PCET-NMVPM's

Nutan College of Engineering and Research, Talegaon, Pune

Department of Computer Science & Engineering- Artificial Intelligence

ACAD-DI-96	INDEX	Academic Year: 2023-24
Rev : 01		Semester: I
Date: 24-01-2023		

Name of Student: Shreya Nitin Pingale

Class: BTech Division: ..... Branch: CSE-AI

Course with code: Data Visualization & Processing (BTAIIC702)

Sl.No	Sl.No	Particulars	Date of Performance	Date of Submission	Obtained marks out of 10	Staff Sign
1	1.1	Data visualization in Python	21/08/23	28/08/23	9	Ruyas
	1.2	connect dataset with Tableau Desktop				Ruyas
	1.3	Filtering & sorting of data	28/08/23	04/09/23	8	Ruyas
2	2.1	creating common visualizations	04/09/23	11/09/23		Ruyas
	2.2	Assembling dashboard layout using dashboard filters	11/09/23	18/09/23	9	Ruyas
3	3.1	Datariz Best Practices	18/09/23	25/09/23		Ruyas
	3.2	create Tableau dashboard with simple calcn using table calcn	25/09/23	09/10/23	8	Ruyas
4	4.1	create an interactive dashboard with text & visual tooltips by using drill down approach	09/10/23	16/10/23	9	Ruyas
	4.2	interactivity with text & visual tooltips, interactivity with actions, drilldown bet^{n} dashboards	16/10/23	23/10/23		Ruyas
5	5.1	create more advanced data visualization dashboards by using simple calcn fields using multiple tables	23/10/23	30/10/23	8	Ruyas
	5.2	Advanced visualization chart: Treemap, waterfall chart, motion chart.	30/10/23	20/11/23		Ruyas
6	6.1	Create data story in Tableau overview of the Tableau ecosym	20/11/23	22/11/23	9	Ruyas
		Design end-to-end dashboards with the help of story telling	20/11/23	22/11/23		Ruyas
	Total Marks Obtained ..... Out of					

Shreya.

Signature of Student

Ruyas

Course Coordinator

Experiment No: 01

Title: Introduction To Tableau

Aim: 1.1 Data Visualization in Python
1.2 Connect dataset with Tableau Desktop (display basic bar chart)
1.3 Filtering and Sorting of data

Lab Objective:

To Understand Tableau Desktop to Connect to various Datasets and prepare data for visualization in Tableau.

Theory:

1.1

Introduction to 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 represented in attractive way.
3. Shows complex relationships.
4. Helps to process large datasets.
5. Minimizes ambiguity.

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.

Basic Visualization in Python:

Basic visualization in Python is a fundamental skill for data analysis and presentation. Python offers several libraries for creating visualizations, with Matplotlib being one of the most popular and versatile options.

Some of the basic data visualization plots:

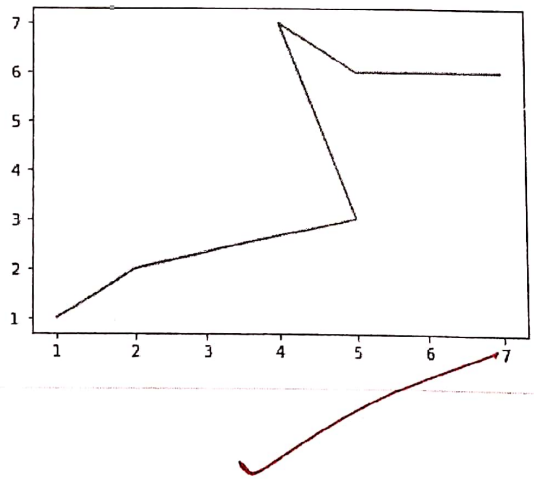
1. Line Plots
2. Histogram
3. Bar Chart
4. Scatter Plots
5. Line charts
6. Matplotlib

1. Line Plots:

Line plots, also known as line charts, are used to visualize trends or changes in data over time or long a continuous scale. They are particularly helpful for displaying data with a sequential or time-based relationship. Line plots connect data points with straight lines, making it easy to observe patterns, fluctuations, or trends. These plots are commonly used in fields like finance to show stock price movements, in scientific research to illustrate experimental results over time, or in weather forecasting to display temperature variations. Line plots provide a clear and intuitive way to understand how a variable evolves, making them a valuable tool for data analysis and communication.

Program and Output:-

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



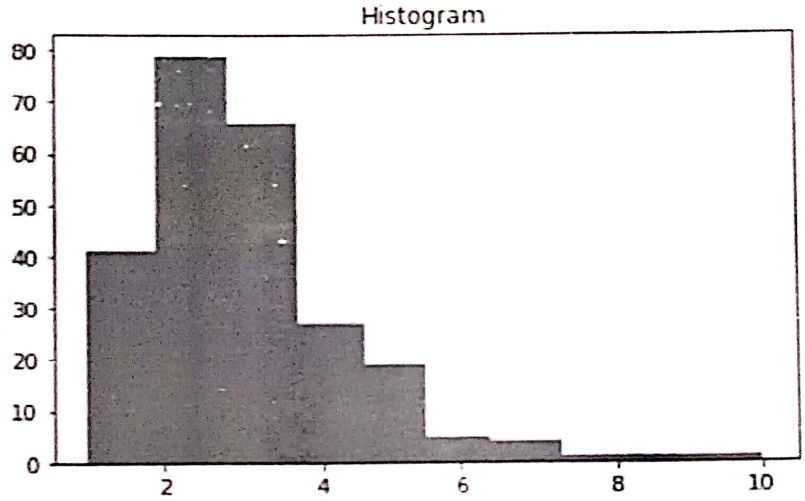
1. Histograms:

Histograms are graphical representations of the distribution of a continuous dataset. They divide the data into intervals or "bins" along the X-axis and display the frequency or count of data points falling into each bin on the Y-axis. Histograms are particularly useful for understanding the underlying distribution of data, whether it's normally distributed, skewed, or has multiple modes. They allow you to identify patterns, outliers, and the central tendency of the data. Histograms are commonly used in statistics, data analysis, and quality control to explore and summarize data distributions, making them a fundamental tool for data exploration and hypothesis testing.

Program:

```
#Histogram
import matplotlib.pyplot as plt

data = pd.read_csv("tips.csv")

# histogram of total_bills
plt.hist(data['tip'])

plt.title("Histogram")

# Adding the legends
plt.show()
```

Output:



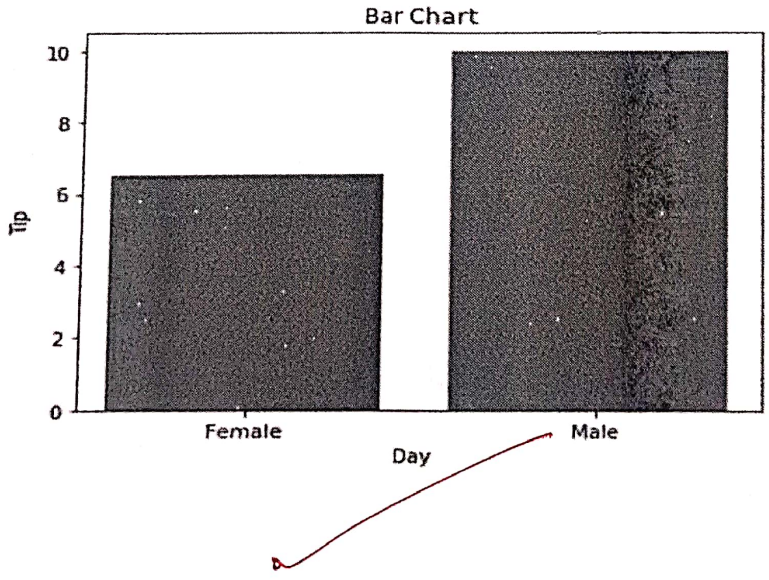
Bar Charts:

Bar charts are designed to compare categorical data or different groups within a dataset. They use rectangular bars, where the length of each bar is proportional to the value it represents. Bar charts are ideal for illustrating the relative sizes, frequencies, or counts of different categories or groups. (column charts) or horizontal (bar charts), depending on the orientation of the bars. They provide a straightforward and effective way to make comparisons among discrete categories and convey information clearly to a wide range of audiences.

Program:

```
#Bar Chart

import matplotlib.pyplot as plt
plt.bar(data['sex'], data['tip'])
plt.title("Bar Chart")
plt.xlabel('Day')
plt.ylabel('Tip')
plt.show()
```

Output:



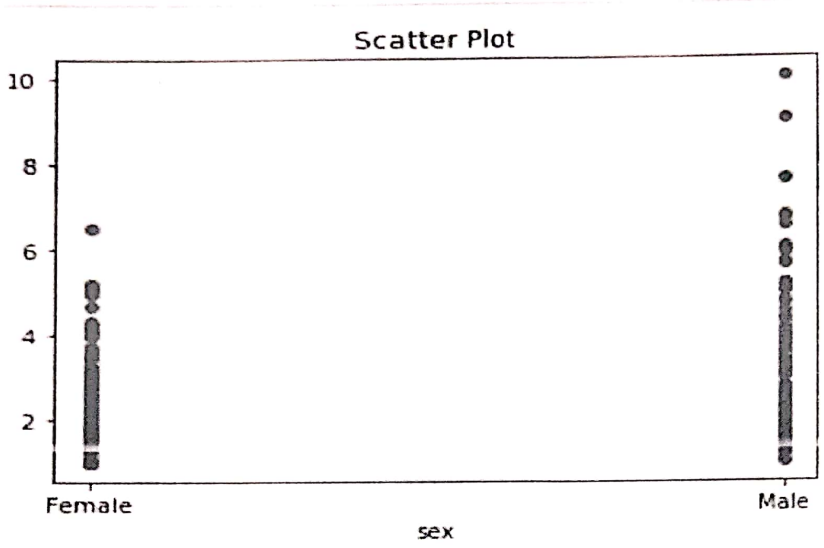
Scatter Plots

Scatter plots are used to visualize the relationship between two continuous variables. Each data point is represented as a point on the plot, with one variable plotted on the X-axis and the other on the Y-axis. Scatter plots are particularly useful for identifying patterns, trends, correlations, oroutliers in data. They help assess the strength and direction of a relationship between variables. Scatter plots are widely used in scientific research, social sciences, and engineering to explore how two variables interact. They are especially effective for detecting linear or non-linear associations between variables, making them an essential tool for data analysis and hypothesis testing. Scatter plots provide valuable insights into the nature of the data and can aid in making informed decisions based on observed relationships.

Program:

```
]: #Scatter plot
   import matplotlib.pyplot as plt

   # Scatter plot with day against tip
   plt.scatter(data['sex'], data['tip'])

   # Adding Title to the Plot
   plt.title("Scatter Plot")

   # Setting the X and Y Labels
   plt.xlabel('sex')
   plt.ylabel('tip')

   plt.show()
```

Output:

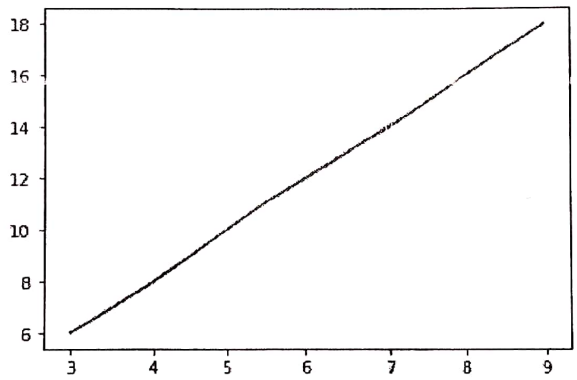


Line charts:

A line chart, also known as a line plot or line graph, is a fundamental type of data visualization used to display data points connected by straight lines. It is primarily employed to depict trends, patterns, or changes in data over a continuous or sequential range, such as time or a numerical scale. In a line chart, the X-axis represents the independent variable, often ordered or timerelated, while the Y-axis represents the dependent variable's values. Data points are plotted at specific X and Y coordinates, and a line is drawn to connect these points, making it easier to observe the progression or fluctuations in the data.

Program and Output :

```
In [11]: # Line Chart

         import matplotlib.pyplot as plt
         import numpy as np

         #define data values
         x = np.array([4, 9, 3, 6, 7])

         y = x*2

         plt.plot(x, y)
         plt.show()
```



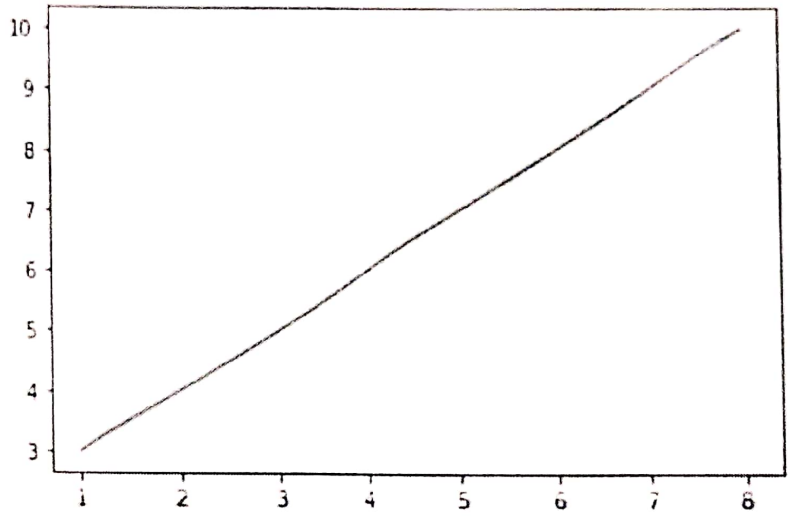
Matplotlib:

Matplotlib is a versatile and widely-used Python library for creating high-quality data visualizations. It provides a comprehensive set of tools for generating various types of plots, charts, and graphs, making it an essential component of data analysis and scientific computing. With Matplotlib, you can create static, animated, or interactive visualizations, making it suitable for a wide range of applications. It offers fine-grained control over every aspect of a plot's appearance, allowing users to customize colors, styles, labels, and annotations. Matplotlib's two main interfaces, the object-oriented API and the state-based pyplot interface, cater to different programming styles and skill levels. The pyplot interface simplifies the creation of basic plots, while the object-oriented API offers more flexibility for advanced customization.

Program:

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

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

plt.plot(xpoints, ypoints)
plt.show()
```

Output:



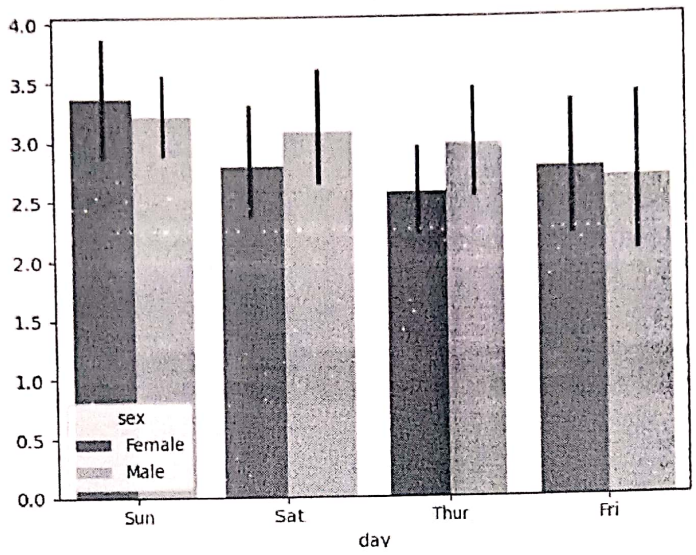
Seaborn:

Seaborn is a Python data visualization library built on top of Matplotlib, designed for creating informative and aesthetically pleasing statistical graphics. It simplifies the process of generating complex, visually appealing plots for exploring and understanding data. Seaborn provides a highlevel interface for creating common plot types like heat maps, pair plots, and violin plots with concise code. It also offers advanced features for statistical estimation and visualization, making it an excellent choice for data analysts and scientists. Seaborn is particularly known for its ability to enhance Matplotlib's default aesthetics and styles, resulting in visually striking and publicationready visualizations with minimal effort, thereby streamlining the data visualization process.

Program:

```
#seaborn
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd


# reading the database
data = pd.read_csv("tips.csv")

sns.barplot(x='day',y='tip', data=data,
            hue='sex')

plt.show()
```

Output:-



Theory:

1.2 (Connect dataset with Tableau Desktop (display basic bar chart)

Tableau offers several ways to connect to data sources, allowing users to visualize and analyze their data effectively. The specific methods and options available may vary depending on your Tableau version and data source. Here are some common ways to connect to data sources in Tableau:

1. File-based Connections:

Excel: You can connect to Excel files (.xls and .xlsx) by selecting the "Excel" option and choosing the file you want to import.

CSV: Tableau allows you to connect to Comma-Separated Values (CSV) files for data analysis.

2. Database Connections:

SQL Server: You can connect to Microsoft SQL Server databases.

MySQL: Tableau supports connecting to MySQL databases.

Oracle: Connect to Oracle databases for data analysis.

PostgreSQL: You can connect to PostgreSQL databases.

Amazon Redshift: For cloud-based data warehousing.

Google BigQuery: To analyze large datasets in Google's BigQuery data warehouse.

3. **Web Data Connectors (WDC):**

WDCs allow you to create custom connectors to web-based data sources that don't have a native Tableau connector. You can use JavaScript and HTML to create these connectors.

4. **Cloud Data Sources:**

Tableau supports cloud-based data sources such as Amazon S3, Google Cloud Storage, and Microsoft Azure Blob Storage.

5. **Web Data Sources:**

You can connect to web data sources using the Web Data Connector, which allows you to

6. **Data Warehouse Connections:**

Tableau can connect to data warehouses like Snowflake, Teradata, and IBM Db2.

7. **Cube Data Sources:**

You can connect to multi-dimensional data sources like Microsoft Analysis Services cubes.

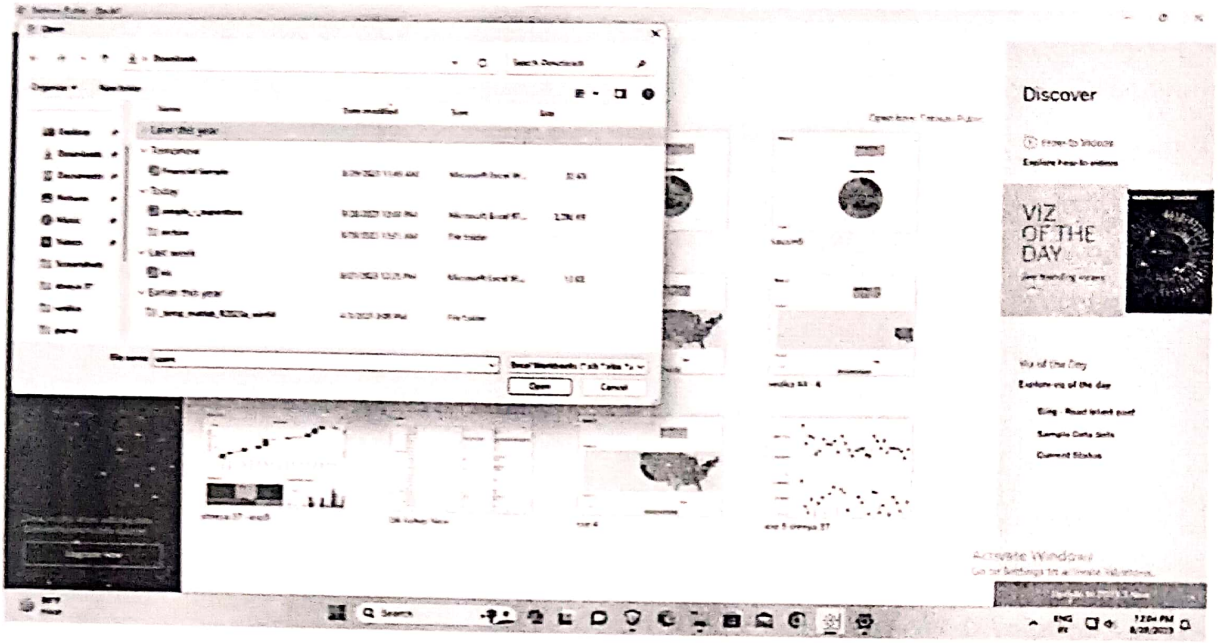
8. **Tableau Server and Tableau Online:**

You can publish data sources to Tableau Server or Tableau Online and then connect to them securely from Tableau Desktop or Tableau Prep.

9. **API Integration:**

Tableau provides APIs (e.g., REST API, JavaScript API) that allow you to programmatically interact with Tableau Server and embed Tableau views into other applications.

Output:



Theory:

1.3(Filtering and sorting of data)

Filtering and sorting are essential techniques in Tableau for data visualization and analysis. They help you control what data is displayed and how it is presented in your visualizations. Here's how filtering and sorting work in the context of data visualization in Tableau:

Filtering in Tableau:

1. Quick Filters: Quick filters are interactive controls that allow users to filter data on a dashboard or worksheet easily. You can add quick filters to your visualization to let users dynamicallycontrol which data they see.

2. Filter Dialogs: You can create more advanced filter conditions using filter dialogs. These allow you to filter data based on specific criteria such as date ranges, numerical ranges, text matching, etc.

3. Context Filters: Context filters are used to create a context for other filters. When you set a filter as a context filter, it affects the data visible to all subsequent filters, helping you control the order of filtering and improve performance.

4. Top N Filters: You can use Top N filters to focus on the top or bottom N items based on a measure, such as the top 10 products by sales.

5. Dimension Filters: These filters allow you to filter data based on categorical variables (dimensions), such as filtering by product category or location.

6. Measure Filters: Measure filters let you filter data based on quantitative variables (measures), such as filtering by sales revenue or profit margin.

7. Set Filters: Sets are custom subsets of data. You can filter data using set filters, which allow you toselect data points that are part of a particular set.

Sorting in Tableau:

a. Default Sorting: By default, Tableau often sorts data based on the order of dimensions or measures in the view. You can change this default sorting by clicking on the field in the view and choosing the desired sorting order (ascending or descending).

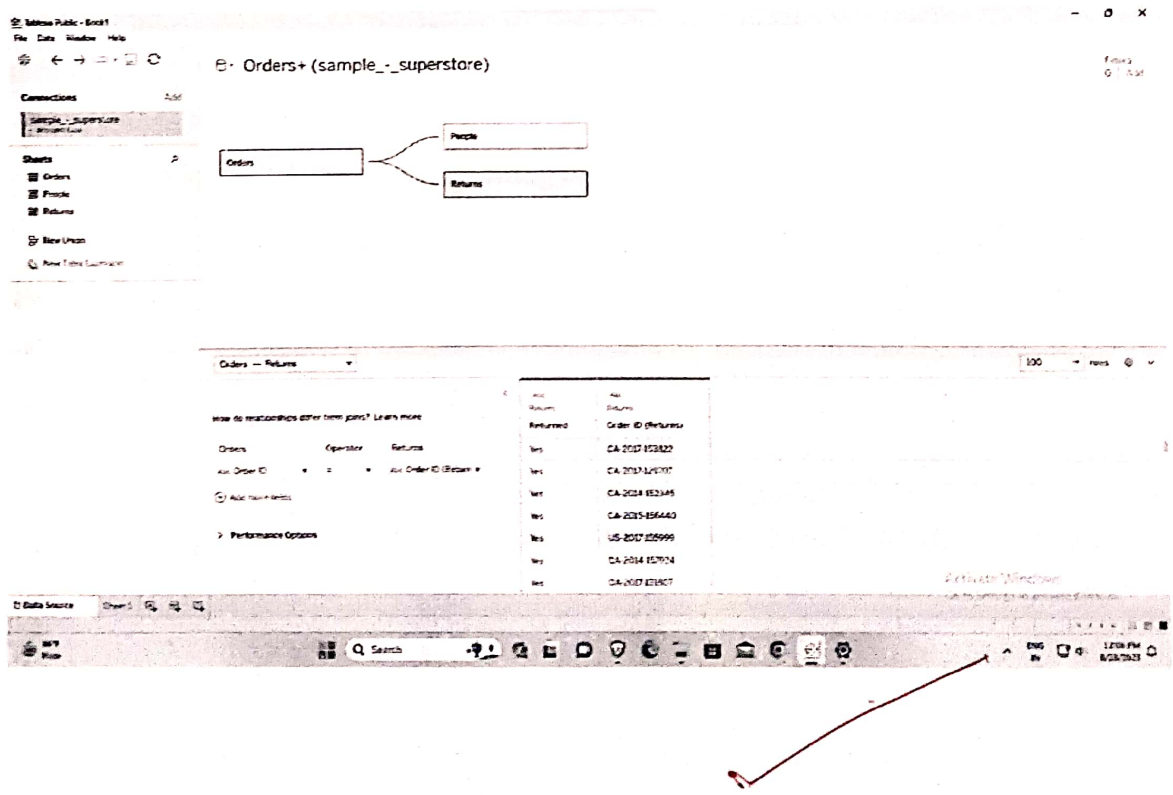
b. Manual Sorting: You can manually reorder items in visualization by dragging and dropping theminto the desired order, especially useful when creating custom hierarchies.

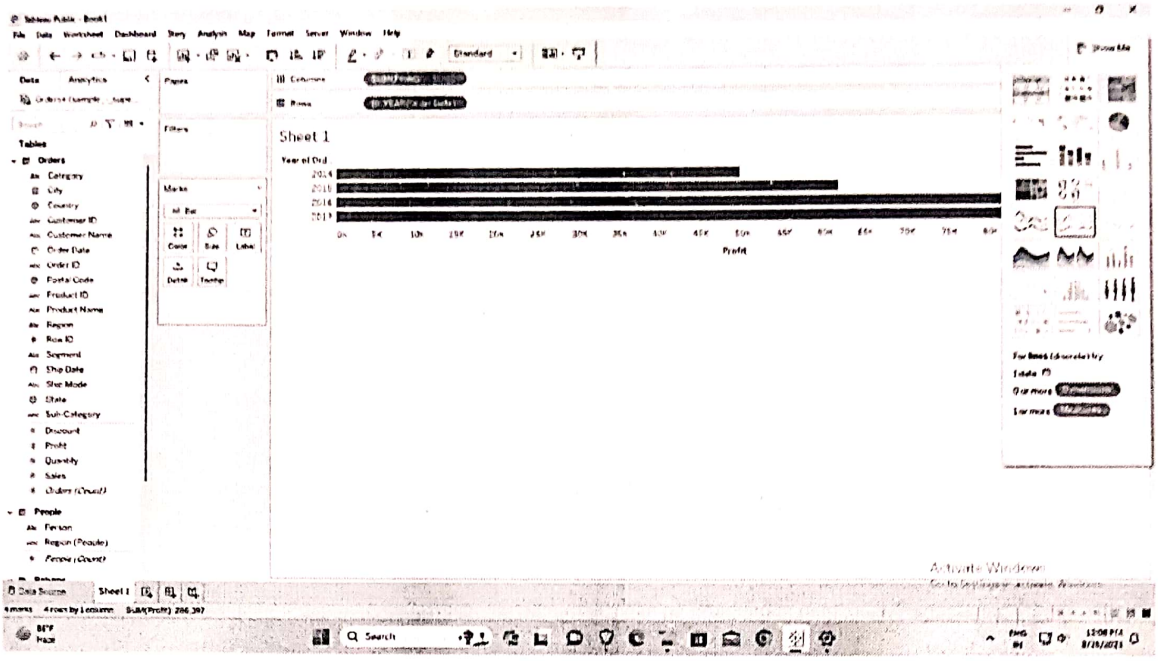
c. Sort Dialog: For more advanced sorting options, you can use the sort dialog. This dialog lets you sort by a specific dimension or measure, and you can control the sorting direction, sort by field, or even create a custom sort order.

d. Sort Buttons: You can add sort buttons to your worksheets or dashboards, allowing users to changethe sorting order dynamically.

e. Combined Sorting: In some cases, you may want to sort data by multiple fields simultaneously.Tableau allows you to combine sorting on dimensions and measures.

Outputs:





Lab Outcome:

Introduction to Tableau Desktop and its installation which connects to various datasets to derive effective data visualization techniques with filtering and sorting of data.

Conclusion:

Thus we understood the introduction to the tableau desktop and its installation which connects to various datasets to derive effective data visualization techniques with filtering and sorting of data.

Experiment No: 2

Title: Common Chart

Aim: 2.1 Creating common visualizations (bar chart, line chart, scatter plot, area chart)

2.2 Assembling dashboard layout using dashboard filters .

Objective: To Create a Tableau dashboard by using various visualization techniques using dashboard filters.

Theory:

2.1

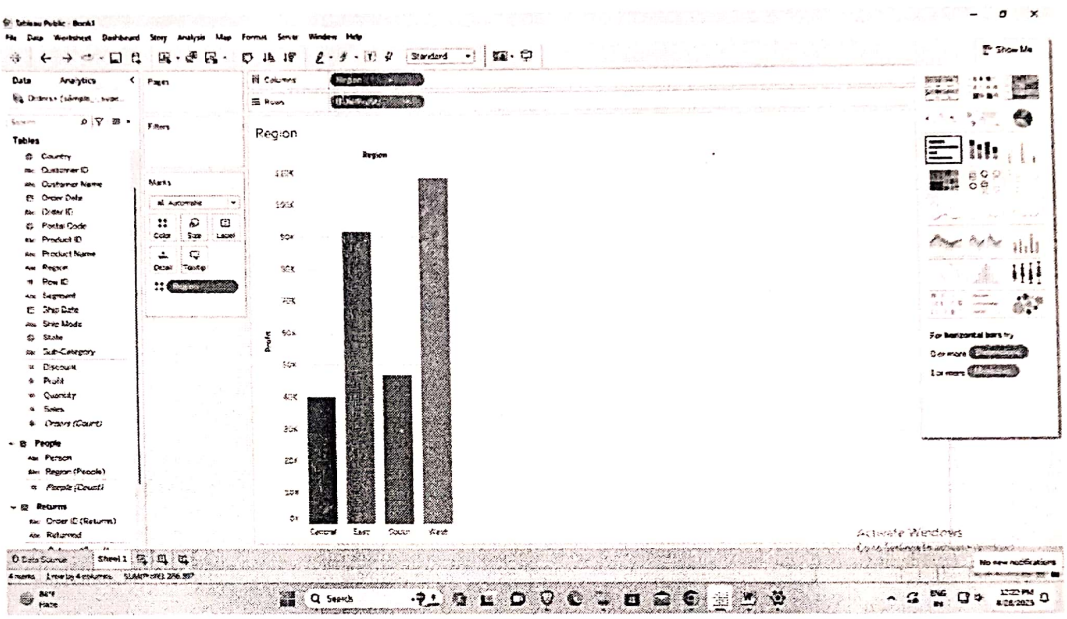
Charts in Tableau help professionals to an exponential level in the field of business intelligence and analytics. The real-time information can be directly represented using a wide variety of readily available and user-readable charts. Implementing data-oriented charts or graphs in your professional life can make result generation efficient for you. While charts are meant to represent an extensive set of information into diagrams, graphs, or tables, a graph displays mathematical relations between various data sets.

Tableau Charts List:

Bar Chart

Bar Charts are one of the standard methods of data visualization across all platforms. It is meant to represent the data in the form of bars. The bar length is proportional to the value of the variable. You can instantly highlight the difference between outliers, show trends, and categories. Also, you can reveal historical lows and highs at a glance. Here we are displaying quantity of Products by Category/Sub Category for the superstore dataset. The bar chart makes it easy to differentiate between highest and lowest quantity of products. The height of the bars give a clear representation for each product category.

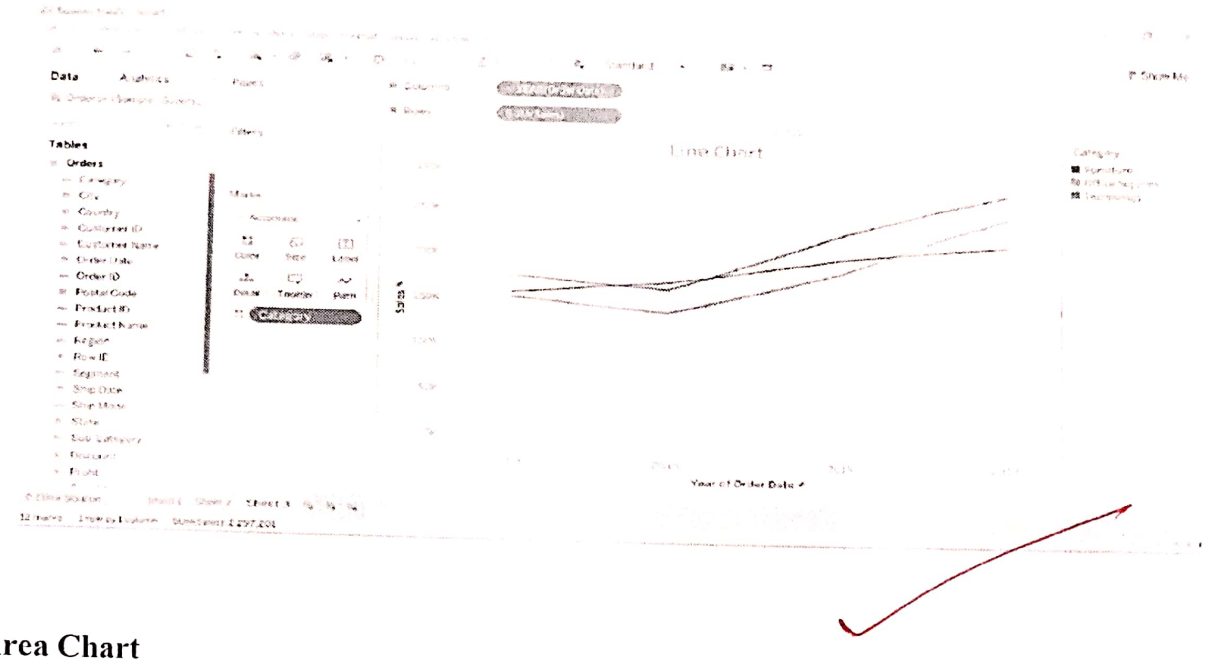
Output:-



Line Chart

Line Graphs, or Line Charts, are majorly used in combining individual points in a comprehensive sequence. It helps connect a variety of data points to display them as one consistent evolution. The result is a straightforward, simple way that visualizes changes in one value that is relative to another.

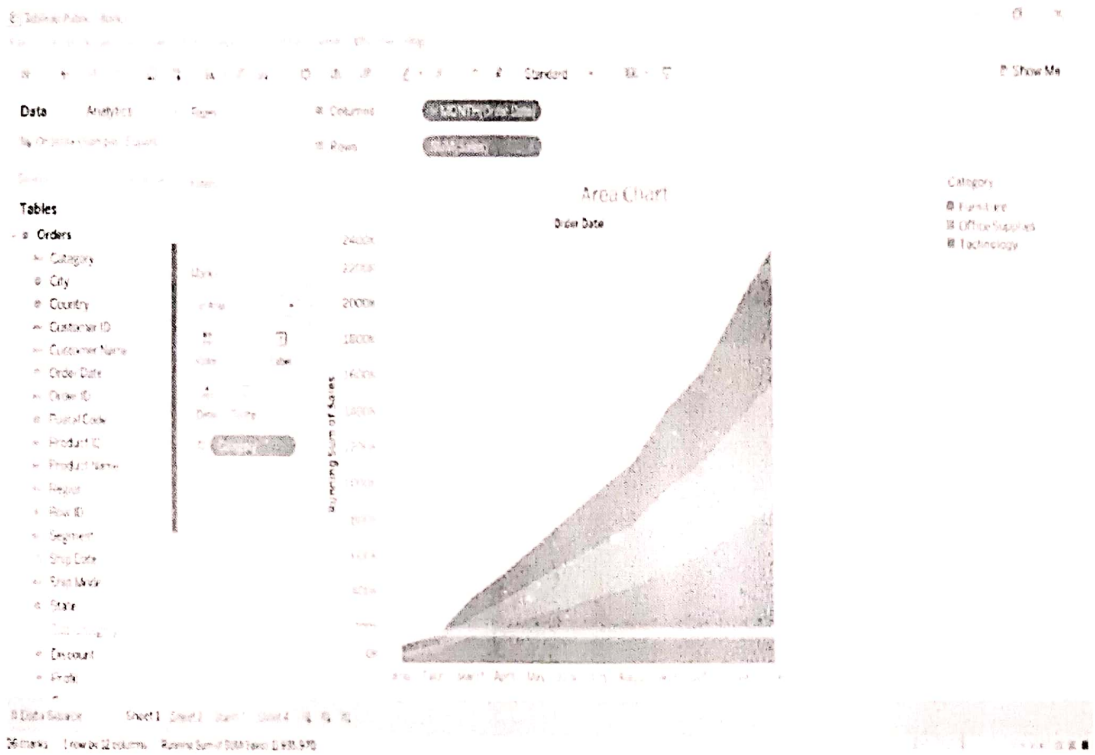
Output:-



Area Chart

Area chart helps represent quantitative data over a variety of periods. An area chart is almost the same as a line chart. What makes them stand apart is the partitions that are done on the basis of nations, regions, or categories in the area chart. Here we are displaying the sales for each category of products (Furniture, Office Supplies, and Technology) by year of order date. Area charthelps to understand the magnitude of sales.

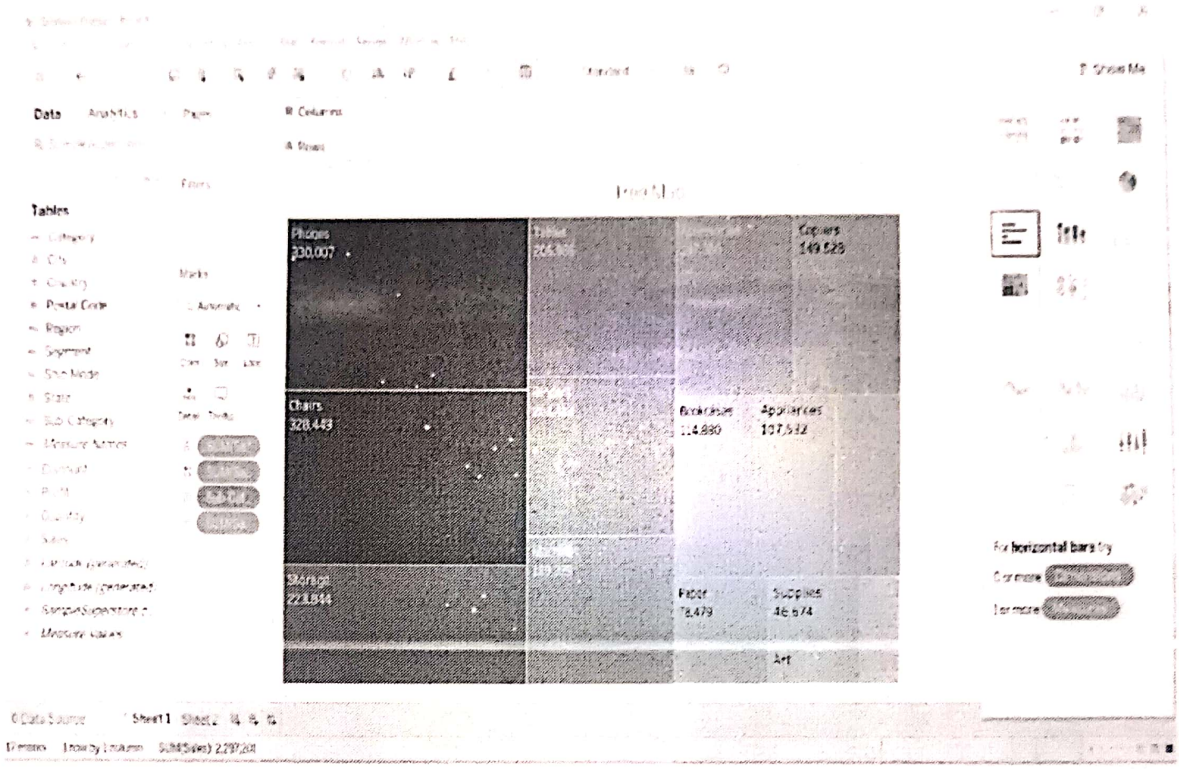
Output:-



Tree Map

A Tree map is used to demonstrate different parts of the data as they relate to the whole. Each rectangle in a Tree map is divided into smaller rectangles based on its proportions to the whole data like branches of a tree. So the total Tree map demonstrates the whole data while individual rectangles show the sub-data in proportion to the whole. Here we are displaying the ship mode and quantity of products segregated by region. There are 4 ship modes – Standard class, First class, Second class and same day.

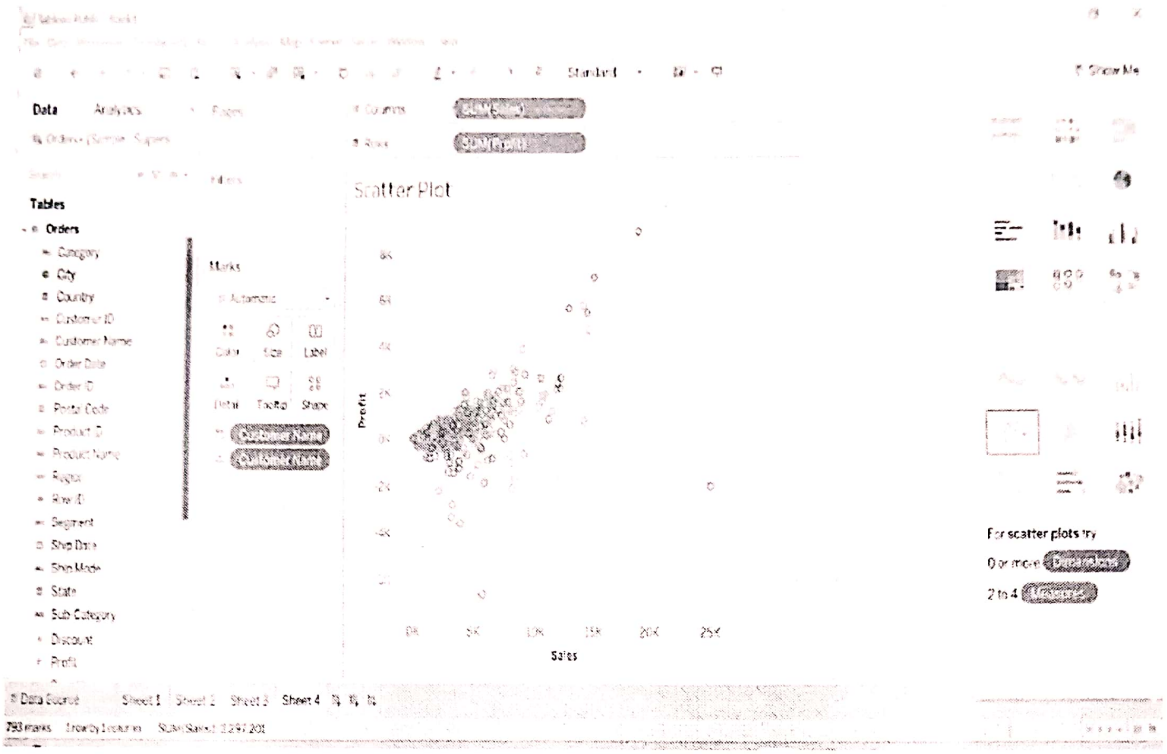
Output:-



Scatter Plot

Scatter plots are used to understand the relationship between two variables in the data. You can also find the outliers in your data or understand the overall distribution by plotting a Scatter plot. If the data moves from lower left to upper right, there might be a positive correlation between the two variables if the data move in the opposite direction, there might be a negative correlation. Here we are displaying the count of orders and profit by city. The scatter plot shows profit and count of orders on the same axis showing relationship between them country wise.

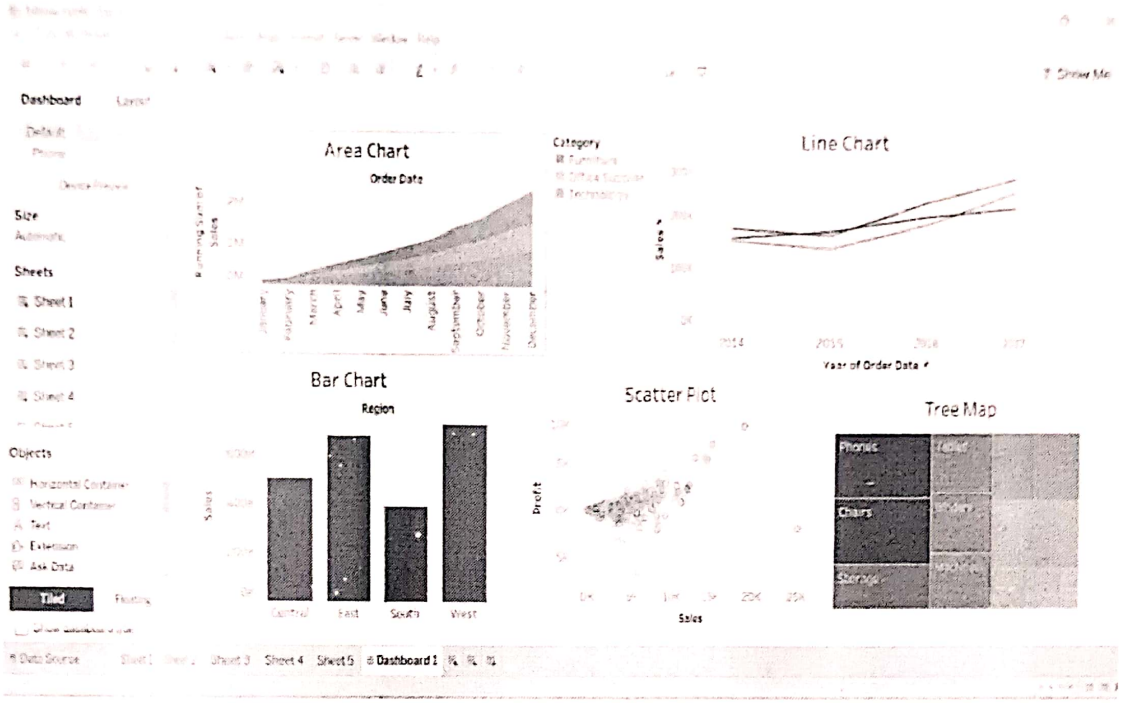
Output:-



2.2 Assembling dashboard layout using dashboard filters

A dashboard is a collection of several views, letting you compare a variety of data simultaneously. For example, if you have a set of views that you review every day, you can create a dashboard that displays all the views at once, rather than navigate to separate worksheets. Like worksheets, you access dashboards from tabs at the bottom of a workbook. Data in sheets and dashboards is connected; when you modify a sheet, any dashboards containing it change, and vice versa. Both sheets and dashboards update with the latest available data from the data source.

Output –



Lab Outcome:

Create a Tableau dashboard by using various visualization charts.

Conclusion:

This lab effectively demonstrated the creation of common data visualizations and the assembly of interactive dashboards with filters in Tableau. These skills enable users to explore an communicate data insights, fostering data-driven decision-making. The focus on user-centric design enhances the overall data analy

Experiment No 3

Title: Transform the data

Aim:

3.1 Data viz Best Practices

Draw a line chart which shows profit and sales by segment , state by quarterly and weekly. Calculate the Actual cost with quick table calculation and visualize it with discount

3.2 Create tableau dashboard with simple calculations using table calculations.

Lab Objective:

To build data model and create simple calculations in Tableau Using table calculations.

Theory:

3.1

Introduction to Data model:

A data model is a conceptual representation of data and the relationships between different pieces of data within a particular domain or system. It serves as a blueprint or framework for organizing, structuring, and managing data in a consistent and meaningful way. Data models are fundamental in database design, software development, and data management, as they help ensure data accuracy, consistency, and integrity.

Relationship between data model

One- to- one relationship

A one-to-one (1:1) relationship is a type of association between two entities in a data model where one record in one entity (table) is related to exactly one record in another entity, and vice other words,

for every record in the first entity, there is a single corresponding record in the second entity, and there are no duplicate relationships.

One- to- many relationship

A one-to-many (1:N) relationship is a fundamental type of association in a data model where one record in one entity (table) can be related to multiple records in another entity, but each record in the second entity can be related to only one record in the first entity. It is one of the most common types of relationships in database modeling.

Many-to-many relationship

A many-to-many (M:N) relationship is a type of association in a data model where multiple records in one entity (table) can be related to multiple records in another entity, and vice versa. In this relationship, there is no strict limit on the number of related records on either side. It is a common type of relationship in database modeling and often requires an intermediary table to represent the relationship.

Table calculation:

A table calculation is a transformation you apply to the values in a visualization. Table calculations are a special type of calculated field that computes on the local data in Tableau. They are calculated based on what is currently in the visualization and do not consider any measures or dimensions that are filtered out of the visualization.

You can use table calculations for a variety of purposes, including:

- Transforming values to rankings
- Transforming values to show running totals
- Transforming values to show percent of total

- Table (down)

Computes down the length of the table and restarts after every partition.

- Table (across then down)

Computes across the length of the table, and then down the length of the table.

- Table (down then across)

Computes down the length of the table, and then across the length of the table.

- Pane (down)

Computes down an entire pane

- Pane (across then down)

Computes across an entire pane and then down the pane

- Pane (down then across)

Computes down an entire pane and then across the pane.

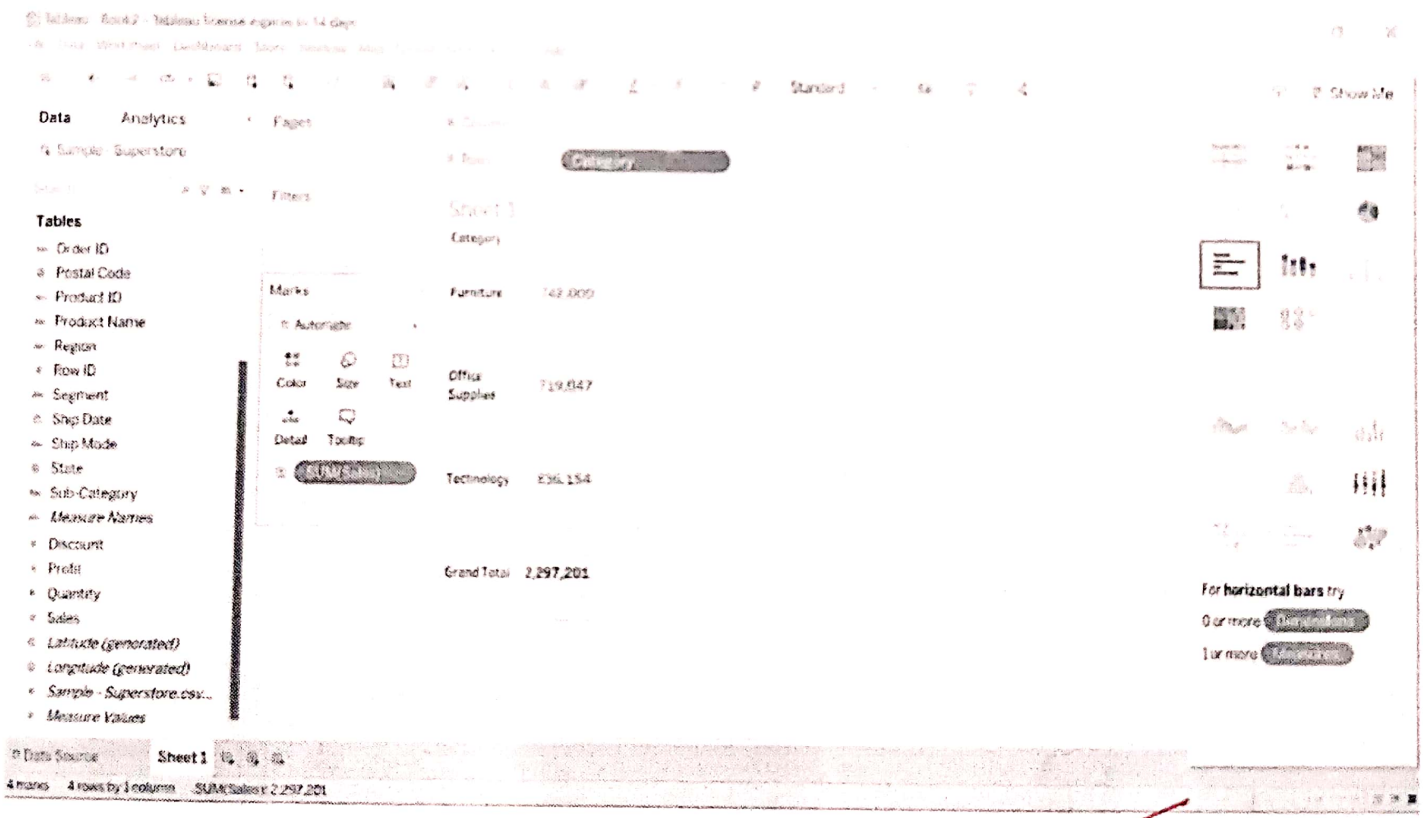
- Cell

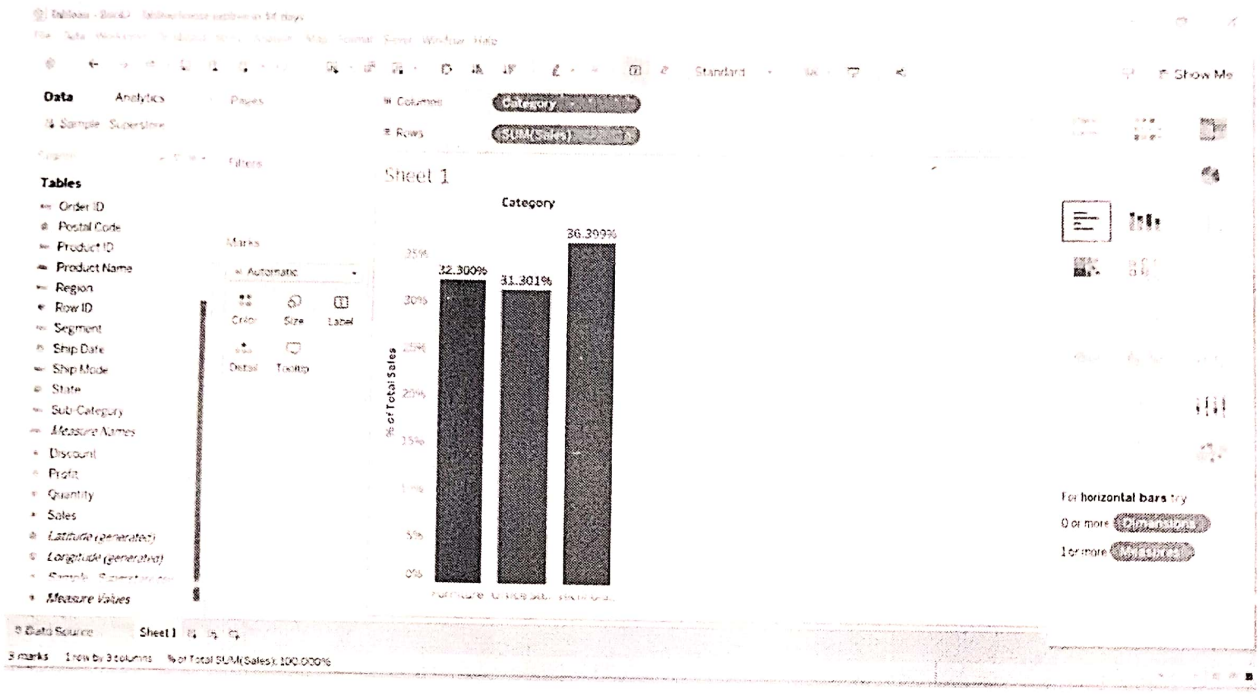
Computes within a single cell.

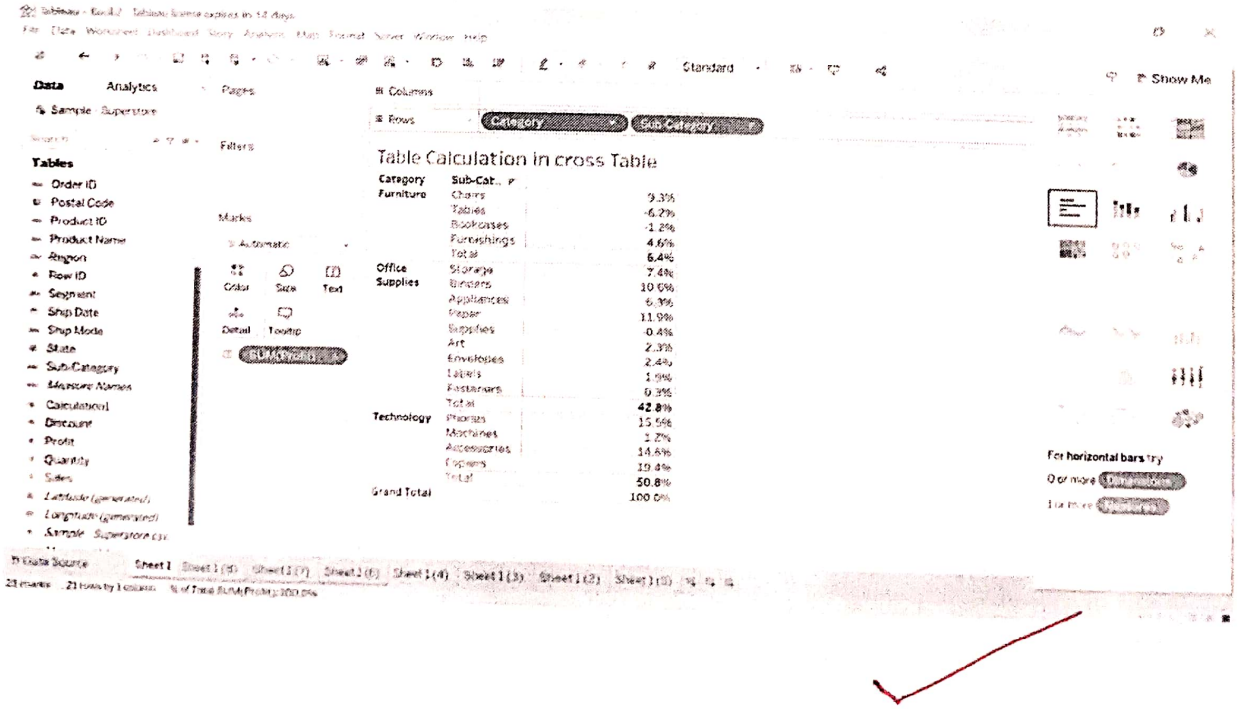
- Specific Dimensions

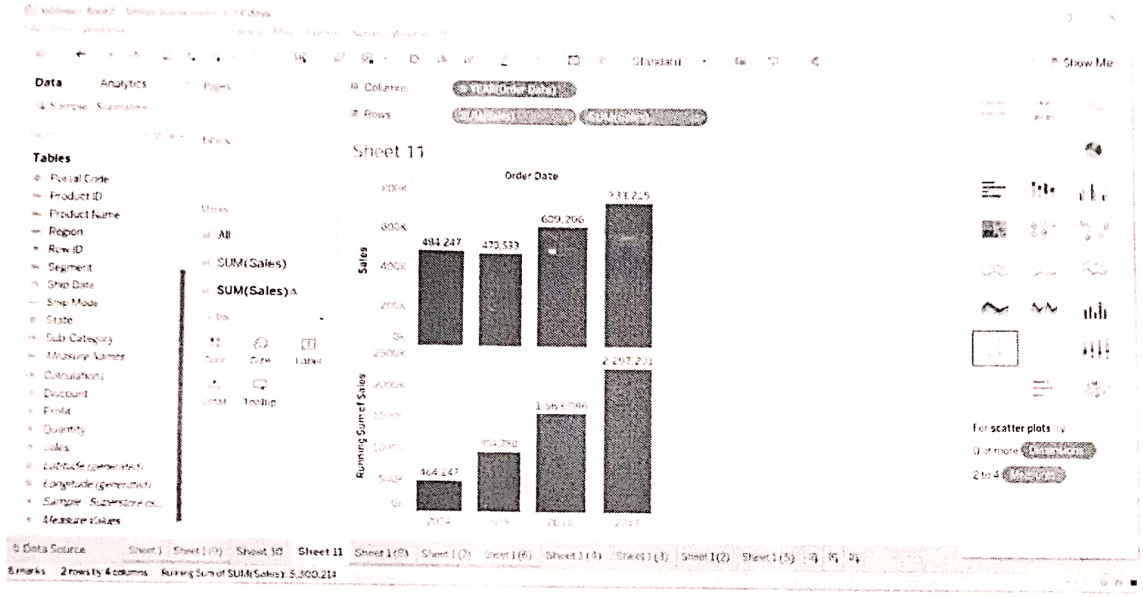
Computes only within the dimensions you specify

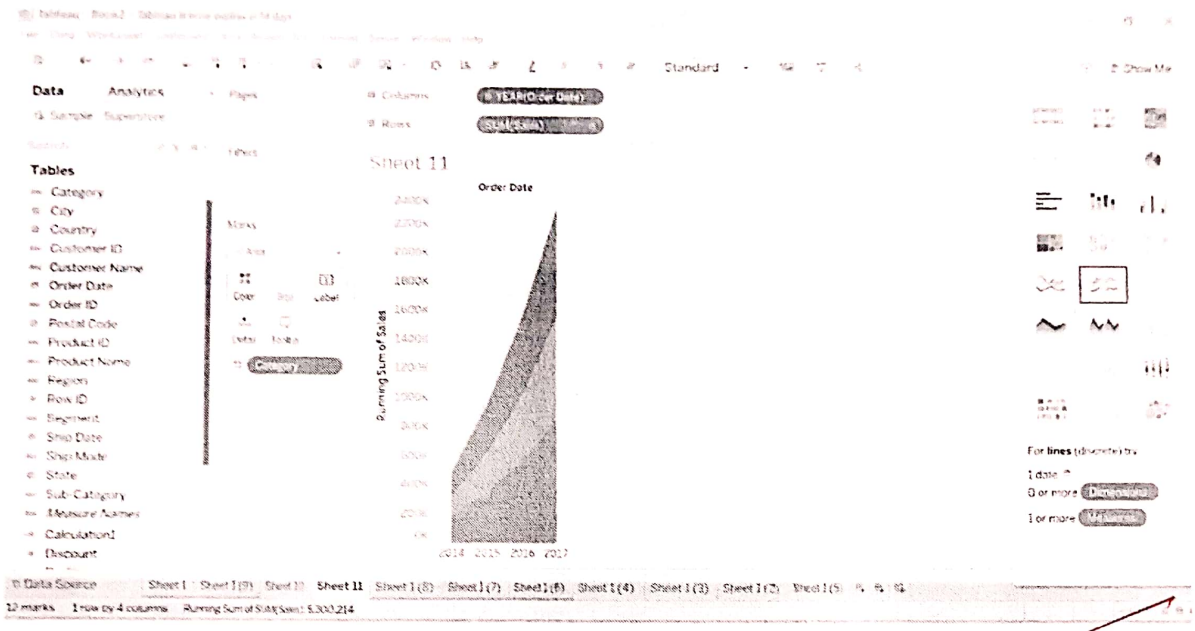
Output:











Lab Outcome:

Establish relationships between various data objects to build a data model and create table calculated data fields to calculate data across or down the table/pane/cell.

Conclusion:

Thus we understood how to establish relationships between various data objects to build a data model and create table-calculated data fields to calculate data across or down the table/pane/cell.

Experiment No 4

Title: Interactions

Aim: 4.1 Create an interactive dashboard with text and visual tooltips by using drill-down
4.2 Interactivity with text and visual tooltips ,Interactivity with actions (filter, highlight,URL),Drilldown between dashboards

Lab Objective: To build an interactive dashboard with text and visual tooltips, supporting actions like filtering and drill-down navigation between dashboards, enhancing the user experience and data exploration.

Theory:

4.1

Creating an interactive dashboard with text and visual tooltips using a drill-down approach involves designing a dynamic data visualization interface that enables users to explore information in a hierarchical manner. It typically commences with a high-level overview, presenting summary statistics or key performance indicators (KPIs) to offer users a top-level perspective of the data. To enhance user engagement and facilitate deeper data exploration, interactive elements are incorporated into the dashboard. These elements, such as clickable charts, graphs, or data points, serve as entry points for users to access more detailed information. Text and visual tooltips are a crucial component of this approach, providing users with additional context and insights when they hover over or click on specific data points. These tooltips may include descriptive text, supplementary statistics, or even pop-up visual aids, such as mini-charts. The drill-down navigation aspect allows users to delve into the data hierarchy by clicking on specific elements, initiating a transition from a broader category to finer details, progressively revealing more data as they navigate deeper into the dashboard. This hierarchical structure should be designed with

user-friendliness in mind, ensuring an intuitive interface that empowers users to explore data at their own pace and according to their specific interests. The ultimate aim of this approach is to engage users effectively and provide them with deeper insights, enabling them to answer questions and make data-driven decisions with ease and confidence. Certainly, creating an interactive dashboard with text and visual tooltips using a drill-down approach is a multifaceted process. Here's a more detailed breakdown:

High-Level Overview: Start with a high-level view of the data, such as summary statistics or key performance indicators (KPIs). This provides users with a bird's-eye view of the information.

Interactive Elements: Design the dashboard with interactive elements, such as clickable charts, graphs, or data points. Users can click on these elements to access more detailed information.

Text and Visual Tooltips: Implement tooltips that display relevant information when a user hovers over or clicks on a specific data point. Tooltips can include text descriptions, additional statistics, or even visual aids like pop-up charts.

Drill-Down Navigation: Allow users to drill down into the data by clicking on specific elements. For example, in a sales dashboard, clicking on a particular region could lead to a more detailed view for that region, including information on individual products or sales representatives.

Hierarchical Structure: Organize your data in a hierarchical structure, with levels of granularity. Users should be able to navigate from broader categories to finer details, revealing more data as they go deeper into the dashboard.

Interactivity with Text and Visual Tooltips:

Interactivity with text and visual tooltips is a critical feature in data visualization and dashboard design. It involves providing users with additional information and context by allowing them to hover over or click on data points, charts, or other elements within a dashboard.

Interactivity with Actions (Filter, Highlight, URL):

Interactivity with actions in a dashboard allows users to manipulate and interact with the data to extract meaningful insights and perform specific tasks. This includes features like filtering, highlighting, and linking to external resources through URLs.

1. Filtering: Users can apply filters to the data to narrow down what they are viewing, focusing on specific subsets or timeframes. Filters enable users to customize their data analysis and answer specific questions. For example, in an e-commerce dashboard, users might filter sales data to view results for a particular product category or location.
2. Highlighting: Highlighting lets users emphasize particular data points, making it easier to identify trends, outliers, or important information. It is a valuable tool for visual data analysis and storytelling within the dashboard.
3. URL Links: Incorporating URL links allows users to access external resources or websites directly from the dashboard. For instance, a dashboard tracking news articles can link to the full articles or related web pages for additional context and information.

These interactive actions empower users to interact with the data and customize their experience, making the dashboard a versatile tool for data exploration, analysis, and decision-making.

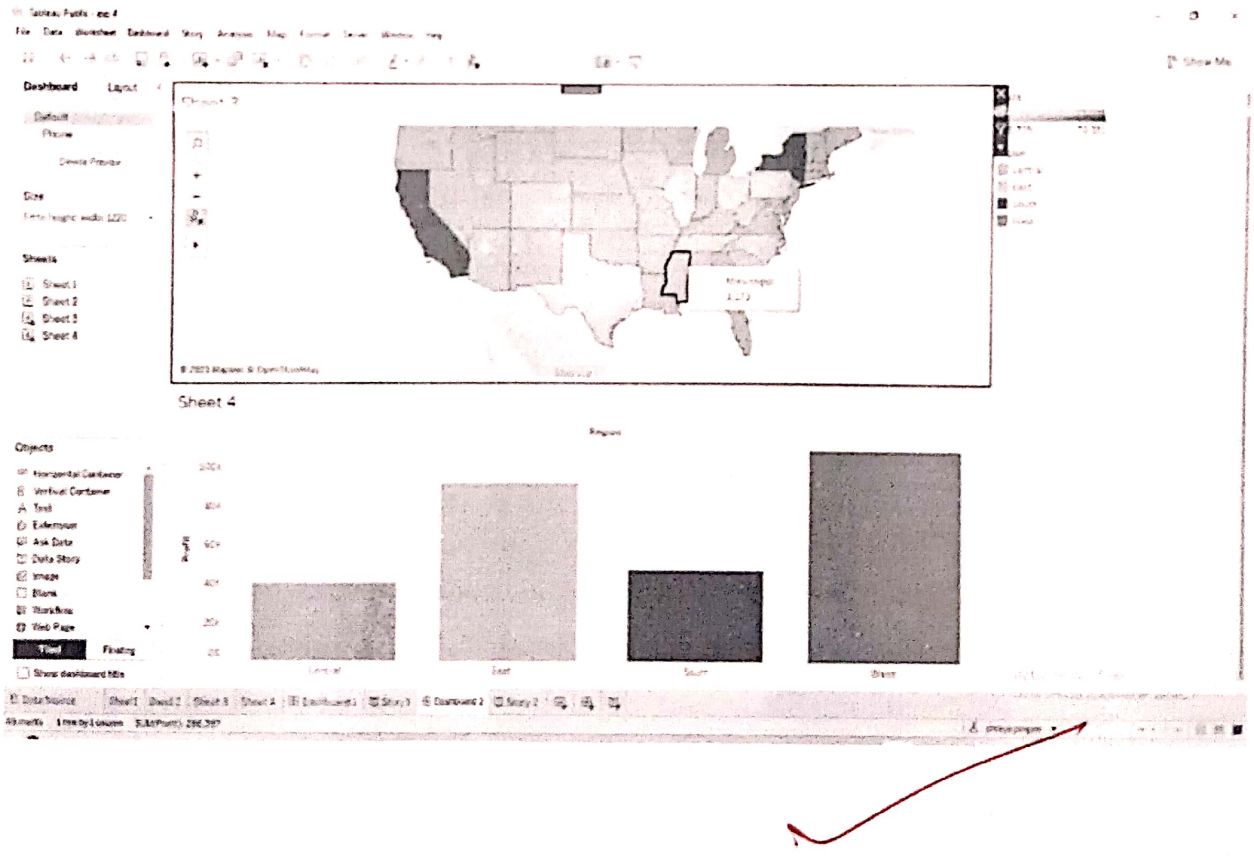
Drilldown Between Dashboards:

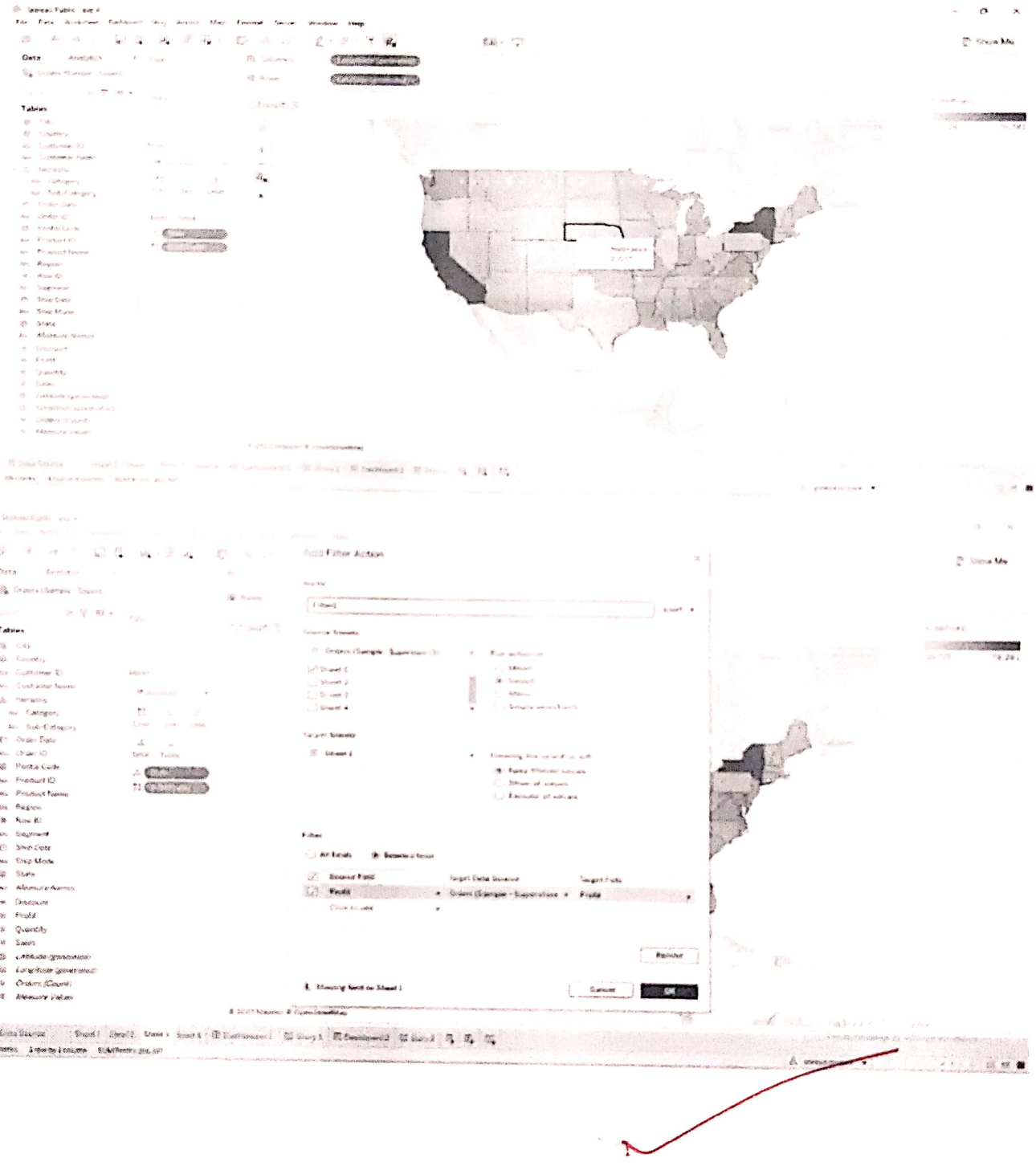
Drilldown functionality is a powerful feature in dashboard design that allows users to navigate between different levels or views of data. It's particularly useful when dealing with complex data hierarchies or presenting data with varying levels of granularity. Users can start at a high-level summary and then drill down into more detailed information. For example, in a geographical sales dashboard, users might begin by viewing sales by continent, then drill down to countrylevel data, and further to individual city data.

Drilldown provides a structured approach to exploring data, allowing users to focus on what matters most to them without overwhelming them with unnecessary details. It enhances the dashboard's flexibility and makes it easier for users to explore and analyze data effectively, tailoring their analysis to their specific needs and interests.

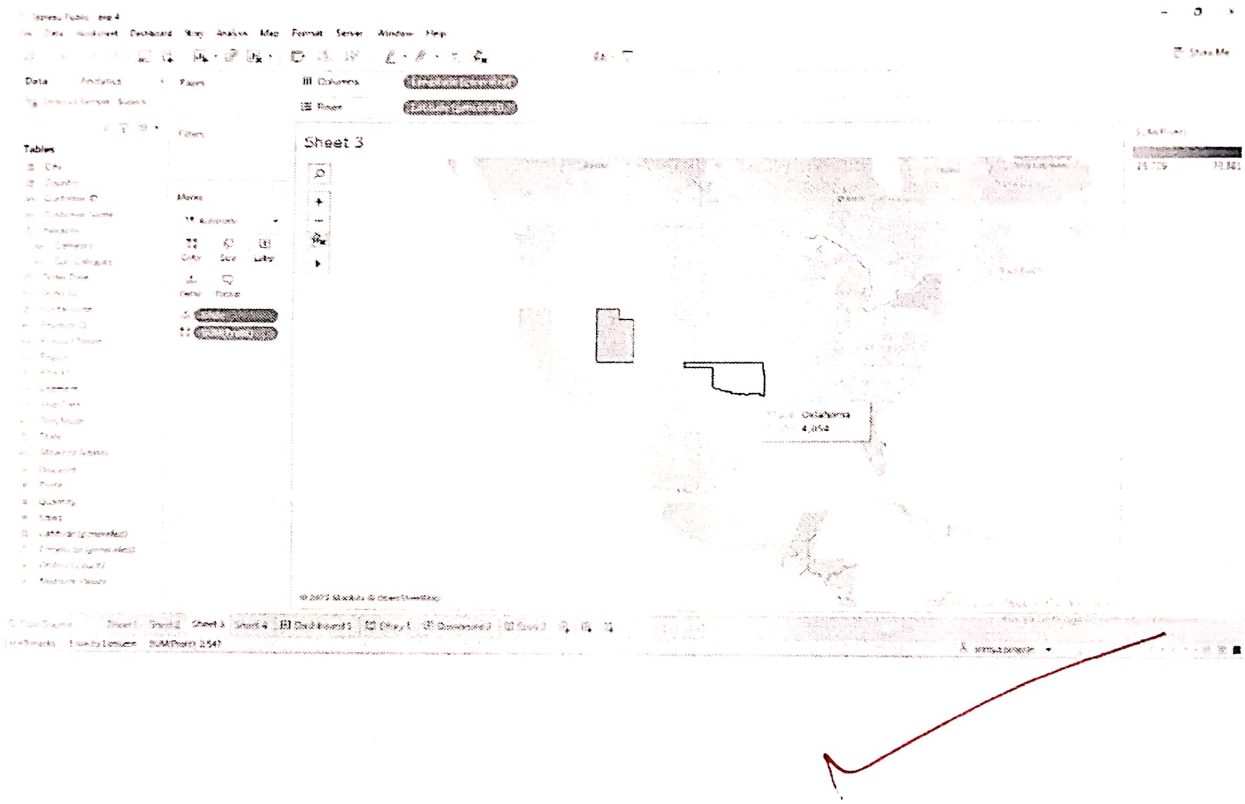
Outputs:-

a) Filter:-





b) Highlight:-



C)URL:-

Lab Outcome:

The lab outcome suggests that using an interactive dashboard with text and visual tooltips in a drill-down approach enhances data exploration and insight generation, providing a valuable tool for data-driven decision-making

Conclusion:

In conclusion, creating an interactive dashboard with text and visual tooltips usin a drill-down approach involves steps such as providing a high-level overview, adding interactive elements, text and visual tooltips, and a hierarchical structure. It aims to empower users to explore data, maintain data consistency, and gain valuable insights for makin informed decisions.

Experiment No 5

Title: Advanced visualizations

Aim: 5.1 Create more advanced data visualization dashboards by using complex Calculation

5.2 Using multiple source tables

Objective: To create more advanced data visualization dashboards by incorporating complex calculated fields for in-depth data analysis and insights, and to optimize user interaction by organizing dashboard filters in a tabular layout, streamlining data exploration for more efficient and customized data-driven decision-making.

Theory:

5.1

Creating more advanced data visualization dashboards using complex calculated fields involves leveraging the power of custom calculations and formulas to extract deeper insights and present data in a more meaningful way.

Complex calculated fields enable the dashboard designer to go beyond basic data presentation and engage in more advanced data manipulation and analysis. This can include:

Advanced Metrics: Calculating sophisticated performance metrics and KPIs that are not readily available in the raw data. For example, creating a custom profitability index based on various financial data points.

Statistical Analysis: Implementing statistical functions to uncover patterns, trends, and anomalies within the data. This may involve calculating correlations, regressions, or statistical significance.

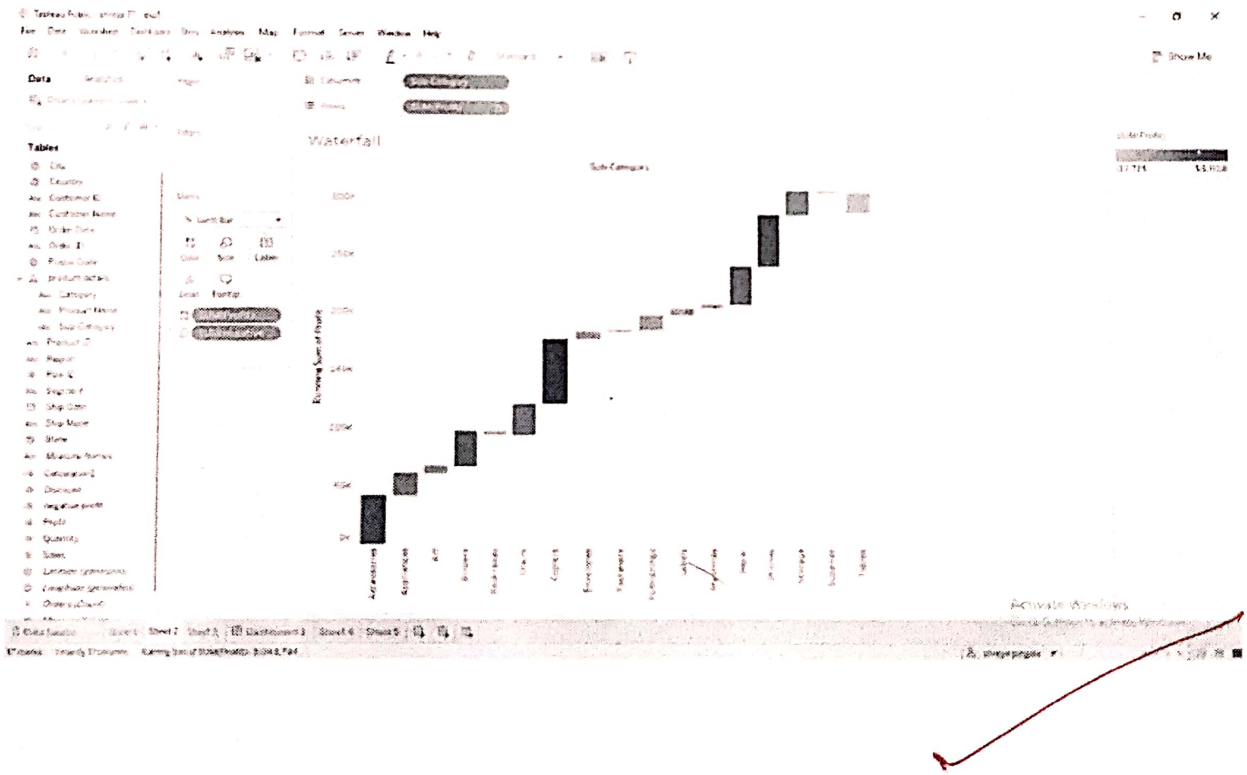
Data Transformations: Using complex formulas to transform the data, such as normalizing values, aggregating data at different levels, or applying mathematical models to predict future trends.

Conditional Logic: Incorporating conditional statements within calculated fields to categorize or label data points based on specific criteria. This can be useful for segmenting data into different groups or identifying outliers.

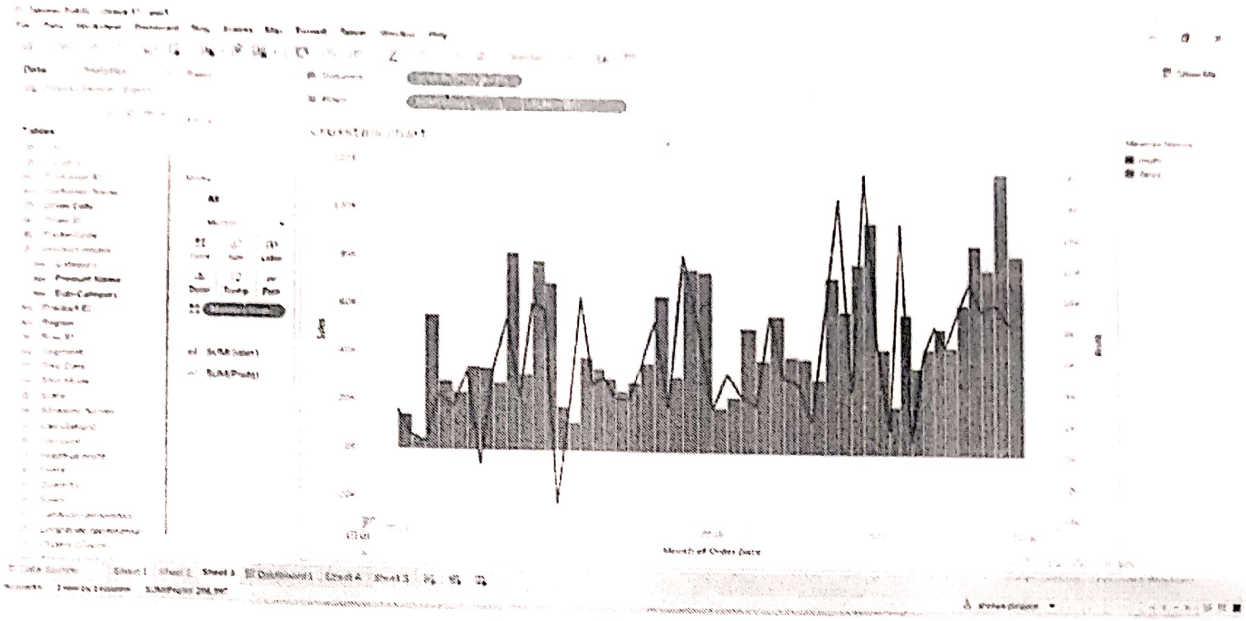
By employing complex calculated fields, dashboard designers can present a richer, more insightful view of the data, enabling users to make more informed decisions and gain a deeper understanding of complex datasets. This advanced approach is especially useful when basic visualizations and simple aggregations are insufficient to convey the complexity and nuances of the data.

Output:-

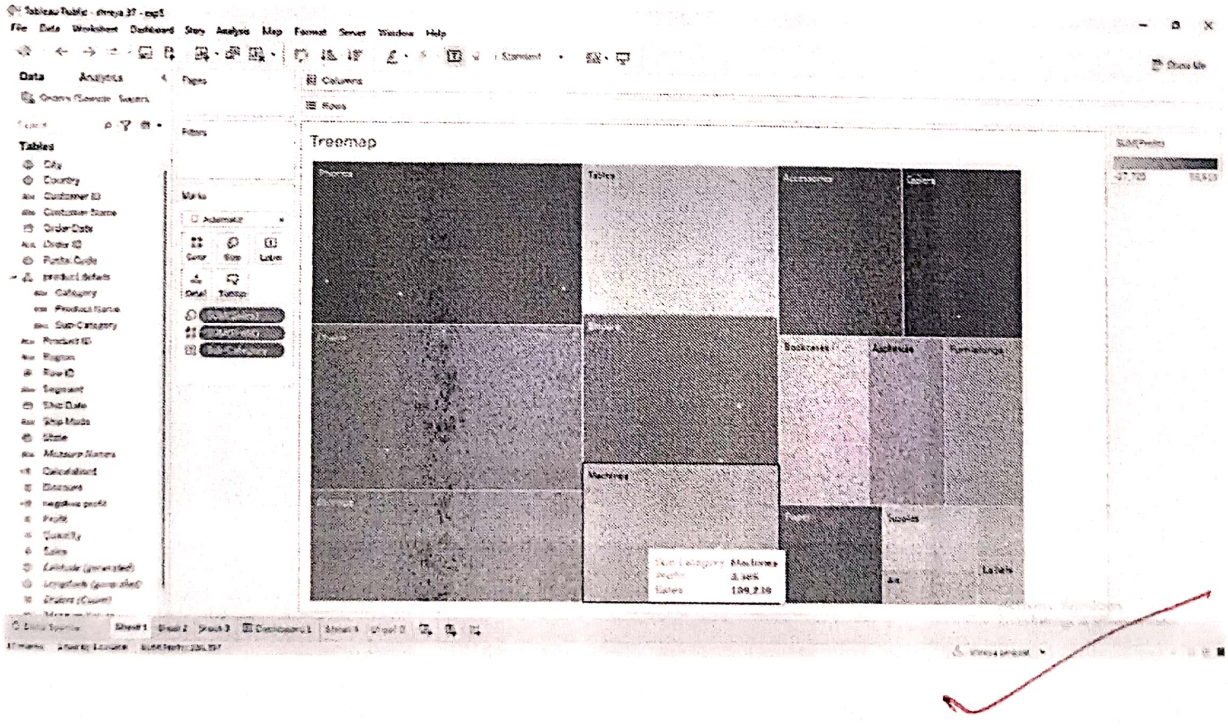
a) Waterfall Chart



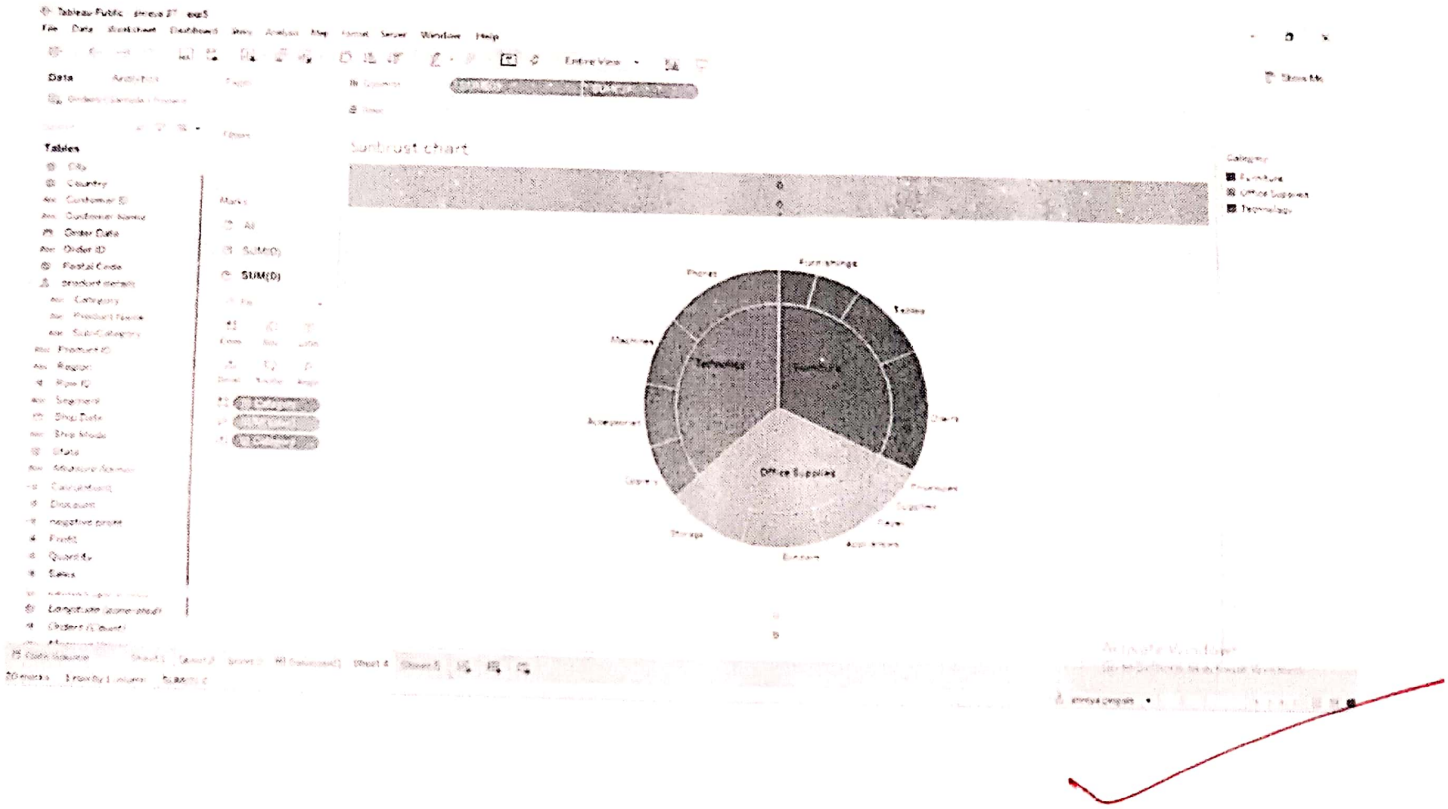
b) Crosstalk Chart



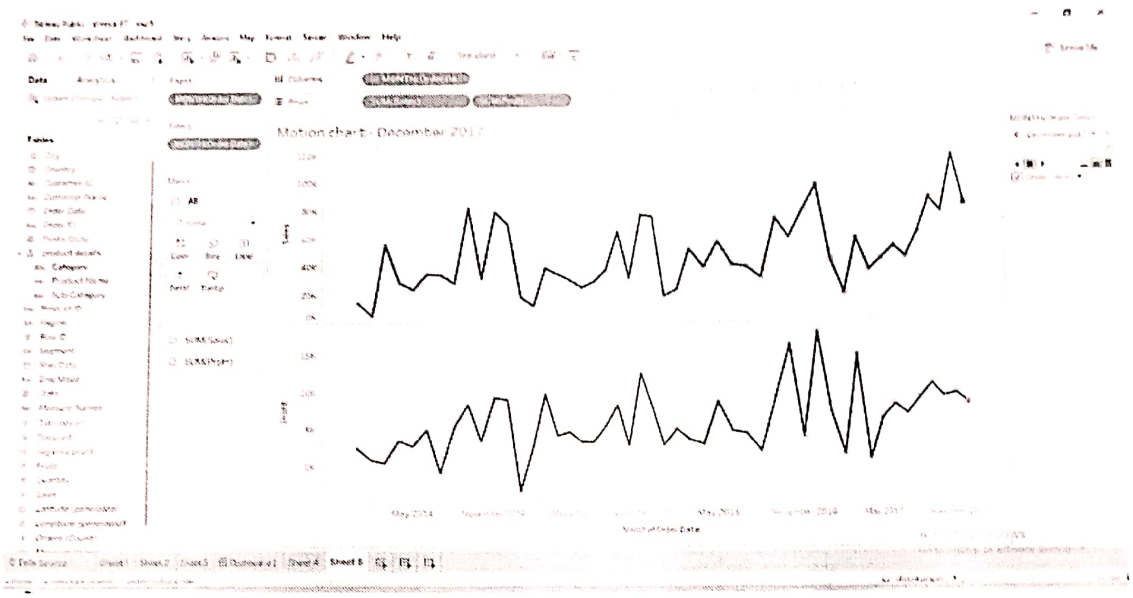
c) Tree Map



d) Sunburst chart



e) Motion Chart



Using multiple source tables in Tableau involves integrating data from various datasets or tables to create a unified and comprehensive dataset for visualization and analysis. This process includes the following key principles:

1. Data Integration: Combining data from multiple source tables allows for a more holistic view of the information. This integration often requires a common key or field that links the data across tables, facilitating the merging process.

2. Data Blending: In cases where there is no direct relationship between the tables, data blending can be used to bring data together. Tableau provides a Data Blending feature to join data from different tables based on related dimensions.

3. Join Types: Different join types (inner, left, right, outer) can be employed to control which data is included in the merged dataset. The choice of join type depends on the specific requirements of the analysis.

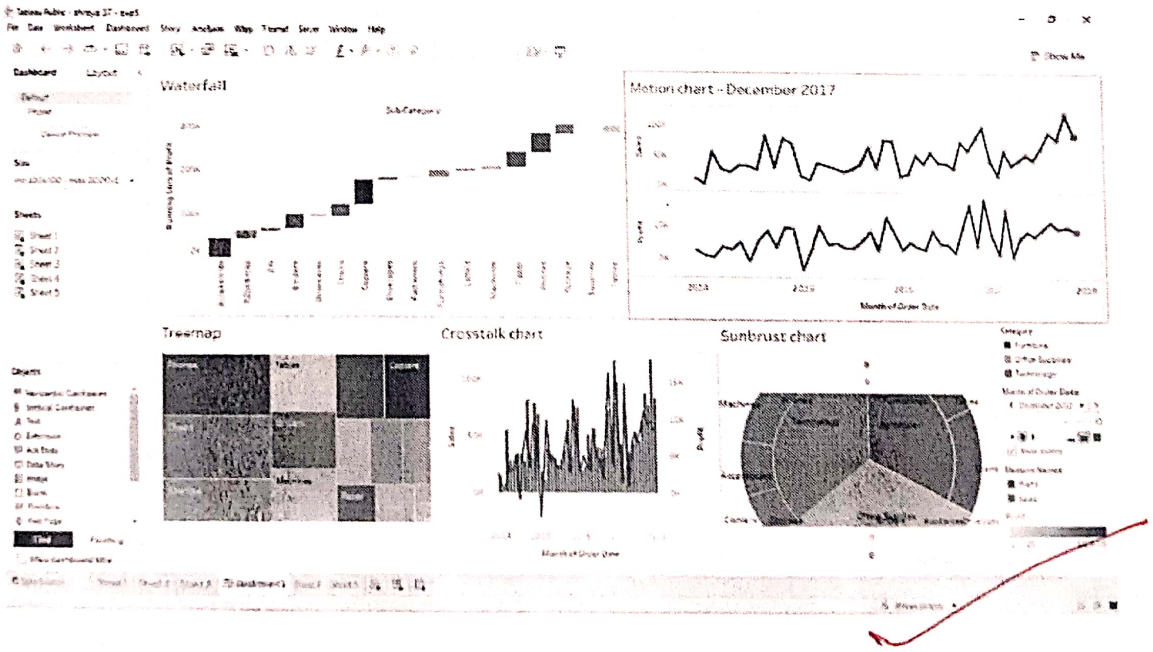
4. Data Cleanup: Prior to integration, it's essential to clean and prepare the data in each source table, ensuring consistency in field names, data types, and formatting to facilitate smooth integration.

5. Relationships: Establishing relationships within Tableau helps define how tables are connected, which fields are used as keys, and how they interact with each other in visualizations. These relationships are important for maintaining data accuracy and structure.

6. Data Performance: When working with multiple source tables, data performance becomes a critical consideration. Optimizing data extracts and query performance can involve aggregation, filtering, or other techniques to ensure that the dashboard remains responsive and efficient

By successfully using multiple source tables in Tableau, analysts and data professionals can create more comprehensive, insightful, and actionable visualizations. This process enables the blending of diverse datasets and empowers users to gain a broader perspective on their data, revealing connections and patterns that may not be apparent when looking at each source table in isolation.

Output:-



Lab Outcome:

Using multiple source tables in Tableau enables the integration of data from various datasets, fostering comprehensive visualizations and informed decision-making while ensuring data accuracy and consistency.

Conclusion:

In summary, using multiple source tables in Tableau enhances data integration and visualization, facilitating informed decision-making. It ensures comprehensive insights while maintaining data accuracy.

Experiment No 6

Title: Data Storytelling

Aim: 6.1 Create data story in Tableau - Overview of the Tableau ecosystem

6.2 Design end-to-end dashboards with the help of storytelling

Objective:

To apply best practices in data storytelling and visualization to create informative and visually appealing content, and to study how to design end-to-end dashboards within Tableau, combining multiple data stories into a unified and interactive visualization platform.

Theory 6.1

Creating a data story in Tableau is a powerful method for effectively communicating data insights, patterns, and trends. It involves the combination of data visualizations, narratives, and interactivity to

engage the audience and deliver a compelling message. Data storytelling is the art of presenting data in a way that captivates your audience, making it easy for them to understand and derive meaningful insights from the data. In Tableau, this involves several key steps:

a. Data Preparation: Begin by loading your data into Tableau. Tableau can connect to various data sources, including databases, spreadsheets, and web data connectors. Ensure that your data is clean, well-

structured, and relevant to the story you want to tell.

b. Visualizations: Visualizations are the building blocks of a data story. Tableau offers a wide range of chart types, including bar charts, line charts, scatter plots, maps, and more. Choose the appropriate
visualization type for your data and the insights you want to convey.

c. Story Structure: A data story should have a clear and logical structure, just like any narrative. It typically consists of the following elements:

Introduction: Begin with an introduction that sets the context for the story. What is the problem or question you're addressing with the data?

Development: In this part, you present the data visualizations and analyses that answer the question or address the problem. You should guide the audience through the data, highlighting key findings and
insights.

Climax: The climax of your data story is where you present the most significant insights or findings. This is the "Aha!" moment that your audience should reach.

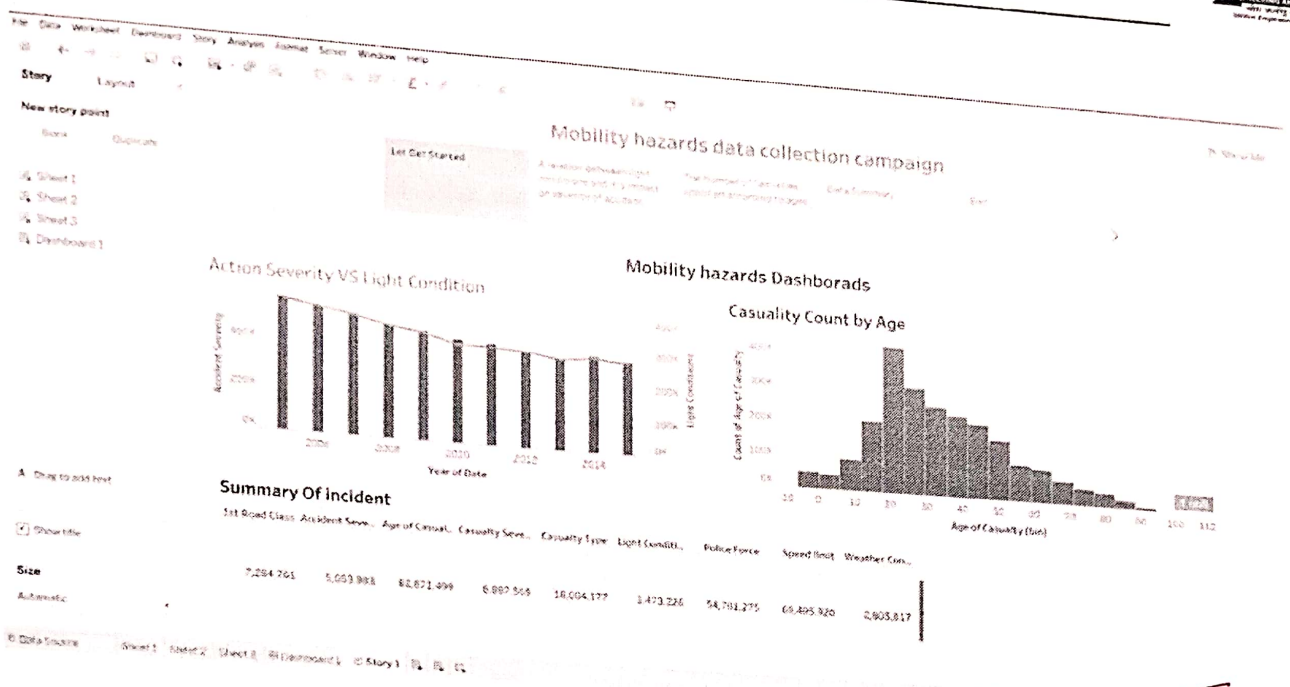
Conclusion: Conclude your story by summarizing the key takeaways and implications. What action or decision should the audience consider based on the data presented?

d. Interactivity: Tableau allows you to add interactivity to your data visualizations. This can include filters, parameters, actions, and tooltips. Interactivity makes your data story engaging and allows the audience to explore the data on their own.

e. Annotations: Annotations provide context and additional information to your visualizations. They can be used to highlight specific data points or explain unusual observations.

f. Consistency: Maintain a consistent style throughout your data story. Use a common color palette, fonts, and formatting to make your story visually appealing and easy to follow.

Output -



6.2

Designing end-to-end dashboards with the help of storytelling is a crucial aspect of data visualization in Tableau. These dashboards serve as a comprehensive platform for presenting a collection of data stories in an organized and interactive manner. An end-to-end dashboard in Tableau is a compilation of multiple data stories, visualizations, and interactive components that provide a holistic view of a dataset or a set of related data. These dashboards can be used to present a range of insights and help the audience make informed decisions. Here are the key steps involved in designing end-to-end dashboards:

a. Dashboard Creation: Start by creating a new dashboard within Tableau. You can do this by selecting "Dashboard" from the menu. A blank canvas will be provided for you to work on.

b. Layout Design: Plan the layout of your dashboard. Decide how you want to arrange your data stories, visualizations, and any additional elements such as titles, filters, and explanatory text. Proper layout
design is critical for user engagement.

c. Data Story Integration: Incorporate your data stories into the dashboard. Each data story can be
represented as a separate dashboard object or a "story point." These story points act as individual pages within your end-to-end dashboard.

d. Interactivity: Enhance the dashboard's interactivity. Utilize filters, parameters, and actions to allow users to explore the data, drill down into specific details, and customize their experience.

e. Consistency: Maintain a consistent design across all the data stories within your dashboard. Use the same color schemes, fonts, and formatting to create a unified and visually appealing presentation.

f. Titles and Descriptions: Provide clear and informative titles and descriptions for each data story or visualization within the dashboard. This helps guide the audience and set the context.

g. Annotations: Use annotations to add context to your visualizations and explain significant trends or outliers within the data.

h. Story Flow: Ensure a logical and coherent flow of data stories within the dashboard. Lead the audience through the data, building up to the most critical insights.

Here are some best practices to consider when designing end-to-end dashboards with storytelling :

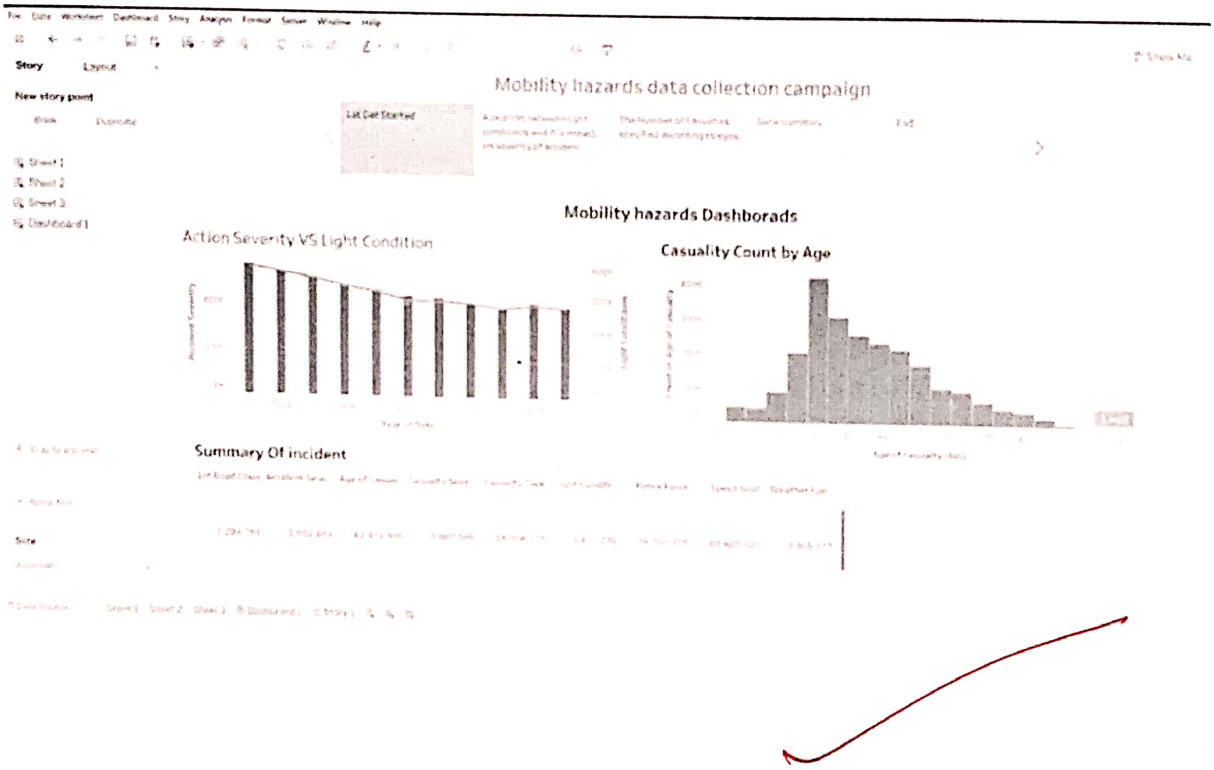
a. User-Centric Design: Always consider the needs and expectations of your target audience when designing your dashboard.
b. Clear Navigation: Ensure that users can easily navigate between data stories and visualizations within the dashboard.

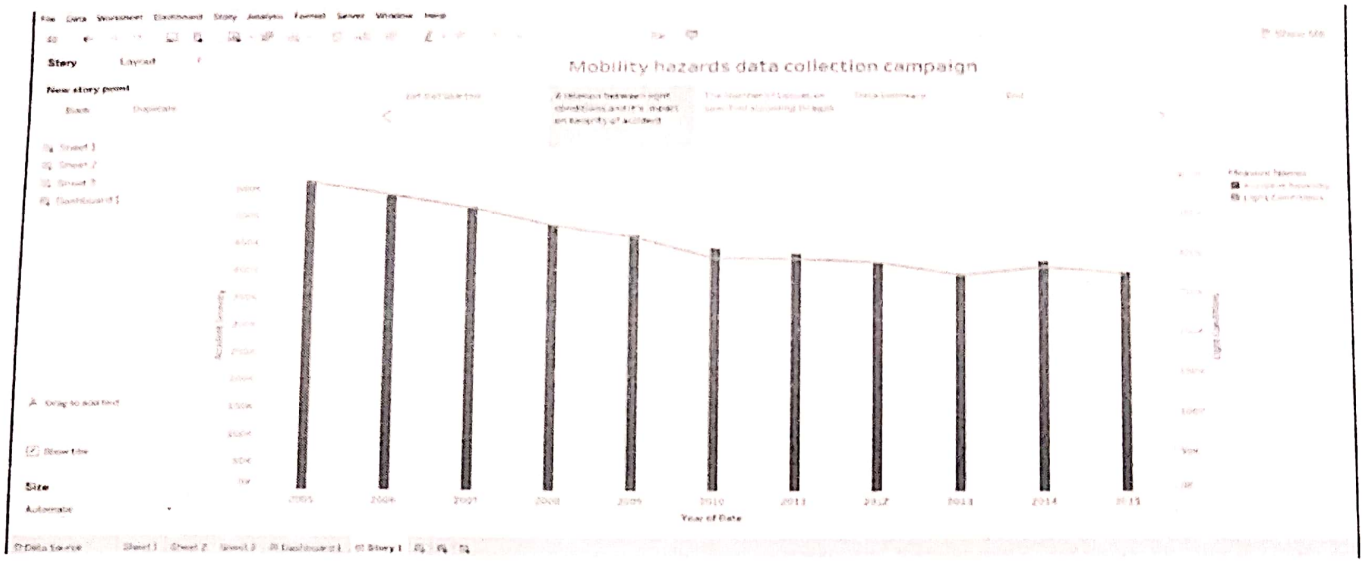
c. Data Integrity: Validate that the data used in your dashboard is accurate and up-to-date.

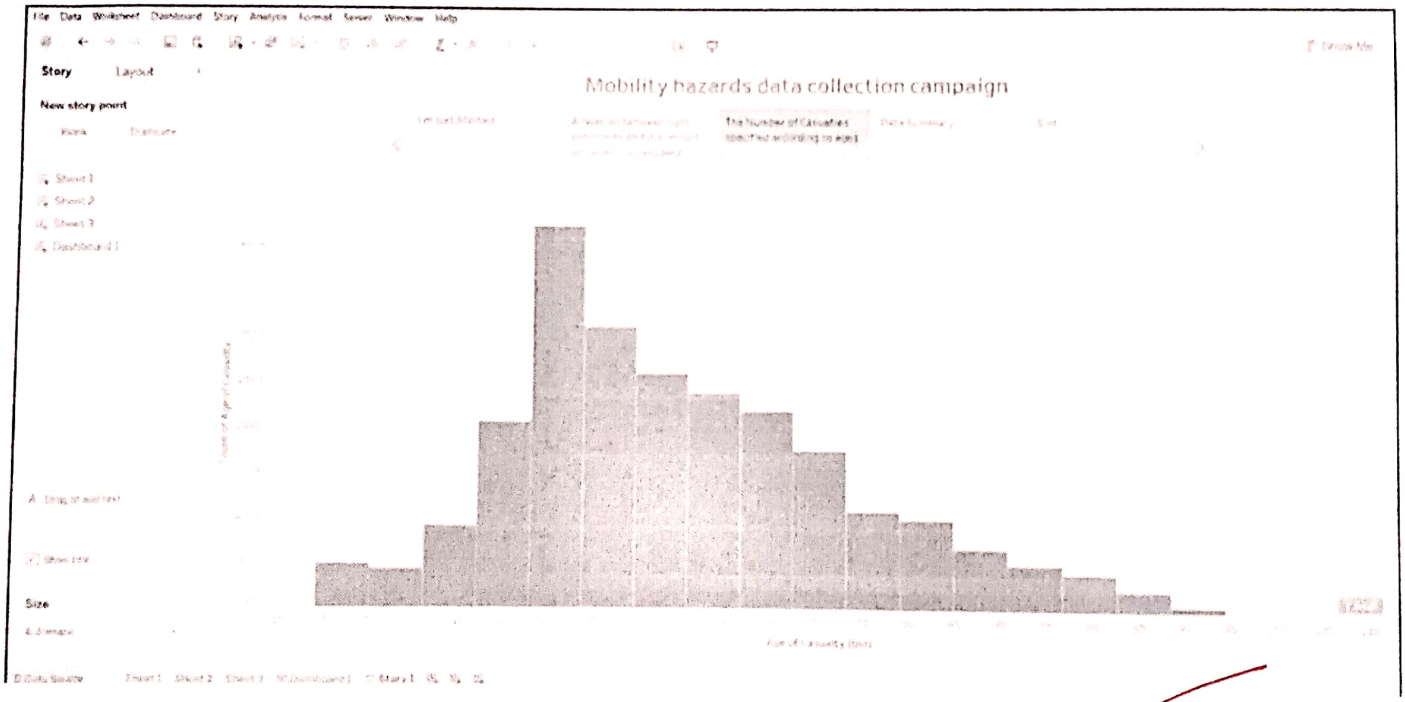
d. Feedback and Iteration: Seek feedback from peers or colleagues to improve the design and storytelling aspects of your dashboard.

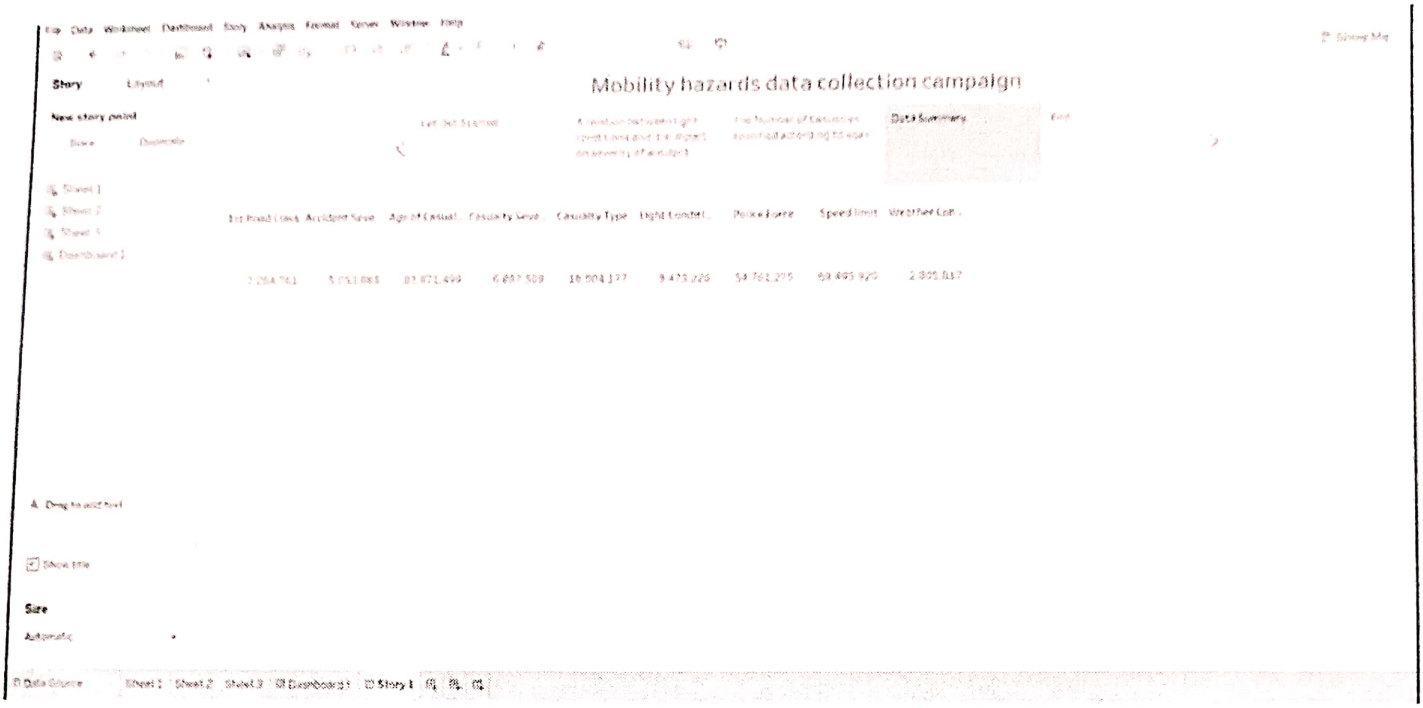
e. Continual Learning: Data storytelling and dashboard design are skills that improve with practice and learning. Explore Tableau's features and experiment with different narrative techniques.

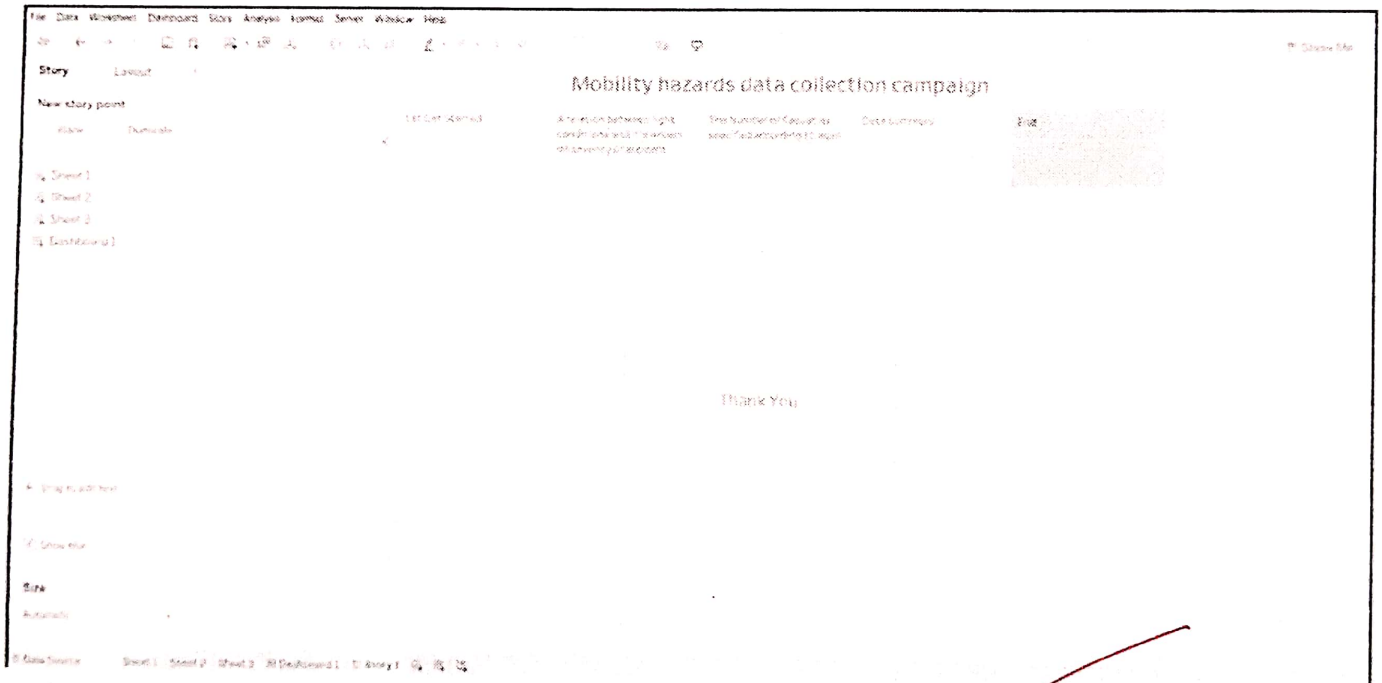
Output –











Lab Outcome:

Utilizing storytelling in Tableau for creating data stories and designing end-to-end dashboards provides a powerful platform for effective data communication and analysis. This approach enables the integration of data visualization principles and narrative techniques to convey insights,
engage the audience, and drive data-informed decision-making.

Conclusion:

Leveraging storytelling in Tableau for data visualization and dashboard design is a powerful method for conveying data insights and facilitating informed decision-making. This approach integrates data visualization best practices with narrative storytelling to create compelling and
informative content. It engages the audience and ensures that data-driven insights are presented in a clear and engaging manner.