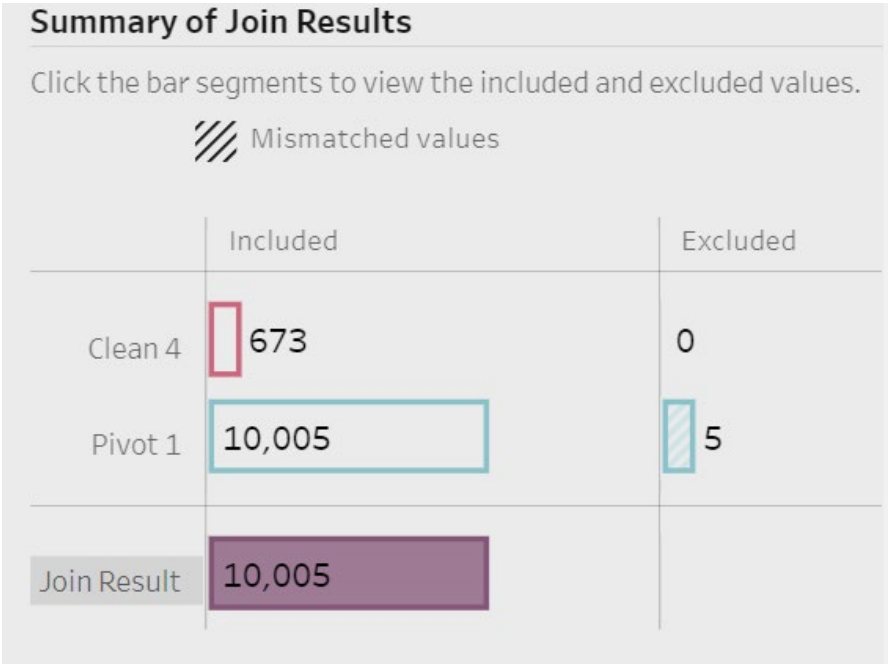


Figure 3.160: Missing customer names in customer orders

29. You can see that there are 5 orders with missing **`Customer_ID`** values. As instructed in *Exercise 3.1*, you should have removed the **`First name`** & **`Last Name`** columns from the **`CustomerNames`** table, as you do not need duplicate columns. This can be done by switching to list view and removing this column:

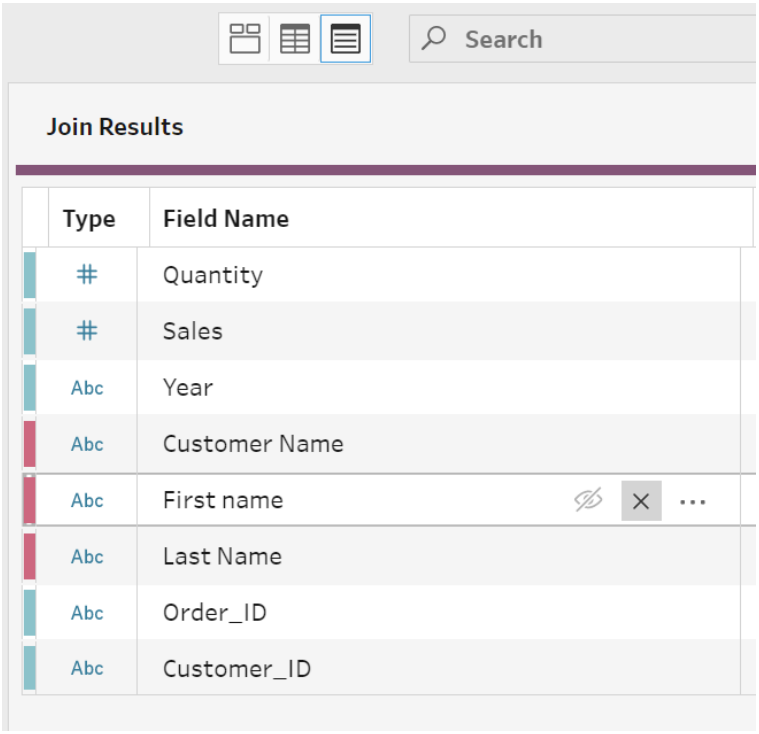


Figure 3.161: Excluding duplicate columns from the join results

30. Join this output with the output of the previous activity, that is, **`Clean 3`** and **`Join 2`**:

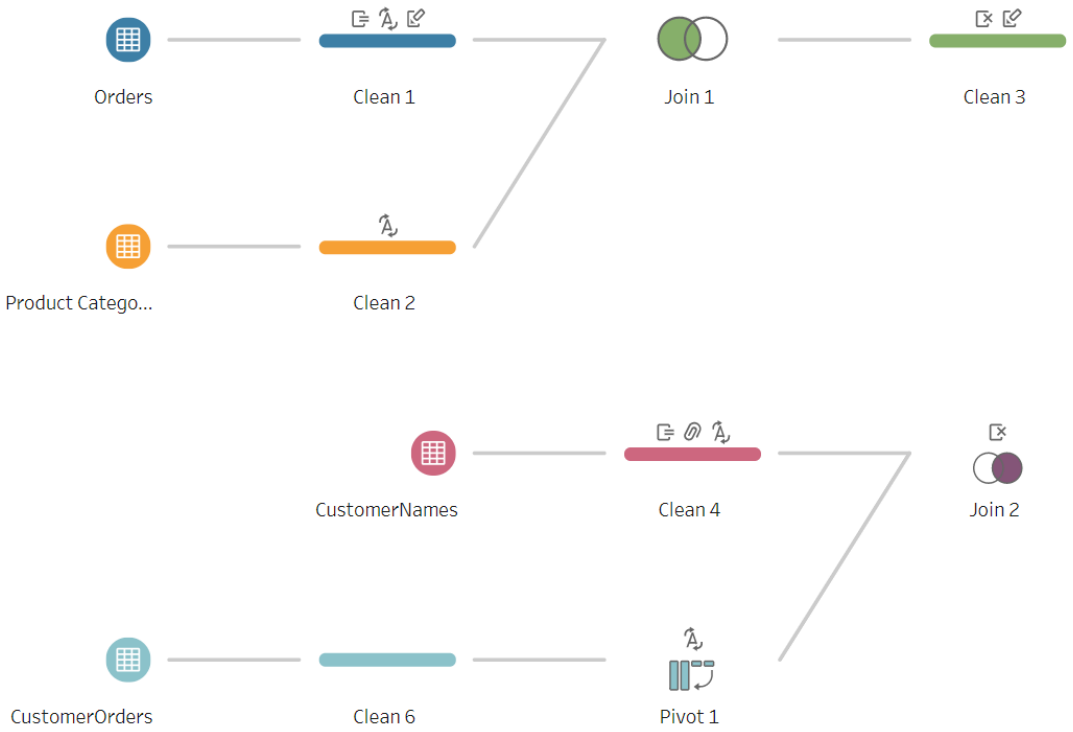


Figure 3.162: Workflow for the Order and Customer tables

You can join these two, as shown in the following screenshot:

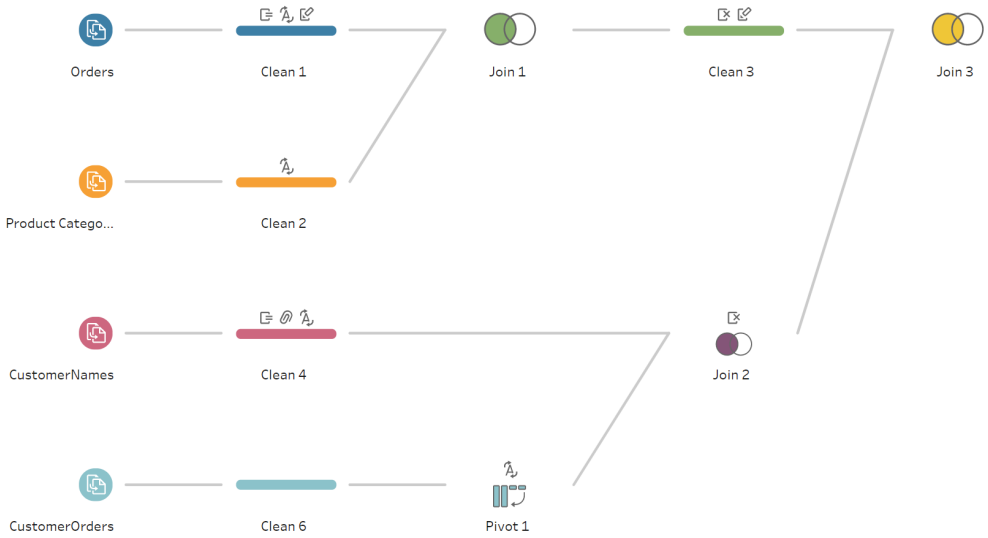


Figure 3.163: Joining customer and orders workflows

31. Prep will use **`Order ID`** to join these steps as the joining clause. Use a left join, as you need all the records from the **`Customer Names`** table and their matching orders information. You can preview this data using the data grid view.

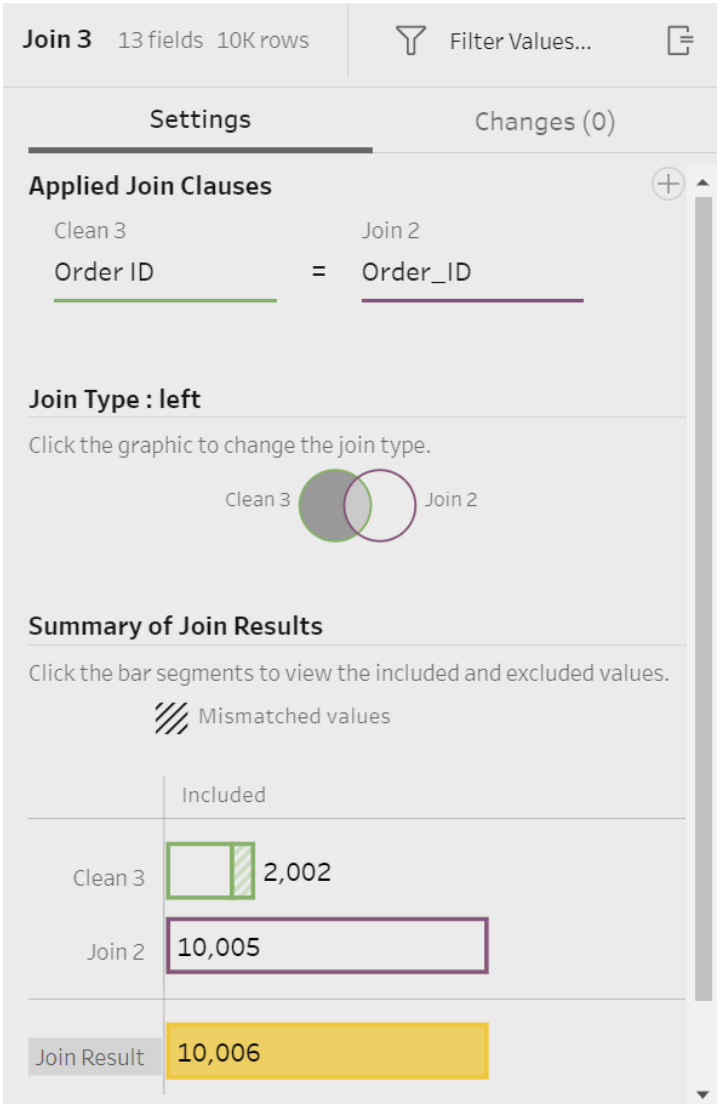


Figure 3.164: Modifying the join type to a left join

32. Now that you have completed the data transformation steps, you can identify the top five customers based on the number of orders. To do that, add a clean step to preview the data. Sort the **`Customer Name`** column in descending order to identify the high-value customers.

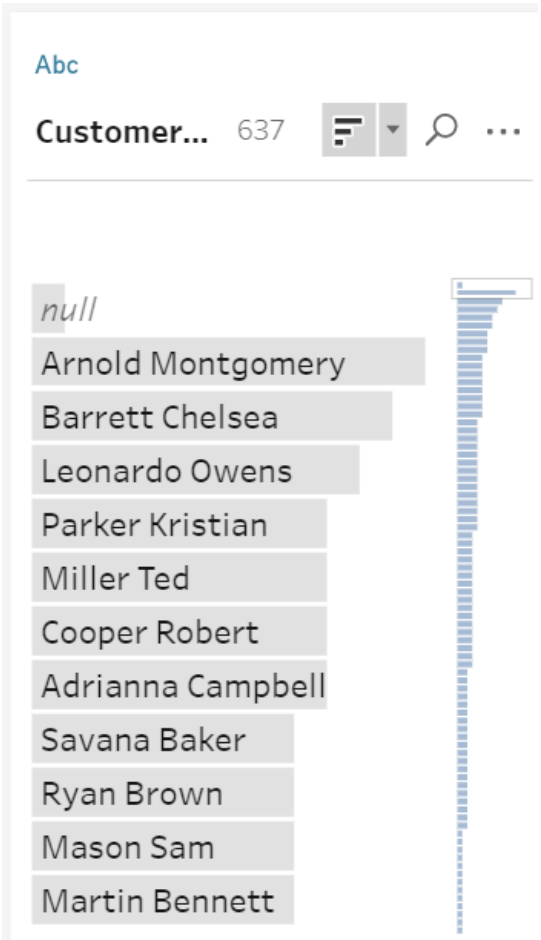


Figure 3.165: Identifying high-value customers in the data preview using the sort option

33. Now that you have identified the top five high-value customers, export this data into a CSV file. Add an output step and click **`Run Flow`**:

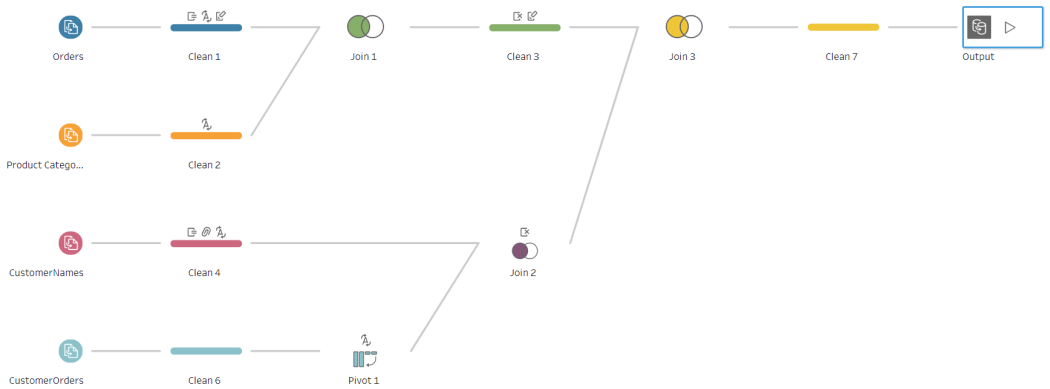


Figure 3.166: Adding an output step to export the data

The exported data will look like the following. As you can see, it contains a unified dataset based on order, products, and customer information:

Quantity	Sales	Customer Name	Month-Year	Month	Year	Order ID	Product Category	Order_ID	Customer_ID
0	0	Amber Richards	01-12-2016	12	01-01-2016	714997-12-2016	Apparel	714997-12-2016	1738
0	0	Tess Campbell	01-07-2016	7	01-01-2016	726827-7-2016	Accessories	726827-7-2016	2075
0	0	Isabella Martin	01-11-2019	11	01-01-2019	653442-11-2019	Departmental	653442-11-2019	2122
61	395	Ashton Brown	01-06-2015	6	01-01-2015	971353-6-2015	Apparel	971353-6-2015	2125
0	0	Dominik Allen	01-01-2017	1	01-01-2017	510196-1-2017	Apparel	510196-1-2017	1772
0	0	Cherry Riley	01-05-2016	5	01-01-2016	703859-5-2016	Travel	703859-5-2016	2004
77	342	Amelia Ellis	01-10-2015	10	01-01-2015	142007-10-2015	Travel	142007-10-2015	1793
0	0	Arnold Montgomery	01-05-2019	5	01-01-2019	167157-5-2019	Travel	167157-5-2019	2023
0	0	Oliver Barrett	01-09-2017	9	01-01-2017	823162-9-2017	Travel	823162-9-2017	2139
0	0	Frederick Myers	01-10-2019	10	01-01-2019	786924-10-2019	Apparel	786924-10-2019	1668
0	0	Lana Johnston	01-03-2018	3	01-01-2018	828138-3-2018	Departmental	828138-3-2018	1836
0	0	Dominik Harris	01-01-2017	1	01-01-2017	145954-1-2017	Dining	145954-1-2017	1974
0	0	Jordan Wilson	01-10-2017	10	01-01-2017	426894-10-2017	Apparel	426894-10-2017	2009
0	0	Maria Wells	01-12-2016	12	01-01-2016	172293-12-2016	Books	172293-12-2016	1905
28	874	Maria Wells	01-01-2015	1	01-01-2015	884933-1-2015	Accessories	884933-1-2015	1905
0	0	Arnold Fowler	01-09-2019	9	01-01-2019	492691-9-2019	Dining	492691-9-2019	1760

Figure 3.167: Exported data from the workflow output preview

The final flow will look like the following:

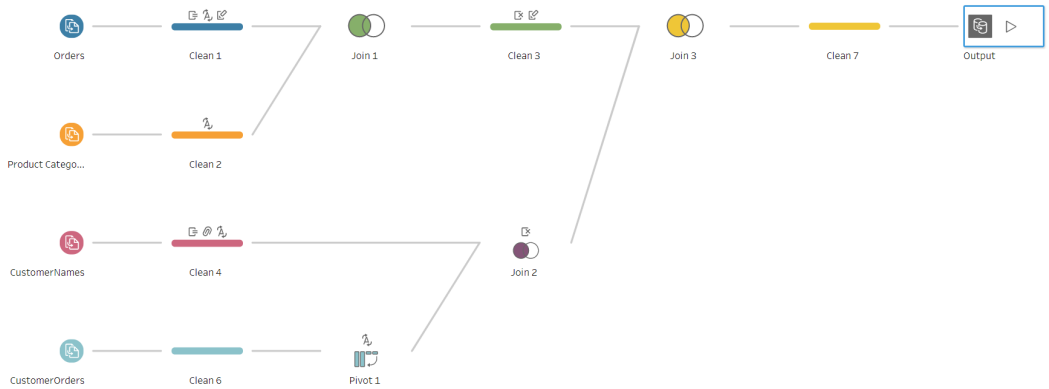


Figure 3.168: Workflow output

In this activity, you cleaned data by creating calculations and automatic groups. You also added data from multiple sources and performed transformations such as pivoting and cleaning the data throughout the flow, to output only the columns that add value to the data. Often, you will need to work on projects that require data to be added from multiple sources and apply joins at multiple levels to reach the final output. This activity has helped you achieve that.

CHAPTER 4: DATA EXPLORATION: COMPARISON AND COMPOSITION

ACTIVITY 4.01: VISUALIZING THE GROWTH OF PASSENGER CARS IN SINGAPORE

Solution:

1. Open the downloaded **`SG_Annual_Vehicle`** population data in your Tableau instance.
2. Drag and drop **`Category`** to the **`Columns`** shelf and **`Number`** to the **`Rows`** shelf as shown here:

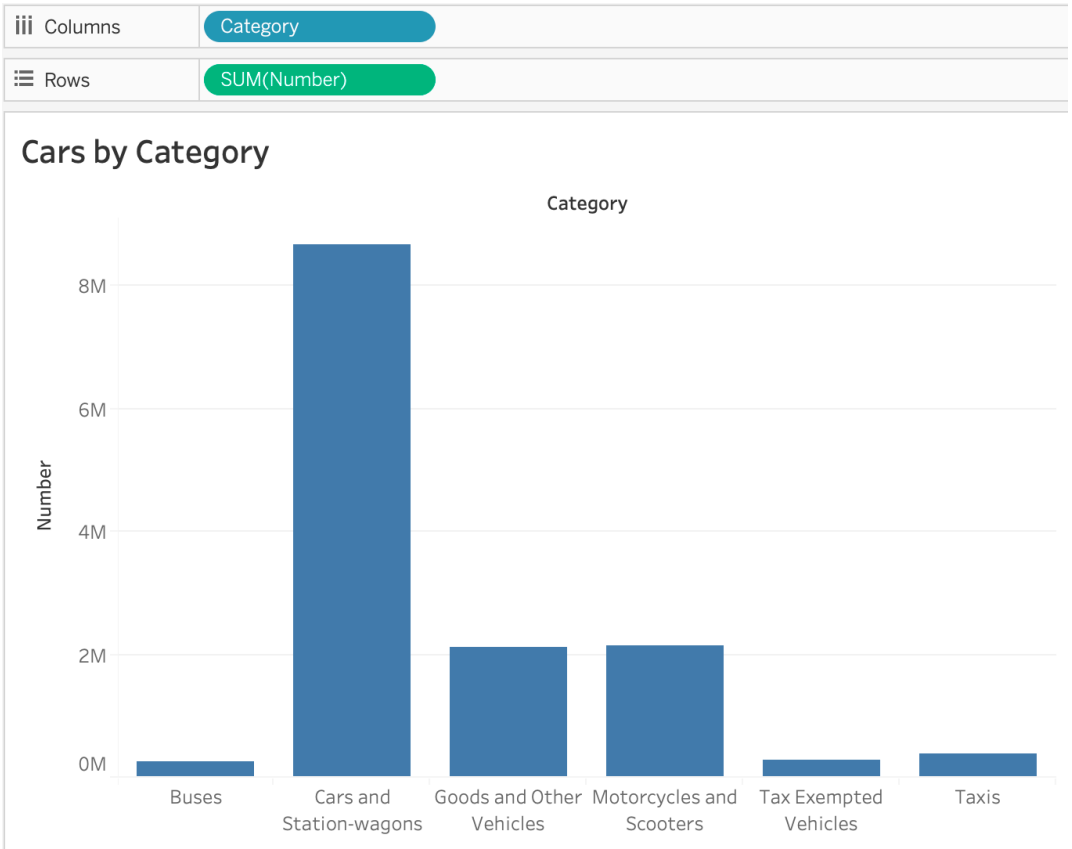


Figure 4.61: Vehicles on the road by category of vehicle

3. Drag and drop **`Year`** to the **`Columns`** shelf and make sure it's continuous (using the green dimension). You want an area chart, so change your mark type from **`Automatic`**/**`Line`** to **`Area`** as shown here:

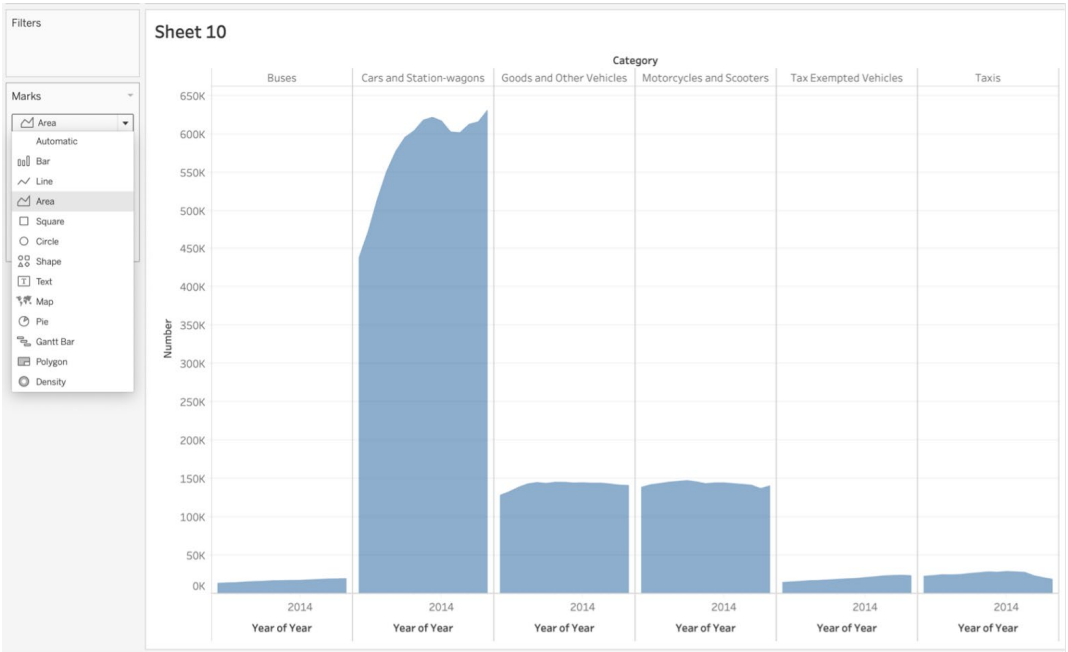


Figure 4.62: Changing from a bar chart to an area chart

4. Drag and drop **`Category`** to the **`Color Marks`** card and **`YEAR(Year)`** to the **`Label Marks`** card:

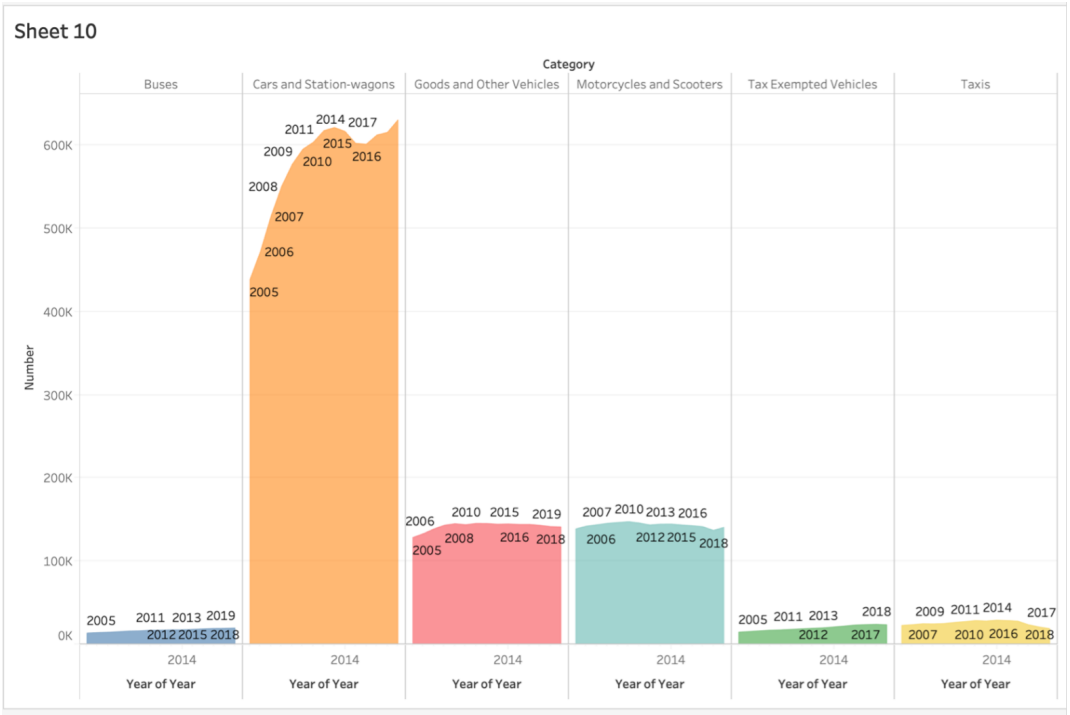


Figure 4.63: Color coding the area chart

5. Edit the axis to change the title and major ticks. Right-click on the axis and change the title to **`Year of Audit`**, then click on the **`Tick Marks`** tab and click on **`Fixed`** under **`Major Tick Marks`**, then set the interval to **`1`**:

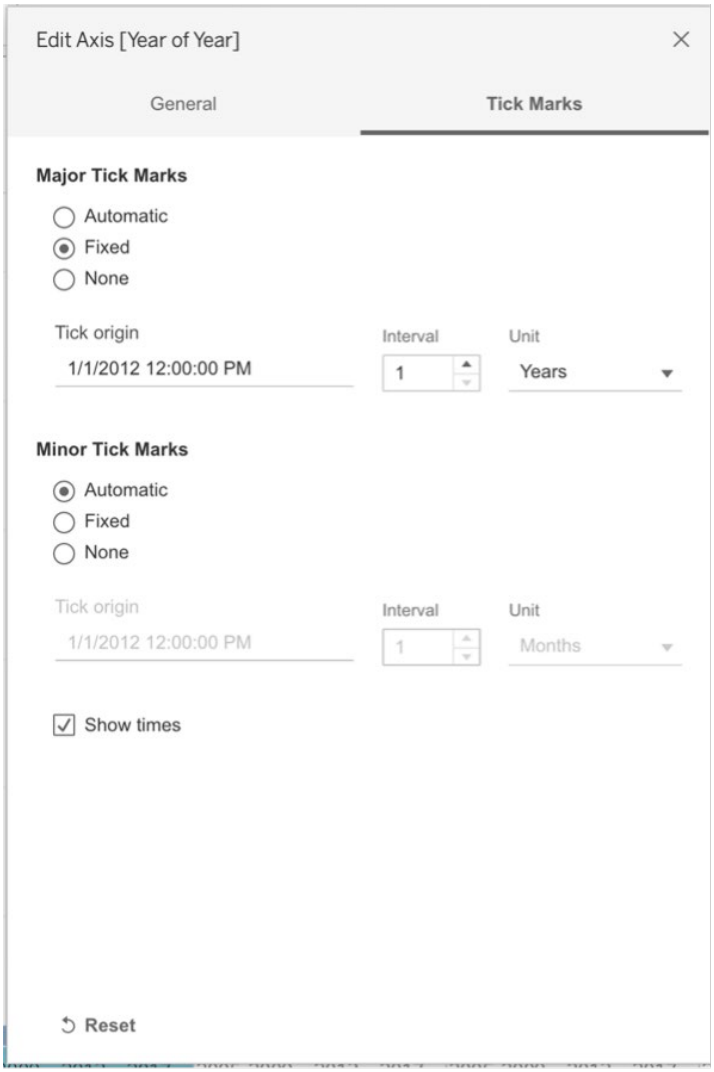


Figure 4.64: Major tick marks for the area chart

6. Wrap up the activity by double-clicking on the title of the worksheet and changing the title to **`Activity 4.1`**.

   Here is the final output:

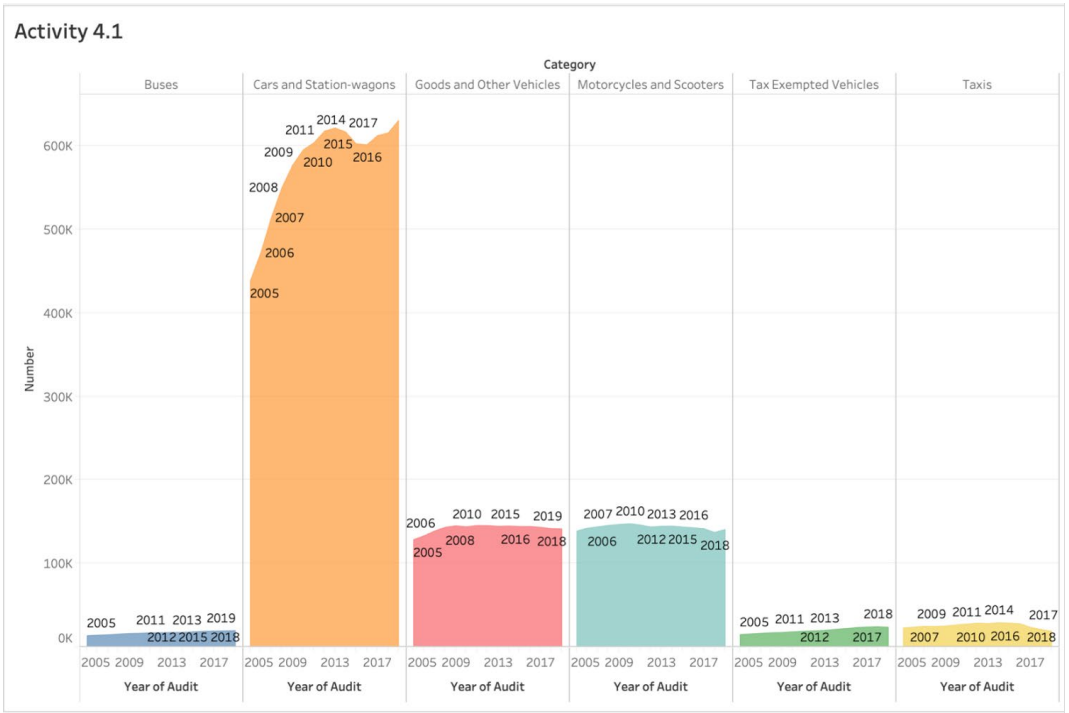


Figure 4.65: Final output for Activity 4.01

In the preceding screenshot, you can see that the number of cars and station wagons on Singapore roads saw explosive growth from 2005 to 2015, whereas growth for other categories of cars remained pretty linear.

In this activity, you practiced adding more granularity and context to your charts by adding more dimensions. Initially, you created an area chart of total vehicles by category and later also added years to the view to look at trends in the data.

CHAPTER 5: DATA EXPLORATION: DISTRIBUTION AND RELATIONSHIPS

ACTIVITY 5.01: CREATING SCATTER PLOTS

Solution:

Imagine you work as an e-commerce analyst and your manager has asked you to create a view of **`Sales`** versus **`Profit Ratio`**. They want to see the metric broken down by **`Segment`** and **`Year`**. You will use scatter plots to achieve this, and will fulfill the requirements using the **`Sample – Superstore`** dataset.

The following steps will help you complete this activity:

1. Open the **`Sample – Superstore`** dataset in your Tableau instance.
2. To add colors to the category and shape the segments, add **`Category`** to the **`Color Marks`** card and add **`Segment`** to the **`Shape Marks`** card as shown in the following figure:

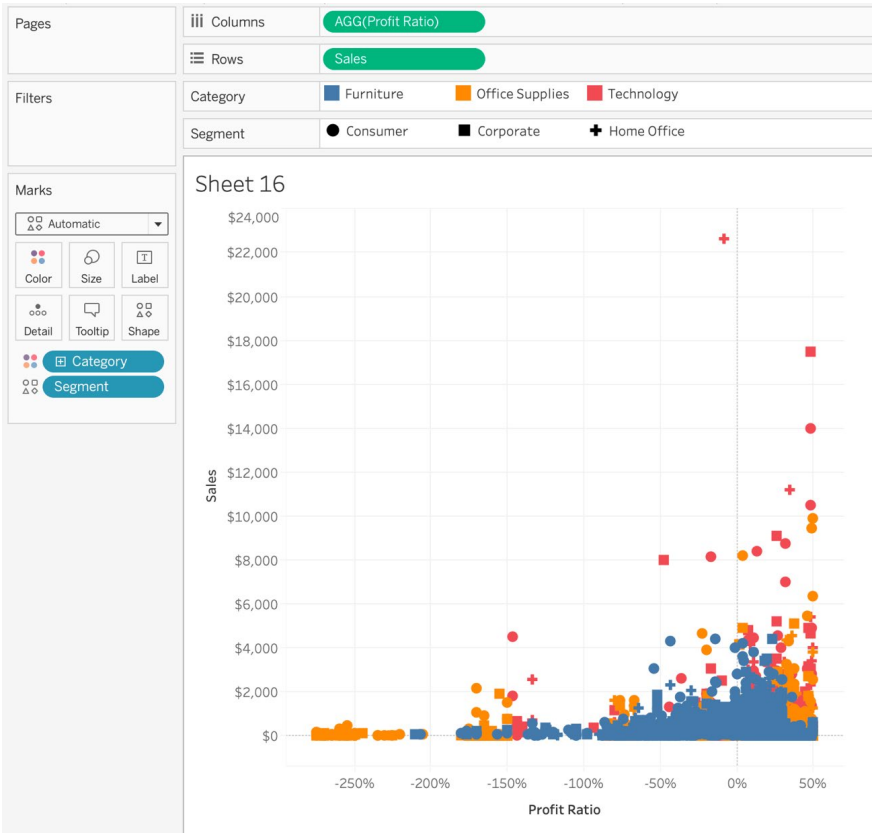


Figure 5.67: Scatter plot colorcoded by category and shape by segment

3. Now, add a couple of filters into the view. Drag **`Segment`** and **`Order Date`** from the **`Data`** pane to the **`Filters`** card. You can manually change **`Order Date`** to **`Discrete Year[Order Date]`** by clicking on the arrow in your filters, or use *Ctrl* + drag for Windows or *Option* + drag for Mac, which will open a window for you to select **`YEAR(Order Date)`** as shown in the following screenshot:

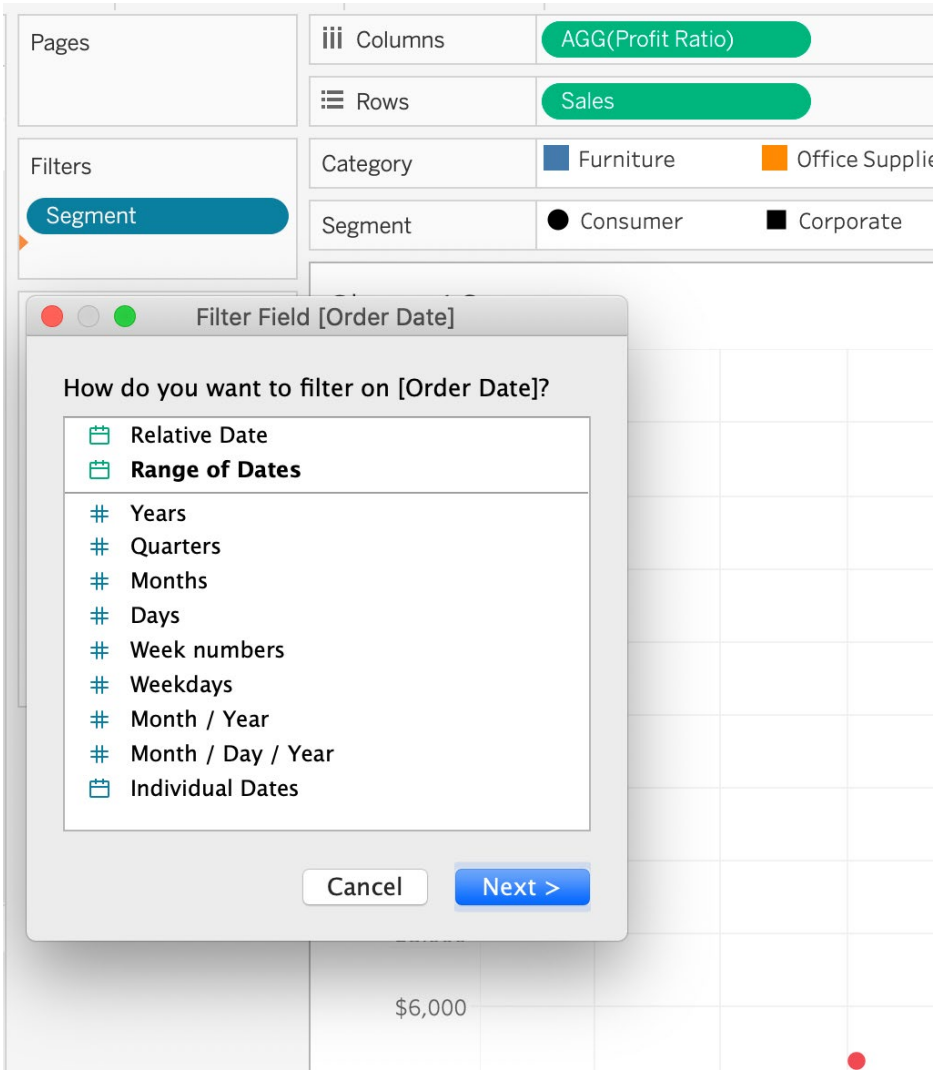


Figure 5.68: Filter Field options for the date dimension

4. Change the filters to be **`Single Value (list)`** by clicking on the right arrow or right-clicking each filter as shown in the following screenshot:

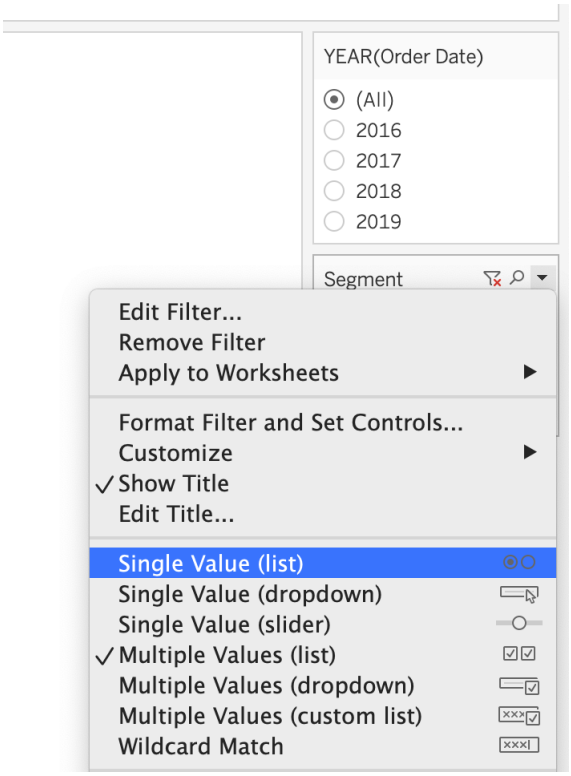


Figure 5.69: Changing the filter selection type

5. Now change the title of the worksheet to **`Scatter Plot by Segment and Year/s`**.

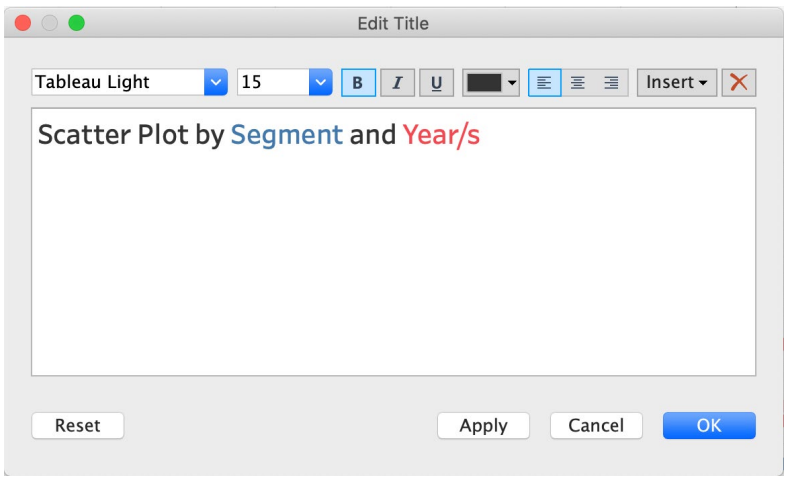


Figure 5.70: Editing the worksheet title

6. Finally, change the opacity to 70% for better readability and you have completed the final scatter plot.

   The expected final output is as follows:

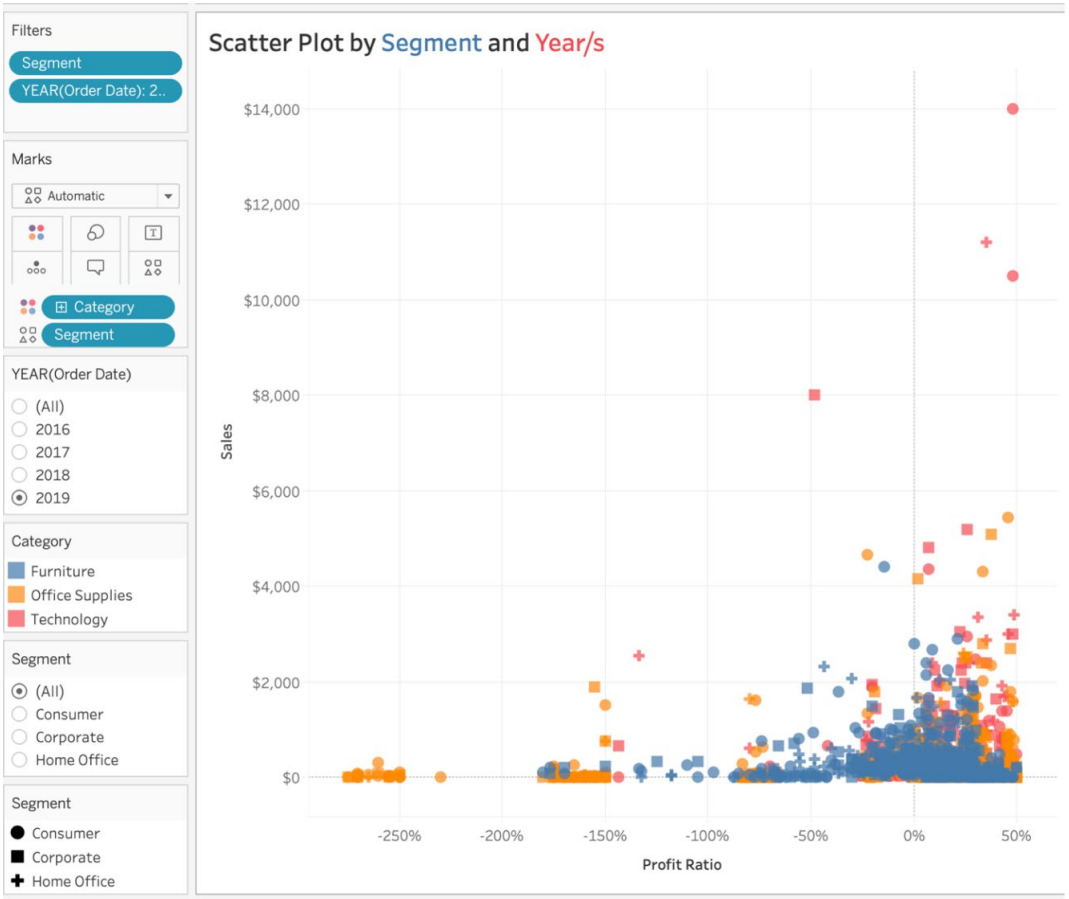


Figure 5.71: Final output of the activity

7. Drag **`Profit Ratio`** to the **`Columns`** shelf and **`Sales`** to the **`Rows`** shelf. De-aggregate the view by navigating to **`Analysis`** and unchecking **`Aggregate Measures`**. This step de-aggregates our measures.

ACTIVITY 5.02 SOLUTION: DUAL AXIS CHART WITH ASYNCHRONOUS AXES

Solution:

This activity continues on from the last. After fulfilling the initial scatter plot requirements, you are now tasked with creating a dual axis chart, that shows how **`Discounts`** affect **`Sales`** month by month. Essentially, you are asked to create a view of sales versus discounts by month using a dual axis chart with an asynchronous dual axis.

The following steps will help you complete this activity:

1. Open the **`Sample – Superstore`** dataset and drag **`Sales`** to the **`Rows`** shelf and *Ctrl* + drag (for Windows) or *Option* + drag (for Mac) **`MONTH(Order Date)`** to the **`Columns shelf`**.
2. By default, a line chart was created because of the continuous **`MONTH(Order Date)`** dimension. Change the mark type from **`Automatic`** to **`Bar`**:

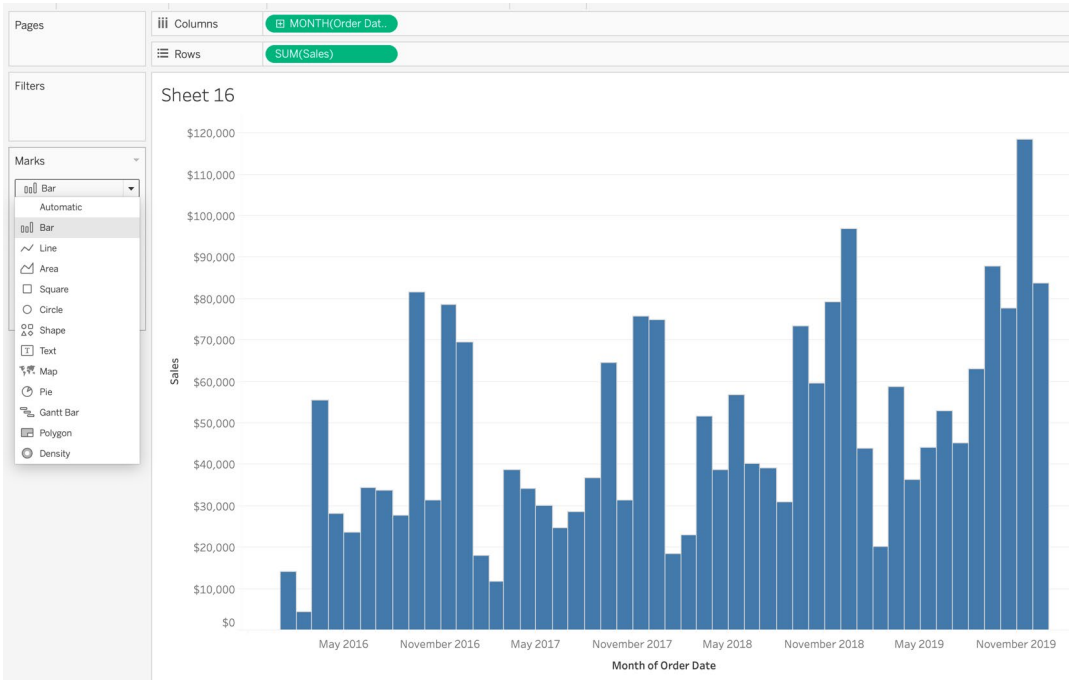


Figure 5.72: Bar chart of sales by month of the year

3. Now drag **`AVG(Discount)`** to the **`Rows`** shelf. Right-click the **`Discount`** or **`Sales`** measures and select **`Dual Axis`**:

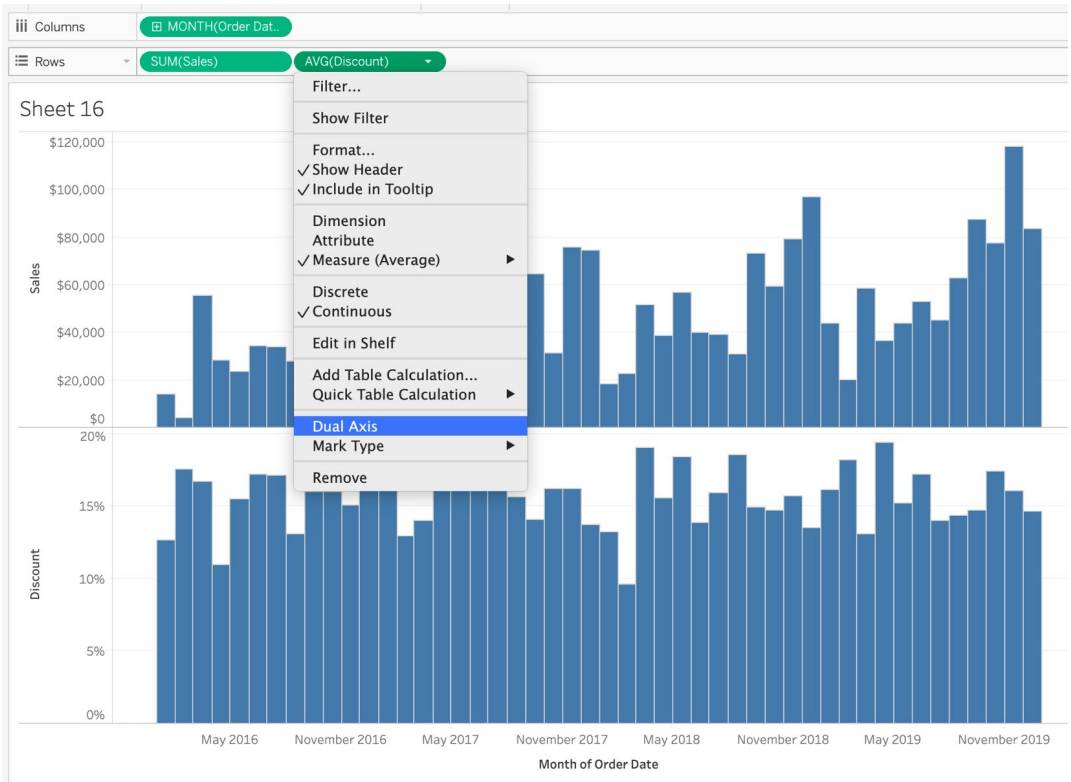


Figure 5.73: Bar charts of Sales and Discount by month of the year

4. Since you manually changed your **Marks** card to **Bar**, Tableau assumes you want all charts to be **Bar** charts in the view. To change this, from your **Marks** card, under the **AVG(Discount) Marks** card, change the mark type from **Bar** to **Line**:

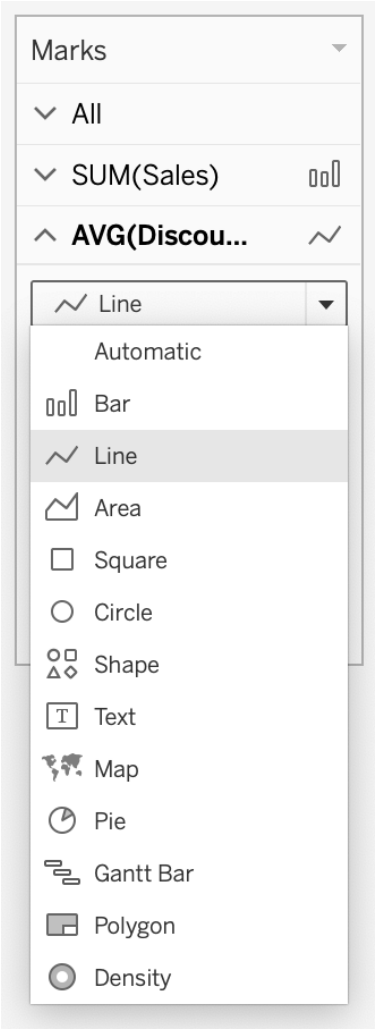


Figure 5.74: Changing the mark type

5. You might be tempted to synchronize the axes, as in *Exercise 5.07, Creating Dual axis Charts*, but synchronized axes in this chart will not make sense because the moment you synchronize, the discount range will go from 0-20% to 0-120,000%, which will change the dual axis to the following view. Notice how the blue **`Discount Measures`** line is literally on the x axis. Hence, do not synchronize your axes in this case:

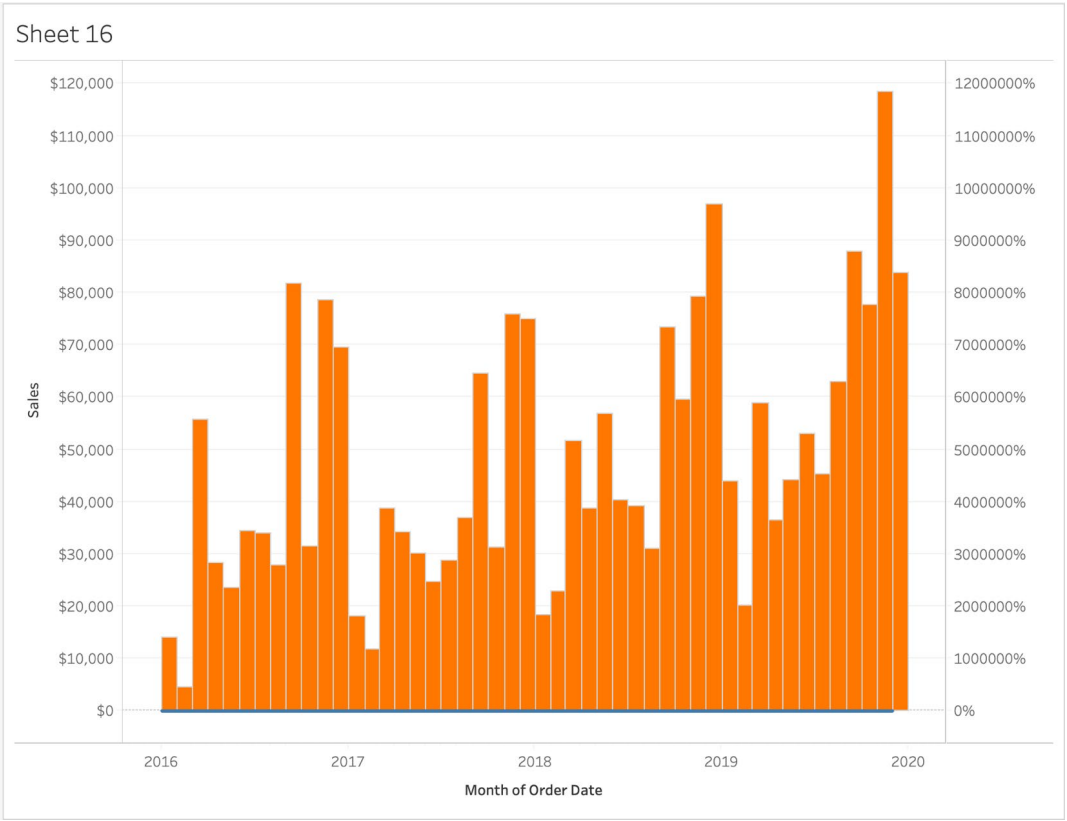


Figure 5.75: Asynchronous dual axis chart

6. Finally, add the **`Discount`** label to your **`Discount`** and **`Min/Max`**. Adding a label to every month would be overkill. The resulting chart is as follows:

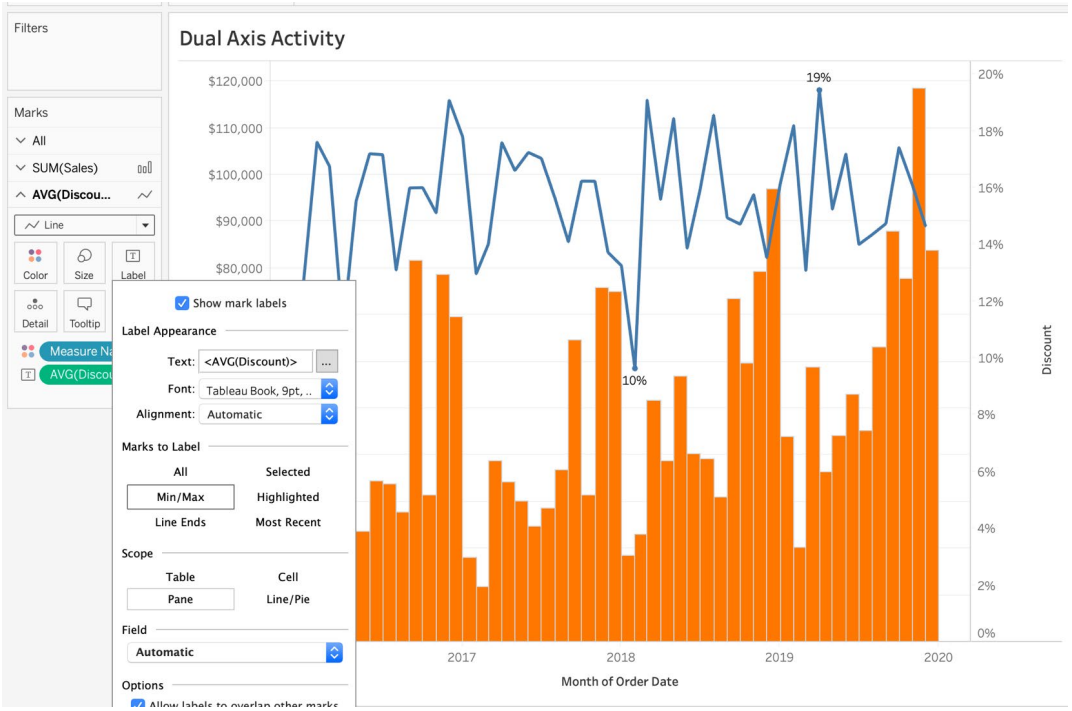


Figure 5.76: Adding Min/Max as labels

The final output will be as follows:

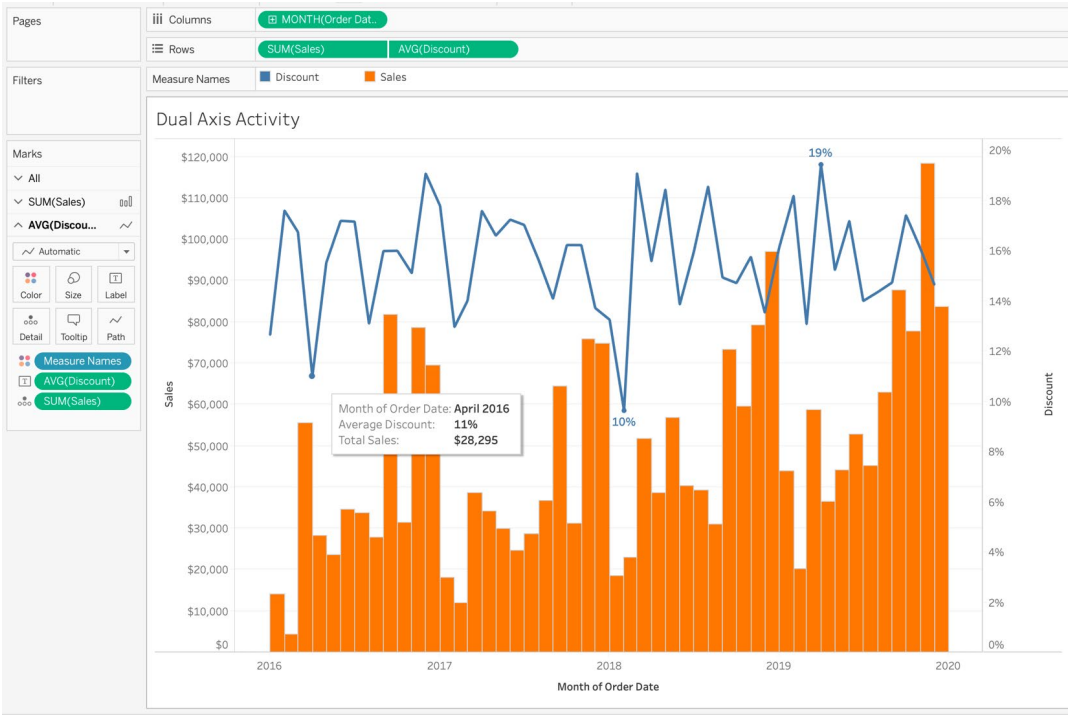


Figure 5.77: Dual axis final output of Activity 5.02

CHAPTER 6: DATA EXPLORATION: EXPLORING GEOGRAPHICAL DATA

ACTIVITY 6.01: CREATING A LOCATION ANALYSIS USING DUAL AXIS AND MAPBOX

Solution

As a data developer for a San Francisco City Department, you are asked to create a report/visualization that will showcase the hotspots of house buyout agreements in the city from a high level and gather contextual information about the house, its neighborhood, its actual address, its buyout date, and its total number of tenants, as well as the buyout amounts for the houses. Stakeholders also want to be able to filter the map data points by buyout dates. You will be using the SF Buyout Agreement data provided in the GitHub link or by downloading the **`.shp`** file from the following link: https://packt.link/ojAf3.

Perform the following steps to complete this activity:

1. Locate the **`SF Buyout Data.shp`** file that you downloaded from GitHub and add it as the spatial data source in Tableau:

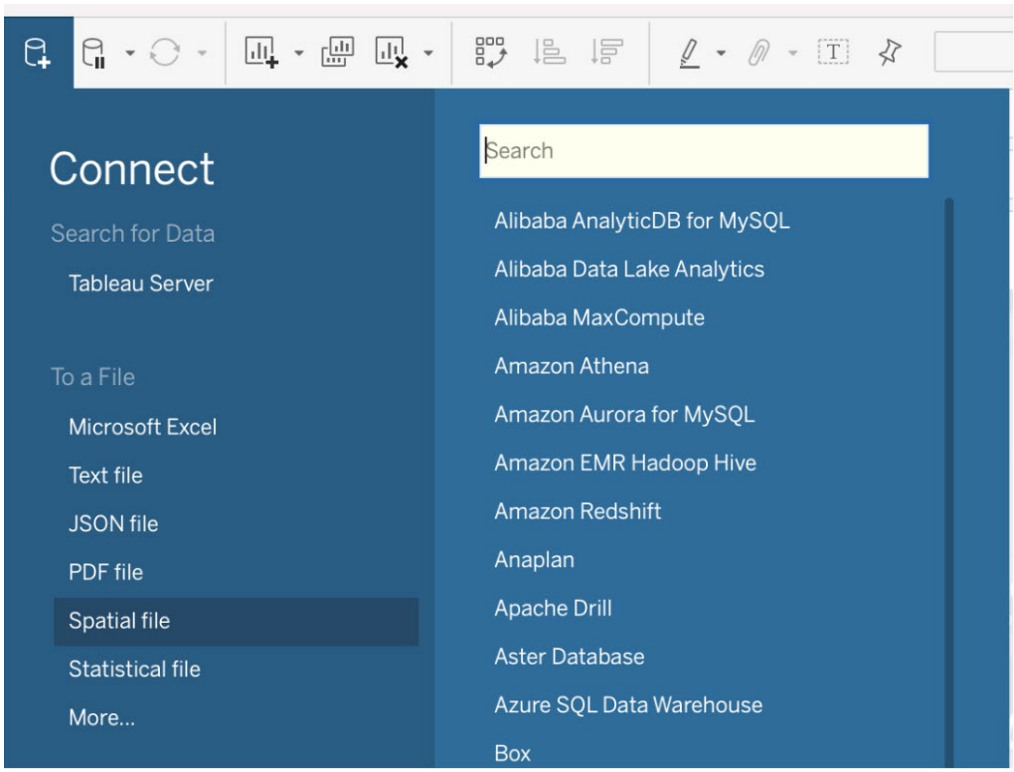


Figure 6.82: Loading the spatial data source

2. Create a new worksheet named **`SF Buyout Map`**.
3. Before proceeding to the next steps, edit the title to **`SF Buyout Map`** and rename some of the column names for easier understanding. The mappings are as follows:

- **`Case Number`** - **`Case Number`**
- **`Date Pre B`** - **`Pre Buyout Disclosure Date`**
- **`Date Buyou`** - **`Buyout Date`**
- **`Buyout Amo`** - **`Buyout Amount`**
- **`Number of`** - **`Tenants`**
- **`Analysis N`** - **`Neighbourhood`**

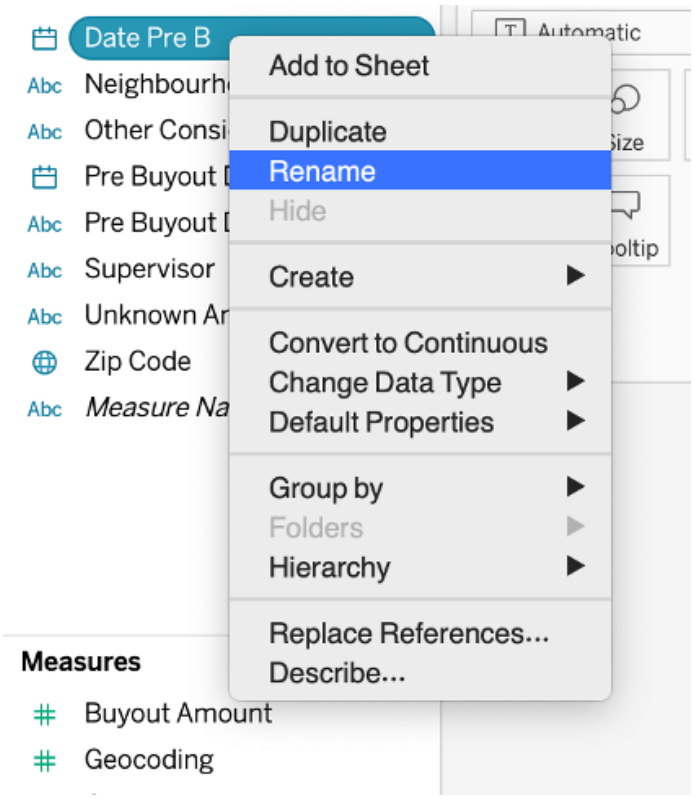


Figure 6.83: Renaming columns in Tableau

4. Add **`Buyout Date`** as a filter. Include only non-null values by selecting **`Range of dates`**. Make sure that **`Include Null Values`** is unticked.

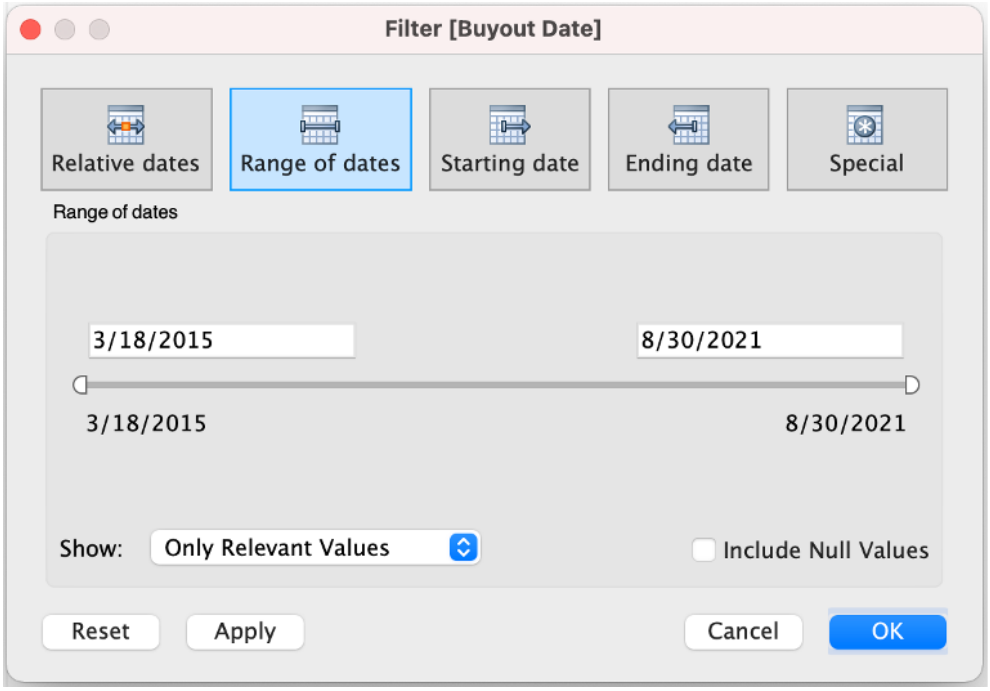


Figure 6.84: Selecting Range of dates in Tableau

5. Show the **`Buyout Date`** filter by right-clicking on the **`Buyout Date`** filter and selecting **`Show Filter`**:

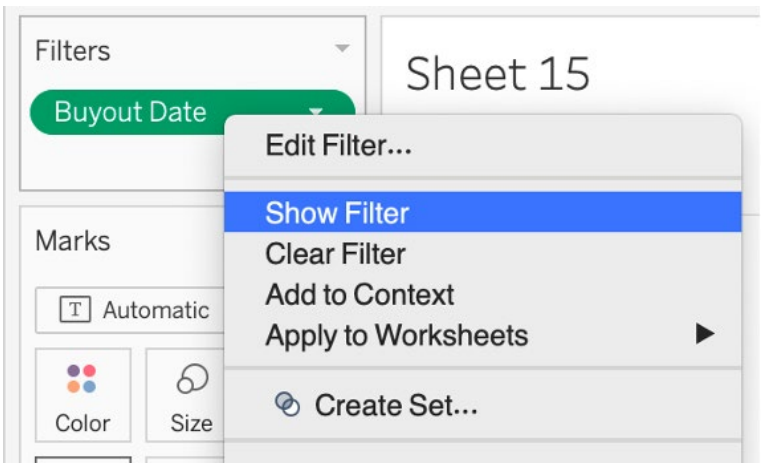


Figure 6.85: Showing the filter for Buyout Date

6. Drag **`Geometry`** onto the **`Detail`** card. The moment you do that, **`Latitude (generated)`** and **`Longitude (generated)`** will be automatically added to the **`Rows`** and **`Columns`** shelves, as shown:

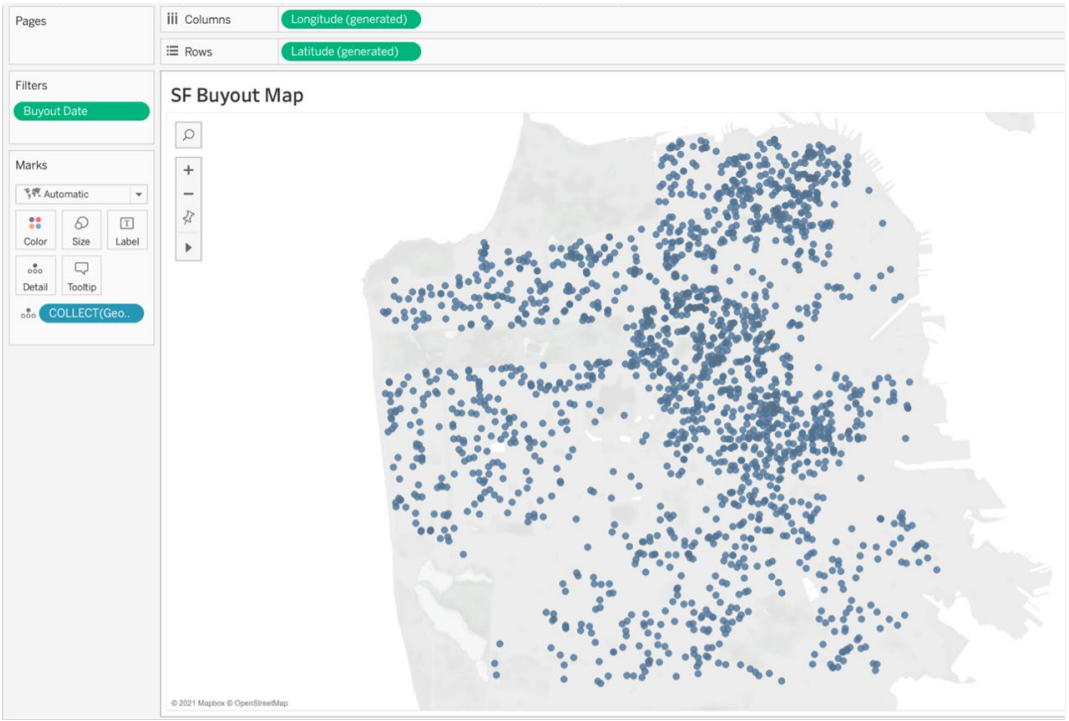


Figure 6.86: Adding the Geometry column to the Detail marks card

7. The goal of this activity is to have two layers of information available for your stakeholder, so you will need two maps as the dual axis to achieve your objective. Duplicate **`Latitude (generated)`** on the **`Columns`** shelf, right-click on the second **`Latitude (generated)`**, and select **`Dual Axis`**:

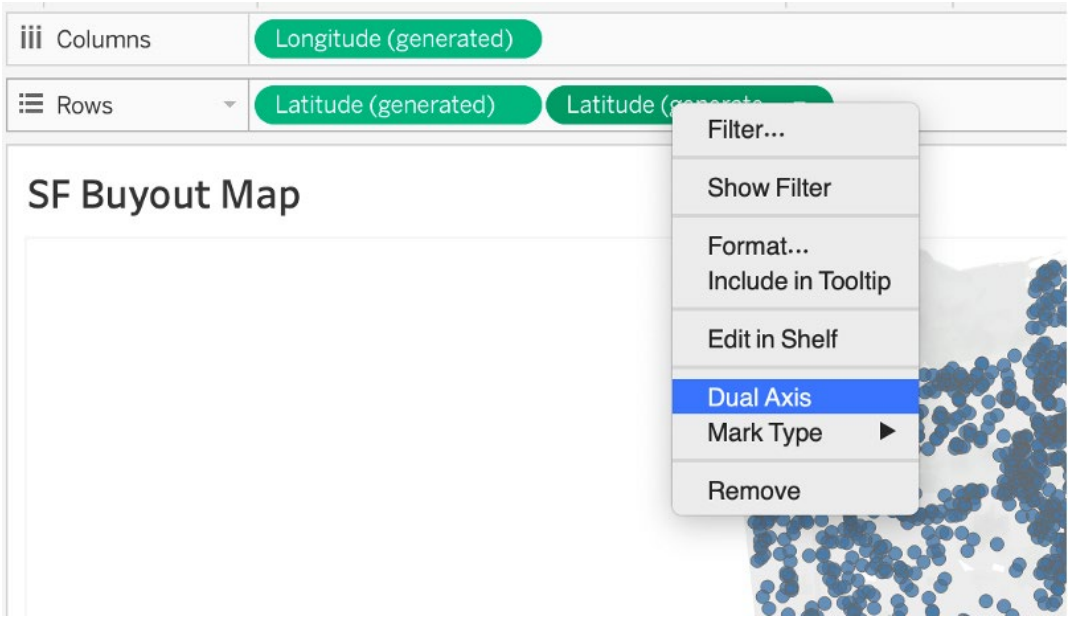


Figure 6.87: Converting the second Latitude to Dual Axis

8. Under the first **`Latitude (generated)`** marks card, change the **`Marks`** type to **`Density`**:

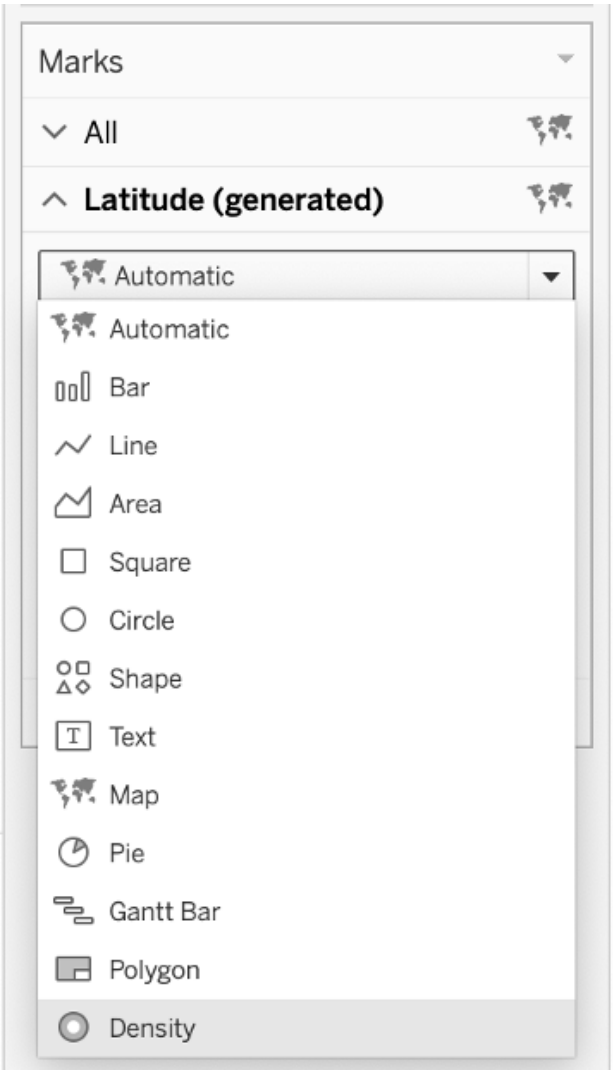


Figure 6.88: Changing the Marks type from Automatic to the Density type

9. Add **`Case Number`** to the **`Detail`** marks card and **`SUM(Tenants)`** to the **`Color`** marks card:

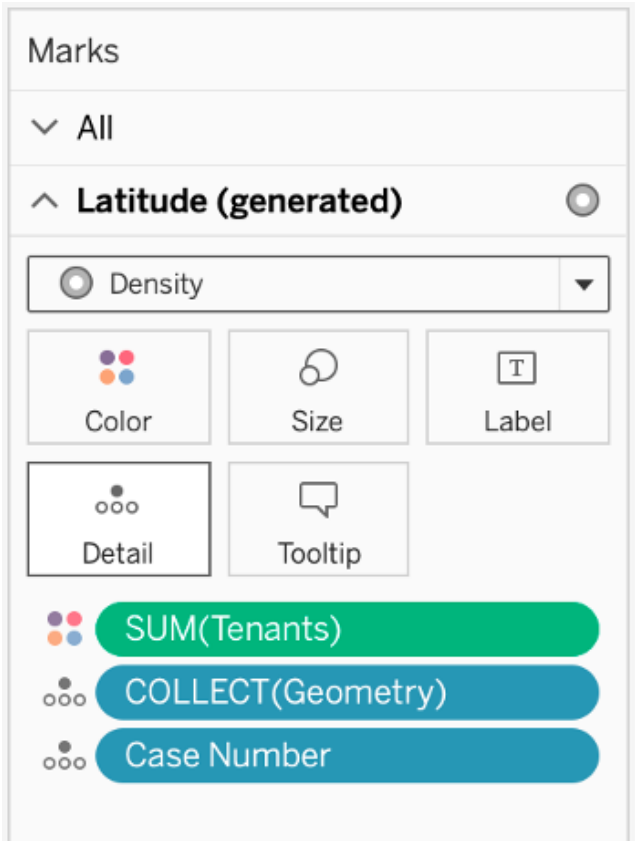


Figure 6.89: Adding Case Number and Tenants to the marks card

10. Add **`SUM(Tenants)`** to the **`Color`** marks card, which essentially adds a density layer to the map, where the darker area represents a greater number of tenants while the lighter represents low tenant buyout. Here is how it looks:

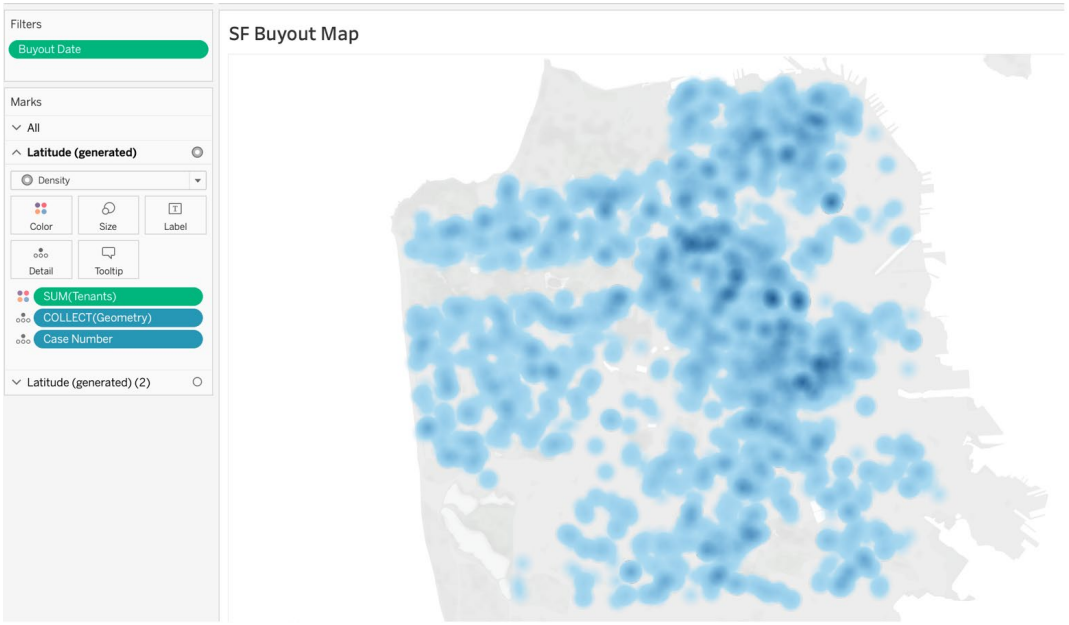


Figure 6.90: Density of tenant buyouts on SF Buyout Map

11. To add the second layer of contextual information/knowledge to your map, first change the **`Marks`** type to the **`Circle`** symbol in the second **`Latitude(generated)`** marks card section:

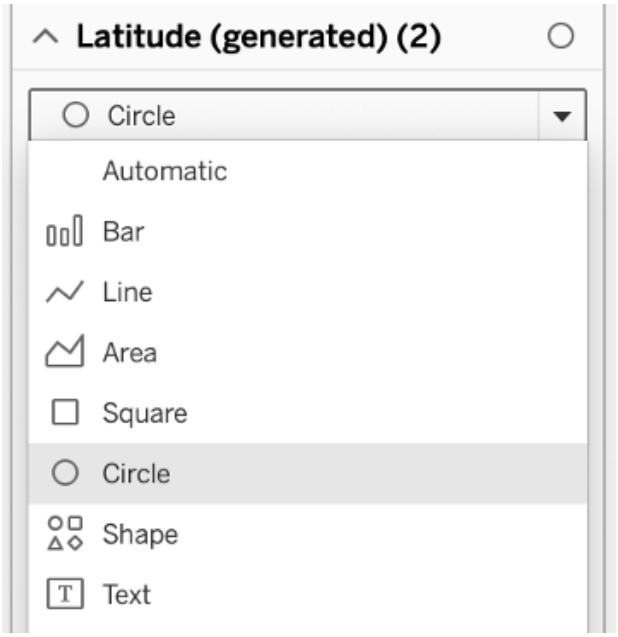


Figure 6.91: Changing the Marks type from Automatic to the Circle type

12. Add **`Case Number`** to the **`Detail`** marks card and **`Neighbourhood`** to the **`Color`** marks card:

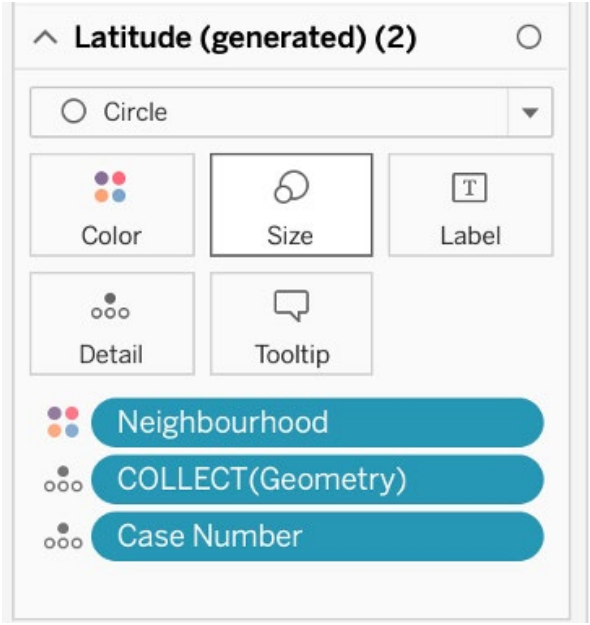


Figure 6.92: Adding Case Number and Neighbourhood to the Marks card

13. If the circles on the map look too big and/or are hiding the density below those circles, decrease the size of the circles and/or decrease the opacity as required by clicking on the **`Color`** marks card:

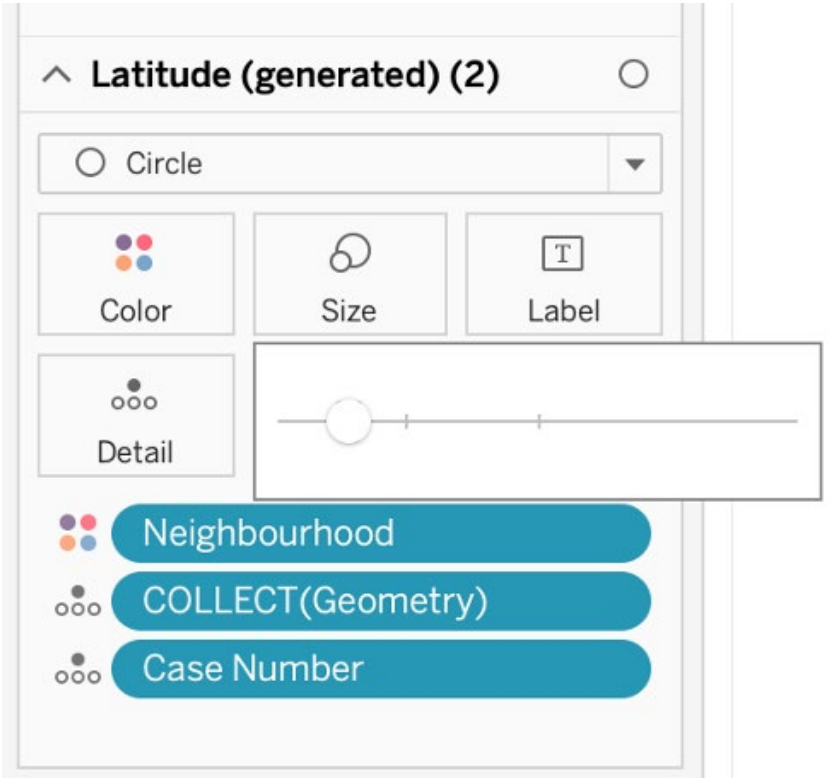


Figure 6.93: Adjusting the size of the circles on the map

For the contextual task, add those columns as part of your tooltip by adding **`Address`**, **`Buyout Date`**, **`Buyout Amount`**, and **`Tenants`** to the second **`Latitude(generated)`** marks card tooltip and edit the tooltip, as shown in the following screenshot:

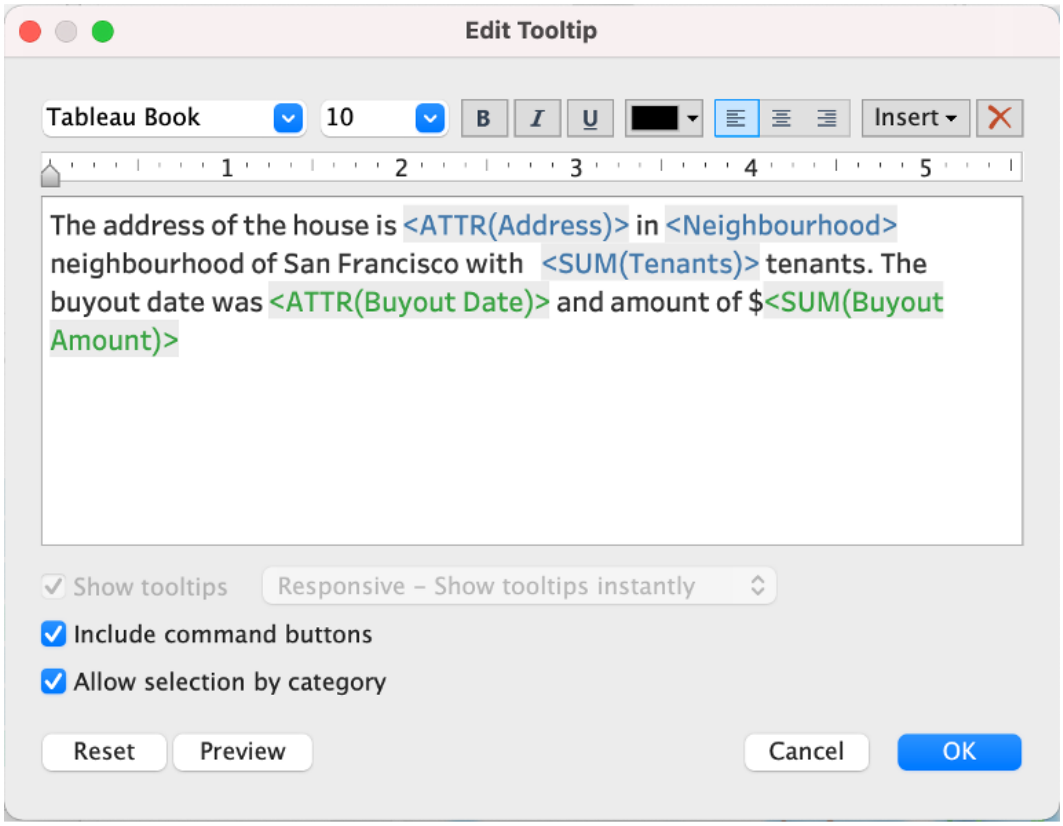


Figure 6.94: Contextual tooltip with added columns

14. Change the background of the map from **`Light`** to **`Normal`** for better aesthetics for the dual-axis maps by clicking on **`Map`** | **`Background Maps`** | **`Normal`**:

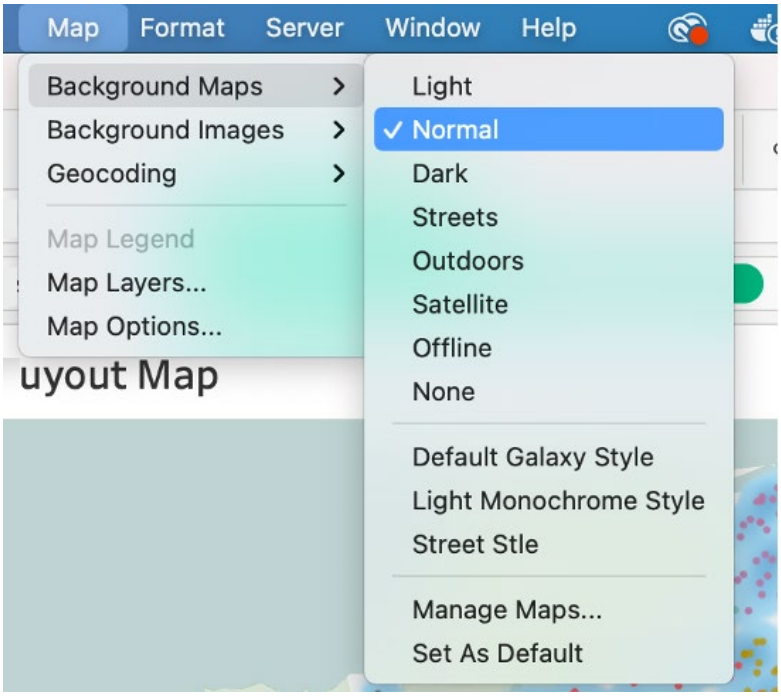


Figure 6.95: Changing background maps

The final output is as follows:

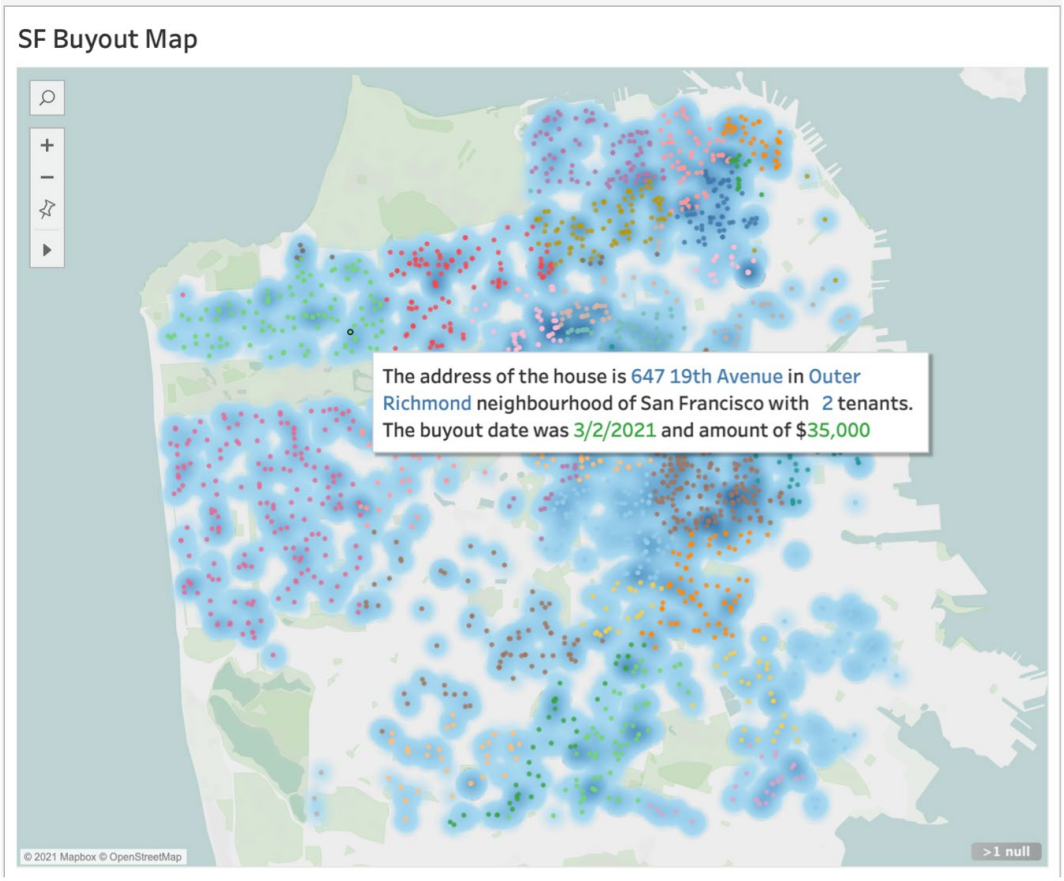


Figure 6.96: Final output for Activity 6.01

The preceding output screenshot shows the contextual information to you and stakeholders while also having a second layer of density of population (hotspots) in those particular neighborhoods. This screen is shown when you click on a particular data point on the map; you get details such as address, neighborhood, total tenants, buyout dates, as well as the buyout amount.

In this activity, you practiced all the concepts that you learned about in the preceding exercises, including creating dual-axis maps, using spatial data (a shapefile) to load a map into Tableau, enhancing maps, as well as adding contextual information for stakeholders who might review this map.

CHAPTER 7: DATA ANALYSIS: CREATING AND USING CALCULATIONS

ACTIVITY 7.01: CALCULATING THE PROFIT MARGIN

Solution:

1. First, connect to the **`Orders`** data from the **`Sample-Superstore.xlsx`** file and then drag and drop the Product Name field into the **`Rows`** shelf.
2. Double-click on **`Profit`**, then double-click on the **`Sales`** field, which should create a text table. Refer to the following screenshot:

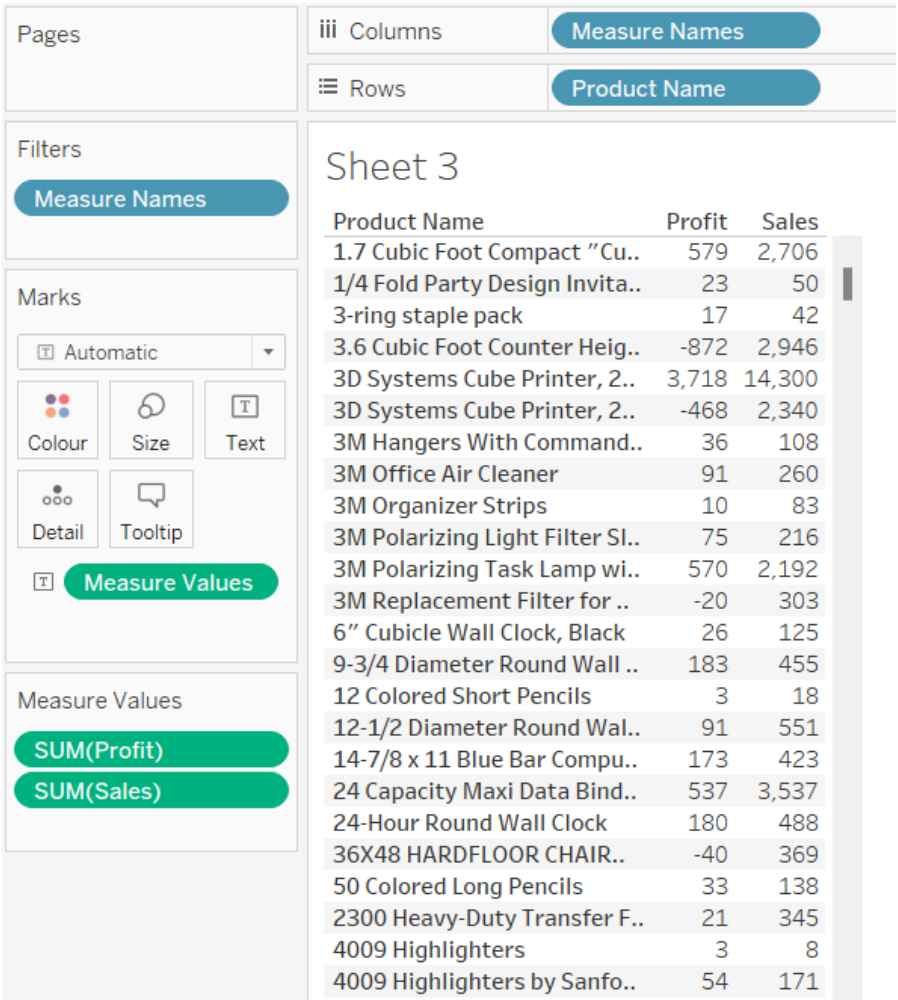


Figure 7.66: A screenshot showing the tabular view of Product Name by Profit and Sales

3. To look at only the **Xerox** products, you must create a new calculated field called **Product=Xerox?**, which will be a string calculation that classifies the products as either **Xerox** products or as **Others** products. The formula will be as follows:

```
IF CONTAINS(LOWER([Product Name]), 'xerox') THEN "Xerox" ELSE
"Others" END
```

Refer to the following screenshot:

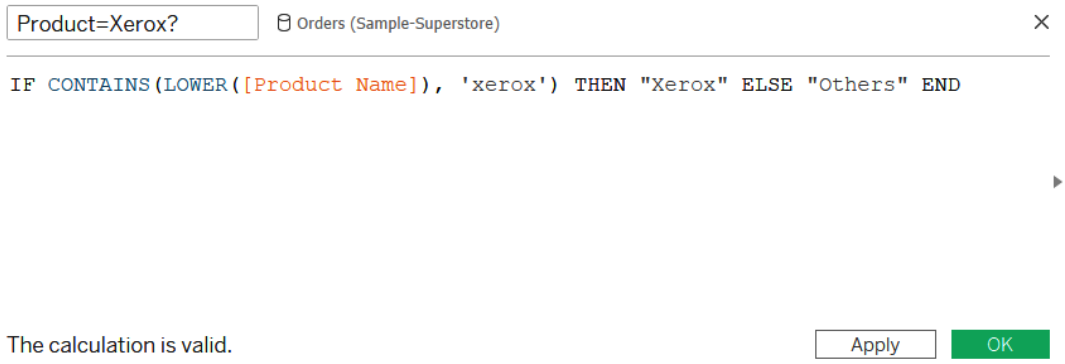


Figure 7.67: A screenshot showing the formula of the Product=Xerox? calculation

4. Use this new calculated field, **`Product=Xerox?`**, by dragging it into the **`Filter`** shelf and selecting **`Xerox`**. Refer to the following screenshot:

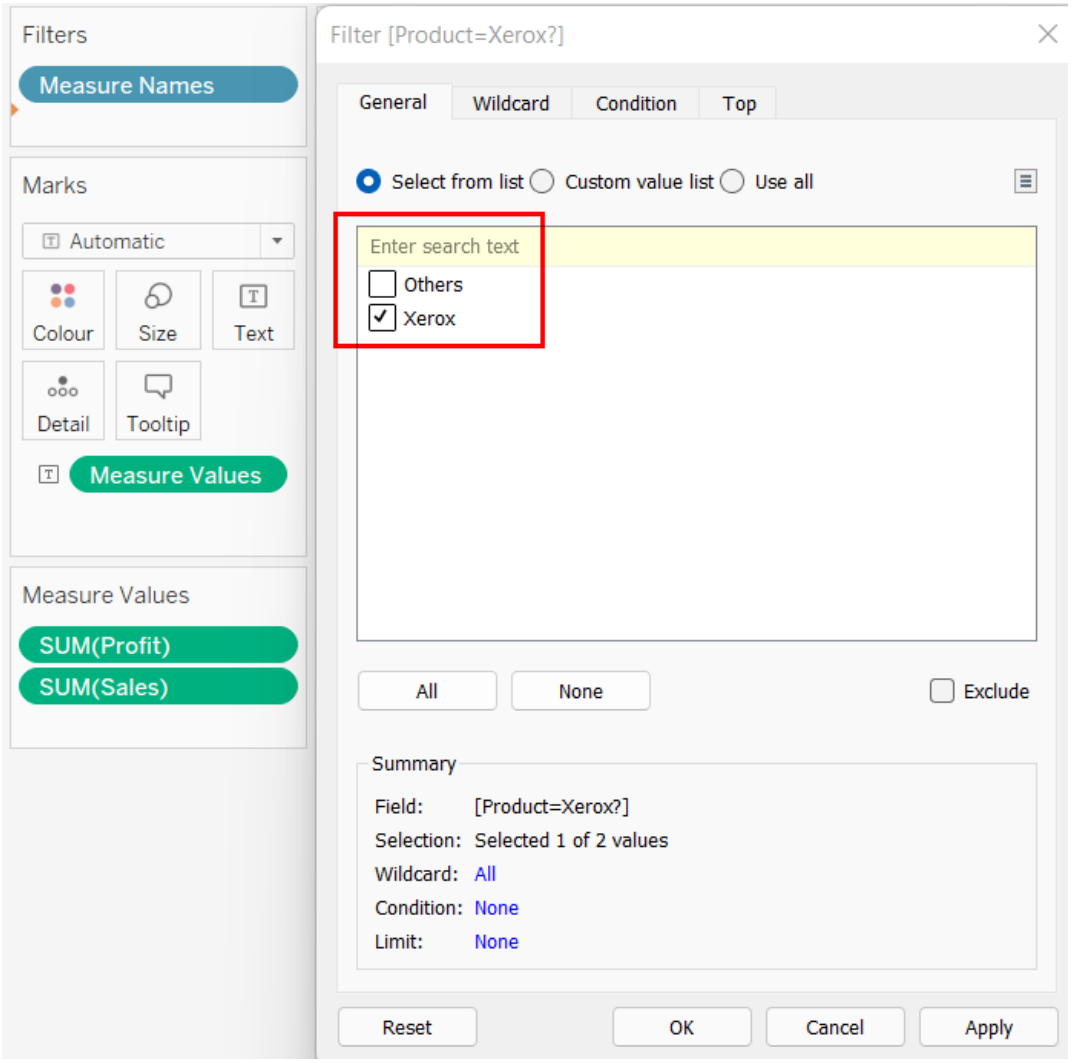


Figure 7.68: A screenshot showing the Product=Xerox? field in the Filter shelf

5. This will update your view to only show the **`Xerox`** products. Next, create another calculation called **`Profit Margin`**, which is an integer calculation that divides **`SUM(Profit)`** by **`SUM(Sales)`**. The formula is as follows:

```
SUM([Profit])/SUM([Sales])
```

Refer to the following screenshot:

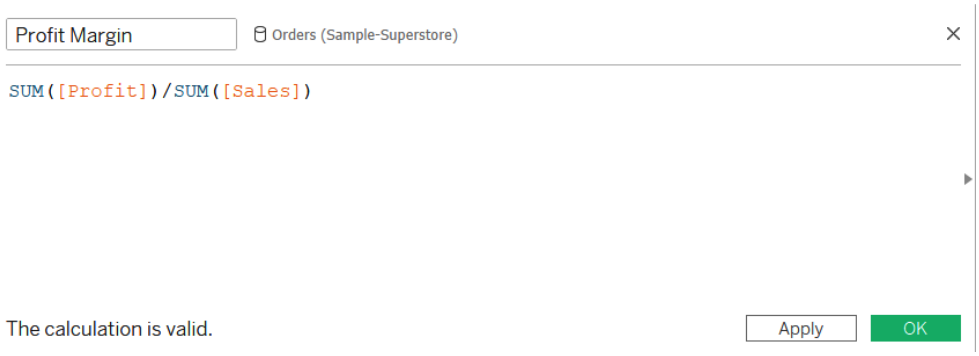


Figure 7.69: A screenshot showing the formula of Profit Margin

6. Once the **`Profit Margin`** is calculated, change **`Default Properties`** to show percentages with two decimals. Refer to the following screenshot:

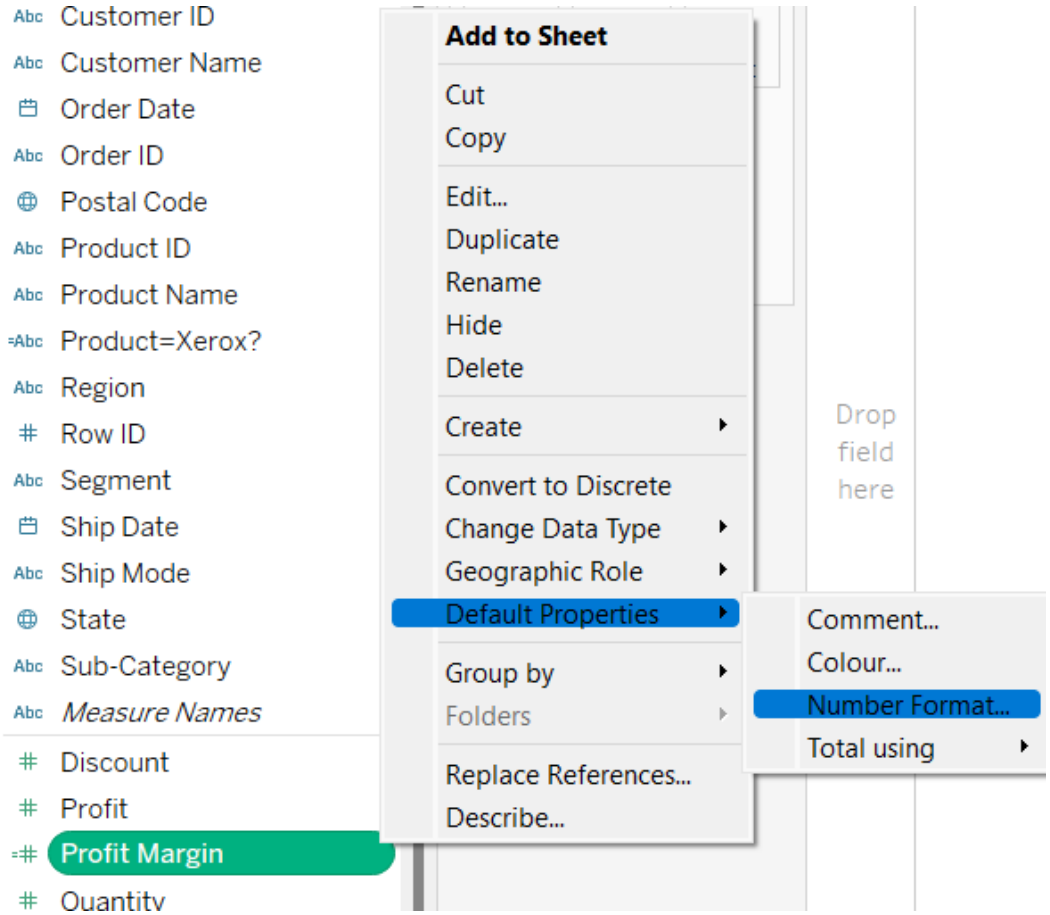


Figure 7.70: A screenshot showing the Default Properties option for Profit Margin

7. Double-click on this new calculated field, that is, **`Profit Margin`** to get it in your tabular view.
8. Drag the **`Profit Margin`** field into the **`Filter`** shelf and choose the **`At least`** option. Enter **`0.45`** and click **`OK`**. Refer to the following screenshot:

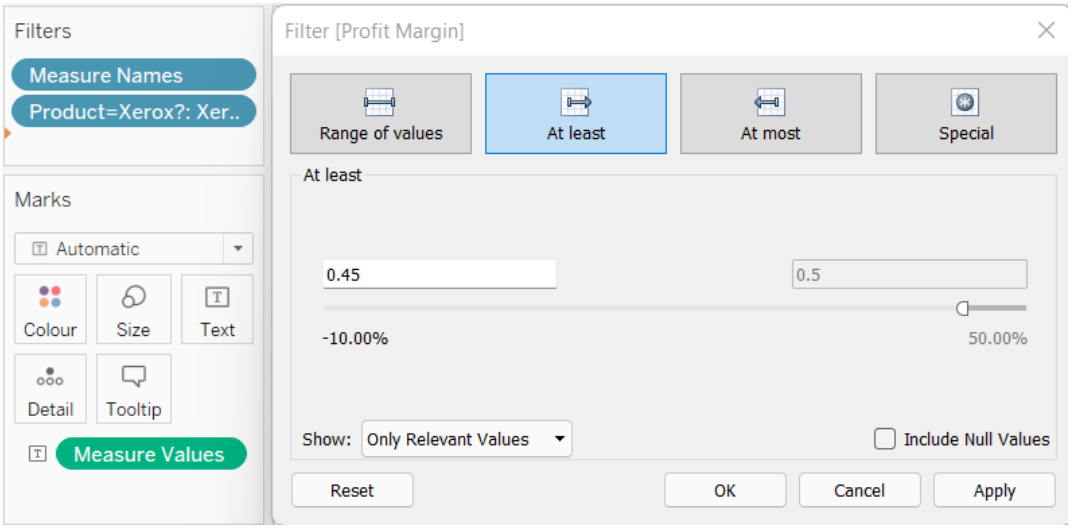
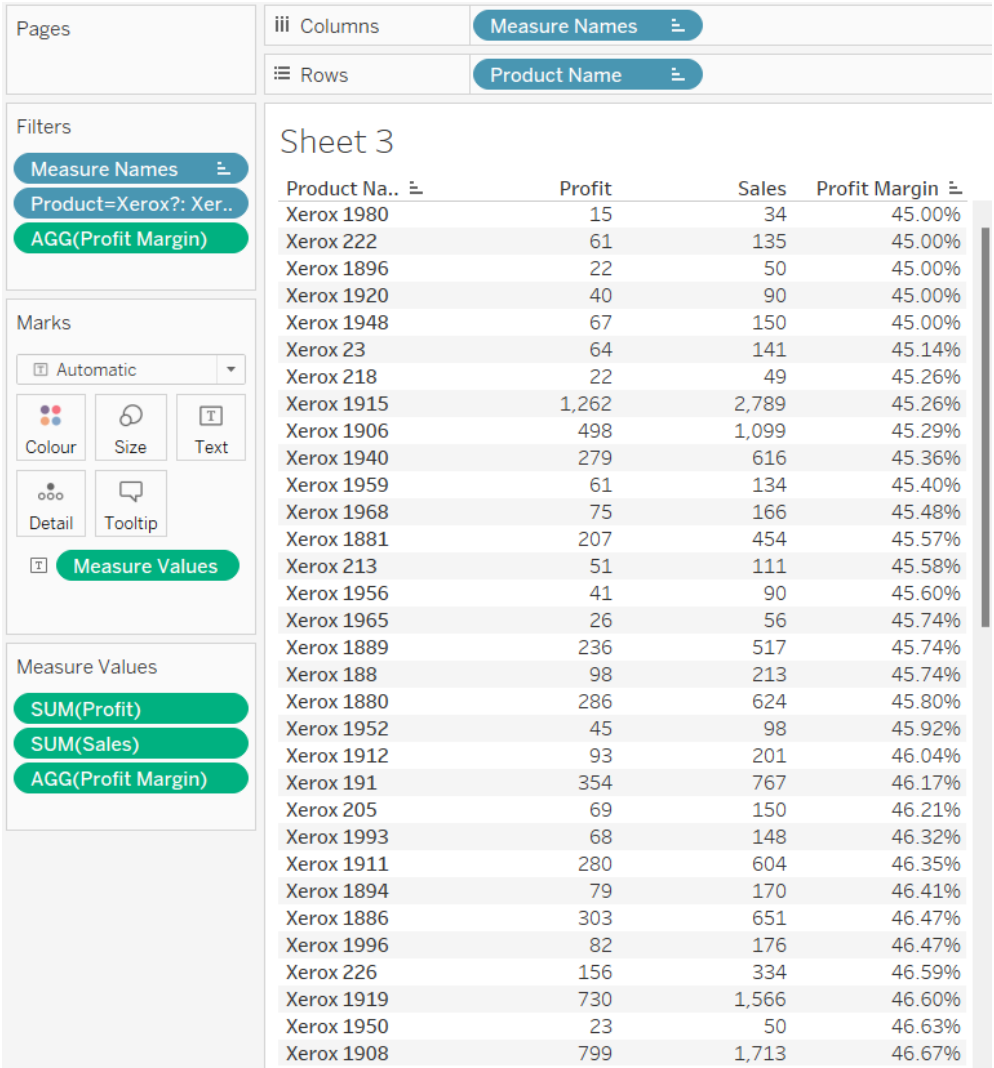


Figure 7.71: A screenshot showing the Profit Margin field in the Filter shelf

9. Finally, sort the final output in ascending order of **`Profit Margin`** and your view should update, as shown in the following screenshot:



Product Na..	Profit	Sales	Profit Margin
Xerox 1980	15	34	45.00%
Xerox 222	61	135	45.00%
Xerox 1896	22	50	45.00%
Xerox 1920	40	90	45.00%
Xerox 1948	67	150	45.00%
Xerox 23	64	141	45.14%
Xerox 218	22	49	45.26%
Xerox 1915	1,262	2,789	45.26%
Xerox 1906	498	1,099	45.29%
Xerox 1940	279	616	45.36%
Xerox 1959	61	134	45.40%
Xerox 1968	75	166	45.48%
Xerox 1881	207	454	45.57%
Xerox 213	51	111	45.58%
Xerox 1956	41	90	45.60%
Xerox 1965	26	56	45.74%
Xerox 1889	236	517	45.74%
Xerox 188	98	213	45.74%
Xerox 1880	286	624	45.80%
Xerox 1952	45	98	45.92%
Xerox 1912	93	201	46.04%
Xerox 191	354	767	46.17%
Xerox 205	69	150	46.21%
Xerox 1993	68	148	46.32%
Xerox 1911	280	604	46.35%
Xerox 1894	79	170	46.41%
Xerox 1886	303	651	46.47%
Xerox 1996	82	176	46.47%
Xerox 226	156	334	46.59%
Xerox 1919	730	1,566	46.60%
Xerox 1950	23	50	46.63%
Xerox 1908	799	1,713	46.67%

Figure 7.72: A screenshot showing the final output of Activity 7.01

You now have your **`Profit Margins`** calculated and displayed only for the **`Xerox`** products, and these **`Xerox`** products are further filtered to eliminate those with **`Profit Margin`** below 45%. You have now computed **`Profit Margin`** and filtered data.

ACTIVITY 7.02: CALCULATING THE PERCENTAGE ACHIEVEMENT WITH RESPECT TO BUDGET SALES

Solution:

1. First, connect to the *CoffeeChain Query* table from the **`Sample-Coffee Chain.mdb`** data and drag the **`Product Name`** field into the **`Rows`** shelf.
2. Create a bar chart to show **`Sales`** for each **`Product`** by dragging and dropping **`Product`** into the **`Rows`**
3. shelf and **`Sales`** into the **`Columns`** shelf. Refer to the following screenshot:

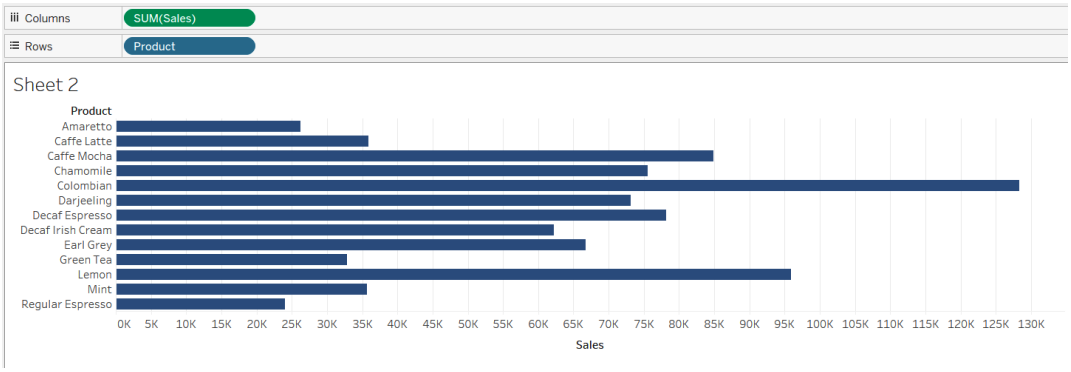


Figure 7.74: A screenshot showing the Sales and Product bar chart

4. Since you are only focusing on the year 2012, drag the **`Date`** field into the **`Filter`** shelf and select the **`Years`** option. Then, select **`2012`**. Refer to the following screenshot:

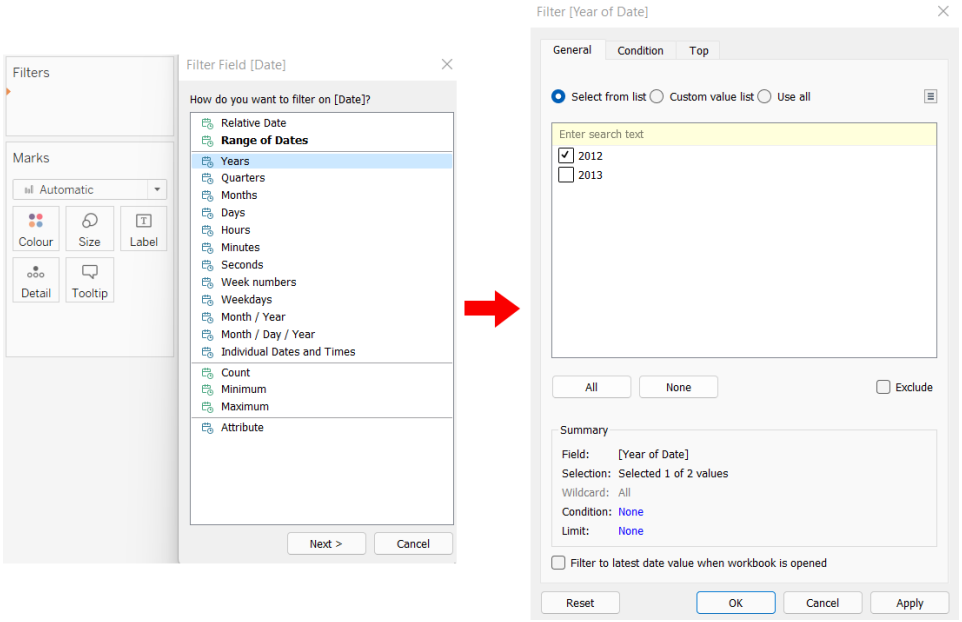


Figure 7.75: A screenshot showing the Date field in the Filter shelf

5. Create a new calculated field called **`% Achievement`** where **`Sales`** is divided by **`Budget Sales`**. The formula will be as follows:

```
SUM([Sales])/SUM([Budget Sales])
```

Refer to the following screenshot:

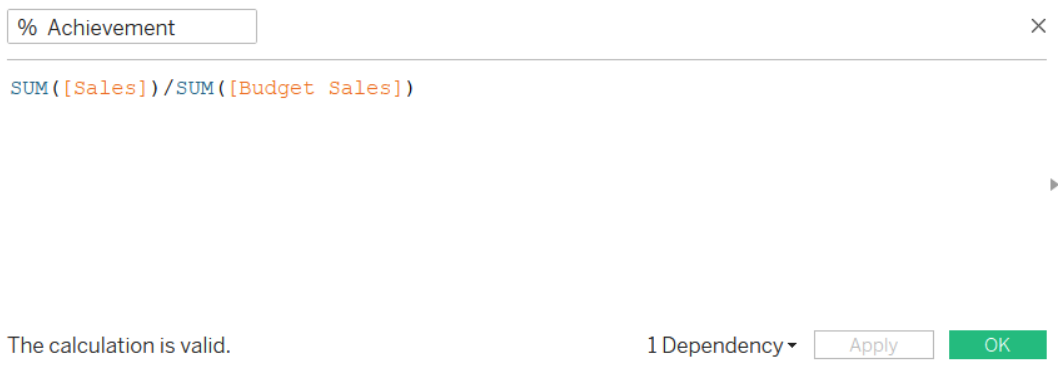


Figure 7.76: A screenshot showing the formula of % Achievement

> **NOTE**
>
> For the steps to change the number format for below *Step 6*, refer to *Figure 7.70*.

6. Change the number format of this new calculated field to show percentages with zero decimals and drop it into the **`Label`** shelf. Your view will update as shown in the following screenshot:

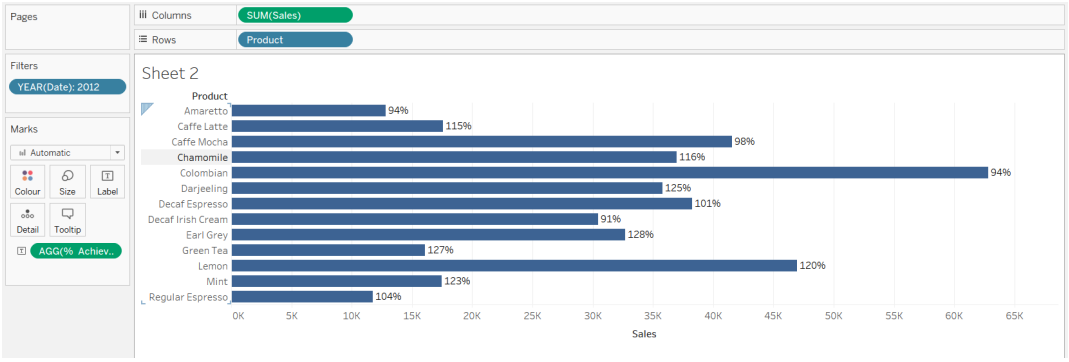


Figure 7.77: A screenshot showing % Achievement in the Label shelf

7. The preceding screenshot just shows a simple bar chart with **`% Achievement`** as labels. Now create another calculated field called **`Performance`**. The formula will be as follows:

```
IF [% Achievement]<0.95 THEN "<95% of Target achieved"
ELSEIF [% Achievement]>1 THEN ">100% of Target achieved"
ELSE "Between 95% to 100% Target achieved"
END
```

Refer to the following screenshot:

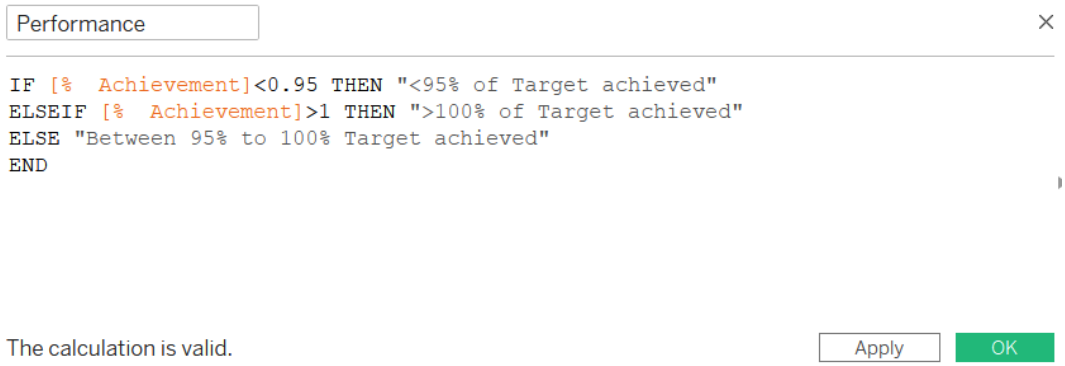


Figure 7.78: A screenshot showing the formula of Performance

8. Drop this new calculated field into the **`Color`** shelf and change the colors as specified. Refer to the following screenshot:

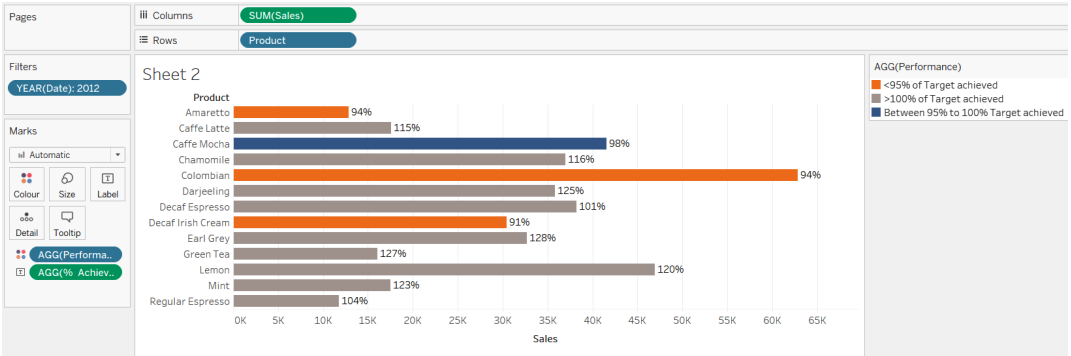


Figure 7.79: A screenshot showing the final output of Activity 7.02

> **NOTE**
>
> Use the stepped color checkbox you first encountered in *Exercise 4.08: Creating Treemaps* to configure the bars with the colors displayed in *Figure 7.79*.

As you see in the preceding screenshot, the orange bars are the products achieving less than 95%, the blue ones have achieved more than 95% but less than 100%, and the gray bars are where they have over-achieved. This analysis quickly identifies products that are under-achieving.

CHAPTER 8: DATA ANALYSIS: CREATING USING TABLE CALCULATIONS

ACTIVITY 8.01 SOLUTION: MANAGING HOSPITAL BED ALLOCATIONS

Solution:

The following steps will help you complete this activity:

1. Connect to the data using the **Connect** pane. Select **Microsoft Excel**, and navigate to the location where you have saved the **Activity1.xls** file. Click **Open** to add it to Tableau Desktop.
2. Create a calculation to find the number of patients currently admitted using the **Open**, **Discharges**, and **Re-open** columns. The number of patients currently admitted will be the new admissions and re-admissions, minus the discharges; you can see the calculation in the following figure:

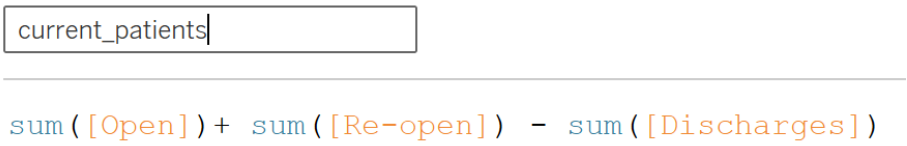


Figure 8.80: Computing the currently admitted patients

3. Add this along with the **`Date`** to the view, as follows:

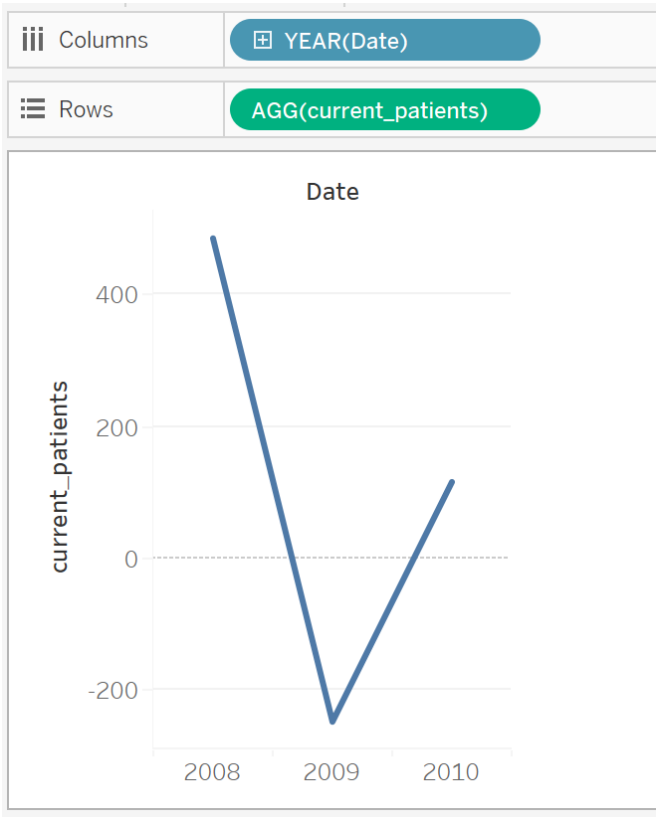


Figure 8.81: Adding a calculation to the view

> **NOTE**
>
> If you are using a version of Tableau later than 2020.1, you may need to convert the date column to data type **`Date`** for this step.

4. Click on the **`Date`** dropdown and select **`Exact Date`**. Click again to open the dropdown and select **`Discrete`**.

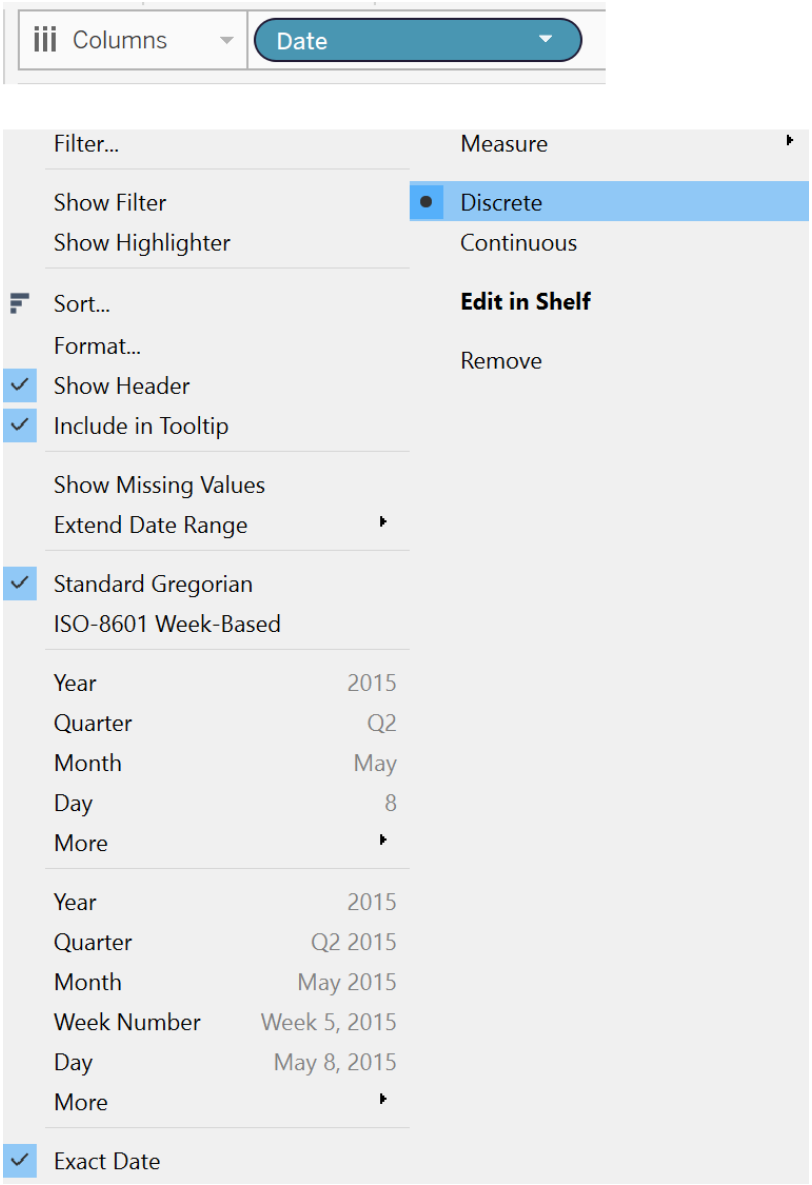


Figure 8.82: Modifying the Date column properties

5. Change the view from **`Automatic`** to a **`Bar`** chart.

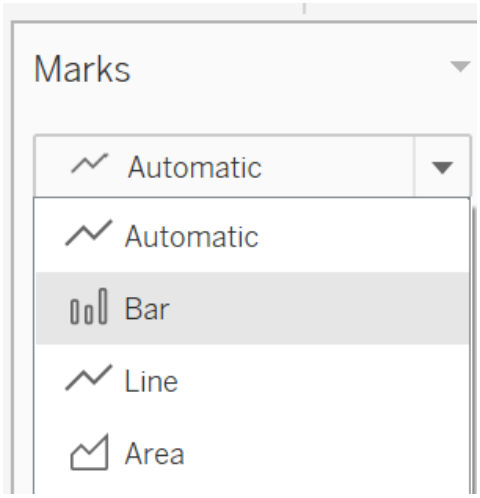


Figure 8.83: Changing the view from Automatic to Bar

You should see the following view when you worksheet is set to '**`Entire View`**':

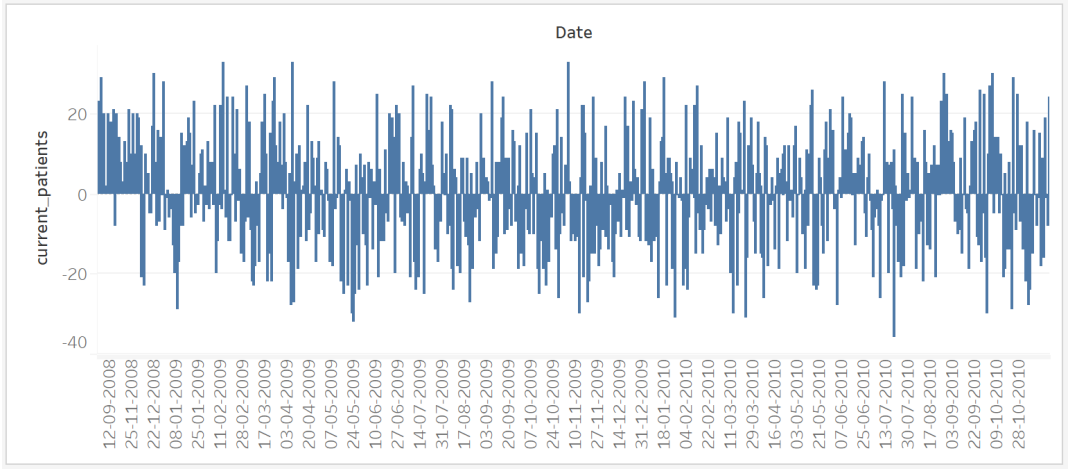


Figure 8.84: Daily admitted patients

Notice here, that there are days with a negative **`current_patients`** value. This simply indicates that the number of discharges was more than the admissions. This view isn't very helpful, though, as it breaks down the number of admissions on the basis of individual days. What is needed is a cumulative number of admissions, which can be done using a table calculation.

6. Add a table calculation to **`current_patients`**, as shown in the following figure similar to how we did it in *Step 3*. Refer to *Figure 8.81*:

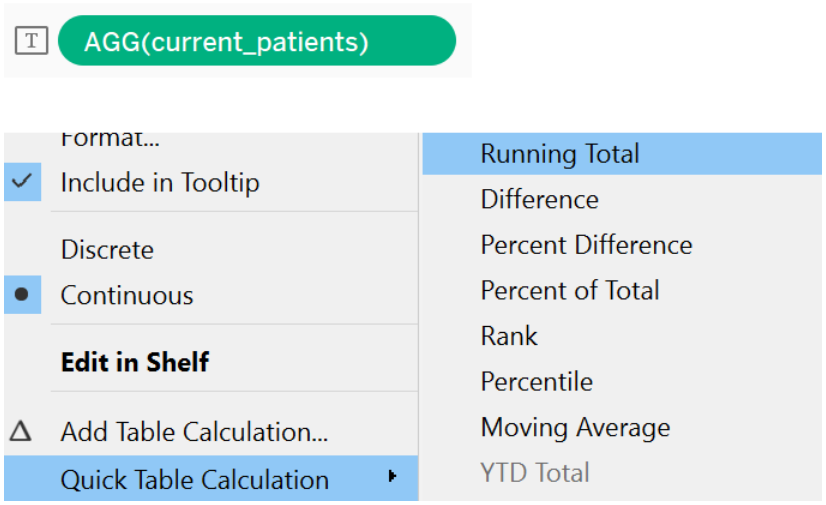


Figure 8.85: Adding Running Total to a current_patients calculation

This adds the cumulative number of currently admitted patients.

7. After doing this, notice there are no negative values, which means that the hospital always had more admissions than discharges.

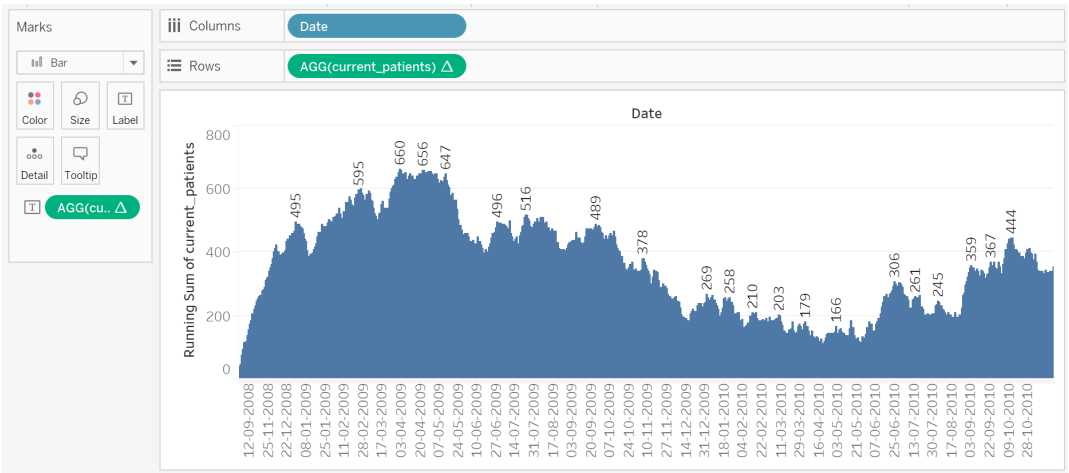


Figure 8.86: Cumulative count of current_patients admitted

Observe that in 2009, the number of current admissions was more than 600. This may have happened due to an unexpected event. Such an occurrence should be highlighted so it can be avoided in future.

8. Now, create a new calculation named **`Alert`** that takes **`RUNNING_SUM`** and checks it against the threshold value of **`600`**, as follows:

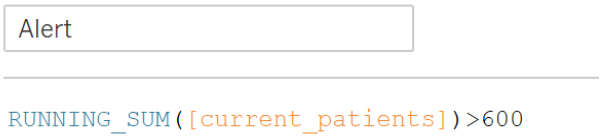


Figure 8.87: Calculation formula for Alert

9. Clicking on the **`current_patients`** in the calculation editor would display the underlying calculation for **`current_patients`** as below:

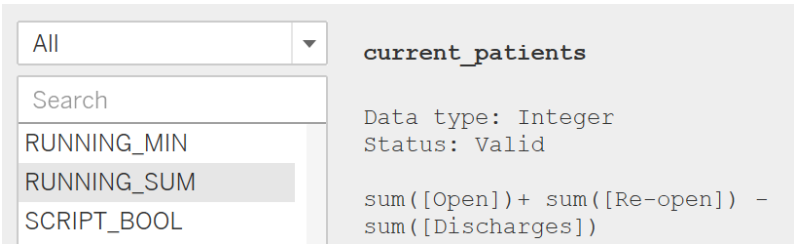


Figure 8.88: Adding Alert to the current_patients view

10. Now, add this to **`Color`** on the **`Marks`** shelf, as follows:

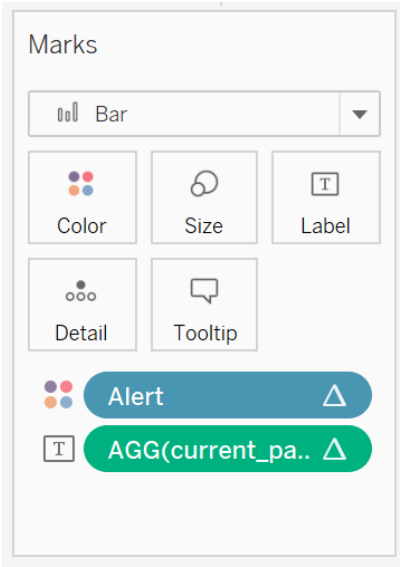


Figure 8.89: Adding Alert to Color

This is the view you generate:

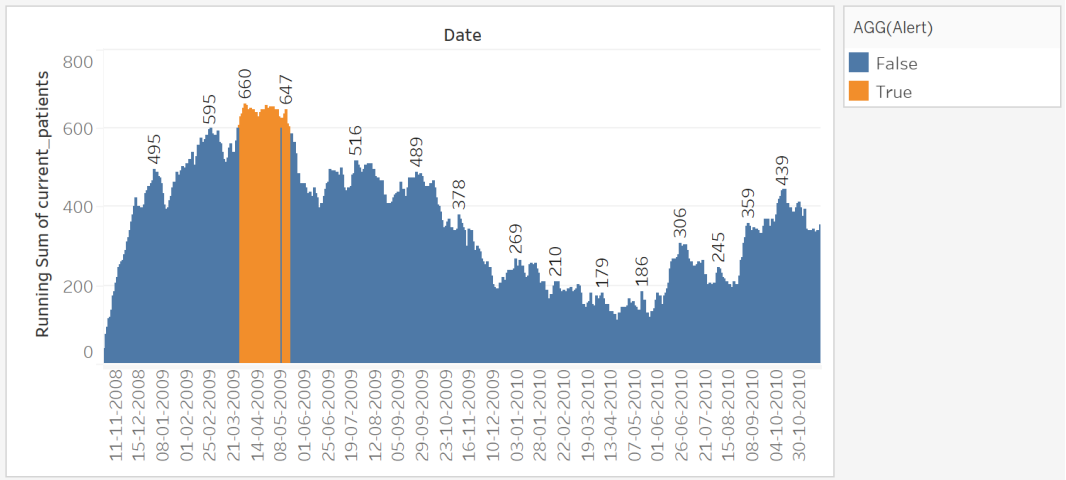


Figure 8.90: Final output

Here, you can see that in 2009, there was a period when the number of patients was more than the number of beds. Although such incidents are rare, it is imperative that they are managed properly.

With this activity, you strengthened your knowledge of creating and using table calculations. This activity helped you see how you can use cumulative values to better analyze data, by highlighting anomalies or events that may have a significant business impact.

ACTIVITY 8.02: PLANNING FOR A HEALTHY POPULATION

Solution

The following steps will help you complete this activity:

1. Reuse the calculations from the previous activity. Here, you need to use the **`RUNNING_SUM`** of **`current_patients`**.
2. Add **`Date`** and **`RUNNING_SUM`** for **`current_patients`** to the view. Also, change the date to the week level and **`Discrete`**, as follows:

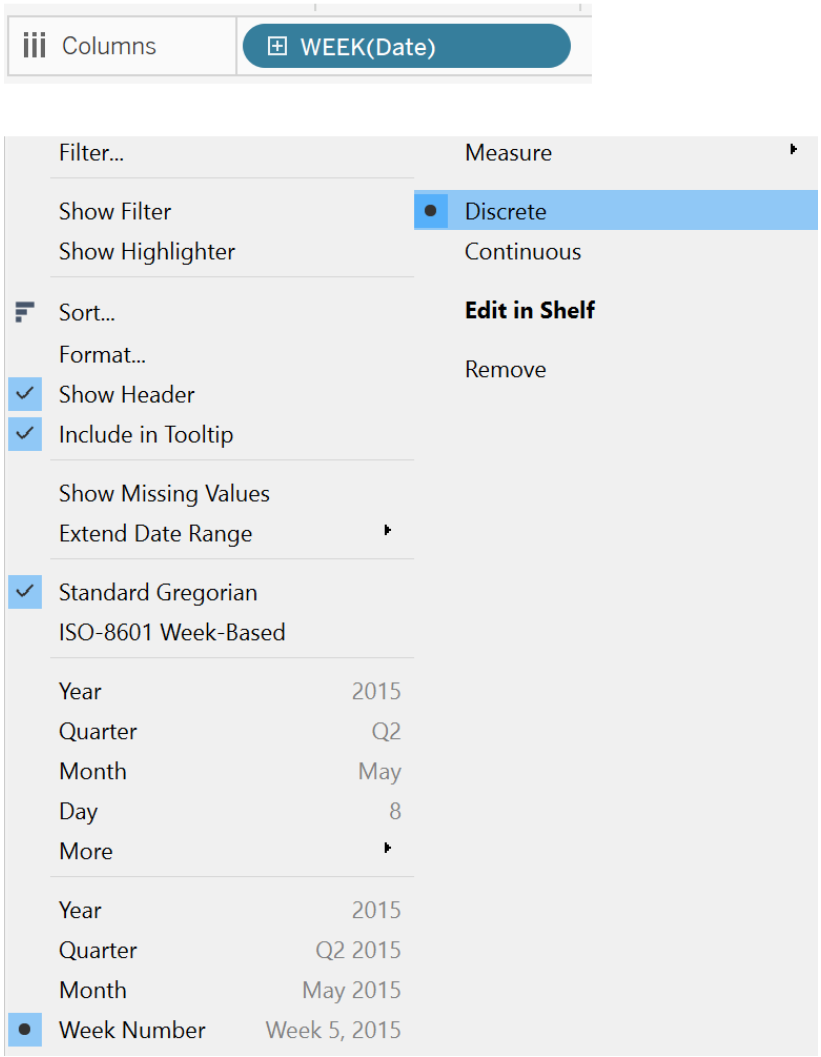


Figure 8.91: Changing the Date properties

3. Create a parameter that acts as the range input from the user. Right-click anywhere in the empty area of the data pane and select **`Create Parameter`**....

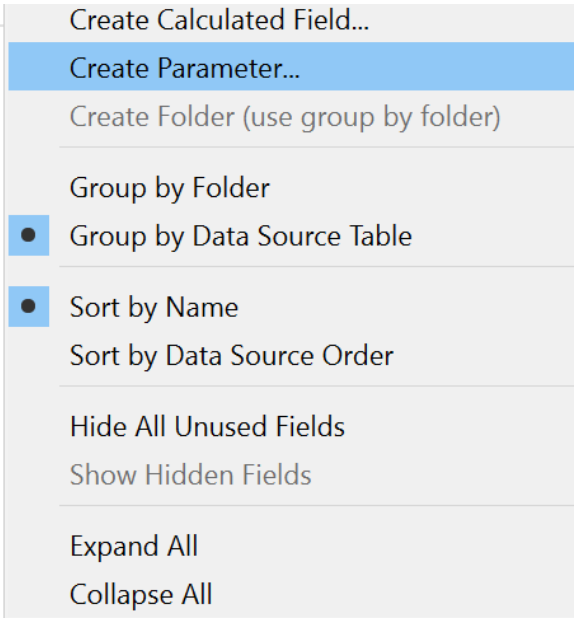


Figure 8.92: Creating a parameter

4. Edit the parameter properties as follows and click **`OK`**:

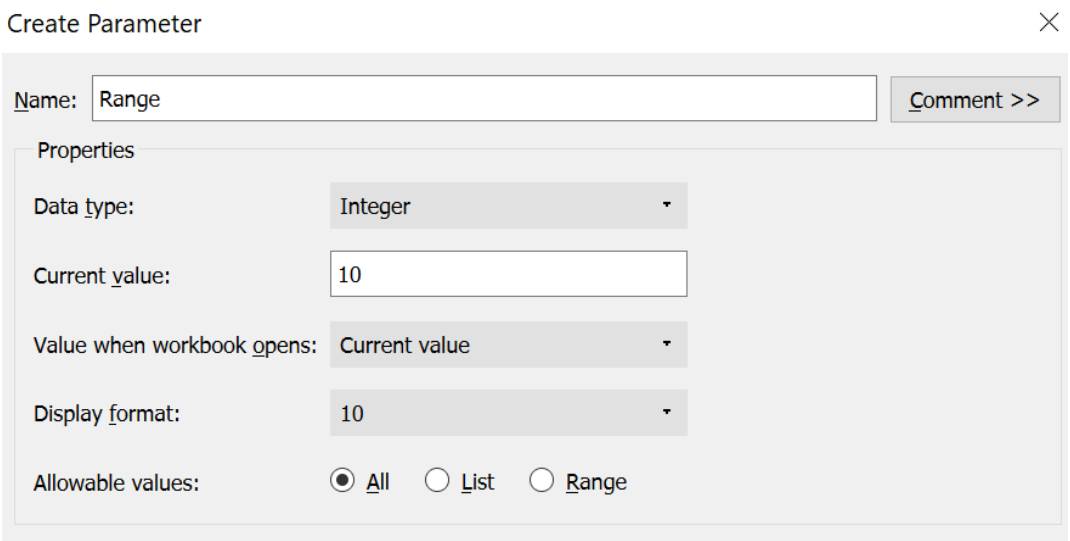


Figure 8.93: Modifying the parameter properties

5. Add it to the view by right-clicking on the parameter and checking **`Show Parameter Control`**.

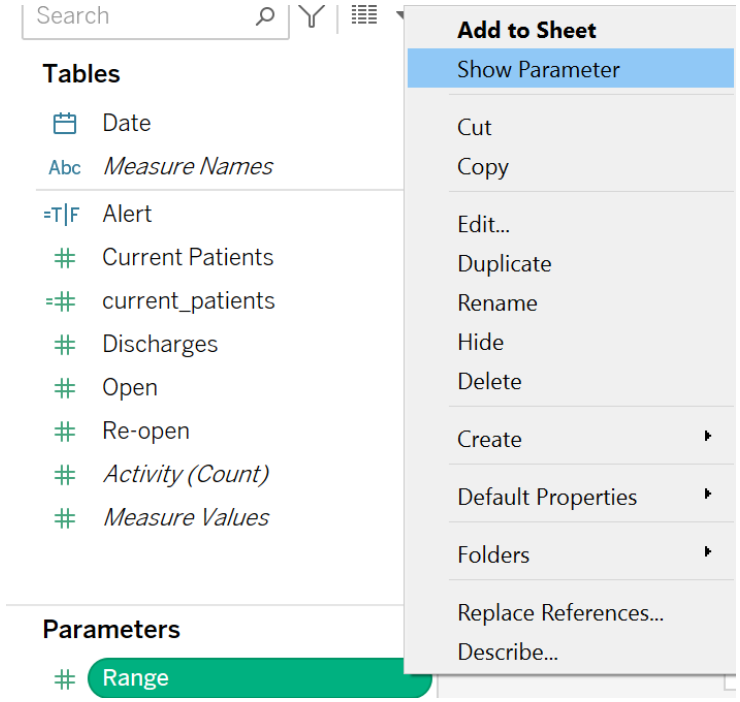


Figure 8.94: Adding a parameter to the view

6. Before using **`WINDOW_AVG`**, it's worth considering its syntax.

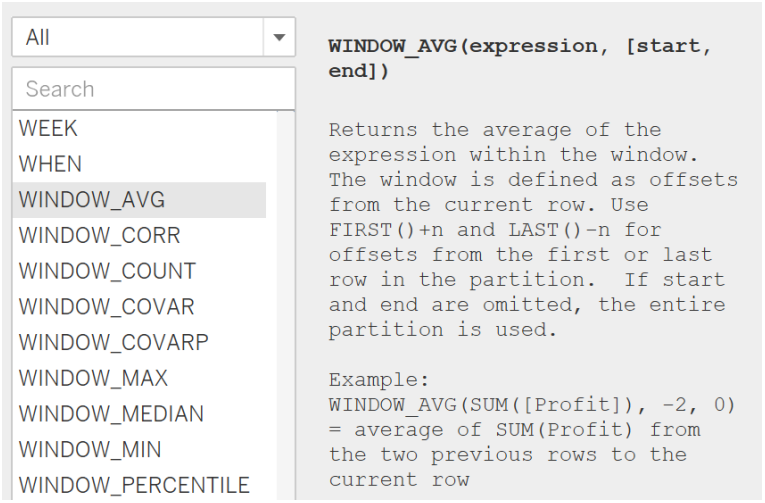


Figure 8.95: WINDOW_AVG syntax

As you see, **WINDOW_AVG** computes the average of an expression based on a window range. In this case, the window is the whole view, the expression is **RUNNING_SUM** of **current_patients**, the start is the last 10 weeks, indicated as **-10** weeks or the **Range** parameter, and the end is the current week, which is week 0. Accordingly, use this parameter as input to the **WINDOW_AVG** calculation and name it **avg_admitted**, as shown in the following figure:

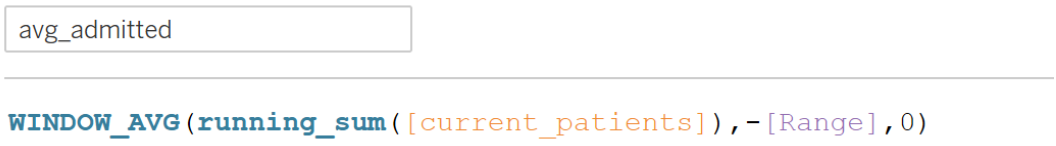


Figure 8.96: Creating WINDOW_AVG avg_admitted syntax

7. Add this calculation to the initial view on the **Rows** shelf.

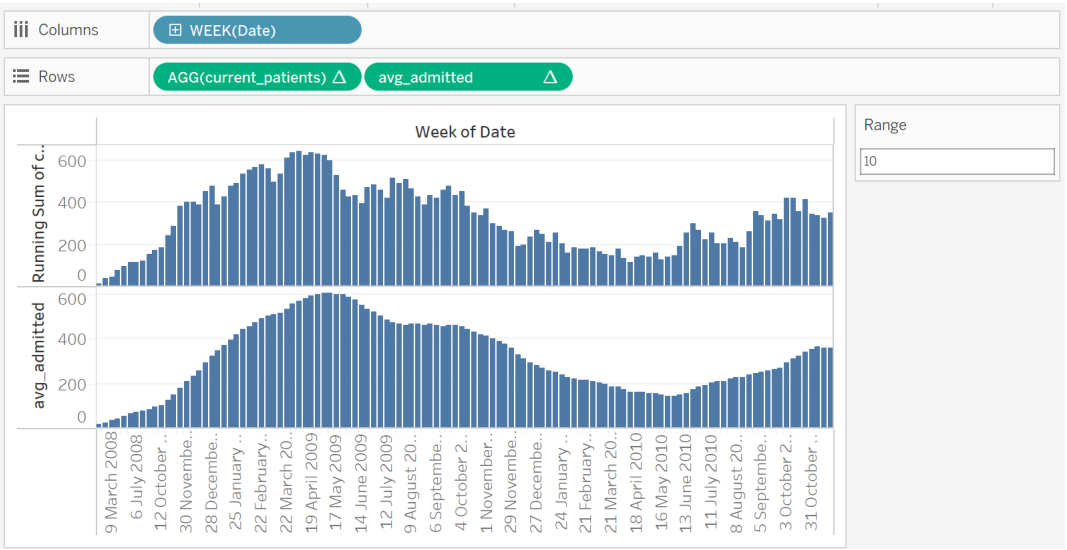


Figure 8.97: Adding avg_admitted to the view

8. Select the axis for **`avg_admitted`** and change it to **`Line`**, as follows:

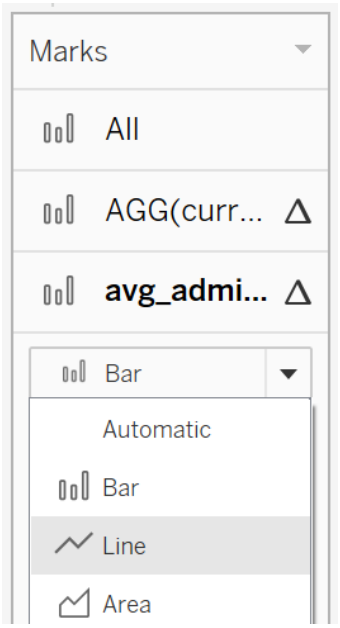


Figure 8.98: Changing avg_admitted marks from Bar to Line in the view

9. Now, right-click on the axis and select **`Dual Axis`** to merge the axes.

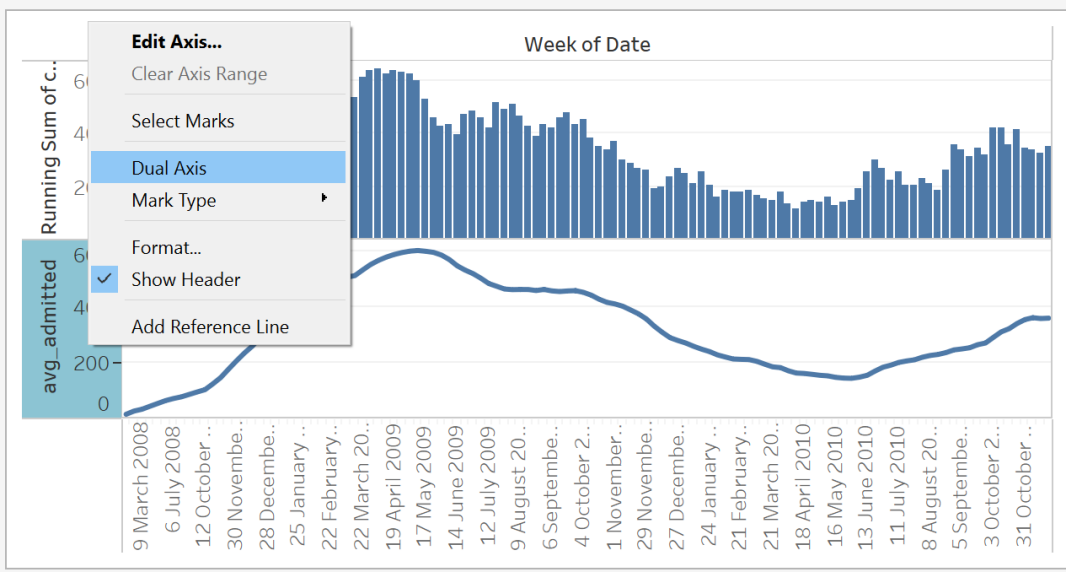


Figure 8.99: Creating a dual-axis

10. Synchronize both axes so they are on the same scale. Right-click on the right axis and select **`Synchronize Axis`**.

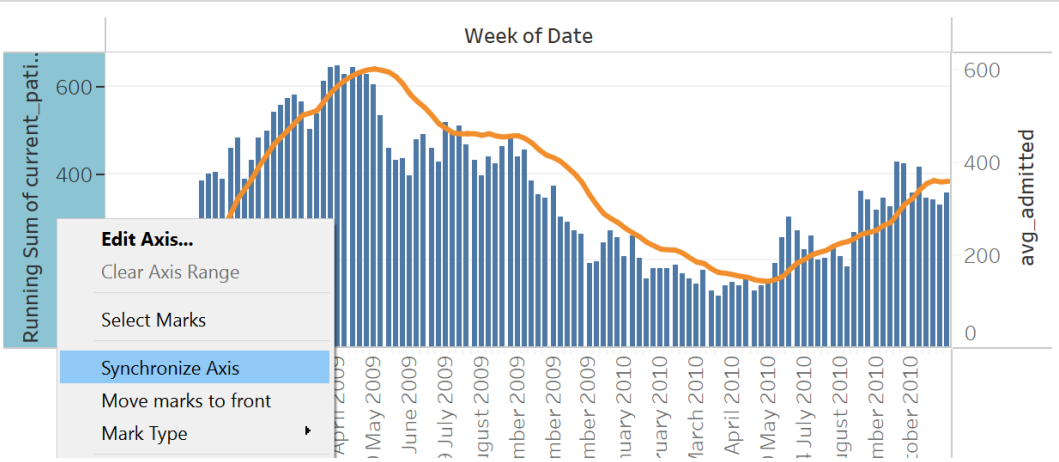


Figure 8.100: Synchronizing a dual-axis

11. Create an alert to check how the average number of patients compares to the 10-week average, as follows:

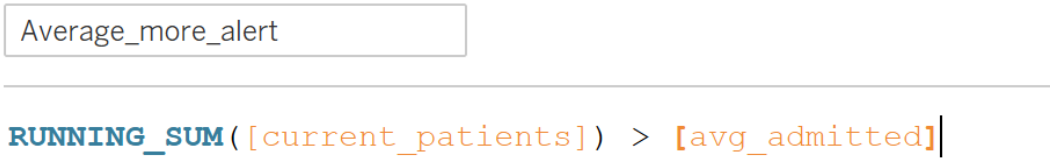


Figure 8.101: Creating an alert for the moving average range

Using this formula, you can check whether the **`RUNNING_SUM`** sum of **`current_patients`** is more than **`avg_admitted`**.

12. Add this to the view to overwrite the **`measure_values`** color.

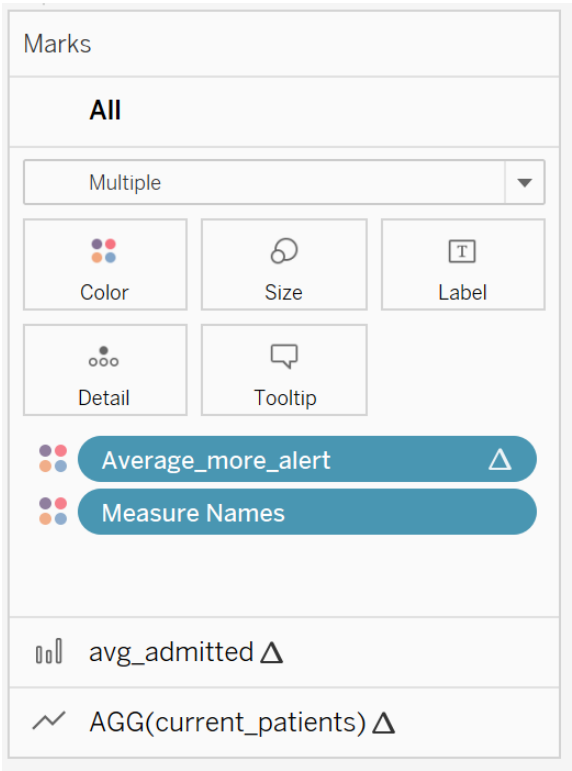


Figure 8.102: Adding an alert to Color

The final output will look like this:

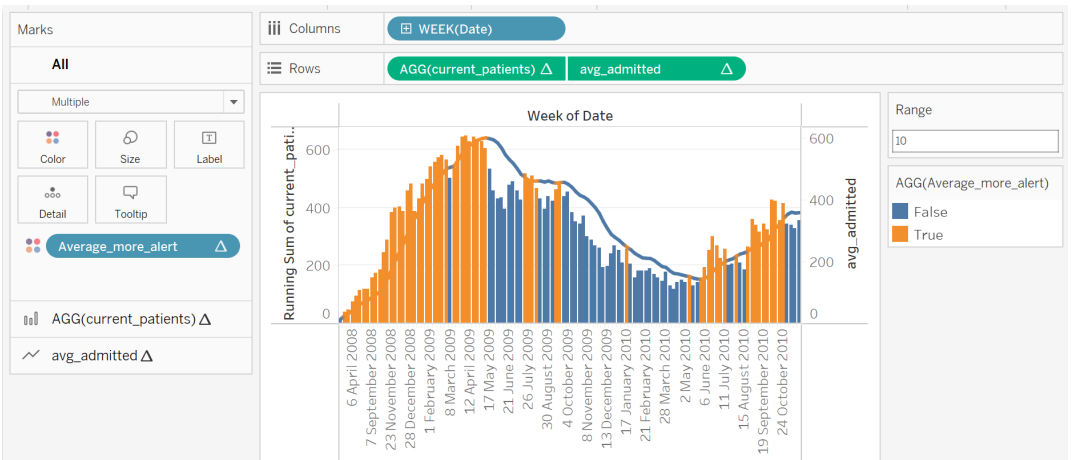


Figure 8.103: Final output

You can now see when the current admitted patient count has gone higher than the 10-week average. The range can be changed based on the requirement by changing the input. An interesting observation is the month of July, which had a higher-than-average number of patients for all of the previous 3 years, indicating the possibility for a similar occurrence for next July.

CHAPTER 9: DATA ANALYSIS: CREATING AND USING LEVEL OF DETAIL (LOD) CALCULATIONS

ACTIVITY 9.01: IDENTIFYING THE TOP-PERFORMING SALES EXECUTIVES

Solution:

1. Connect to the data using the **`Connect pane`**. Select **`Microsoft Excel`** and navigate to the location where you saved the *Activity1.xlsx* file. Click **`Open`** to add it to Tableau Desktop.
2. Now drag the **`State`**, **`City`**, **`Order ID`**, and **`Sales Executive`** dimensions on the **`Rows`** shelf. Add the **`Sales`** measure to the **`Columns`** shelf. The default aggregation for **`Sales`** will be **`SUM`**; keep it that way:

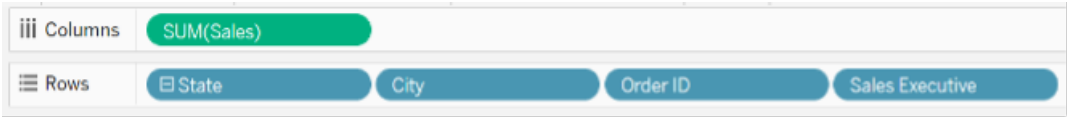


Figure 9.37: Adding fields to the view

3. Add **`SUM(Sales)`** to the **`Label`** card to show the bar labels:

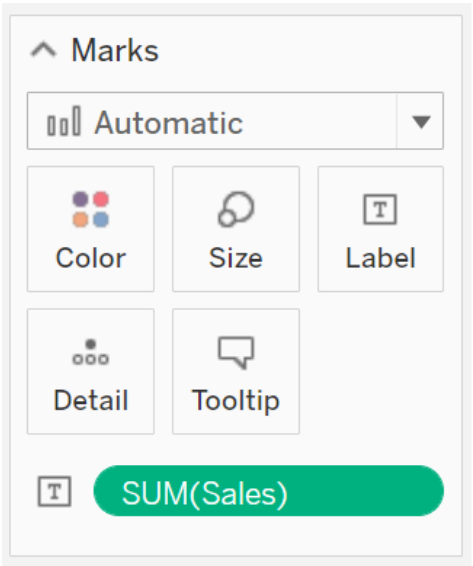


Figure 9.38: Adding SUM(Sales) to the Label card

4. Now, sort this data in descending order of sales per city per state, using the quick sort option:

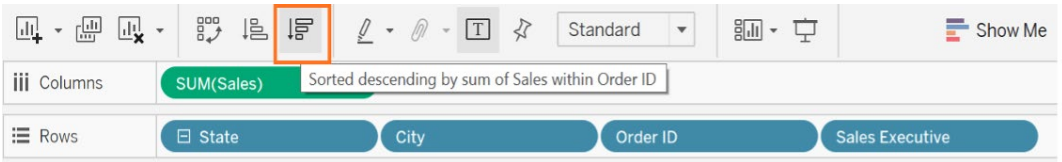


Figure 9.39: Sorting the view

5. You have now created an initial view. List the orders with the highest sales values in each city and state, along with the sales executive responsible for each order:

Figure 9.40: Initial view

You can clearly see that in **`Auburn`**, **`Maurice Satty`** has the order with the highest sales value. Similarly, **`Stewart Carmichael`** is the top sales executive for **`Decatur`**.

6. Now, your goal is to display only those sales executives with the highest sales value for an order. Create a **`FIXED`** LOD by using **`State`**, **`City`** and the **`Max(Sales)`** value. Using this, you get the maximum sales value per state and city:

Figure 9.41: LOD calculation expression

7. You can use this LOD calculation to identify the sales executive whose sales value matches this LOD calculation value. For this, create the calculation shown in the following figure:

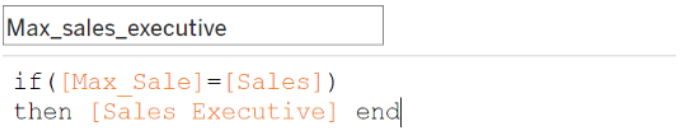


Figure 9.42: Using a LOD calculation with another calculation

Here, the (**`Max_Sale`**) LOD calculation is compared with the sales of each Order ID. When a match is found, the value of Sales Executive is saved. A null is saved if the condition is not satisfied.

8. Add this calculation to **`Filters`** and exclude the **`Null`** value. Click on **`OK`** to add this filter:

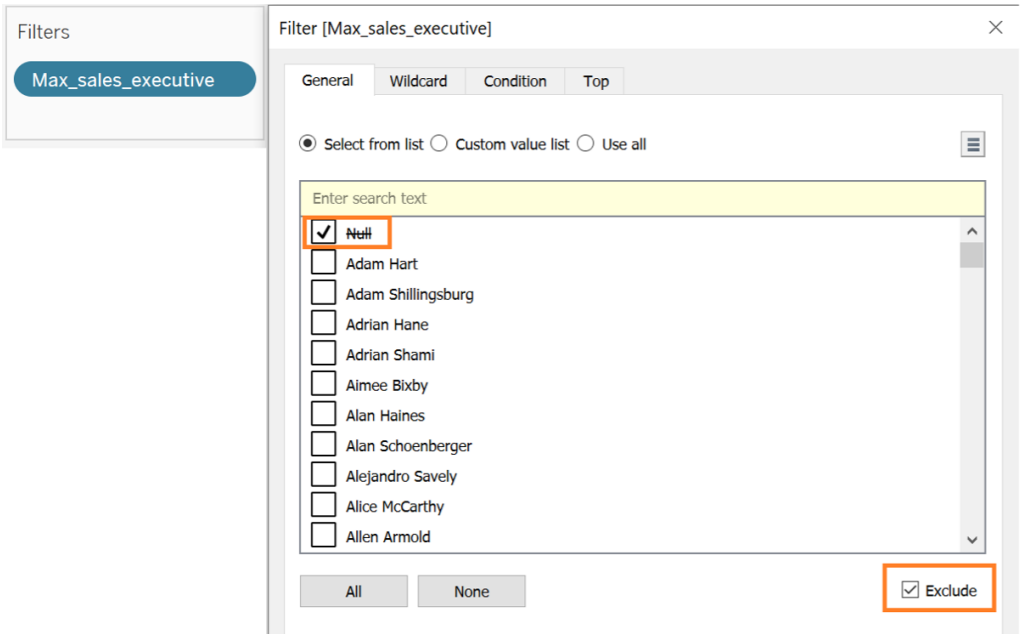


Figure 9.43: Excluding the Null value from the filter

You now get only the sales executive whose sales value matches the maximum sales value per city and state. You can verify the results using the initial view. The following figure shows the final view:

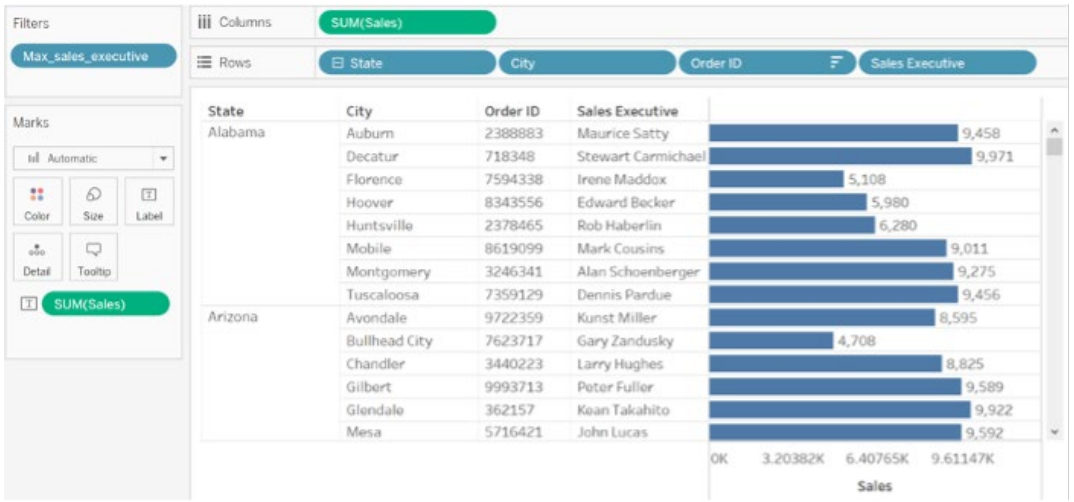


Figure 9.44: Final solution for Activity 1

ACTIVITY 9.02: PERFORMING A COMPARATIVE ANALYSIS

Solution:

1. Connect to the data using the **Connect** pane. Select **Microsoft Excel**, and navigate to the location you have saved the *Activity2.xls* file in. Click **Open** to add it to Tableau Desktop.
2. Drag **Name** to the **Rows** shelf. Add the **Cook Time** measure to the **Columns** shelf. The default aggregation will be **SUM**. Keep it that way, as you have only one record per dish:

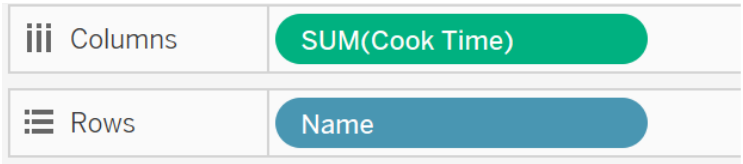


Figure 9.45: Adding fields to the view

3. Add **Course** to the **Filters** shelf and select **dessert** as the selected value. Click **OK** to add this to **Filters**:

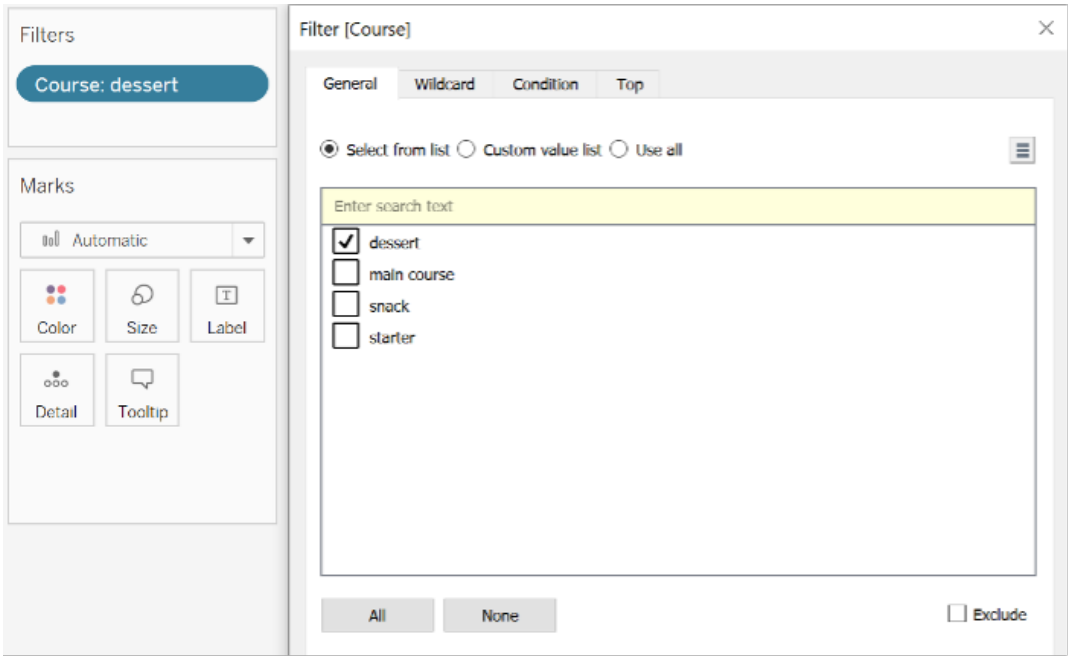


Figure 9.46: Adding a filter on dessert

4. Add **`SUM(Cook Time)`** to the **`Label`** card:

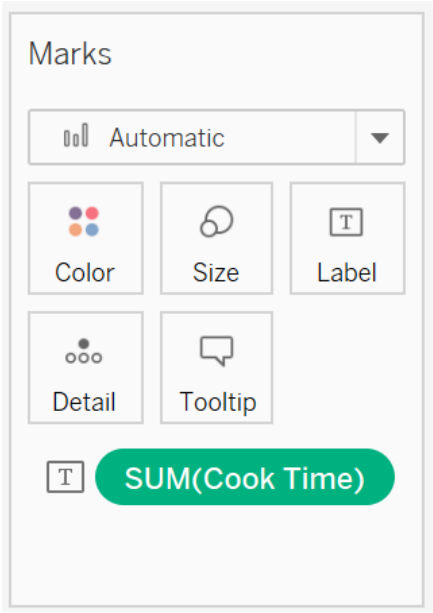


Figure 9.47: Adding Cook Time to the Label card

5. This becomes your initial view for analysis:

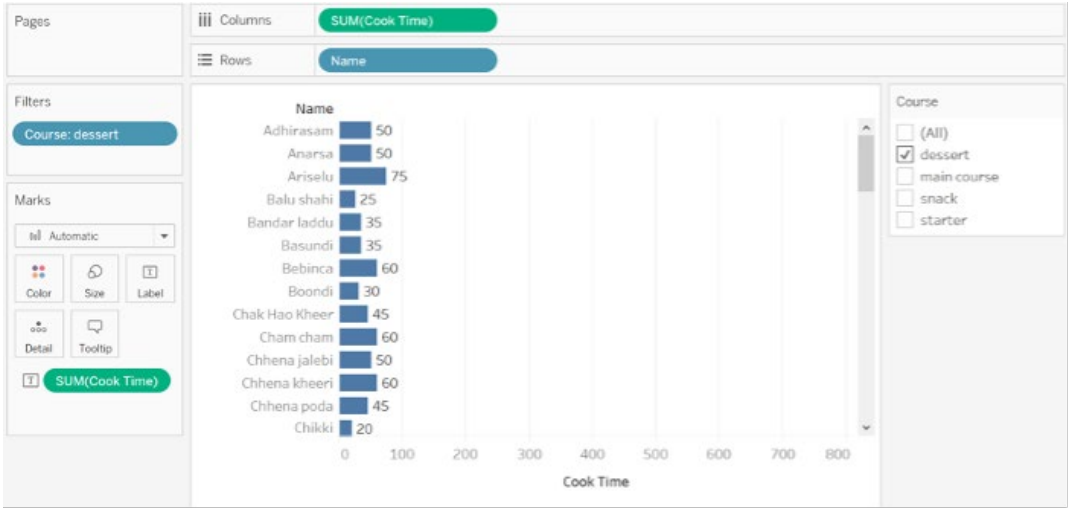


Figure 9.48: Initial view

In this view, you can observe the cooking times of various desserts.

6. Create a calculation that keeps just the dessert names and removes all other course type names. A null value will be stored if the course type is not **`dessert`**:

Figure 9.49: Dessert-only calculation expression

7. Use this calculation to create a parameter with only desserts as selection values by right-clicking on this calculation, and selecting **`Create`** | **`Parameter`**...:

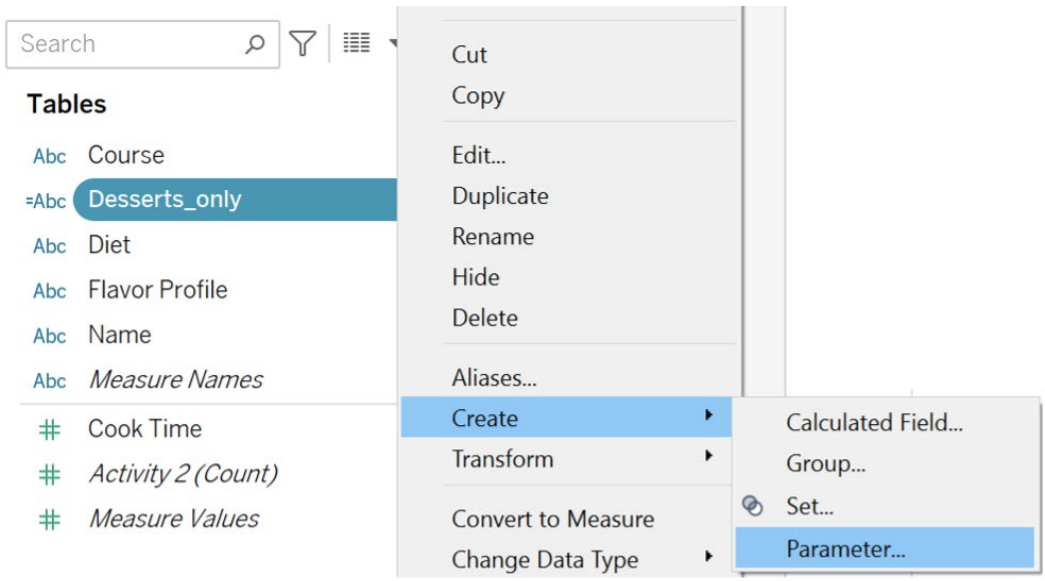


Figure 9.50: Creating a parameter

8. Rename this parameter as follows and click **OK**:

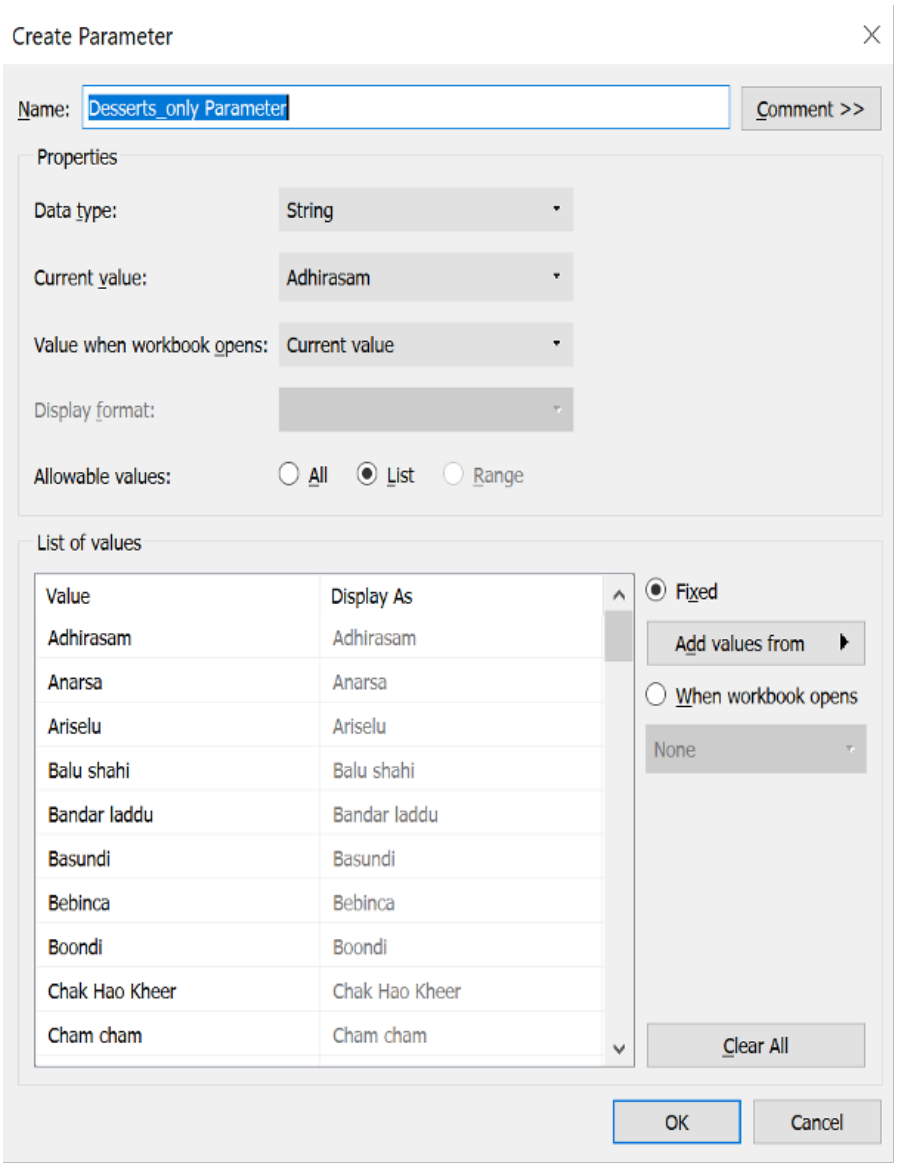


Figure 9.51: Updating the parameter details

9. You should be able to see the parameter in the data pane:

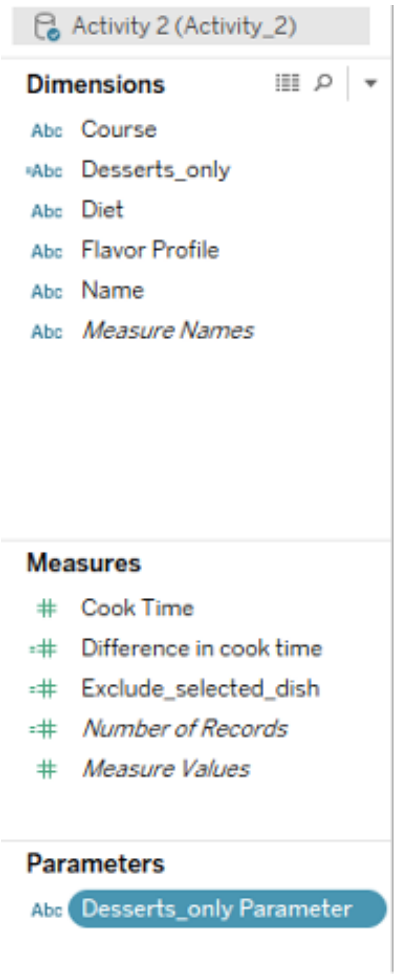


Figure 9.52: Viewing the parameter in the data pane

10. Right-click on this parameter and select **`Show Parameter Control`** to add this as a selection in the view:

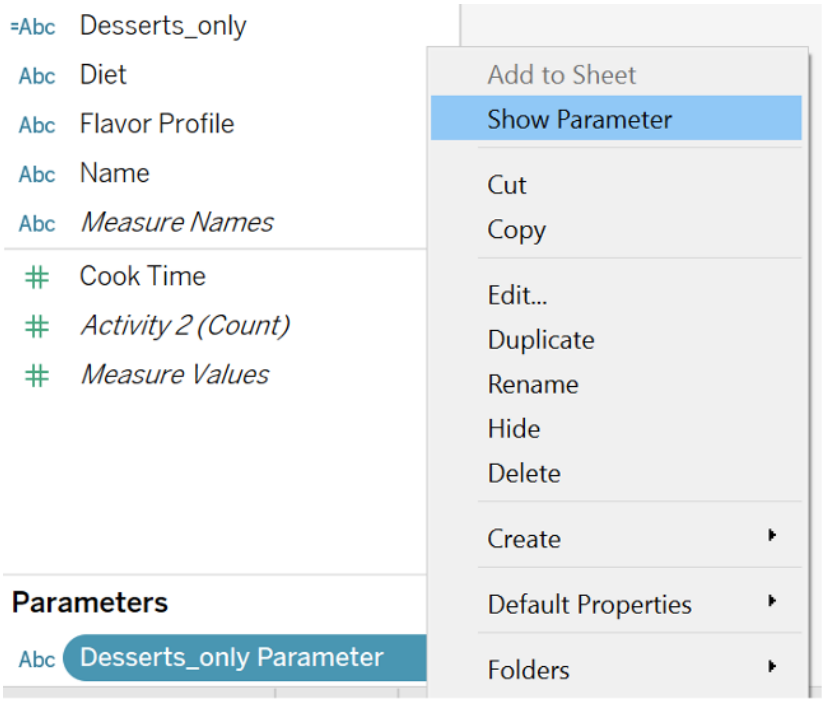


Figure 9.53: Adding the parameter to the view

11. To create a calculation, right-click on the parameter and select **`Create Calculated Field...`**. You can also right-click on any field to do the same:

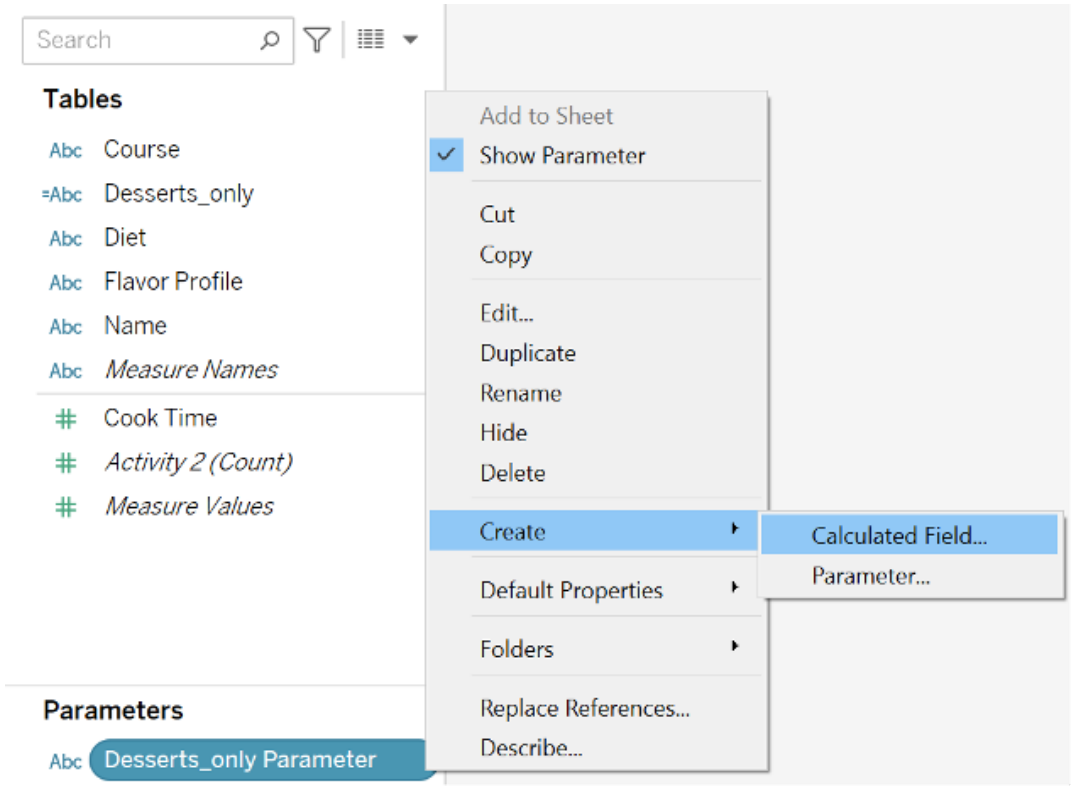


Figure 9.54: Creating a calculated field from the data pane

12. Create an **EXCLUDE** LOD calculation that only keeps cooking times for dishes of the selected parameter, and excludes cooking times of other dishes. Create a calculation as follows:

Figure 9.55: Expression for the EXCLUDE LOD calculation

The expression first excludes the names in the view. The initial view consists of the dish names and the cooking time. So, this can be excluded as the calculation must only consider the cooking time of the selected dish. Then, the dish name is compared with the selected parameter dish. If there is a match, the cooking time is saved; otherwise, the value is saved as 0.

13. Create another calculation that calculates the difference between the cooking time of all other dishes and the selected dish, as follows:

Figure 9.56: Cook Time difference calculation expression

14. Add this calculation to the **Rows** shelf, as follows:

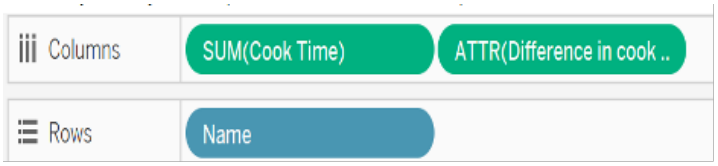


Figure 9.57: Adding a difference calculation field to the view

Make sure to add it to the **`Label`** card as well:

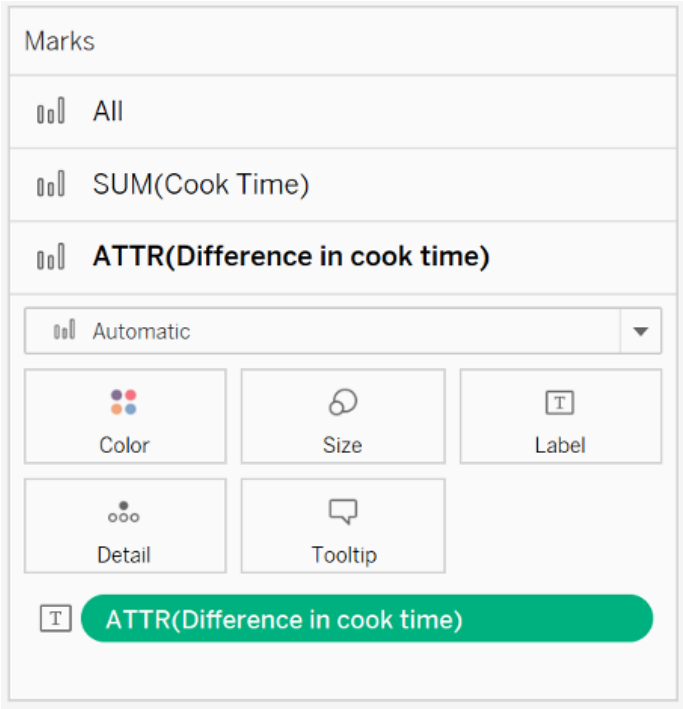


Figure 9.58: Adding the difference calculation field to the Label card

This is what the final view looks like:

Figure 9.59: Final view with various fields and the selection parameter

You can now change the selection in the dropdown, and compare the cooking times of the other dishes. A positive difference means that the other dish takes longer to cook compared to the selected dish, whereas a negative difference indicates the opposite.

15. Try comparing the cooking times of various dishes by changing the selected dish in the parameter selection.

    The final view should look like the following:

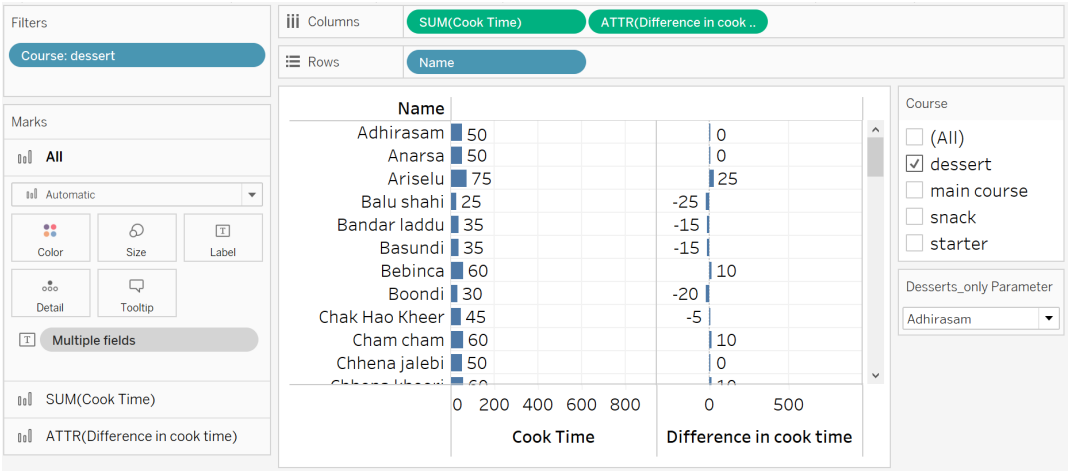


Figure 9.60: Final solution

CHAPTER 10: DASHBOARDS AND STORYBOARDS

ACTIVITY 10.01: BUILDING A COMPLETE DASHBOARD

Solution:

1. Open and load the **`European Airports 2015`** data into your Tableau instance.
2. Familiarize yourself with the data and the dimensions/measures.
3. Create a new worksheet and drag **`2014 Passengers`** onto the **`Text`** marks card and format the marks card as follows:

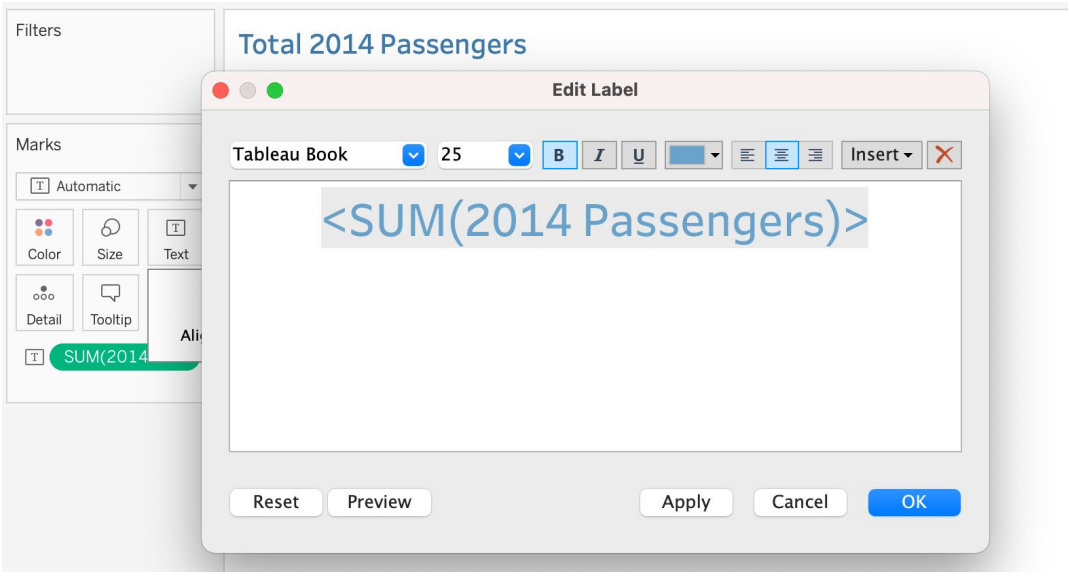


Figure 10.74: Formatting the total 2014 passengers worksheet

4. Repeat the same steps for the **`2015 Passengers`** scorecard.
5. Create the third worksheet and a new calculated field, naming this field **`Passenger Growth/Decline`**.

6. Essentially, if you want a percentage change from 2014 to 2015, use the following formula:

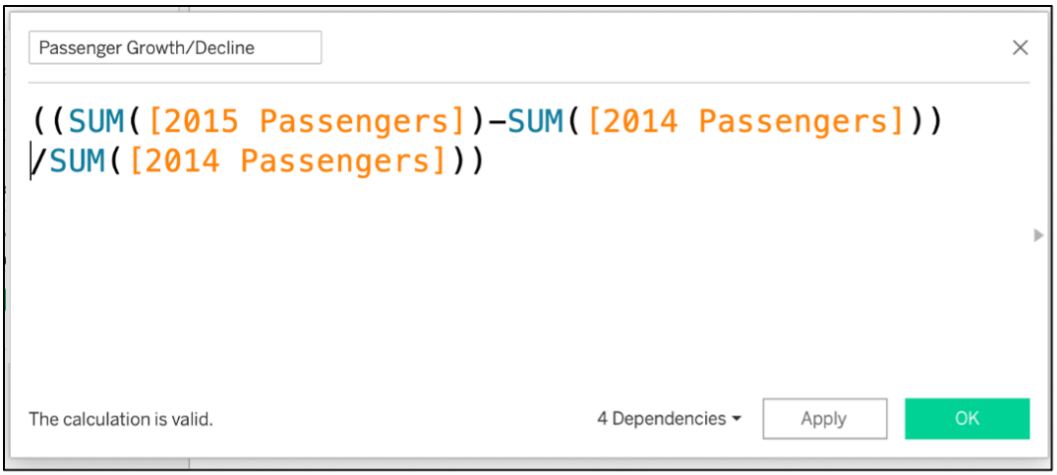


Figure 10.75: Calculating the percentage difference from 2014 to 2015

7. Drag the newly created **`Passenger Growth/Decline`** field onto the **`Text`** marks card and format the text as you did for other scorecard metrics. Rename the worksheet to **`Delta`** or something descriptive.

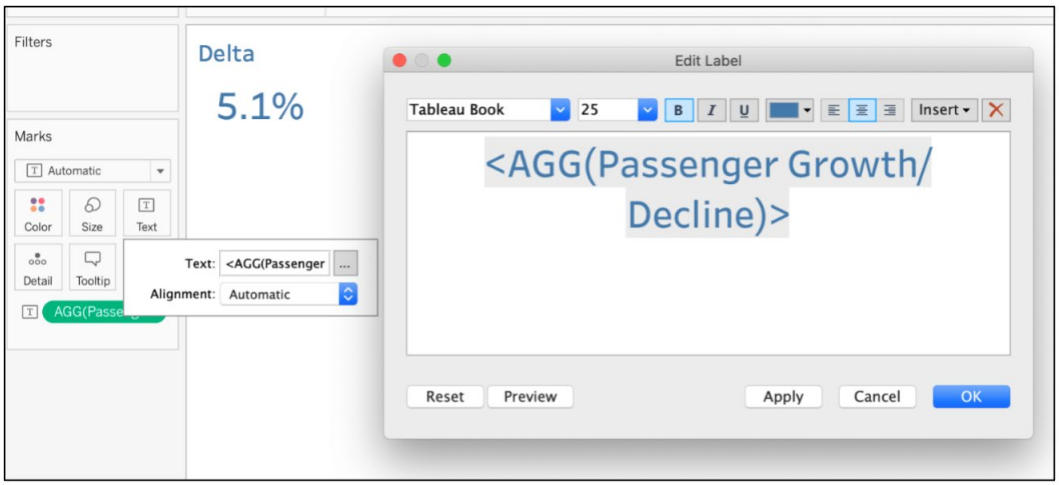


Figure 10.76: Formatting the scorecard worksheet

8. Create the **`Top N Airports by 2014`** worksheet. For that, you also need to create a parameter and add a **`Top N`** filter to the view. Create a new worksheet and name the worksheet **`Top N Airports by 2014`**.

9. Right-click on the data pane to create a new parameter with the following settings:

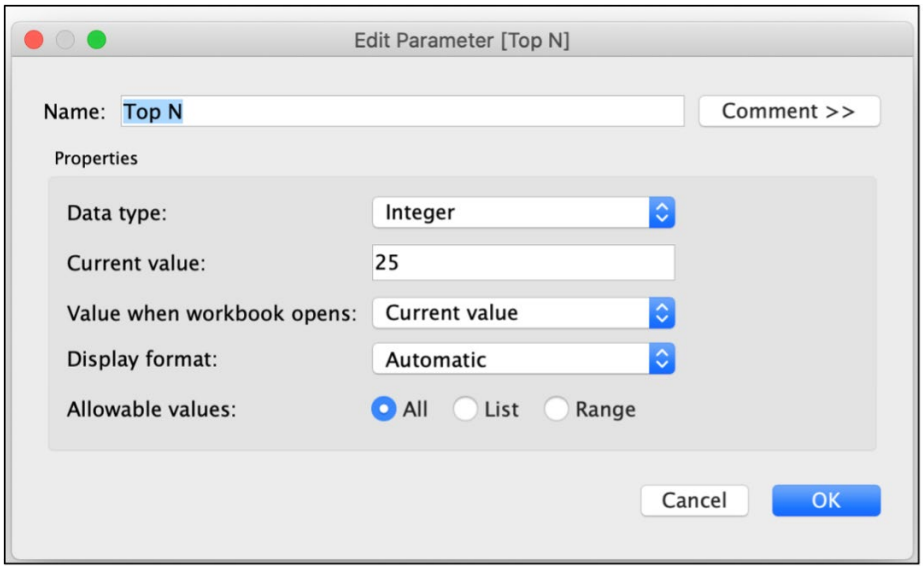


Figure 10.77: Creating a top N parameter

10. Next, add **`Airport Name`** to the **`Rows`** shelf, and **`2014 Passengers`** and **`2015 Passengers`** to the **`Columns`** shelf. Sort both columns appropriately as shown below:

Figure 10.78: Two bar charts using two dimensions

11. Convert both charts into **`Dual Axis`** and synchronize the axis by right-clicking on one of the green pills and selecting **`Dual Axis`**, as shown here:

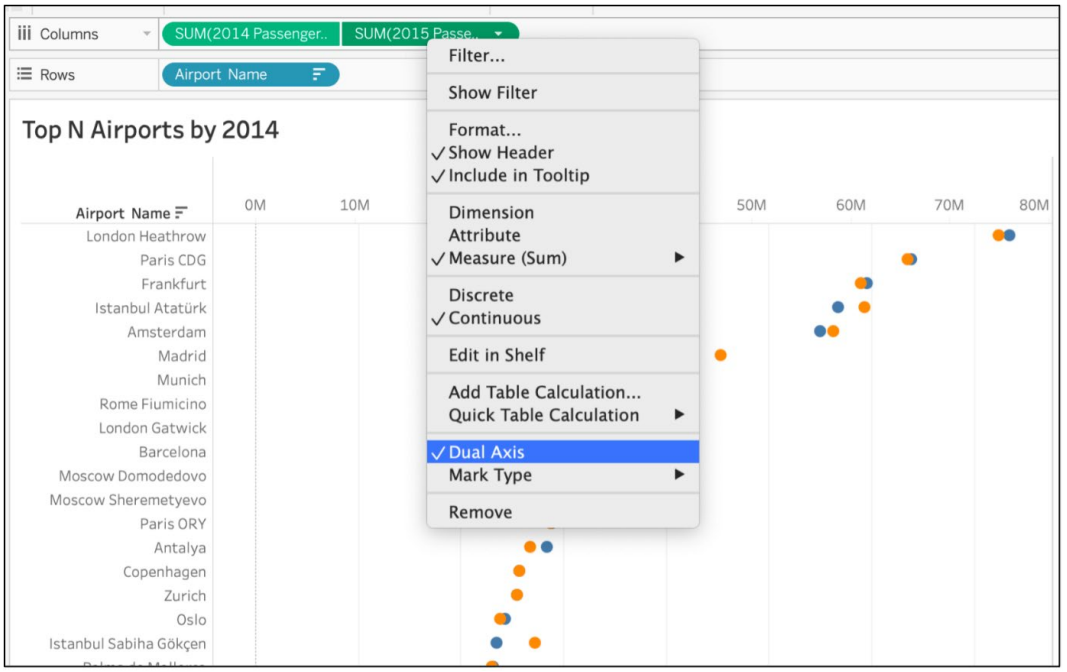


Figure 10.79: Converting the two charts to dual axis

12. Manually select the marks type for each of the axes, for the **`2014 Passengers`** metric, and select **`Bar Chart`** as our preferred chart type. For **`2015 passengers`**, select **`Circle`** as the preferred mark type, as shown here:

Figure 10.80: Changing the marks type to Circle

13. Now, add the **`Airport Name`** dimension to your filters so that you can filter using **`Top N`** parameters. Drag **`Airport Name`** to the filters card, click on **`Top`** -> **`By Field`**, and use the following settings:

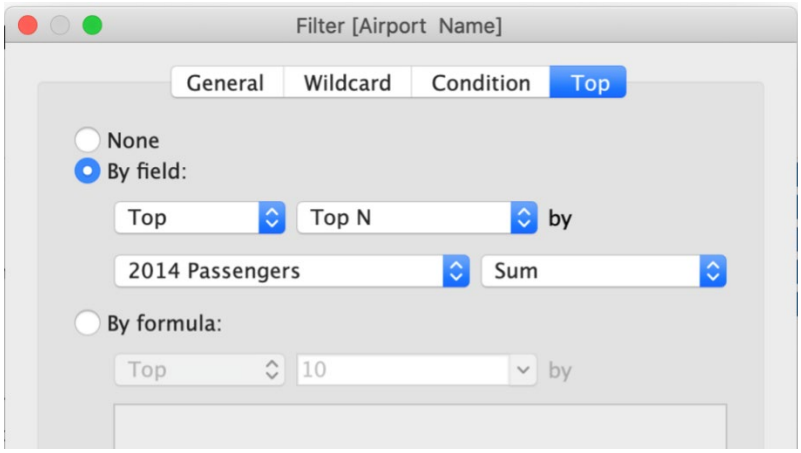


Figure 10.81: Adding the top N filter

14. Next, show the parameter in the view to filter our worksheet by **`Top N`**, as per the data in the parameter field:

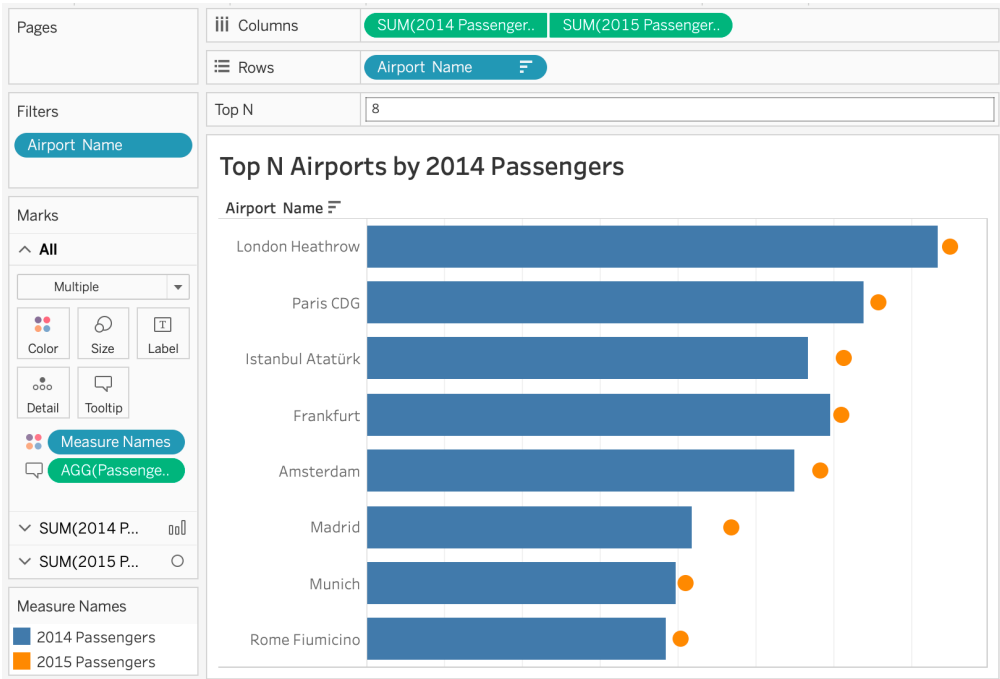


Figure 10.82: Showing a parameter in the view

15. You have now completed all the individual pieces. Aggregate all these pieces into the dashboard.
16. Create a new dashboard and drag the tiled **`Vertical`** object to the view first.
17. Drag another tiled **`Vertical`** object to the top of the previous object and then drag a tiled **`Horizontal`** object to create sections for your header.
18. Now, drag the **`Text`** object to the object created previously and then drag the **`Image`** object to the right of the **`Text`** object to add your own branding.

Figure 10.83: Adding a header to the dashboard

19. Create the middle section of the dashboard, drag another **`Vertical`** tiled object below the header, and then overlay the **`Vertical`** object with another **`Horizontal`** tiled object for better control.
20. Drag all three scorecards one after another to the right of each other, as shown here:

Figure 10.84: All scorecards under one object

21. You want your scorecards to be distributed evenly across the middle section, so click on the drop-down arrow and select **`Distribute Content Evenly`**.

22. Next, drag the **`Top N Airports by 2014`** sheet below the middle section. This should add the legends as well as **`Top N`** parameters to a separate section on the right (if not, add them manually by selecting the worksheet and adding these parameters/legends).

Figure 10.85: Adding the last worksheet to the dashboard

23. You want to place the **`Top N`** parameter and the legend between the scorecards and the worksheets themselves, so drag the parameter right in between the middle section and the worksheet.

24. Next, drag the legend similarly and place it to the right of the parameter that you incorporated in the previous step.

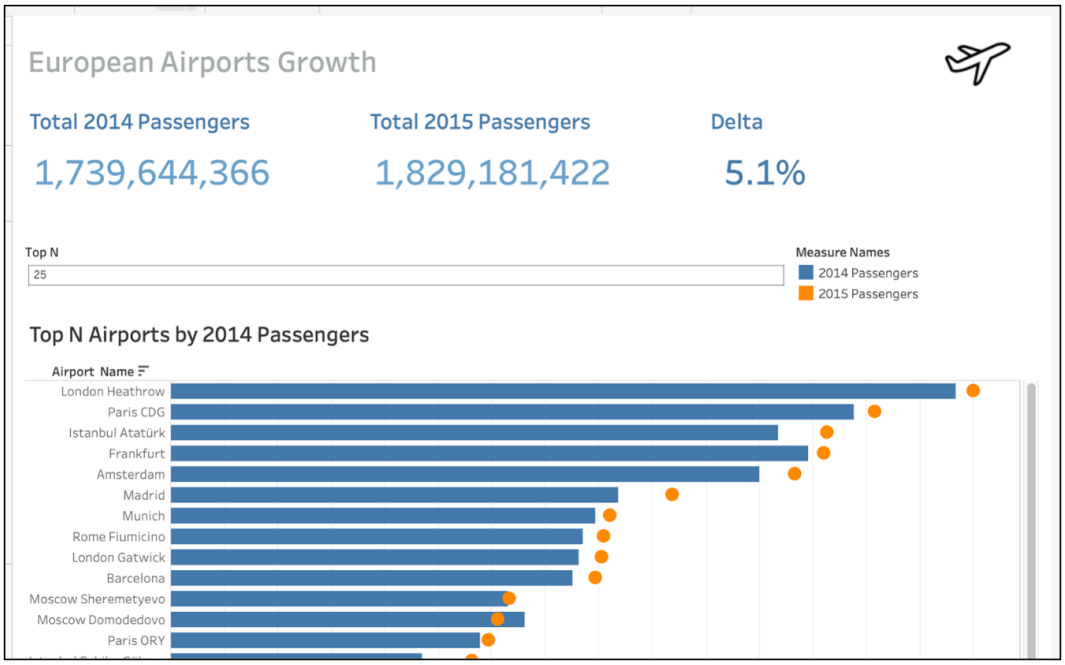


Figure 10.86: Placing legends

25. To finalize the dashboard, add more formatting to the view. Right-click on the **Legend** title and hide the title. In each of the scorecard sheets, add a thin solid line border to differentiate between each sheet, as shown here:

Figure 10.87: Adding a thin line to the scorecard worksheet

26. Next, fit each of your scorecard worksheets to Entire View so that the sheets can make better use of the whitespace.

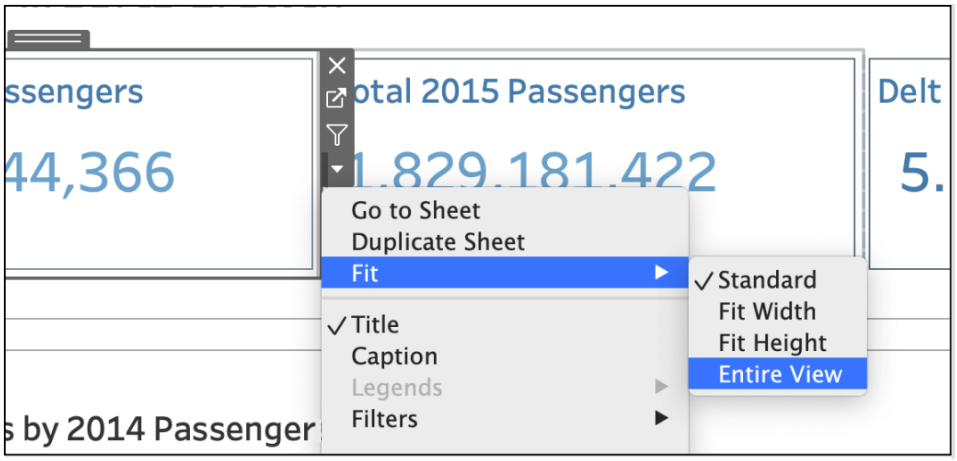


Figure 10.88: Selecting the entire view to fit the scorecard worksheet

27. To test the dashboard out, enter any other number in the **`Top N`** parameter.

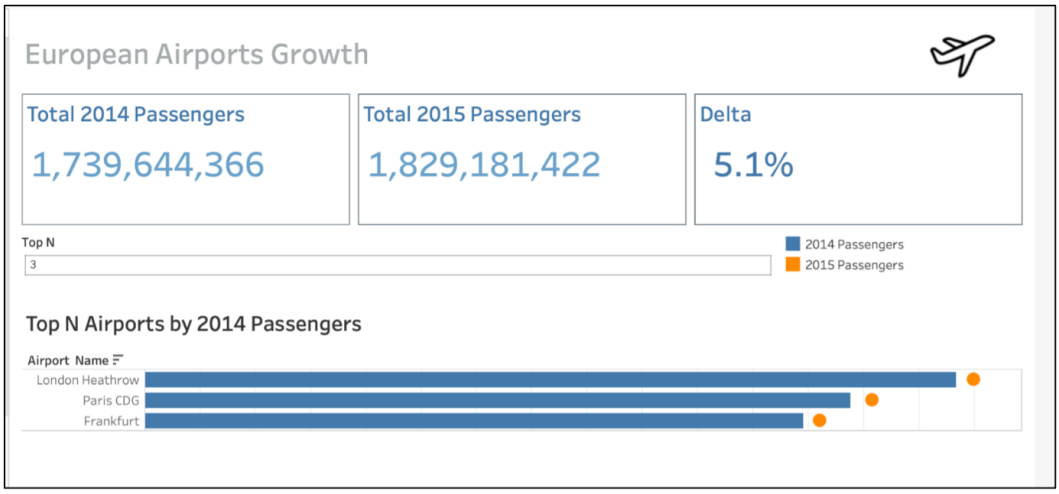


Figure 10.89: European airports dashboard with the top 3 airports example

In this activity, you added a couple of extra elements to your dashboard using a text-based parameter that was dynamic enough to filter the view with the Top N number of the stakeholder's choice. You also played with color, text, and sizes, as well as placement for better readability.

CHAPTER 11: TABLEAU INTERACTIVITY: PART 1

ACTIVITY 11.01: TOP N COUNTRIES USING PARAMETERS, SETS, AND FILTERS

Solution:

Link to dataset: http://packt.link/eLmSX.

1. Connect the **`WorldIndicator.hyper`** dataset downloaded from the project/book folder in Tableau.
2. Drag **`Country/Region`** to the **`Rows`** shelf and **`SUM[Energy Usage]`** to the **`Columns`** shelf to create the bar chart and sort by descending order.

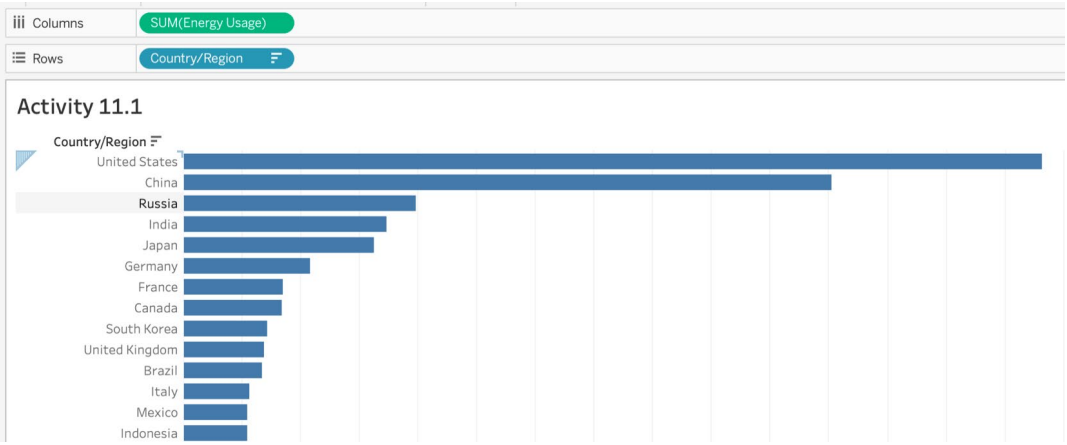


Figure 11.114: Energy usage by country

In the next step, you want to give your end users the ability to select the top N countries by energy usage, so that your view eventually reflects those changes in the bar chart. To accomplish that, you will create an integer list parameter first and later use the parameter in the set to appropriately color your bars, as shown in the final expected output (*Figure 11.111*).

3. Right-click on the arrow in the **Dimensions** pane and click **Create Parameter**. As mentioned, since users will be selecting **Top N Countries**, you want your **Data type** to be **Integer** and **Allowable values** to be **List**. Add **5**, **10**, **15**, and **20** as **Top N Countries** values in the parameter as shown here:

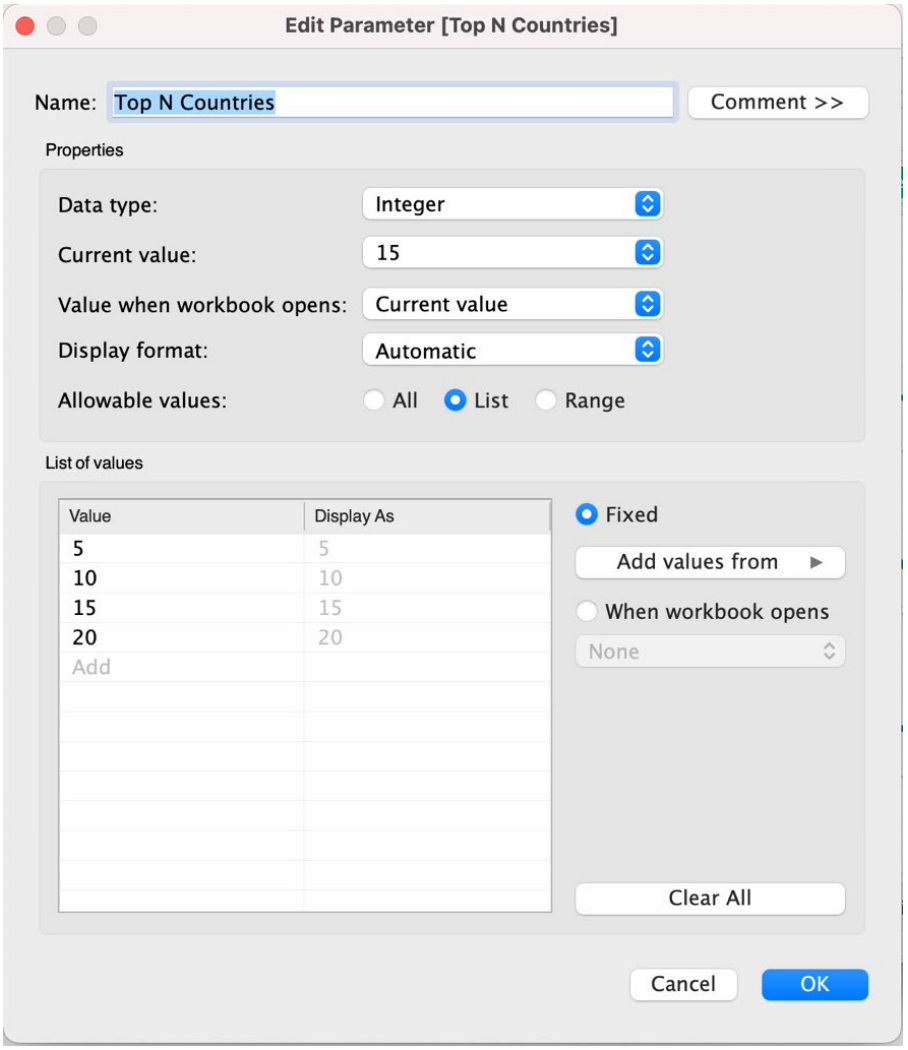


Figure 11.115: Creating the Top N Countries parameter

Now, as previously discussed, parameters by themselves are not of much use if you don't use the placeholder variables in calculated fields, sets, filters, or reference lines. You want your view to update the colors based on the number of countries your end users select. The best way to achieve that will be via a set.

4. Right-click on the **Country/Region** dimension and navigate to **Create** | **Set...**. Since you want to show **Top N Countries**, use the **Top** tab and select **Top countries by SUM[Energy Usage]**. Instead of a fixed top N value of 5 or 10 and so on, use the parameter created in the previous step to dynamically update the set depending on the end user's input. Select the **Top N Countries** parameter from the dropdown besides **By field** – **Top**. The **Create Set** window should resemble the following:

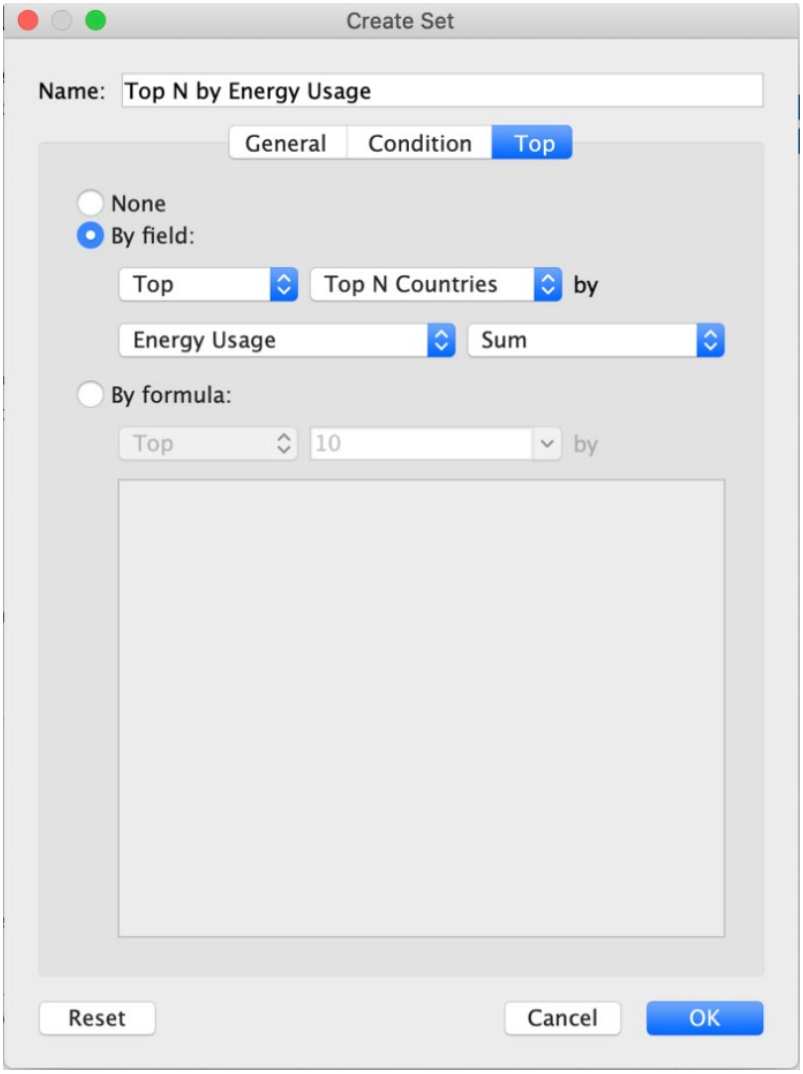


Figure 11.116: Creating the Top N by Energy Usage set

5. Add the set you just created to the **`Color Marks`** shelf card. Change the **`In/Out`** color according to your preference.

6. Press *Command* and drag **`Year`** on a Mac or *Ctrl* and drag **`Year`** in Windows to the **`Filters`** shelf, then select **`Year[Years]`** continuous as shown here and select all values from the list:

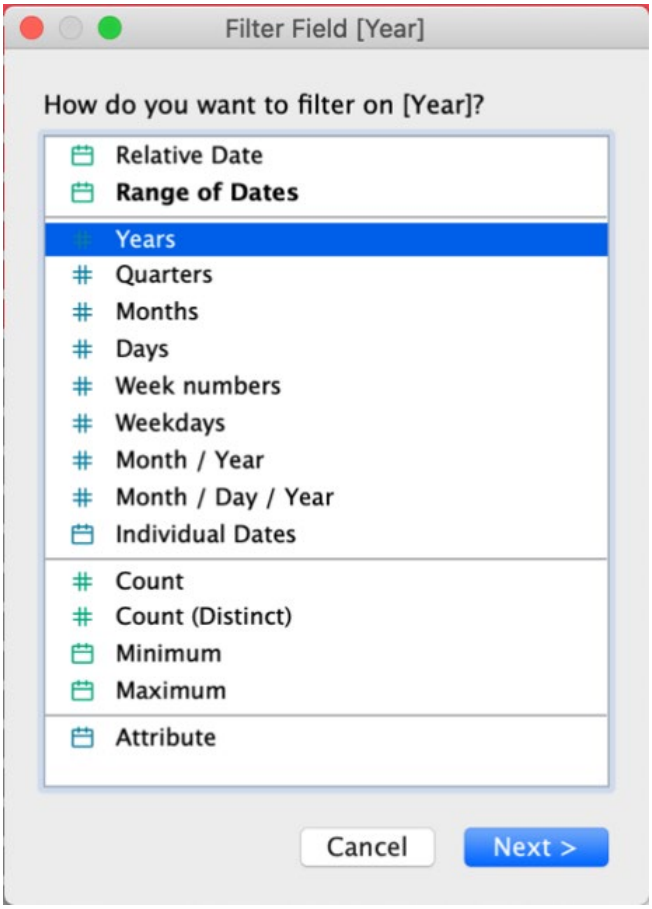


Figure 11.117: Using discrete years as a quick filter

7. From the **`Filters`** shelf, right-click **`Year(Year)`** and click on **`Show Filter`**. By default, Tableau assumed that you wanted **`Multi Values (list)`**, but since, as per the requirement, you want **`Single Value (dropdown)`**, you'll need to select this from the dropdown, as shown here:

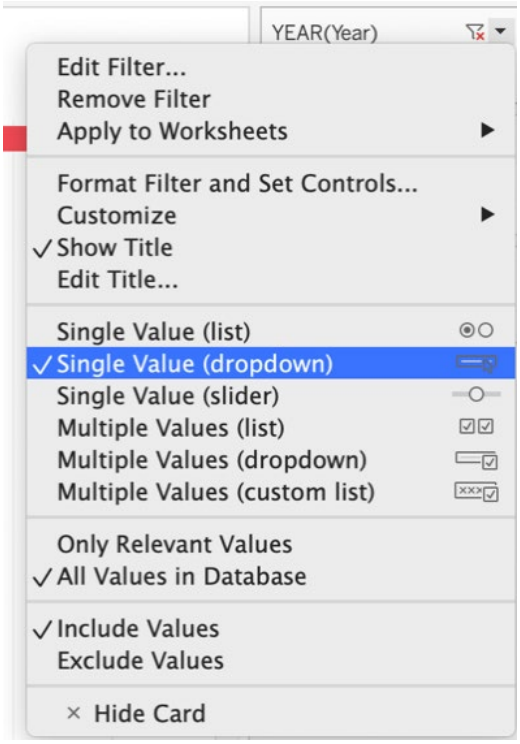


Figure 11.118: Selecting Single Value for the date filter

8. To show the parameter that you created previously in your view, right-click on the **`Top N Countries`** parameter from the **`Parameter`** shelf and select **`Show Parameter Control`**. Here is the current view:

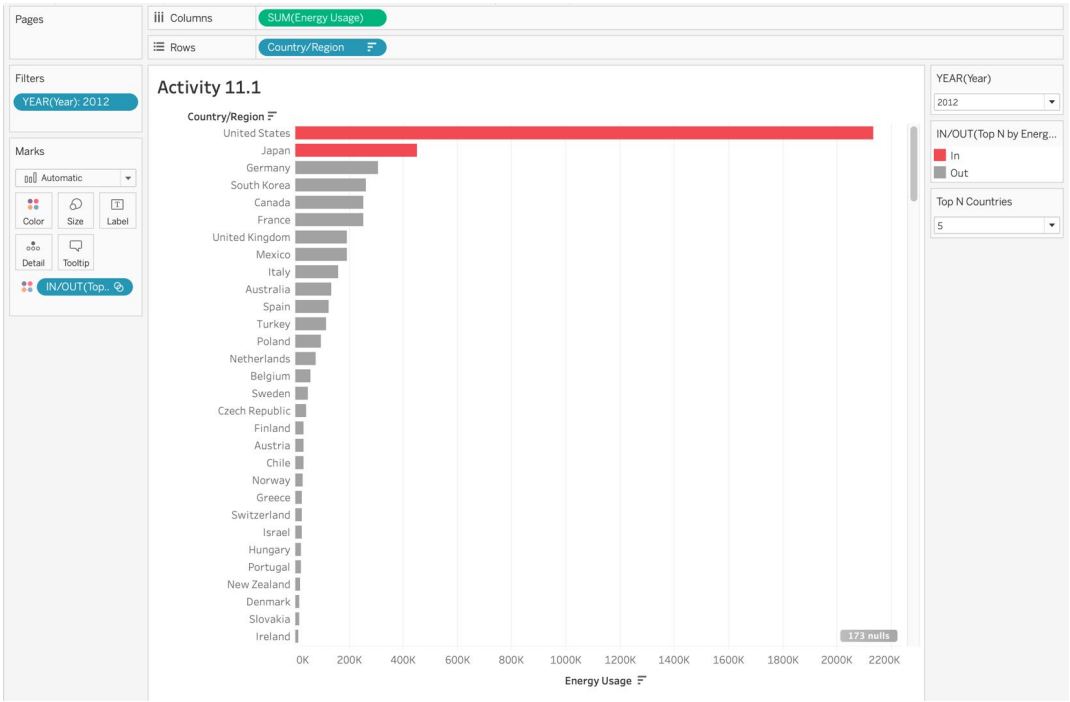


Figure 11.119: Non-context view of the data

Note that the parameter says **`Top 5 Countries`**, though there are only two countries **`In`** the set instead of five. Why do you think that is happening? If you guessed context filters, you are correct.

The reason this happens is that, if you look at the order of operations in Tableau, top N gets filtered first before dimension filtering is applied to the view. So whenever you use a top N filter, Tableau has already calculated the top N for the dimension in the view, and when you use a secondary dimension for filtering, it gets filtered on the top N data and not the whole dataset. Hence, you see just two countries instead of five because the top countries of China, US, India, Russia, and Japan had data from 2000 to 2011, but there was no data for India, Russia, and Japan for 2012. The missing data meant that, when you filtered on the top five countries, the filter was applied to those five countries; and, since three had no data, your view only showed two countries.

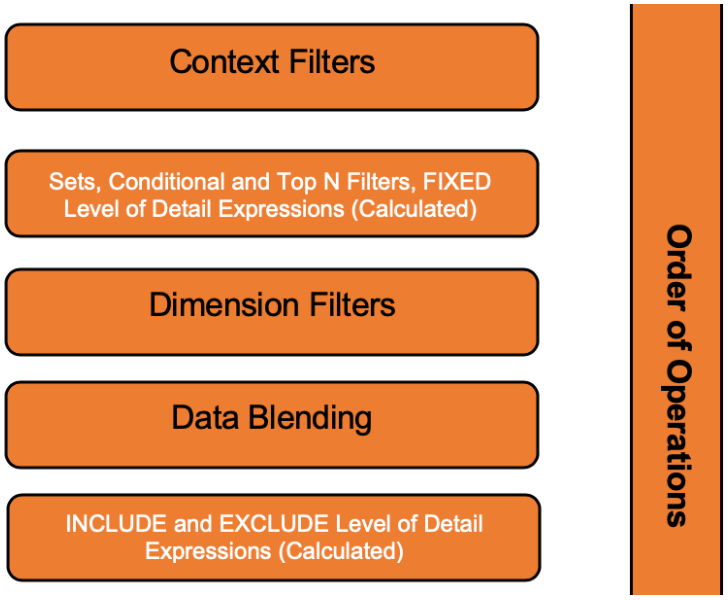


Figure 11.120: Sectional view of the tableau order of operations

9. To fix this problem, convert the **`YEAR(Year)`** dimension in the **`Filters`** shelf to **`Context Filter`** as shown here:

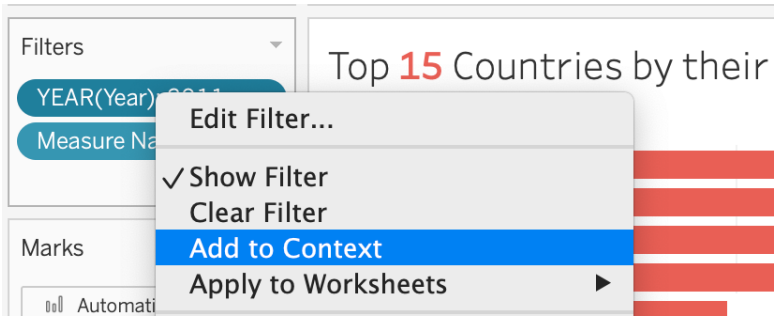


Figure 11.121: Adding Year to the context filter

10. Make **`YEAR(Year)`** a context filter, and the view will be updated. The current view now has the top five countries as expected:

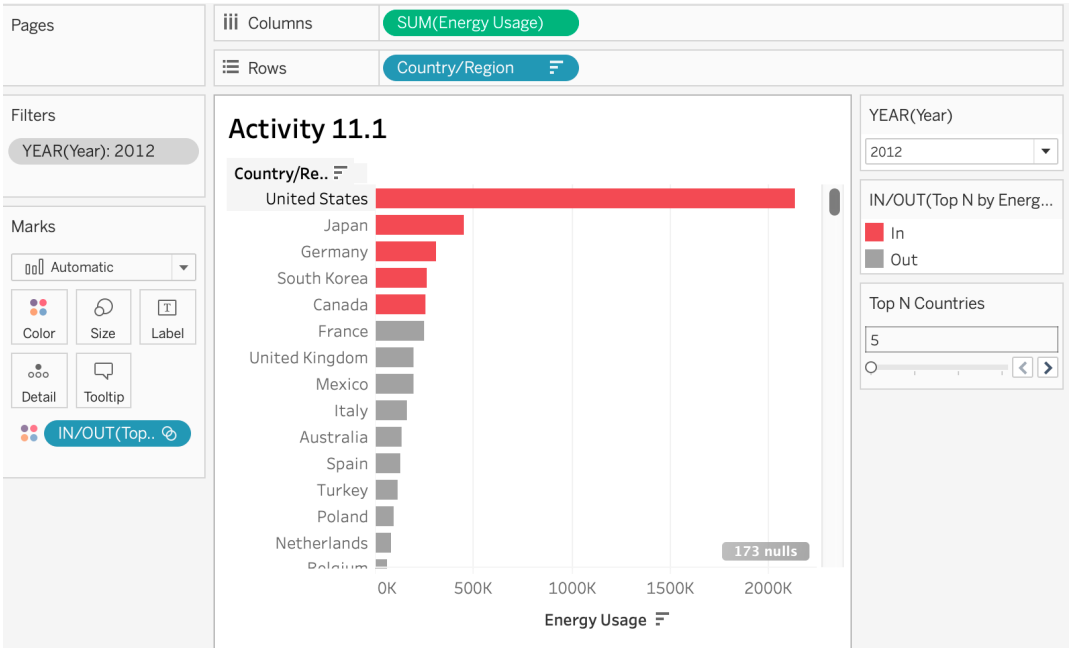


Figure 11.122: Final output after adding the context filter

11. Finally, edit your title and insert parameter values into your title as shown here:

> **NOTE**
>
> You don't have to type **<Parameters.Top N Countries>** manually; you can use the **Insert** button at the top and select the dynamic fields you want to add to your title.

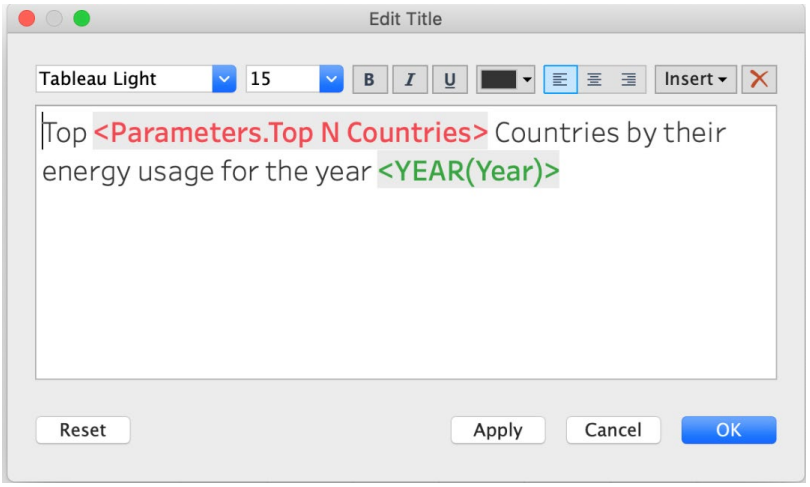


Figure 11.123: Editing the title using the dynamic parameter variable

The final output will be as follows:

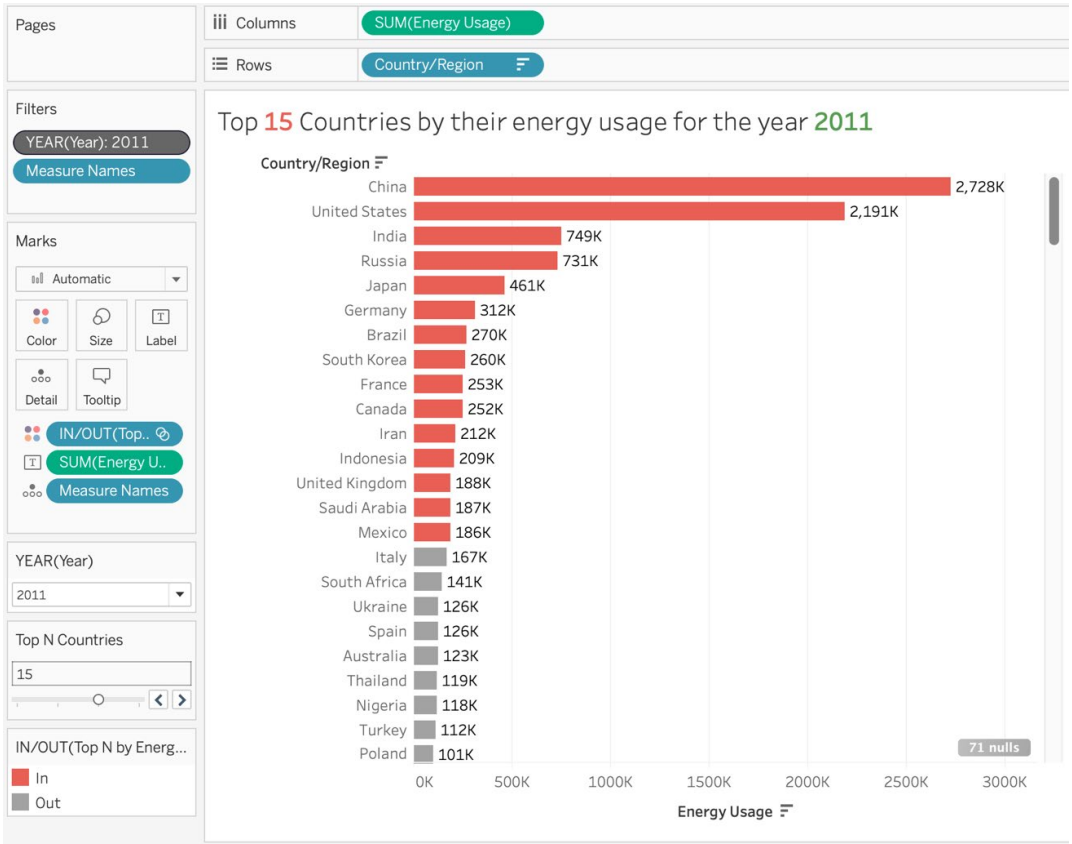


Figure 11.124: Top 15 countries by their energy usage

In this activity, you utilized all the major concepts and expanded your knowledge of parameters used as part of the top N set. You also utilized context filters with date filters,and saw why they are necessary. You then worked through dynamic sheet titles, which used parameter values as dynamic variables in your view.

> **NOTE**
>
> The following activity solutions relate to those three additional chapters referenced in the preface. Find these online at https://packt.link/CTCxk.

CHAPTER 12: TABLEAU INTERACTIVITY: PART 2

ACTIVITY 12.01: USING ACTIONS

Solution:

1. Load the **`Sample_Coffee_Chain`** dataset in the Tableau instance if it is not already open.
2. Create a new worksheet, rename the sheet as **`'Pie Chart'`**, Change the mark type to **`Pie`** first and later drag **`Market Size`** on **`Color`** and **`SUM(Sales)`** on **`Angle`**. Also, use them both as labels. Convert **`SUM(Sales)`** to **`Table Calculation`** and calculate it as **`Percent of Total`**:

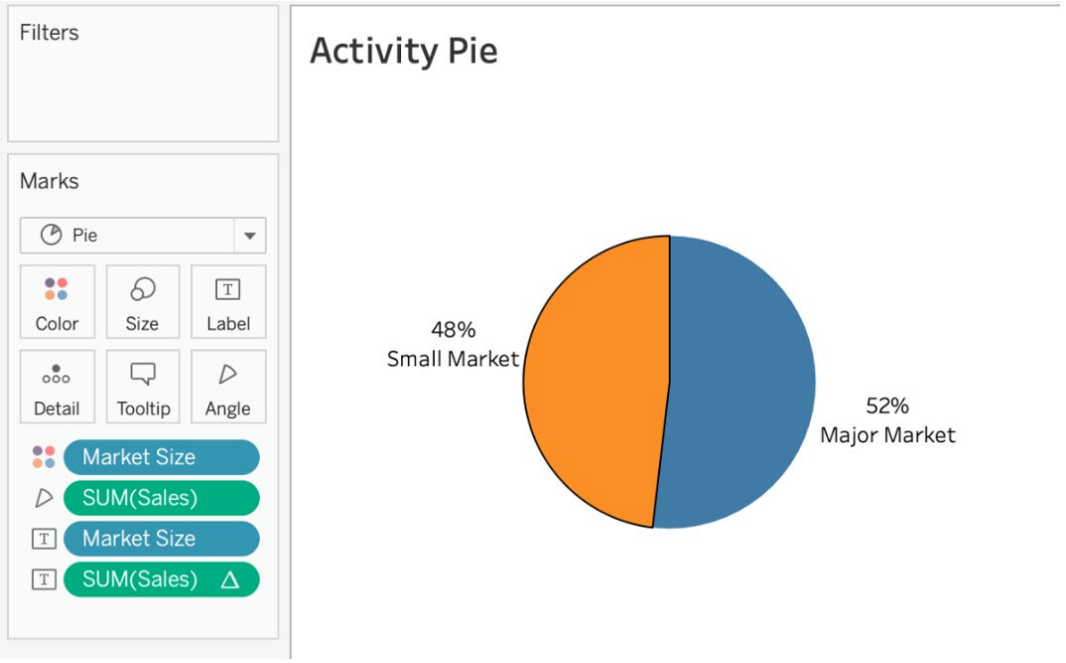


Figure 12.59: Small versus major market pie chart

3. Create another worksheet, rename the sheet as **`'Bar Chart'`**, Create a bar chart by dragging **`Product Type`** to the **`Rows`** shelf and **`SUM(Sales)`** to the **`Columns`** shelf, using **`SUM(Sales)`** as the label, as shown:

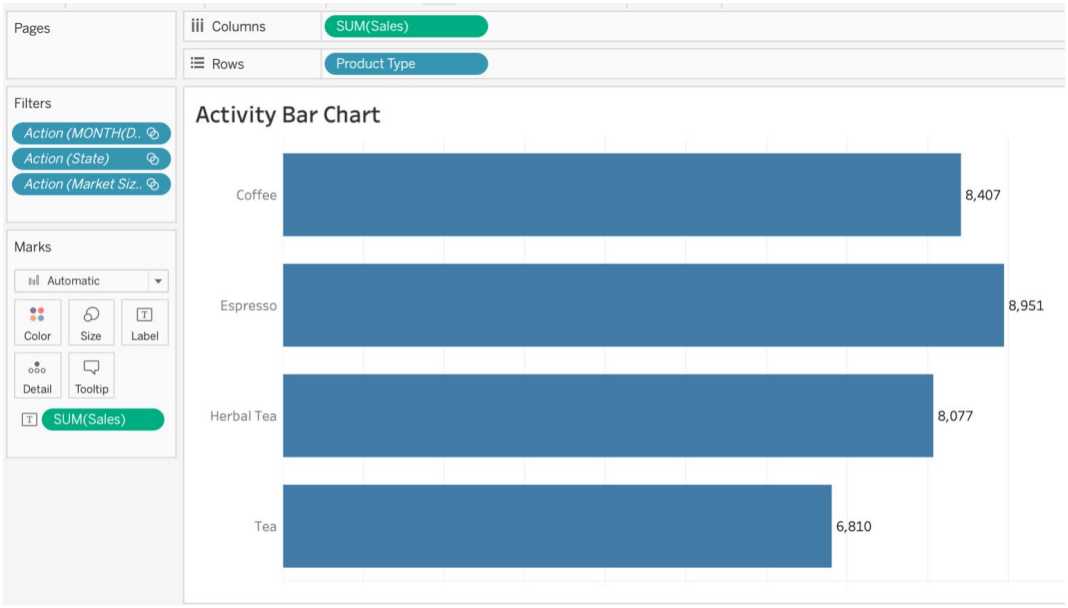


Figure 12.60: Beverage type bar chart

4. Create the third worksheet, rename the sheet as **`'Line Chart'`**, Create a line chart by dragging **`Sum(Sales)`** to the **`Rows`** shelf and **`Continuous [MONTH(Date)]`** to the **`Columns`** shelf. When you load the data, you might have loaded the date column as a measure; make sure to convert it to a date. On the label, show **`SUM(SALES)`**. However, only show the labels on min/max values, as shown:

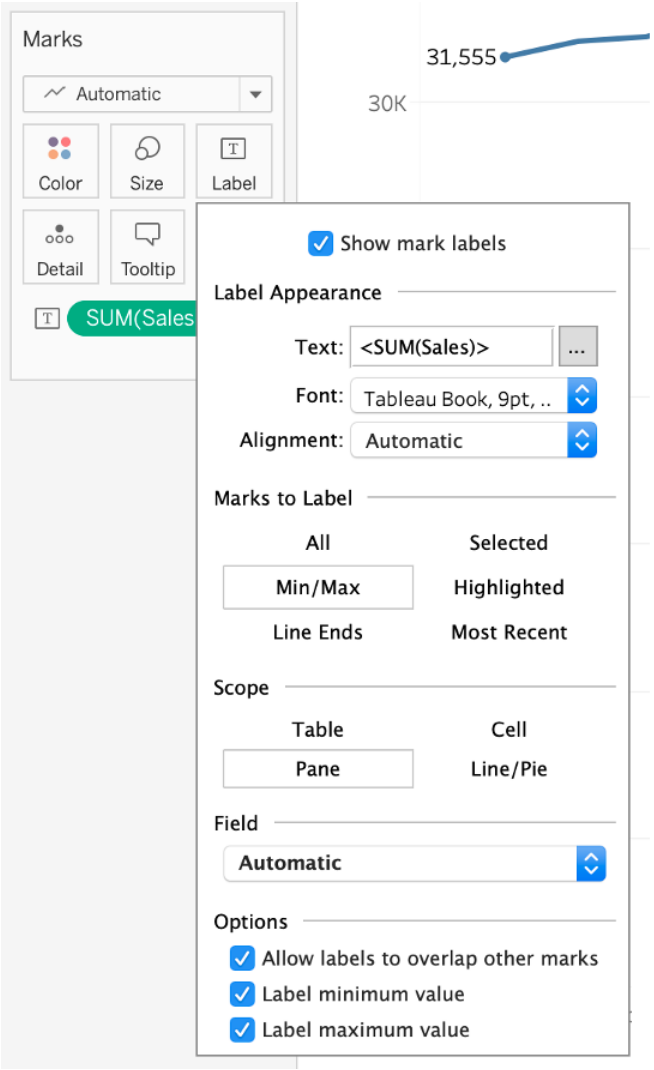


Figure 12.61: Label marks card options

In the preceding screenshot, you have selected **Min/Max** for **Marks to Label** because your goal is to only show the labels at the min/max values and not across all data points. **Scope** is defined as **Pane** as you want your marks label to change when the data in the pane/view changes. Select the **Allow labels to overlap other marks**, **Label minimum value**, and **Label maximum value** options:

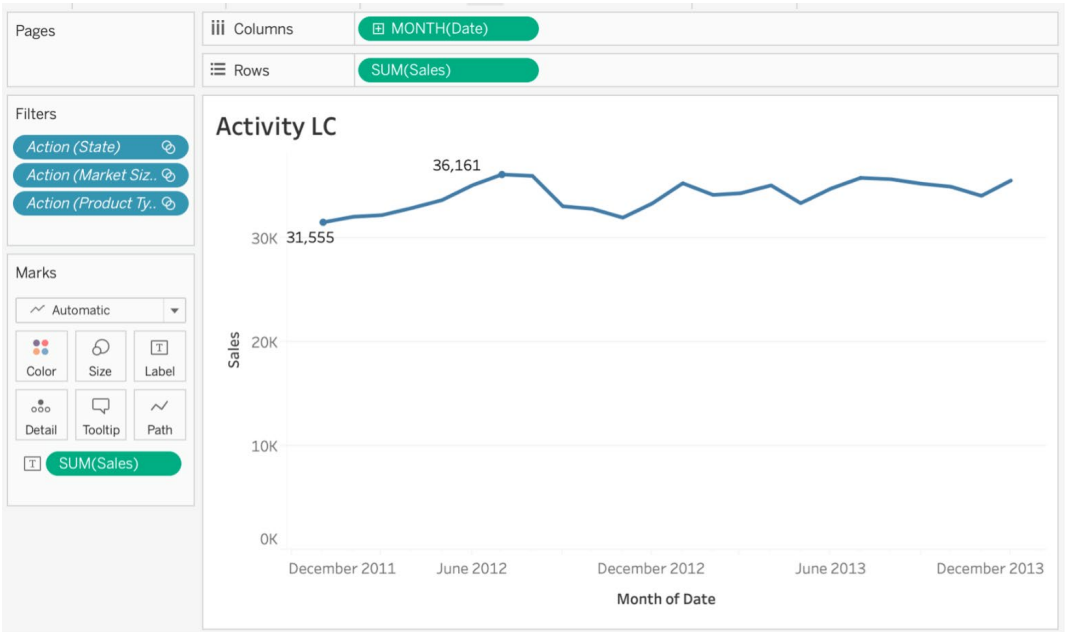


Figure 12.62: Sales by Month line chart

5. Change the mark type to **Map**, drag **State** onto the **Detail** marks card, and drag **SUM(Profit)** onto the **Color** marks card. Also, drag **SUM(Profit)** onto the **Label** marks card and **SUM(Sales)** onto the **Tooltip** marks card:

> **NOTE**
>
> Your default view for maps may not be United States if you are not from the United States. To change the default country to United States, go to **Map** under **Menus** and click on **Edit Locations** and change your country to **United States**.

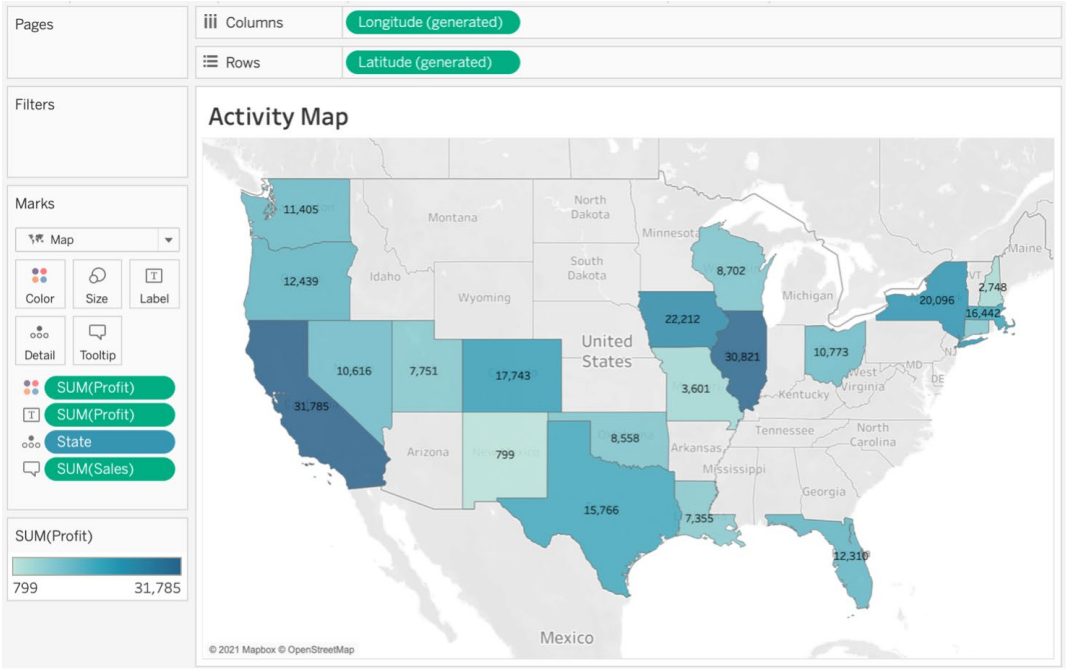


Figure 12.63: Activity map with profit colored

6. Create a dashboard view by clicking on the **`Create dashboard`** button at the bottom right of the screen. Add a couple of text boxes to your view: a title as well as a contextual info box. Once that is done, drag all four worksheets you created in previous steps onto your dashboard view, and you should end up with the following dashboard without any filter actions:

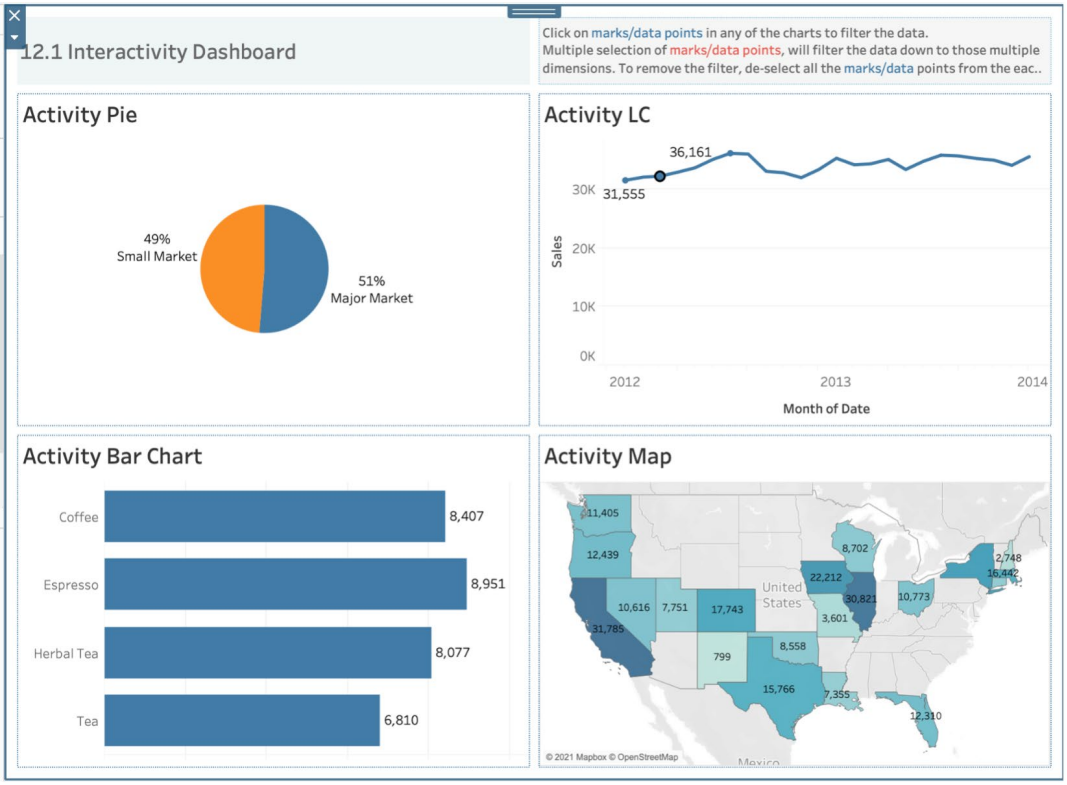


Figure 12.64: Adding a worksheet to the dashboard

7. Navigate to the dashboard menu and add actions by clicking on **`Actions`** | **`Add Action`** | **`Filter...`**:

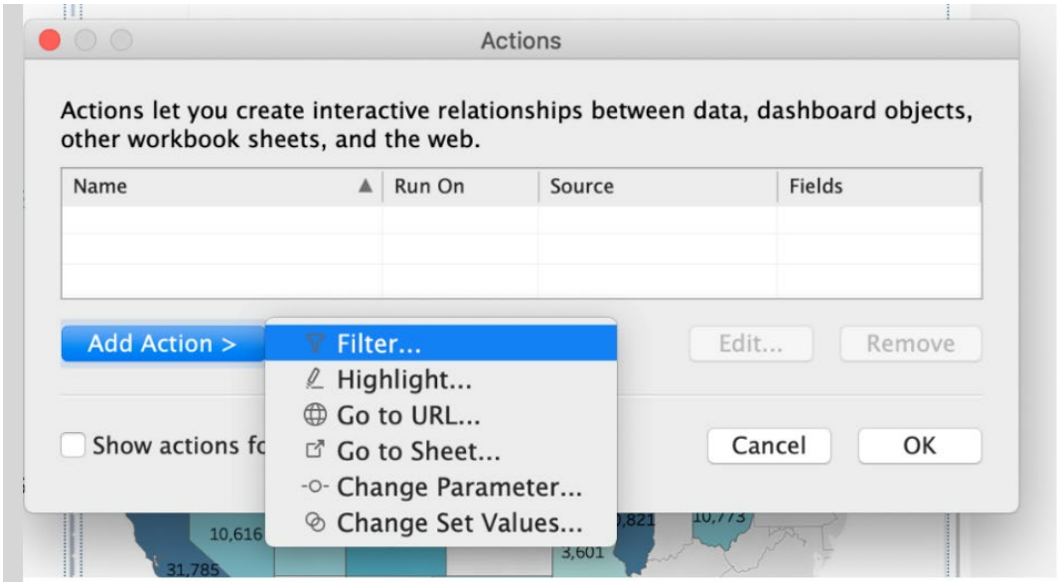


Figure 12.65: Adding filter actions

8. You want all your sheets to act as a filter across all other worksheets. You could individually add each worksheet as a separate filter action and source sheets, but that will increase the load as well as steps. Instead, choose the more efficient method by creating one filter action where all worksheets will act as filter worksheets (the target sheets will be all four of these worksheets). Additionally, select the following options for this filter action:

- **Name**: **`All in One Filter Action`**.
- **Source Sheets**: Use all worksheets as our source sheet as we want all sheets to act as a filter.
- **Run action on**: Use **`Select`** as the method on which the action runs as using **`Hover`** might add unnecessary complexity to the dashboard.
- **Target Sheets**: Again, as mentioned in **`Source Sheets`**, you want every interaction on any of the worksheets to filter on all other sheets, so select all the sheets for **`Target Sheets`**.
- **Clearing the selection will**: Select **`Show all values`** as you want users to be able to revert to the original dashboard stage when they clear their selection.

- Target Filters: Since you are not targeting a specific dimension/field in this dashboard, keep All Fields selected:

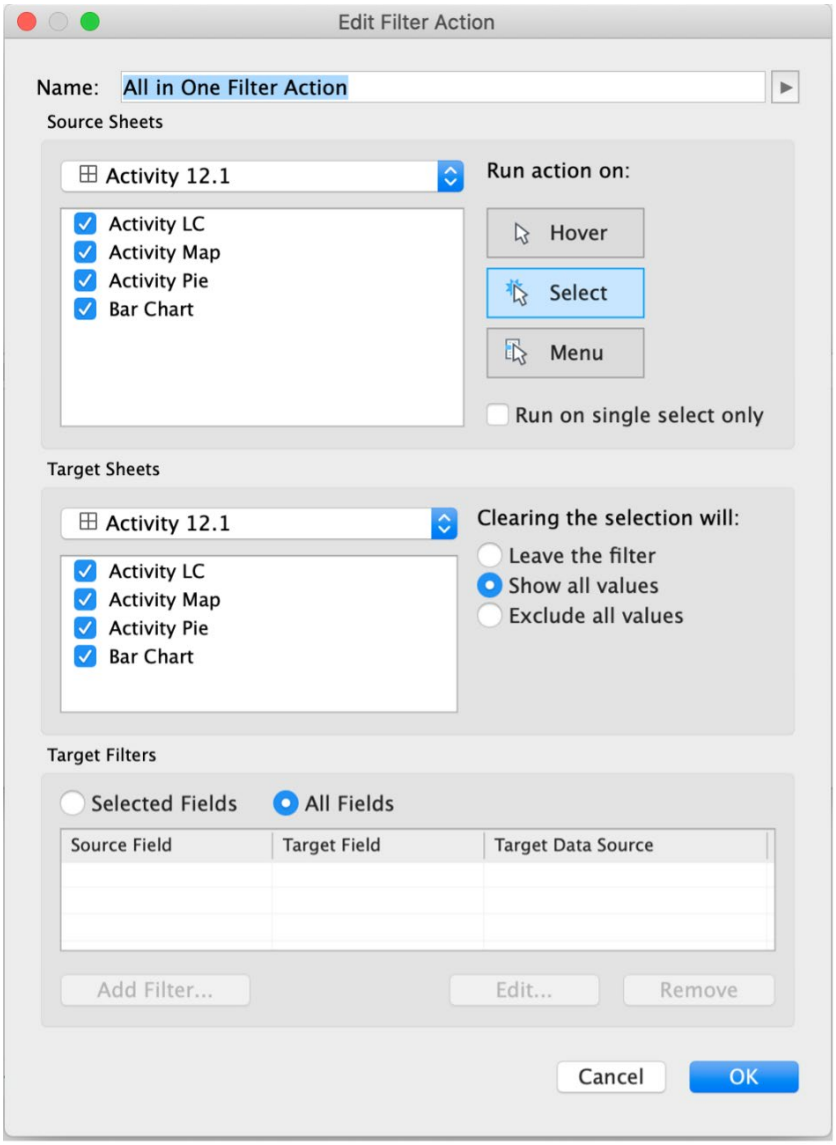


Figure 12.66: Filter action options with all worksheets selected

9. Test the filter action by clicking on any of the marks/data points as desired, as illustrated in the following screenshots. First, select **`Dates`** from the line chart. The output will be as follows:

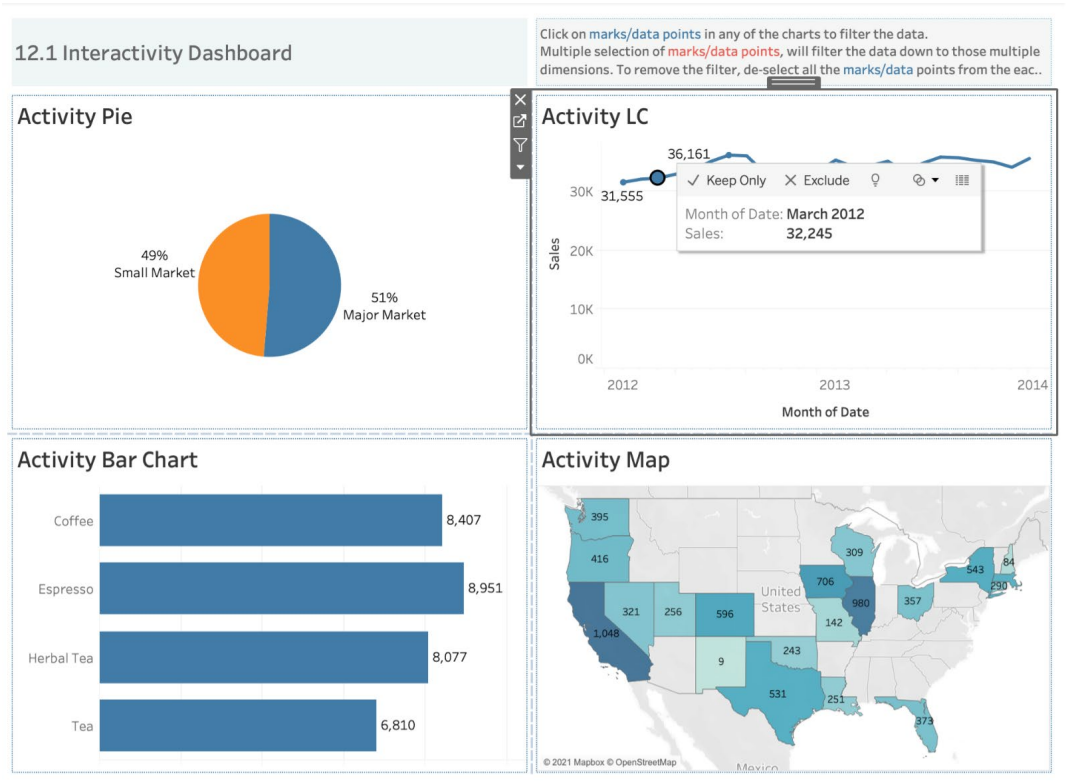


Figure 12.67: Testing the filter action

10. Next, select **`Market Size`** and **`Product Type`**:

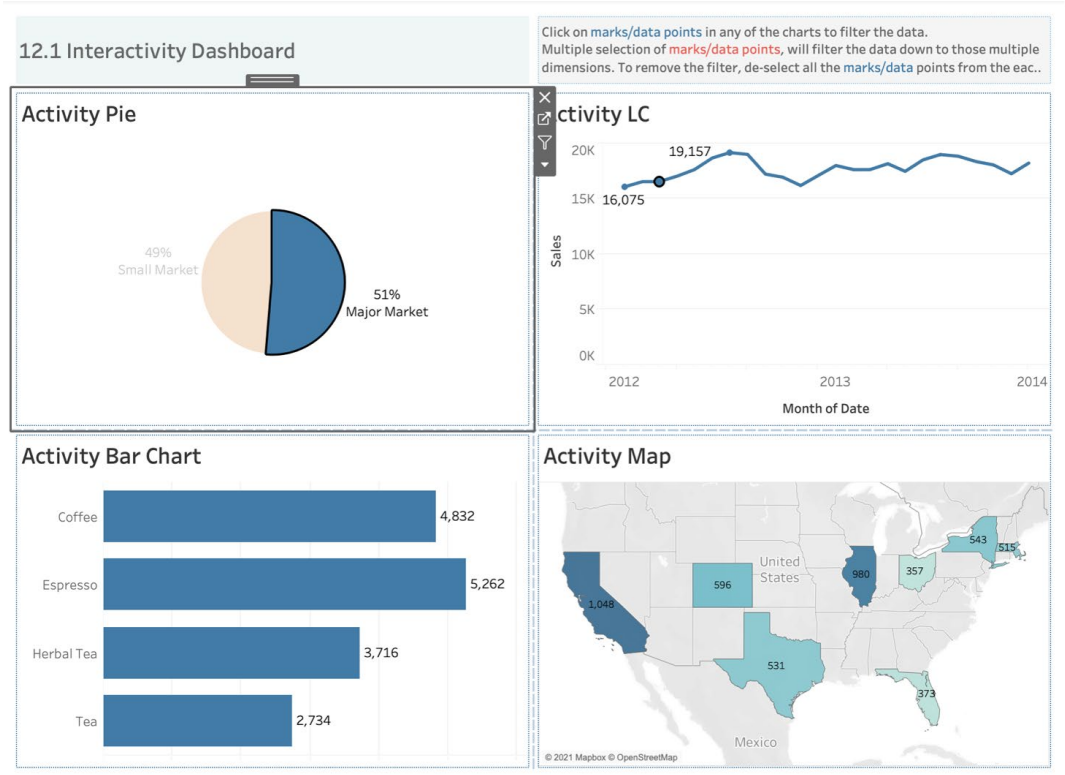


Figure 12.68: Demonstrating interactivity on the dashboard

In this activity, you created multiple worksheets and filter actions wherein every source sheet acted as a target sheet, as well as a master filter action through which any marks/data point a user selects will filter the dashboard based on the dimension/field.

CHAPTER 13: DISTRIBUTION OF DASHBOARDS

ACTIVITY 13.01: SAVING AND SHARING YOUR WORK

Solution:

1. Open a new Tableau workbook using the **File** | **New** menu or using the *Ctrl* + *N* or *Command* + *N* shortcut.
2. Click **Connect to Data** in the **Data** pane or use the **Data** | **New Data Source** menu (*Ctrl* + *D*/*Command* + *D*).
3. Click **World Indicators**, under **Saved Data Sources**. If it is not available there, go to https://packt.link/MjCGb and open this file using the **Connect** | **To a File** | **Text file** option.

Figure 13.33: Saved Data Sources

4. Double-click the **`Country/Region`** field in the **`Dimensions`** pane to create a quick map view.

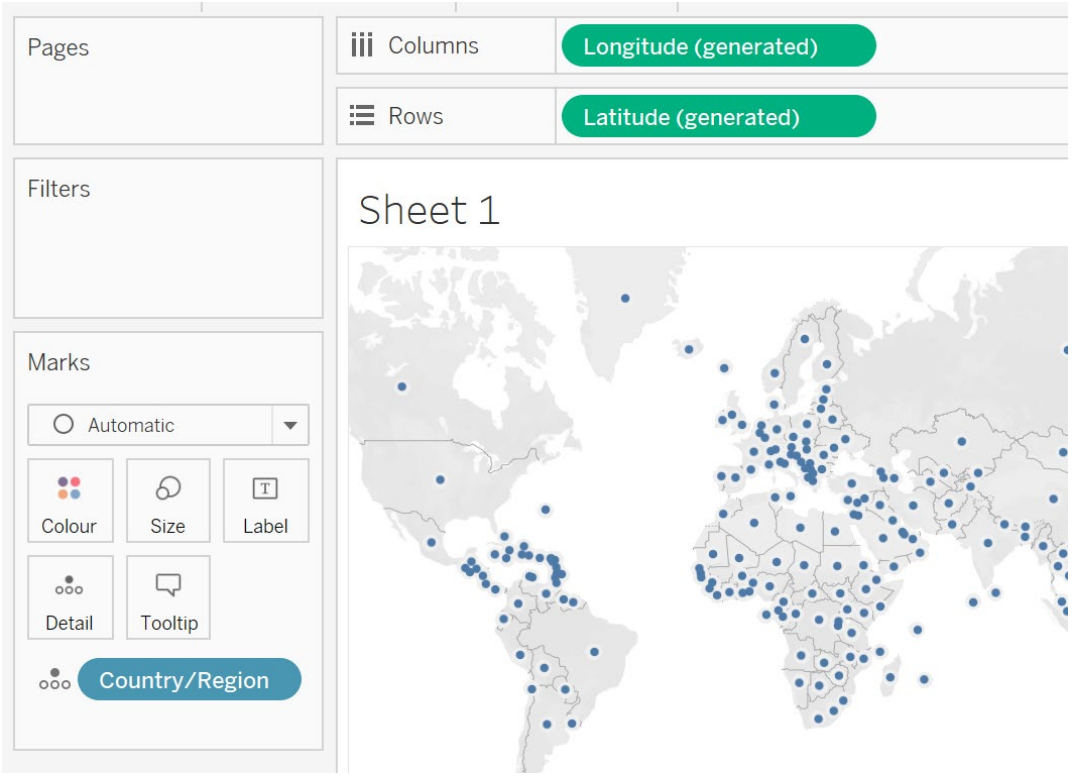


Figure 13.34: Adding Country/Region to the canvas

5. Drag and drop the **`Population Total`** measure to the **`Colour`** box in the **`Marks`** pane. The view will change, as shown in the following screenshot (including changing the calculation to **`SUM`**):

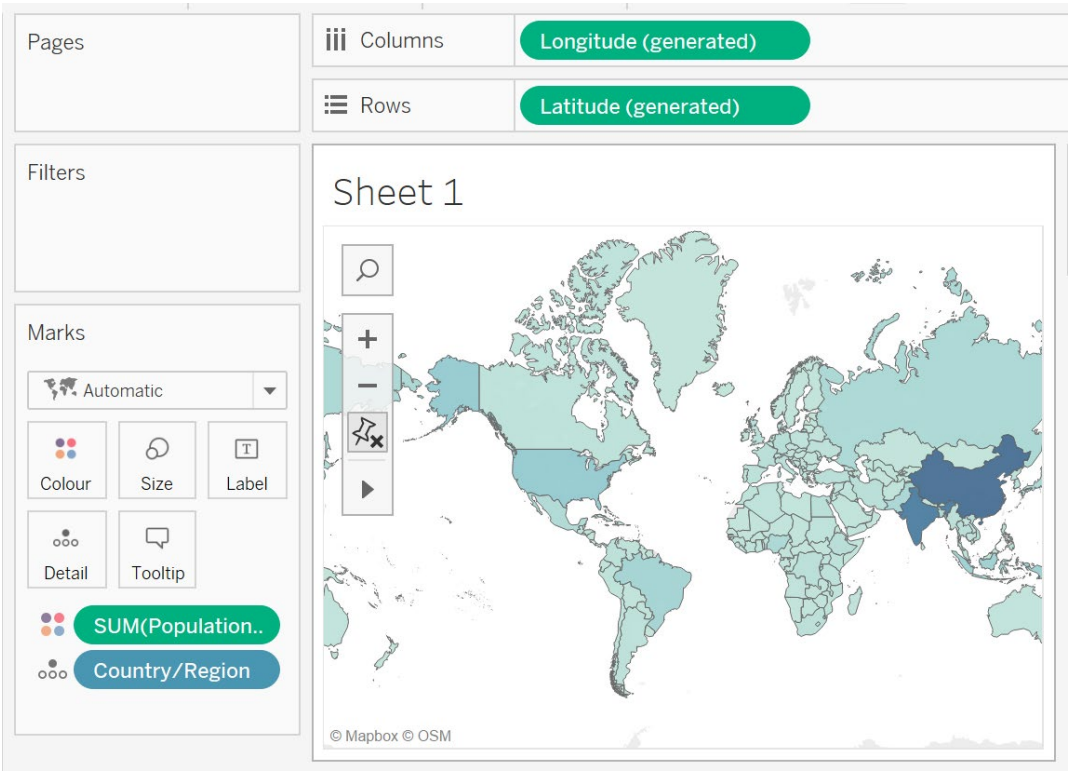


Figure 13.35: Population by Country view

6. Right-click the **`Year`** measure and use the **`Show Filter`** option.

7. In the **`YEAR(Year)`** filter on the right, untick **`(All)`** and then select the **`2009`** checkbox to filter for 2009 data only.

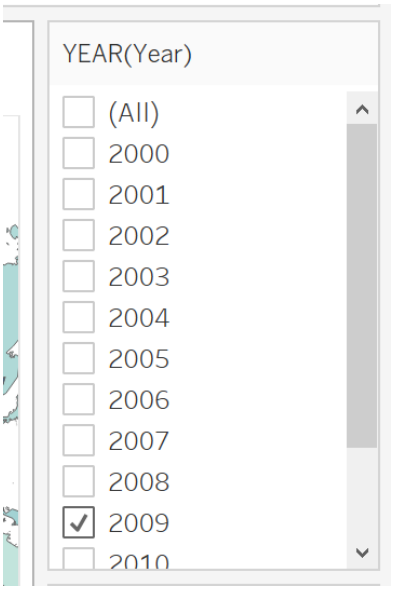


Figure 13.36: Filters applied

8. Right-click the **World Indicators** data source in the **Data** pane and choose **Extract Data...**. Alternatively, use the **Data** | **World Indicators** | **Extract Data...** menu.
9. Leave the options as they are and click **Extract**.
10. If prompted, save the extract in the proposed folder, which should be your **Datasources** repository.
11. Rename your worksheet **World Population (2009)** by right-clicking the **Sheet 1** tab and using the **Rename** option.
12. Use the **File** | **Save** or **File** | **Save As...** menu, or *Ctrl* + *S*/*Command* + *S*.
13. Navigate to your desktop, change the **Save As** type to **Tableau Packaged Workbook (*.twbx)**, and use **World Population in 2009.twbx** as the filename.
14. Click **Save**. You now have a packaged workbook containing this data.
15. If you already have a Tableau Public account, skip to *step 17*. Otherwise, navigate to https://public.tableau.com/ in your browser and click the **SIGN UP** button at the top right of your screen.

16. Enter your name, email, and password in the relevant fields to create your account.

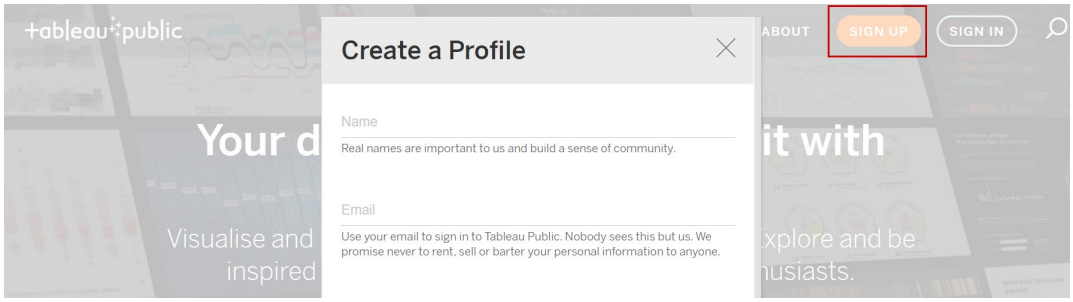


Figure 13.37: Creating a Tableau Public profile

17. Once you have created an account, use the **`Server`** | **`Tableau Public`** | **`Save to Tableau Public...`** menu.

18. If necessary, log in to Tableau Public using your credentials.

19. Save the workbook under the name **`World Population in 2009`**.

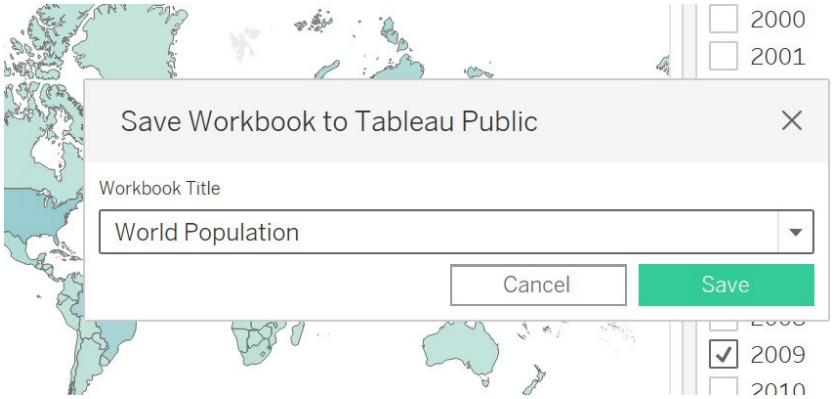


Figure 13.38: Save Workbook to Tableau Public

20. Click **`Save`**. A new browser tab will open with your dashboard published on Tableau Public.

CHAPTER 14: CASE STUDY

ACTIVITY 14.01: MERGING WORKSHEETS TO CREATE A DASHBOARD

Solution:

1. Add a new dashboard page and title it **`Airline/Airport Performance Scoreboard`**.
2. Set the size to 1200 x 1000px (or any canvas size you feel will fit the size of your screen).
3. Add the title text, labeling it **`<Parameters.Origin Airport.P> Performance Scoreboard`** and referencing the **`Origin Airport`** parameter to complete the title:

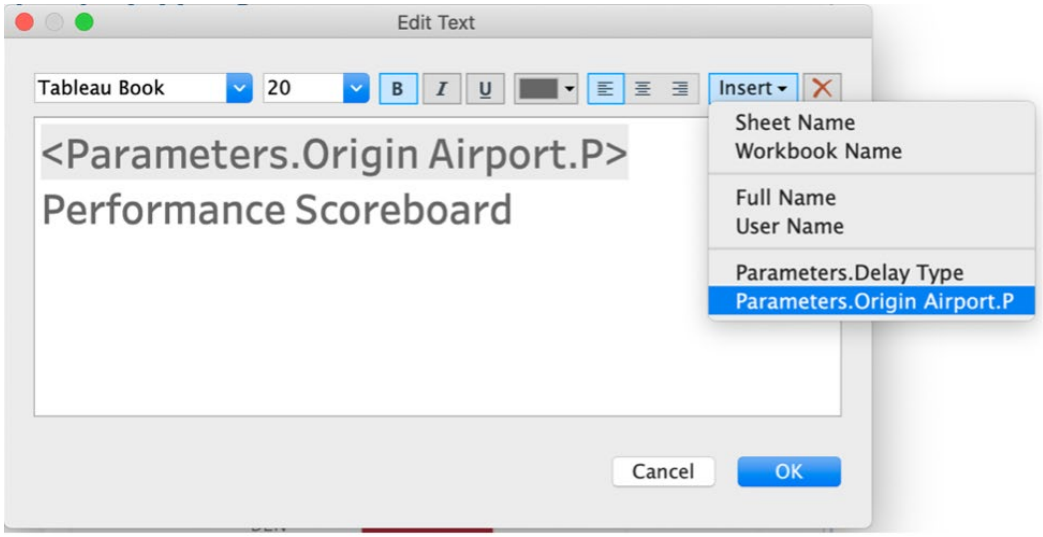


Figure 14.84: Add/edit title for dashboard

4. Add a vertical container that occupies the remainder of the space below the title.

5. Add a second vertical container to the right of the first one. Size it to accommodate filters and parameters. Next, add a horizontal container inside the large vertical container. Your canvas should look like the following screenshot:

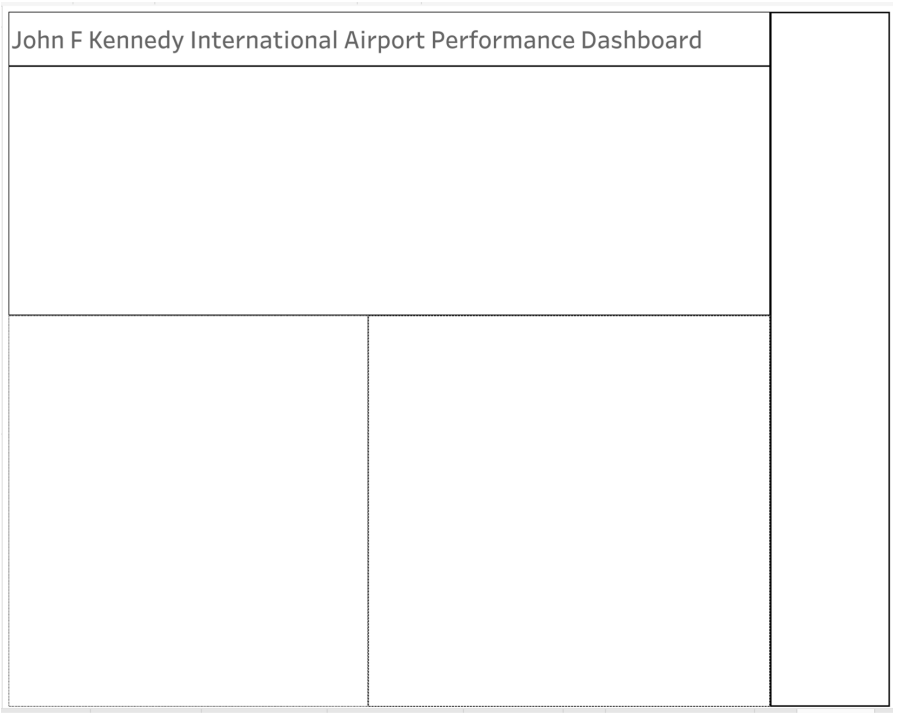


Figure 14.85: Containerizing the dashboard

6. Add the **`Delays by Origin Airport`** worksheet to the large vertical container.
7. Add the **`Performance vs. All Airports`** worksheet beneath the priorworksheet.
8. Add the **`Carrier Delay by Airport Trend`** worksheet beneath the two scoreboard items. Things might need some re-arrangements as your filters/parameters can be shifted to the top-right vertical container for easier use, and you can hide the title for each of the worksheets as required.

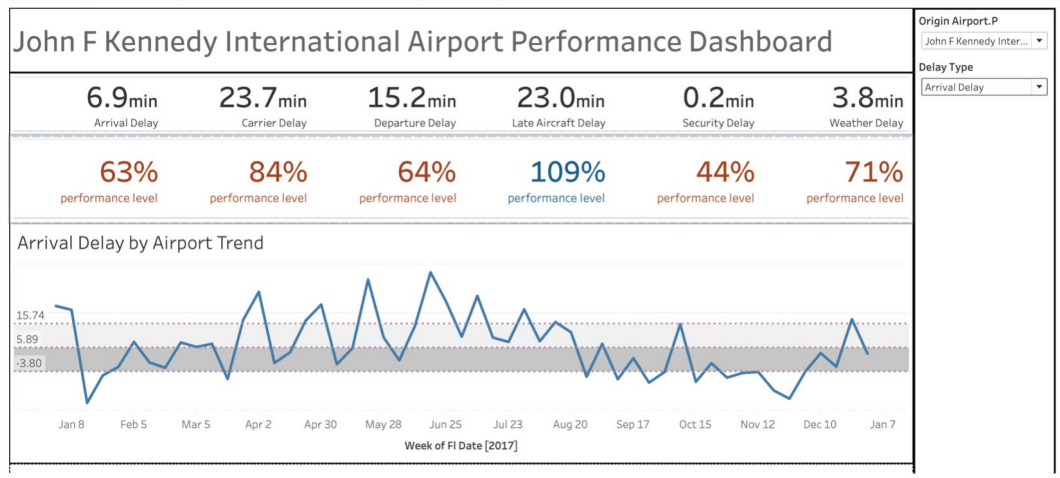


Figure 14.86: Adding both the scoreboard and By Airport Trend in the dashboard

9. Add the **`by Airport Carrier`** and **`Flight Delays by Destination Airport`** charts. At this point, you might decide to delete the color and size legends as you only want filters and parameters as the options.

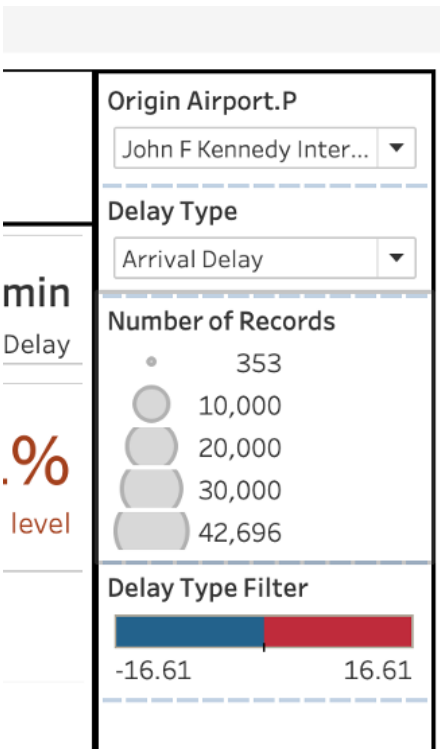


Figure 14.87: Filter and color legends

10. Make sure your filters, parameters, and the **`Airport Map`** sheet are all contained in the narrow vertical container to the right.

11. You previously have tested each worksheet separately. Now, change the airport as well as the delay type to see if the dashboard gets updated with the appropriate numbers (visual cues):

    Test 1: La Guardia Airport and Carrier Delay

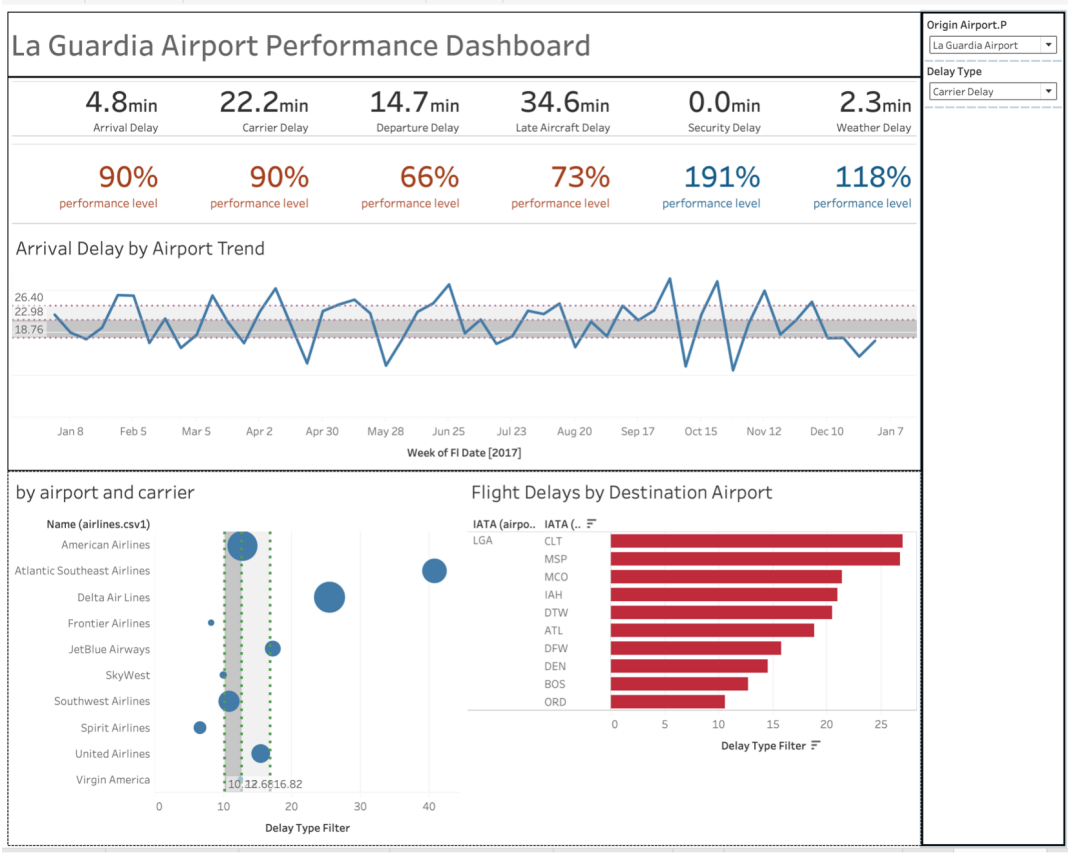


Figure 14.88: Test 1: La Guardia Airport and Carrier Delay view

The view above is for La Guardia Airport, where you can see the average arrival delay for LGA is 4.8 min. The second row, the percentage KPI, compares LGA with the rest of the dataset. The charts at the bottom break down the **`LGA Carrier Delay by Airlines`**, while the flight delays by destination airport show the arrival delay for flights into other airports.

Test 2: San Francisco and Security Delay Options

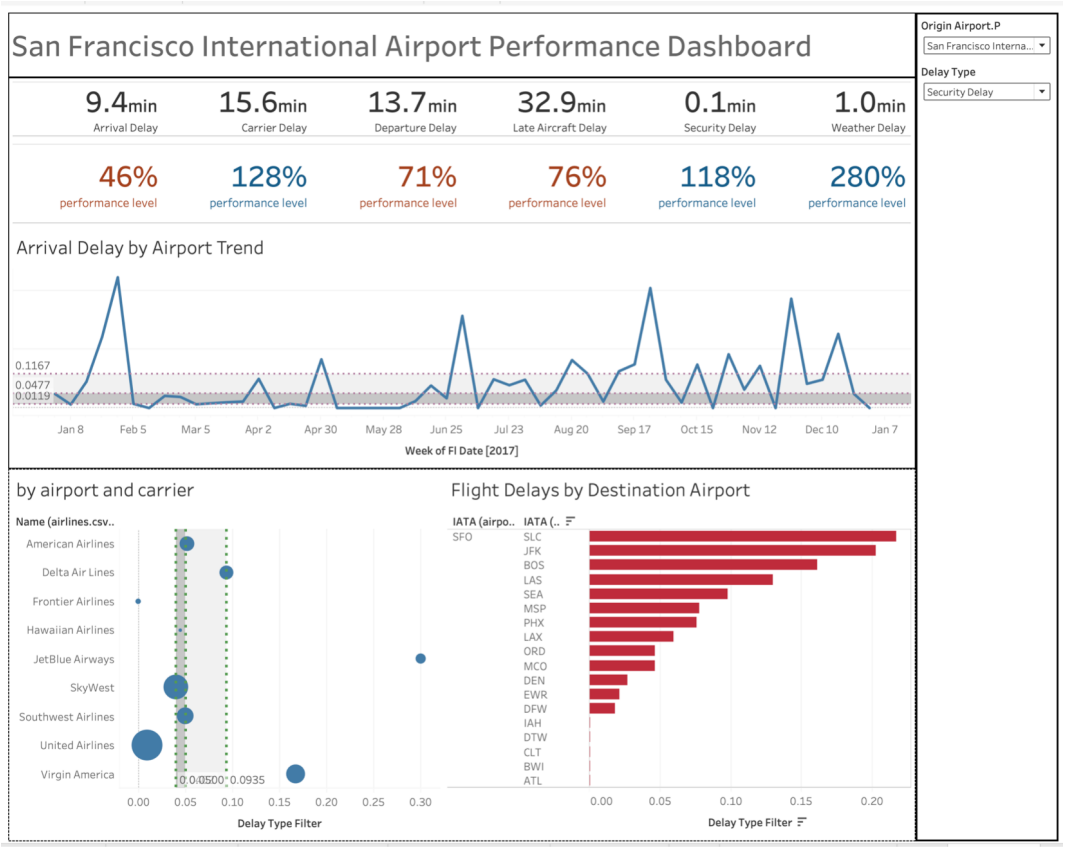


Figure 14.89: Test 2: San Francisco and Security Delay Options view

The preceding view is for San Francisco, where we can see the average arrival delay for SFO is 9.4 min. The second row, the percentage KPI, compares SFO with the rest of the dataset. The **`by airport and carrier`** chart at the bottom breaks down the **`SFO Carrier Delay by Airlines`**, while **`Flight Delays by Destination Airport`** shows the arrival delays for flights into other airports.