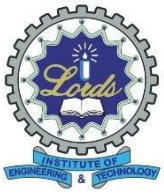


LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

Approved by AICTE | Affiliated to Osmania University | Estd.2003.

Survey No.32, Himayath Sagar, Near TSPA Junction, Hyderabad-500091

Department of Computer Science and Engineering

LAB MANUAL

Data Visualization Lab

Course Code: U21CR3L1

B.E. (Bachelor of Engineering)

Branch: - Computer Science and Engineering

Semester – III A.Y: 2024-2025

Prepared by:

Dr Ruhiat Sultana

Associate Professor

Department of Computer Science and Engineering

Department of Computer Science and Engineering

VISION

To emerge as a center of excellence for quality Computer Science and Engineering education with innovation, leadership and values.

MISSION

- DM1: Provide fundamental and practical training through learner – centric Teaching-Learning Process and state-of-the-art infrastructure
- DM2: Develop design, research, and entrepreneurial skills for successful career
- DM3: Promote training and activities through Industry-Academia interactions

Course Name: Data Visualization Lab

Course Objectives:

The objectives of this course are :

1. Effective use of Business Intelligence (BI) technology (Tableau/Power BI/Google Data Cloud Studio) to apply data visualization.
2. To discern patterns and relationships in the data.
3. To build Dashboard applications.
4. To communicate the results clearly and concisely.
5. To be able to work with different formats of data sets

Course Outcomes:

On Successful completion of this course, students will be able to:

1. Learn how to import data into Tableau.
2. Enumerate Tableau concepts of Dimensions and Measures.
3. Develop Programs and understand how to map Visual Layouts and Graphical Properties.
4. Create a Dashboard that links multiple visualizations.
5. Use graphical user interfaces to create Frames for providing solutions to real-world problems

II B.E CSE-III Semester

Data Visualization Lab

CO-PO Mapping

Student will be able to

CO. No.	Description	Bloom's Taxonomy Level
C238.1	Learn how to visualize data in python	BTL3
C238.2	Enumerate Tableau concepts of Dimensions and Measures.	BTL3
C238.3	Develop Programs and understand how to map Visual Layouts and Graphical Properties.	BTL3
C238.4	Create a Dashboard that links multiple visualizations.	BTL4
C238.5	Use graphical user interfaces to create Frames for providing solutions to real-world problems	BTL4

Course Outcomes (CO)	Program Outcomes (PO)												Program Specific Outcomes (PSO's)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
C238.1	2	3	-	-	2	-	-	-	2	-	-	-	2	-
C238.2	3	-	-	2	3	-	-	-	-	2	-	2	2	-
C238.3	2	-	2	-	3	-	-	2	-	-	-	-	2	3
C238.4	-	-	3	-	3	3	-	2	-	-	-	-	2	3
C238.5	2	3	-	2	3	-	-	-	2	-	2	-	2	2
Avg. C23	**2.2**	**3.0**	**2.5**	**2.0**	**2.8**	**3.0**	**-**	**2.0**	**2.0**	2.0	2.0	**2.0**	**2.0**	**2.6**

List of Experiments:

1. Understanding Data, What is data, where to find data, and Creating Your First Visualization in Python.

2. Getting started with Tableau Software using Data file formats, connecting your Data to Tableau, creating basic charts(line, bar charts, Tree maps), Using the Show me panel.

3. Tableau Calculations, Overview of SUM, AVR, and Aggregate features, Creating custom calculations and fields.

4. Applying new data calculations to your visualizations, Formatting Visualizations, Formatting Tools and Menus, and Formatting specific parts of the view.

5. Editing and Formatting Axes, Manipulating Data in Tableau data, Pivoting Tableau data.

6. Structuring your data, Sorting and filtering Tableau data, and Pivoting Tableau data.

7. Creating Dashboards & Storytelling, creating your first dashboard and Story, and Design for different displays, and publish your visualization.

8. Creating custom charts, cyclical data, circular area charts, and Dual Axis charts.

9. Advanced Visualization Tool Power BI: Using Filters, Using the Detail panel, using the Size panels, customizing filters, And customizing tooltips, and Formatting your data with colors.

10. Visualization data supported chart types in Power BI, Map Visualizations, Color palettes in Charts, Loading Shapes, text boxes, and images.

11. Explore the Google Data Studio Dashboard and connect to your first source.

12. Create a Report on Google Data Studio and share the report.

Introduction to Data Visualization tools

Data visualization is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from. It is the representation of information and data through use of common graphics, such as charts, plots, infographics, and animations. Data visualization is a powerful way for people, especially data professionals, to display data so that it can be interpreted easily.

Data Visualization enables decision-makers of any enterprise or industry to look into analytical reports and understand concepts that might otherwise be difficult to grasp.

Benefits of Data Visualization:

1. It is easy to understand the information with graphics

2. It made data to be represented in attractive way

3. Shows complex relationships

4. Helps to process large datasets

5. Useful for identifying trends

6. Minimizes ambiguity

Data visualization tools provide the ability to see and understand data trends, outliers, and patterns in an easy, intuitive way. There are various data visualization tools available. One must choose the tool based on various factors such as its ease of use, types of graphical representations the tool can produce, size of the dataset the tool can handle etc. some of Data Visualization tools are Tableau, Power BI, Google Charts, Python, R, etc.

The following are some common types of data visualizations:

- ✓ Table: A table is data displayed in rows and columns, which can be easily created in a Word document or Excel spreadsheet.
- ✓ Chart or graph: Information is presented in tabular form with data displayed along an x and y axis, usually with bars, points, or lines, to represent data in comparison.
- ✓ Geospatial visualization: Data is depicted in map form with shapes and colours that illustrate the relationship between specific locations, such as a choropleth or heat map.
- ✓ Dashboard: Data and visualizations are displayed, usually for business purposes, to help analysts understand and present data

Experiment-1:

1. Understanding Data, What is data, where to find data, and Creating Your First Visualization in Python.

What is Data?

Data refers to raw facts, statistics, or information collected or stored in a structured or unstructured form. Data can take various forms, such as text, numbers, images, videos, and more. It is the foundation of all information and knowledge and is used in various fields for analysis, decision-making, and understanding trends and patterns.

Data can be categorized into two main types:

Structured Data: This type of data is organized into a specific format, such as tables or databases, and is easily searchable and analyzable. Examples include spreadsheets, relational databases, and CSV files.

Unstructured Data: Unstructured data lacks a specific format and can include text documents, social media posts, images, audio recordings, and more. Analyzing unstructured data often requires advanced techniques like natural language processing and image recognition.

Where to Find Data?

You can find data from various sources, depending on your specific needs:

Open Data Portals: Many governments and organizations provide free access to a wide range of data through open data portals. Examples include Data.gov (United States) and data.gov.uk (United Kingdom).

Data Repositories: Academic institutions, research organizations, and data enthusiasts often share datasets on platforms like Kaggle, GitHub, and the UCI Machine Learning Repository.

APIs (Application Programming Interfaces): Some websites and services offer APIs that allow you to programmatically access and retrieve data. Examples include Twitter API, Google Maps API, and financial market APIs.

Web Scraping: You can extract data from websites using web scraping tools and libraries like Beautiful Soup and Scrapy. However, be mindful of the website's terms of use and legal restrictions.

Surveys and Surveys: You can conduct your own surveys or collect data through questionnaires and interviews.

IoT Devices: Internet of Things (IoT) devices generate vast amounts of data that can be used for various purposes.

Commercial Data Providers: Some companies specialize in selling datasets for specific industries, such as market research, finance, and healthcare.

Foundations for Building Data Visualizations:

Creating effective data visualizations requires a strong foundation in several key areas:

Data Analysis: Before creating visualizations, you should thoroughly analyze your data to understand its structure, relationships, and any patterns or trends. Exploratory data analysis (EDA) techniques can help with this.

Statistical Knowledge: Understanding basic statistics is essential for making meaningful interpretations of data. Concepts like mean, median, standard deviation, and correlation are commonly used in data visualization.

Domain Knowledge: Having knowledge of the specific domain or subject matter related to your data is crucial for creating contextually relevant visualizations. It helps you ask the right questions and provide valuable insights.

Visualization Tools: Familiarize yourself with data visualization tools and libraries such as matplotlib, Seaborn, ggplot2, D3.js, and Tableau. Each tool has its strengths and can be used for different types of visualizations.

Design Principles: Study design principles, including color theory, typography, and visual hierarchy, to create visually appealing and effective visualizations. Avoid common pitfalls like misleading visualizations.

Interactivity: Learn how to add interactive elements to your visualizations to engage users and allow them to explore the data. This can be achieved using tools like JavaScript, Python libraries, or dedicated visualization software.

Creating Your First Visualization:

To create your first data visualization, follow these general steps:

Select Your Data: Choose a dataset that aligns with your goals and interests. Ensure that the data is clean and well-structured.

Define Your Objective: Clearly define what you want to communicate or explore with your visualization. Are you looking to show trends, comparisons, or distributions?

Choose the Right Visualization Type: Select a visualization type that suits your data and objectives. Common types include bar charts, line charts, scatter plots, histograms, and pie charts.

Prepare and Transform Data: Preprocess your data as needed. This may involve aggregating, filtering, or transforming the data to fit the chosen visualization.

Create the Visualization: Use a suitable tool or library to create your visualization. Customize it with labels, colors, and other design elements.

Interactivity (Optional): If appropriate, add interactive features to your visualization to allow users to interact with the data.

Test and Iterate: Review your visualization for accuracy and clarity. Seek feedback from others and make improvements as necessary.

Publish or Share: Once you are satisfied with your visualization, publish it on a platform, embed it in a report, or share it with your intended audience.

Document and Explain: Provide context and explanations for your visualization. Clearly communicate what the viewer should take away from it.

Maintain and Update: If the data changes or new insights emerge, update your visualization accordingly.

Python has different modules for visualizing data such as matplotlib, seaborn.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It presents data in 2D graphics. Seaborn is a visualization library that is built on top of Matplotlib.

It provides data visualizations that are typically more aesthetic and statistically sophisticated.

Matplotlib can be installed using the following command:

pip install matplotlib

Once the module installed, it must be imported into the program using the following command import matplotlib as mpl, where mpl is the alias name given to matplotlib library.

matplotlib.pyplot is a state-based interface to matplotlib.

matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels etc. pyplot can be imported into the program using following command

import matplotlib.pyplot as plt

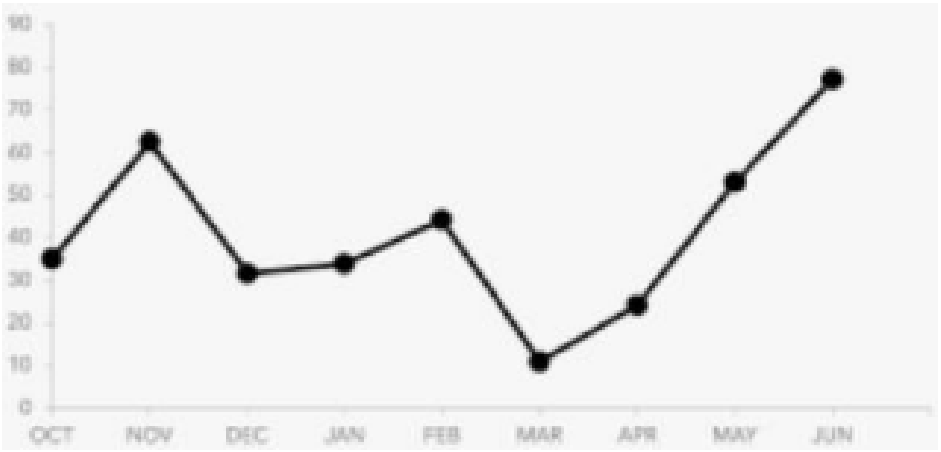
Following are some of the basic data visualization plots

1. Line plots
2. Area plots
3. Histograms
4. Bar charts
5. Pie charts
6. Box plots

Line Plots:

A line plot is used to represent quantitative values over a continuous interval or time period. It is generally used to depict trends on how the data has changed over time.

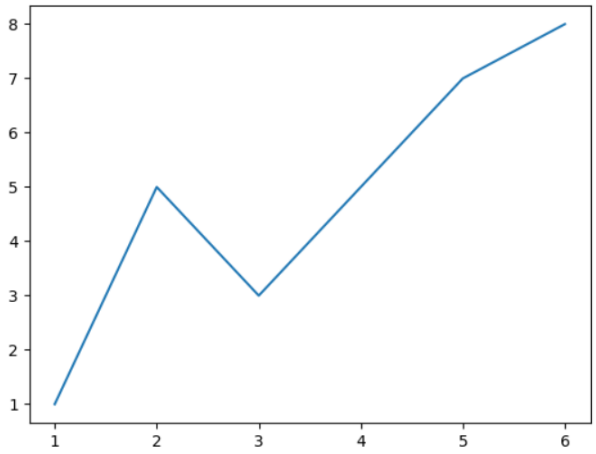
Example:



Program:

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5, 6]
y = [1, 5, 3, 5, 7, 8]
plt.plot(x, y)
plt.show()
```

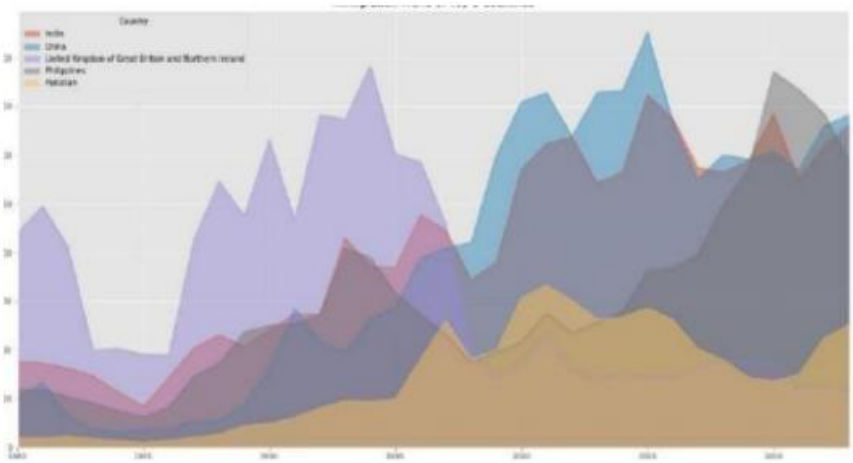
Output:



Area Plots:

An Area Plot is also called as Area Chart which is used to display magnitude and proportion of multiple variables.

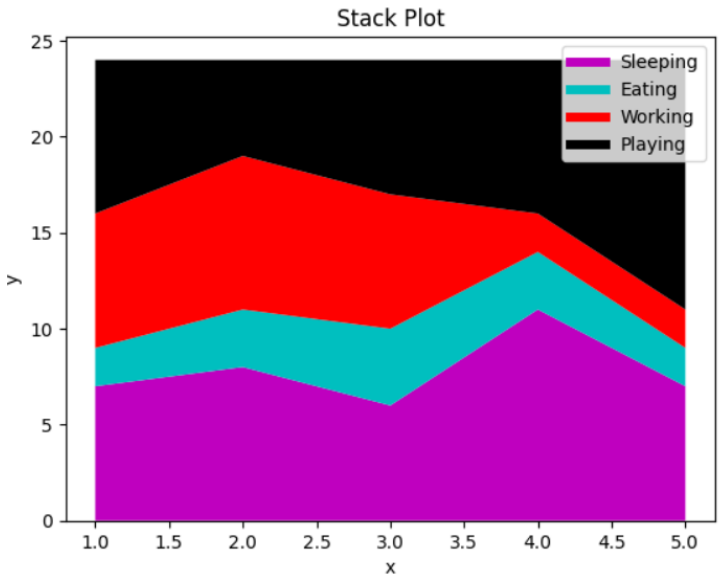
Example:



Program:

```
import matplotlib.pyplot as plt
days = [1,2,3,4,5]
sleeping =[7,8,6,11,7]
eating = [2,3,4,3,2]
working =[7,8,7,2,2]
playing = [8,5,7,8,13]
plt.plot([],[],color='m', label='Sleeping', linewidth=5)
plt.plot([],[],color='c', label='Eating', linewidth=5)
plt.plot([],[],color='r', label='Working', linewidth=5)
plt.plot([],[],color='k', label='Playing', linewidth=5)
plt.stackplot(days, sleeping,eating,working,playing, colors=['m','c','r','k'])
plt.xlabel('x')
plt.ylabel('y')
plt.title('Stack Plot')
plt.legend()
plt.show()
```

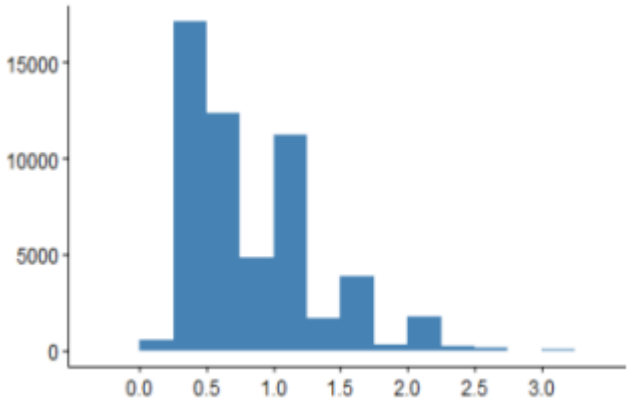
Output:



Histograms:

Histograms represents the frequency distribution of a dataset. It is a graph showing the number of observations within each given interval.

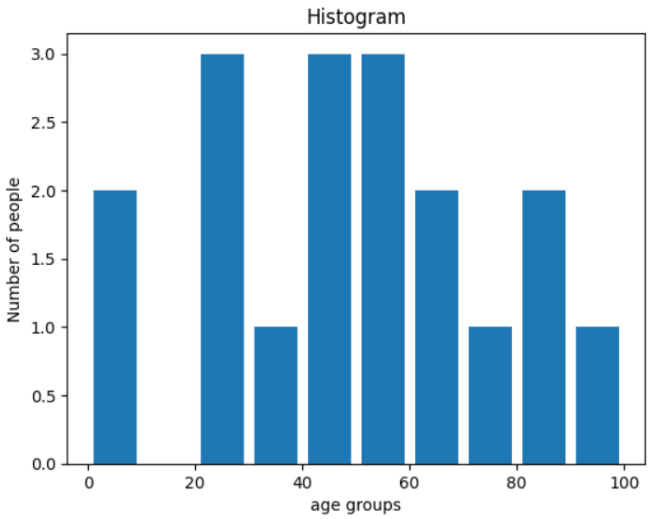
Example:



Program:

```
import matplotlib.pyplot as plt
population_age=[22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,120,70,65,55,111,115,80]
bins = [0,10,20,30,40,50,60,70,80,90,100]
plt.hist(population_age, bins, histtype='bar', rwidth=0.8)
plt.xlabel('age groups')
plt.ylabel('Number of people')
plt.title('Histogram')
plt.show()
```

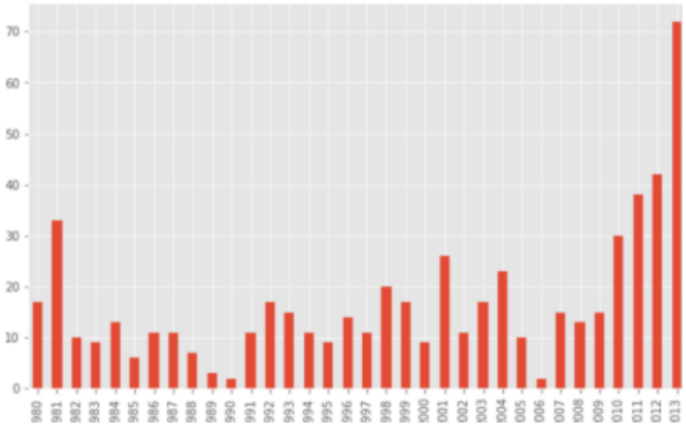
Output:



Bar Charts:

A Bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. A bar plot is a way of representing data where the length of the bars represents the magnitude/size of the feature/variable.

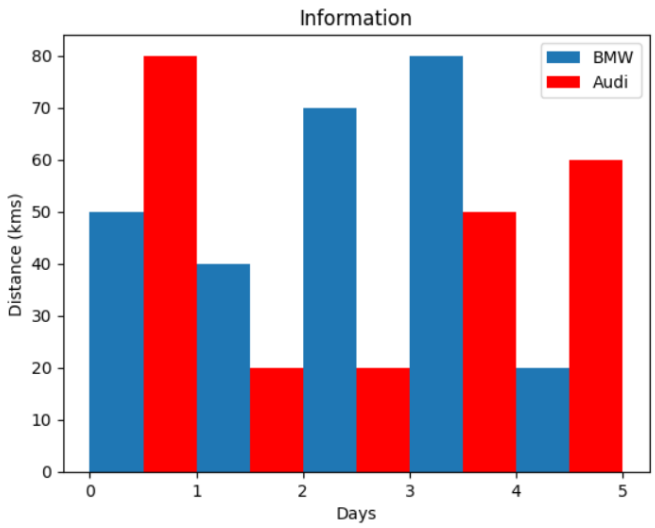
Example:



Program:

```
from matplotlib import pyplot as plt
plt.bar([0.25,1.25,2.25,3.25,4.25],[50,40,70,80,20],label="BMW",width=.5)
plt.bar([.75,1.75,2.75,3.75,4.75],[80,20,20,50,60],label="Audi", color='r',width=.5)
plt.legend()
plt.xlabel('Days')
plt.ylabel('Distance (kms)')
plt.title('Information')
plt.show()
```

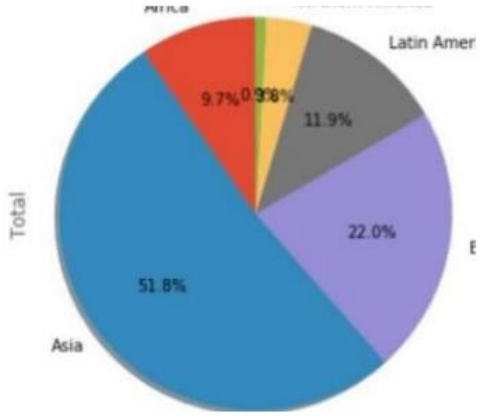
Output:



Pie Charts: A

Pie chart is a circular statistical chart, which is divided into sectors to illustrate numerical proportion.

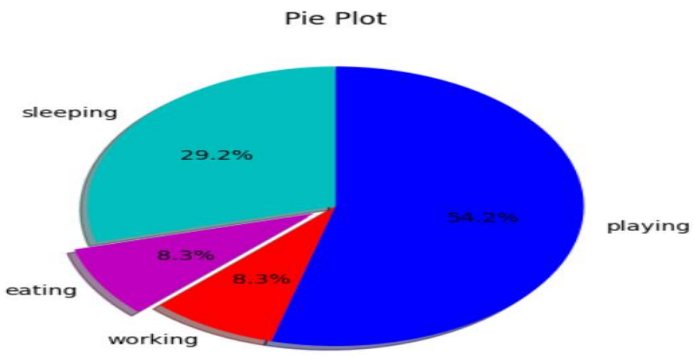
Example:



Program:

```
import matplotlib.pyplot as plt
days = [1,2,3,4,5]
sleeping =[7,8,6,11,7]
eating = [2,3,4,3,2]
working =[7,8,7,2,2]
playing = [8,5,7,8,13]
slices = [7,2,2,13]
activities = ['sleeping','eating','working','playing']
cols = ['c','m','r','b']
plt.pie(slices, labels=activities, colors=cols, startangle=90, shadow= True,
explode=(0,0.1,0,0), autopct='%1.1f%%')
plt.title('Pie Plot')
plt.show()
```

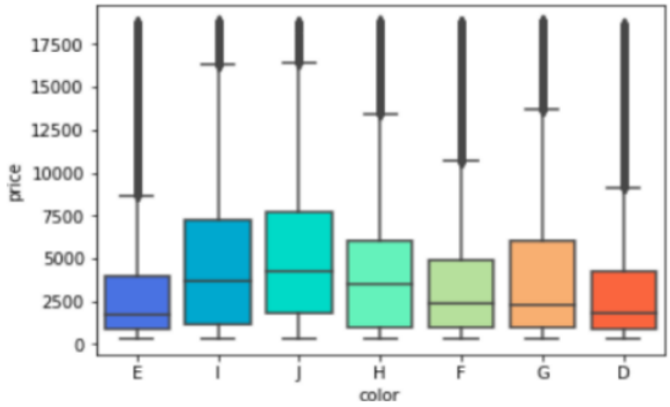
Output:



Box Plots:

A Box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. Box plot shows the quartiles of the dataset while the whiskers extend encompass the rest of the distribution but leave out the points that are the outliers.

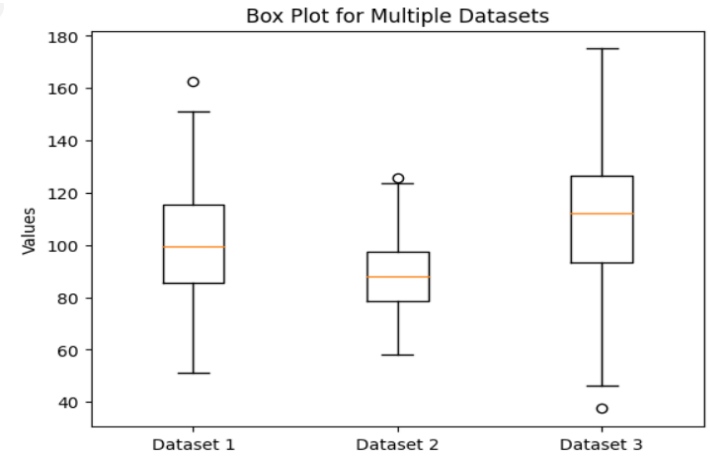
Example



Program:

```
import numpy as np
import matplotlib.pyplot as plt
data1 = np.random.normal(100, 20, 200)
data2 = np.random.normal(90, 15, 200)
data3 = np.random.normal(110, 25, 200)

# Create a box plot for multiple datasets
data = [data1, data2, data3]
plt.boxplot(data, labels=['Dataset 1', 'Dataset 2', 'Dataset 3'])
plt.title('Box Plot for Multiple Datasets')
plt.ylabel('Values')
plt.show()
```

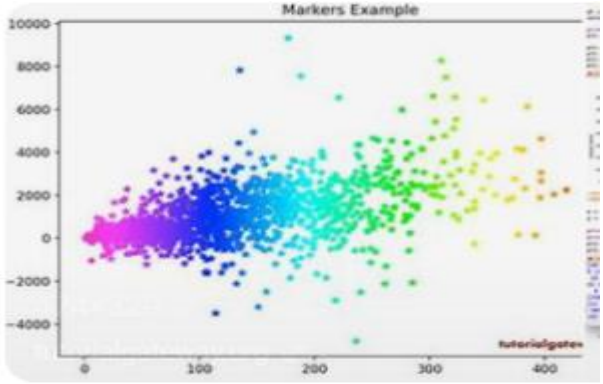
Output:



Scatter Plots:

A Scatter chart, also called a scatter plot, is a chart that shows the relationship between two variables

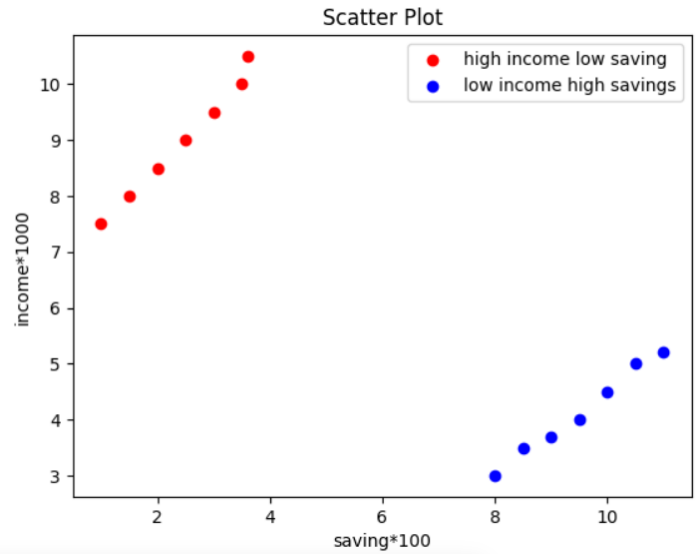
Example:



Program:

```
import matplotlib.pyplot as plt
x=[1,1.5,2,2.5,3,3.5,3.6]
y=[7.5,8,8.5,9,9.5,10,10.5]
x1=[8,8.5,9,9.5,10,10.5,11]
y1=[3,3.5,3.7,4,4.5,5,5.2]
plt.scatter(x,y, label='high income low saving',color='r')
plt.scatter(x1,y1,label='low income high savings',color='b')
plt.xlabel('saving*100')
plt.ylabel('income*1000')
plt.title('Scatter Plot')
plt.legend()
plt.show()
```

Output:



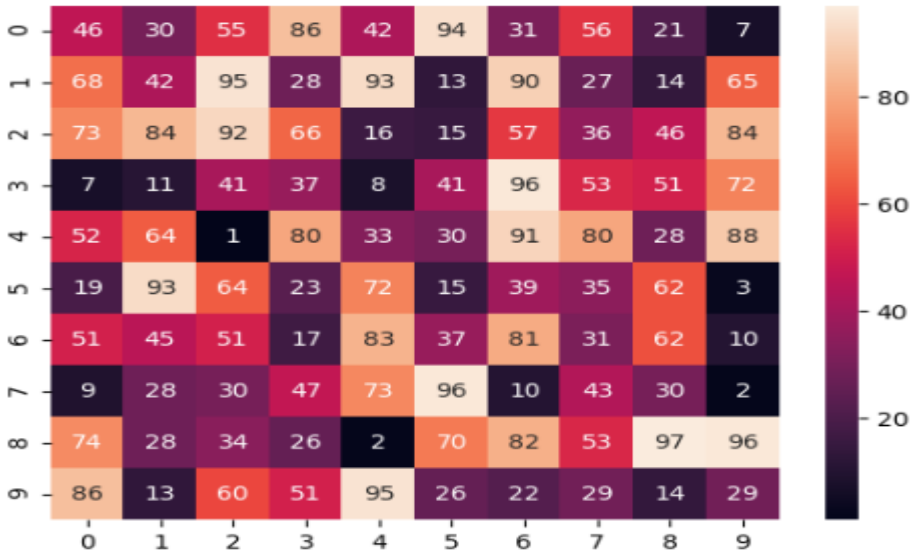
Heatmap:

Heatmap is defined as a graphical representation of data using colors to visualize the value of the matrix. In this, to represent more common values or higher activities brighter colors basically reddish colors are used and to represent less common or activity values, darker colors are preferred.

Program:

```
import numpy as np
import seaborn as sn
import matplotlib.pyplot as plt

# generating 2-D 10x10 matrix of random numbers
# from 1 to 100
data = np.random.randint(low=1,
                         high=100,
                         size=(10, 10))

# setting the parameter values
annot = True

# plotting the heatmap
hm = sn.heatmap(data=data,
                annot=annot)

# displaying the plotted heatmap
plt.show()
```

Output:



Experiment-2

Getting started with Tableau Software using Data file formats, connecting your Data to Tableau, creating basic charts(line, bar charts, Tree maps), Using the Show me panel.

Aim:- Getting started with Tableau Software using Data file formats, connecting your Data to Tableau,creating basic charts(line, bar charts, Tree maps),Using the Show me panel.

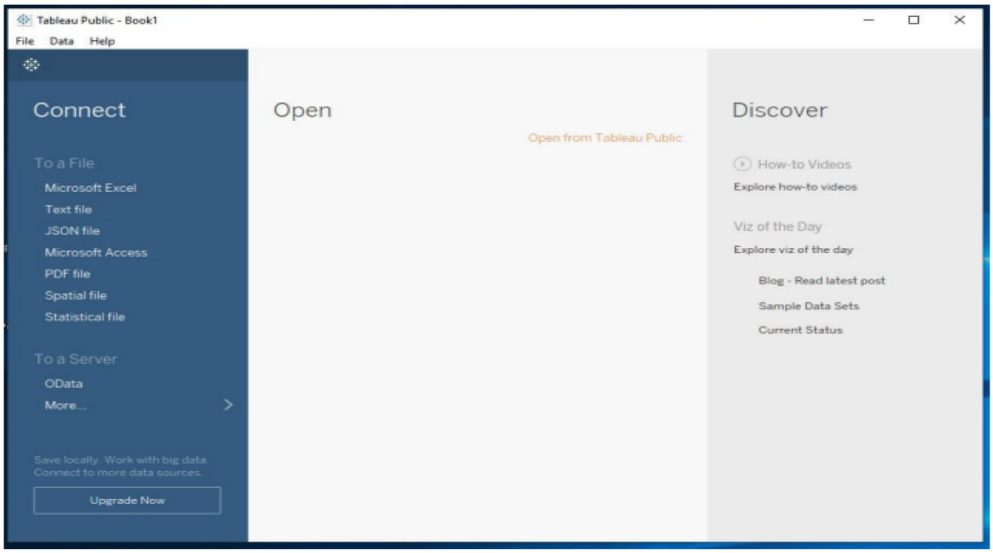
Solution:- Getting started with Tableau software is a great way to create data visualizations quickly and efficiently. Here are the steps to get started, including connecting your data to Tableau, creating basic charts like line charts, bar charts, and treemaps, and using the Show Me panel:

Tableau is a data visualization tool that provides pictorial and graphical representations of data. It is used for data analytics and business intelligence. Tableau provides limitless data exploration without interrupting flow of analysis. With an intuitive drag and drop interface, user can uncover hidden insights in data and make smarter decisions faster.

1. Download and Install Tableau: First, you'll need to download and install Tableau Desktop or Tableau Public (a free version). Follow the installation instructions provided on the Tableau website for your specific operating system.

Tableau can be downloaded from the following website: https://www.tableau.com/products/public/download after downloading, the following is the screen appear

Click the licence agreement checkbox and then click on install button. After installation, click on Tableau Public icon to run Tableau. Following is the Tableau Public home screen.

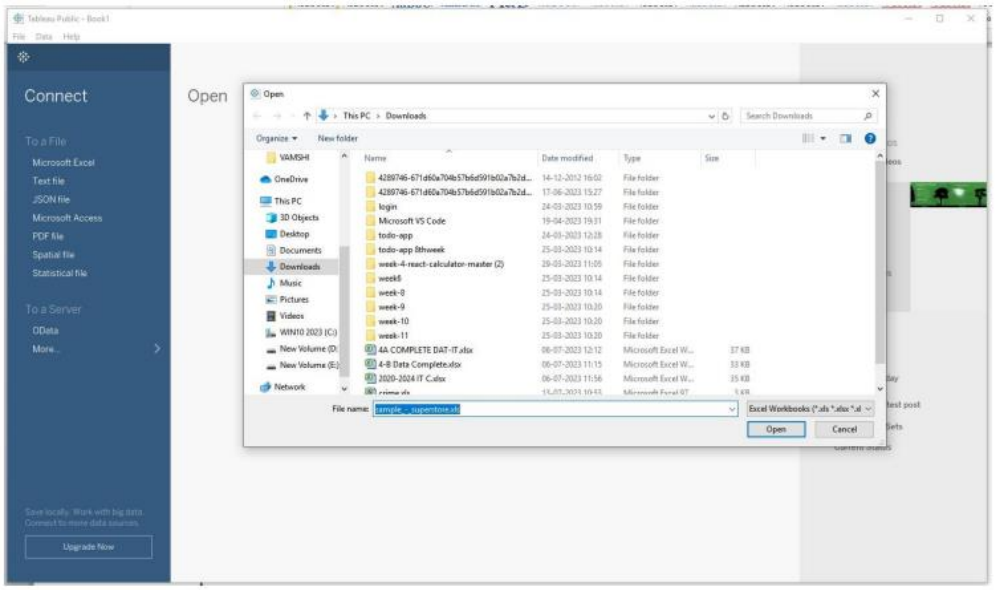


2. Prepare Your Data: Before connecting your data to Tableau, ensure that your data is in a suitable format. Common data file formats that Tableau supports include Excel (.xlsx), CSV (.csv), and text files (.txt). Make sure your data is organized with headers for each column.

3. Connect Your Data to Tableau:

3.1 Launch Tableau Desktop.

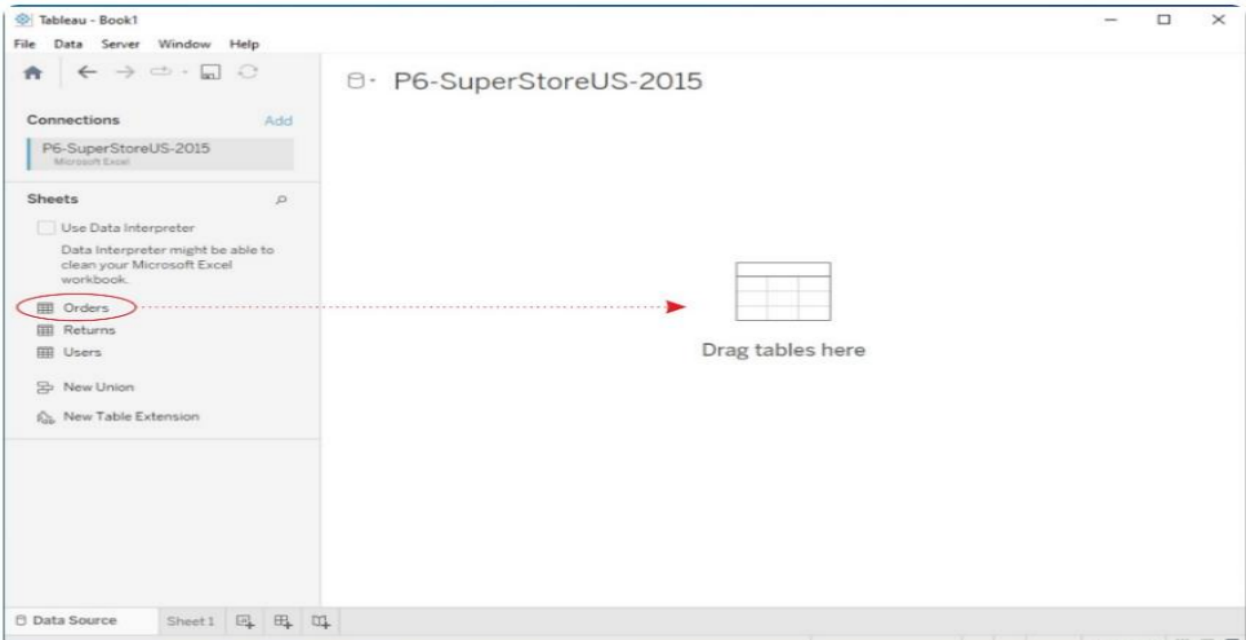
3.2 Go to "File" Menu and then click on "Open".



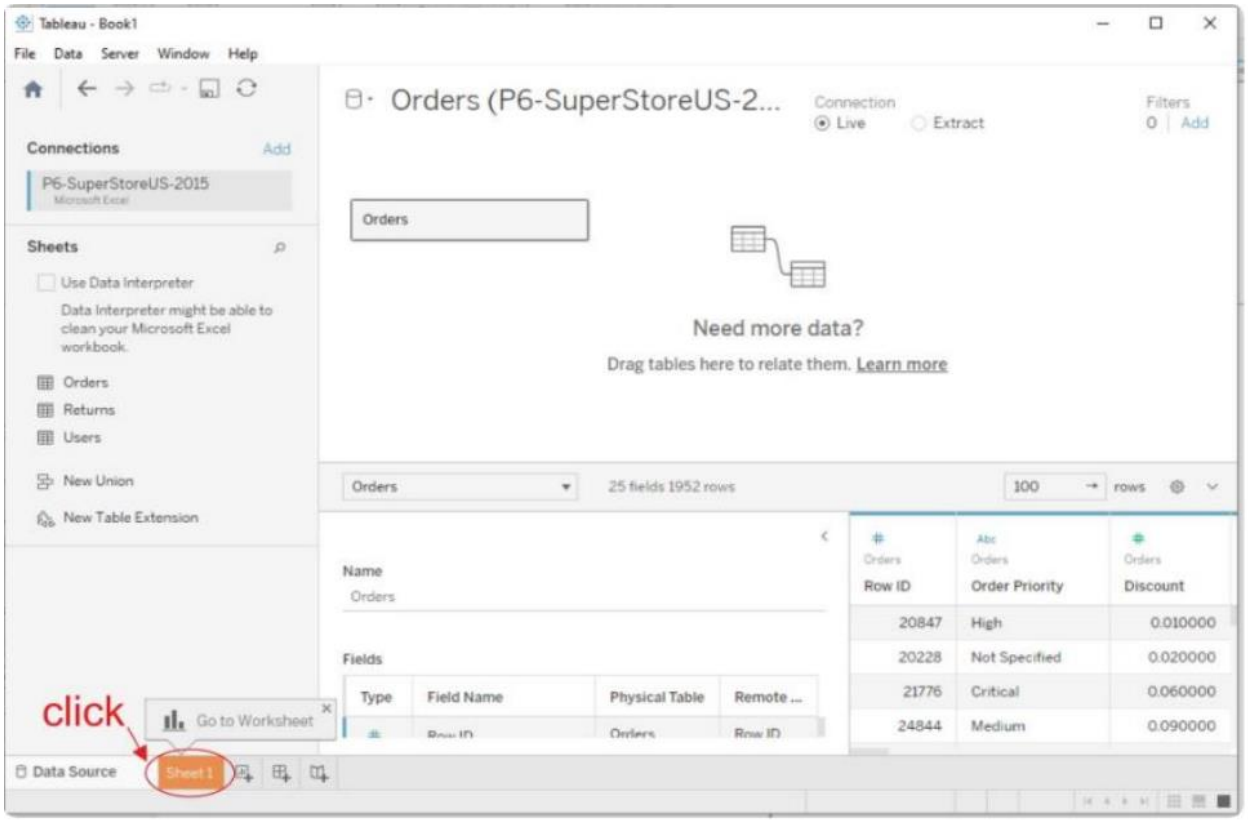
After clicking on open, screen is as follows:

3.3 Choose the data source type (e.g., Excel, CSV, text file) and Select the data file(P6-SuperStoreUS-2015.xls) and click "Open".

3.4 Drag any table(e.g. Orders) into working area.

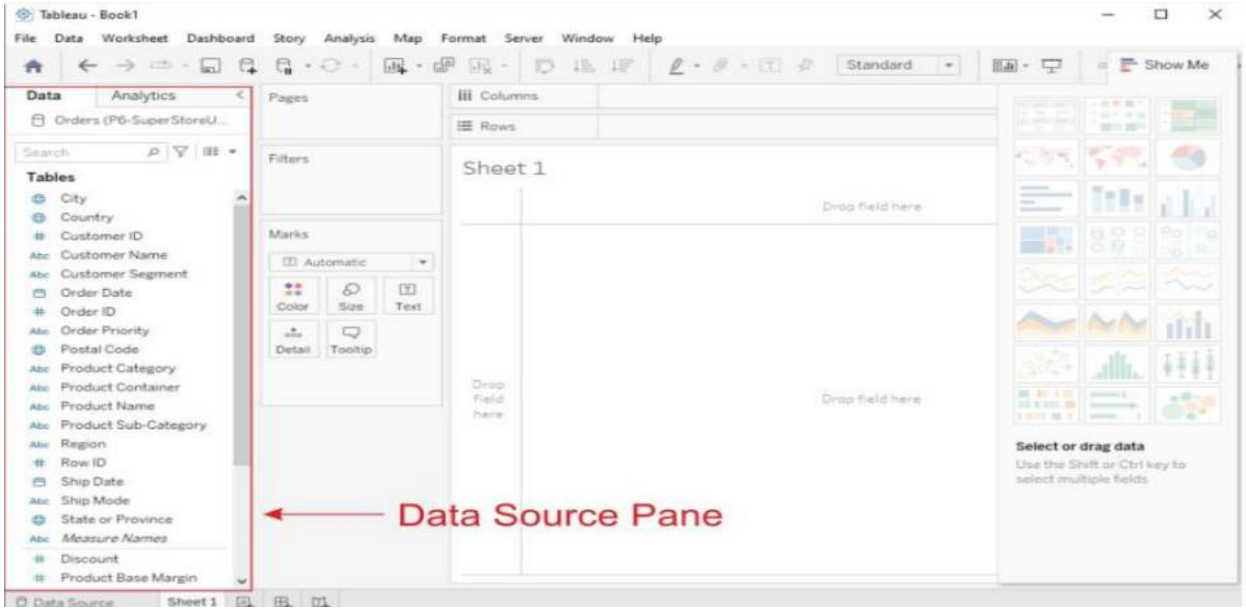


3.5 Click on Worksheet(Sheet1).



4. Data Source Pane:

Once your data is connected, the Data Source Pane will appear on the left-hand side of the Tableau interface. Here, you can see a preview of your data and perform data transformations or join multiple data sources if necessary.

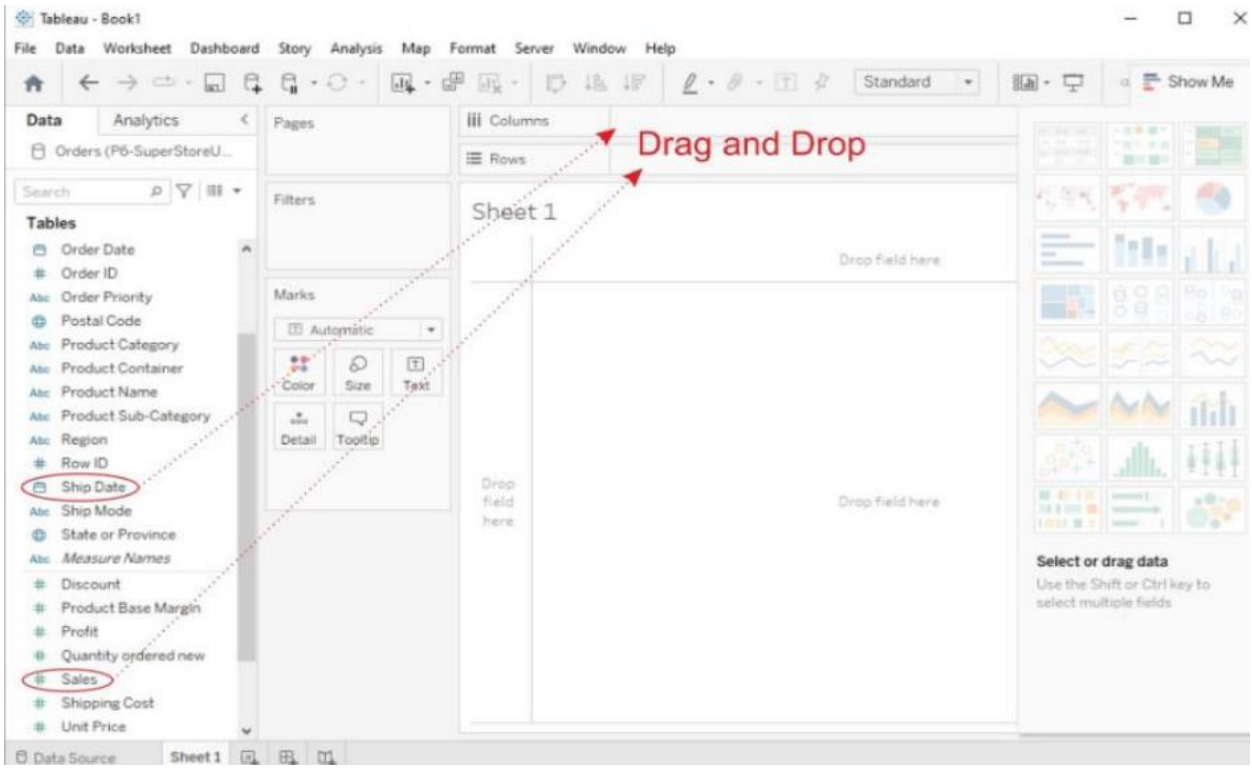


5. Creating Basic Charts: Data Visualization

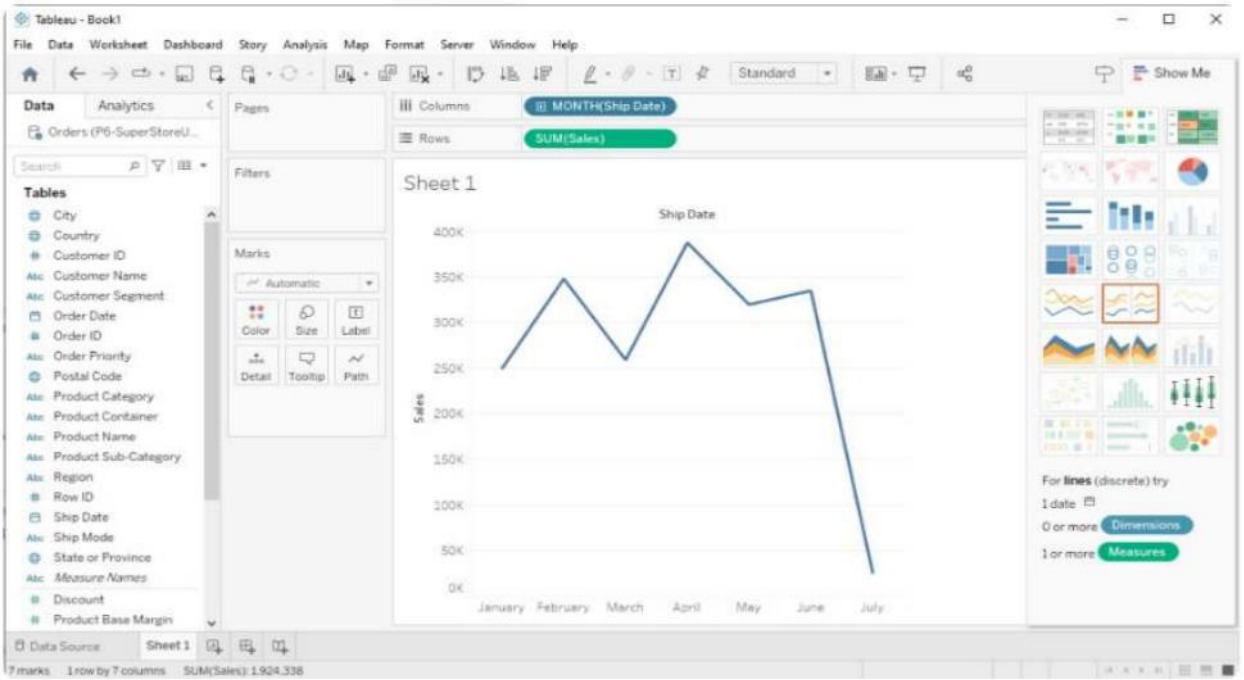
Now, let's create some basic charts using Tableau:

a. Line Chart:

1. From the "Data Source pane", drag and drop the date field to the Columns shelf and a numeric field (e.g., sales, revenue) to the Rows shelf.

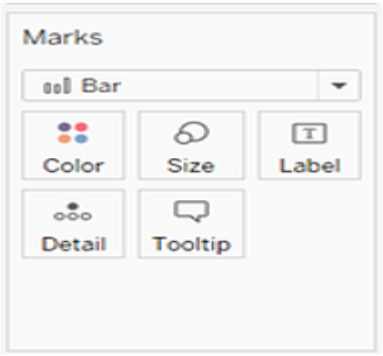


2. Then Tableau will automatically create a line chart. You can customize it by adding labels, titles, and formatting



We can perform various visualization operations on data in Tableau. Some of them are bar cart, histogram, bubble chart, gantt chart, scatter plot, heat map etc.

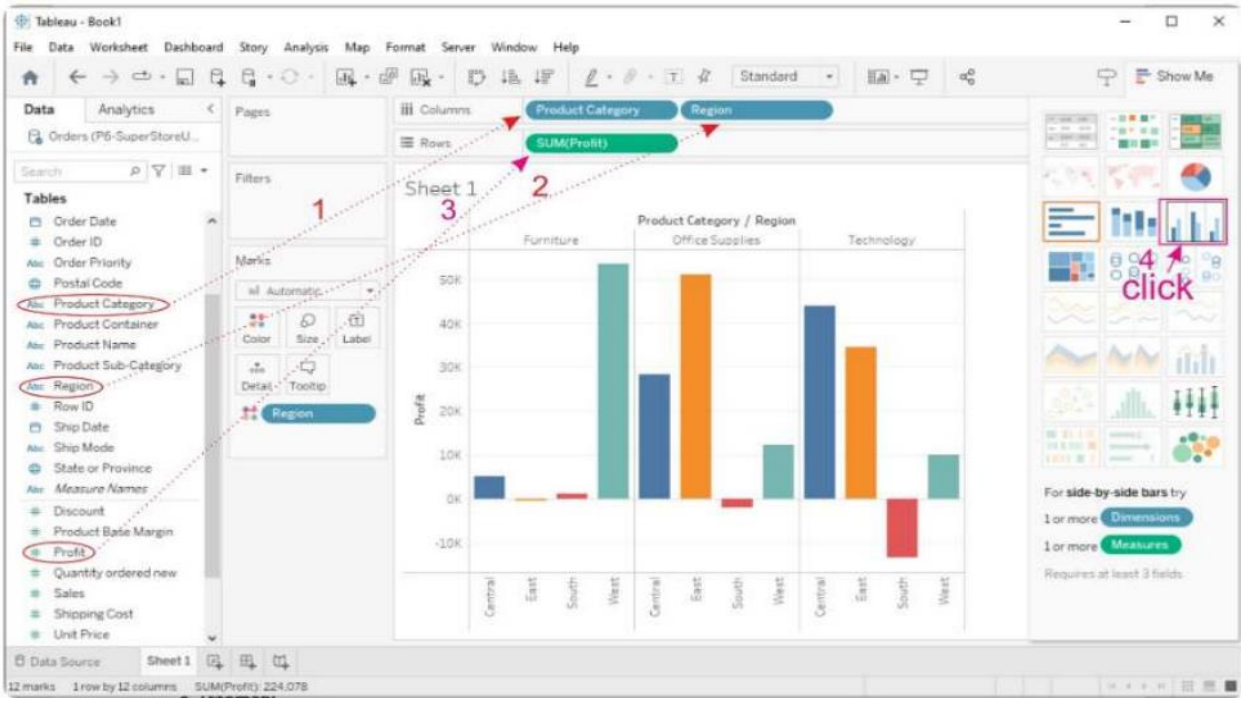
On the **Marks** card, select **Bar** from the drop-down list.



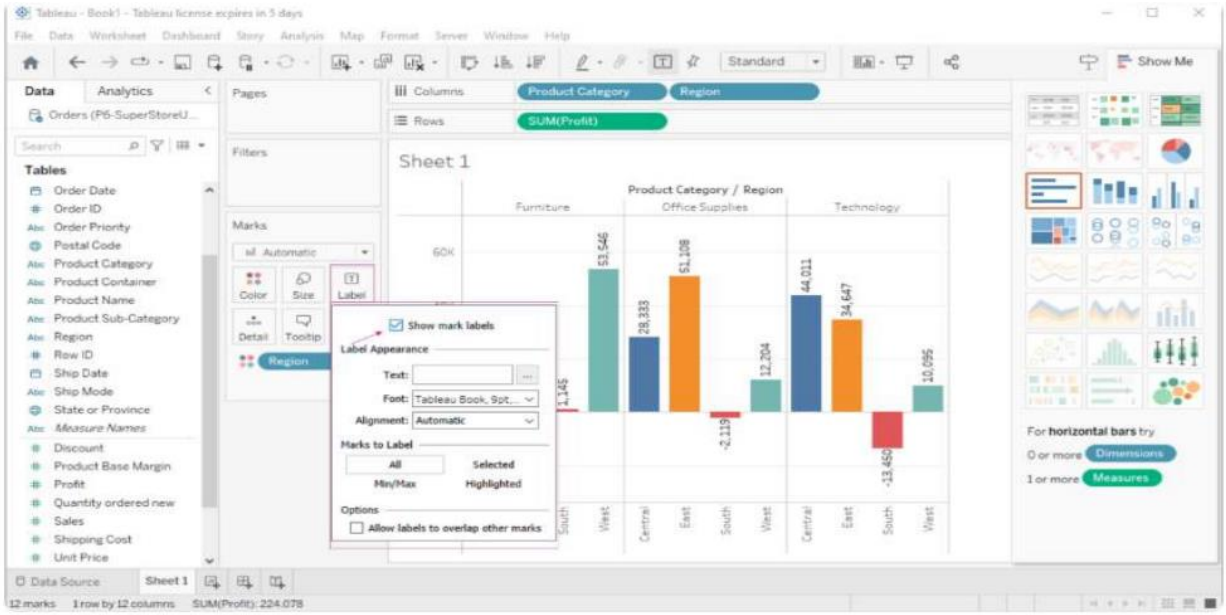
b. Bar chart:

Bar charts can be created in 3 variations in Tableau: Horizontal bars, stacked bars, side-by-side bars. Horizontal bars can be created by selecting that type of chart from Show Me menu on right hand side of Canvas. The type of chart in box on right hand side represents horizontal bar graph.

1. Drag and drop a categorical field (e.g., product category, region) to the Columns shelf and a numeric field to the Rows shelf.

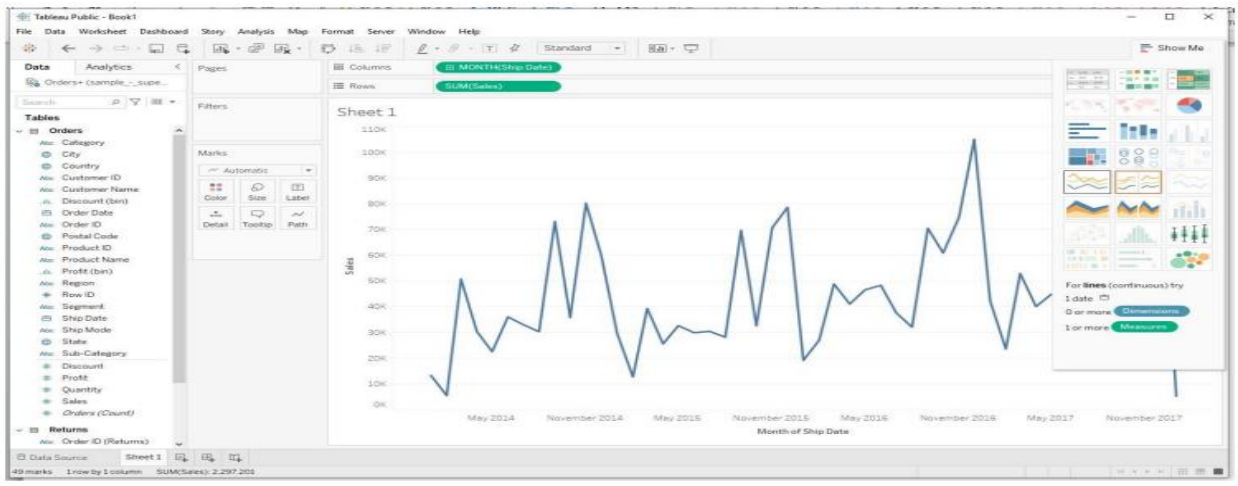


2. Then Tableau will create a bar chart. You can adjust the orientation and formatting as needed. To display Labels on the bars click on Lables and select "Show mark lables".

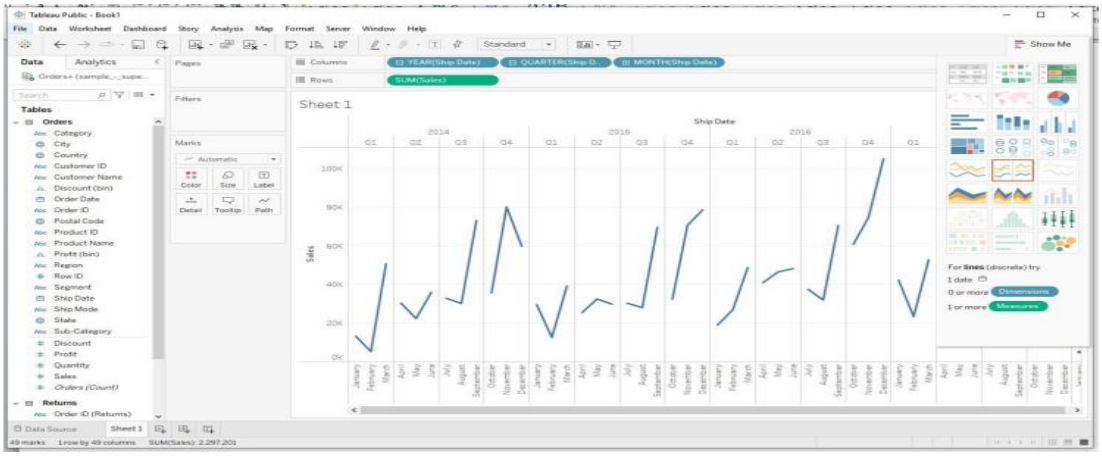


c. Line graph:

Line graph can be continuous or discrete. Continuous line graph is shown below

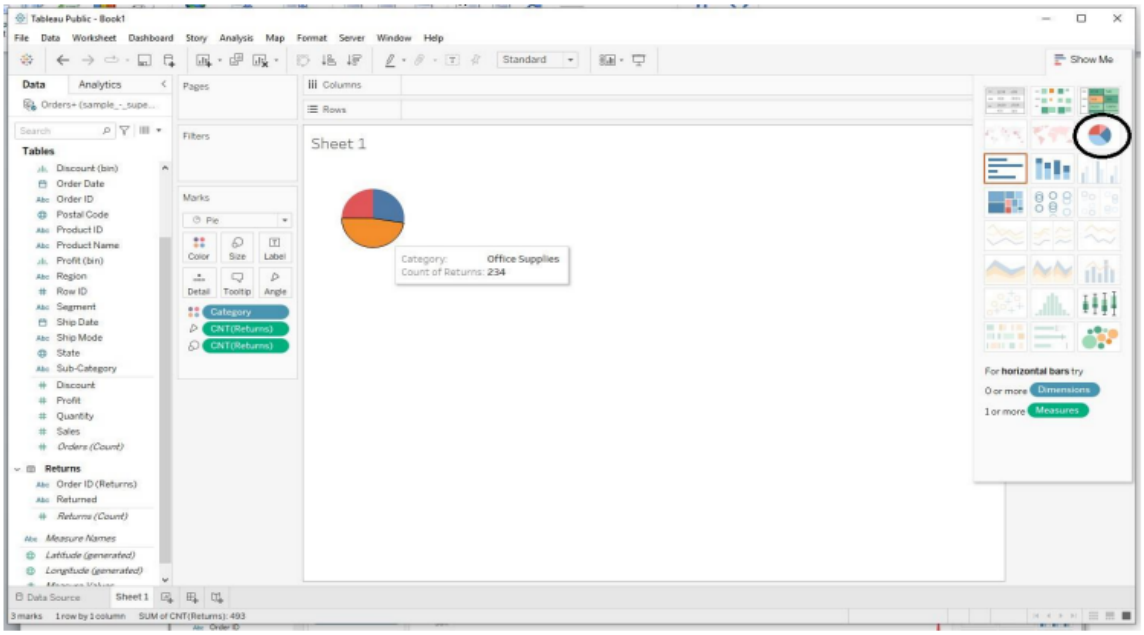


Discrete line graph is shown below

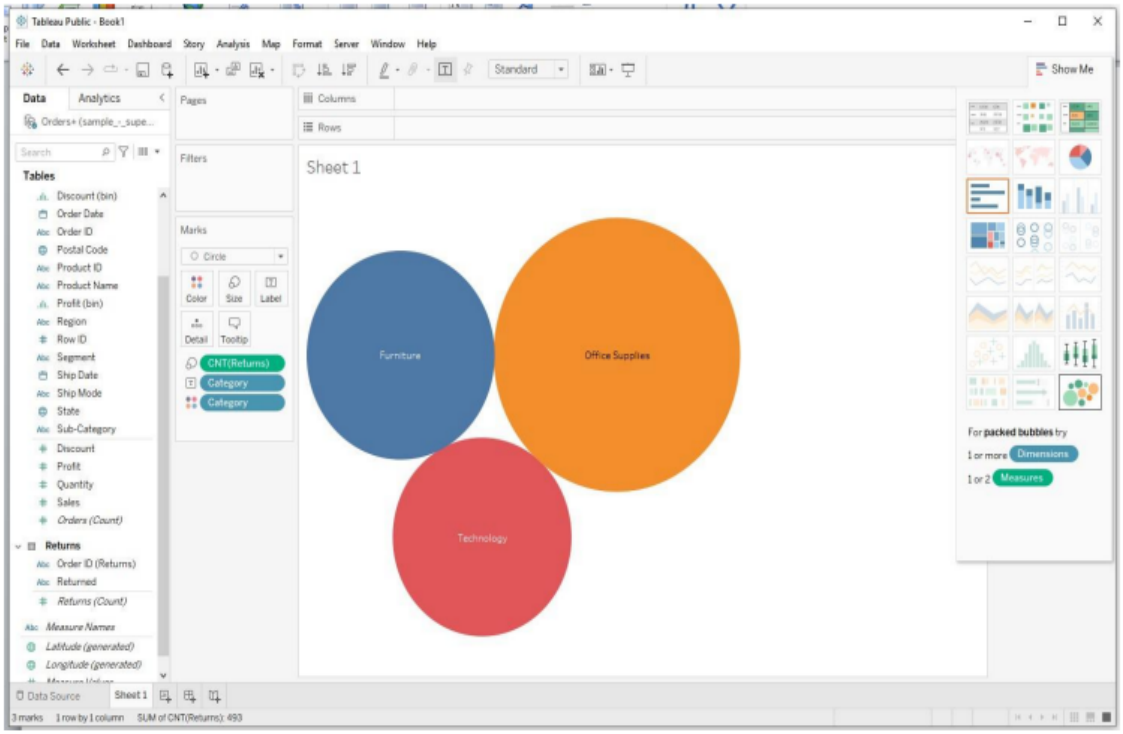


d. Piechart

Pie chart:



Bubble chart:



f. Treemap:

1. Drag and drop a categorical field to the Columns shelf.

2. Drag and drop a numeric field to the Size shelf.

3. Tableau will create a treemap visualization. You can further customize it by adjusting colors and labels.

6. Using the Show Me Panel:-

The Show Me panel in Tableau helps you explore various chart types based on your data and the fields you select. Here's how to use it:

1. After adding fields to the Rows and Columns shelves, click on the "Show Me" panel located on the left side of the Tableau interface.

2. In the Show Me panel, you'll see a variety of chart options that Tableau recommends based on your data. Click on a chart type to create it.

3. Tableau will automatically generate the selected chart type with your data. You can further customize it as needed.

4. To go back to the regular worksheet view, click the "Clear" button in the Show Me panel.

Experiment 3: Tableau Calculations, Overview of SUM, AVR, and Aggregate features, Creating custom calculation sand fields.

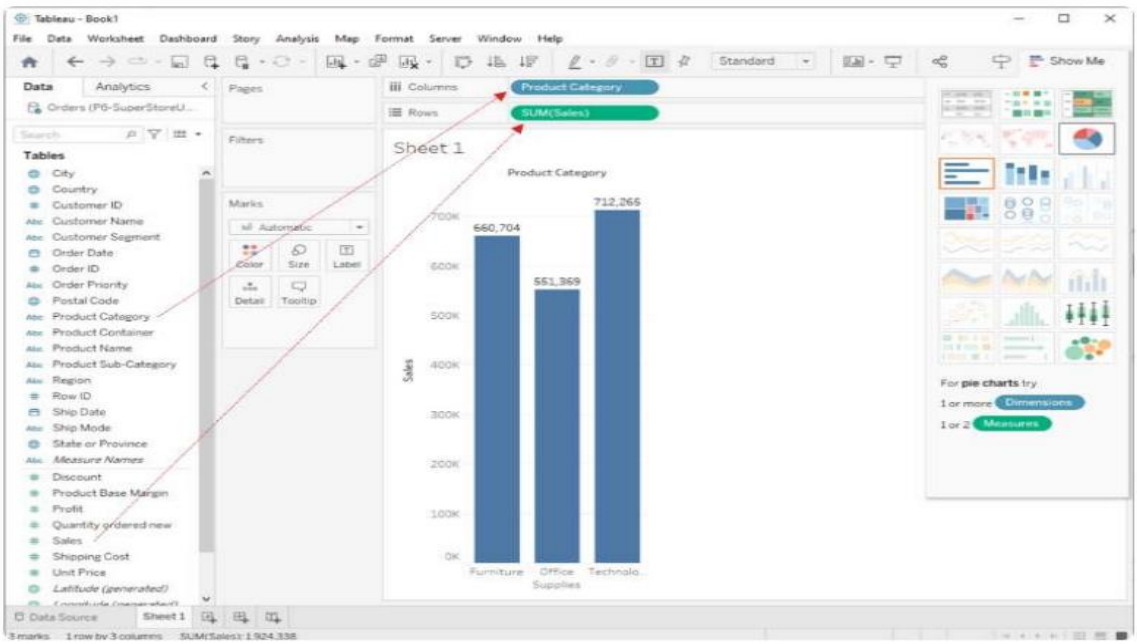
Aim:- Tableau Calculations, Overview of SUM, AVR, and Aggregate features, Creating custom calculations and fields.

Solution:- Tableau offers a powerful set of calculation tools that allow you to manipulate, transform, and analyze your data in various ways. Here's an overview of some key concepts related to Tableau calculations, including SUM, AVG (average), and aggregate functions, as well as creating custom calculations and fields.

SUM and AVG (Average) Functions:-

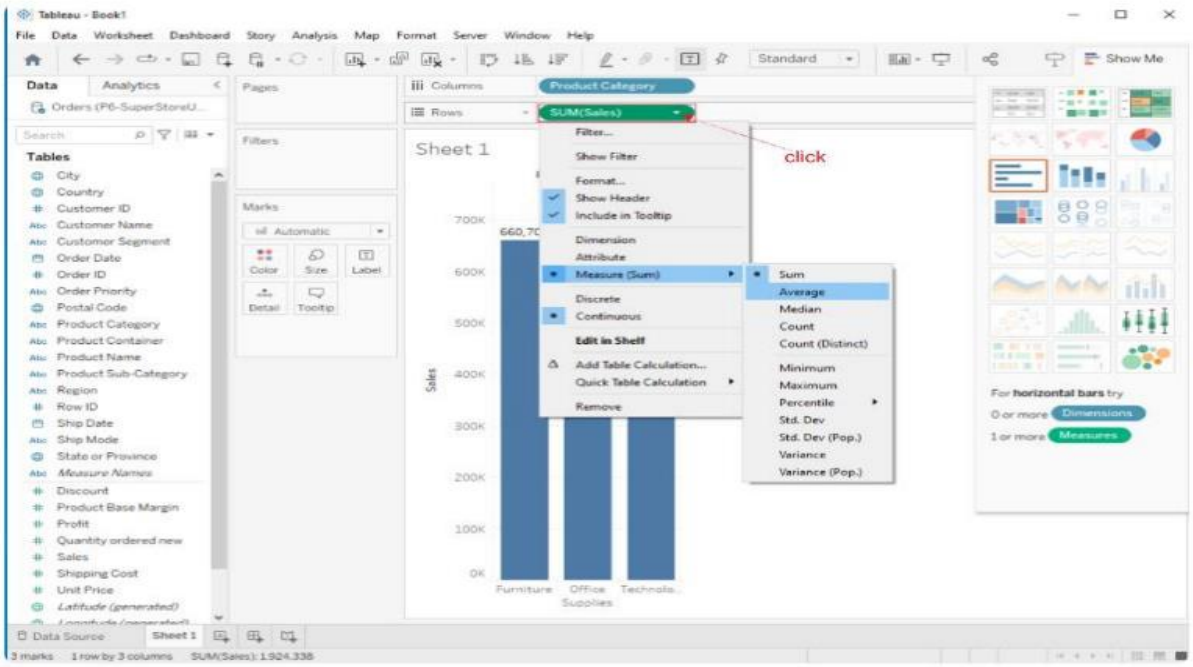
SUM Function

The SUM function in Tableau calculates the total sum of a numeric field. You can use it to find the sum of values in a column or as part of a more complex calculation. To use SUM, simply drag and drop a numeric field into the "SUM" shelf, or you can create a calculated field using the SUM function.



AVG (Average) Function

The AVG function calculates the average (mean) value of a numeric field. Like SUM, you can use it by dragging a numeric field into the "AVG" shelf or creating a calculated field with the AVG function.

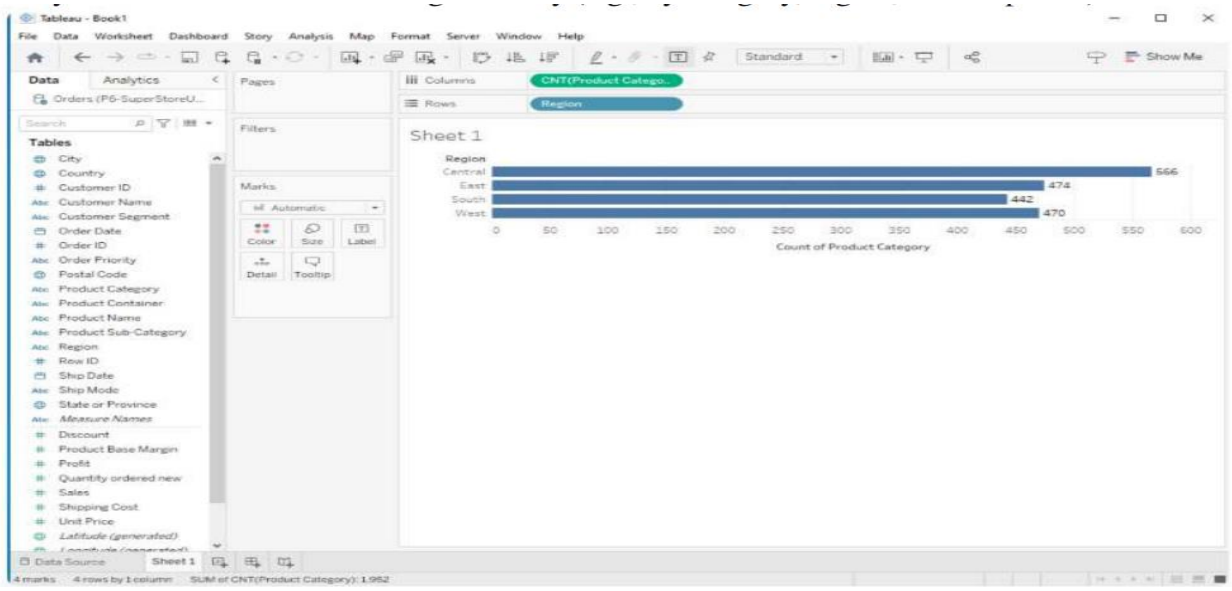


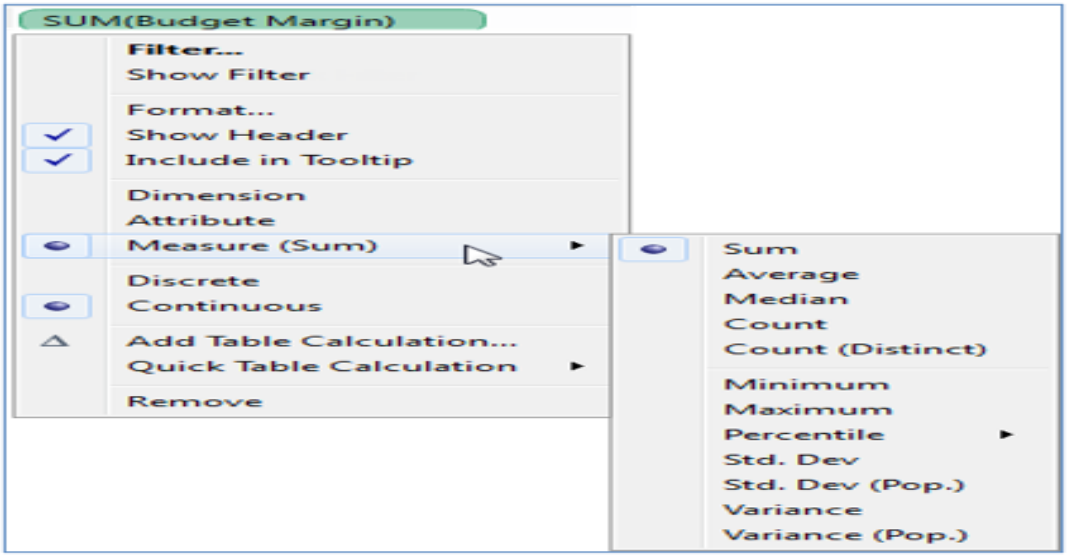
Aggregate Functions:

Tableau provides a range of aggregate functions that allow you to perform calculations on groups of data. Common aggregate functions include SUM, AVG, COUNT, MIN (minimum value), and MAX (maximum value). These functions are particularly useful when you want to analyze data at different levels of granularity (e.g., by category, region, or time period).

Aggregate functions allow you to summarize or change the granularity of your data.

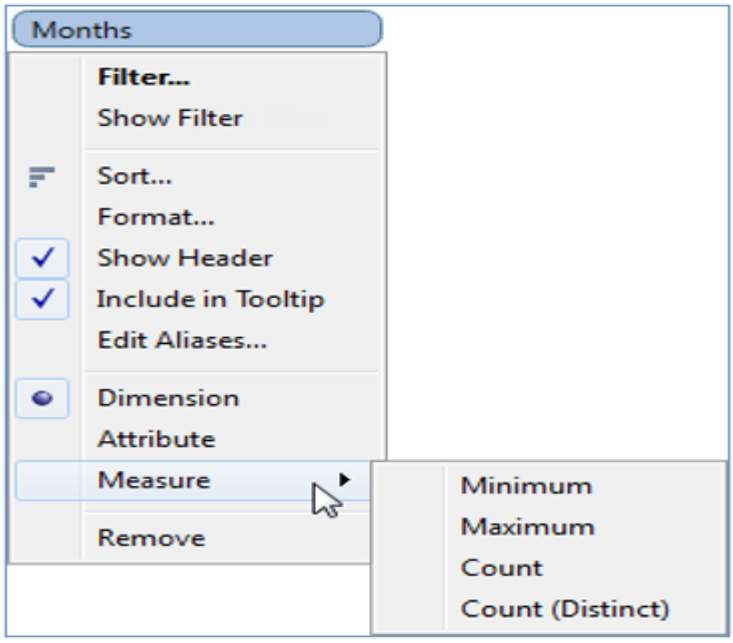
For example, you might want to know exactly how many orders your store had for a particular year. You can use the COUNTD function to summarize the exact number of orders your company had, and then break the visualization down by year.





Aggregating Dimensions

You can aggregate a dimension in the view as **Minimum**, **Maximum**, **Count**, or **Count (Distinct)**. When you aggregate a dimension, you create a new temporary measure column, so the dimension actually takes on the characteristics of a measure.

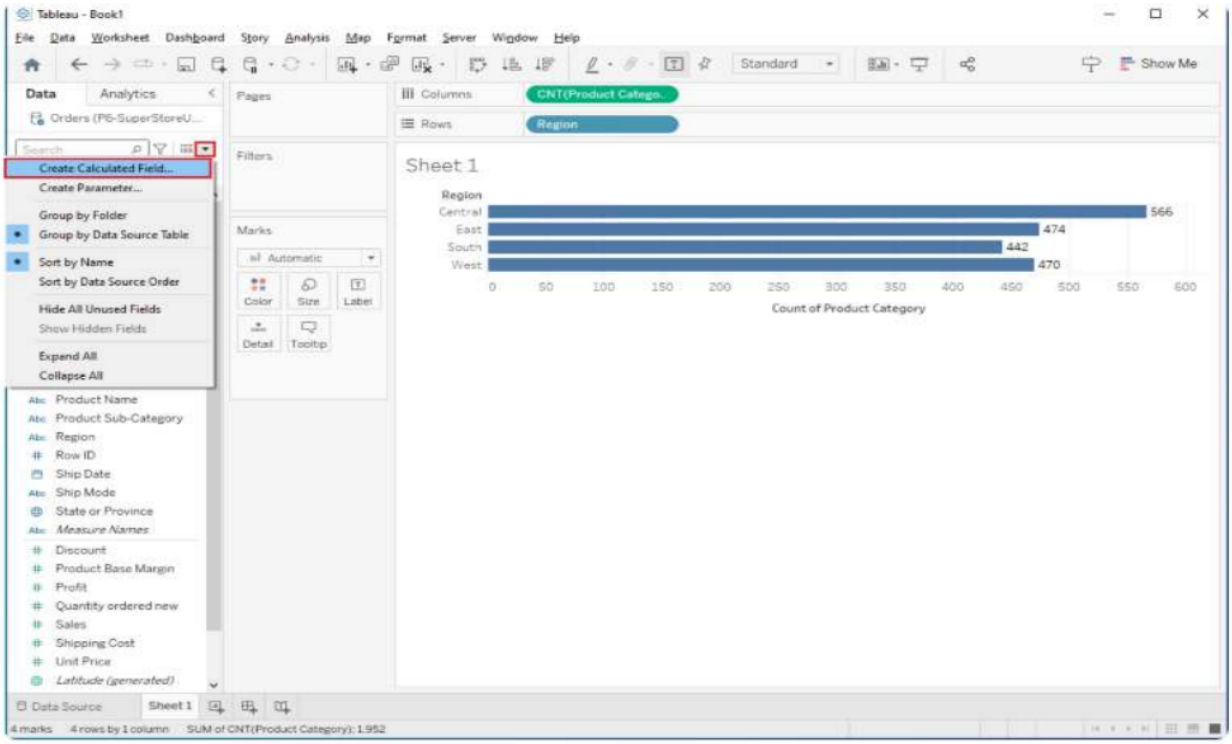


Creating Custom Calculations:

Tableau allows you to create custom calculations using calculated fields. Here's how to create a custom calculation: 1. Create a New Calculated Field In the Data Source Pane, right-click on your data source and select "Create Calculated Field".

1. Create a New Calculated Field

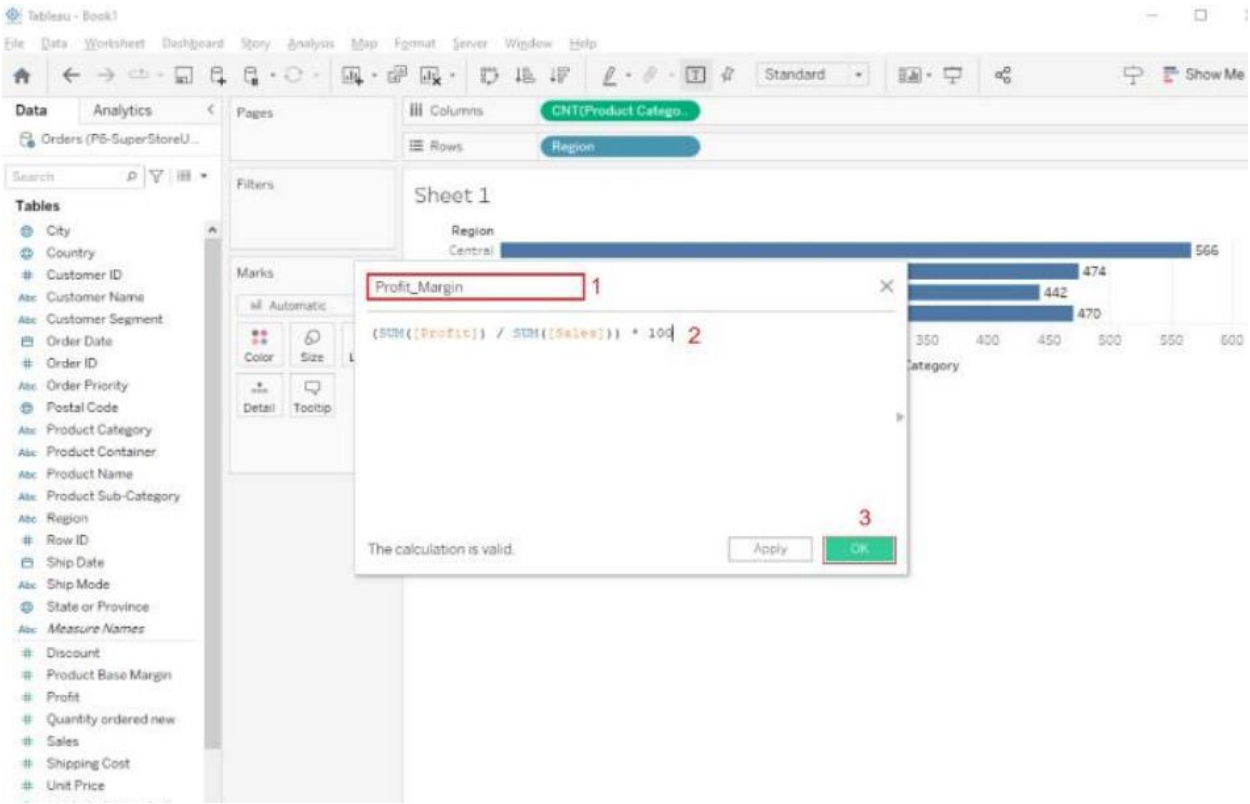
In the Data Source Pane, right-click on your data source and select "Create Calculated Field".



Alternatively, you can create a calculated field by right-clicking on a shelf in your worksheet and choosing "Create Calculated Field".

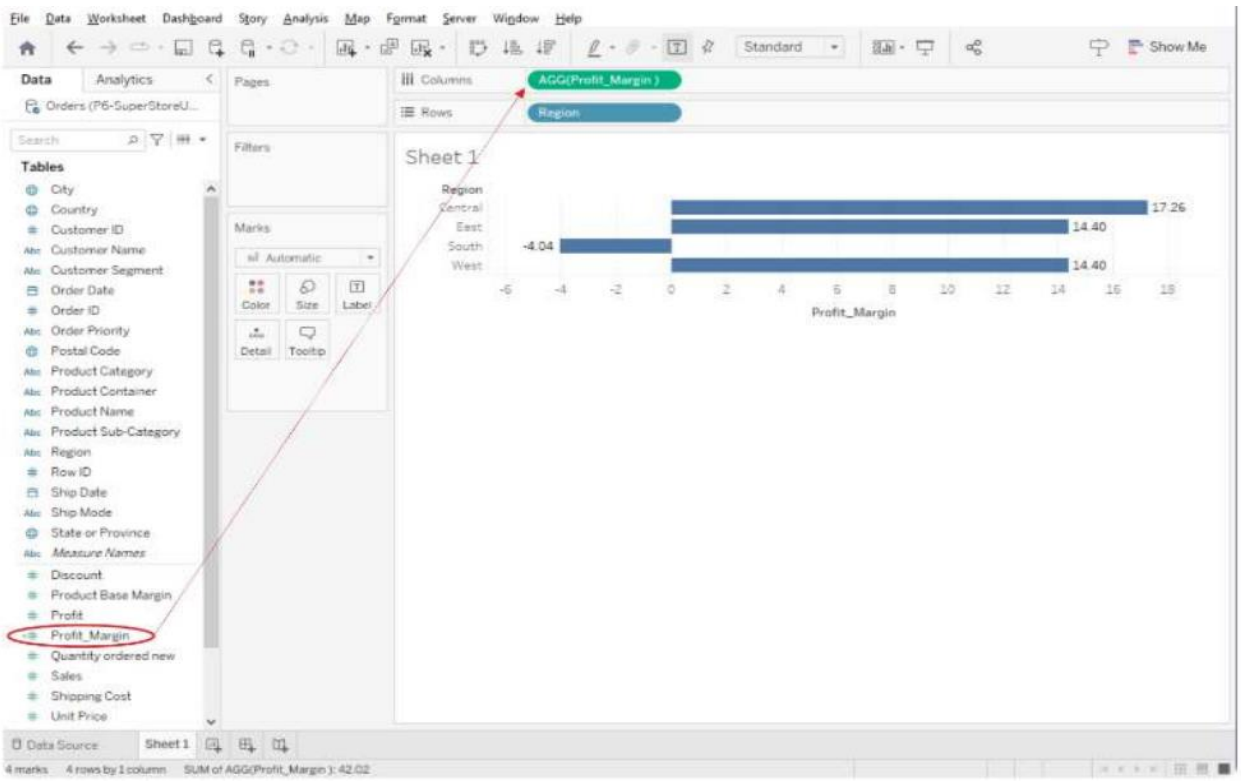
2. Enter Your Calculation: In the calculated field editor, you can use functions, operators, and field references to define your calculation. For example, you can create a calculated field to calculate profit margin as (SUM([Profit]) / SUM([Sales])) * 100.

3. Name and Save the Calculated Field: Give your calculated field a meaningful name. Click the "OK" or "Apply" button to save the calculated field.



4. Use the Calculated Field in Your Worksheet:

You can now use the calculated field like any other field in your worksheet. Drag it to the Rows or Columns shelf, use it in filters, or create visualizations based on it.



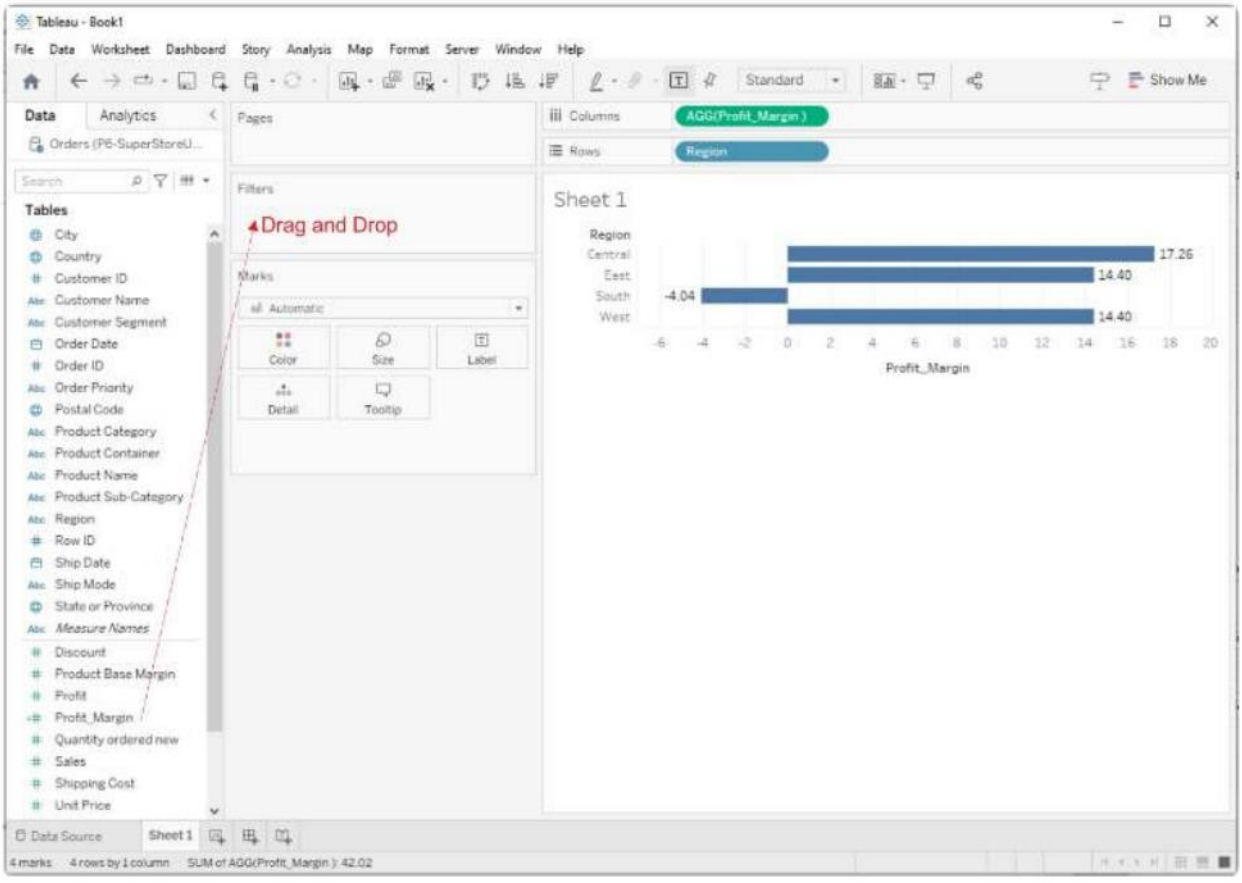
Experiment 4:

Applying new data calculations to your visualizations, Formatting Visualizations, Formatting Tools and Menus, and Formatting specific parts of the view.

Aim:- Applying new data calculations to your visualizations, Formatting Visualizations, Formatting Toolsand Menus, Formatting specific parts of the view.

Solution:- Applying New Data Calculations to Visualizations.

1. Drag and Drop Calculated Fields: To apply your newly created calculated fields to a visualization, simply drag and drop them onto the appropriate shelves in your worksheet. For example, you can drag a calculated field to the Rows or Columns shelf, use it in filters, or place it on the Marks card to control the appearance of marks.



2. Filter with Calculated Fields:

Create filters using calculated fields to control which data points are displayed in your visualization. You can use calculated fields to filter by specific criteria, such as a calculated date range or a custom ranking.

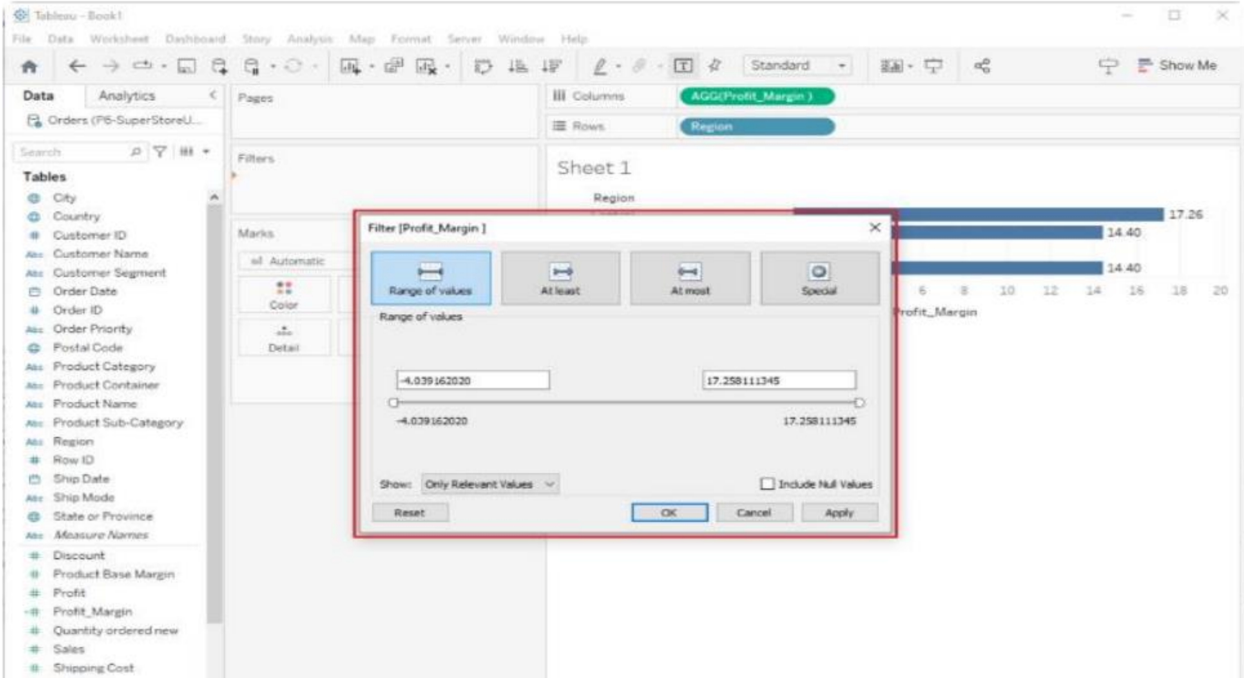
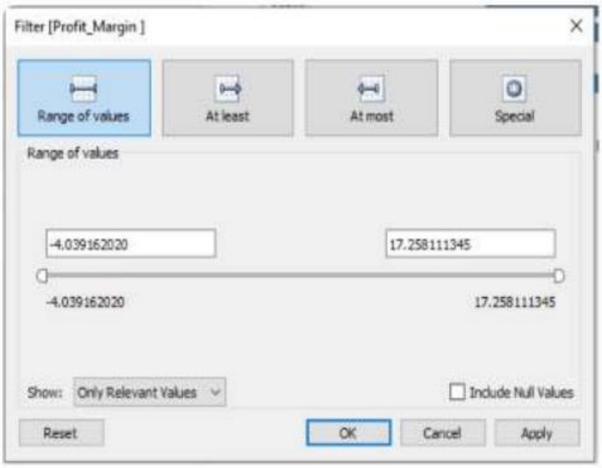
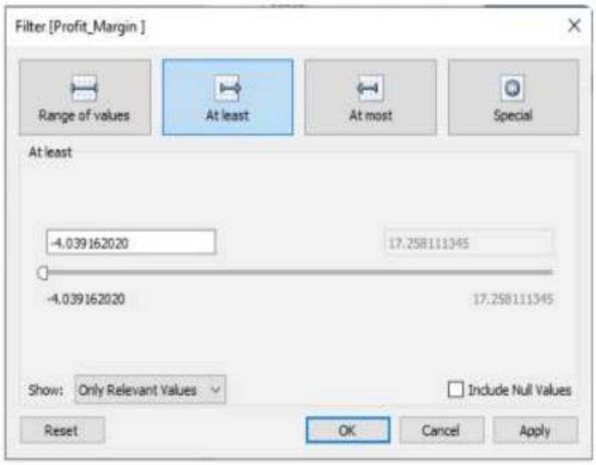
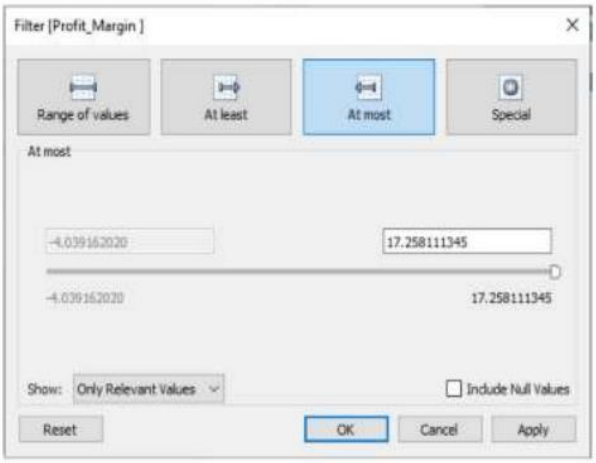
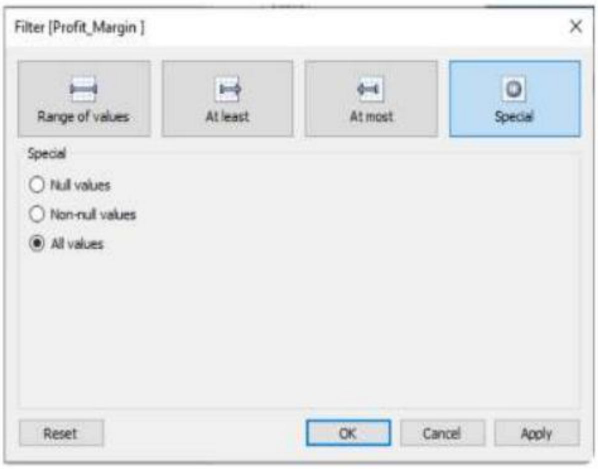


Tableau - Book1
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Show Me
Data
Analytics
Orders (P6-SuperStoreU...
Pages
Columns
AGG(Profit_Margin)
Rows
Region
Filters
Sheet 1
Region
Tables
City
Country
Customer ID
Customer Name
Customer Segment
Order Date
Order ID
Order Priority
Postal Code
Product Category
Product Container
Product Name
Product Sub-Category
Region
Row ID
Ship Date
Ship Mode
State or Province
Measure Names
Discount
Product Base Margin
Profit
Profit_Margin
Quantity ordered new
Sales
Shipping Cost
Marks
Automatic
Color
Detail
Filter [Profit_Margin]
Range of values
At least
At most
Special
Range of values
-4.039162020
17.258111345
-4.039162020
17.258111345
Show: Only Relevant Values
Include Null Values
Reset
OK
Cancel
Apply
17.26
14.40
14.40
6 8 10 12 14 16 18 20
Profit_Margin


Filter [Profit_Margin]
Range of values
At least
At most
Special
Range of values
-4.039162020
17.258111345
-4.039162020
17.258111345
Show: Only Relevant Values
Include Null Values
Reset
OK
Cancel
Apply


Filter [Profit_Margin]
Range of values
At least
At most
Special
At least
-4.039162020
17.258111345
-4.039162020
17.258111345
Show: Only Relevant Values
Include Null Values
Reset
OK
Cancel
Apply

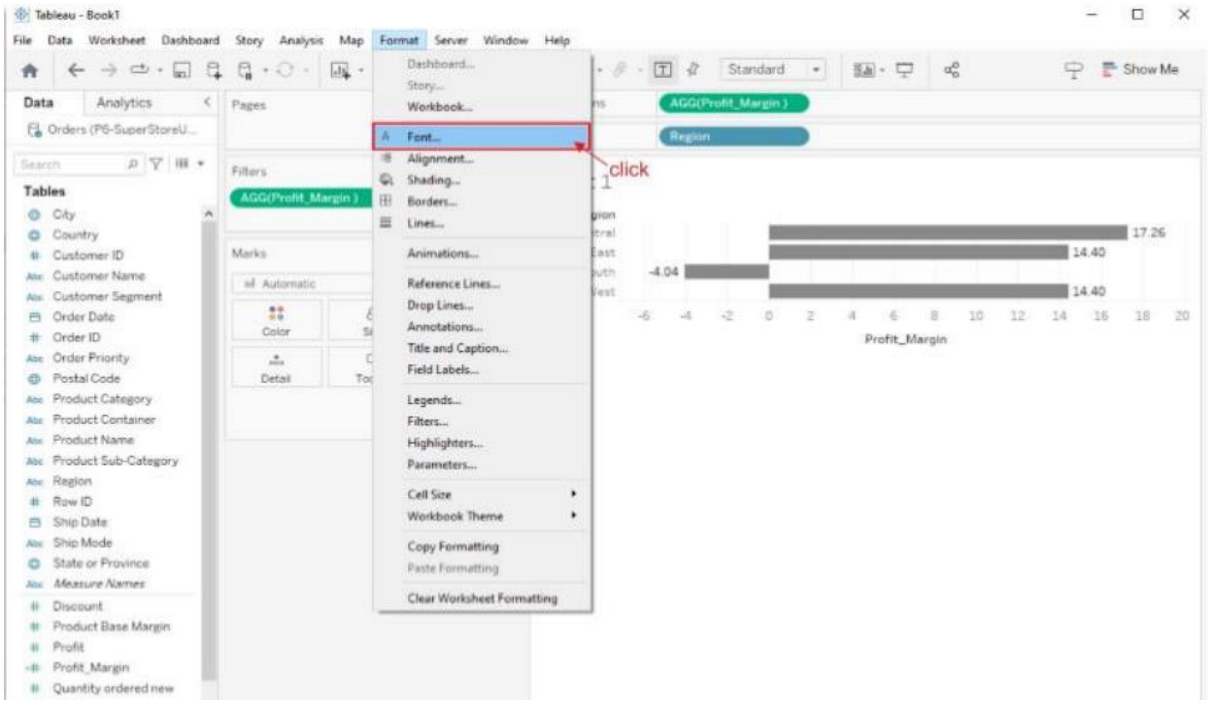

Filter [Profit_Margin]
Range of values
At least
At most
Special
At most
-4.039162020
17.258111345
-4.039162020
17.258111345
Show: Only Relevant Values
Include Null Values
Reset
OK
Cancel
Apply

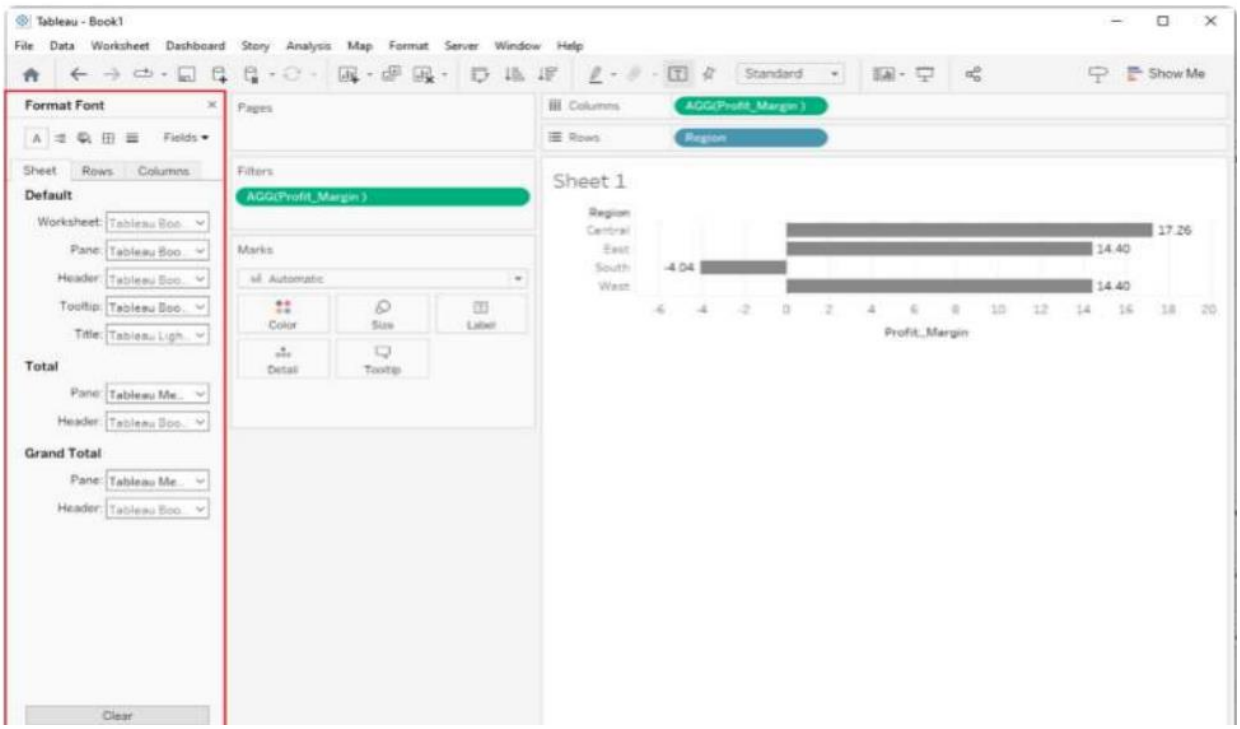

Filter [Profit_Margin]
Range of values
At least
At most
Special
Special
Null values
Non-null values
All values
Reset
OK
Cancel
Apply

Formatting Visualizations

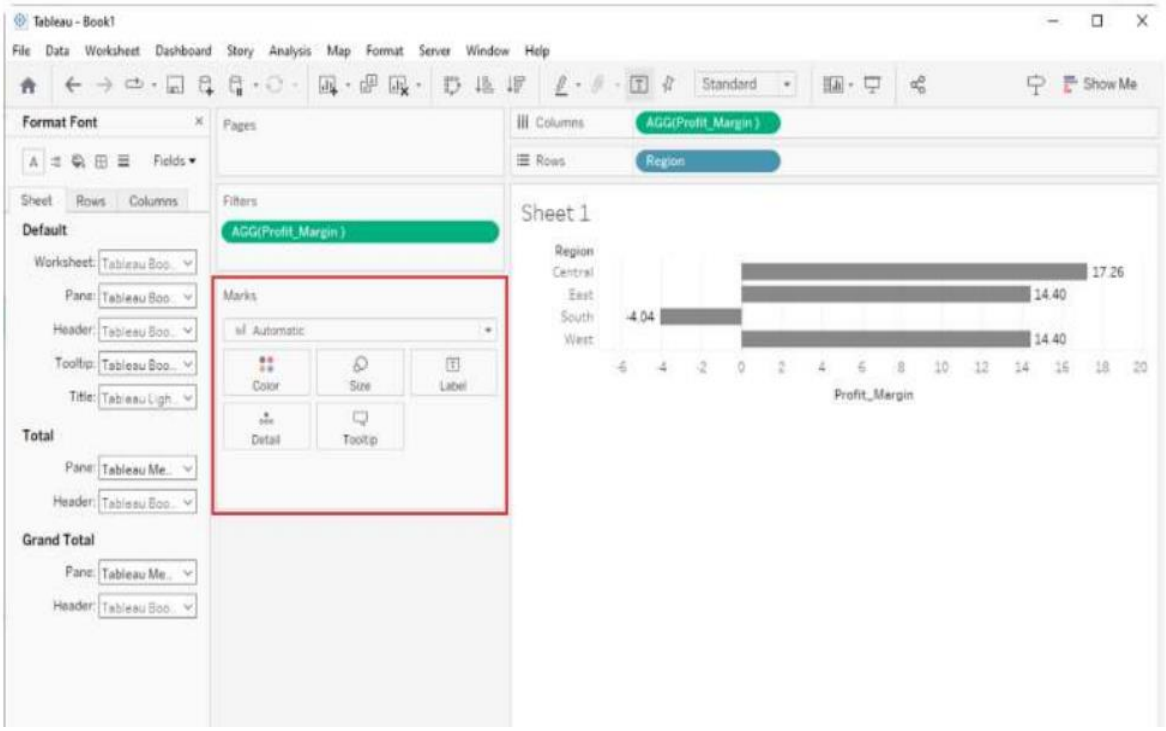
Tableau provides a wide range of formatting options to make your visualizations more appealing and informative:

1. **Format Pane:** On the left side of the Tableau interface, you'll find the Format pane. It allows you to format various aspects of your visualization, such as fonts, colors, lines, shading, and borders. Simply select the element you want to format and use the options in the Format pane to make changes.

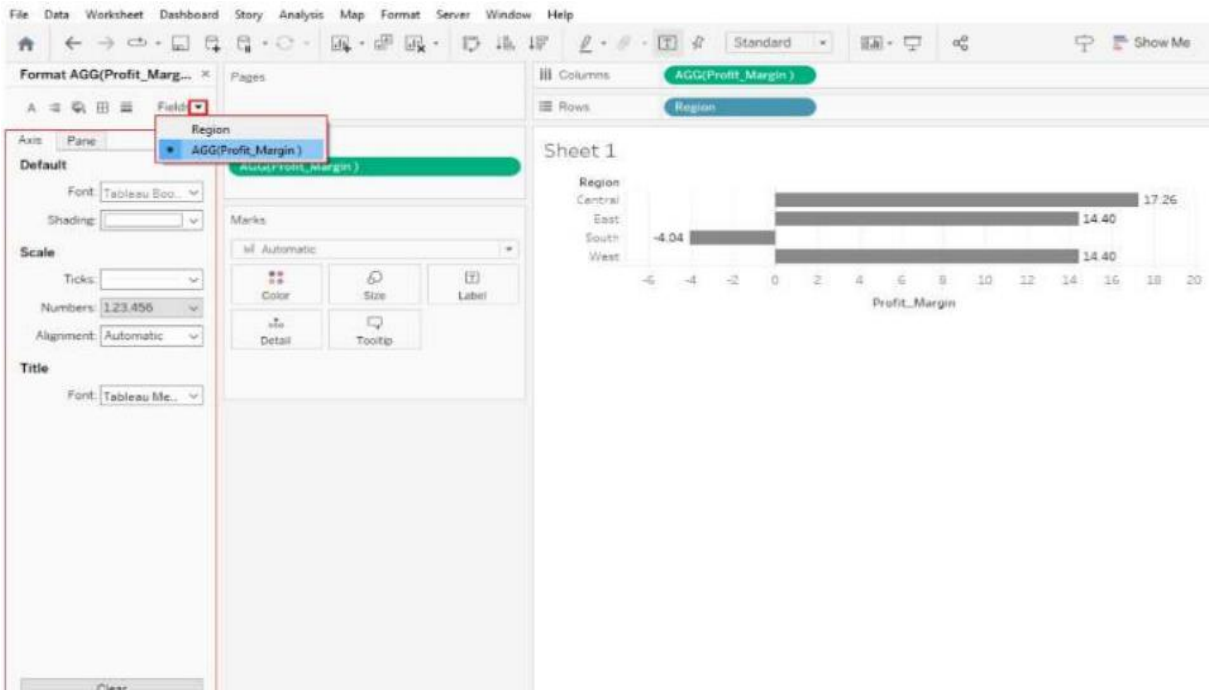




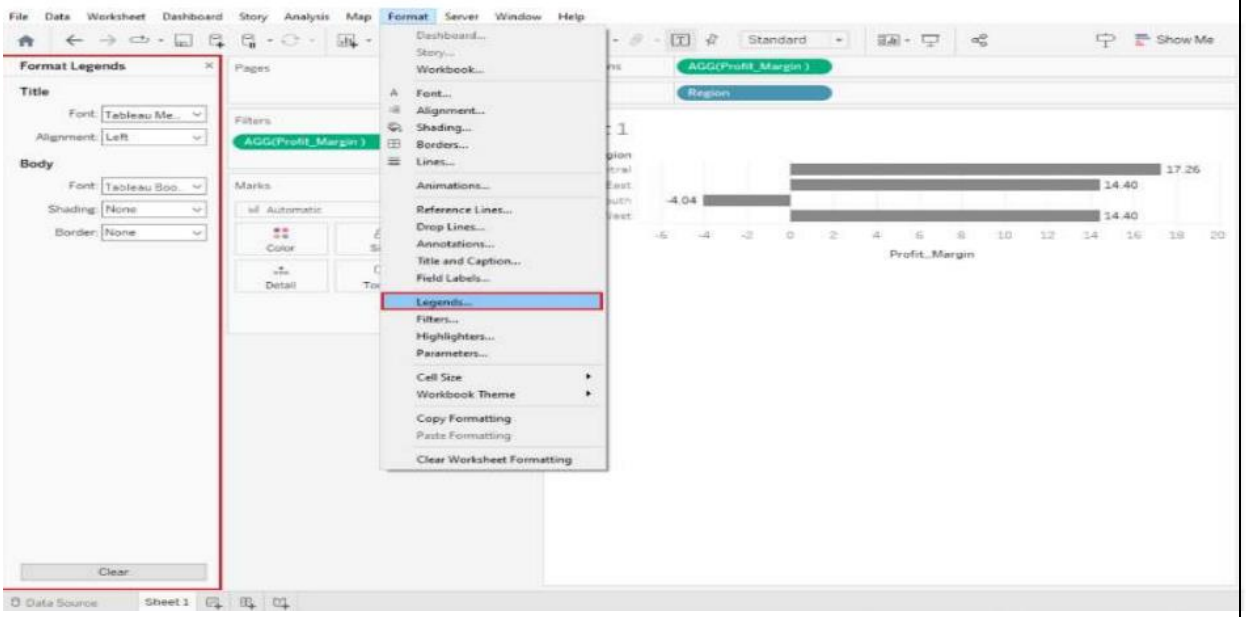
2. **Marks Card:** The Marks card, located above your visualization, offers formatting options specific to the type of marks you're using (e.g., color, size, label). Click on the Marks card to access these options and modify how your data is represented.



3. **Axis and Gridlines:** You can format axis labels, titles, and gridlines to improve the readability of your visualization. Right-click on an axis or gridline to access formatting options.



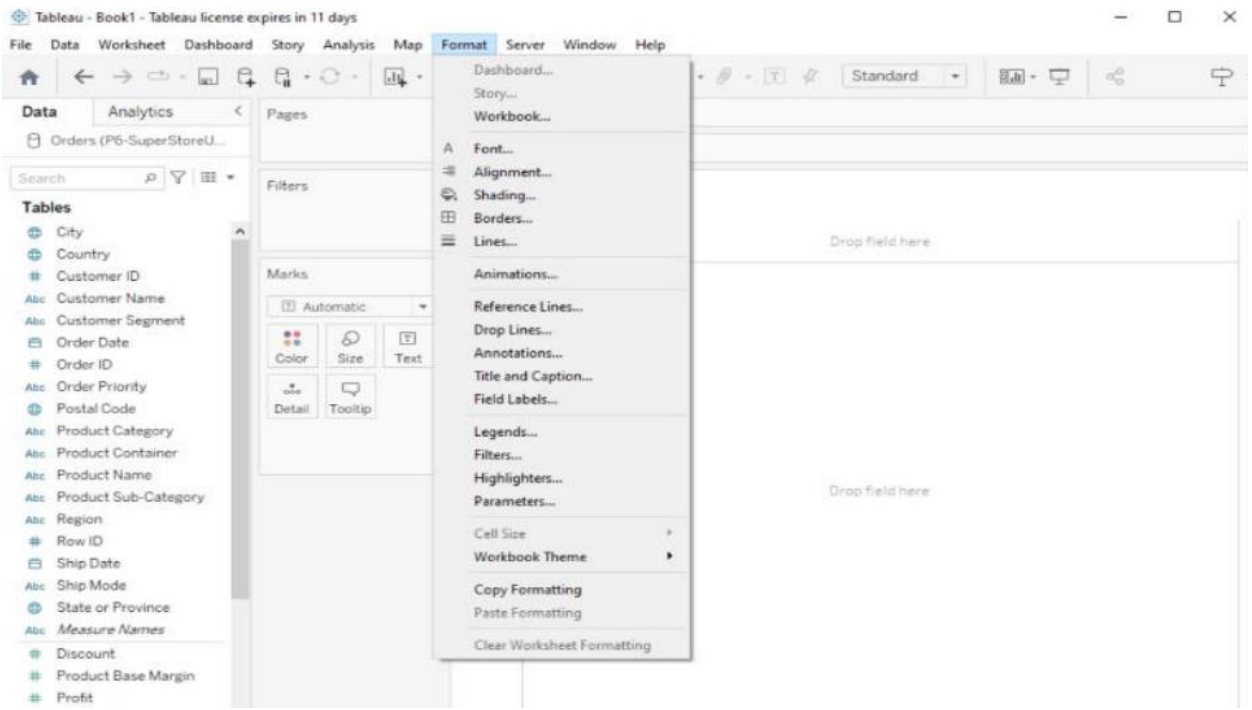
4. **Legends and Color Scales:** Customize legends and color scales to provide context for your visualizations. You can change colors, labels, and the position of legends to match your data.



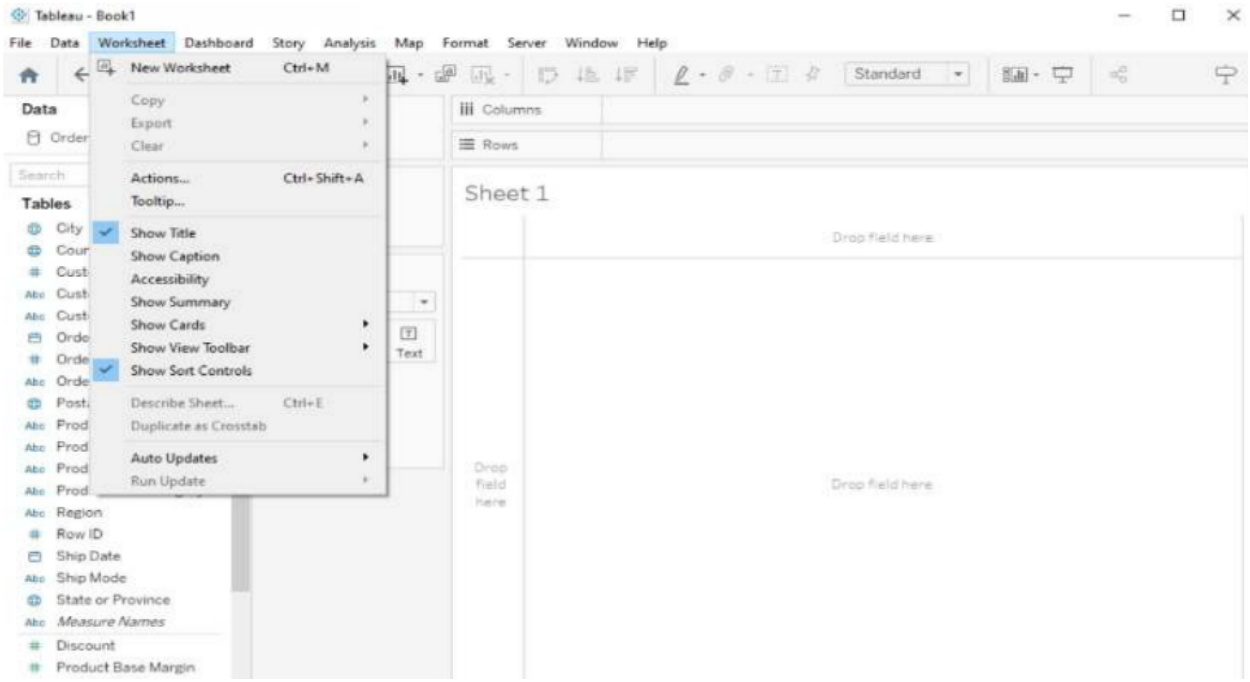
Formatting Tools and Menus:

Tableau provides several formatting tools and menus to help you refine the appearance of your visualizations:

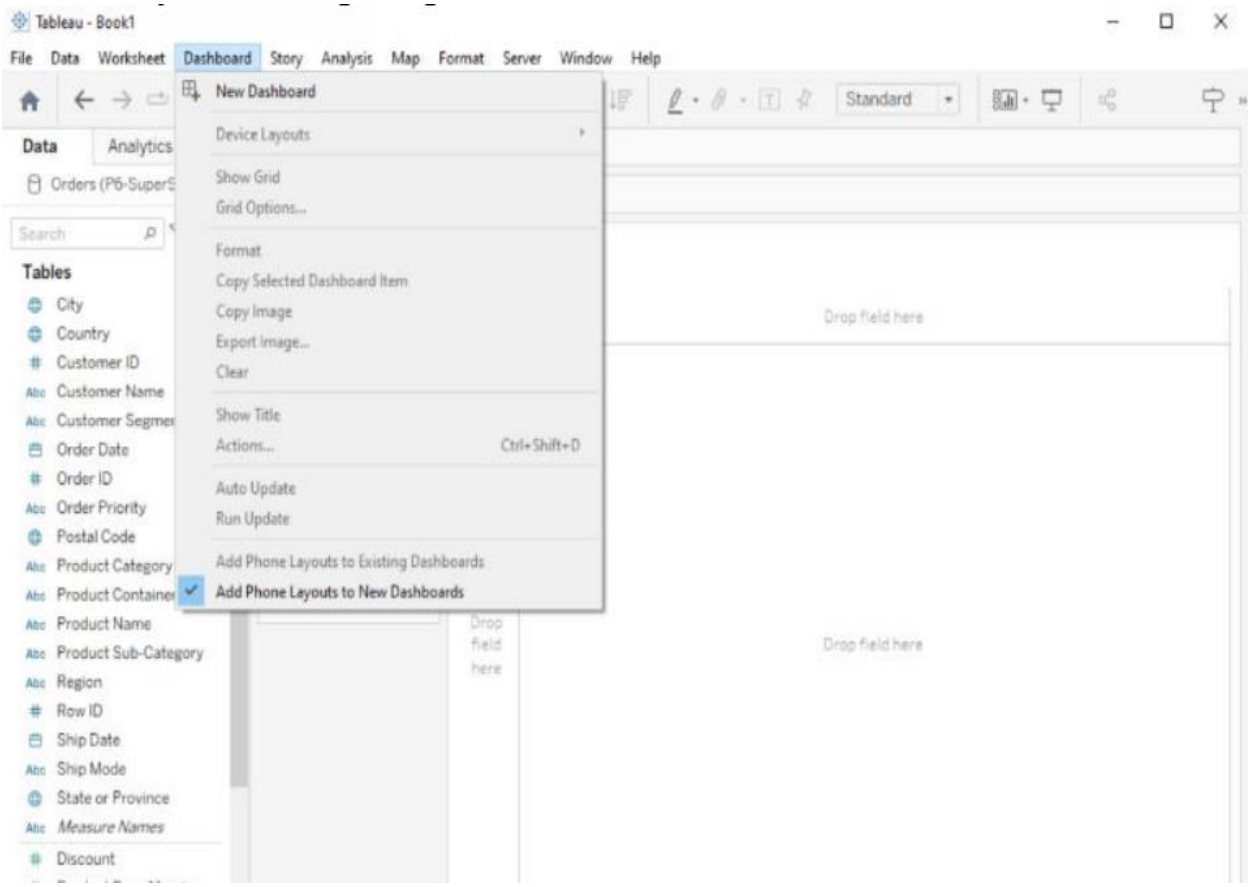
1. **Format Menu:** The Format menu at the top of the Tableau interface provides access to various formatting options, including font styles, shading, borders, alignment, and more. You can use this menu to format text, labels, and other elements.



2. **Worksheet Menu:** In the Worksheet menu, you'll find options to format the entire worksheet, including background color, borders, and worksheet title. You can also adjust the worksheet size.



3. **Dashboard Menu:** If you're working with dashboards, the Dashboard menu allows you to format the entire dashboard layout, including background, size, and title.



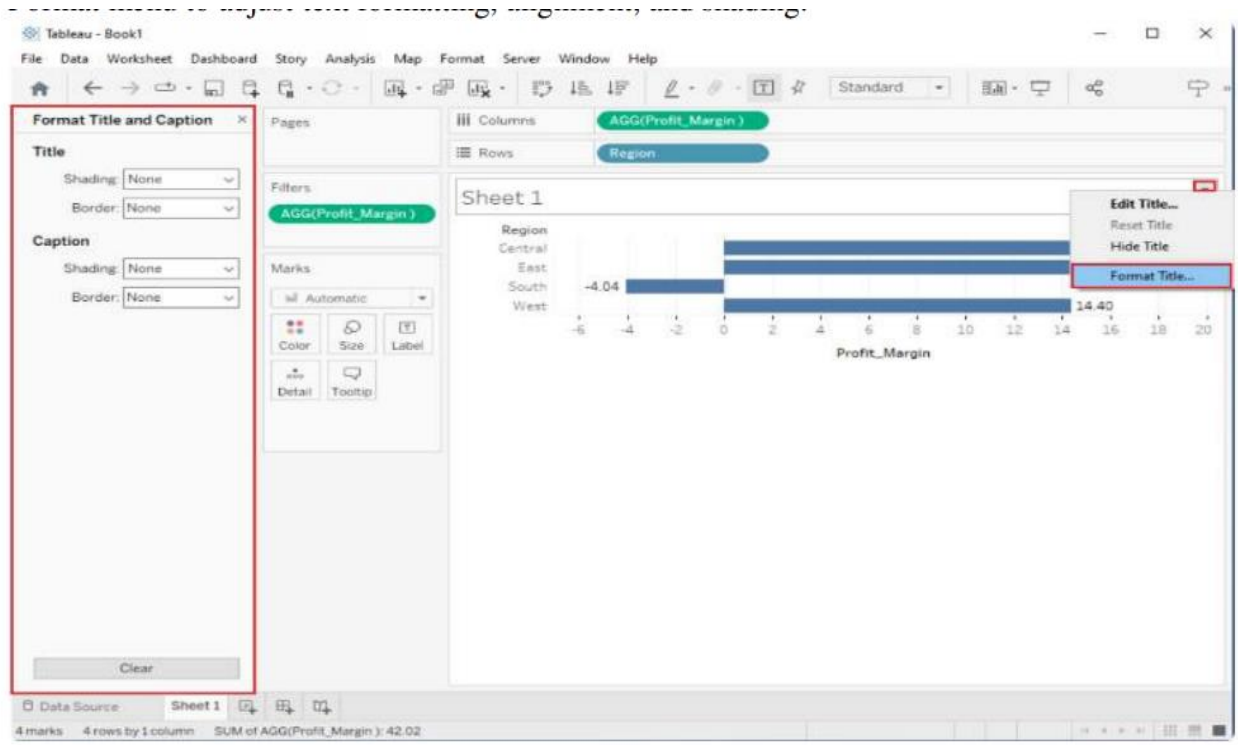
Formatting Specific Parts of the View:-

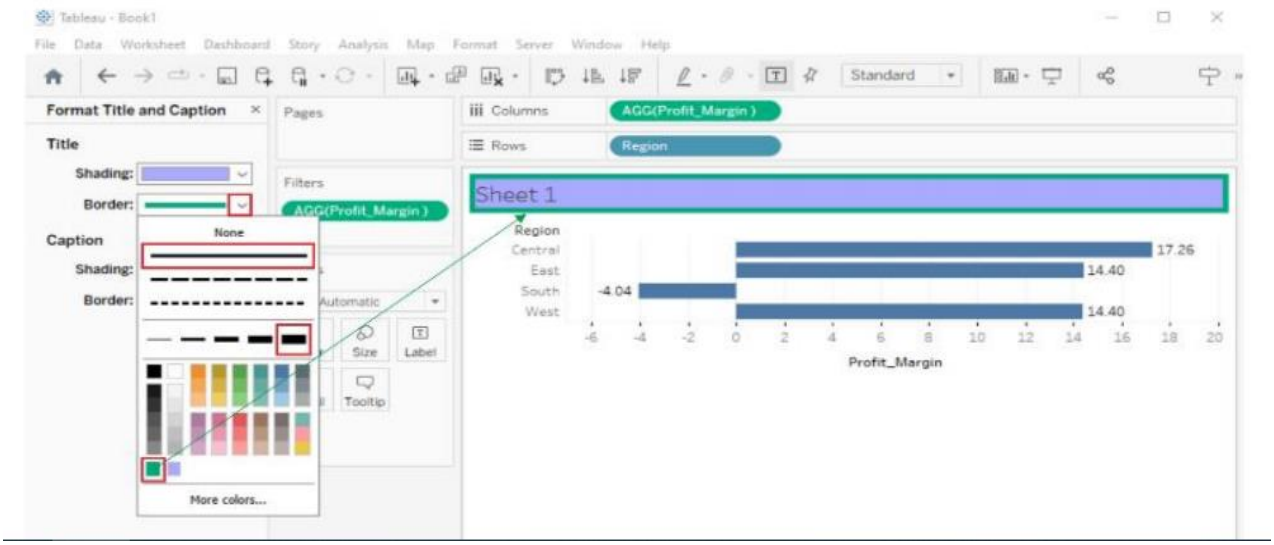
Tableau lets you format specific elements of your visualization:

1. Annotations: You can add annotations to your visualizations to highlight important points or provide additional context. Format these annotations using the options available when you right-click on an annotation.

2. Tooltips: Customize tooltips to display relevant information when users hover over data points. You can format tooltips to show or hide specific fields and control their appearance.

3. Headers and Titles: Format headers, titles, and subtitles for clarity and consistency. Use the Format pane or the Format menu to adjust text formatting, alignment, and shading.

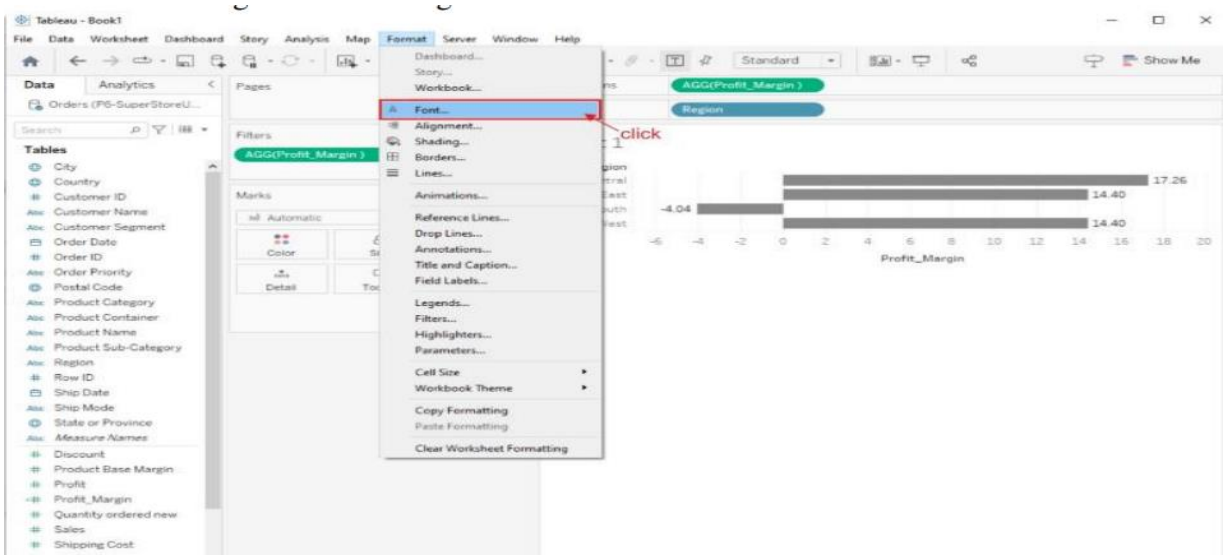




Experiment 5:

Editing and Formatting Axes, Manipulating Data in Tableau data, Pivoting Tableau data.

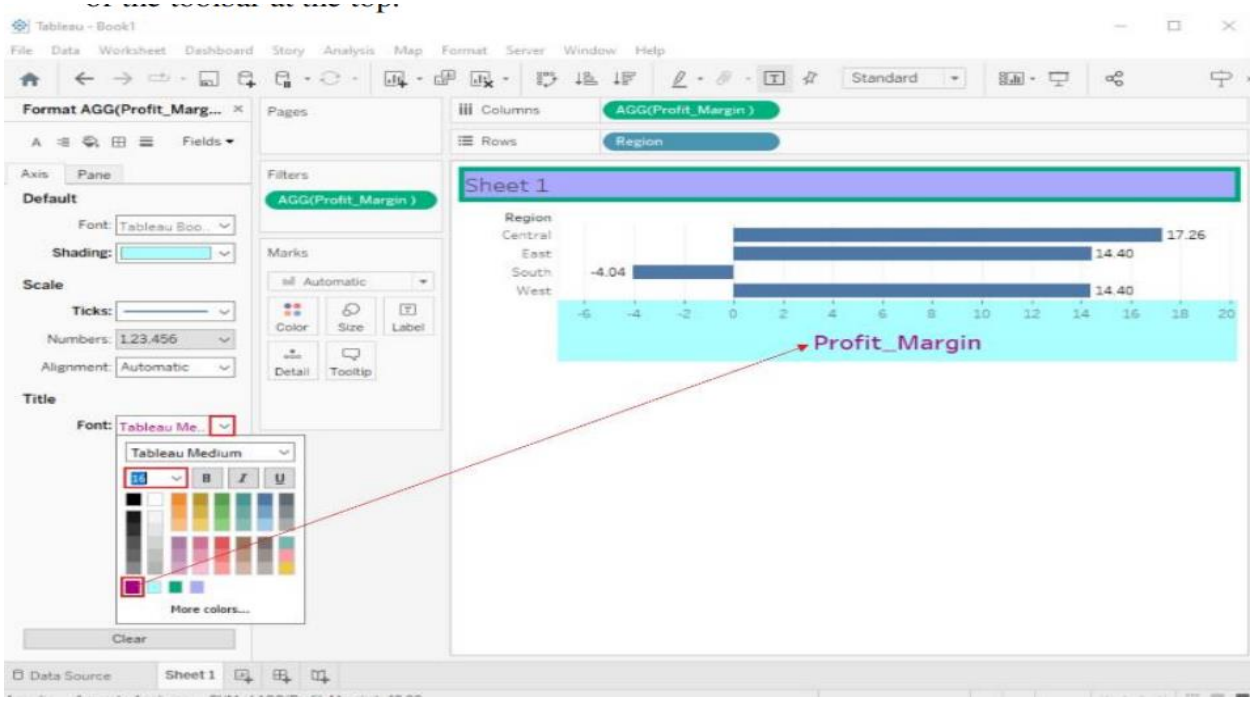
Aim:- Editing and Formatting Axes, Manipulating Data in Tableau data, Pivoting Tableau data.
Solution:- Editing and Formatting Axes:



1. Edit Axis Title:

- Click on the axis title you want to edit.

- You can now modify the title text, font, size, color, and alignment using the Format pane or the toolbar at the top.



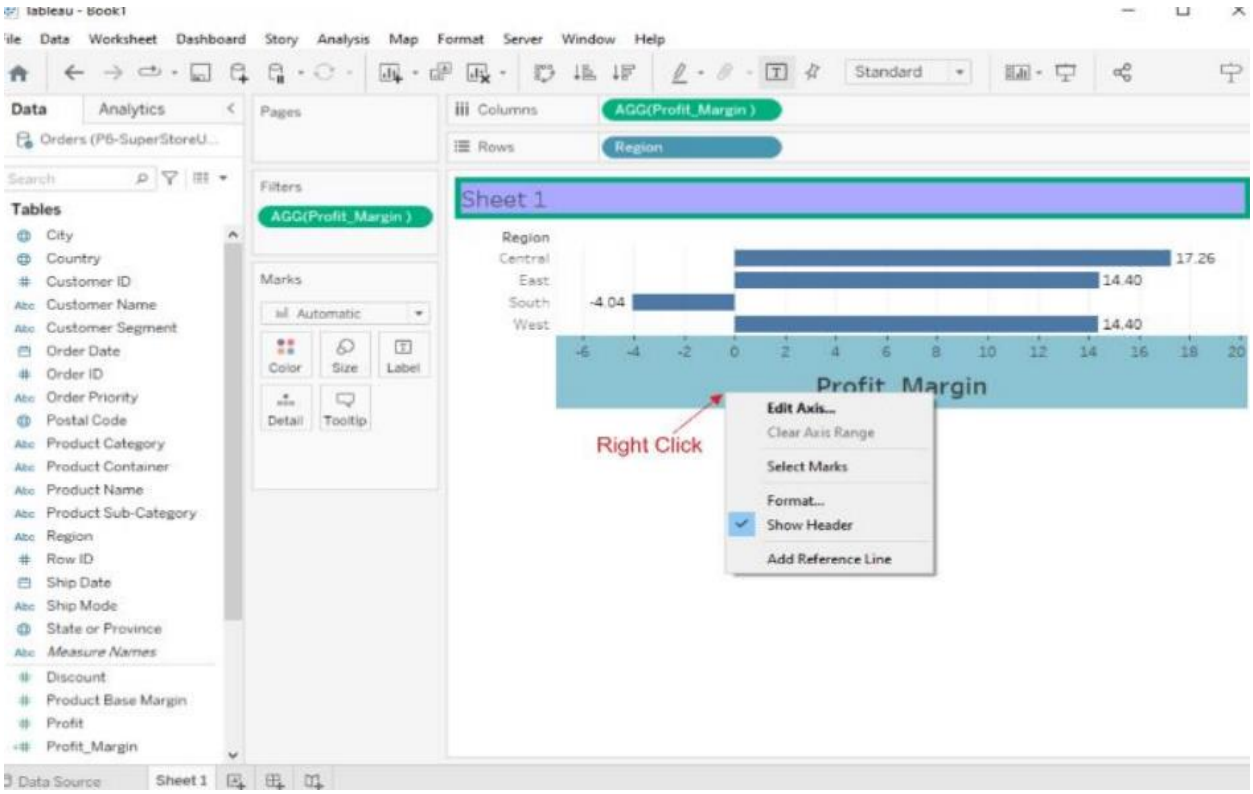
2. **Edit Axis Labels:**

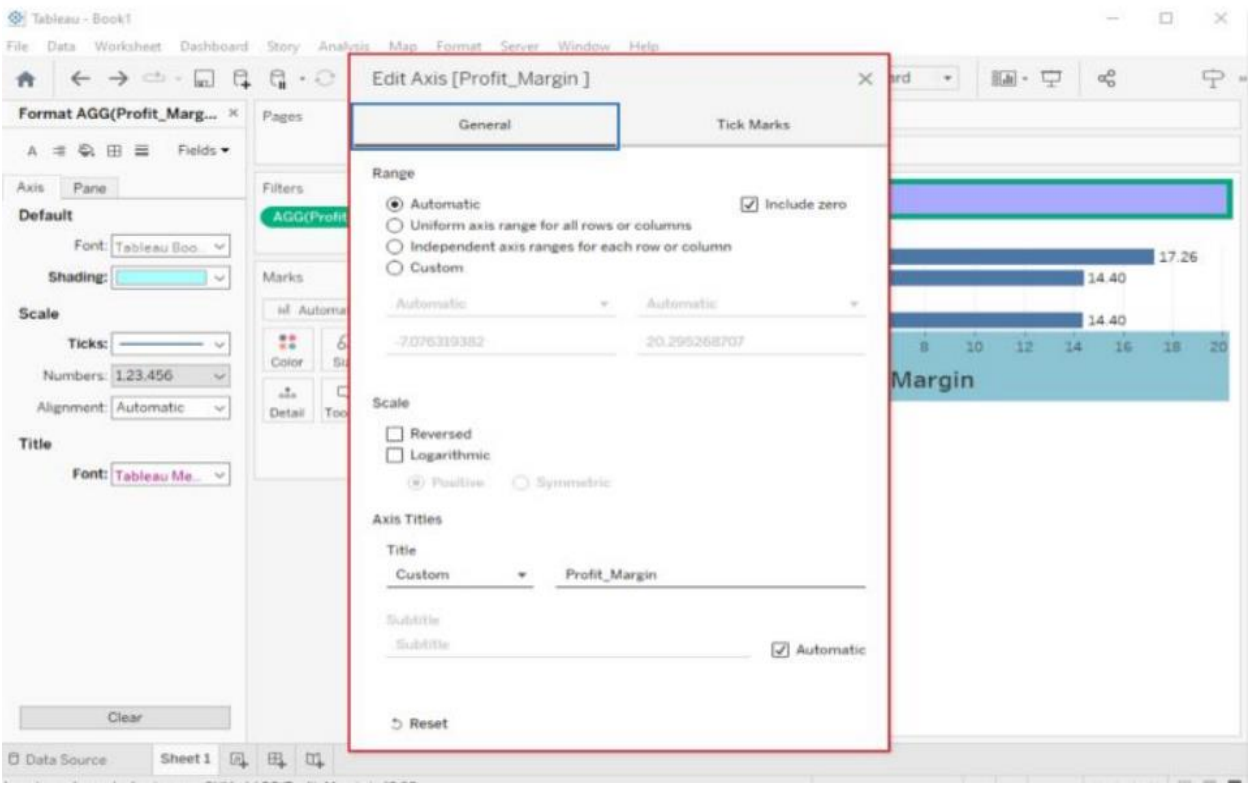
Right-click on an axis and select "Edit Axis."

- In the Edit Axis dialog box, you can change the formatting of labels, tick marks, and other axis-related properties.

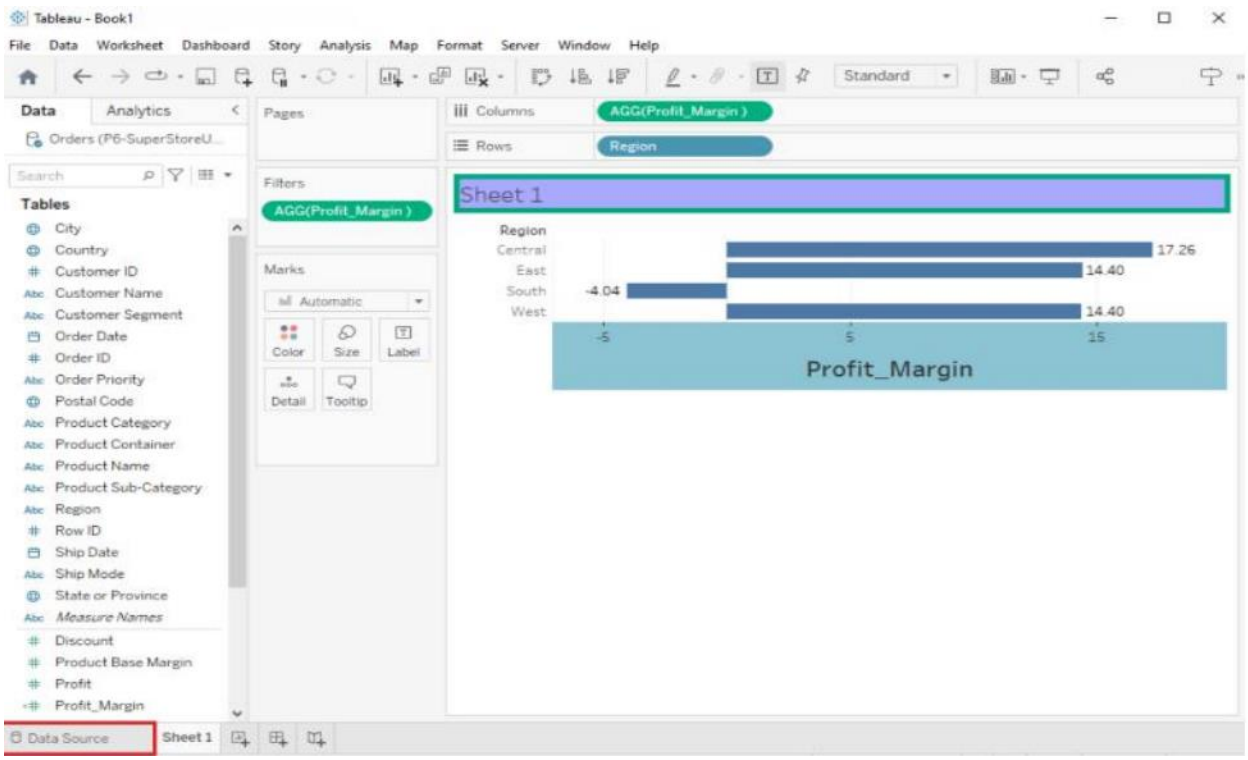
3. Scale and Range:

- To change the scale or range of an axis, right-click on it and select "Edit Axis."
- In the dialog box, adjust the Minimum and Maximum values, scale, or range according to your needs.



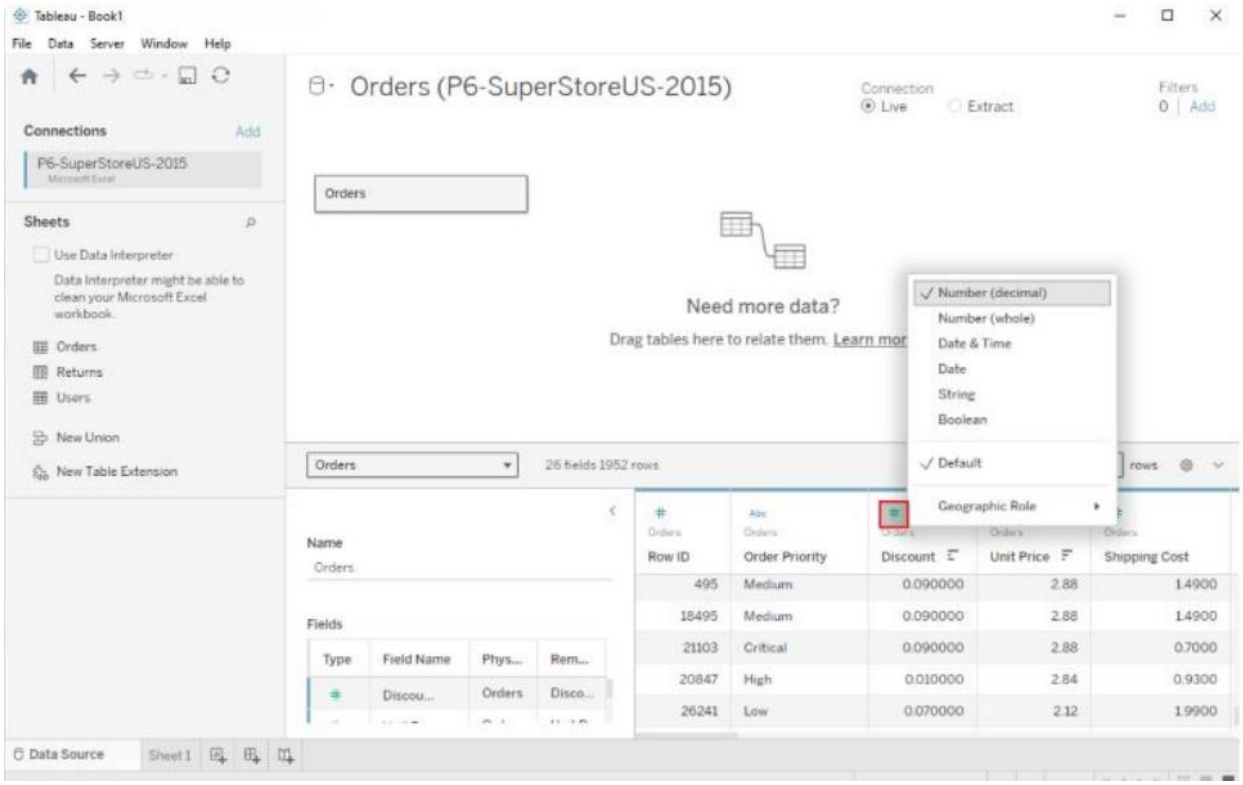


Manipulating Data in Tableau data



Change Data Type:

If Tableau has inferred a wrong data type for a column, the data type can be changed by clicking on the data type symbol in the column header.



New Column(Calculated Fields)

Calculated fields can be used if you need to create customized logic for manipulating certain data types or data values. There are a large-range of functions available in Tableau that can used individually or collectively for data manipulation.

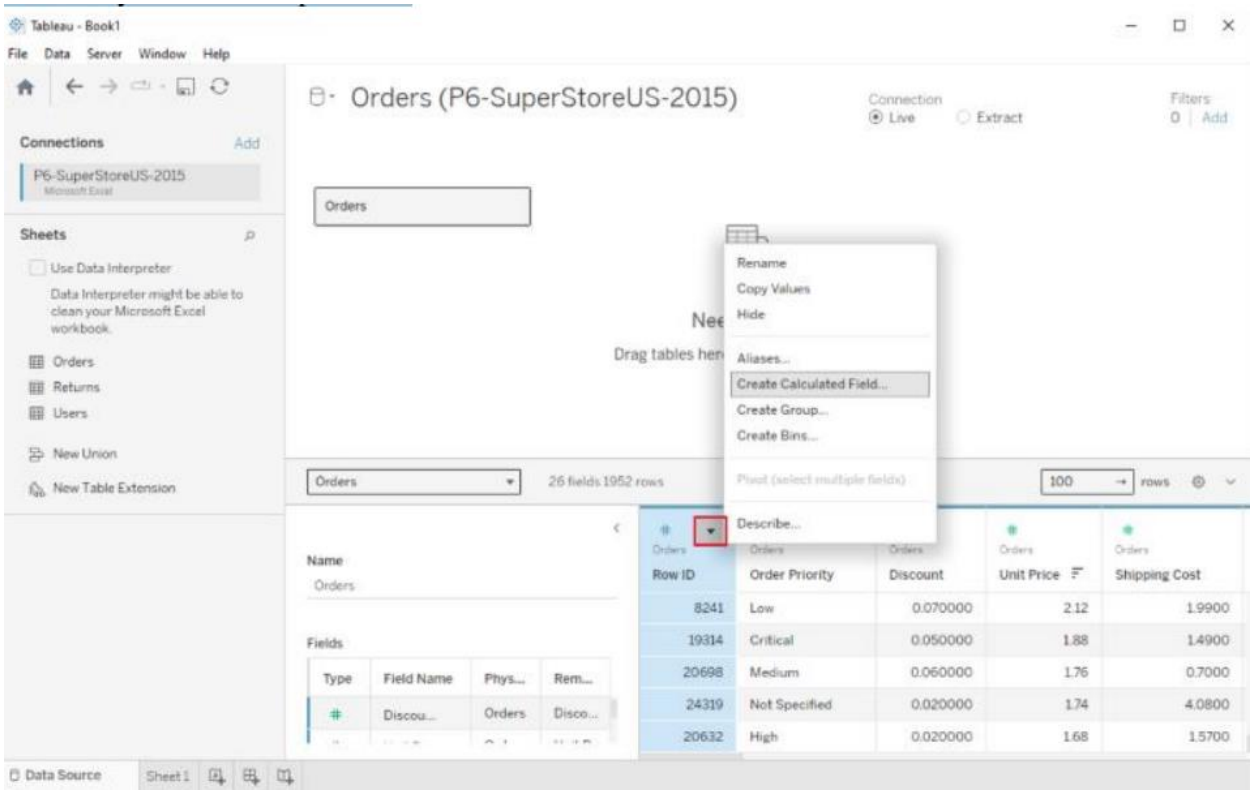
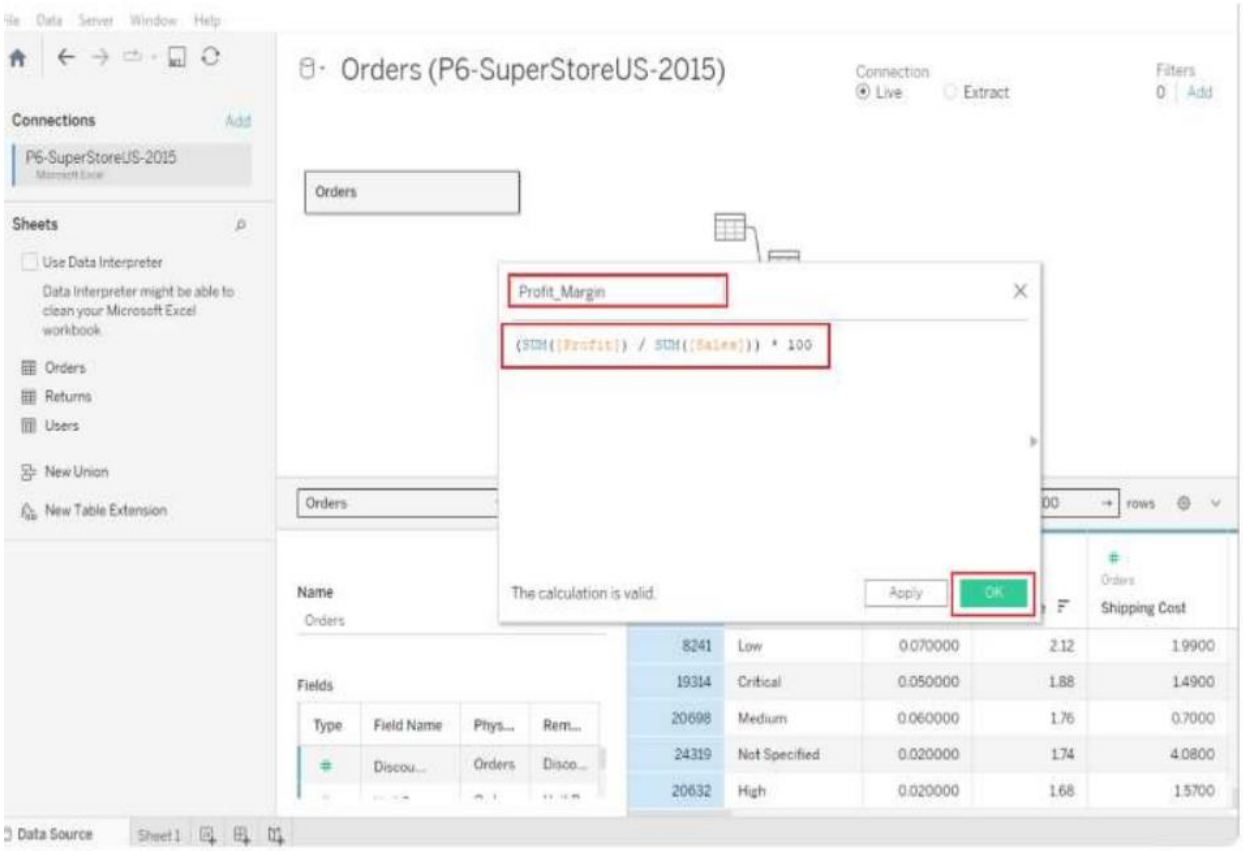
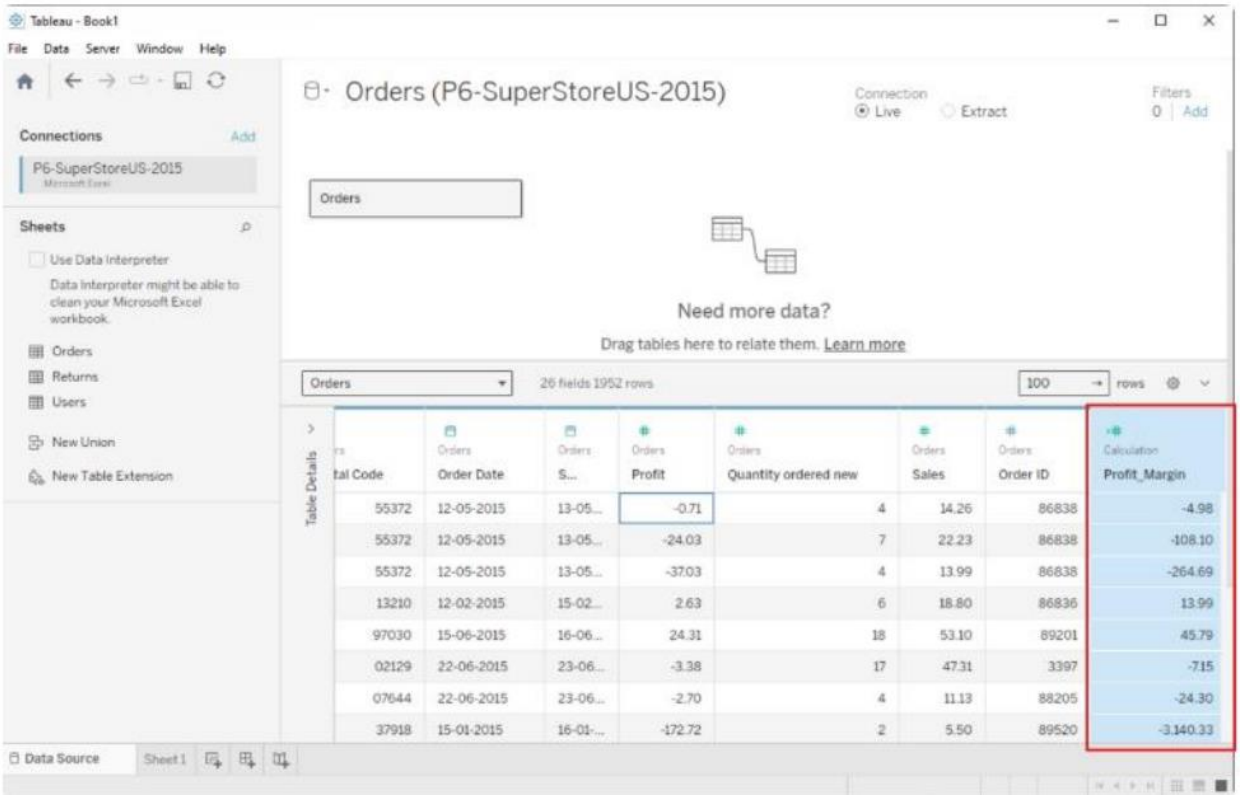


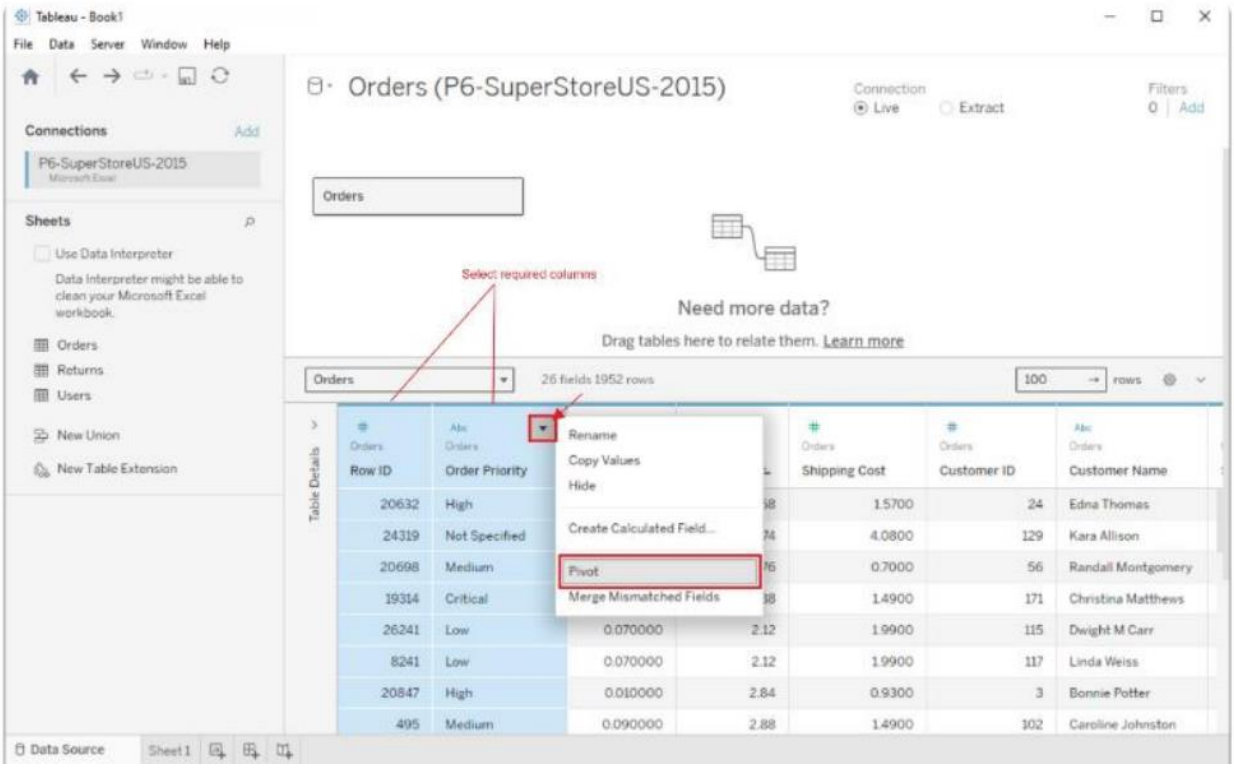
Tableau - Book1
File Data Server Window Help
Orders (P6-SuperStoreUS-2015)
Connection
Live
Extract
Filters
0 Add
Connections
Add
P6-SuperStoreUS-2015
Sheets
Use Data Interpreter
Data Interpreter might be able to clean your Microsoft Excel workbook.
Orders
Returns
Users
New Union
New Table Extension
Orders
Rename
Copy Values
Hide
Aliases...
Create Calculated Field...
Create Group...
Create Bins...
Pivot (select multiple fields)
Describe...
26 fields 1952 rows
100 rows
Name
Orders
Fields
Type
Field Name
Phys...
Rem...
Row ID
Order Priority
Discount
Unit Price
Shipping Cost
8241 Low 0.070000 2.12 1.9900
19314 Critical 0.050000 1.88 1.4900
20698 Medium 0.060000 1.76 0.7000
24319 Not Specified 0.020000 1.74 4.0800
20632 High 0.020000 1.68 1.5700
Data Source
Sheet 1


Orders (P6-SuperStoreUS-2015)
Connection
Live
Extract
Filters
0 Add
Connections
Add
P6-SuperStoreUS-2015
Sheets
Use Data Interpreter
Data Interpreter might be able to clean your Microsoft Excel workbook.
Orders
Returns
Users
New Union
New Table Extension
Profit_Margin
(SUM([Profit]) / SUM([Sales])) * 100
The calculation is valid.
Apply
OK
Name
Orders
Fields
Type
Field Name
Phys...
Rem...
Shipping Cost
8241 Low 0.070000 2.12 1.9900
19314 Critical 0.050000 1.88 1.4900
20698 Medium 0.060000 1.76 0.7000
24319 Not Specified 0.020000 1.74 4.0800
20632 High 0.020000 1.68 1.5700
Data Source
Sheet 1



Pivoting Tableau data

Data pivoting enables you to rearrange the columns and rows in a report so you can view data from different perspectives.



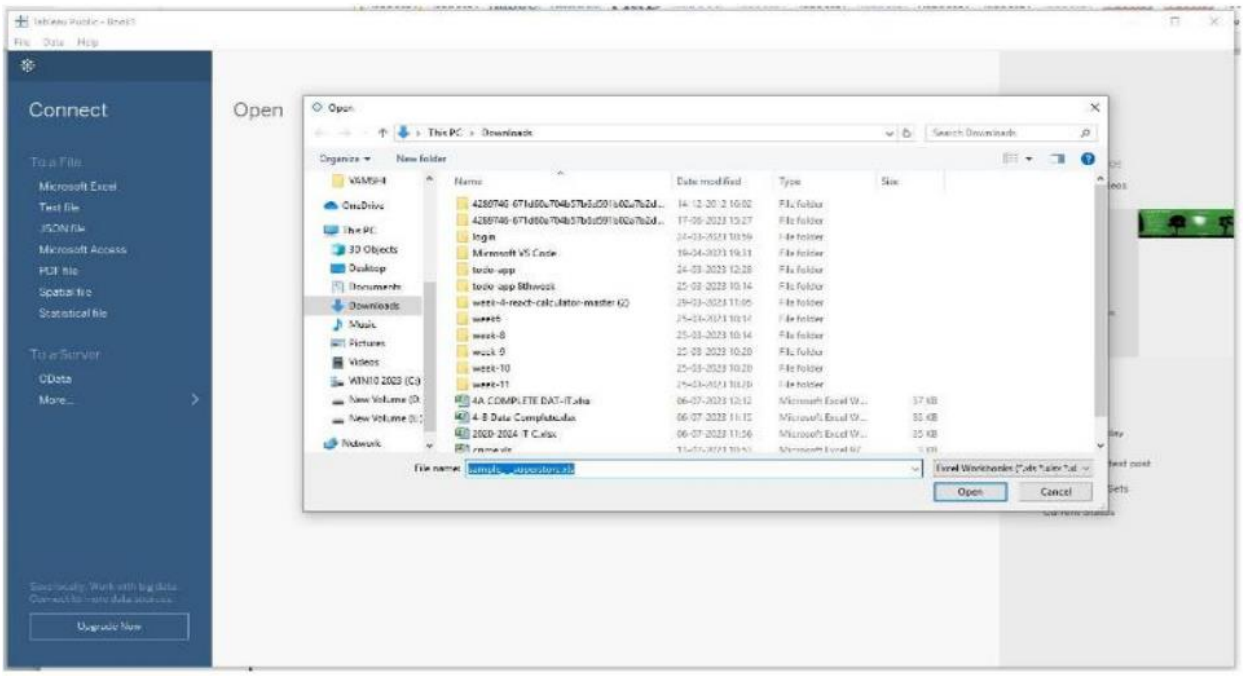
Experiment 6:

Structuring your data, Sorting and filtering Tableau data, Pivoting Tableau data.

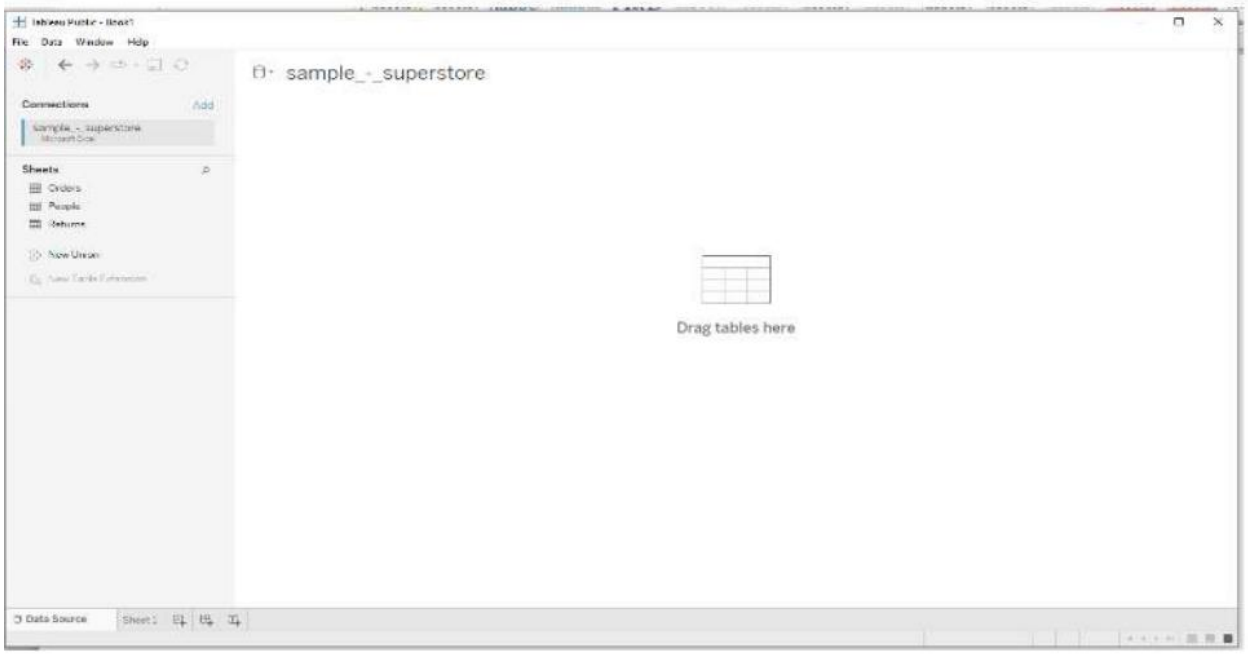
Aim:- Structuring your data, Sorting and filtering Tableau data, Pivoting Tableau data.
Solution:- Creating Sorting and filtering Tableau data, Pivoting Tableau data:

Tableau supports connecting to a wide variety of data, stored in a variety of places. For example, data might be stored on computer in a spread sheet or a text file, or in a big data, relational, or cube (multidimensional) database on a server in enterprise or the data can be from a public domain available on the web. Data can be imported in Tableau Public from Connect panel on left side.

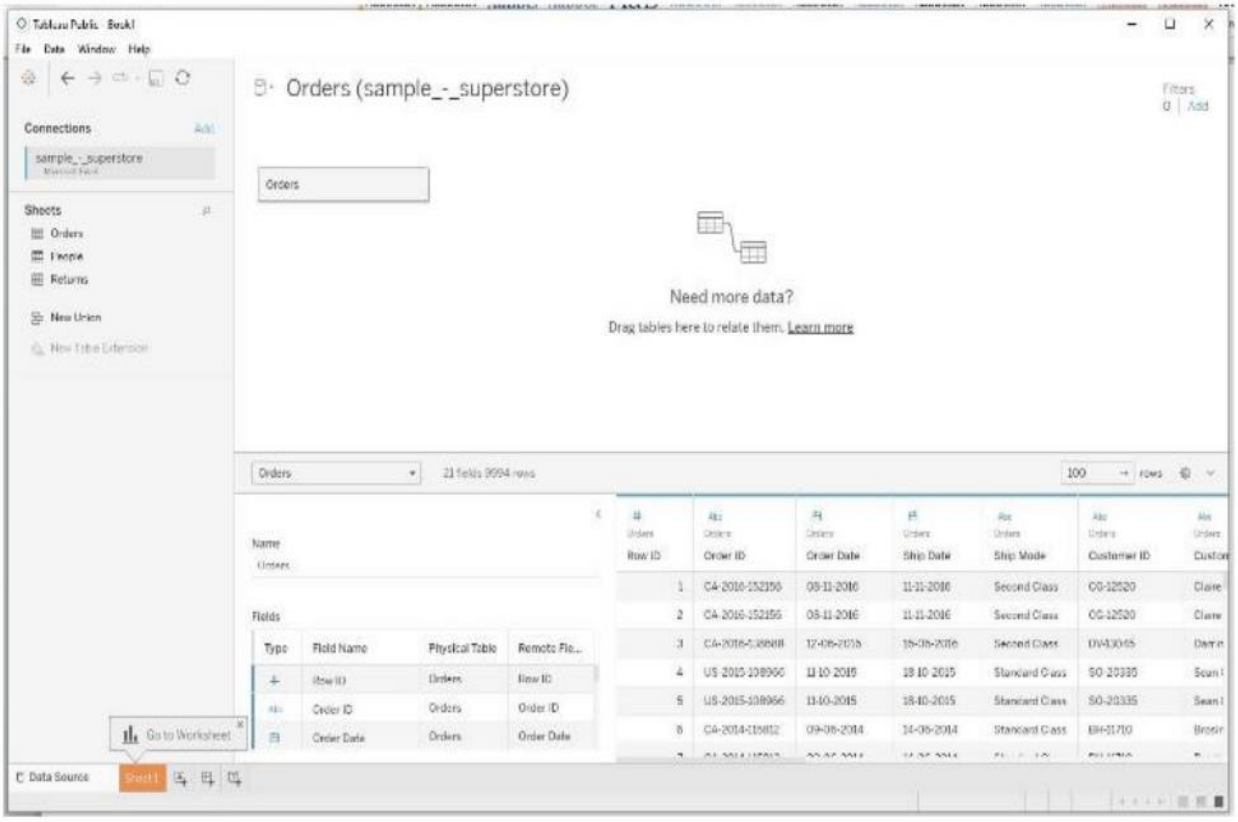
For example, an Excel sample data set was loaded into Tableau as follows:



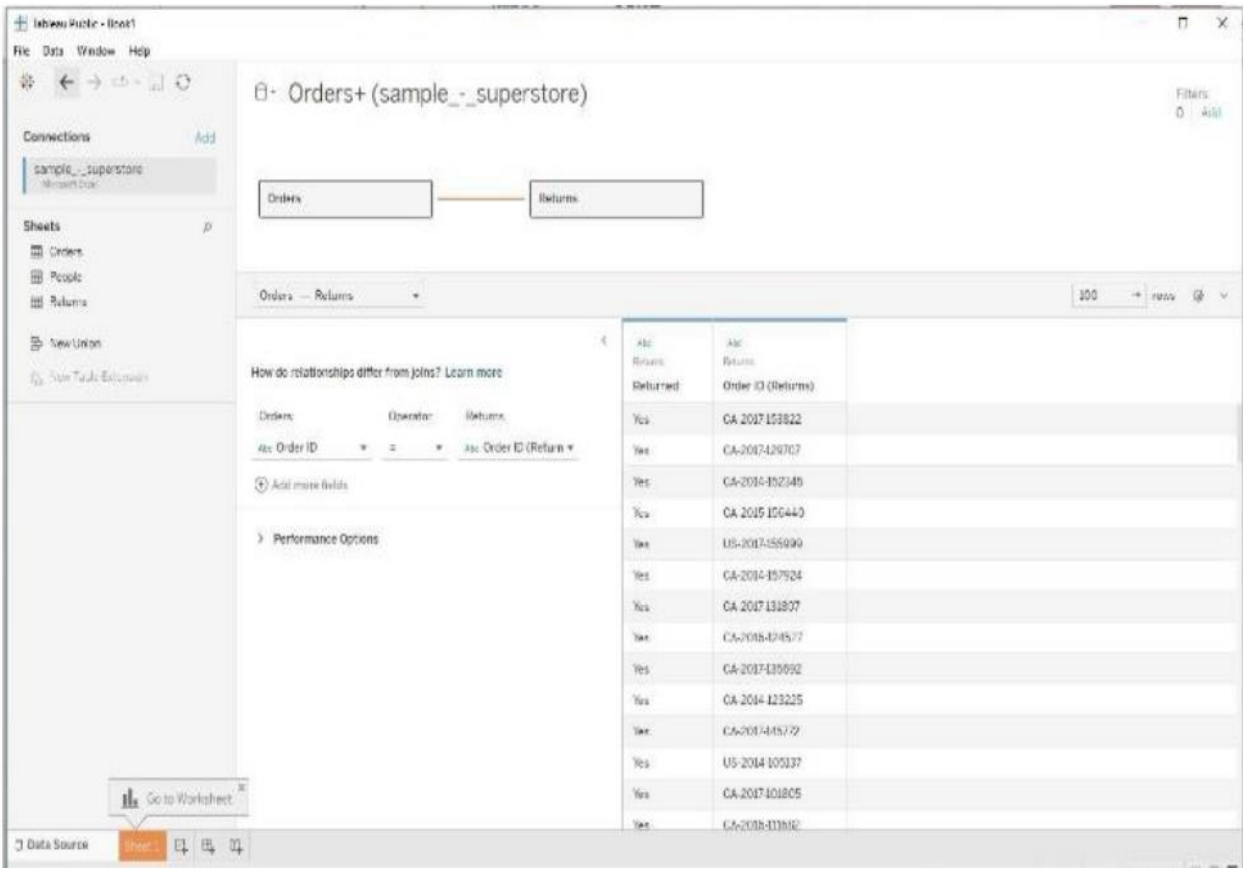
After clicking on open, screen is as follows:



The data store page appears as above. The left pan shows that above dataset consists of 3 worksheets. If we drag orders table, screen appears as follows: Tableau automatically identifies the data type of each column.



Now drag Returns table onto the Canvas to the right of Orders table. This shows the relation between the two tables Orders and Returns.



If we click on the link between Orders and Returns table names at the top gives the summary of the relationship between the tables. Now rename the data store and click on Sheet1 at the bottom left to proceed. This step creates a data extract which improves query performance.

Experiment 7:

Creating Dashboards & Storytelling, creating your first dashboard and Story, and Design for different displays, and publish your visualization.

Aim: Creating your first dashboard and Story, and Design for different displays, and publish your visualization**.**

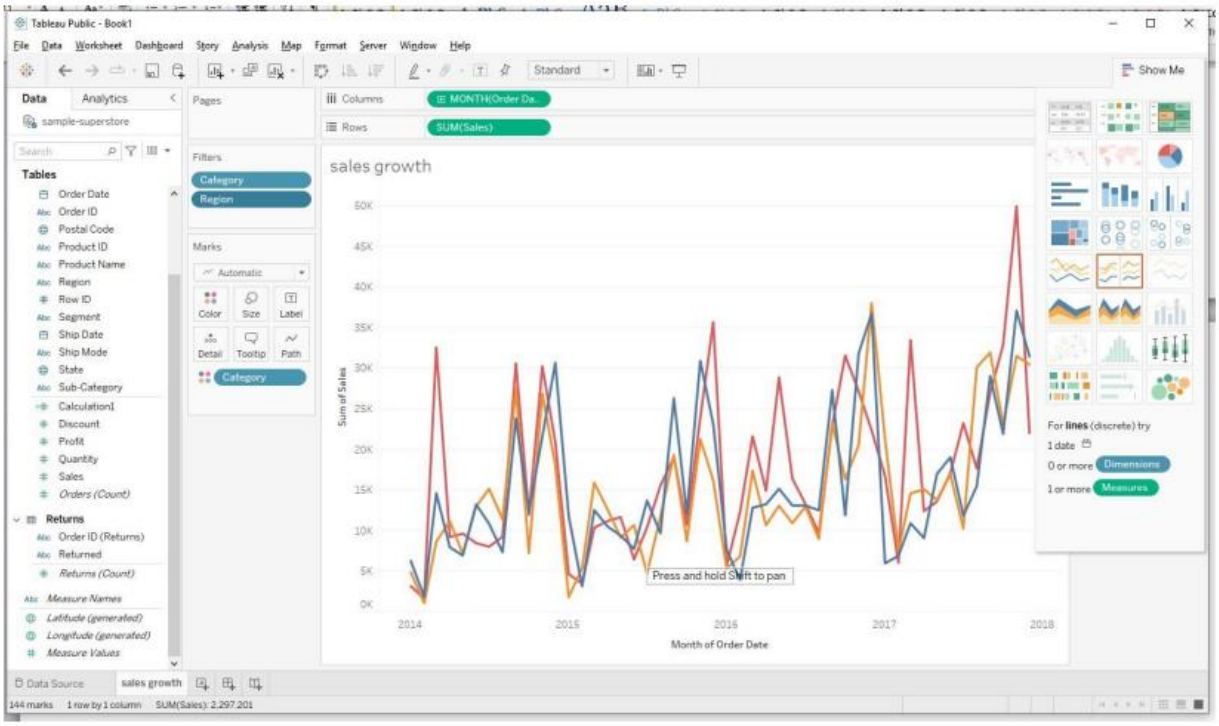
Solution:

A dashboard is a way of displaying various types of visual data in one place. Usually, a dashboard is intended to convey different, but related information in an easy-to-digest form. And oftentimes, this includes things like key performance indicators (KPI)s or other important business metrics that stakeholders need to see and understand at a glance.

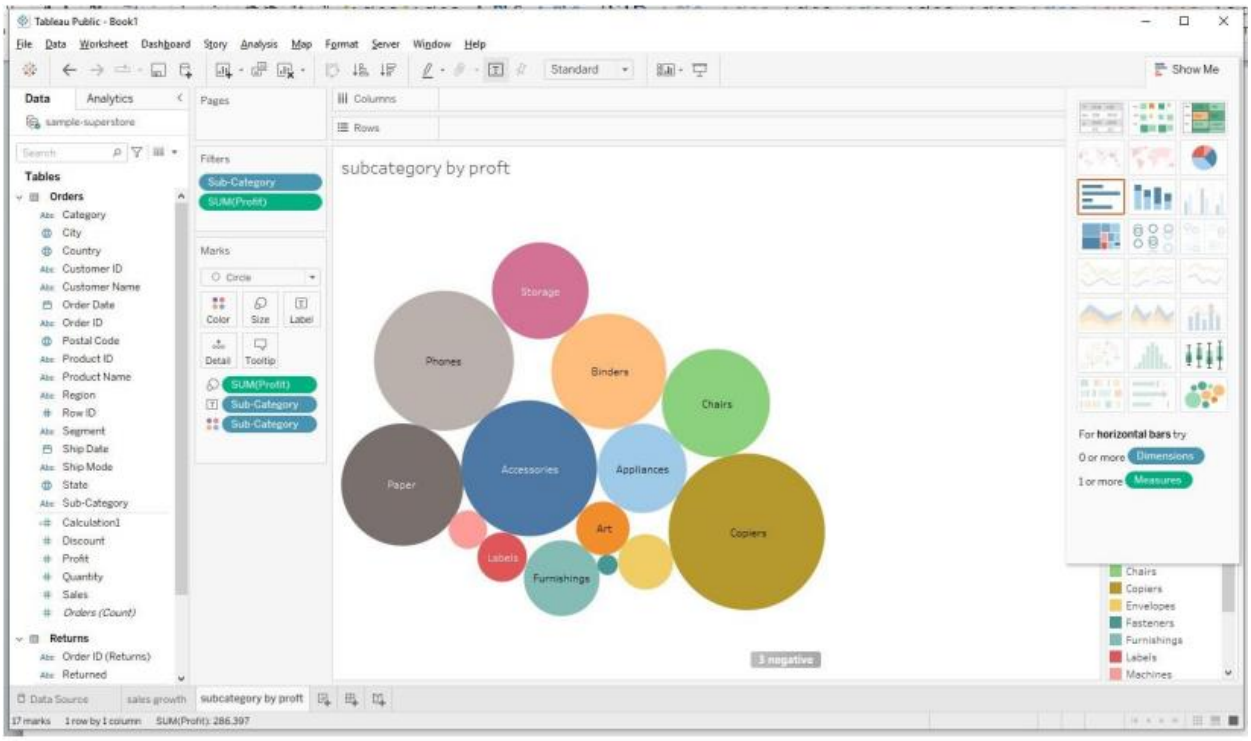
Dashboards are useful across different industries and verticals because they're highly customizable. They can include data of all sorts with varying date ranges to help you understand: what happened, why it happened, what may happen, and what action should be taken.

For example, category of sales across months in a year, region is the field added.

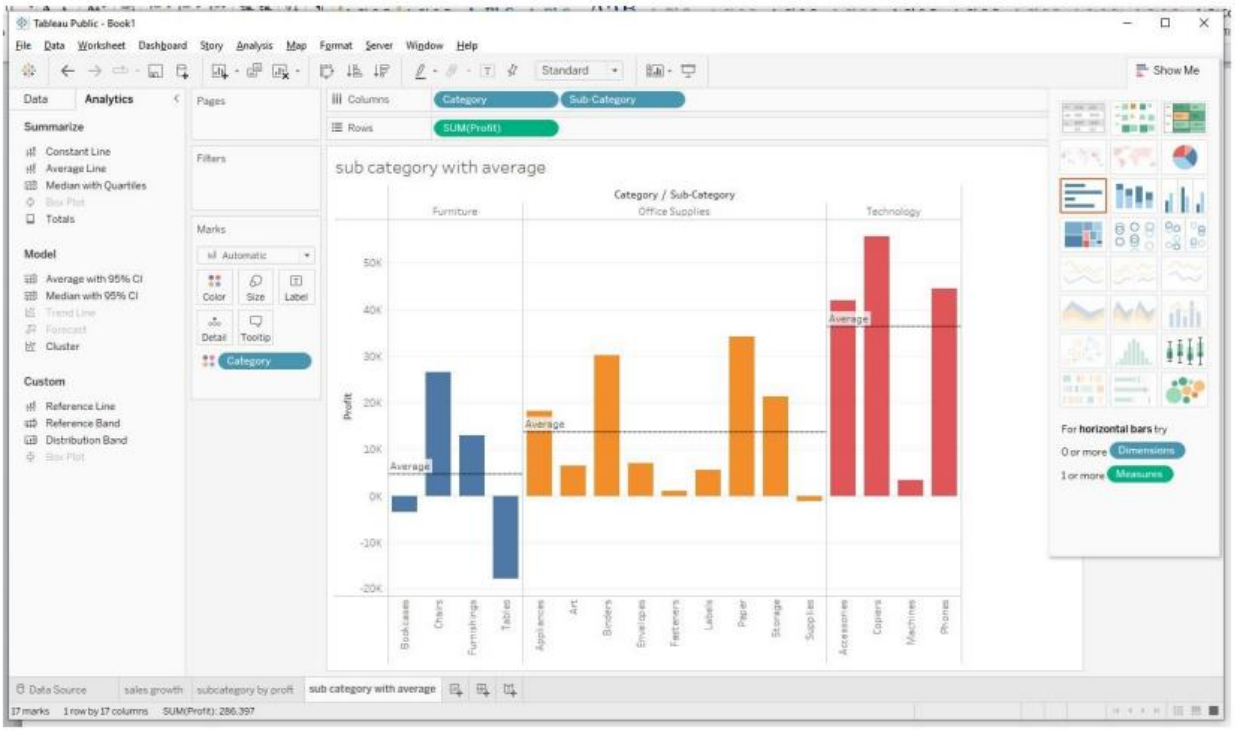
The **first view** is shown below. This can be renamed at the bottom of the screen.



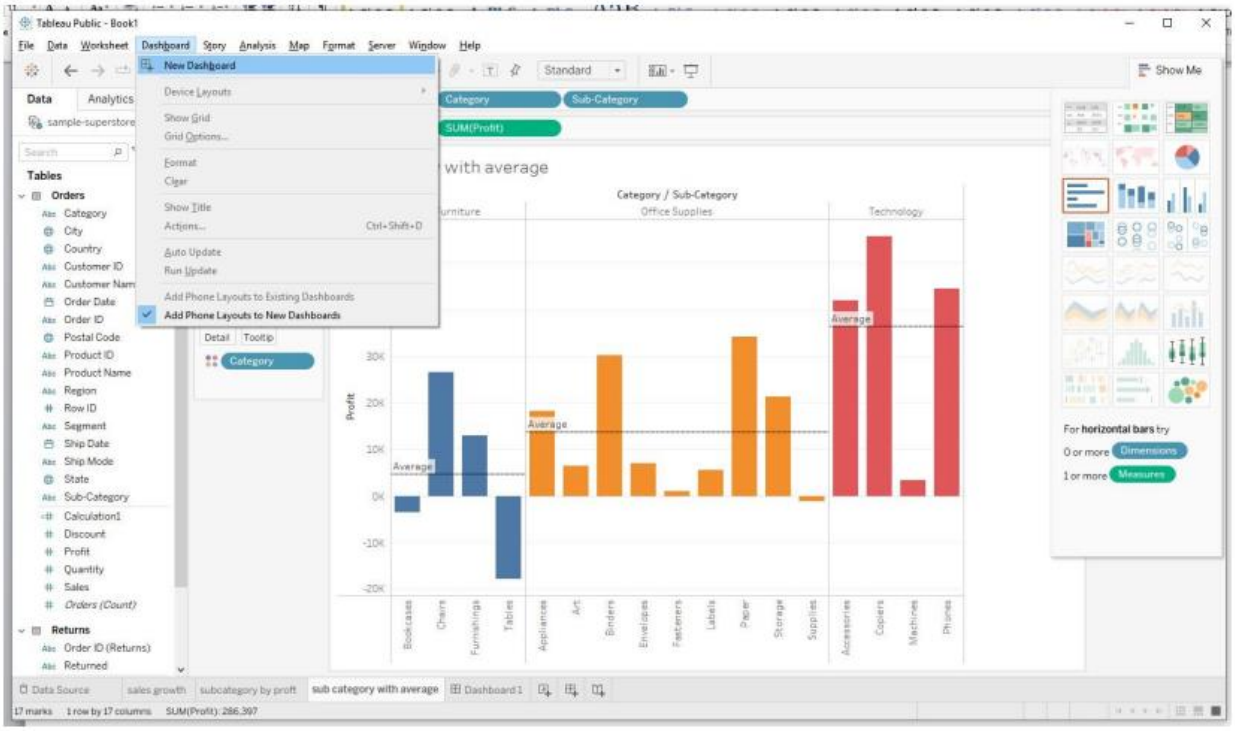
Now go to 2nd sheet for creating **second view**. The second view is shown below. A bubble chart was drawn between profit and subcategory. Then rename the sheet



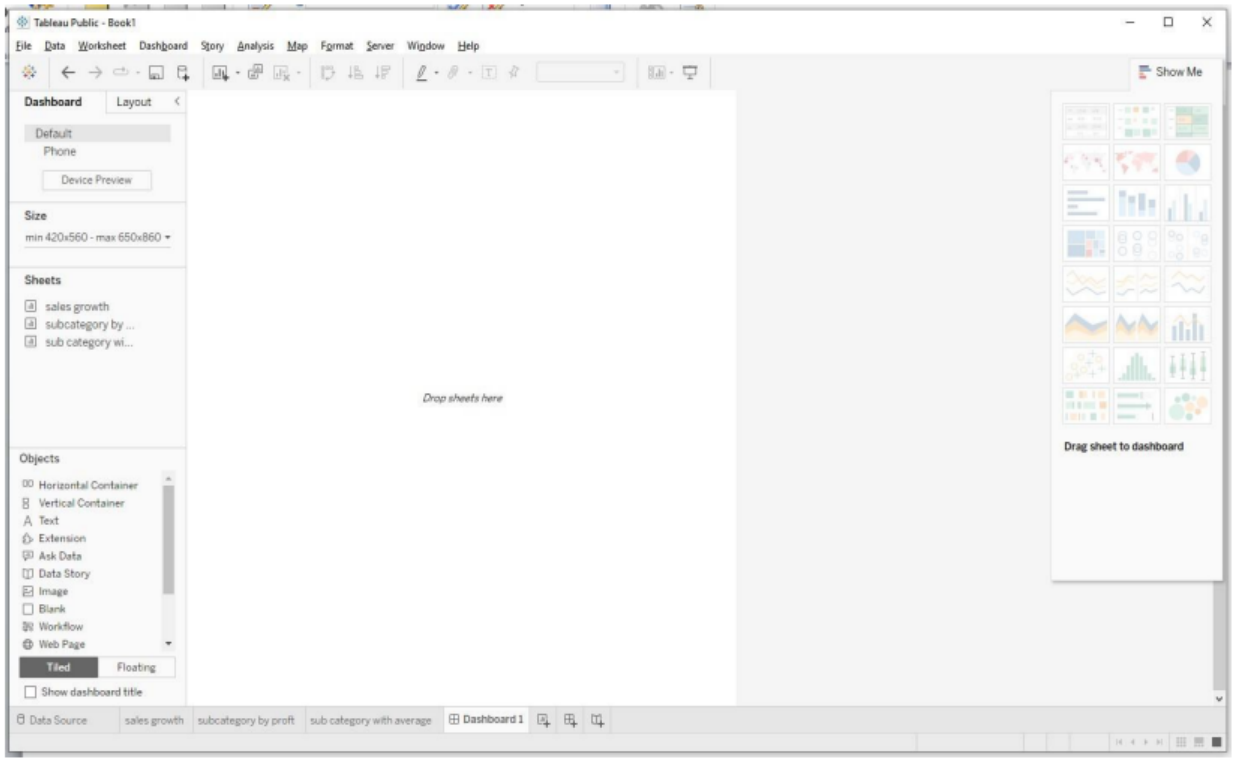
Next **third** view is created as follows for profit for each subcategory in the category with averages.



After creating individual views, now a **Dashboard** can be created by clicking on create dashboard at the toolbar.



after clicking on new dashboard option, the screen is shown below



Now the sheets or views which are created earlier can be drag and dropped on this dashboard. The above three created views are placed in the dashboard as follows.

One can follow their own way of importing sheets on the dashboard.

After creating dahsboard, title can be given to the dashboard from Dashboard tab.

Dahsboard can be customized in terms of its appearance by the user if required. Dashboard once created can be saved on user's system and can be retrieved whenever required