**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY FACULTY OF SCIENCE AND HUMANITIES DEPARTMENT OF COMPUTER APPLICATIONS**

****

**PRACTICAL RECORD NOTE**

**STUDENT NAME :**

**REGISTER NUMBER :**

**CLASS : MSc ADS SEC : C**

**YEAR & SEMESTER : I YEAR & I SEM**

**SUBJECT CODE :**

**SUBJECT TITLE : DATA VISUALIZATION AND CONCEPTS**

**OCT - 2025**



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY FACULTY OF SCIENCE AND HUMANITIES DEPARTMENT OF COMPUTER APPLICATIONS**

SRM Nagar, Kattankulathur – 603 203

**CERTIFICATE**

*Certified to be the bonafied record of practical work done by*

*Register No. of* ***I MSc ADS*** *Degree course for*

*–* ***DATA VISUALIZATION AND CONCEPTS*** *in the Computer lab*

*in SRM Institute of Science and Technology during the academic year 2024-2025.*

***Staff In-charge Head of the Department***

*Submitted for Semester Practical Examination held on \_.*

***Internal Examiner 1 Internal Examiner2***

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **DATE** | **TITLE** | **PAGE NO** | **SIGN** |
| 1 | 11-12-2024 | Getting Started with Tableau | 1-2 |  |
| 2 | 18-12-2024 | Working with Sample Datasets in Tableau | 3-4 |  |
| 3 | 06-01-2025 | Creating and Manipulating Data Tables | 5-6 |  |
| 4 | 20-01-2025 | Filters, Sorting, and Data Preparation | 7-9 |  |
| 5 | 23-01-2025 | Creating and Using Calculated Fields | 10-11 |  |
| 6 | 03-02-2025 | Building and Organizing Dashboards | 12-13 |  |
| 7 | 10-02-2025 | Real-Time Data Dashboards in Tableau | 14-15 |  |
| 8 | 14-02-2025 | Working with Storytelling in Tableau | 16-17 |  |
| 9 | 18-02-2025 | Plotting and Configuring Aesthetics in Tableau | 18-19 |  |
| 10 | 21-02-2025 | Integrating Tableau with External Data Sources (SQL, APIs, Google Sheets) | 20-21 |  |
| 11 | 25-02-2025 | Analyzing Bivariate Distributions & Multiple Variables | 22-23 |  |
| 12 | 28-02-2025 | Using Color Palettes and Customizing Visuals | 24-25 |  |
| 13 | 04-03-2025 | Advanced Analytics: Regression, Correlation, and Covariance | 26-27 |  |
| 14 | 07-03-2025 | Plotting and Configuring Aesthetics in Seaborn | 28-29 |  |
| 15 | 14-03-2025 | Customizing Matplotlib Visualizations (Aesthetics, Labels, Legends, Colors) | 30-32 |  |

**Date: 13/12/2024**

# Getting Started with Tableau

### Aim:

To understand the basics of Tableau, including installing the software, connecting to a dataset, and creating a simple visualization.

### Procedure:

1. **Install Tableau**
   * Download Tableau Desktop (trial) or Tableau Public (free).
   * Run the installer, accept the License Agreement, and install.
   * Open the software and sign in (if required).

### Connect to a Dataset

* + Click "Connect to Data" on the welcome screen.
  + Select Microsoft Excel and open Sample - Superstore.xlsx.
  + The dataset opens in the Data Source tab.

### Explore the Interface

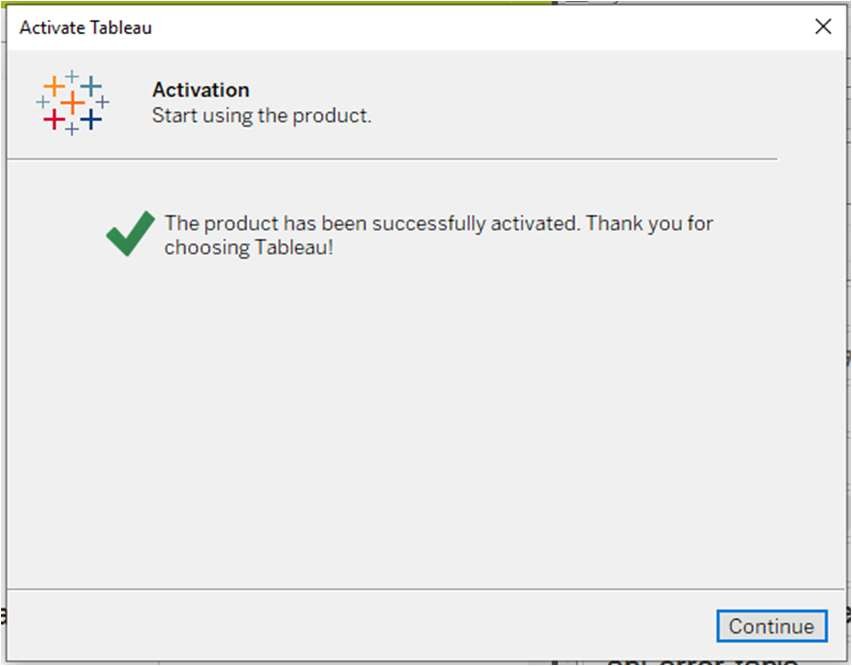
* + Left panel: Data Pane (shows fields and datasets).
  + Bottom: Sheets (for creating visualizations).
  + Top right: "Show Me" panel (chart options).

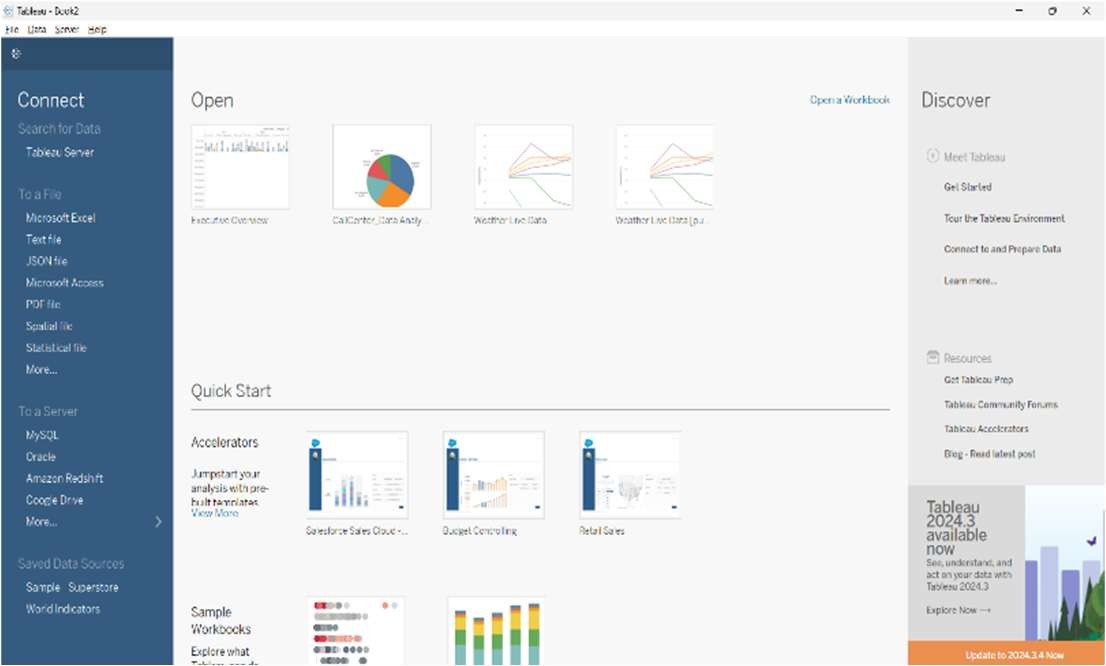
### Create a Basic Visualization

* + Click **Sheet 1** at the bottom.
  + Drag **Category** (Dimension) to **Columns**.
  + Drag **Sales** (Measure) to **Rows** → A bar chart appears.
  + Customize: Sort bars, add **Region** to Color shelf, apply filters.

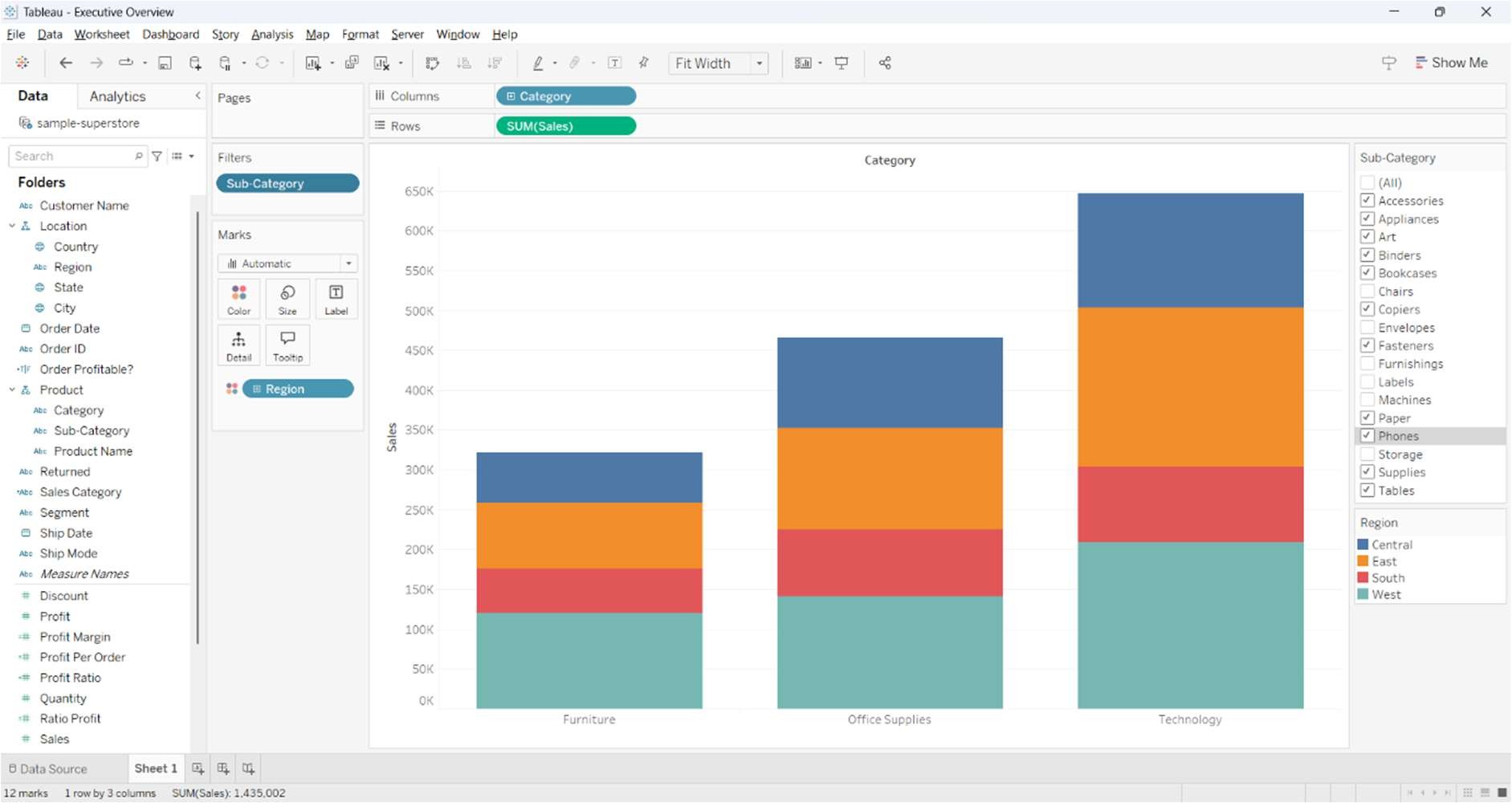
### Save & Export

* + Save as Getting\_Started.twb.
  + Export as **PNG** (File > Export > Image) or **PDF** (File > Print to PDF).





**Load Data**



## Result:

Successfully installed Tableau, connected to a sample dataset, and created a basic bar chart.

# Working with Sample Datasets in Tableau

### Aim:

To learn how to import, clean, and manipulate datasets in Tableau for analysis.

### Procedure:

1. **Open Tableau and Connect to Data**
   * Launch Tableau.
   * Click **"Connect to Data"** (Left Panel).
   * Select **Microsoft Excel** and open Sample - Superstore.xlsx.

### Explore the Data Source Tab

* + The dataset appears like a spreadsheet.
  + Dimensions (blue) = Categorical fields (e.g., Region, Product Name).
  + Measures (green) = Numerical fields (e.g., Sales, Profit).
  + Modify incorrect data types (e.g., change Order Date to Date).

### Create Extracts for Faster Performance

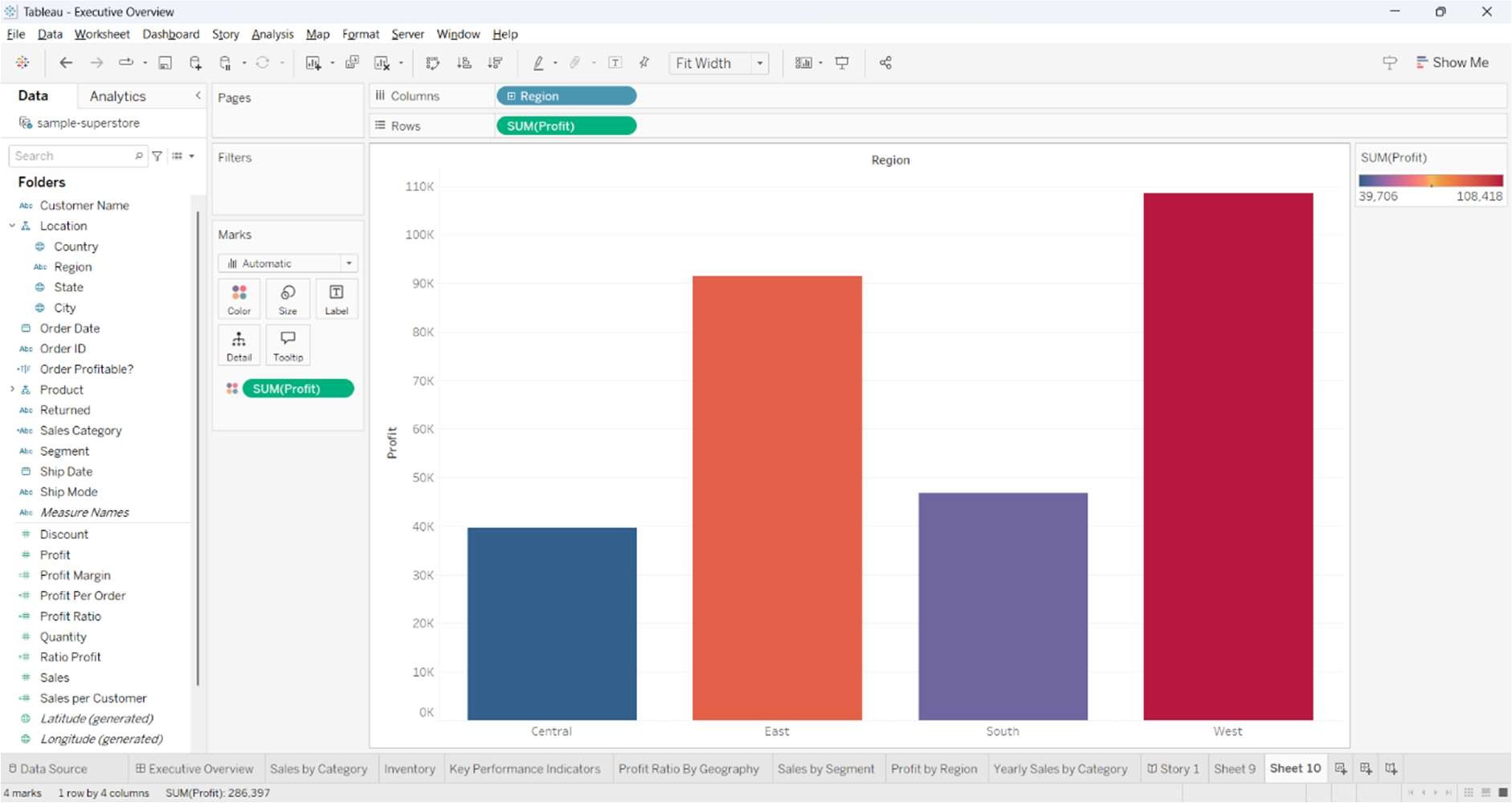
* + Click **Extract** (Top Right) instead of **Live** for better speed.
  + Click **Sheet 1** to start working.

### Create an Initial Visualization

* + Drag **Region** to Columns.
  + Drag **Profit** to Rows → A bar chart appears.
  + Click **Show Me** (top right) to choose a chart type (Map or Bar Chart).
  + Customize colors: Drag **Profit** to Color shelf.
  + Click **Sort** to arrange regions in descending order of profit.

### Save & Export

* + Save the workbook as Sample\_Dataset.twb.
  + Export as **PNG** (Worksheet > Export > Image) or **PDF** (File > Print to PDF).



## Result:

Successfully imported and explored a dataset in Tableau, created an extract, and built a visualization.

# Creating and Manipulating Data Tables in Tableau

### Aim:

To create custom data tables, manage fields, and use calculations for enhanced analysis in Tableau.

### Procedure:

1. **Open Tableau and Connect to Data**
   * Launch Tableau.
   * Click **"Connect to Data"** (Left Panel).
   * Select **Microsoft Excel** and open Sample - Superstore.xlsx.
   * Click **Sheet 1** to start working.

### Build a Data Table

* + Drag **Category** to Rows.
  + Drag **Sub-Category** to Rows (below Category).
  + Drag **Sales** to Columns.
  + Drag **Profit** to Columns (next to Sales).

### Apply Table Calculations

* + Right-click **Sales** > **Quick Table Calculation** > **Running Total** (shows cumulative sales).
  + Right-click **Profit** > **Quick Table Calculation** > **Percent of Total** (converts profit into percentage).

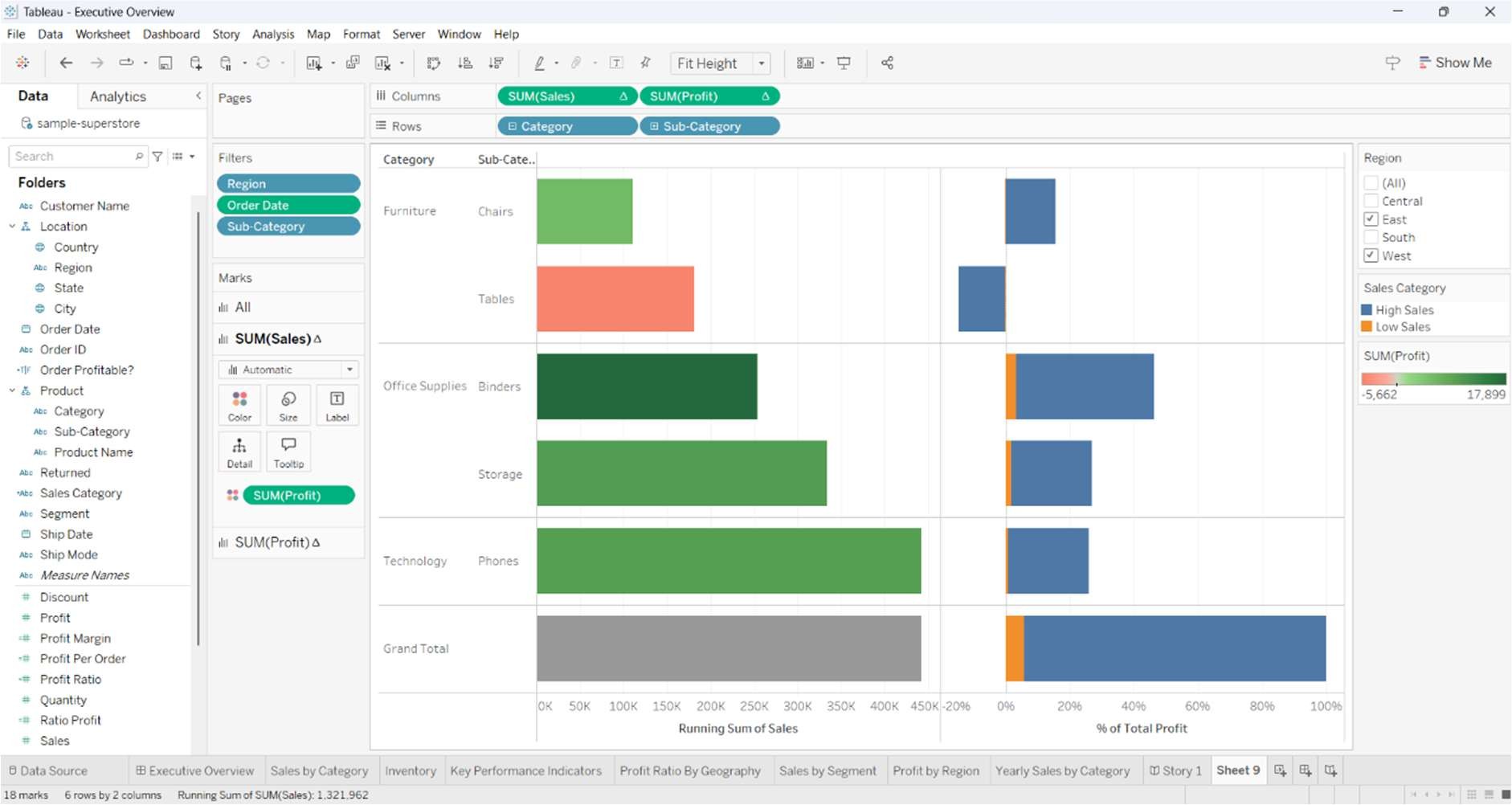
### Format the Table

* + Click **Format** (Top Menu).
  + Set **numeric fields** to Currency for Sales and Profit.
  + Drag **Profit** to Color shelf and use a **Green-Red diverging scale**.

### Click Analysis > Totals > Show Column Grand Totals.

1. **Save & Export**
   * Save as Data\_Table.twb.
   * Export as **Excel** (Worksheet > Export > Crosstab to Excel).
   * Alternatively, export as **PDF** or **Image**.





## Result:

Successfully created and formatted a data table in Tableau with calculations and styling.

# Filters, Sorting, and Data Preparation in Tableau

### Aim:

To learn how to filter and sort data in Tableau for better analysis and data preparation.

### Procedure:

1. **Open Tableau and Connect to Data**
   * Launch Tableau.
   * Click **"Connect to Data"** (Left Panel).
   * Select **Microsoft Excel** and open Sample - Superstore.xlsx.
   * Click **Sheet 1** to start working.

### Sorting Data

* + Drag **Category** to Columns.
  + Drag **Sales** to Rows → A bar chart appears.
  + Click the **Sort** icon above the chart to arrange bars in descending order of Sales.
  + To sort by another field: Click **Sort > Sort by Field > Choose Profit > Descending**.

### Applying Filters

* + **Basic Filtering**:
    - Drag **Region** to the Filters Shelf.
    - Select only **East and West** regions and click **OK**.

### Interactive Filters:

* + - Right-click **Region** in Filters Shelf > **Show Filter** → Users can now toggle regions dynamically.

### Date Filtering:

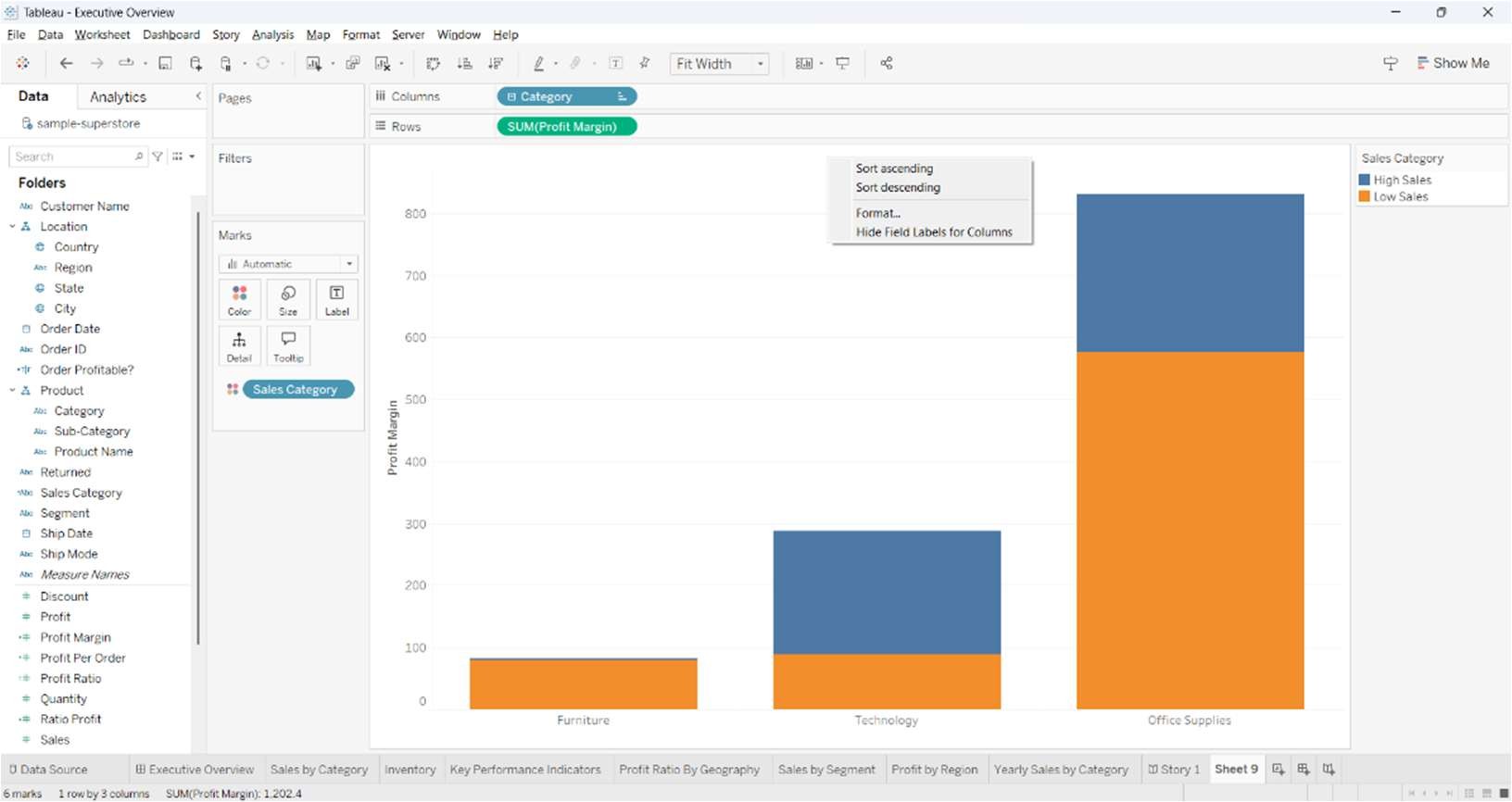
* + - Drag **Order Date** to Filters Shelf.
    - Choose **Relative Date > Last 2 Years** → Displays only recent data.

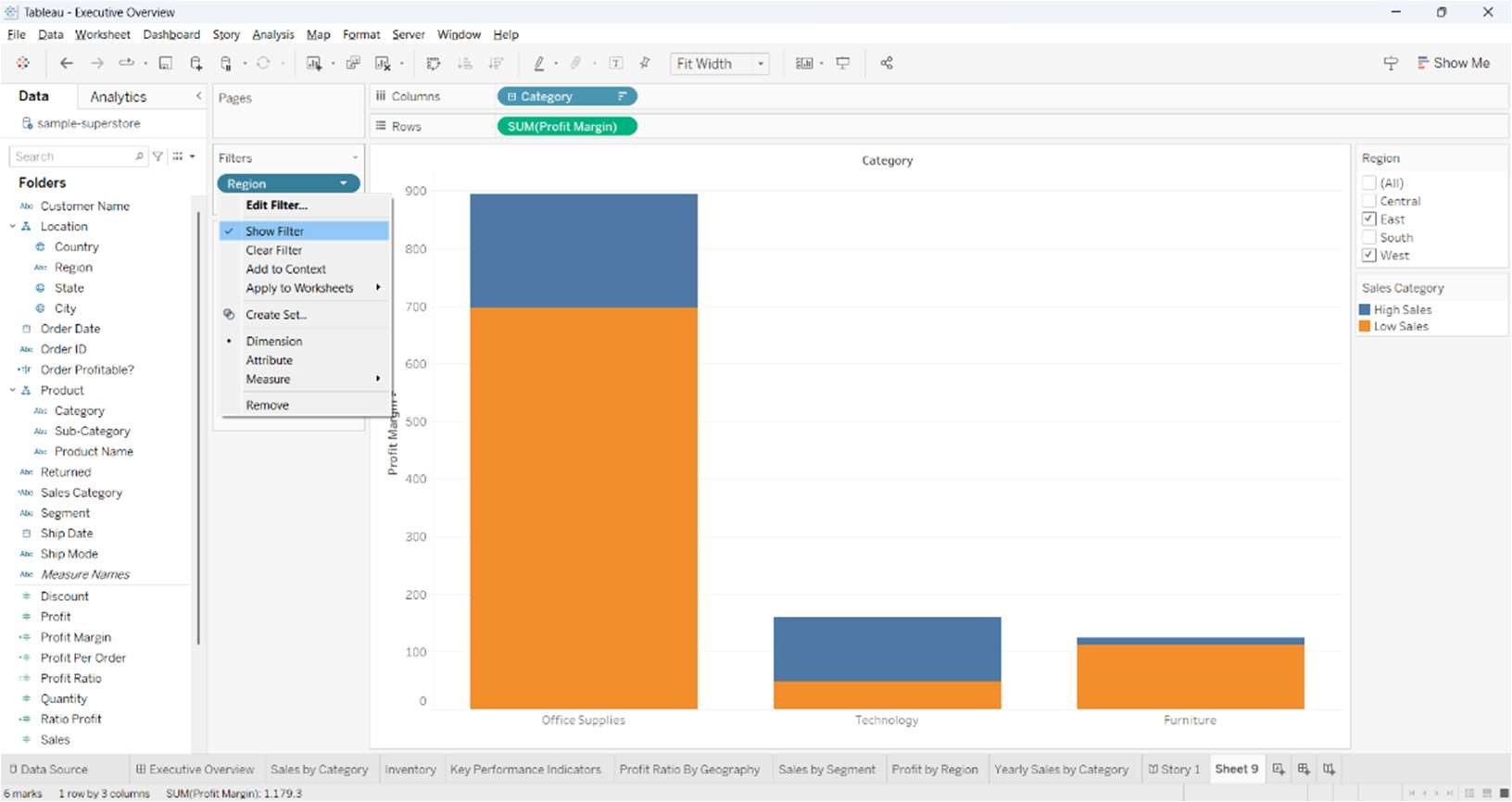
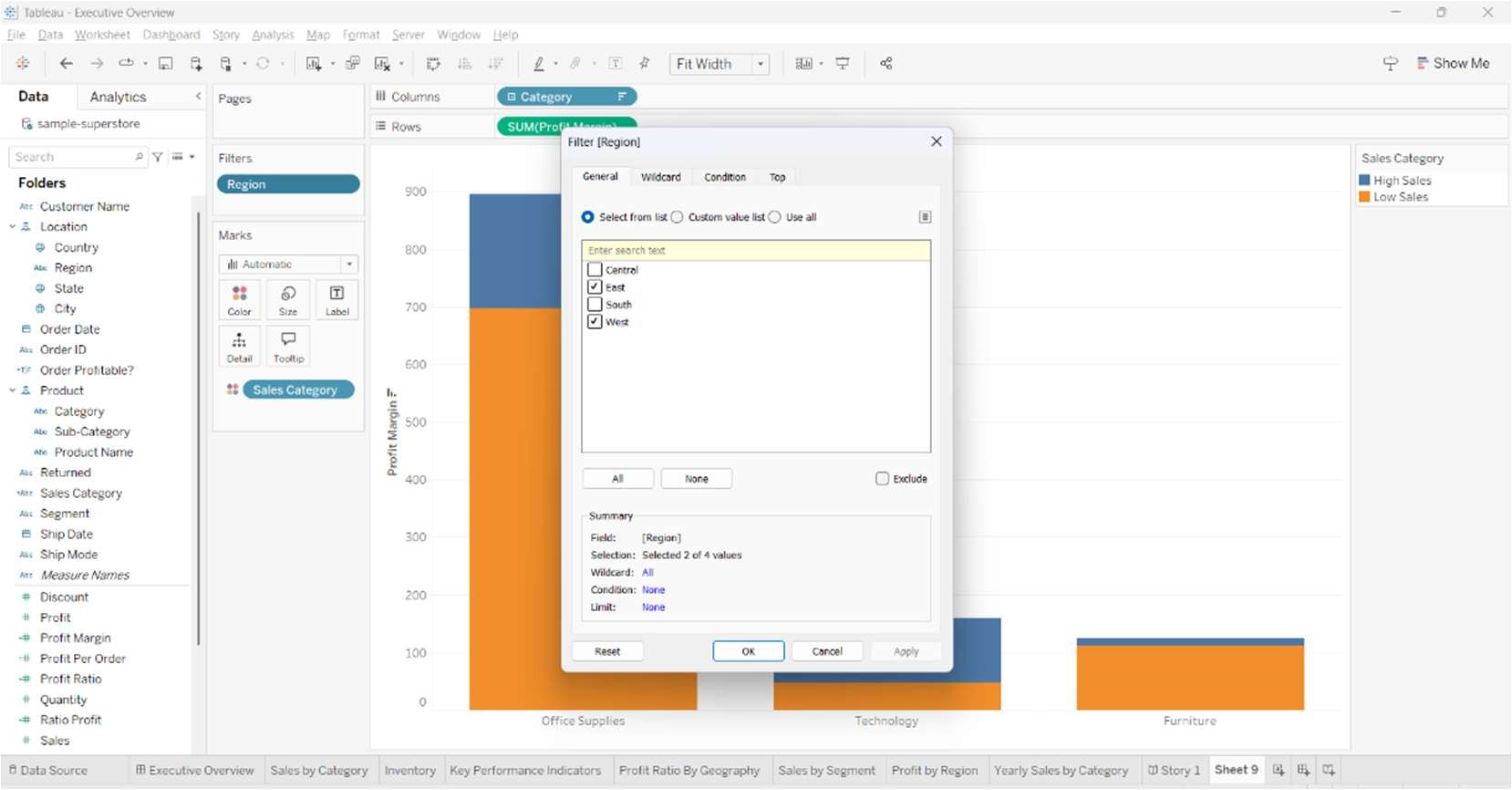
### Creating a Top N Filter

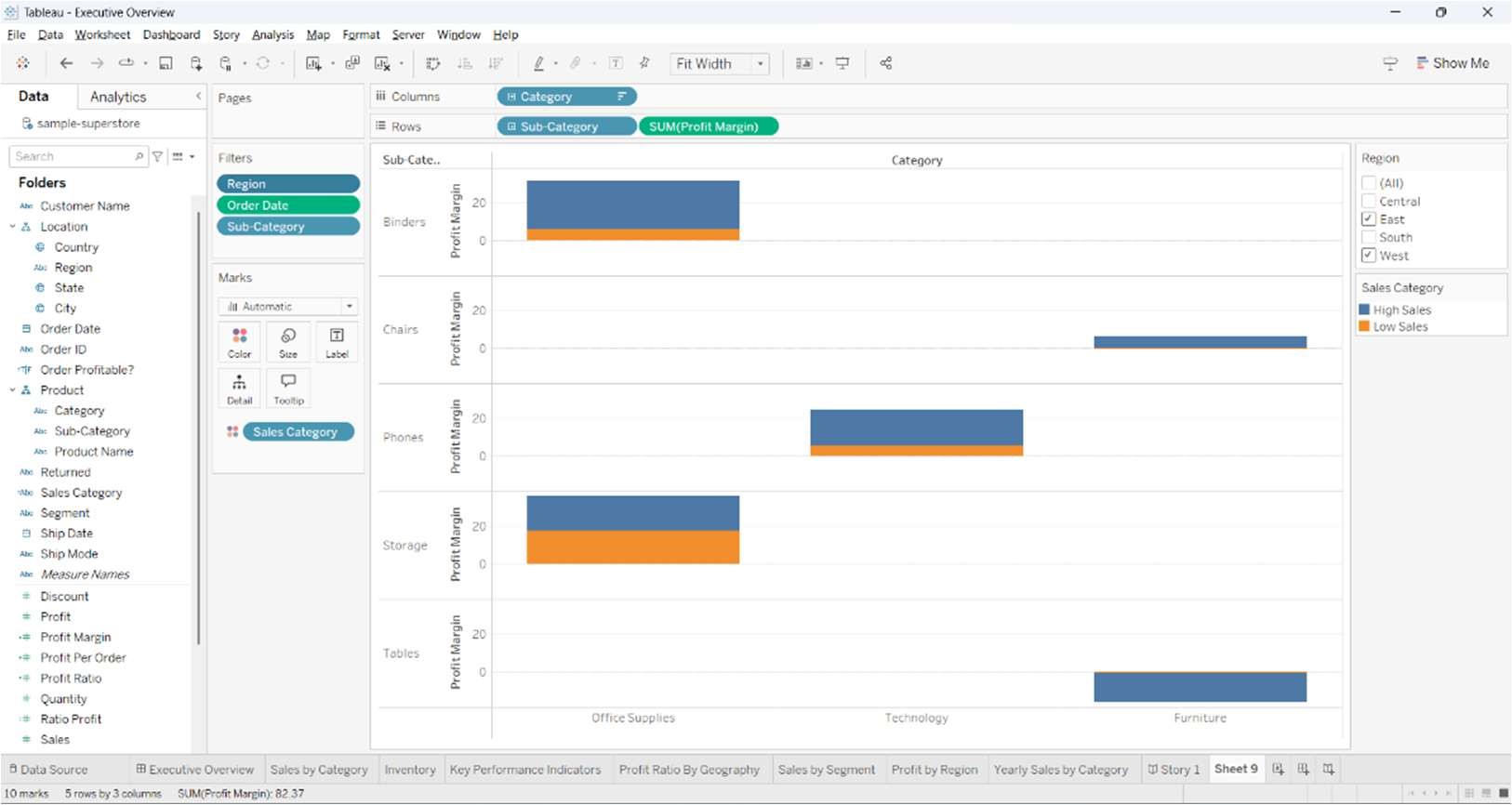
* + Drag **Sub-Category** to Rows.
  + Right-click **Sub-Category** > **Filter**.
  + Go to the **Top** Tab > Select **By Field**.
  + Enter **Top 5 by Sales** → Displays only the top 5 sub-categories.

### Save & Export

* + Save as Filters\_and\_Sorting.twb.
  + Export as **PNG** (Worksheet > Export > Image) or **PDF** (File > Print to PDF).







**Result:** Successfully applied filters and sorting in Tableau to refine data for better insights.

# Creating and Using Calculated Fields in Tableau

### Aim:

To learn how to create and use calculated fields in Tableau to derive new insights from data.

### Procedure:

1. **Open Tableau and Connect to Data**
   * Launch Tableau.
   * Click **"Connect to Data"** (Left Panel).
   * Select **Microsoft Excel** and open Sample - Superstore.xlsx.
   * Click **Sheet 1** to start working.

### Creating a Basic Calculated Field

* + Click on the **Data Pane** (Left Panel).
  + Click the dropdown next to **Sales** > **Create Calculated Field**.
  + Enter the formula:

Profit Margin = [Profit] / [Sales]

* + Click **OK** → A new field called **Profit Margin** appears under Measures.

### Using the Calculated Field in a Visualization

* + Drag **Category** to Columns.
  + Drag **Profit Margin** to Rows.
  + Convert it to a **Percentage**:
    1. Right-click **Profit Margin** in the Data Pane.

### Select Default Properties > Number Format > Percentage.

* + Click **Show Me** (top right) and select **Bar Chart**.

### Creating an IF Condition Calculation

* + Click **Create Calculated Field** again.
  + Name it **Sales Category**.
  + Enter the formula:

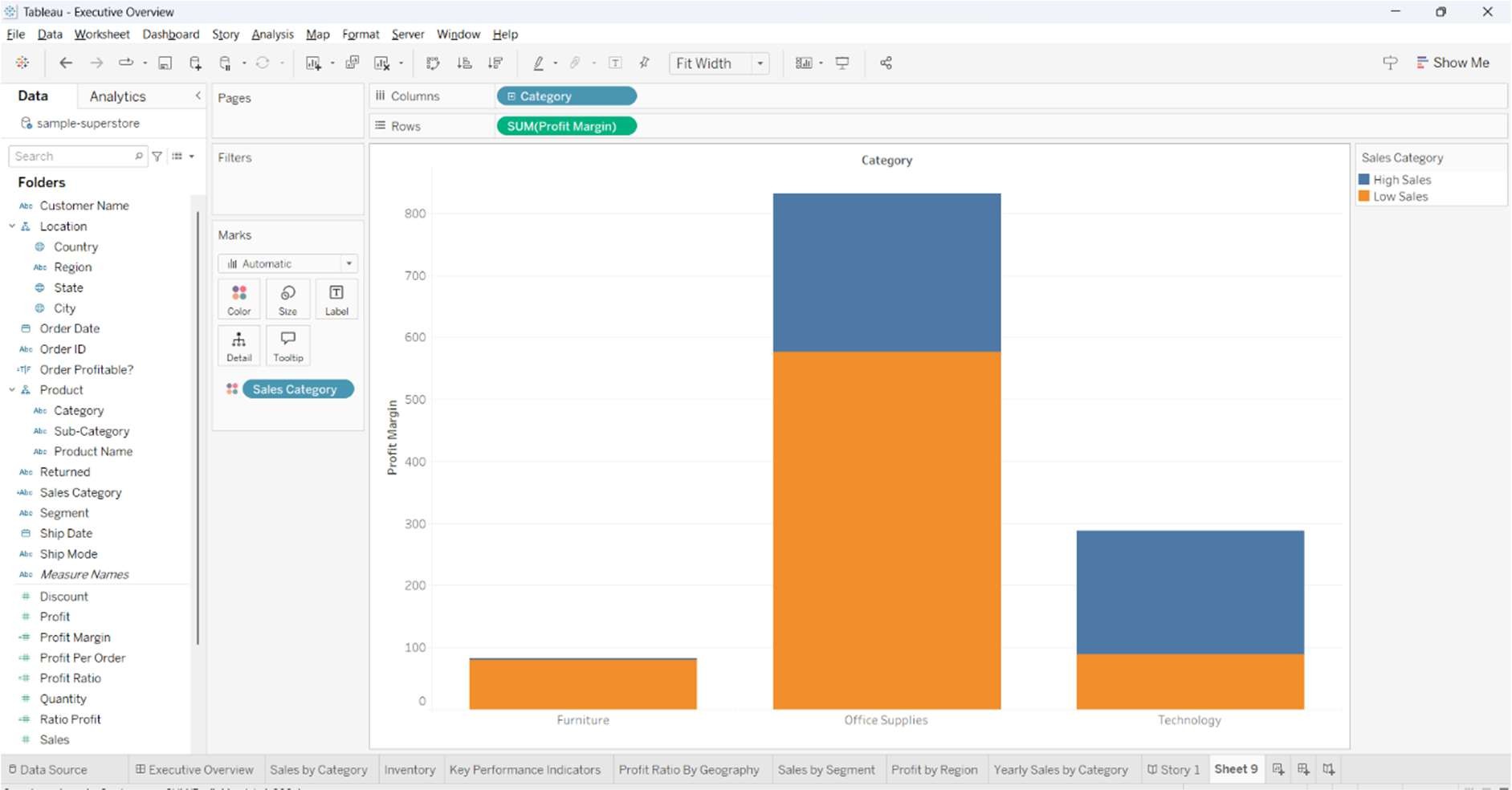
IF [Sales] > 90 THEN "High Sales" ELSE "Low Sales" END

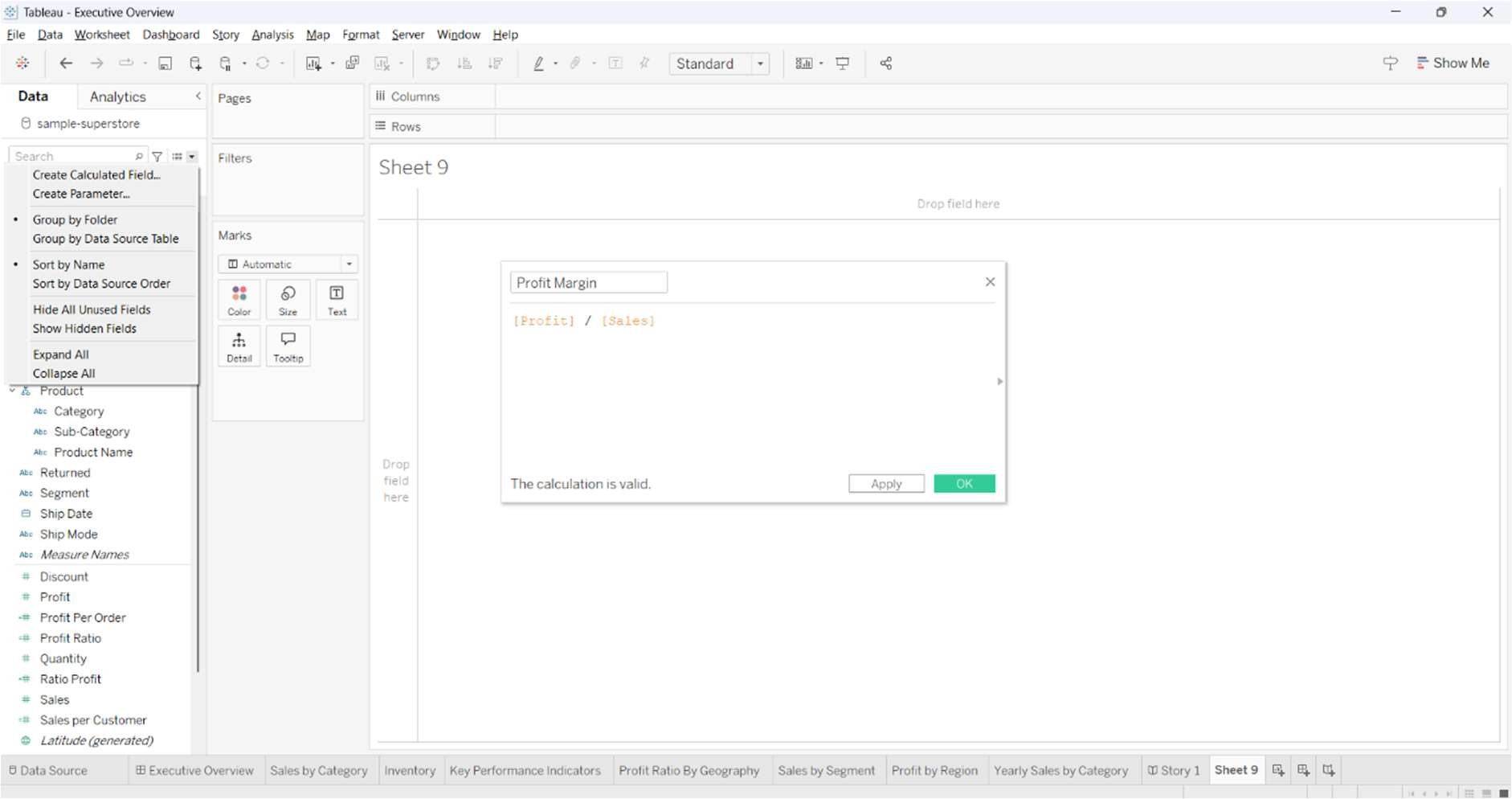
* + Click **OK**.
  + Drag **Sales Category** to the **Color Shelf** → Categories are now color-coded based on sales performance.

### Save & Export

* + Save as Calculated\_Fields.twb.
  + Export as **PNG** (Worksheet > Export > Image) or **PDF** (File > Print to PDF).

## Implementation:

****



**Calculated Field**

**Formula Pane**

**Result:** Successfully created and used calculated fields in Tableau for enhanced data analysis.

# Building and Organizing Dashboards in Tableau

### Aim:

To create a dashboard in Tableau by combining multiple visualizations for better data representation.

### Procedure:

1. **Open Tableau and Connect to Data**
   * Load Sample - Superstore.xlsx.

### Create Charts

* + **Bar Chart (Sales by Category):** Drag **Category** to Columns and **Sales** to Rows.
  + **Map (Profit by Region):** Drag **State** to Detail and **Profit** to Color.
  + **Line Chart (Sales Trend):** Drag **Order Date** to Columns and **Sales** to Rows.

### Build the Dashboard

* + Click **New Dashboard**.
  + Drag and arrange the charts.

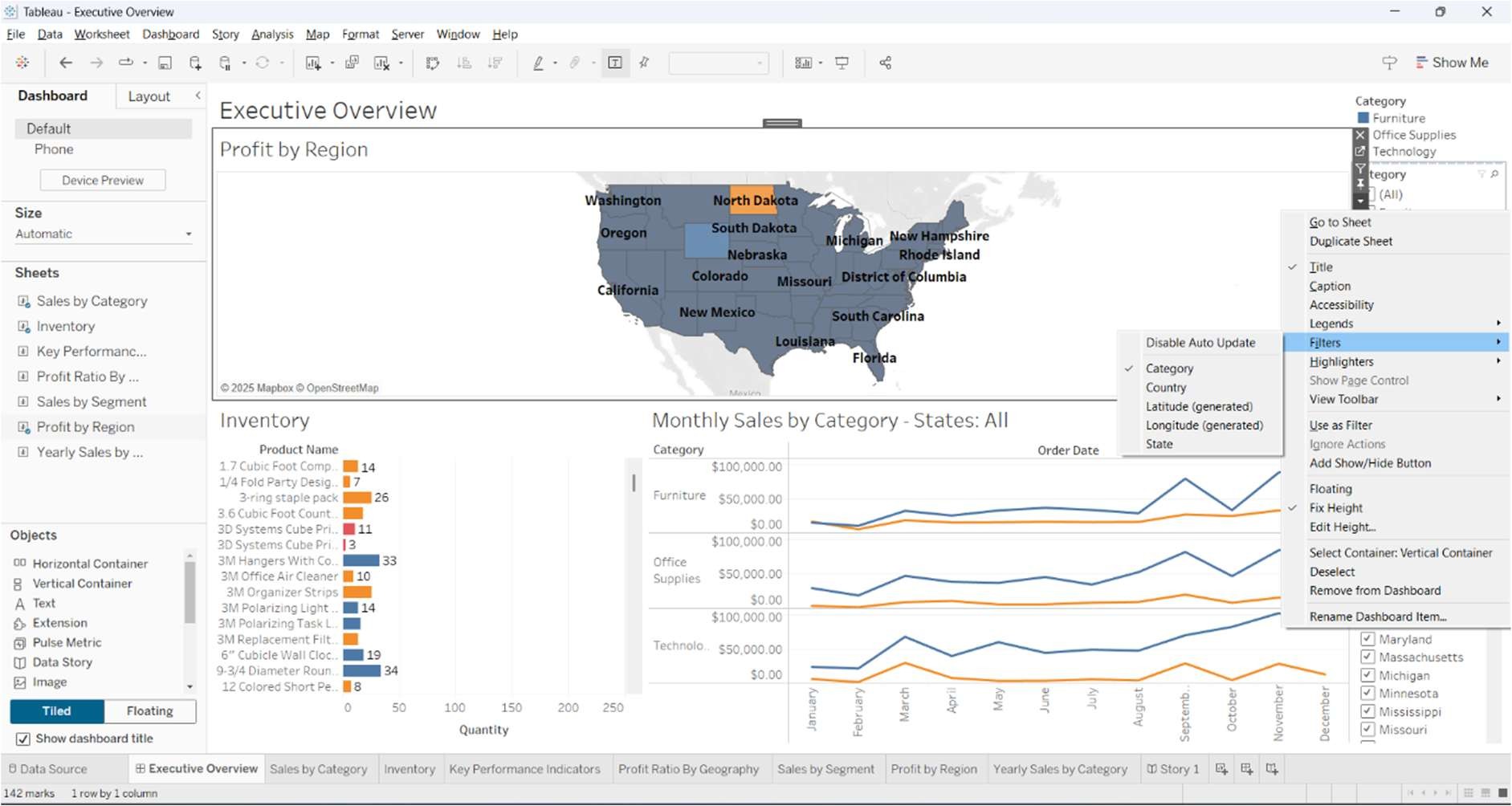
### Add Filters & Title

* + Click **Show Filter** for interactive controls.
  + Click **Dashboard > Show Title** and rename it.

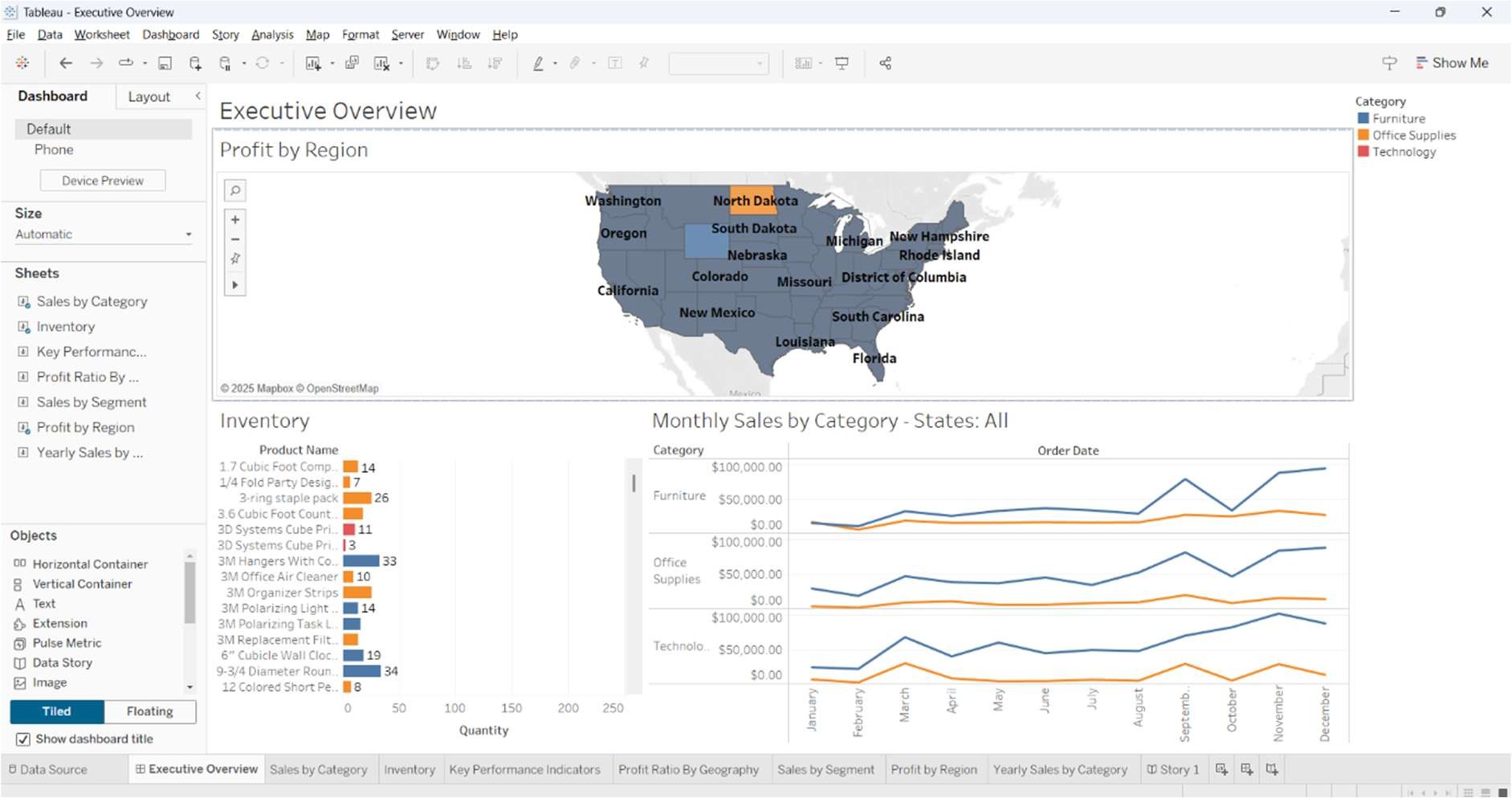
### Save & Export

* + Save as Dashboard\_Analysis.twb.
  + Export as **PNG** or **PDF**.

## Implementation:



**Apply Filters**



**Result:**

Successfully built an interactive dashboard in Tableau combining multiple visualizations.

# Real-Time Data Dashboards in Tableau

### Aim:

To create a real-time data dashboard in Tableau by connecting to live data sources and enabling automatic updates.

**Procedure:**

1. **Open Tableau and Connect to Live Data**
   * Select **MySQL, SQL Server, Google Sheets, or Web Data Connector**.
   * Enter connection details and load the data.

### Create Real-Time Charts

* + **Line Chart (Sales Trend):**
    - Drag **Order Date** to Columns and **Sales** to Rows.

### Bar Chart (Inventory):

* + - Drag **Product Name** to Rows and **Stock Quantity** to Columns.

### Build the Dashboard

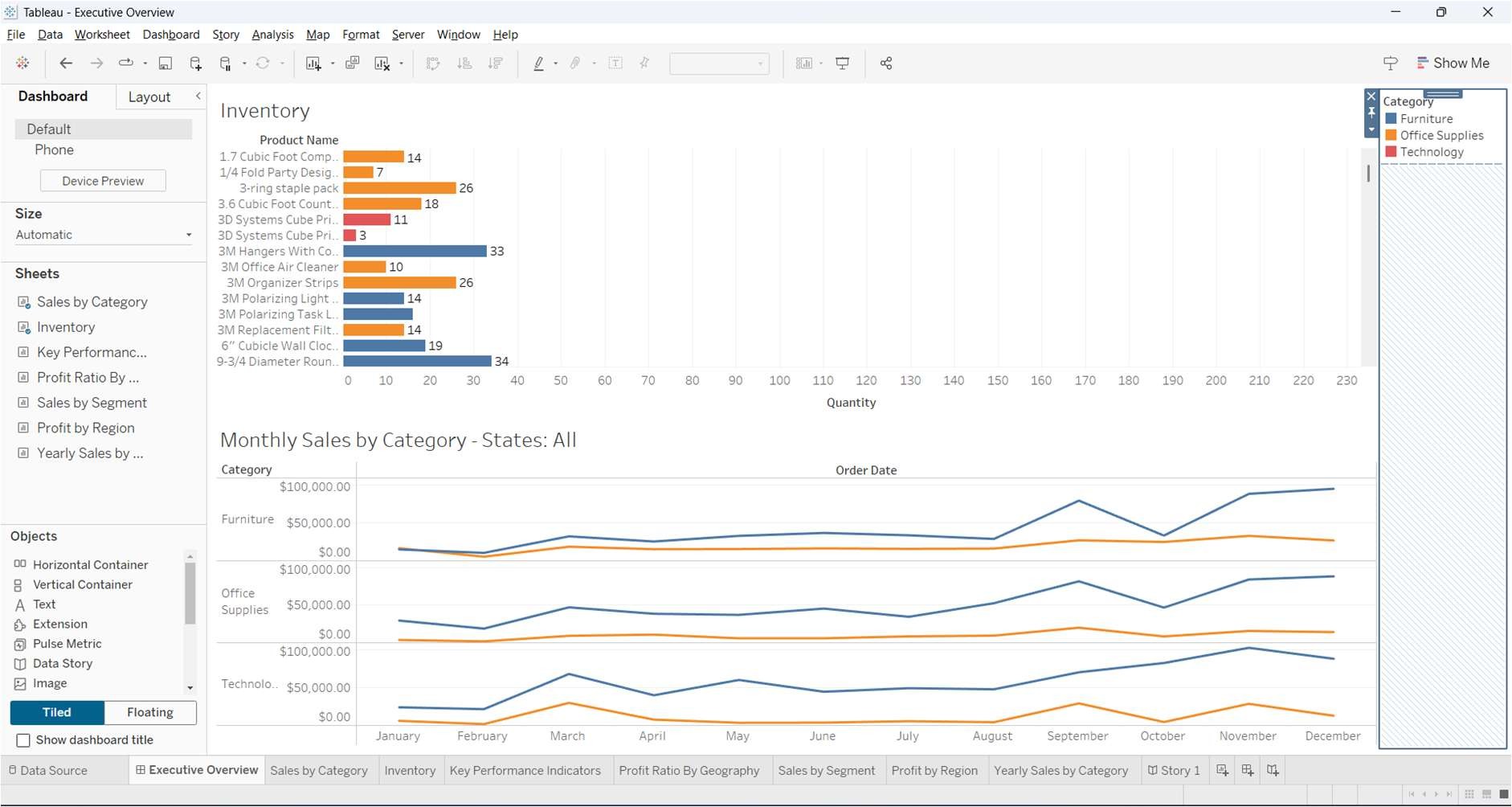
* + Click **New Dashboard**.
  + Drag and arrange the charts.

### Enable Auto-Refresh

* + Click **Data > Refresh Data Automatically**.
  + For **Google Sheets**, enable **Auto-Refresh Every 1 Minute**.

### Save & Share

* + Save as RealTime\_Dashboard.twb.
  + Publish to **Tableau Public** for online access.



## Result:

Successfully created a real-time dashboard in Tableau with live data updates.

# Working with Storytelling in Tableau

### Aim:

To create a story in Tableau using multiple dashboards and visualizations to present data- driven insights effectively.

### Procedure:

1. **Open Tableau and Connect to Data**
   * Load Sample - Superstore.xlsx.

### Create Key Charts

* + **Bar Chart (Sales by Category):** Drag **Category** to Columns and **Sales** to Rows.
  + **Map (Profit by Region):** Drag **State** to Detail and **Profit** to Color.
  + **Line Chart (Yearly Sales Trend):** Drag **Order Date** to Columns and **Sales**

to Rows.

### Create a Story

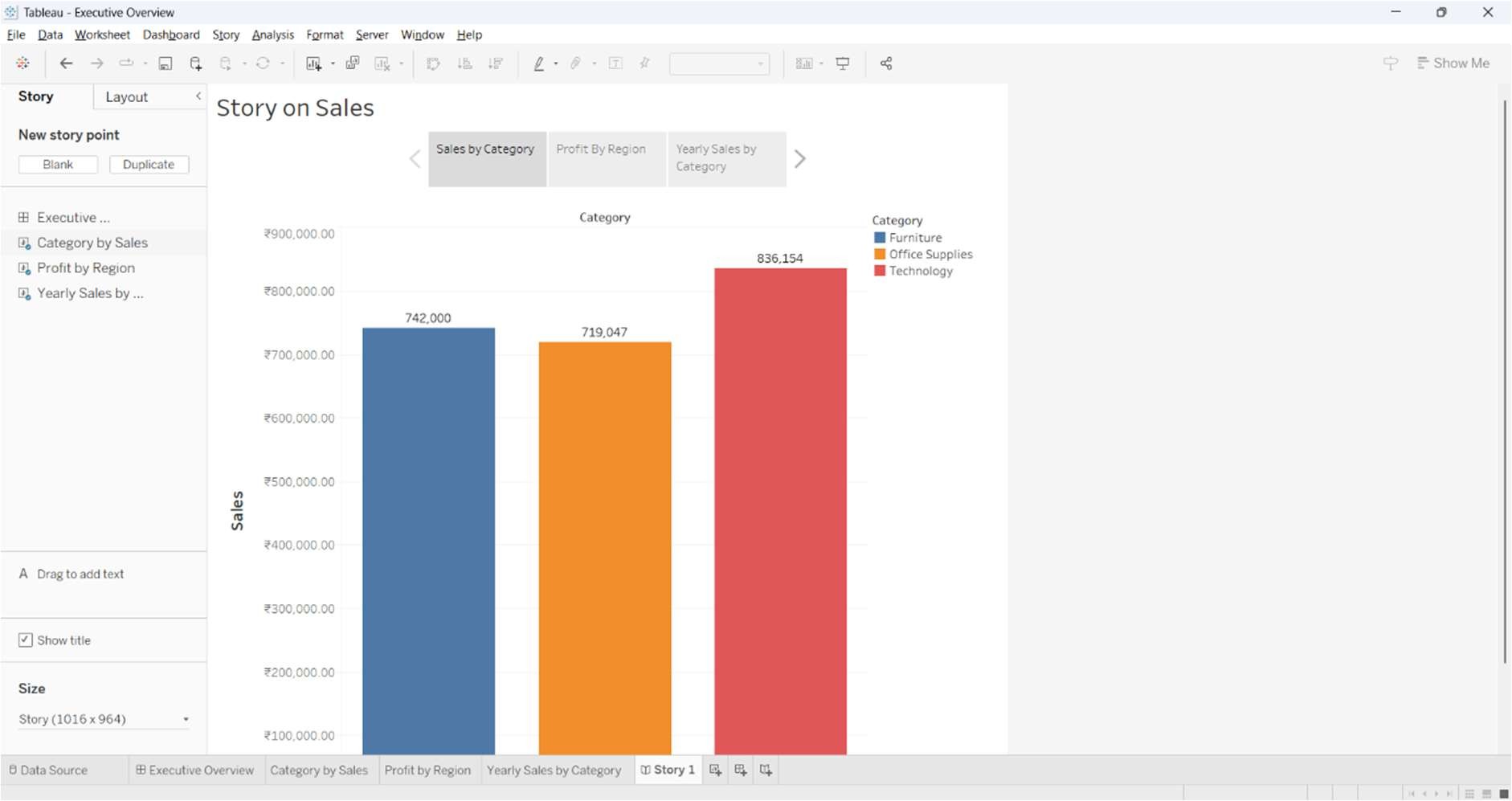
* + Click **New Story**.
  + Drag **Sales by Category** onto the canvas.
  + Click **Add Caption** and enter a title.

### Add More Story Points

* + Click **New Blank Point** and add other visualizations.
  + Add captions to describe insights.

### Save & Export

* + Save as Storytelling\_Analysis.twb.
  + Export as **PNG** or **PDF**.



**Load Data**

**Captions**

**Chart Sheets**

**Create individual Sheets**

**Create Story**

**New Blank Point**

## Result:

Successfully created a data-driven story in Tableau to present insights effectively.

# Plotting and Configuring Aesthetics in Tableau

### Aim:

To enhance the appearance of visualizations in Tableau by customizing colors, labels, tooltips, and layouts.

### Procedure:

1. **Open Tableau and Connect to Data**
   * Load Sample - Superstore.xlsx.

### Create a Basic Chart

* + Drag **Category** to Columns and **Sales** to Rows.
  + Click **Show Me** and select **Bar Chart**.

### Customize Appearance

* + **Change Colors:** Drag **Category** to the **Color Shelf**.
  + **Add Labels:** Click **Label** in Marks Card and check **Show Mark Labels**.
  + **Modify Tooltips:** Click **Tooltip** and add details like **Profit and Sales**.

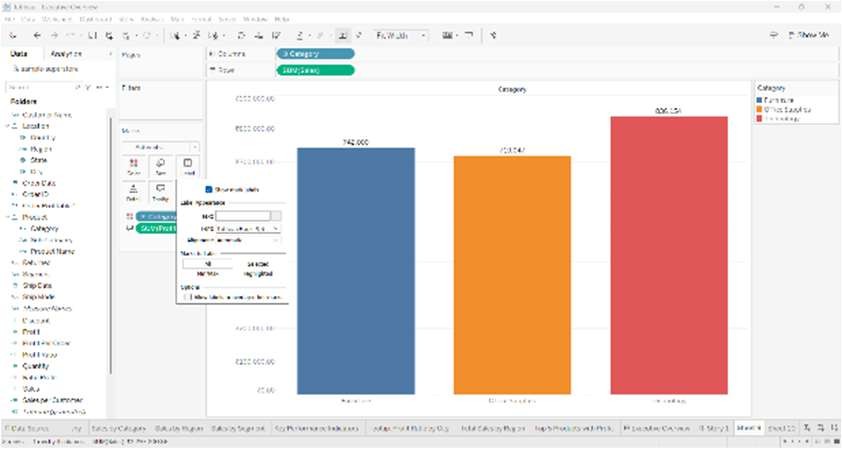
### Adjust Axis, Fonts, and Layout

* + Right-click **Sales Axis > Format** > Change to **Currency**.
  + Change font styles using **Format > Fonts**.
  + Resize and align elements for better readability.

### Save & Export

* + Save as Customized\_Visualization.twb.
  + Export as **PNG** or **PDF**.

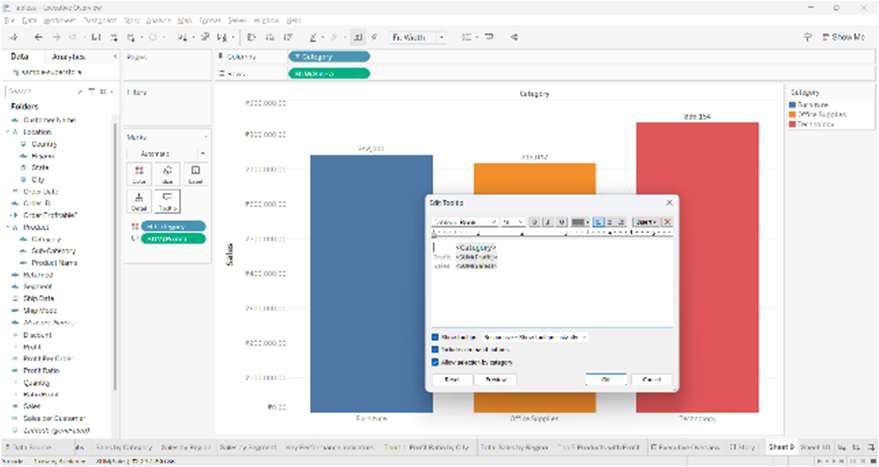
## Implementation:



**Enable Labels**

**Drag Required columns**

**Load Dataset**



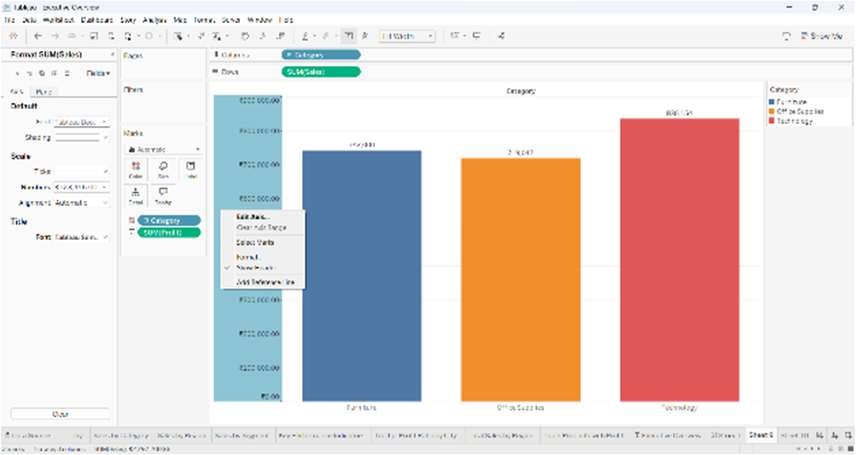
**Customize Tooltip**

**Drag Column**

**to Color**

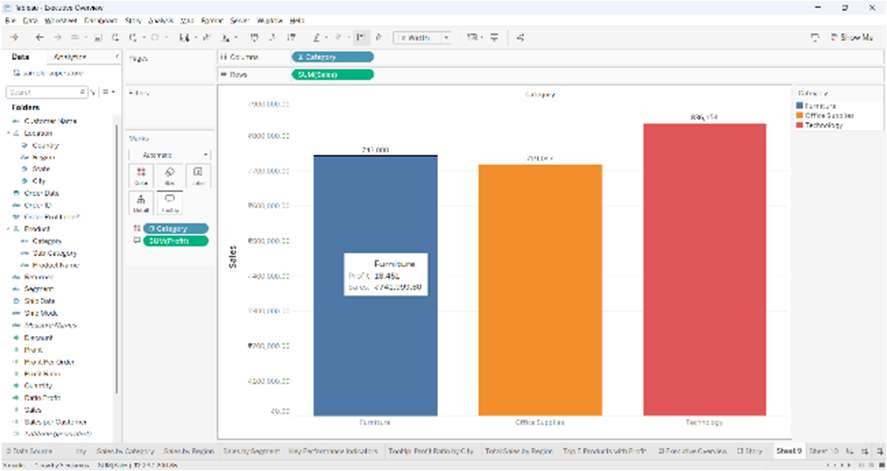
**Drag Column**

**to Tooltip**



**Format Pane**

**Format Axis**



**Formatted Axis**

**Labels**

**Tooltip**

## Result:

Successfully customized the aesthetics of a Tableau visualization using colors, labels, tooltips, and formatting.

# Integrating Tableau with External Data Sources (SQL, APIs, Google Sheets)

### Aim:

To connect Tableau with external data sources such as SQL databases, APIs, and Google Sheets for dynamic data visualization.

**Procedure:**

1. **Connect to an SQL Database**
   * Click **"Connect to Data"** > **Microsoft SQL Server/MySQL/PostgreSQL**.
   * Enter **server name, database name, username, and password**.
   * Select tables and click **Update Now** to load data.

### Connect to Google Sheets

* + Click **"Connect to Data"** > **Google Sheets**.
  + Sign in and select a spreadsheet.

### Connect to a Web API

* + Click **"Connect to Data"** > **Web Data Connector**.
  + Enter the **API URL** and click **Load**.

### Create a Visualization

* + Drag **Date** to Columns and **Sales** to Rows.
  + Select a **Line Chart** from **Show Me**.
  + Apply filters for better analysis.

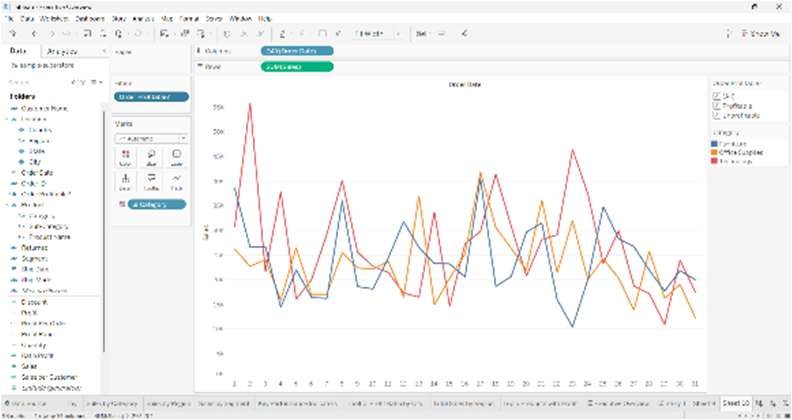
### Enable Live Connection & Auto Refresh

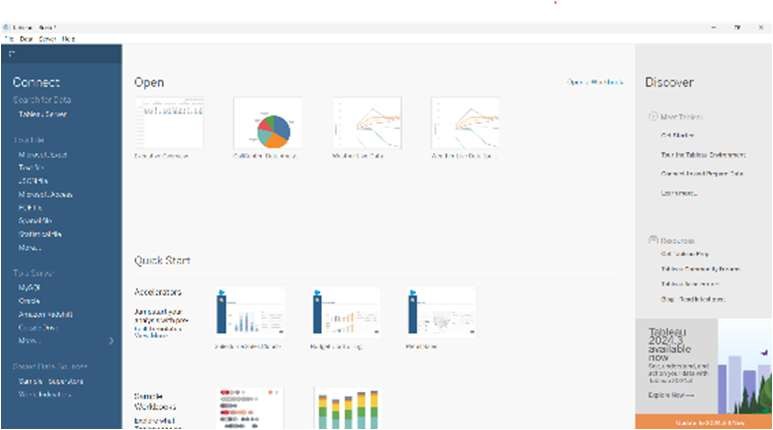
* + In **Data Source Tab**, select **Live** instead of **Extract** for real-time updates.
  + For **Google Sheets**, set auto-refresh every few minutes.

### Save & Export

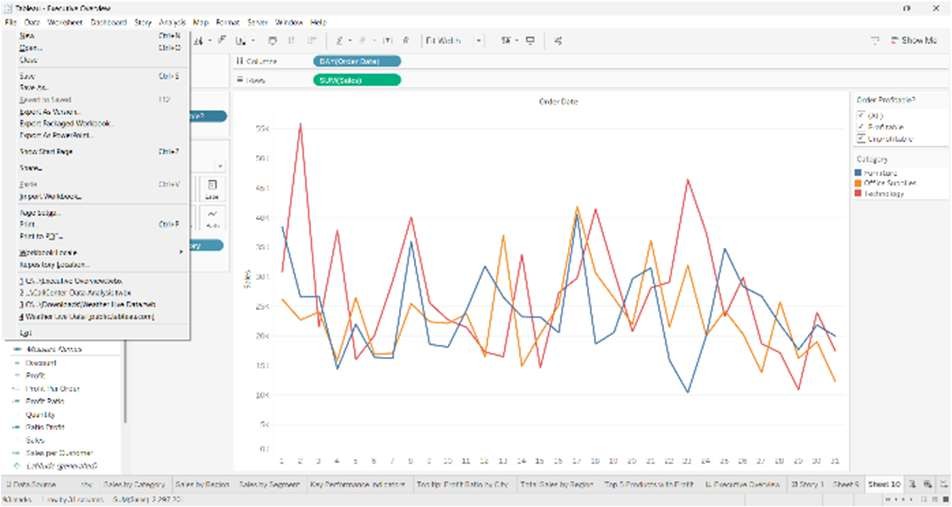
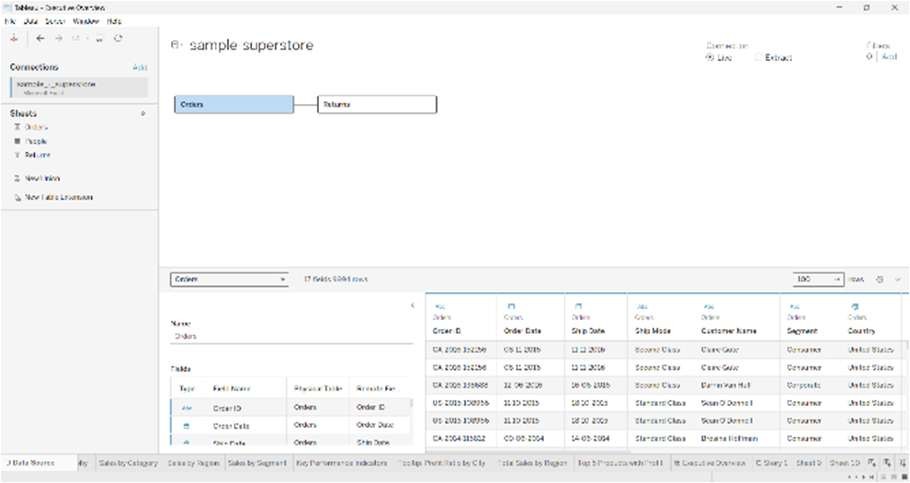
* + Save as External\_Data\_Integration.twb.
  + Export as **PNG** or **PDF**.

## Implementation:

****



**Live Data sources**



**Export**

## Result:

Successfully integrated Tableau with an external data source and created a visualization from live data.

# Analyzing Bivariate Distributions & Multiple Variables

## Aim:

To analyze bivariate distributions and relationships between multiple variables using Seaborn visualizations such as scatter plots, pair plots, and heatmaps.

## Procedure:

1. Install **Seaborn** and **Matplotlib** if not already installed.
2. Import the required libraries.
3. Load a sample dataset or create random data.
4. Create various visualizations such as:
   * Scatter Plot with regression line
   * Pair Plot for multivariable analysis
   * Heatmap for correlation analysis

## Code:

# Step 1: Import required libraries import seaborn as sns

import matplotlib.pyplot as plt import pandas as pd

# Step 2: Load a sample dataset

df = sns.load\_dataset("tips") # Built-in dataset in Seaborn

# Step 3: Scatter Plot - Analyzing two variables plt.figure(figsize=(8, 6))

sns.scatterplot(x="total\_bill", y="tip", data=df, hue="sex", style="time", palette="coolwarm")

plt.title("Total Bill vs Tip Amount", fontsize=14) plt.xlabel("Total Bill ($)", fontsize=12) plt.ylabel("Tip ($)", fontsize=12) plt.legend(title="Gender & Time")

plt.show()

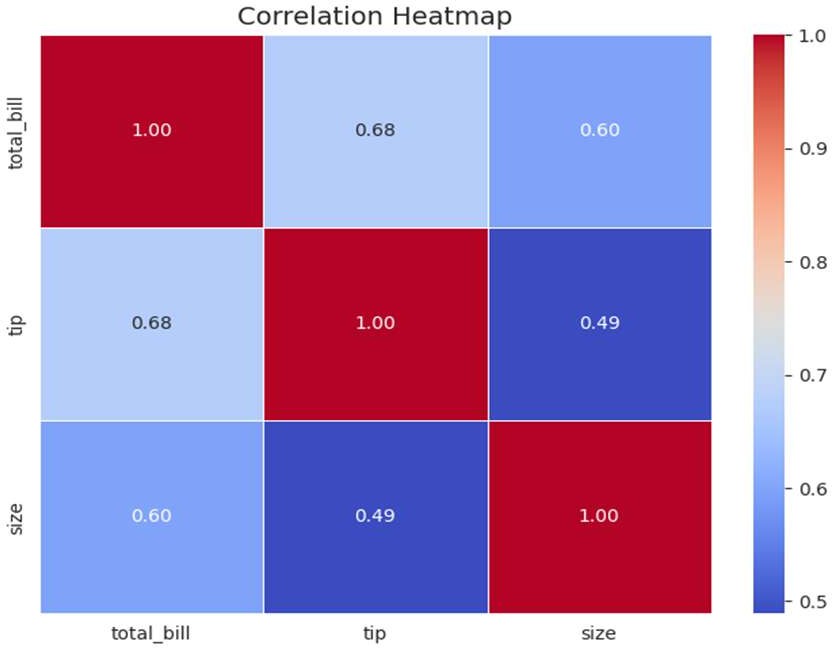
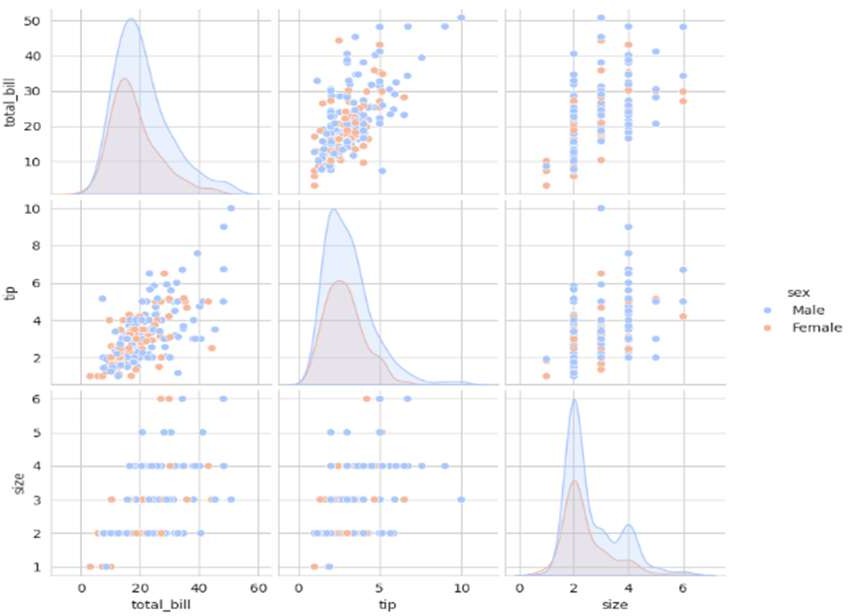
# Step 4: Pair Plot - Exploring relationships between multiple variables sns.pairplot(df, hue="sex", diag\_kind="kde", palette="coolwarm") plt.show()

# Step 5: Heatmap - Analyzing correlation between numeric variables plt.figure(figsize=(8, 6))

sns.heatmap(df.drop(columns=['sex', 'smoker', 'day', 'time']).corr(), annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5) plt.title("Correlation Heatmap", fontsize=14)

plt.show()

## Implementation:

****

**Result:**

Successfully analyzed **bivariate distributions and multiple variables** using **scatter plots, pair plots, and heatmaps** in Seaborn. These visualizations help in understanding data relationships effectively.

# Using Color Palettes and Customizing Visuals

## Aim:

To explore different color palettes in Seaborn and customize visualizations for better readability and aesthetics.

## Procedure:

1. Install **Seaborn** and **Matplotlib** if not installed.
2. Import required libraries.
3. Load a sample dataset.
4. Apply different color palettes to Seaborn visualizations such as:
   * Bar plot with a **custom color palette**
   * Box plot with a **set style and color palette**
   * Violin plot with **hue-based colors**
   * Customizing **figure size, labels, and grid styles**

## Code:

# Step 1: Import required libraries import seaborn as sns

import matplotlib.pyplot as plt

# Step 2: Load a sample dataset

df = sns.load\_dataset("penguins") # Built-in dataset in Seaborn

# Step 3: Set a Seaborn style and color palette sns.set\_style("whitegrid") # Options: darkgrid, whitegrid, dark, white, ticks

sns.set\_palette("pastel") # Options: deep, muted, bright, pastel, dark, colorblind

# Step 4: Bar Plot with Custom Colors plt.figure(figsize=(8, 6))

sns.barplot(x="species", y="body\_mass\_g", data=df, palette="viridis") plt.title("Average Body Mass of Penguin Species", fontsize=14) plt.xlabel("Species", fontsize=12)

plt.ylabel("Body Mass (g)", fontsize=12) plt.show()

# Step 5: Box Plot with Different Color Palettes plt.figure(figsize=(8, 6))

sns.boxplot(x="species", y="flipper\_length\_mm", data=df, palette="coolwarm", hue="sex")

plt.title("Flipper Length by Species and Sex", fontsize=14) plt.xlabel("Species", fontsize=12)

plt.ylabel("Flipper Length (mm)", fontsize=12) plt.legend(title="Sex")

plt.show()

# Step 6: Violin Plot with Hue-based Coloring plt.figure(figsize=(8, 6))

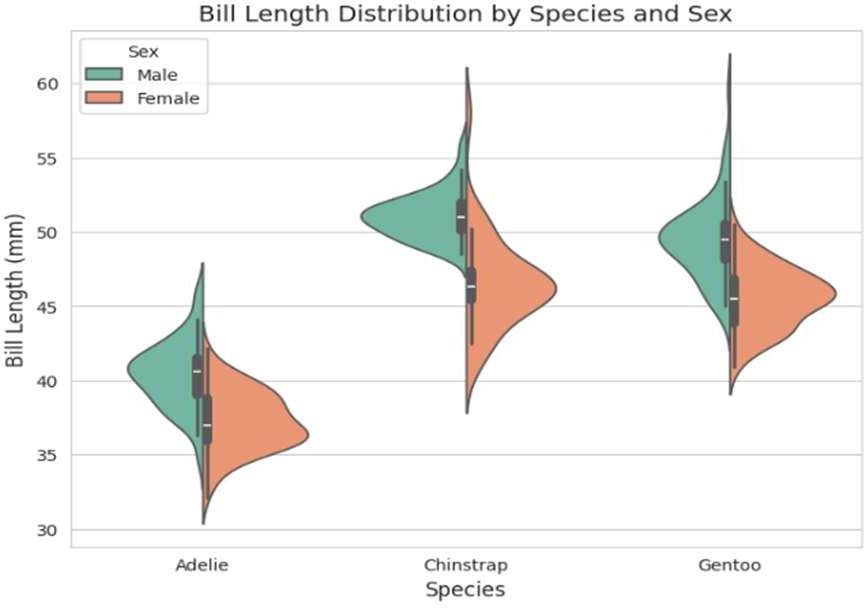
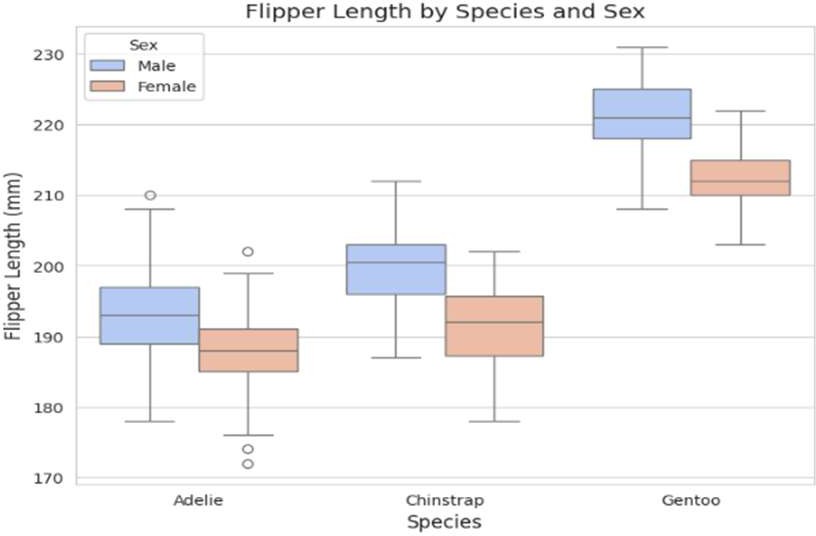
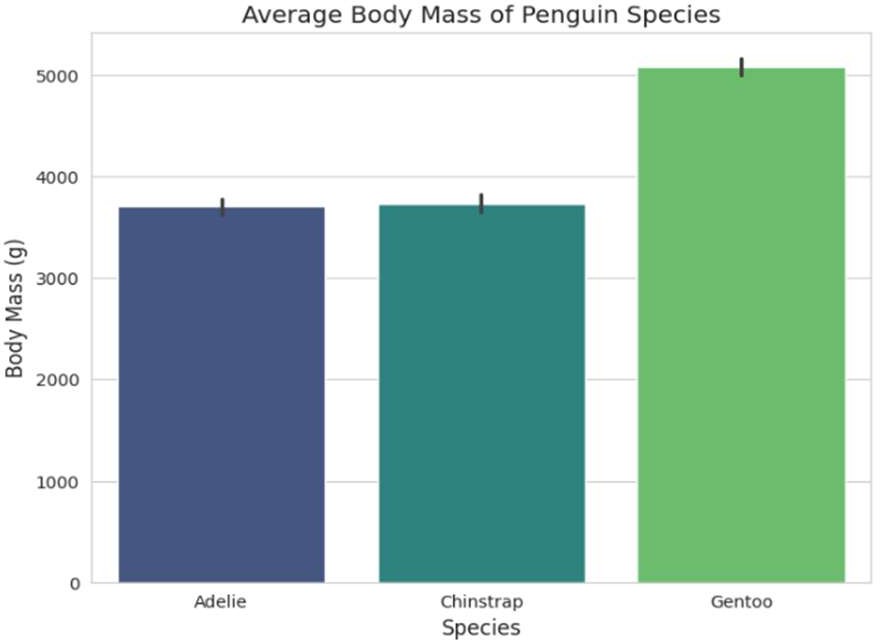
sns.violinplot(x="species", y="bill\_length\_mm", data=df, palette="Set2", hue="sex", split=True)

plt.title("Bill Length Distribution by Species and Sex", fontsize=14) plt.xlabel("Species", fontsize=12)

plt.ylabel("Bill Length (mm)", fontsize=12) plt.legend(title="Sex")

plt.show()

## Implementation:

****

**Result:**

Successfully applied **various Seaborn color palettes** and **customized visualization aesthetics** using styles, figure sizes, and legend placements.

# Advanced Analytics: Regression, Correlation, and Covariance

## Aim:

To analyze relationships between variables using **regression plots, correlation heatmaps, and covariance matrices** in Seaborn.

## Procedure:

1. Install **Seaborn**, **Matplotlib**, and **Pandas** if not installed.
2. Import required libraries.
3. Load a dataset containing numerical values for correlation and regression analysis.
4. Create different **advanced analytics visualizations**:
   * **Regression Plot (regplot)**
   * **Correlation Heatmap (heatmap)**
   * **Pair Plot with Regression Lines (pairplot)**
   * **Covariance Matrix (calculated with Pandas)**

## Code:

# Step 1: Import required libraries import seaborn as sns

import matplotlib.pyplot as plt import pandas as pd

# Step 2: Load a sample dataset

df = sns.load\_dataset("penguins") # Built-in dataset in Seaborn

# Step 3: Drop missing values (important for correlation analysis) df = df.dropna()

# Step 4: Regression Plot - Relationship between Bill Length & Bill Depth plt.figure(figsize=(8, 6))

sns.regplot(x="bill\_length\_mm", y="bill\_depth\_mm", data=df, scatter\_kws={'color':'blue'}, line\_kws={'color':'red'}) plt.title("Regression: Bill Length vs Bill Depth", fontsize=14) plt.xlabel("Bill Length (mm)", fontsize=12)

plt.ylabel("Bill Depth (mm)", fontsize=12) plt.show()

# Step 5: Correlation Heatmap - Understanding variable relationships plt.figure(figsize=(8, 6))

sns.heatmap(df.iloc[:,2:6].corr(), annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)

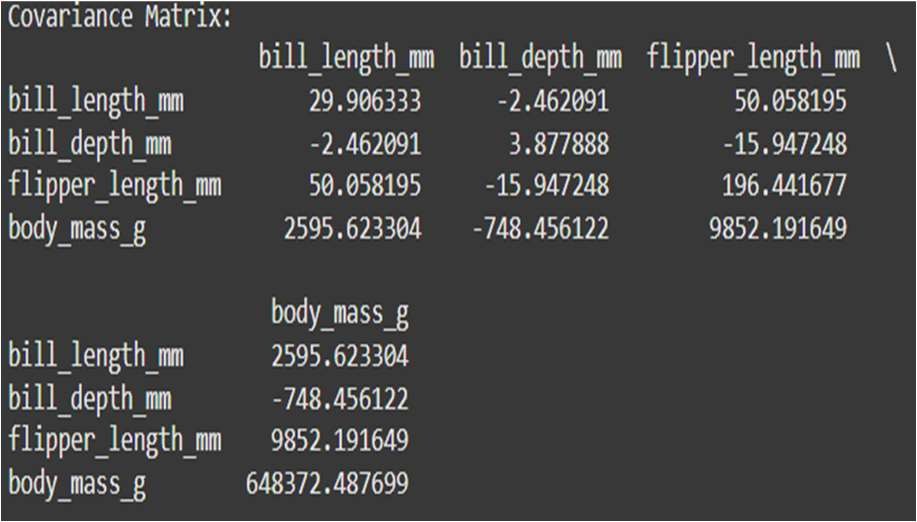
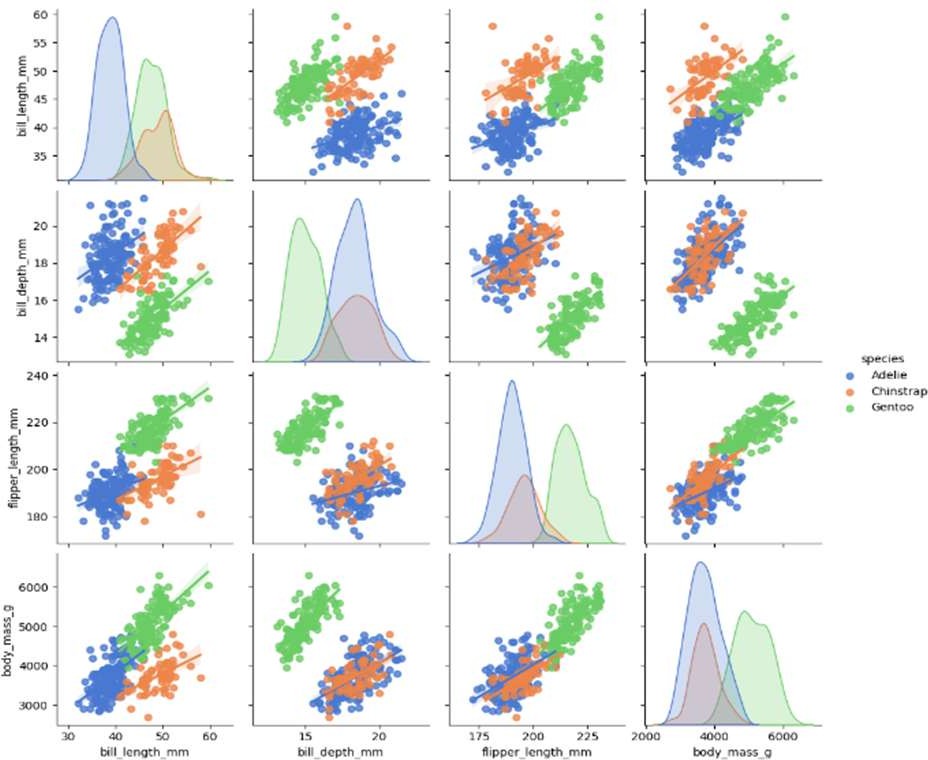
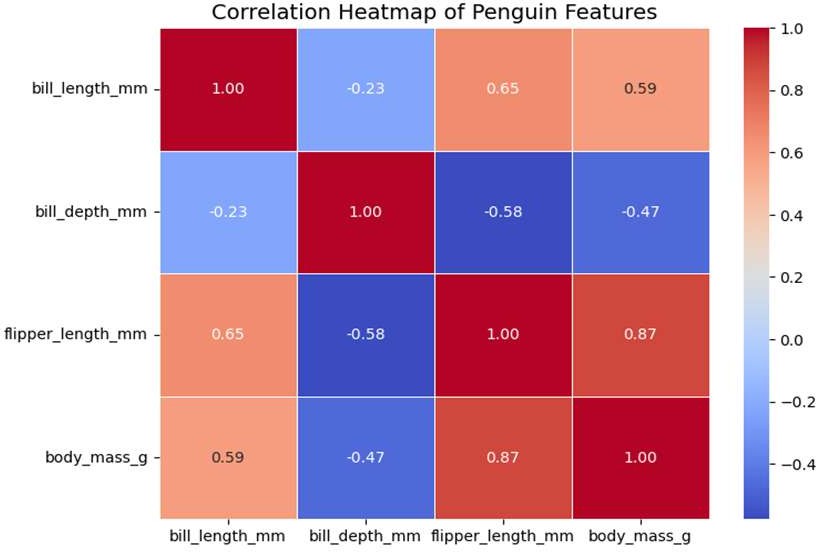
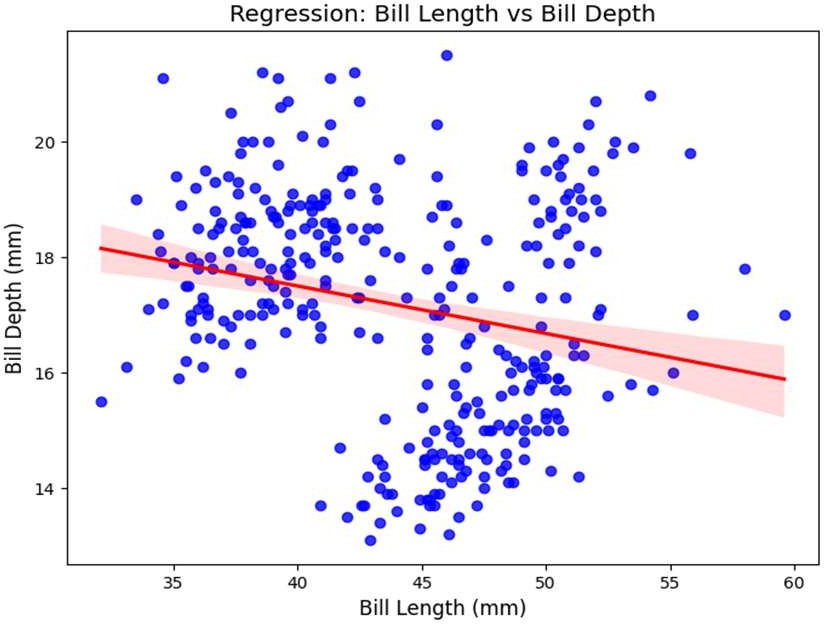
plt.title("Correlation Heatmap of Penguin Features", fontsize=14) plt.show()

# Step 6: Pair Plot - Relationships between multiple variables with regression lines

sns.pairplot(df, kind="reg", hue="species", palette="muted") plt.show()

# Step 7: Covariance Matrix Calculation cov\_matrix = df.iloc[:,2:6].cov() print("Covariance Matrix:\n", cov\_matrix)

## Implementation:

****

**Result:**

Successfully performed **regression analysis, correlation visualization, and covariance computation** using Seaborn, helping in data-driven insights.

# Plotting and Configuring Aesthetics in Seaborn

## Aim:

To explore different ways of customizing Seaborn visualizations using themes, font styles, grid settings, and figure adjustments for better readability.

## Procedure:

1. Install **Seaborn** and **Matplotlib** if not installed.
2. Import required libraries.
3. Load a dataset for visualization.
4. Customize aesthetics by:
   * Changing **Seaborn themes**
   * Adjusting **font size and style**
   * Configuring **grid lines and figure size**
   * Modifying **axes labels and titles**
   * Using **despine()** to remove unnecessary borders

## Code:

# Step 1: Import required libraries import seaborn as sns

import matplotlib.pyplot as plt

# Step 2: Load a sample dataset

df = sns.load\_dataset("penguins") # Built-in dataset in Seaborn

# Step 3: Set a Seaborn theme for better aesthetics sns.set\_theme(style="darkgrid") # Options: whitegrid, darkgrid, white, dark, ticks

# Step 4: Customize font size, style, and figure size plt.figure(figsize=(8, 6))

sns.set\_context("talk") # Options: paper, notebook, talk, poster

# Step 5: Create a Histogram with custom aesthetics sns.histplot(df["body\_mass\_g"], bins=30, kde=True, color="purple", edgecolor="black")

plt.title("Penguin Body Mass Distribution", fontsize=14, fontweight="bold") plt.xlabel("Body Mass (g)", fontsize=12)

plt.ylabel("Count", fontsize=12) plt.show()

# Step 6: Create a Scatter Plot with Grid Customization plt.figure(figsize=(8, 6))

sns.scatterplot(x="flipper\_length\_mm", y="bill\_length\_mm", data=df, hue="species", palette="Set1", s=100)

plt.title("Flipper Length vs Bill Length", fontsize=14) plt.xlabel("Flipper Length (mm)", fontsize=12) plt.ylabel("Bill Length (mm)", fontsize=12)

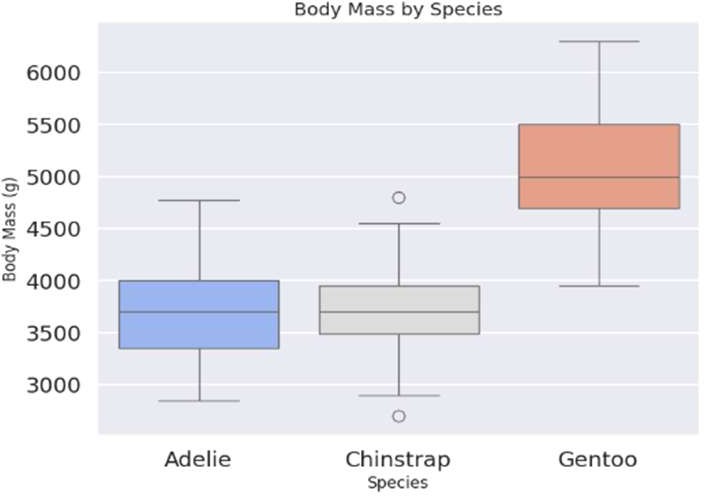
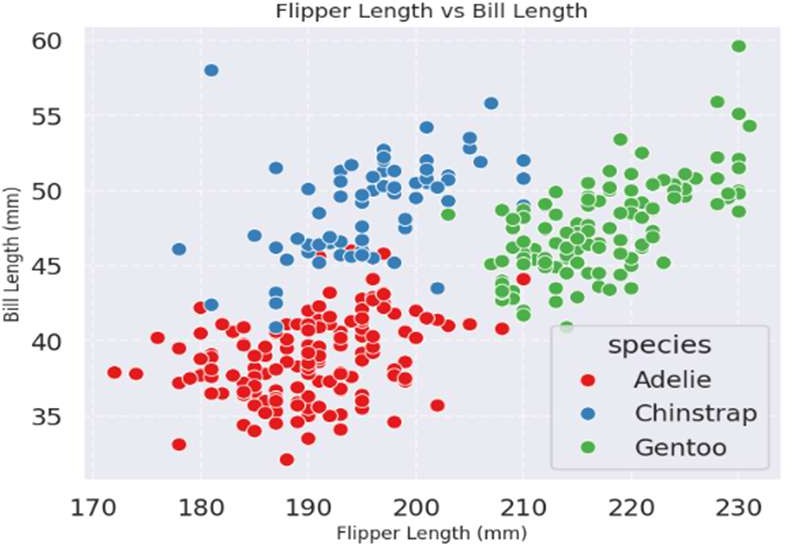
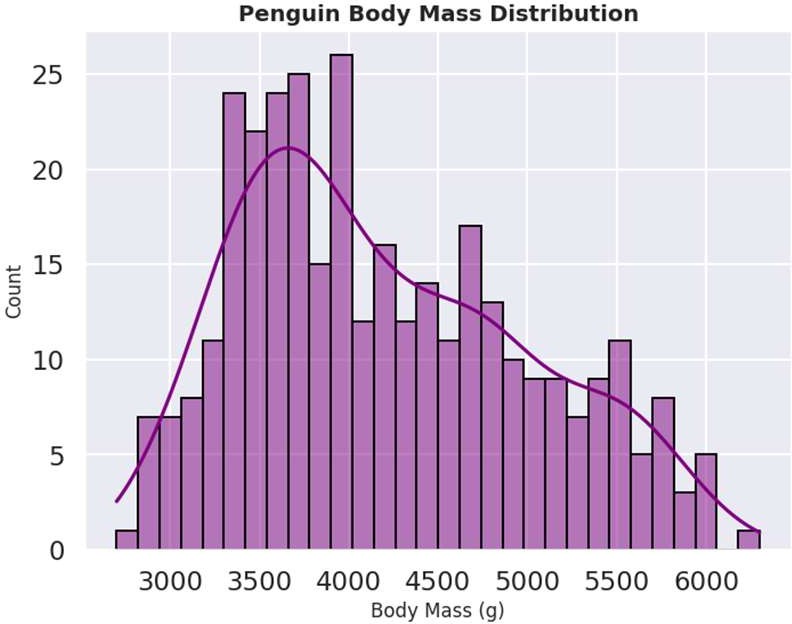
plt.grid(True, linestyle="--", alpha=0.5) plt.show()

# Step 7: Remove unnecessary borders using despine() plt.figure(figsize=(8, 6))

sns.boxplot(x="species", y="body\_mass\_g", data=df, palette="coolwarm") sns.despine(left=True, bottom=True) # Removes left and bottom spines plt.title("Body Mass by Species", fontsize=14)

plt.xlabel("Species", fontsize=12) plt.ylabel("Body Mass (g)", fontsize=12) plt.show()

## Implementation:

****

**Result:**

Successfully customized Seaborn visualizations using **themes, fonts, grid styles, and axis formatting** to enhance clarity and aesthetics.

# Customizing Matplotlib Visualizations (Aesthetics, Labels, Legends, Colors)

## Aim:

To customize **Matplotlib** visualizations by adjusting aesthetics, labels, legends, colors, and grid settings for better data representation.

## Procedure:

1. Install **Matplotlib** if not installed.
2. Import required libraries.
3. Load a dataset or generate sample data.
4. Create different **customized visualizations**:
   * **Line Plot with Labels and Grid**
   * **Bar Chart with Custom Colors**
   * **Scatter Plot with Annotations**
   * **Pie Chart with Explode Effect**

## Code:

# Step 1: Import required libraries import matplotlib.pyplot as plt import numpy as np

# Step 2: Generate sample data x = np.linspace(0, 10, 100)

y1 = np.sin(x) y2 = np.cos(x)

# Step 3: Create a Line Plot with Customization plt.figure(figsize=(8, 6))

plt.plot(x, y1, label="Sine Wave", color="blue", linestyle="--", linewidth=2)

plt.plot(x, y2, label="Cosine Wave", color="red", linestyle="-.", linewidth=2)

plt.title("Sine and Cosine Waves", fontsize=14, fontweight='bold') plt.xlabel("X values", fontsize=12)

plt.ylabel("Y values", fontsize=12) plt.legend(loc="upper right") plt.grid(True, linestyle="--", alpha=0.5) plt.show()

# Step 4: Create a Bar Chart with Custom Colors categories = ["A", "B", "C", "D", "E"]

values = [23, 45, 56, 78, 34]

colors = ['#ff9999','#66b3ff','#99ff99','#ffcc99','#c2c2f0']

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

plt.bar(categories, values, color=colors, edgecolor="black", linewidth=1.5)

plt.title("Category-wise Values", fontsize=14) plt.xlabel("Categories", fontsize=12) plt.ylabel("Values", fontsize=12)

plt.show()

# Step 5: Create a Scatter Plot with Annotations np.random.seed(0)

x = np.random.rand(10) \* 10 y = np.random.rand(10) \* 10

labels = ["Point " + str(i) for i in range(1, 11)]

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

plt.scatter(x, y, c="green", marker="o", s=100, edgecolor="black", alpha=0.7)

plt.title("Scatter Plot with Annotations", fontsize=14) plt.xlabel("X-axis", fontsize=12)

plt.ylabel("Y-axis", fontsize=12)

# Add annotations

for i, txt in enumerate(labels):

plt.annotate(txt, (x[i], y[i]), textcoords="offset points", xytext=(5,5), ha='right', fontsize=10, color="black")

plt.grid(True, linestyle="--", alpha=0.5) plt.show()

# Step 6: Create a Pie Chart with Custom Labels and Explode Effect labels = ['Apple', 'Banana', 'Grapes', 'Orange', 'Mango']

sizes = [25, 15, 30, 10, 20]

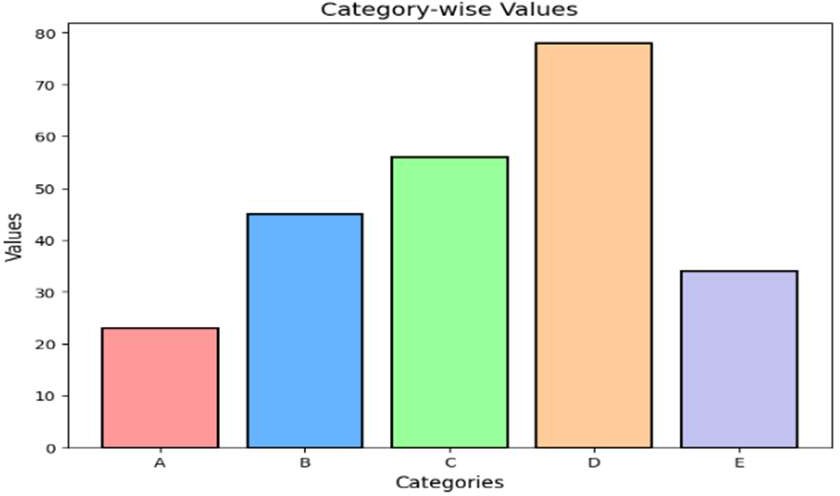
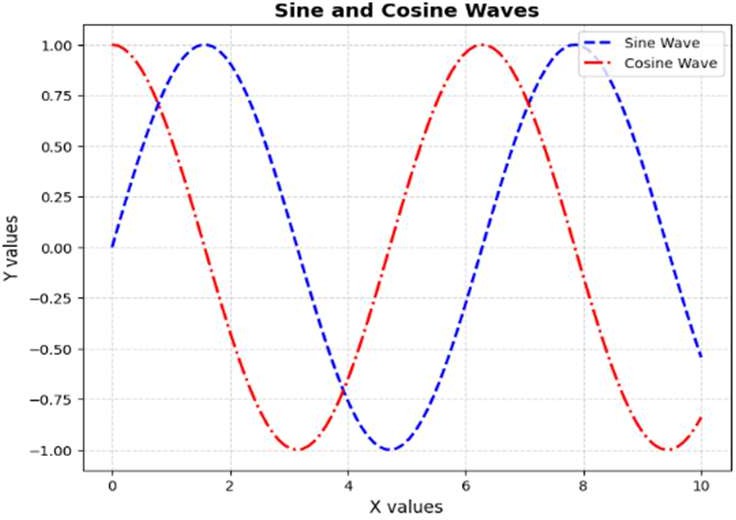
explode = (0, 0, 0.1, 0, 0) # Slightly separate the third slice

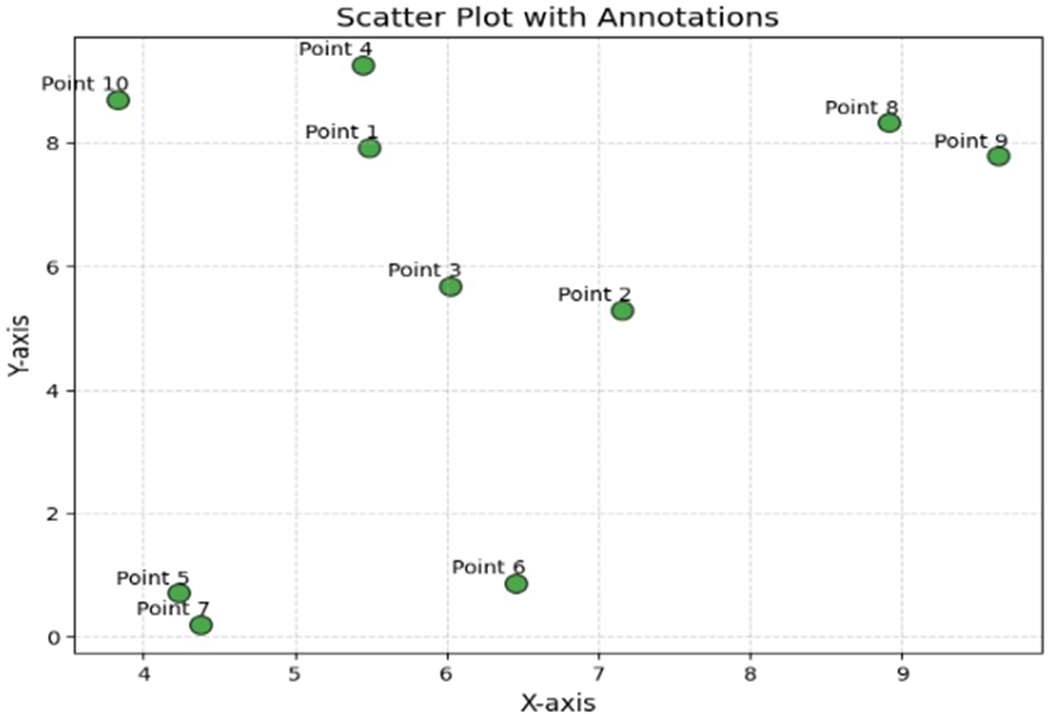
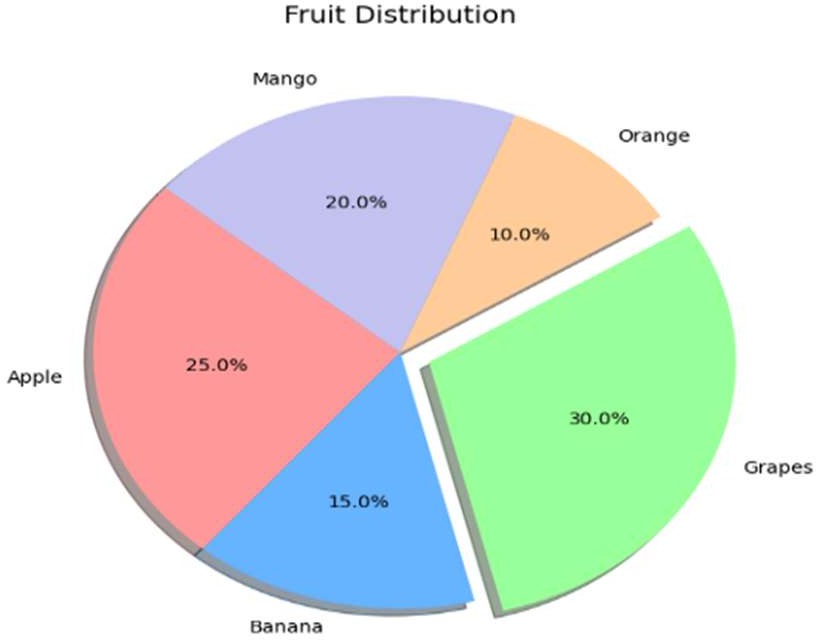
plt.figure(figsize=(7, 7))

plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=colors, explode=explode, startangle=140, shadow=True)

plt.title("Fruit Distribution", fontsize=14) plt.show()

## Implementation:

****

## Result:

Successfully customized **Matplotlib** visualizations using **labels, legends, colors, grid styling, and annotations** for enhanced presentation.