Hindi Vidya Prachar Samiti's

RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ARTS ,SCIENCE AND COMMERCE (EMPOWERED AUTONOMOUS)

Data Analysis and Visualization

Name : Vaibhavi Shailesh Takale

Roll no : 10332

Class : Msc. Data Science and Artificial Intelligence Part I (SEM – 1)

HINDI VIDYA PRACHAR SAMITI'S

RAMNIRANJAN JHUNJHUNWALA COLLEGE

GHATKOPAR (WEST) MUMBAI 400086

CERTIFICATE

DEPARTMENT OF DATA SCIENCE AND ARTIFICIAL INTELLIGENCE

This is to certify of **Vaibhavi Shailesh Takale** of Msc. Data Science and Artificial Intelligence, Roll no: **10332**, has successfully completed the practical of Data **Analysis and Visualization** during the Academic Year 2025-2026.

Date:

(Prof. Shweta Ma'am) **External Examiner**

Prof – In - Charge

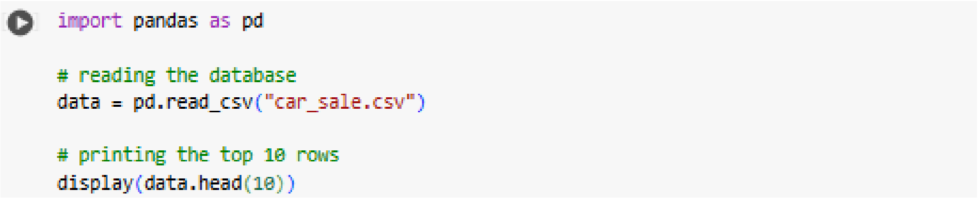
INDEX

Practical no. 1 Show Basic Visualization in Python.

Aim: To perform data visualization on the dataset `car_sale.csv` using different Python libraries such as Pandas, Matplotlib, Seaborn, Bokeh, and Plotly**.**

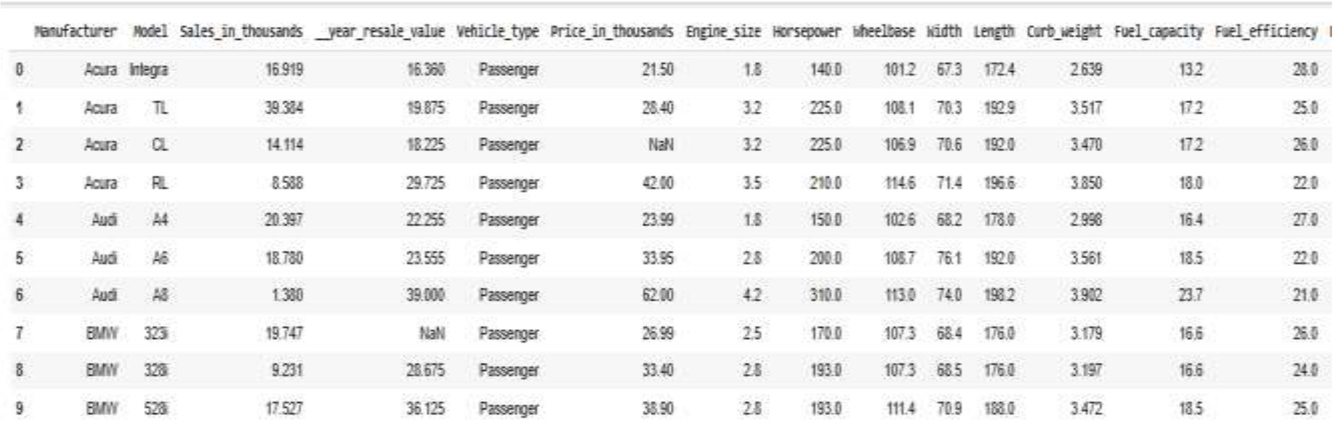
Implementation:

- **Import libraries**



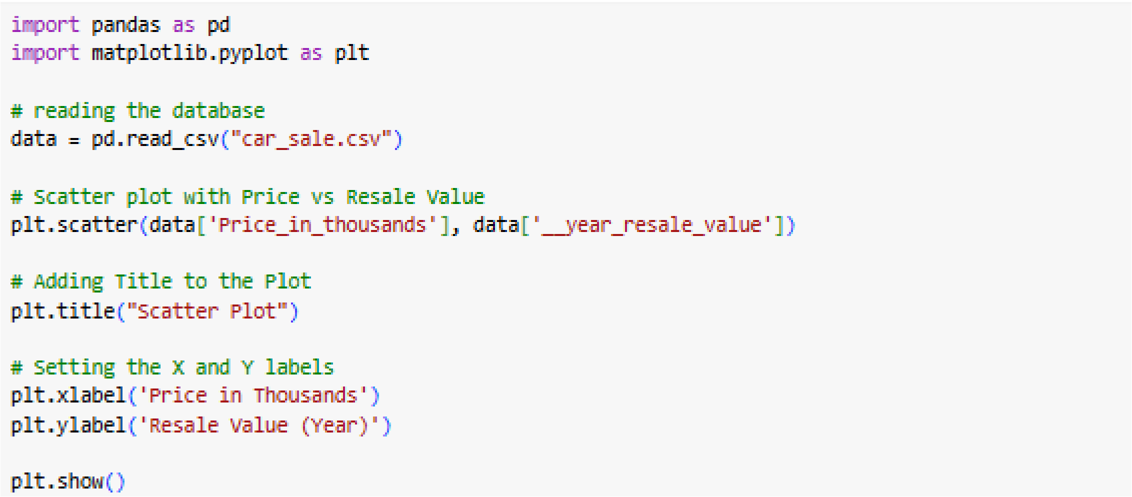
```
import pandas as pd

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

# printing the top 10 rows
display(data.head(10))
```

Output:



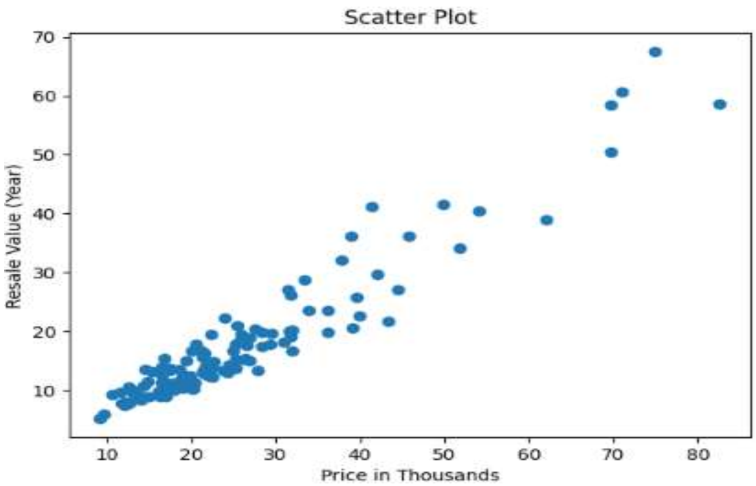
	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thousands	Engine_size	Horsepower	Wheelbase	Width	Length	Curb_weight	Fuel_capacity	Fuel_efficiency
0	Acura	Integra	16.919	16.360	Passenger	21.50	1.8	140.0	101.2	67.3	172.4	2.639	13.2	28.0
1	Acura	TL	39.384	19.875	Passenger	28.40	3.2	225.0	108.1	70.3	192.9	3.517	17.2	25.0
2	Acura	CL	14.114	18.225	Passenger	NaN	3.2	225.0	106.9	70.6	192.0	3.470	17.2	26.0
3	Acura	RL	8.588	29.725	Passenger	42.00	3.5	210.0	114.6	71.4	196.6	3.850	18.0	22.0
4	Audi	A4	20.397	22.255	Passenger	23.99	1.8	150.0	102.6	68.2	178.0	2.998	16.4	27.0
5	Audi	A6	18.780	23.555	Passenger	33.95	2.8	200.0	108.7	76.1	192.0	3.561	18.5	22.0
6	Audi	A8	1.380	39.000	Passenger	62.00	4.2	310.0	113.0	74.0	198.2	3.902	23.7	21.0
7	BMW	323i	19.747	NaN	Passenger	26.99	2.5	170.0	107.3	68.4	176.0	3.179	16.6	26.0
8	BMW	328i	9.231	28.675	Passenger	33.40	2.8	193.0	107.3	68.5	176.0	3.197	16.6	24.0
9	BMW	528i	17.527	36.125	Passenger	38.90	2.8	193.0	111.4	70.9	188.0	3.472	18.5	25.0

- **Scatter**

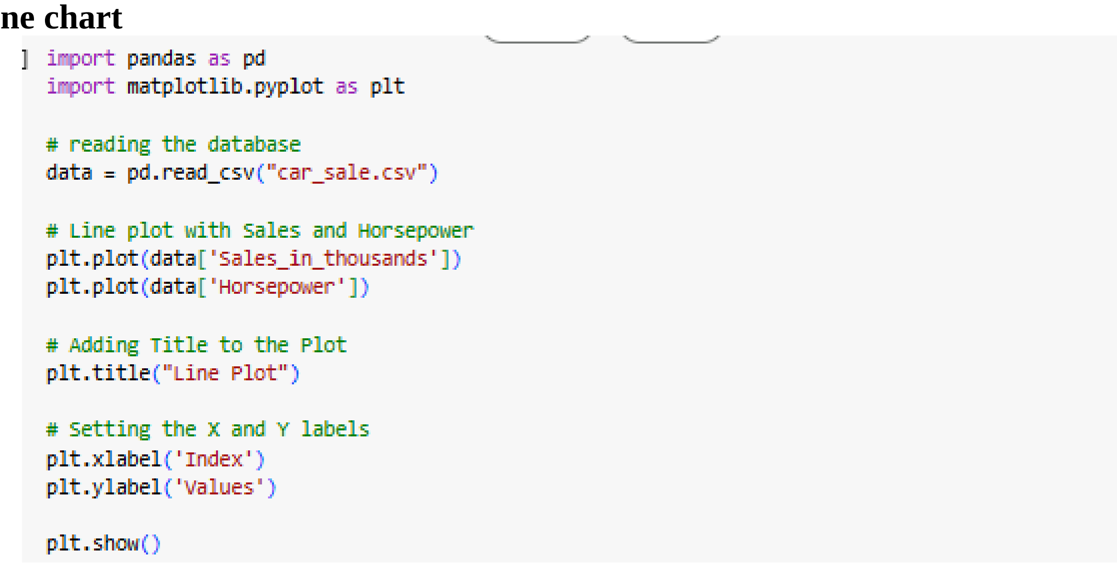


```
import pandas as pd
import matplotlib.pyplot as plt

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

# Scatter plot with Price vs Resale Value
plt.scatter(data['Price_in_thousands'], data['__year_resale_value'])

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

# Setting the X and Y labels
plt.xlabel('Price in Thousands')
plt.ylabel('Resale Value (Year)')

plt.show()
```

Output:

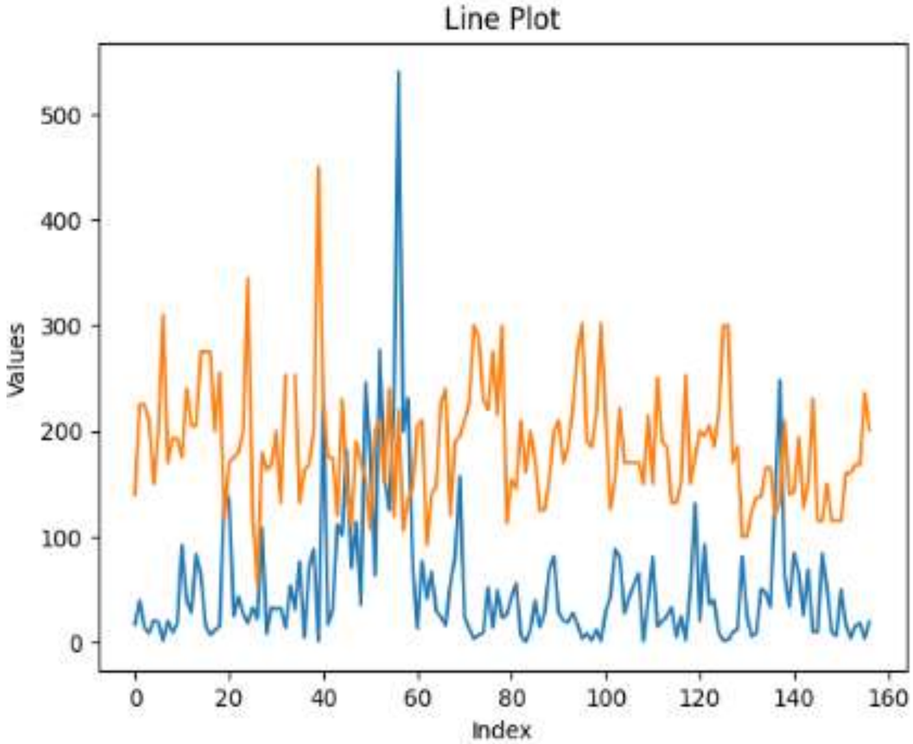


- **Line chart**

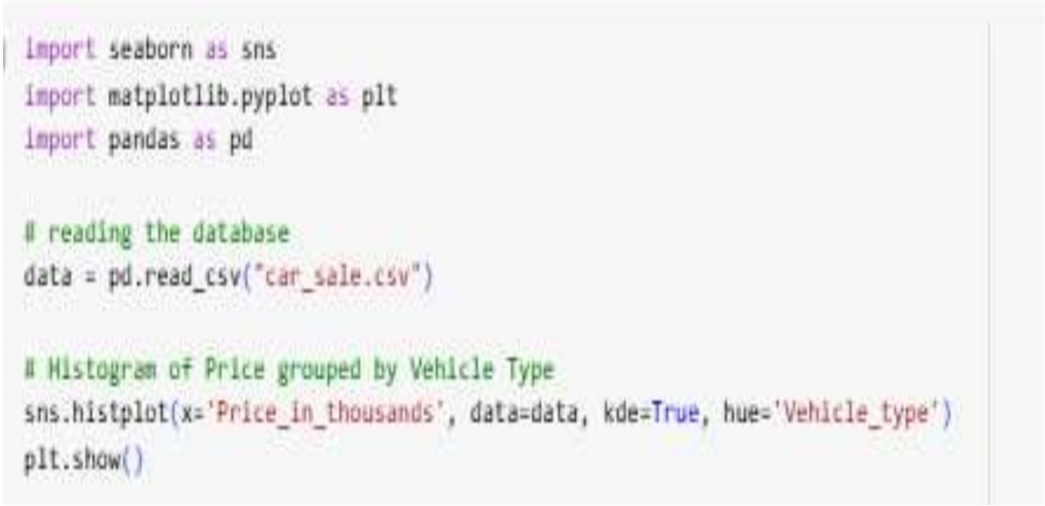


```
] import pandas as pd
  import matplotlib.pyplot as plt

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

  # Line plot with Sales and Horsepower
  plt.plot(data['Sales_in_thousands'])
  plt.plot(data['Horsepower'])

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

  # Setting the X and Y labels
  plt.xlabel('Index')
  plt.ylabel('Values')

  plt.show()
```

Output:

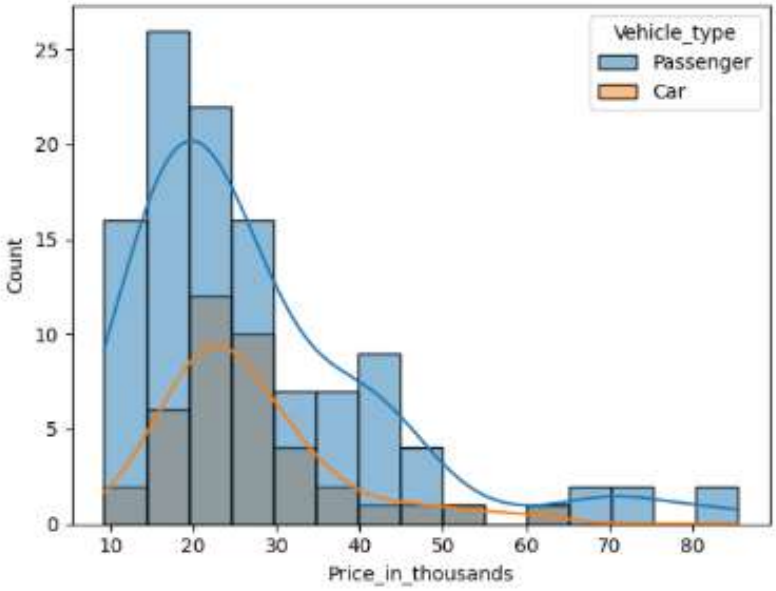


- **Seaborn Histplot**

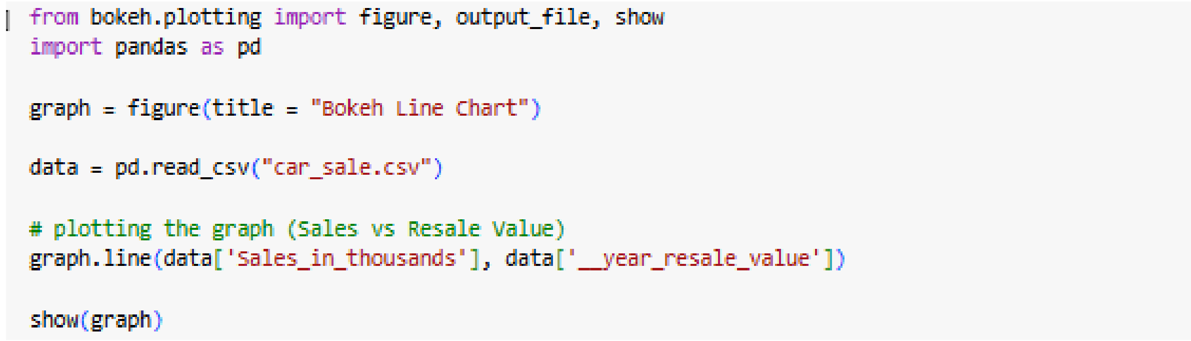


```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

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

# Histogram of Price grouped by Vehicle Type
sns.histplot(x='Price_in_thousands', data=data, kde=True, hue='Vehicle_type')
plt.show()
```

Output:

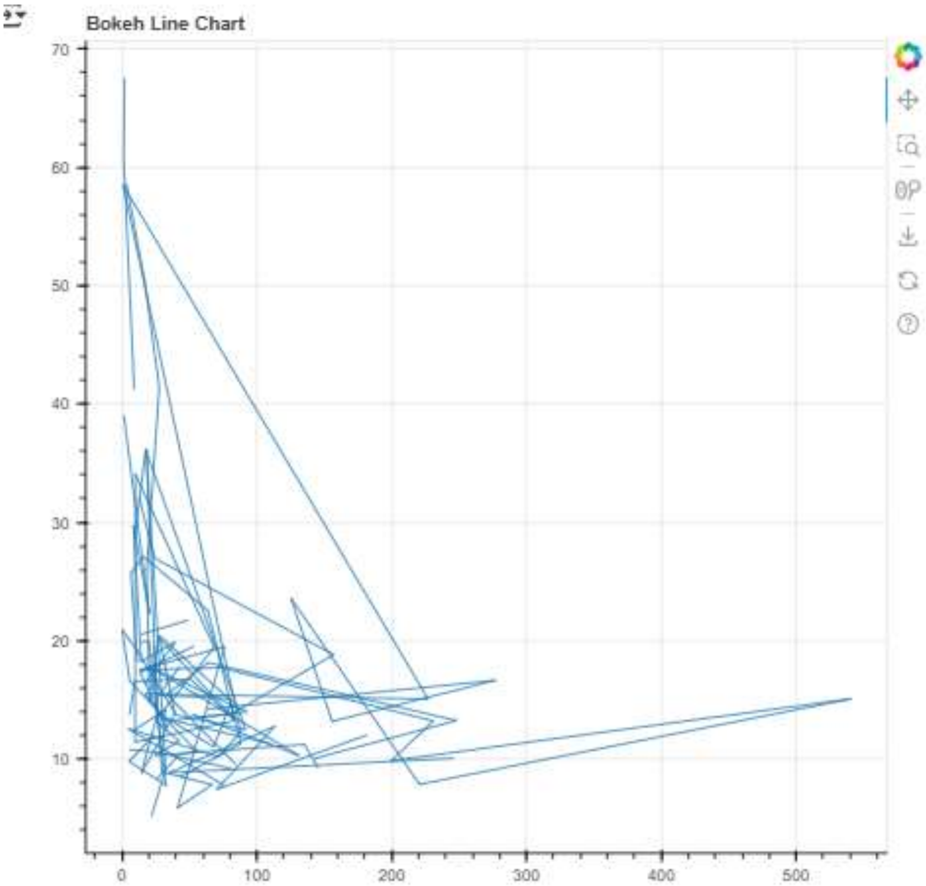


- **Boken line**

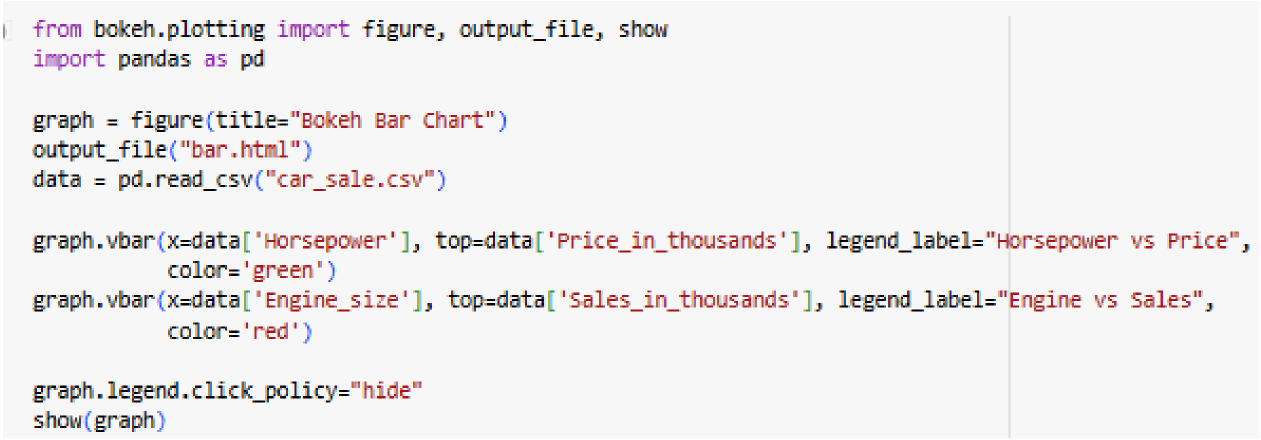


```
from bokeh.plotting import figure, output_file, show
import pandas as pd

graph = figure(title = "Bokeh Line Chart")

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

# plotting the graph (Sales vs Resale Value)
graph.line(data['Sales_in_thousands'], data['__year_resale_value'])

show(graph)
```

Output:

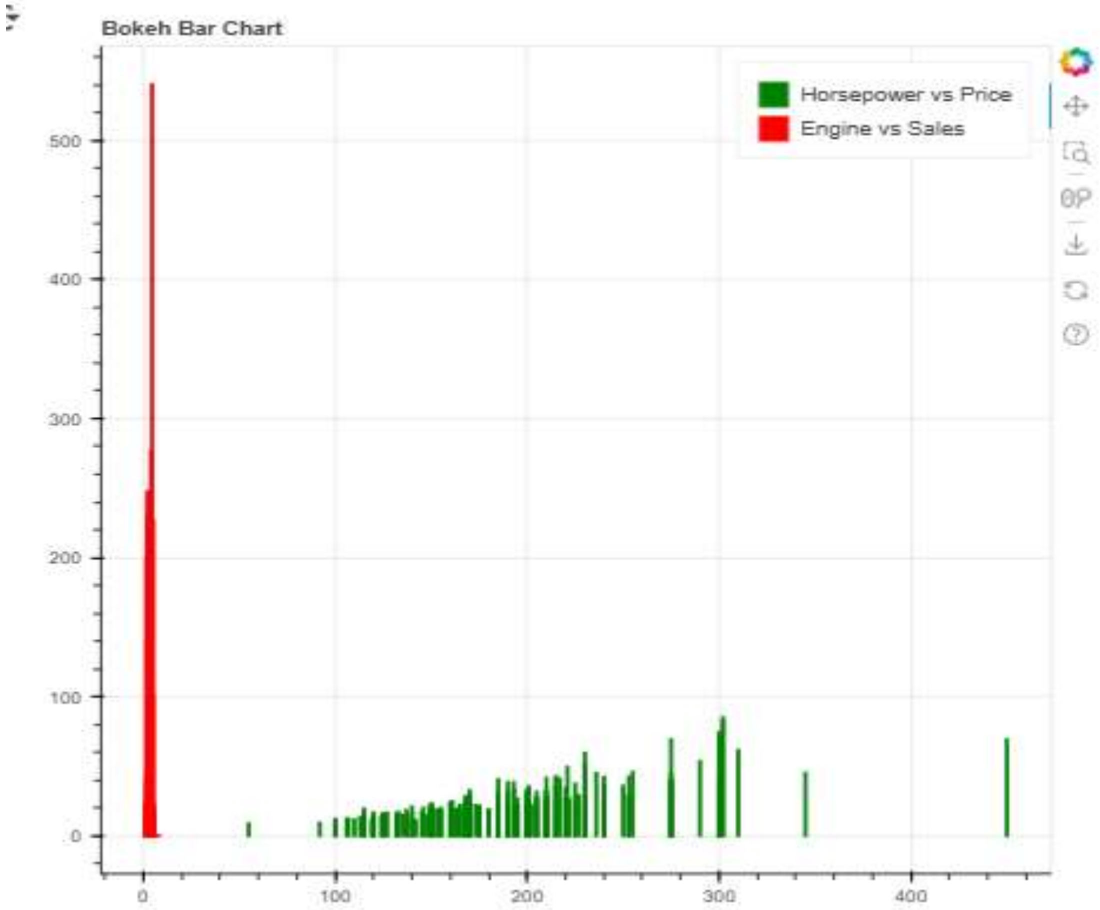


- **Bokeh vbar**

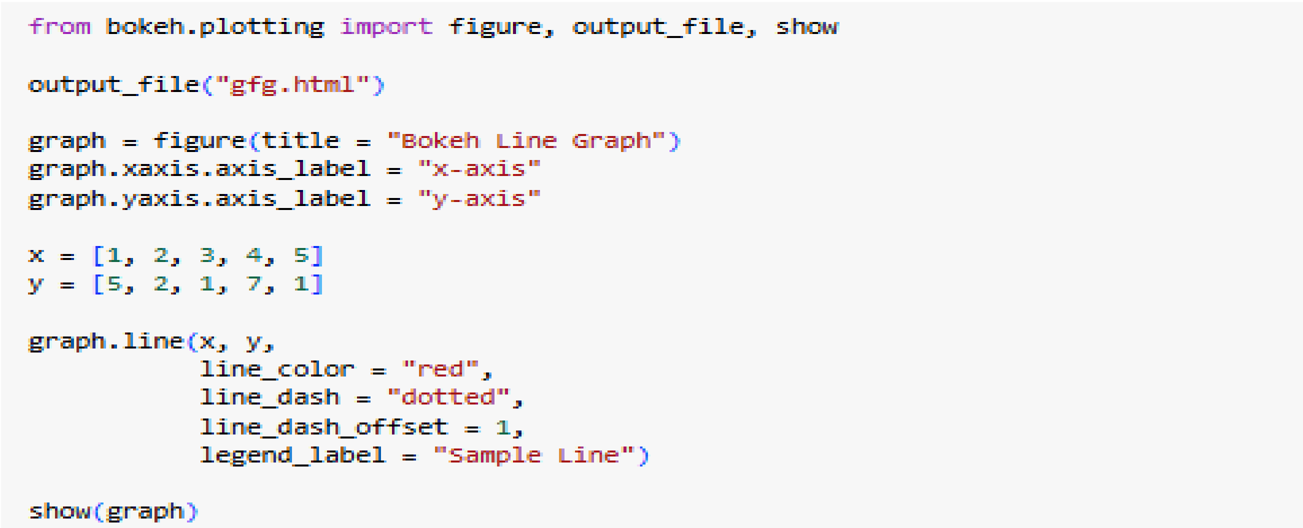


```
from bokeh.plotting import figure, output_file, show
import pandas as pd

graph = figure(title="Bokeh Bar Chart")
output_file("bar.html")
data = pd.read_csv("car_sale.csv")

graph.vbar(x=data['Horsepower'], top=data['Price_in_thousands'], legend_label="Horsepower vs Price",
           color='green')
graph.vbar(x=data['Engine_size'], top=data['Sales_in_thousands'], legend_label="Engine vs Sales",
           color='red')

graph.legend.click_policy="hide"
show(graph)
```

Output:

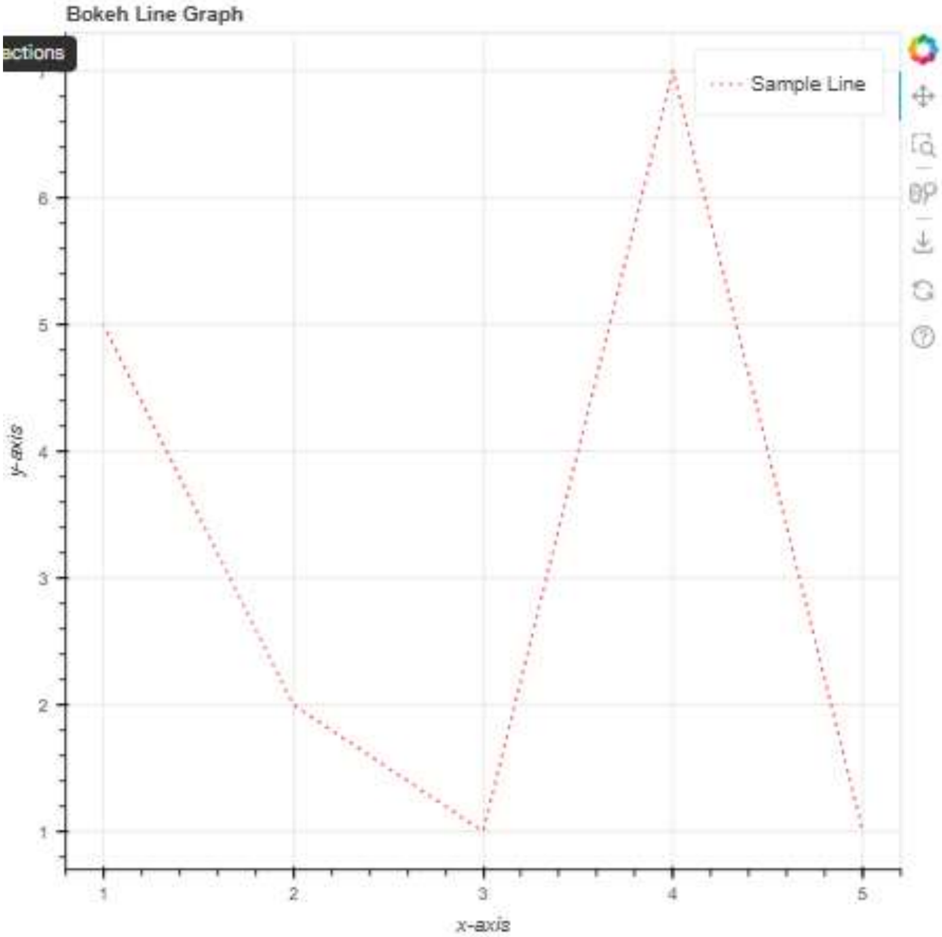


- **Bokeh custom line**

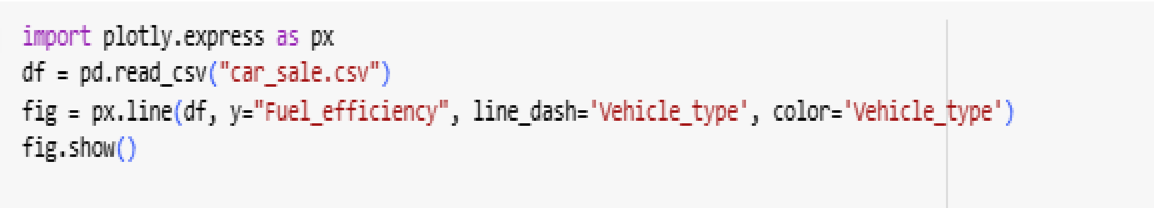


```
from bokeh.plotting import figure, output_file, show

output_file("gfg.html")

graph = figure(title = "Bokeh Line Graph")
graph.xaxis.axis_label = "x-axis"
graph.yaxis.axis_label = "y-axis"

x = [1, 2, 3, 4, 5]
y = [5, 2, 1, 7, 1]

graph.line(x, y,
           line_color = "red",
           line_dash = "dotted",
           line_dash_offset = 1,
           legend_label = "Sample Line")

show(graph)
```

Output:

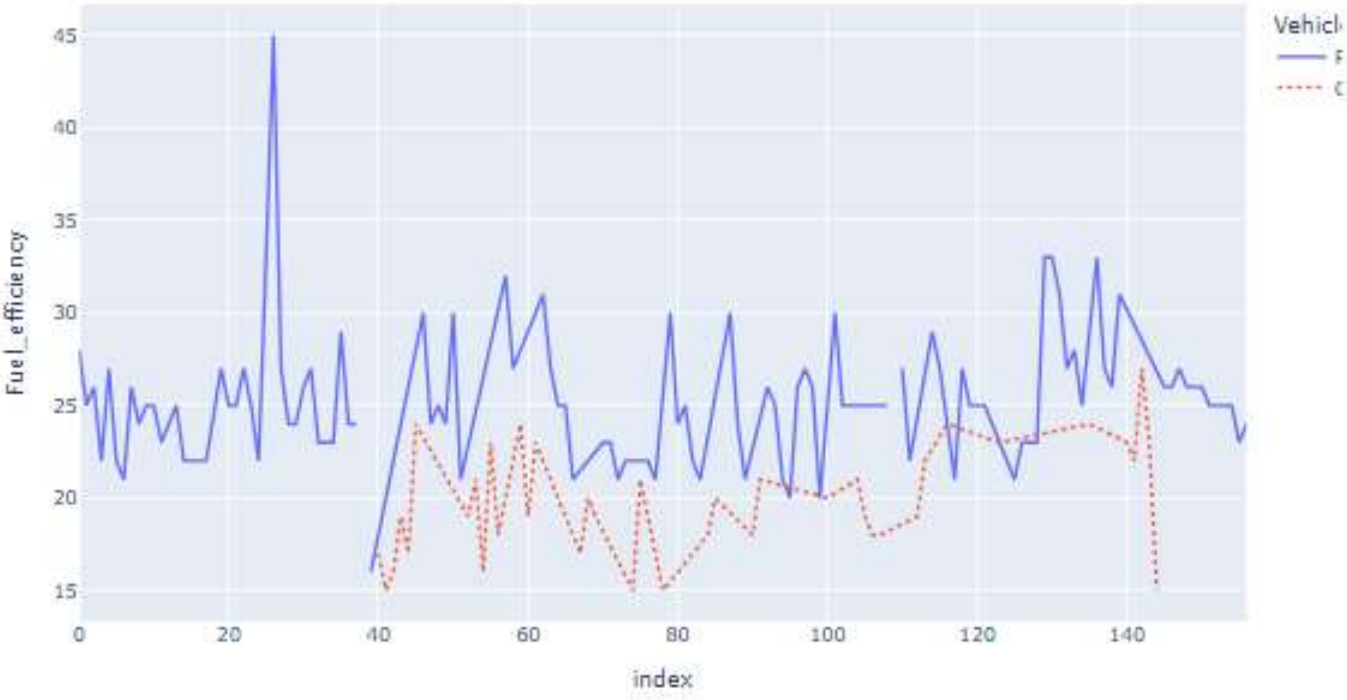


- **Plotly Line Chart**

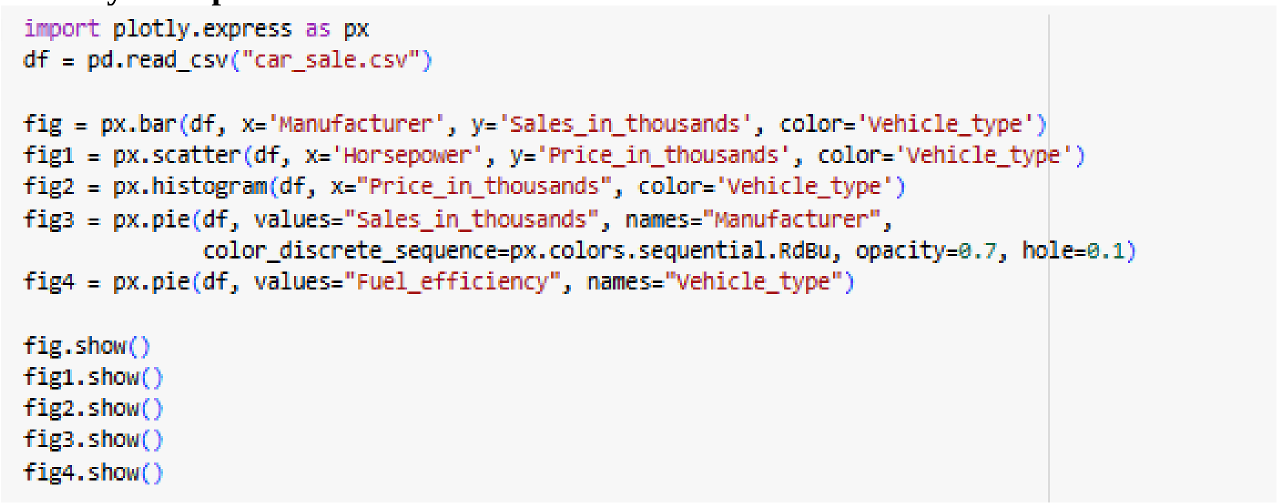


```
import plotly.express as px
df = pd.read_csv("car_sale.csv")
fig = px.line(df, y="Fuel_efficiency", line_dash='Vehicle_type', color='Vehicle_type')
fig.show()
```

Output:

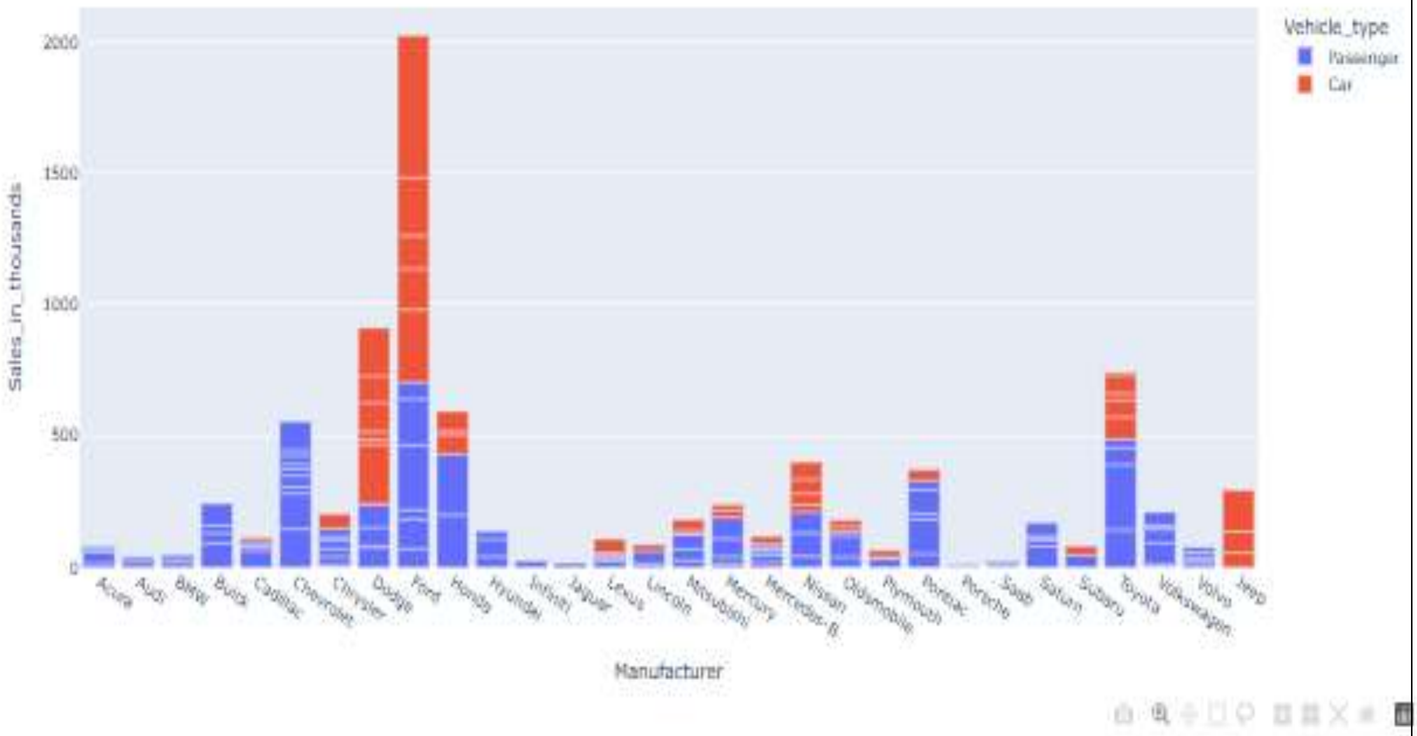


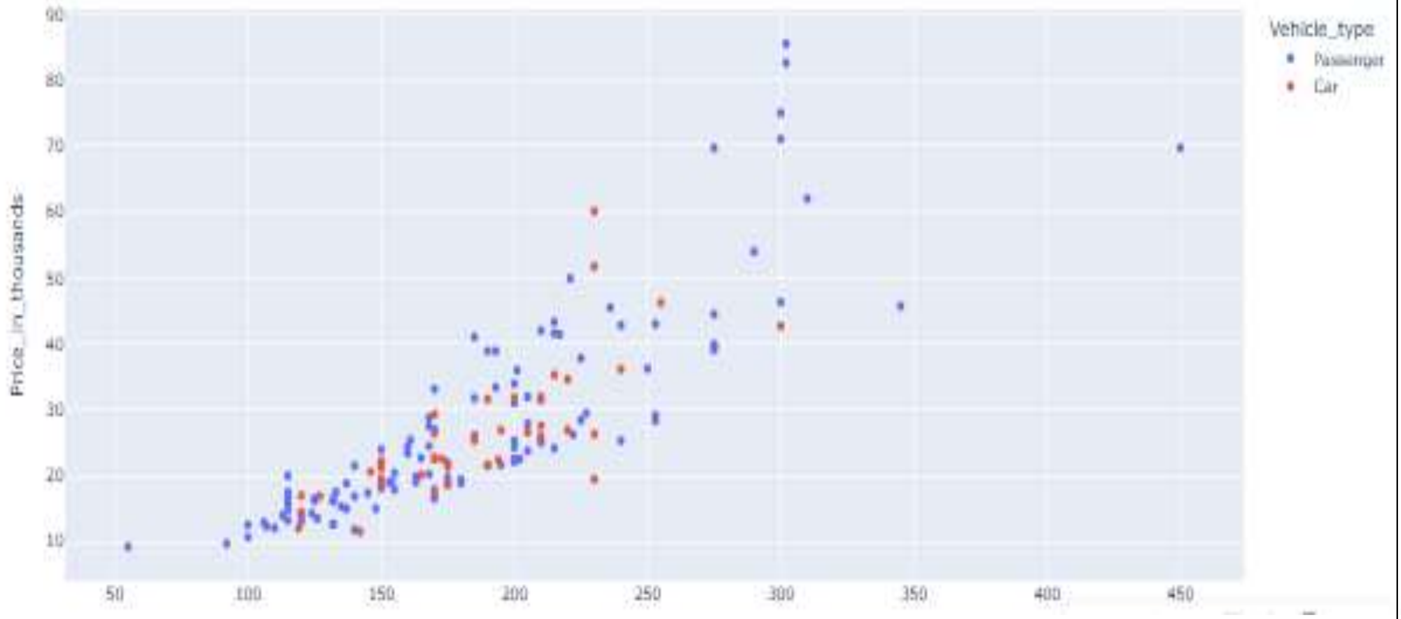
- **Plotly multiple**

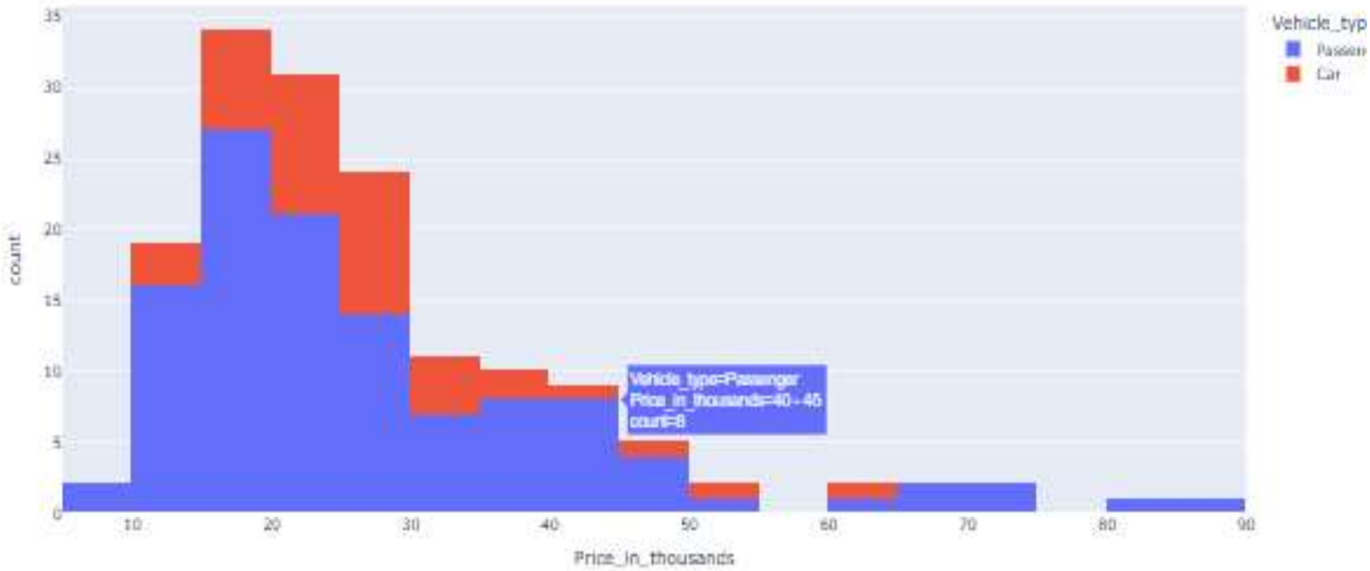


```
import plotly.express as px
df = pd.read_csv("car_sale.csv")

fig = px.bar(df, x='Manufacturer', y='Sales_in_thousands', color='Vehicle_type')
fig1 = px.scatter(df, x='Horsepower', y='Price_in_thousands', color='Vehicle_type')
fig2 = px.histogram(df, x="Price_in_thousands", color='Vehicle_type')
fig3 = px.pie(df, values="Sales_in_thousands", names="Manufacturer",
              color_discrete_sequence=px.colors.sequential.RdBu, opacity=0.7, hole=0.1)
fig4 = px.pie(df, values="Fuel_efficiency", names="Vehicle_type")

fig.show()
fig1.show()
fig2.show()
fig3.show()
fig4.show()
```

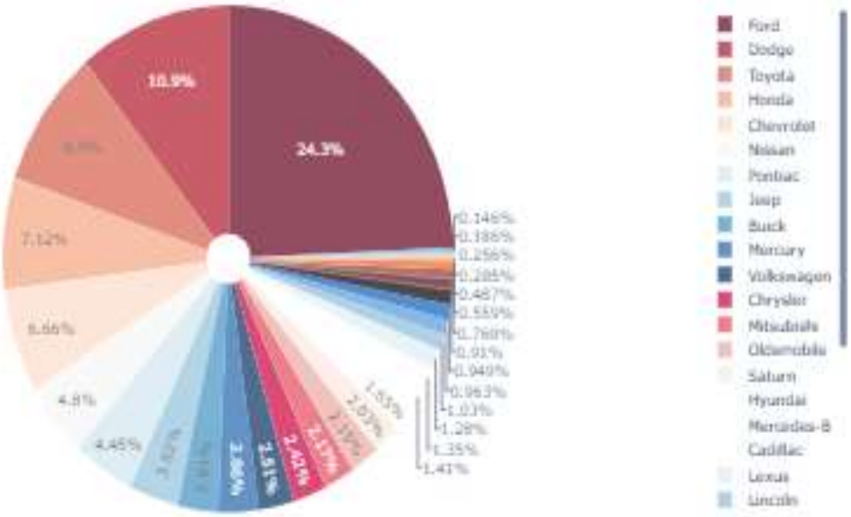
output:



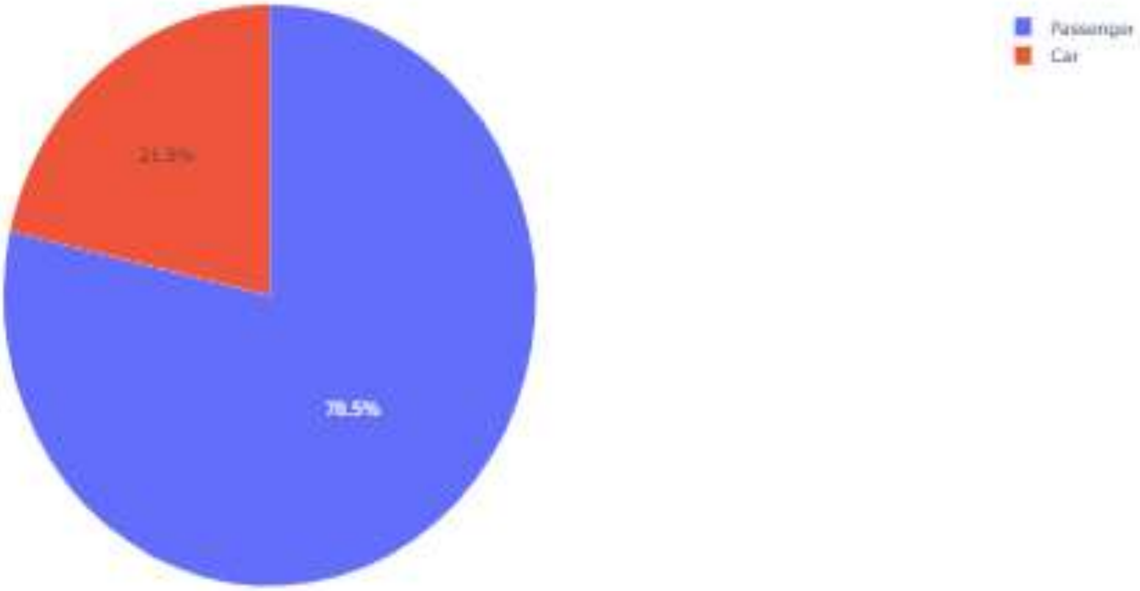




count
0
5
10
15
20
25
30
35
10
20
30
40
50
60
70
80
90
Price_in_thousands
Vehicle_typ
Passen
Car
Vehicle_type=Passenger
Price_in_thousands=40 - 45
count=8



24.3%
10.9%
7.12%
6.66%
4.8%
4.45%
3.82%
2.86%
2.51%
2.42%
2.07%
1.55%
0.146%
0.186%
0.256%
0.285%
0.487%
0.559%
0.768%
0.91%
0.949%
0.963%
1.03%
1.28%
1.35%
1.41%
Ford
Dodge
Toyota
Honda
Chevrolet
Nissan
Pontiac
Jeep
Buick
Mercury
Volkswagen
Chrysler
Mitsubishi
Oldsmobile
Saturn
Hyundai
Mercedes-B
Cadillac
Lexus
Lincoln



21.5%
78.5%
Passenger
Car

Practical no. 2 Show Basic Visualization in R.

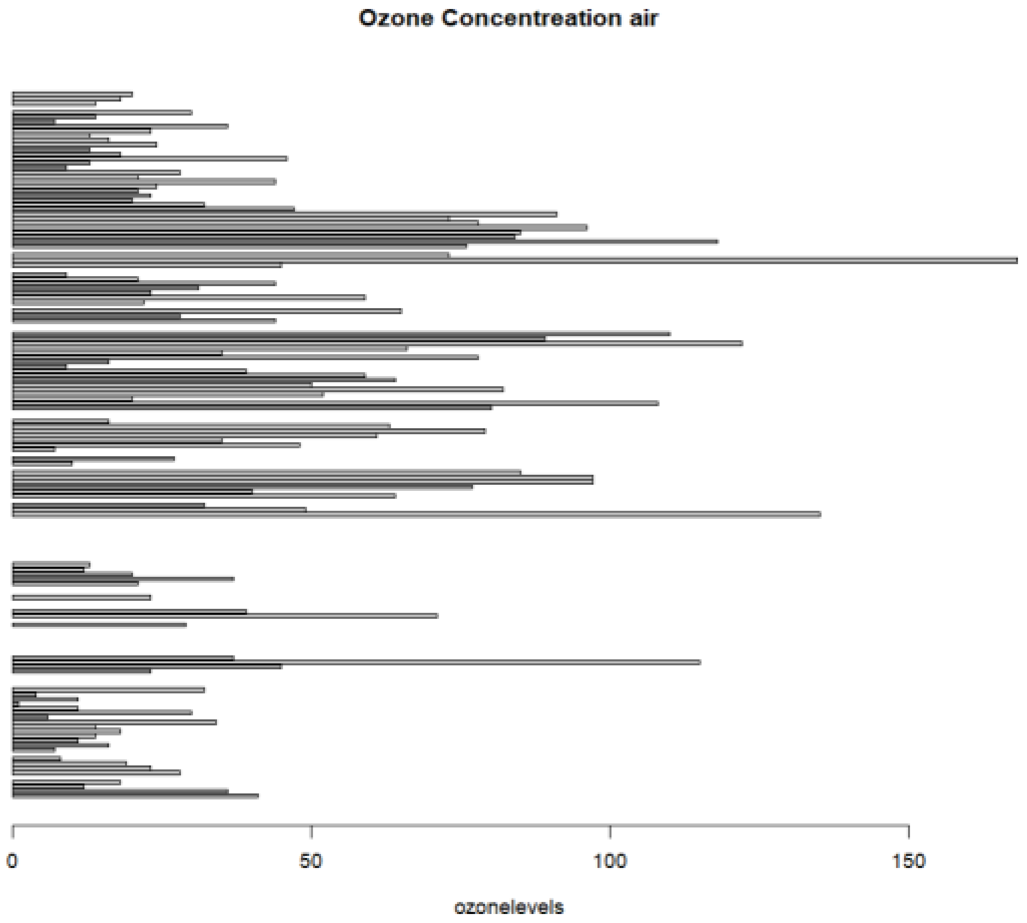
Aim: To perform basic data visualization in **R programming language** using functions like `plot()`, `hist()`, and `barplot()` for better understanding of data distribution and relationships.

Source Code:-

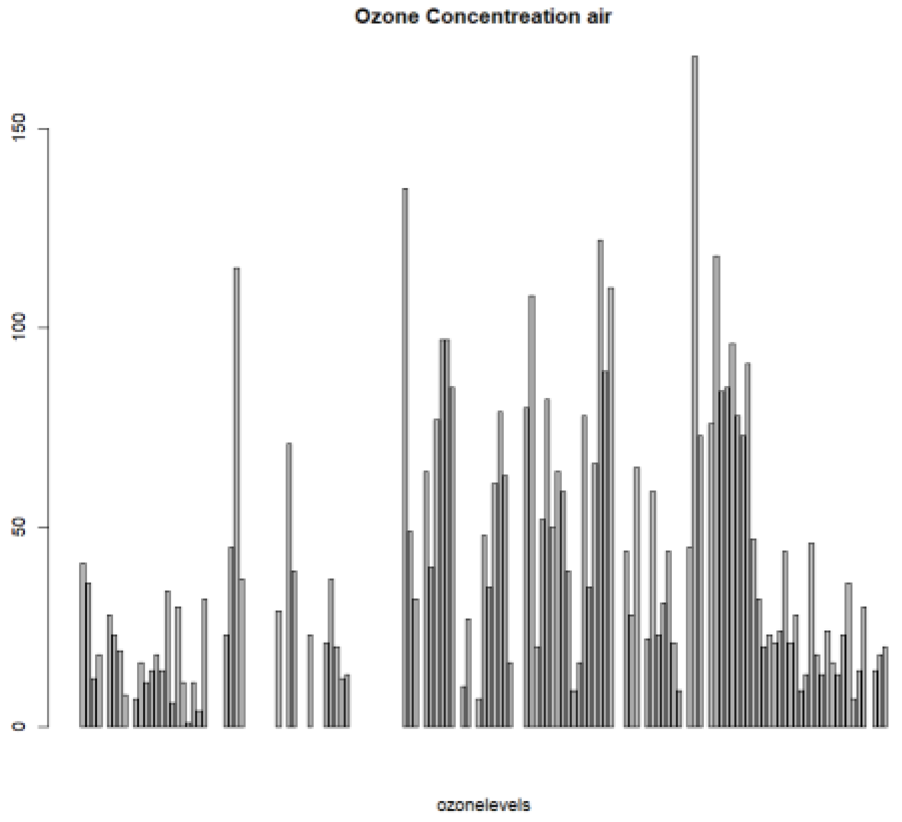
```
>barplot(airquality$Ozone,main='Ozone Concentreation air',xlab="ozonelevels",horiz=TRUE)

>barplot(airquality$Ozone,main='Ozone Concentreation air',xlab="ozonelevels",horiz=FALSE)

>hist(airquality$Temp,main = "Ozone Concentreation air",xlab="Temperature(Fahrenheit)",xlim=c(50,125),col="yellow",freq=TRUE)

>boxplot(airquality$Wind,main="AVG wind speed",xlab="Miles per hrs",ylab="wind",col='yellow',border = 'red',horizontal = TRUE,notch = TRUE)

>data<-matrix(rnorm(25,0,5),nrow=5,ncol=5)

colnames(data)<-paste0("col",1:5)

row.names(data)<-paste0("row",1:5)

heatmap(data)

>mycolors<-colorRampPalette(c("violet","cyan"))

heatmap(data,col=mycolors(100))
```

<u>Output:-</u>

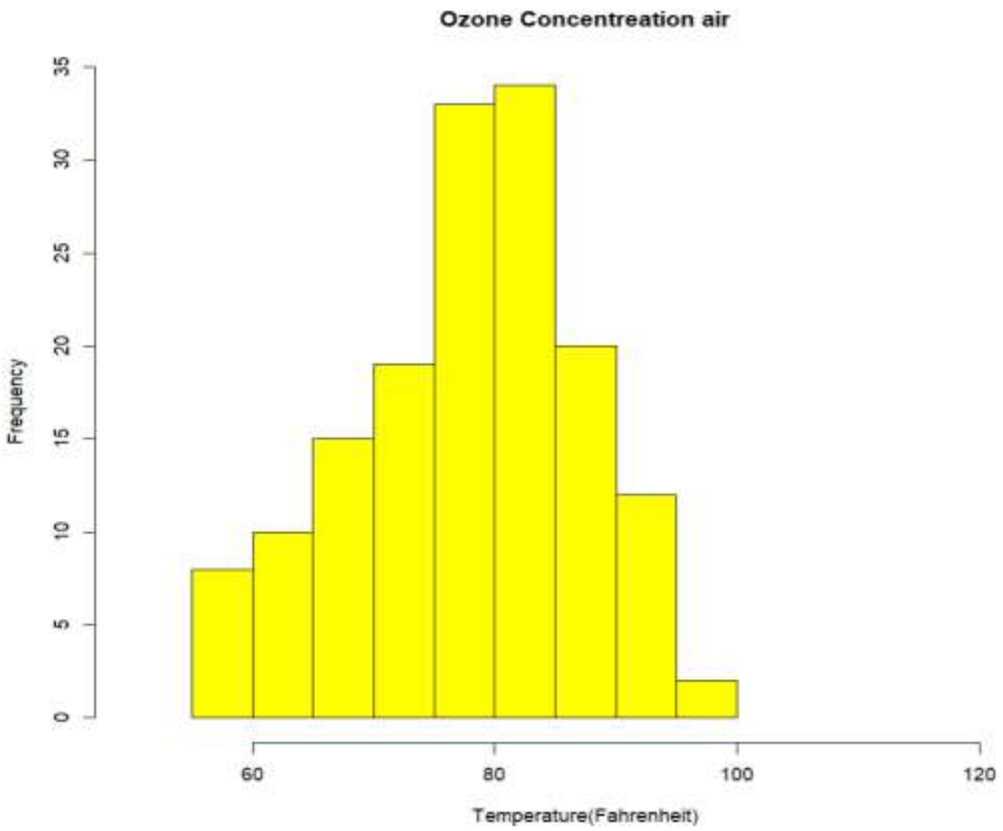
```
> barplot(airquality$Ozone,main='Ozone Concentreation air',xlab="ozonelevels",horiz=TRUE)
```



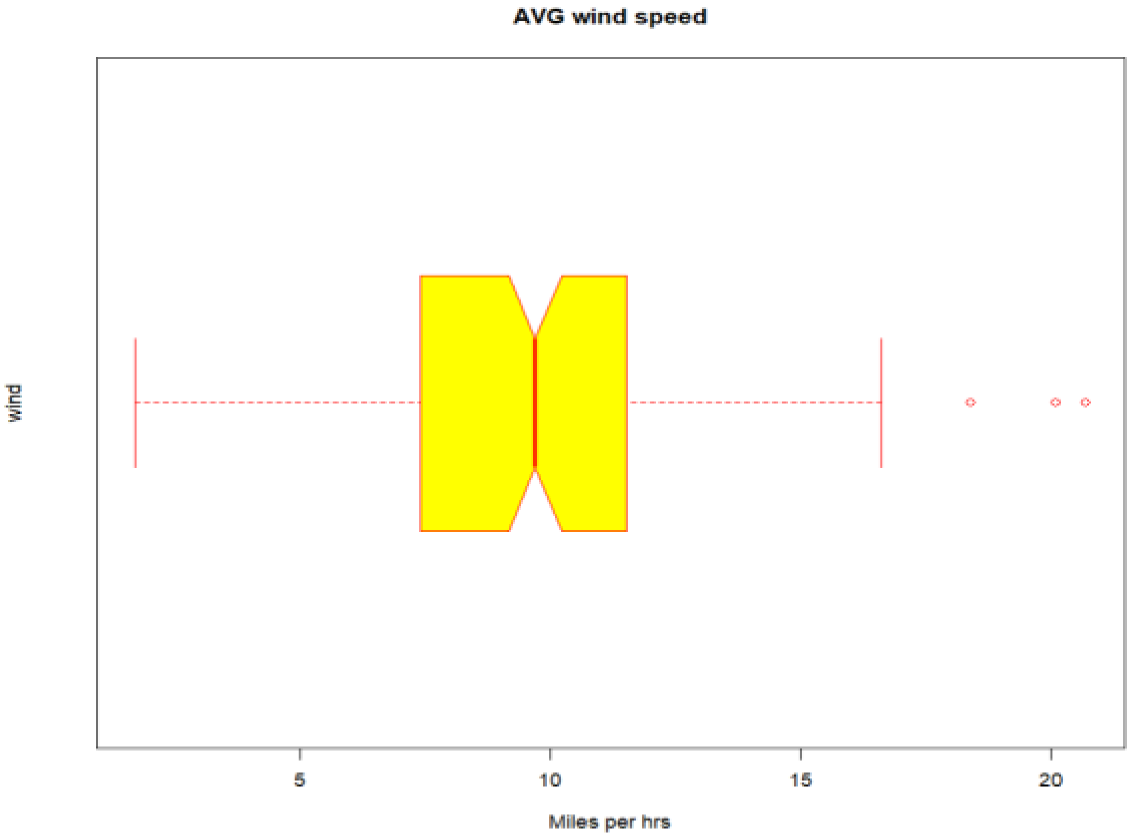
```
> barplot(airquality$Ozone,main='Ozone Concentreation air',xlab="ozonelevels",horiz=FALSE)
```



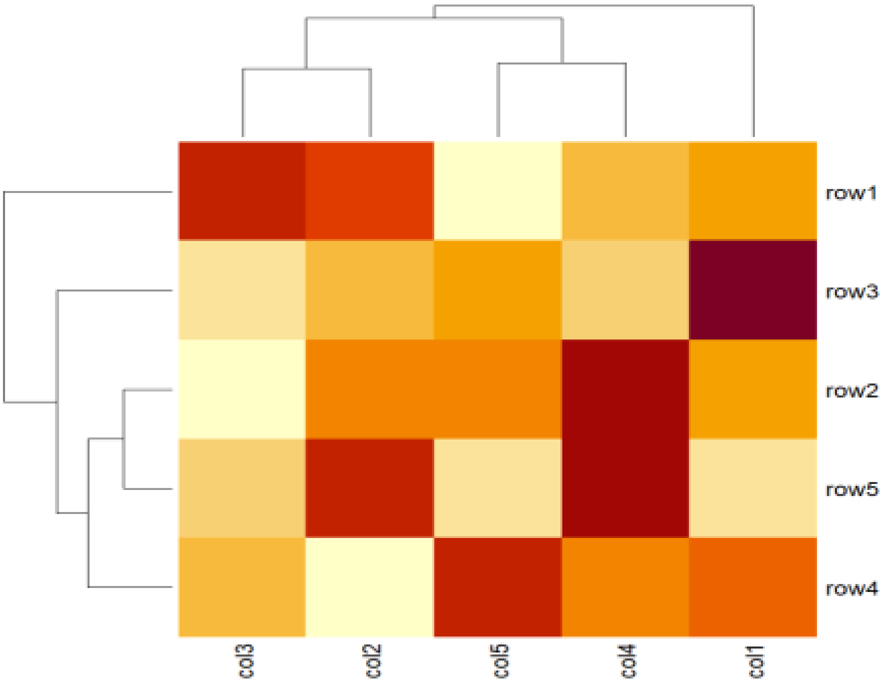
```
> hist(airquality$Temp,main = "Ozone Concentreation air",xlab="Temperature(Fahrenheit)",xlim=c(50,125),col="yellow",freq=TRUE)
```



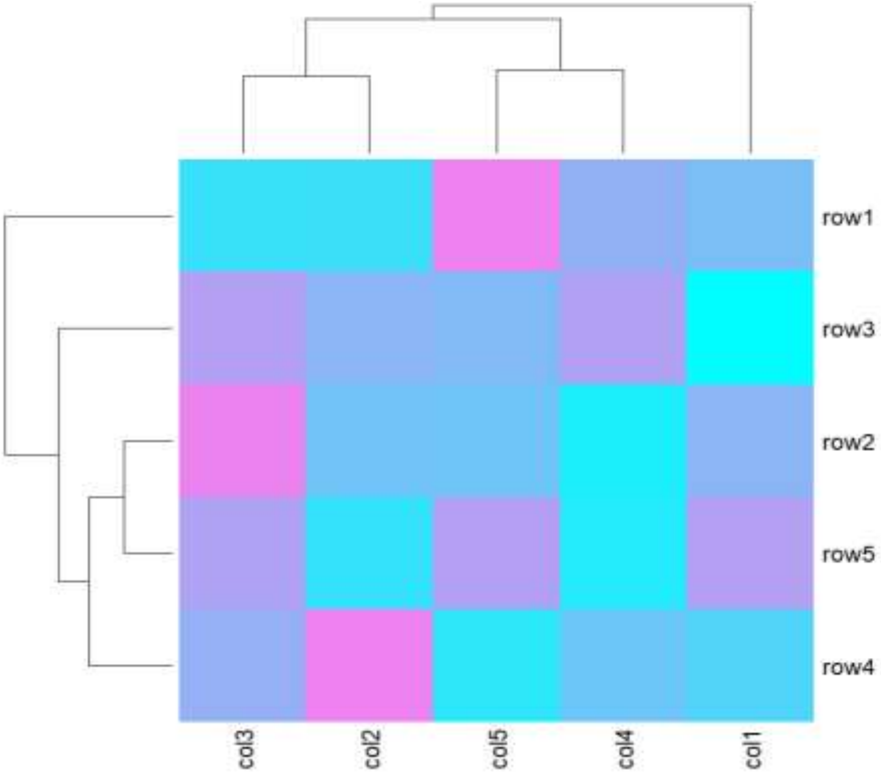
```
> boxplot(airquality$Wind,main="AVG wind speed",xlab="Miles per hrs",ylab="wind",col='yellow',border = 'red',horizontal = TRUE,notch = TRUE)
```



```
> data<-matrix(rnorm(25,0,5),nrow=5,ncol=5)

> colnames(data)<-paste0("col",1:5)

> row.names(data)<-paste0("row",1:5)

> heatmap(data)
```



```
>mycolors<-colorRampPalette(c("violet","cyan"))

heatmap(data,col=mycolors(100))
```

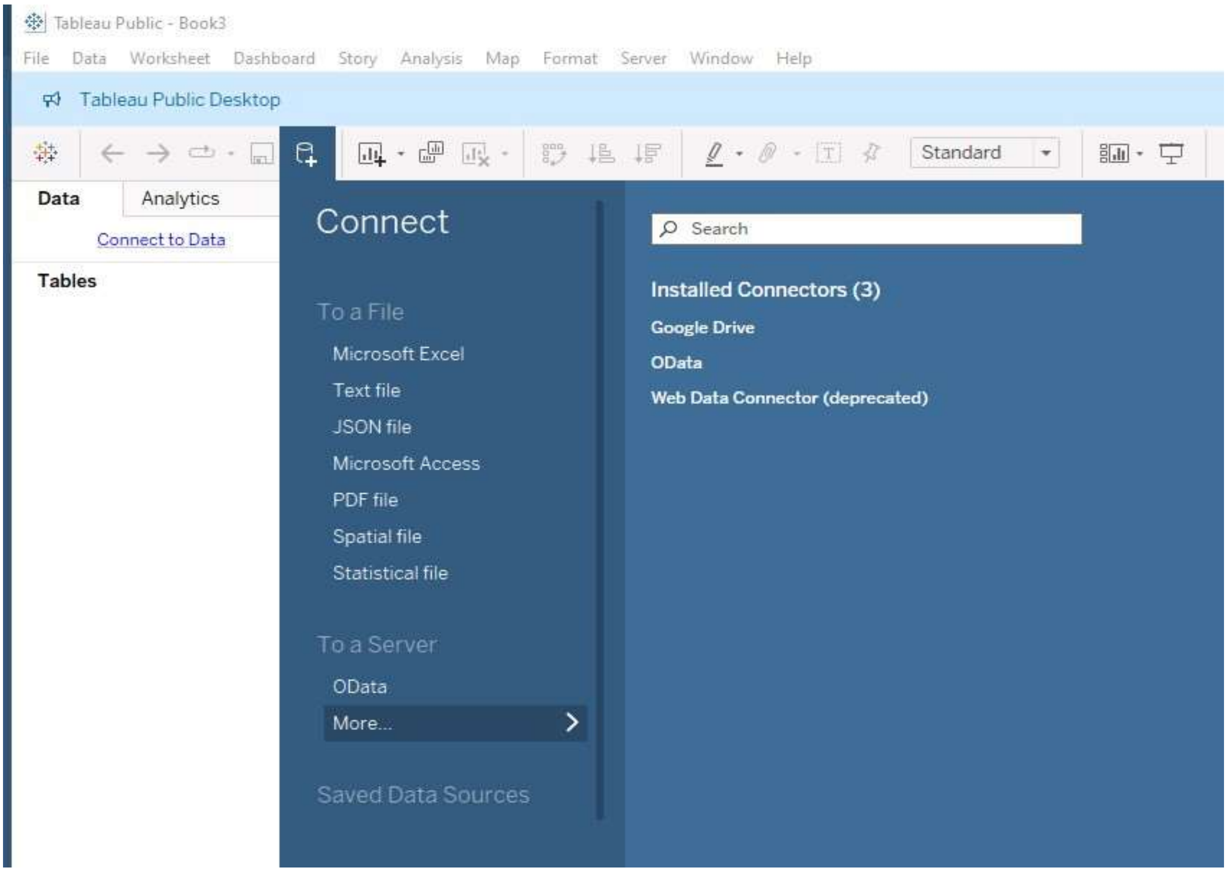


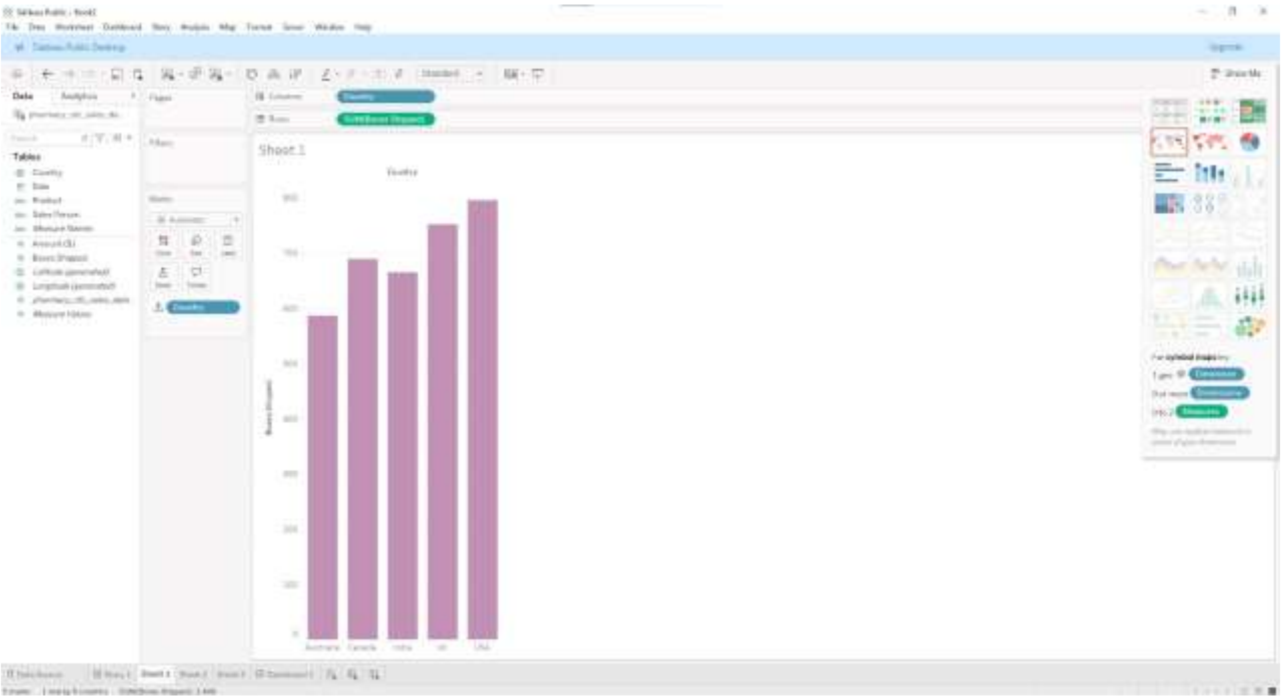
Practical no. 3 Connecting to Data and preparing data for visualization in Tableau

Aim: To connect Tableau to different data sources (CSV, Excel, Database) and perform **data cleaning and preparation** for visualization.

Steps:

1. **Open Tableau Desktop / Tableau Public – Start a new project.**





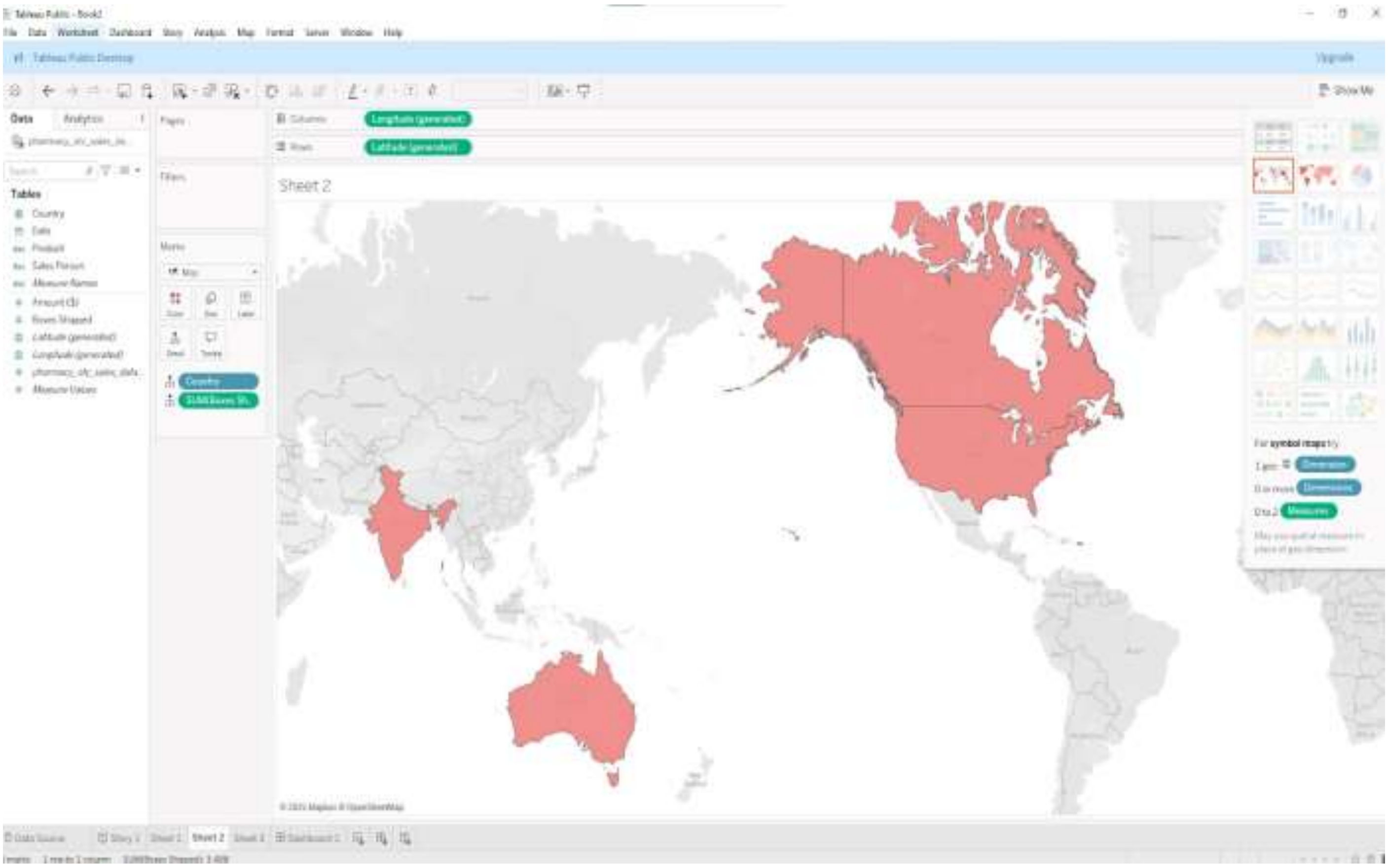


Tableau Public - Book2
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Tableau Public Desktop
Upgrade
Show Me
Data Analytics
Pages
Filters
Marks
Map
Columns Longitude (generated)
Rows Latitude (generated)
Sheet 2
Tables
Country
Date
Product
Sales Person
Measure Names
Amount ($)
Boxes Shipped
Latitude (generated)
Longitude (generated)
Measure Values
Color Size Label
Detail Tooltip
Country
SUM(Boxes Sh..
Dimension
Dimensions
Measures
Data Source Story 1 Sheet 1 Sheet 2 Sheet 3 Dashboard 1

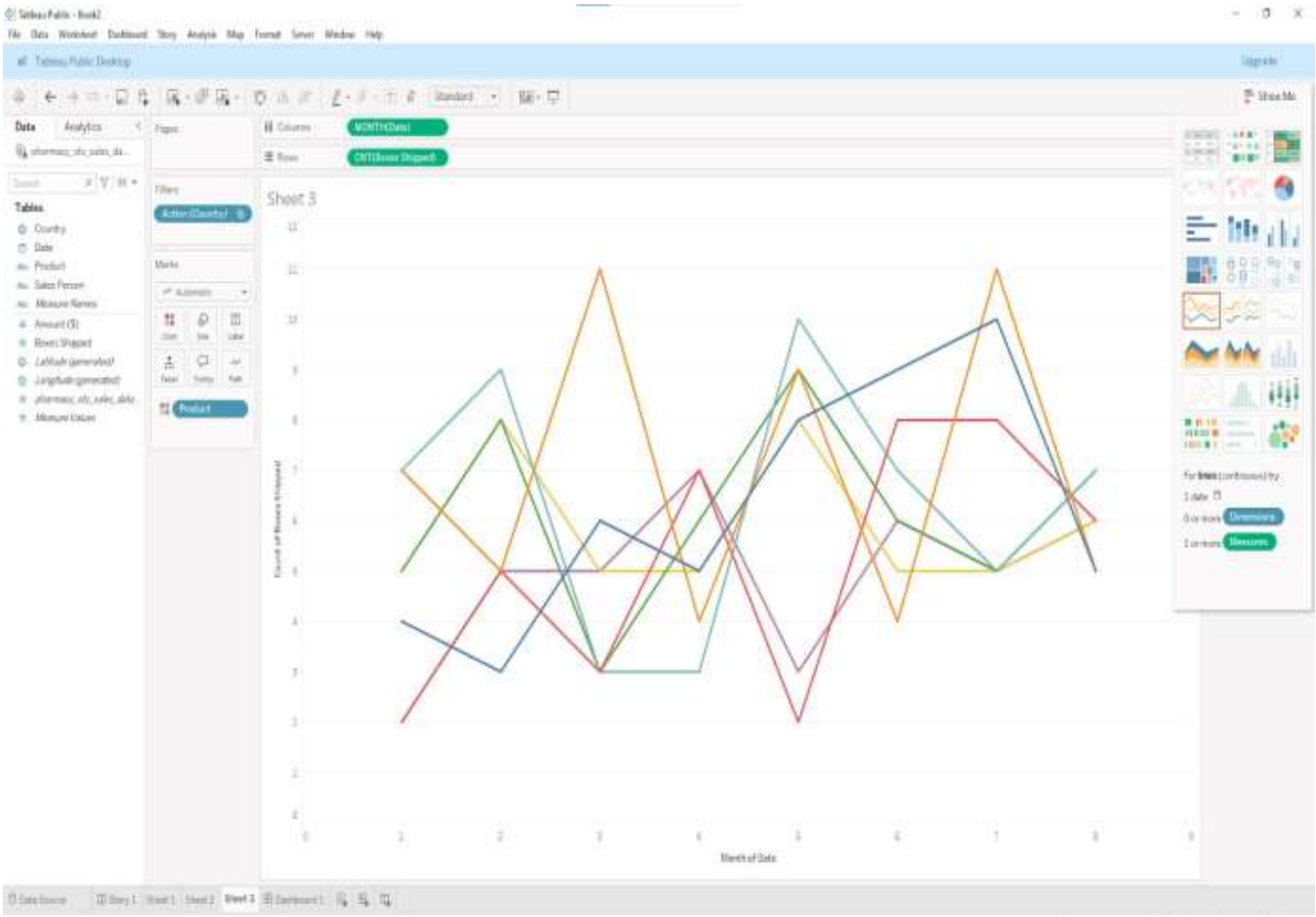
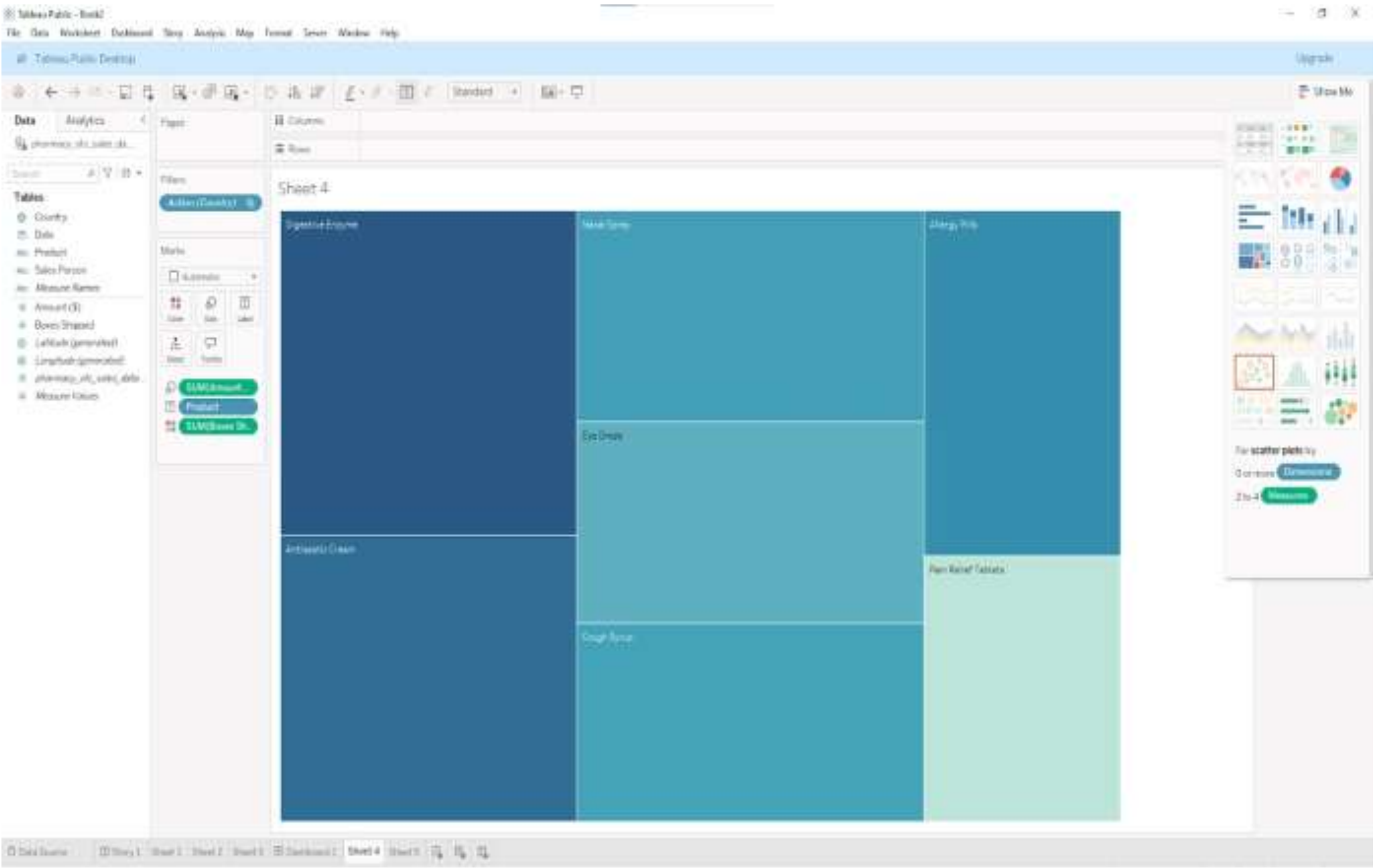
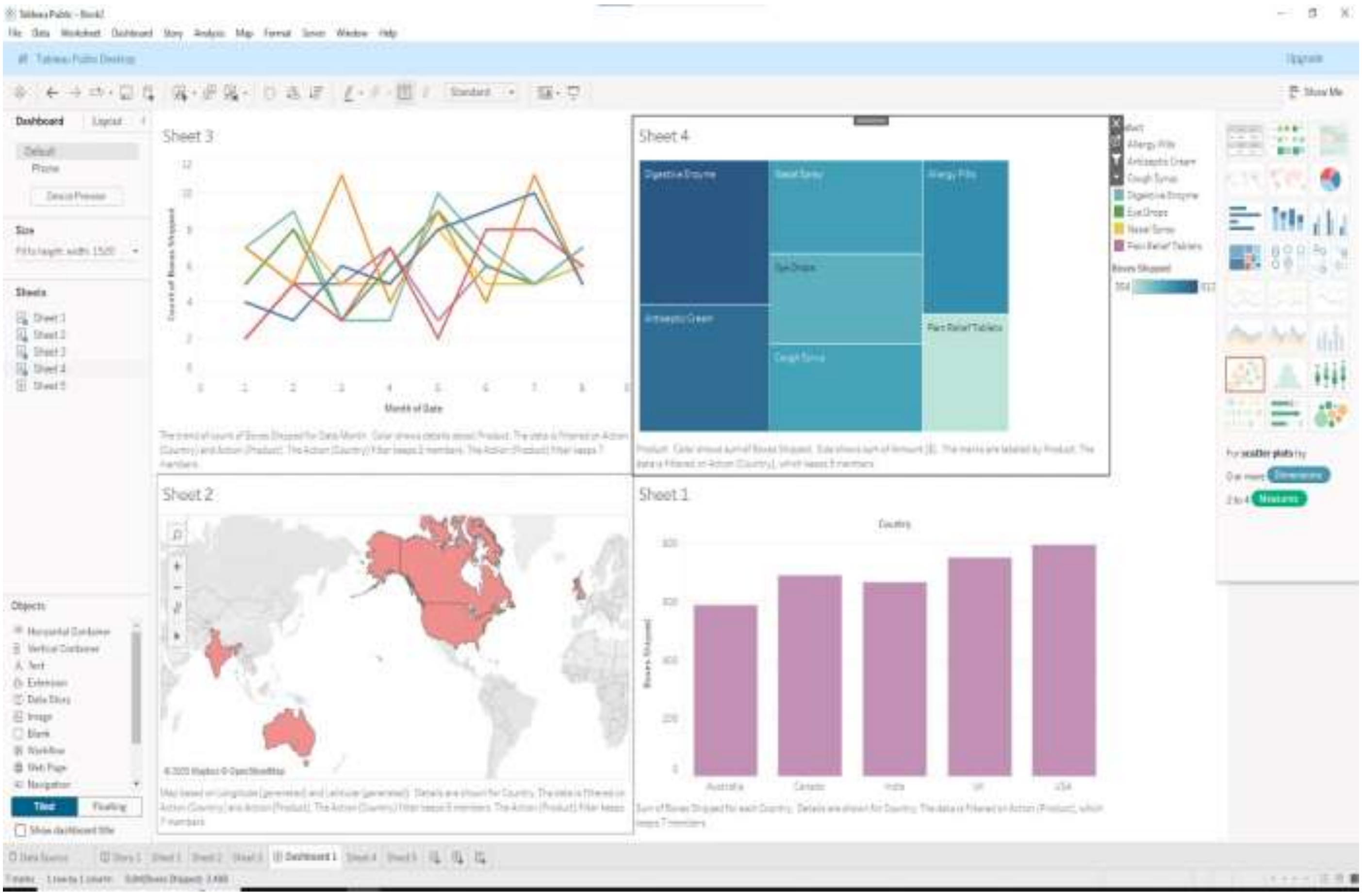


Tableau Public - Book2
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Tableau Public Desktop
Upgrade
Standard
Show Me
Data Analytics
Pages
Filters
Action (Country)
Marks
Automatic
Columns MONTH(Date)
Rows CNT(Boxes Shipped)
Sheet 3
Tables
Country
Date
Product
Sales Person
Measure Names
Amount ($)
Boxes Shipped
Latitude (generated)
Longitude (generated)
Measure Values
Color Size Label
Detail Tooltip Path
Product
Count of Boxes Shipped
Month of Date
0
1
2
3
4
5
6
7
8
9
Dimensions
Measures
Data Source Story 1 Sheet 1 Sheet 2 Sheet 3 Dashboard 1



DASHBOARD



The trend of count of Boxes Shipped for Date Month. Color shows details about Product. The data is filtered on Action (Country) and Action (Product). The Action (Country) filter keeps 5 members. The Action (Product) filter keeps 7 members.

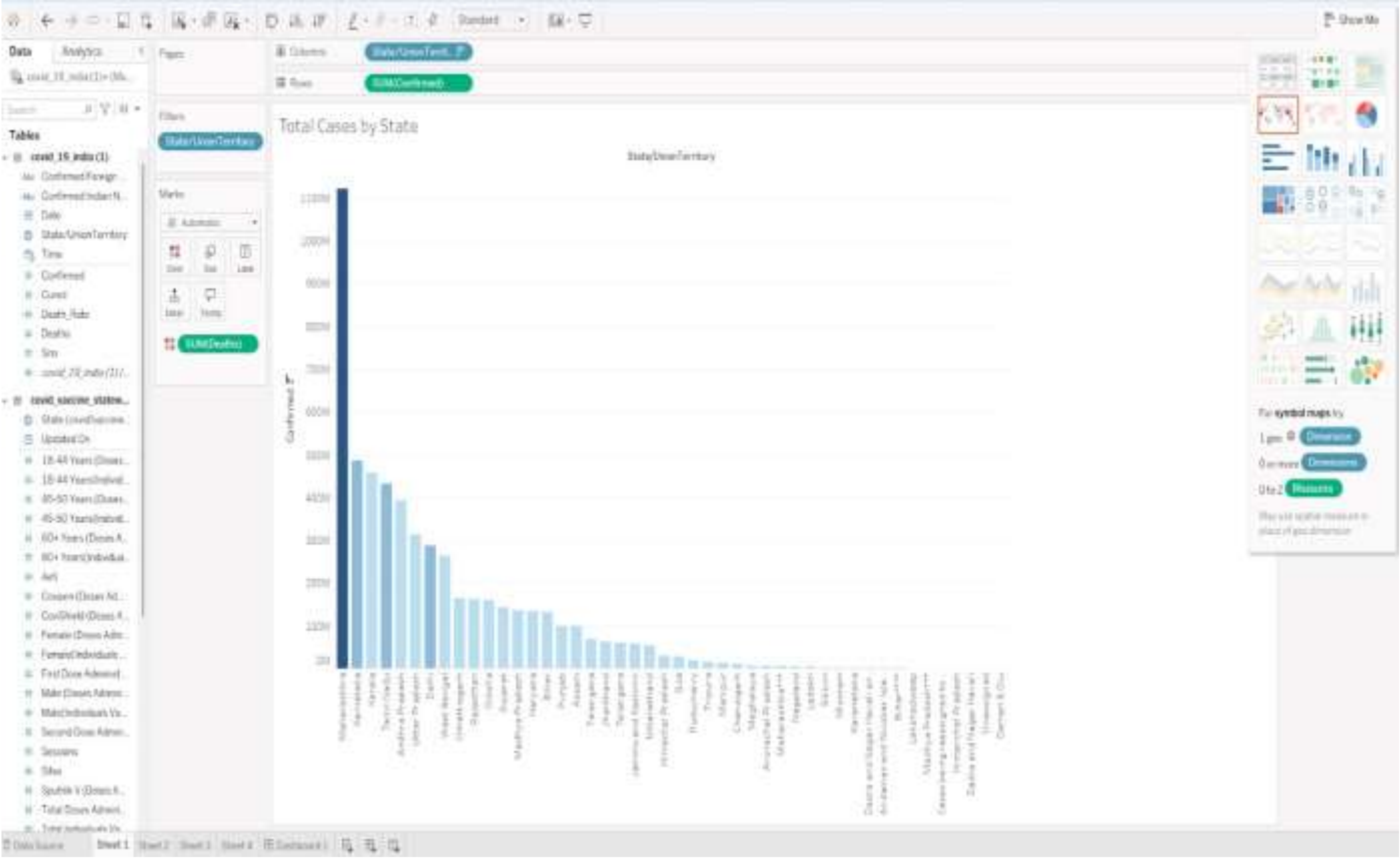
Product. Color shows sum of Boxes Shipped. Size shows sum of Amount ($). The marks are labeled by Product. The data is filtered on Action (Country), which keeps 5 members.

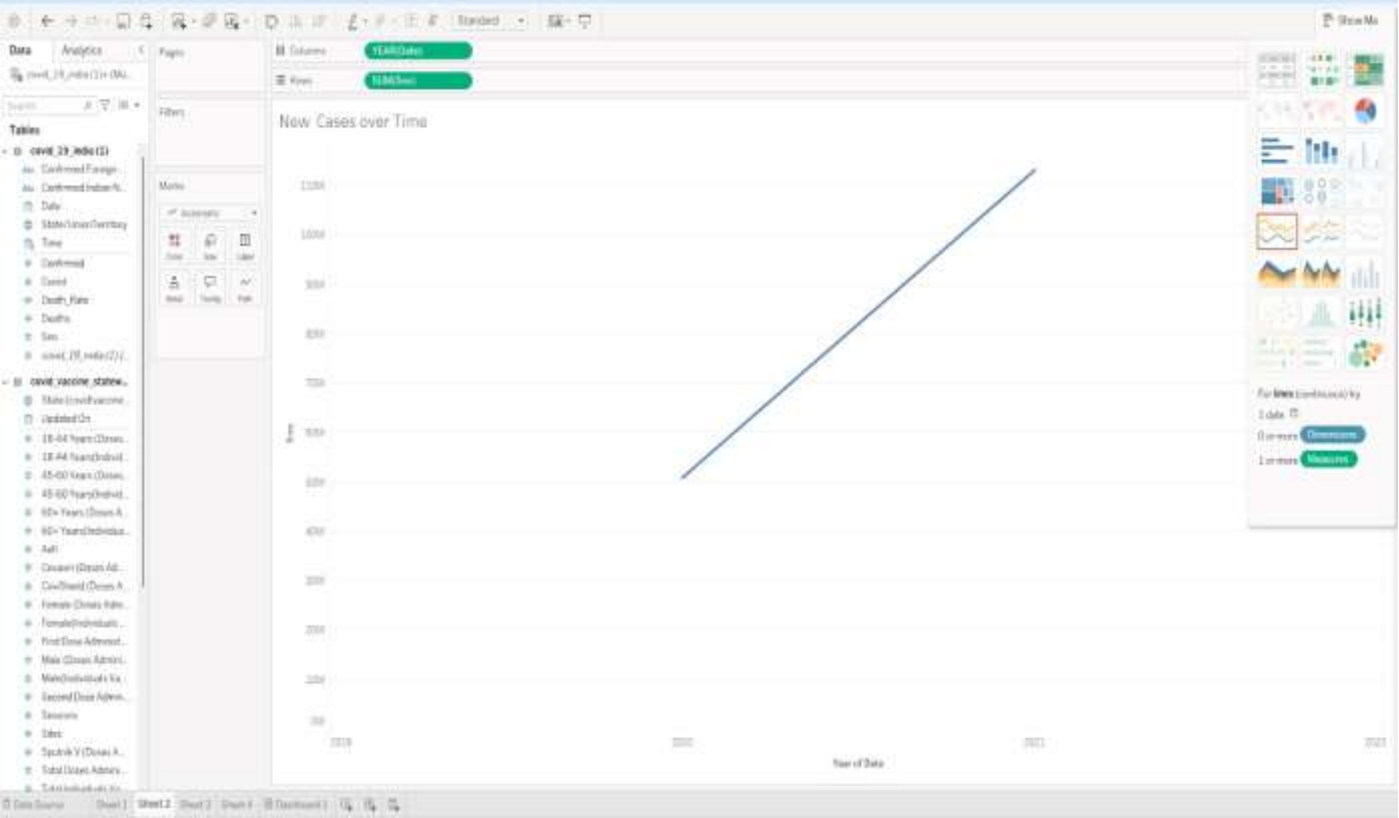
Map based on Longitude (generated) and Latitude (generated). Details are shown for Country. The data is filtered on Action (Country) and Action (Product). The Action (Country) filter keeps 5 members. The Action (Product) filter keeps 7 members.

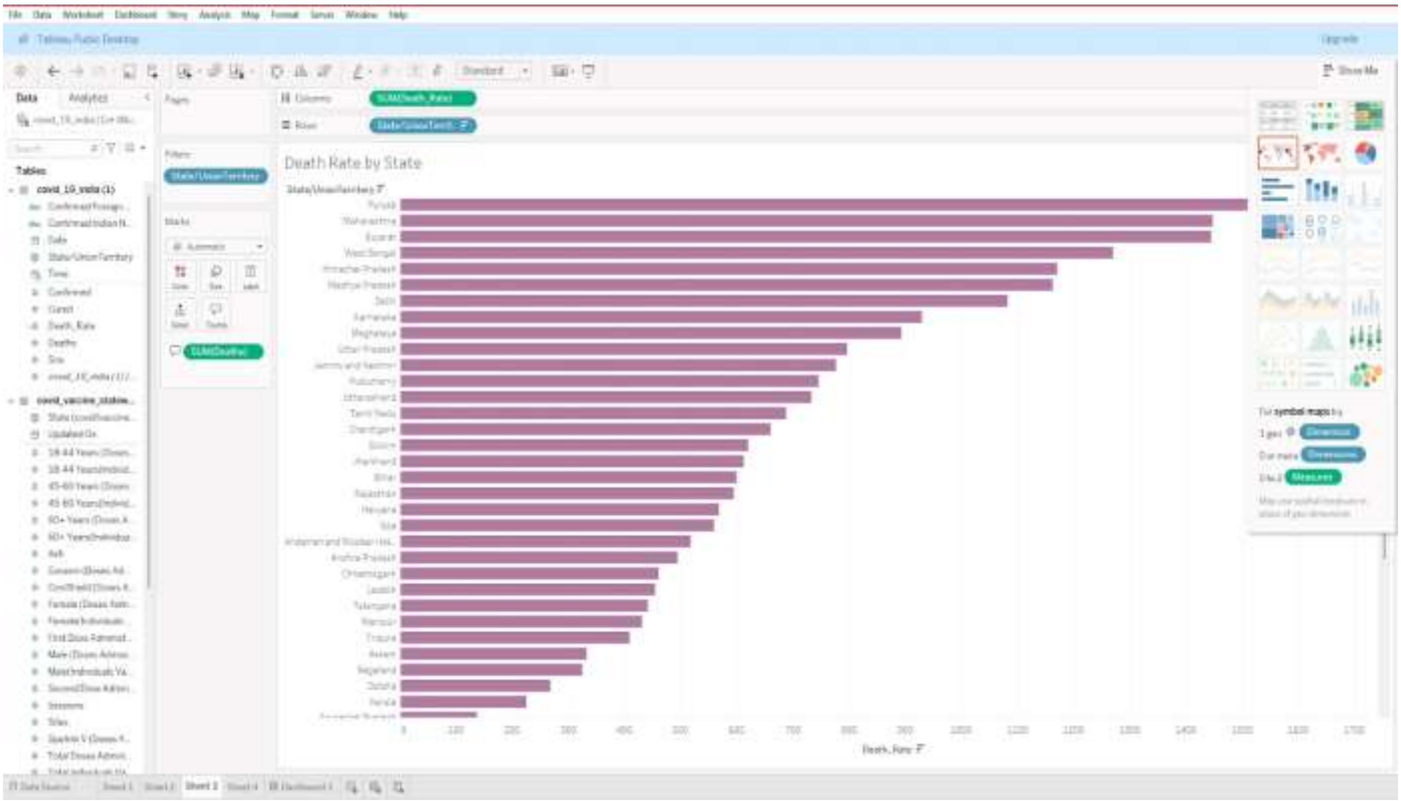
Sum of Boxes Shipped for each Country. Details are shown for Country. The data is filtered on Action (Product), which keeps 7 members.

Practical no 4. Use Data aggregation and statistical functions in Tableau

Aim: To apply **aggregation functions (SUM, AVG, COUNT)** and statistical functions (MIN, MAX, MEDIAN) in Tableau for meaningful data analysis.





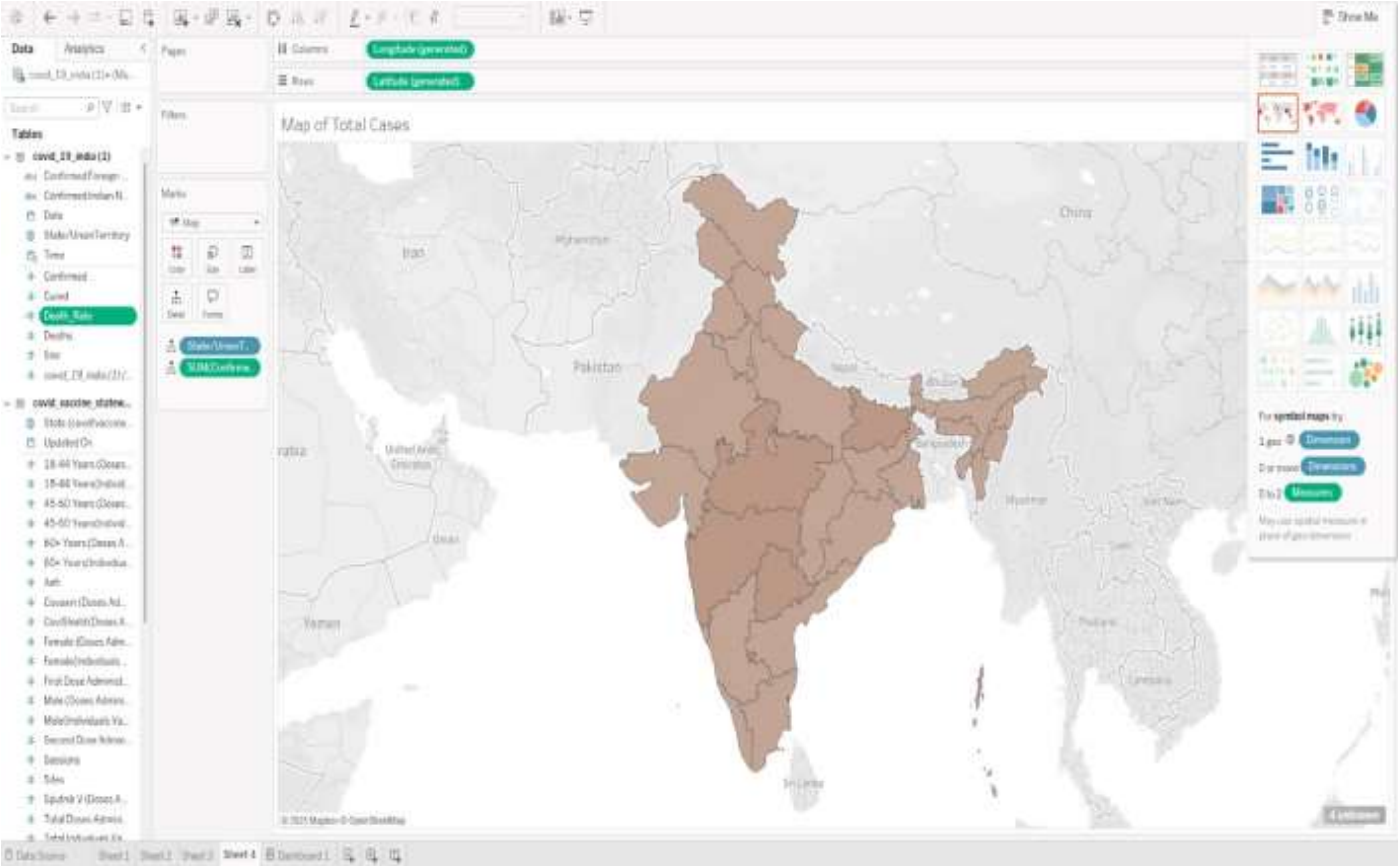


Columns: SUM(Death_Rate)

Rows: State/UnionTerritory

Filters: State/UnionTerritory

Death Rate by State



Columns: Longitude (generated)

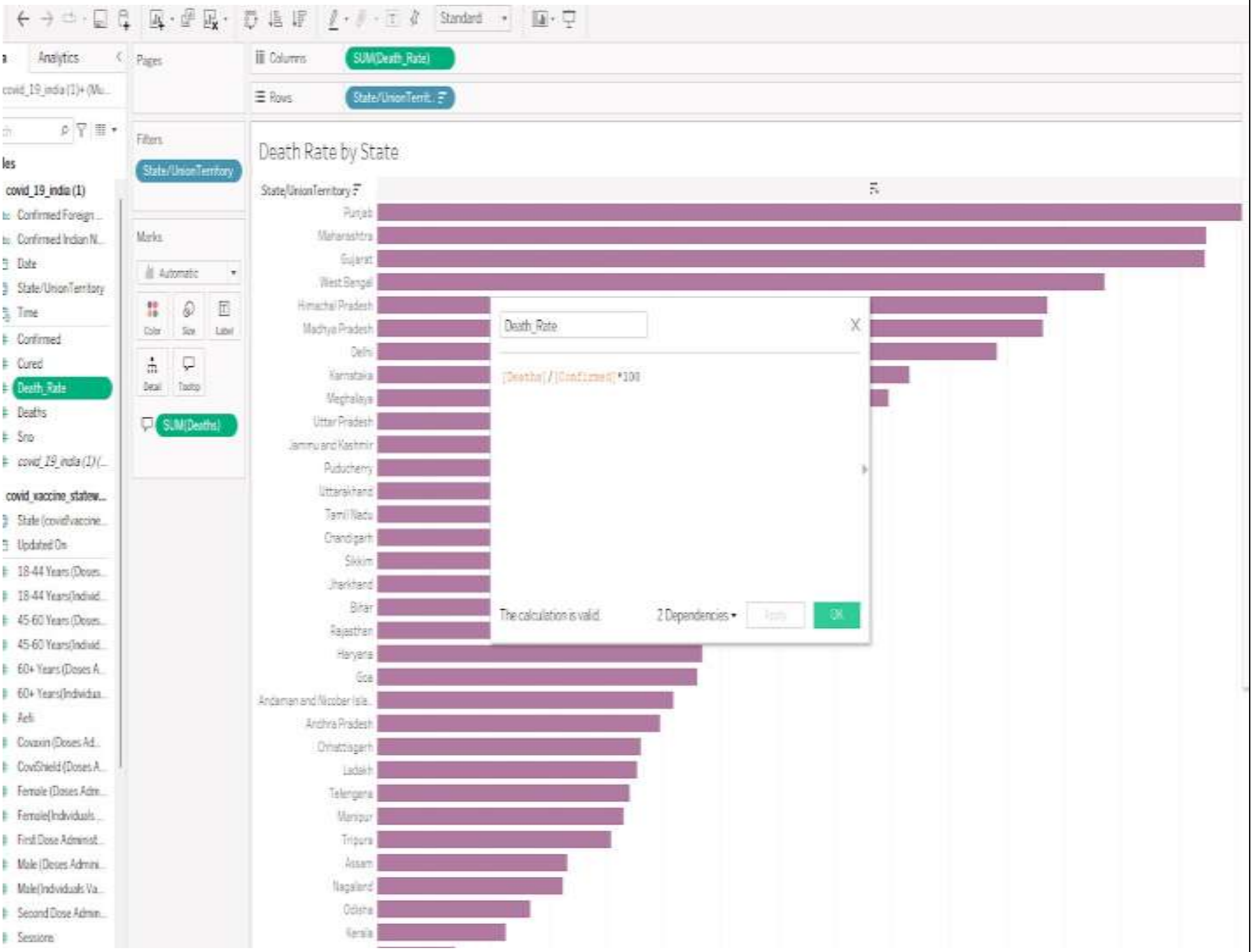
Rows: Latitude (generated)

Marks: Map — State/UnionT..., SUM(Confirmed)

Map of Total Cases

Practical no 5. Show Data Visualization using Tableau

Aim: To create **interactive data visualizations** such as bar charts, line charts, scatter plots, and pie charts using Tableau.



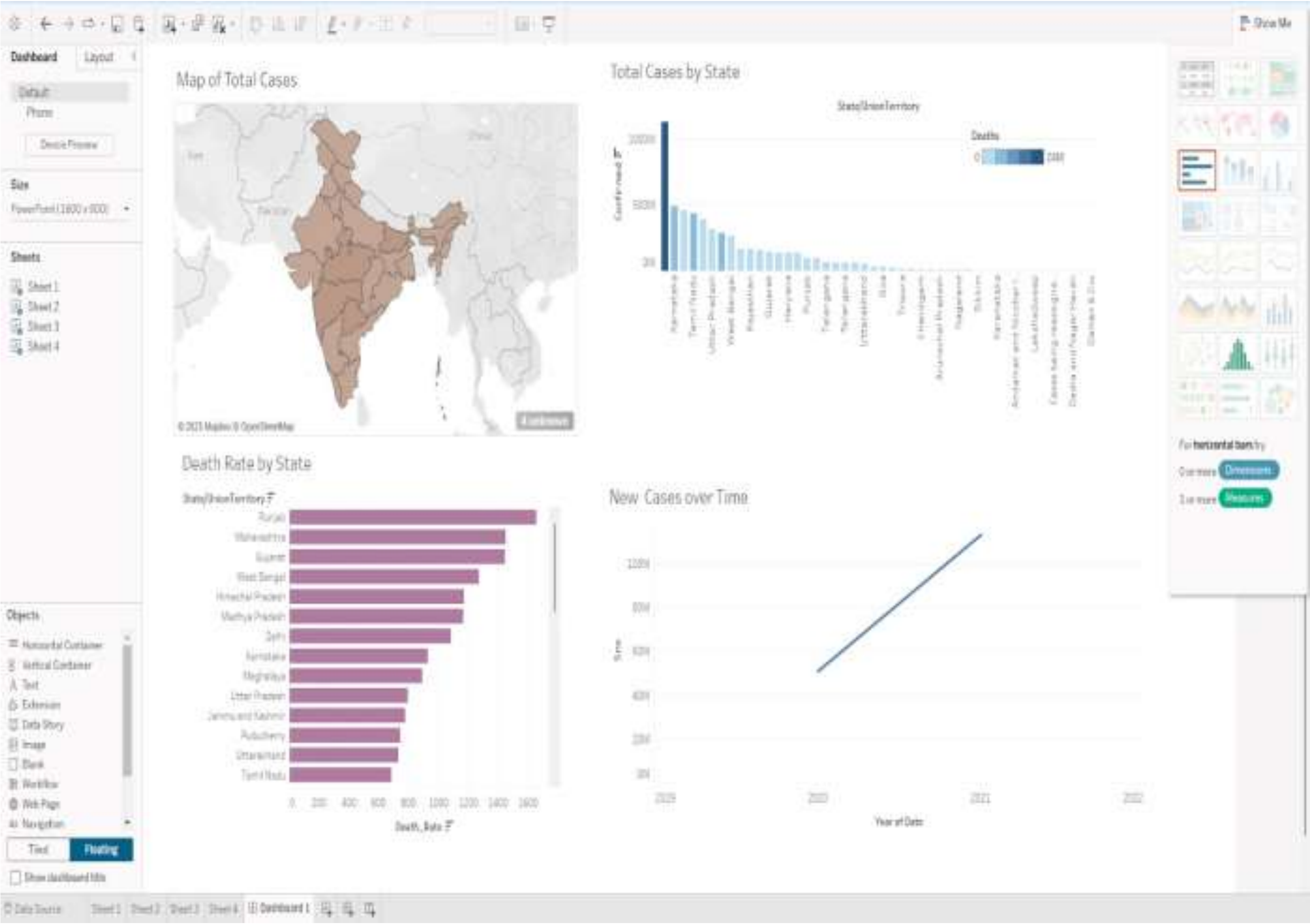
```
[Deaths]/[Confirmed]*100
```

The calculation is valid.

Practical no 6. Use dashboards of Tableau

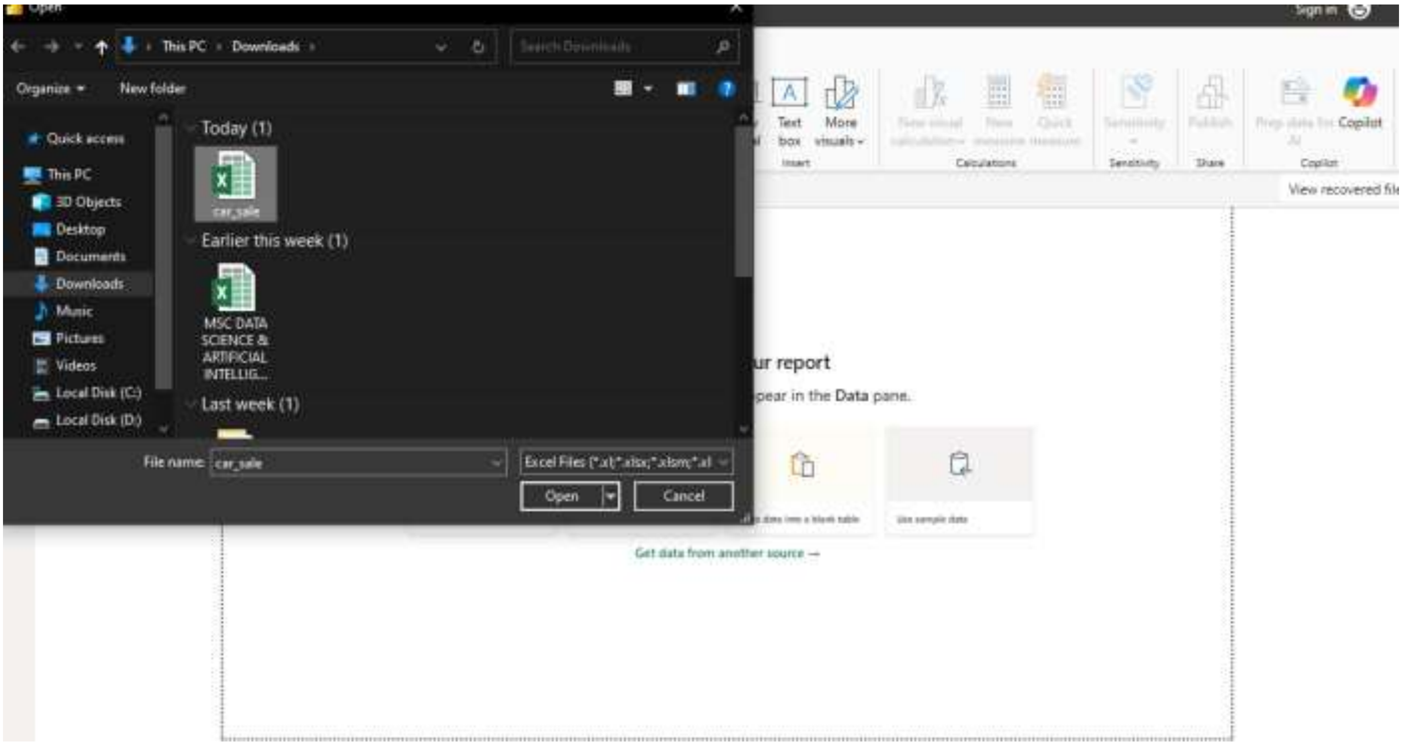
Aim: To design and implement **dashboards** in Tableau for combining multiple visualizations and presenting insights effectively.

DASHBOARD OF COVID-19(2020-2022)



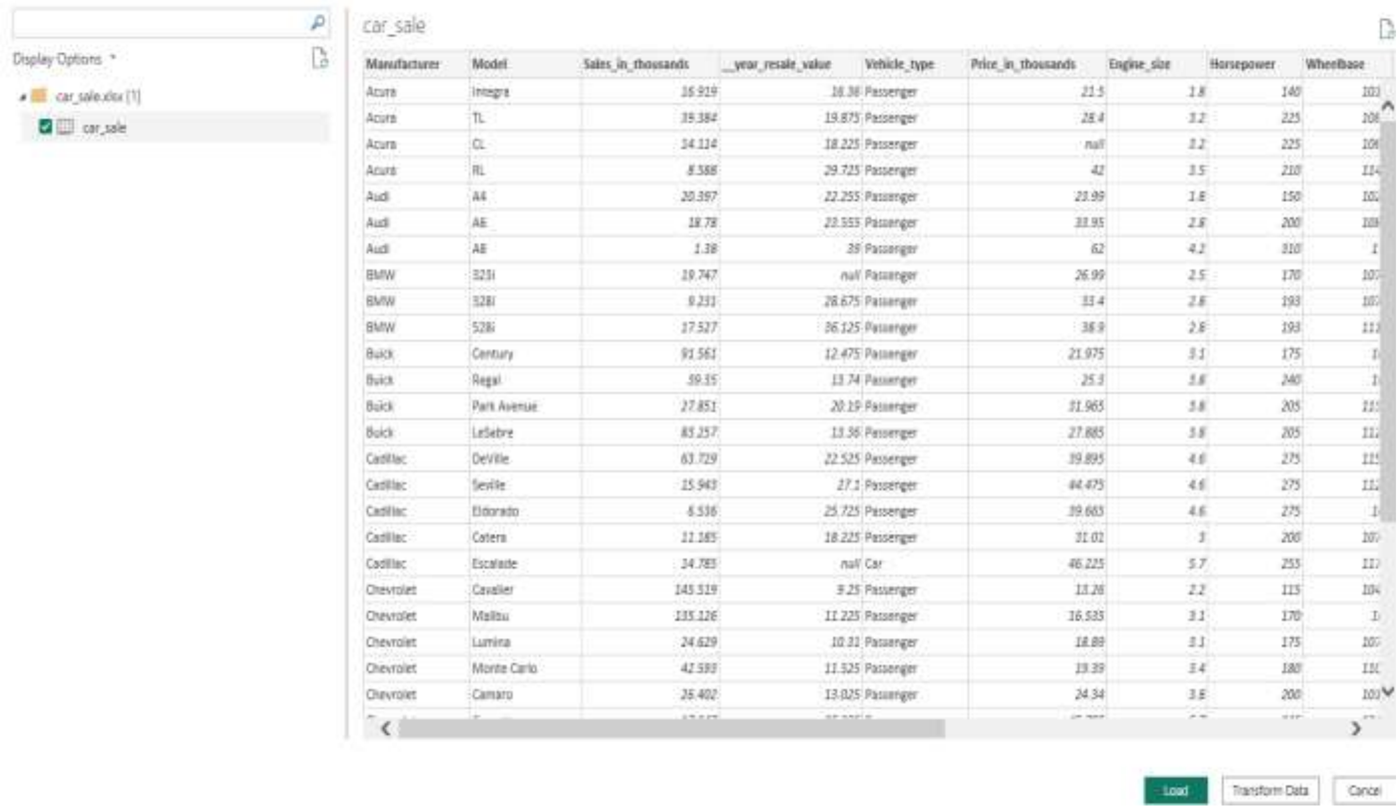
Practical no 7. Show Data Visualization using Power BI

Aim: To perform **data visualization** using Microsoft Power BI and generate interactive reports from a given dataset.



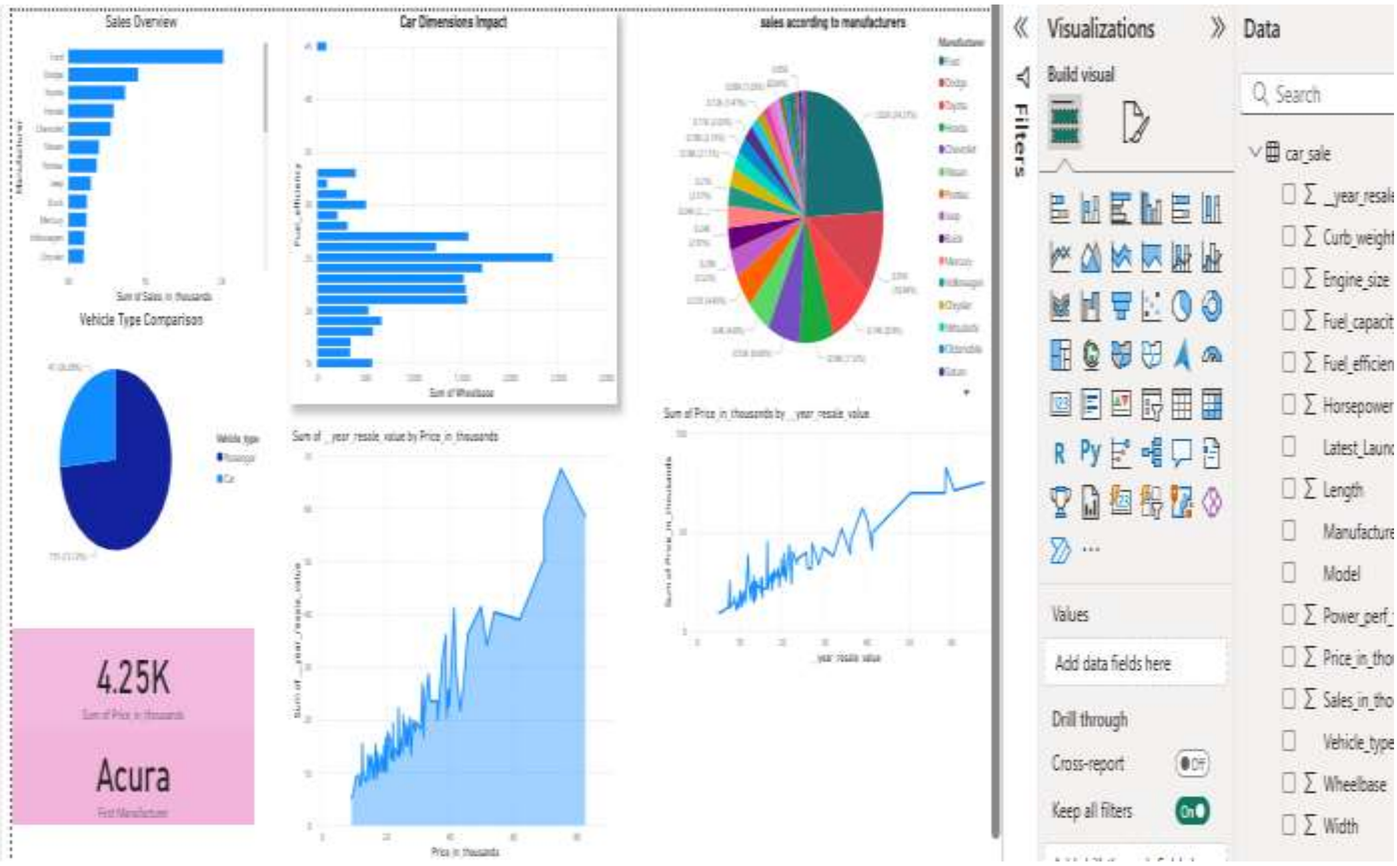
File name: car_sale

Excel Files (*.xl;*.xlsx;*.xlsm;*.xl



car_sale

Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thousands	Engine_size	Horsepower	Wheelbase
Acura	Integra	16.919	16.36	Passenger	21.5	1.8	140	101
Acura	TL	39.384	19.875	Passenger	28.4	3.2	225	108
Acura	CL	14.114	18.225	Passenger	null	3.2	225	106
Acura	RL	8.588	29.725	Passenger	42	3.5	210	114
Audi	A4	20.397	22.255	Passenger	23.99	1.8	150	102
Audi	A6	18.78	23.555	Passenger	33.95	2.8	200	108
Audi	A8	1.38	39	Passenger	62	4.2	310	1
BMW	323i	19.747	null	Passenger	26.99	2.5	170	107
BMW	328i	9.231	28.675	Passenger	33.4	2.8	193	107
BMW	528i	17.527	36.125	Passenger	38.9	2.8	193	111
Buick	Century	91.561	12.475	Passenger	21.975	3.1	175	1
Buick	Regal	39.35	13.74	Passenger	25.3	3.8	240	1
Buick	Park Avenue	27.851	20.19	Passenger	31.965	3.8	205	11
Buick	LeSabre	83.257	13.36	Passenger	27.885	3.8	205	112
Cadillac	DeVille	63.729	22.525	Passenger	39.895	4.6	275	115
Cadillac	Seville	15.943	27.1	Passenger	44.475	4.6	275	112
Cadillac	Eldorado	6.536	25.725	Passenger	39.665	4.6	275	1
Cadillac	Catera	11.185	18.225	Passenger	31.01	3	200	107
Cadillac	Escalade	14.785	null	Car	46.225	5.7	255	117
Chevrolet	Cavalier	145.519	9.25	Passenger	13.26	2.2	115	104
Chevrolet	Malibu	135.126	11.225	Passenger	16.535	3.1	170	1
Chevrolet	Lumina	24.629	10.31	Passenger	18.89	3.1	175	107
Chevrolet	Monte Carlo	42.593	11.525	Passenger	19.39	3.4	180	110
Chevrolet	Camaro	26.402	13.025	Passenger	24.34	3.8	200	101

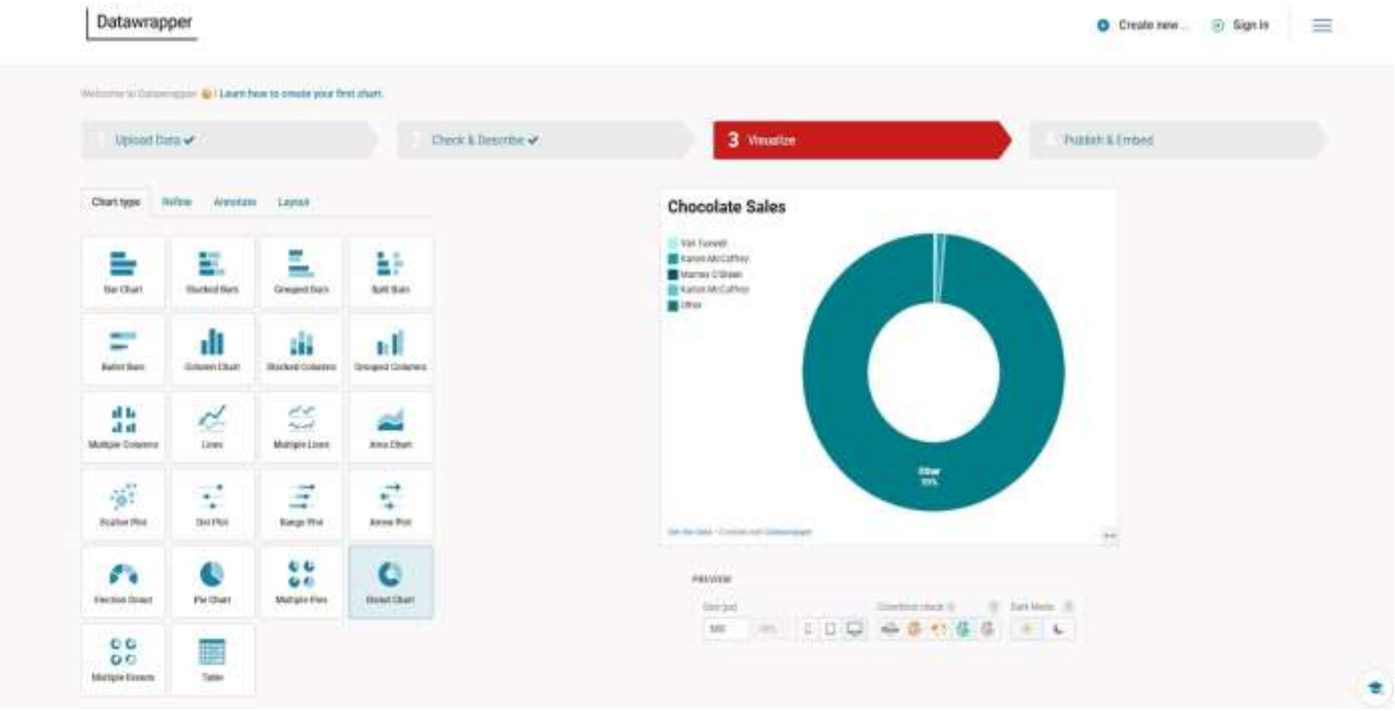


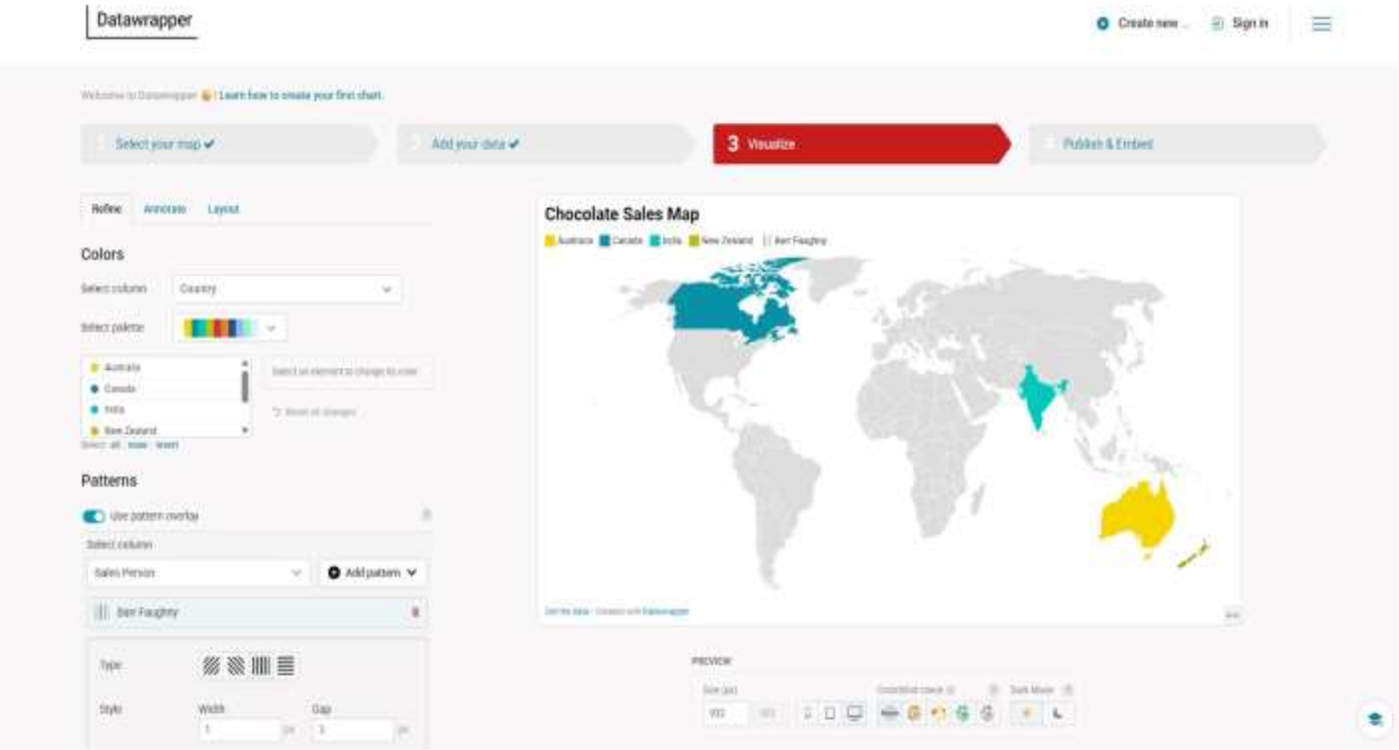
Cross-report Off

Keep all filters On

Practical No 8. Show Data Visualization using DataWrapper

Aim: To use **DataWrapper** for creating quick, interactive charts, maps, and tables and publishing them online.





Select column: Country

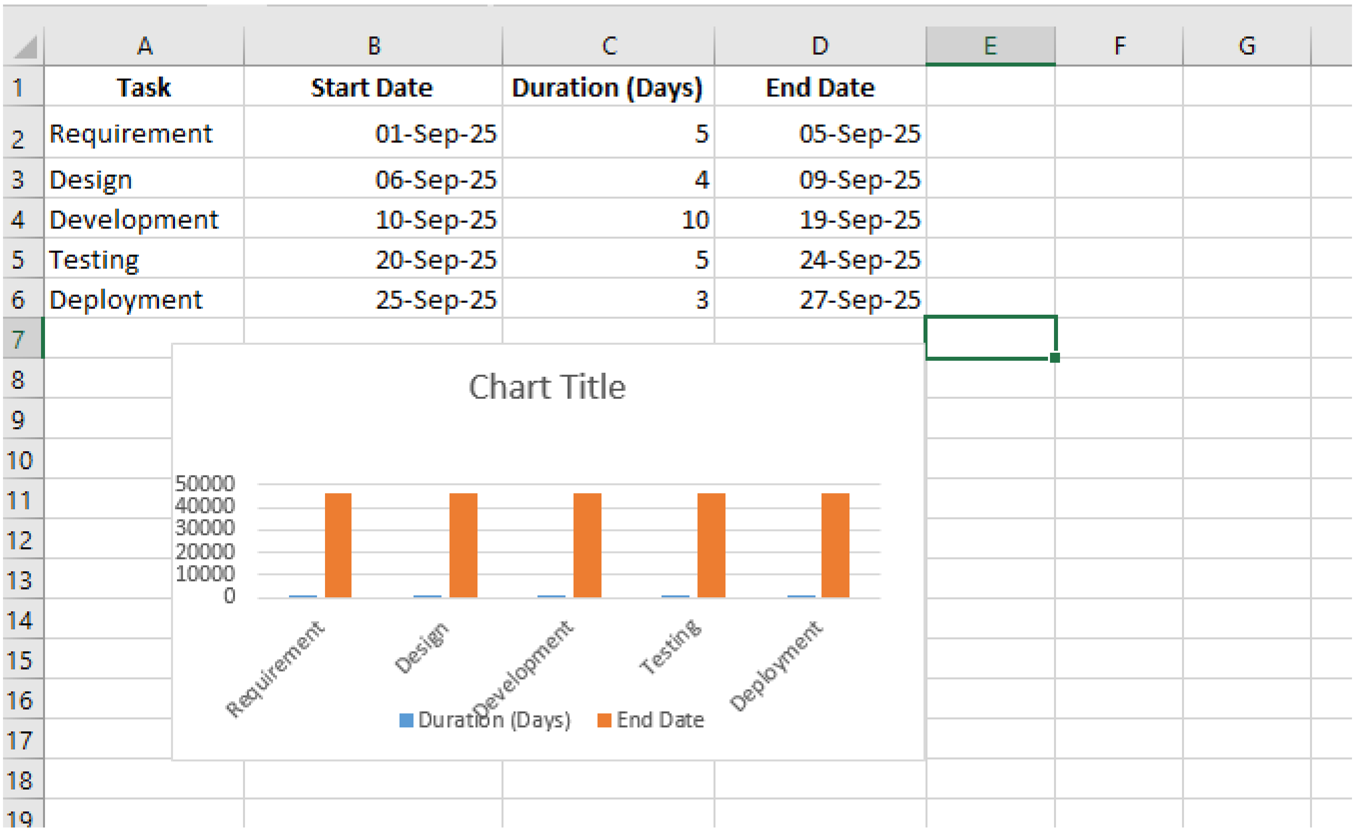
Select column: Sales Person

Style: Width 1 px Gap 3 px

Practical no 9. Show Data Visualization using Gantt Chart

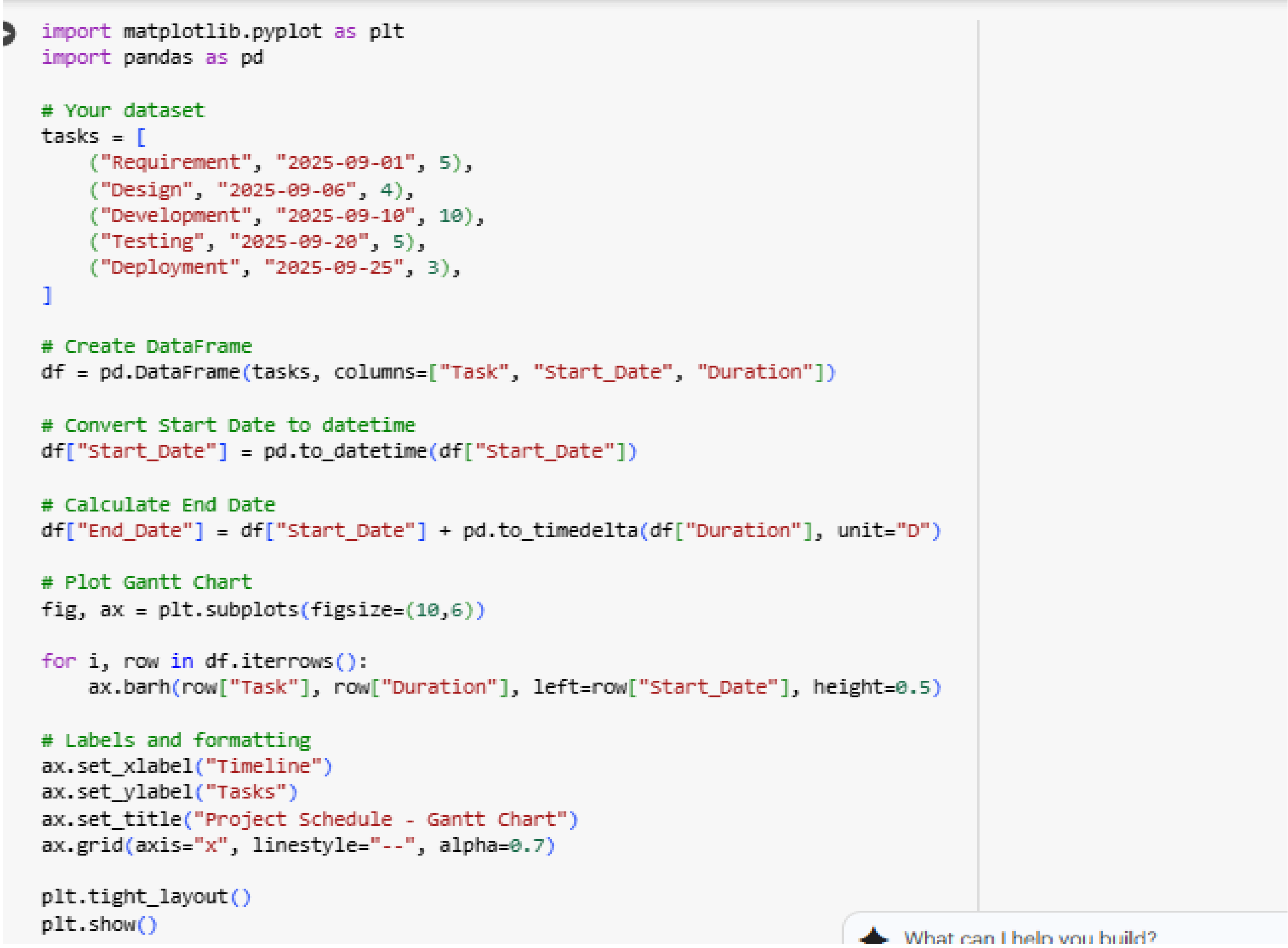
Aim: To visualize **project scheduling and task timelines** using a Gantt Chart for project management.

- **Excel sheet:**



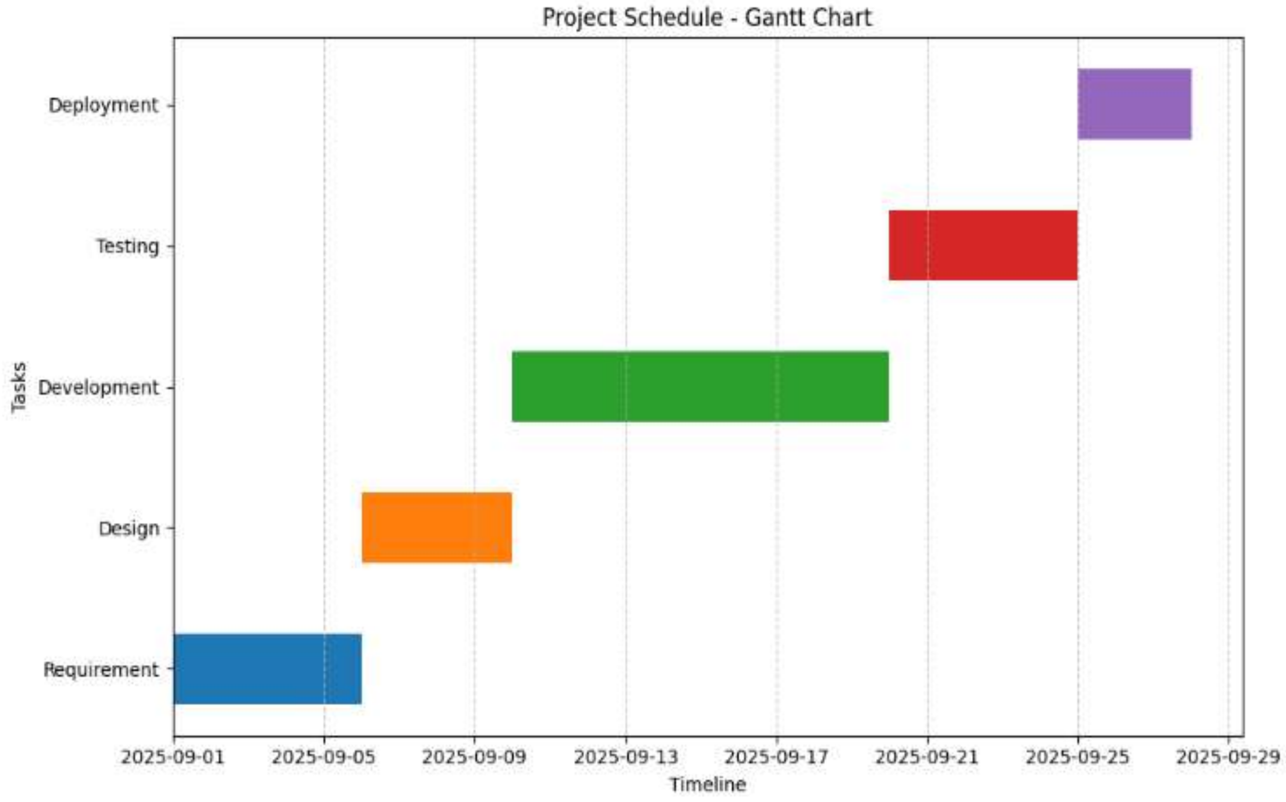
Task	Start Date	Duration (Days)	End Date
Requirement	01-Sep-25	5	05-Sep-25
Design	06-Sep-25	4	09-Sep-25
Development	10-Sep-25	10	19-Sep-25
Testing	20-Sep-25	5	24-Sep-25
Deployment	25-Sep-25	3	27-Sep-25

- **Code:**



```
import matplotlib.pyplot as plt
import pandas as pd

# Your dataset
tasks = [
    ("Requirement", "2025-09-01", 5),
    ("Design", "2025-09-06", 4),
    ("Development", "2025-09-10", 10),
    ("Testing", "2025-09-20", 5),
    ("Deployment", "2025-09-25", 3),
]

# Create DataFrame
df = pd.DataFrame(tasks, columns=["Task", "Start_Date", "Duration"])

# Convert Start Date to datetime
df["Start_Date"] = pd.to_datetime(df["Start_Date"])

# Calculate End Date
df["End_Date"] = df["Start_Date"] + pd.to_timedelta(df["Duration"], unit="D")

# Plot Gantt Chart
fig, ax = plt.subplots(figsize=(10,6))

for i, row in df.iterrows():
    ax.barh(row["Task"], row["Duration"], left=row["Start_Date"], height=0.5)

# Labels and formatting
ax.set_xlabel("Timeline")
ax.set_ylabel("Tasks")
ax.set_title("Project Schedule - Gantt Chart")
ax.grid(axis="x", linestyle="--", alpha=0.7)

plt.tight_layout()
plt.show()
```

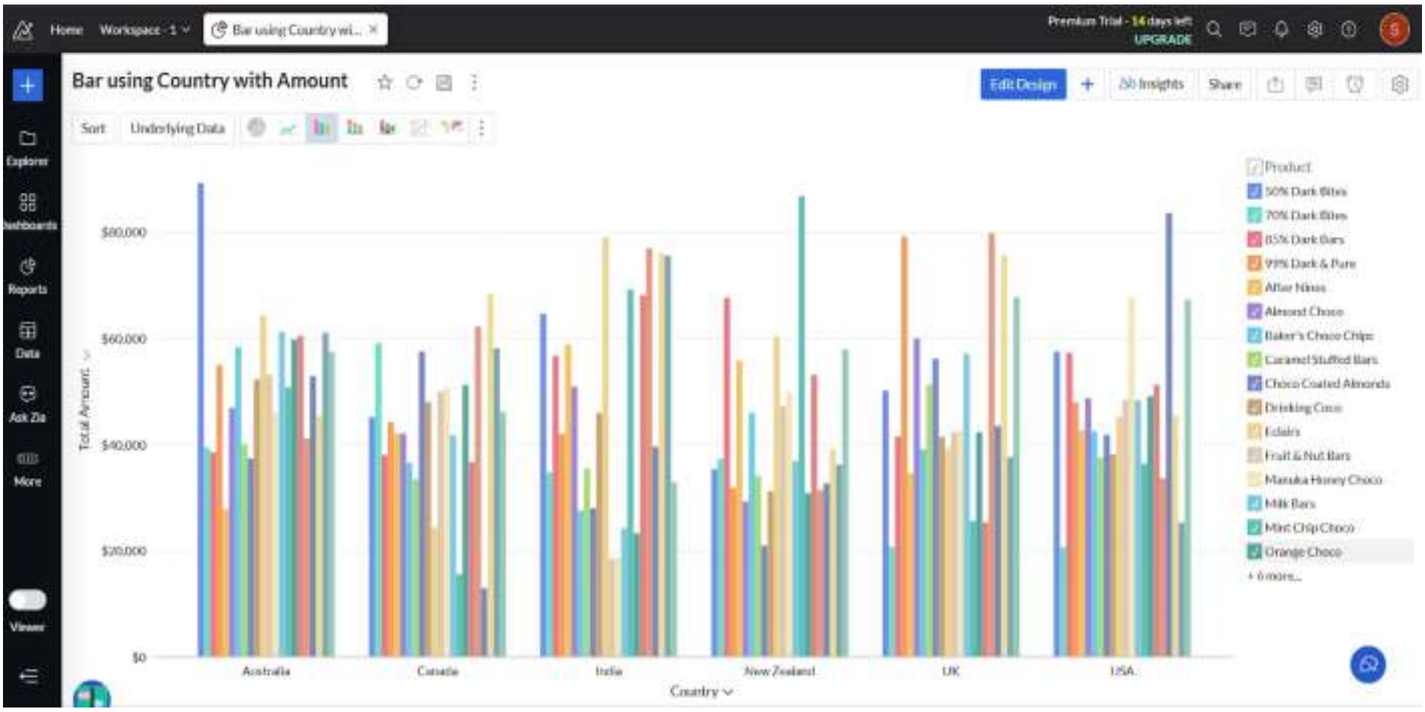
Output:



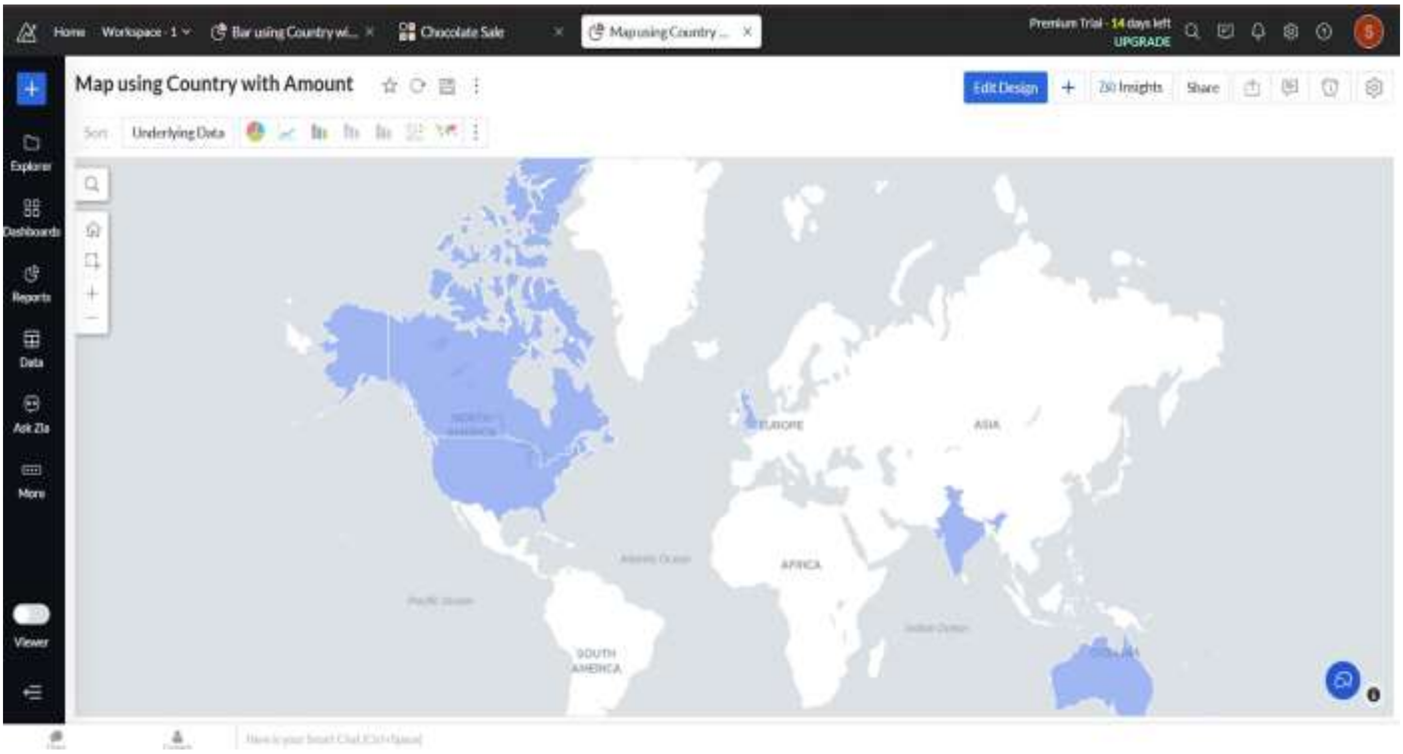
Practical no 10. Show Data Visualization using Zoho Analytics

Aim: To create **visual reports and dashboards** using Zoho Analytics by connecting to datasets and applying visualization tools.

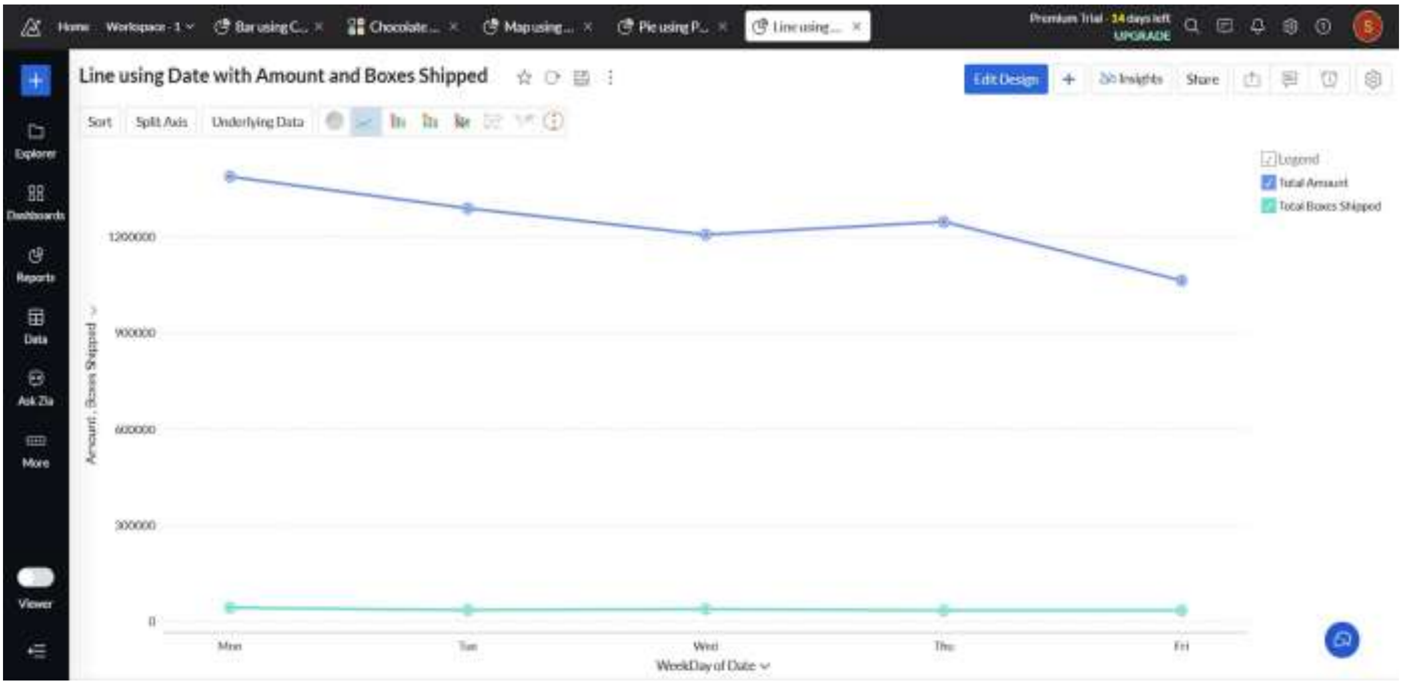
- **BAR CHART**



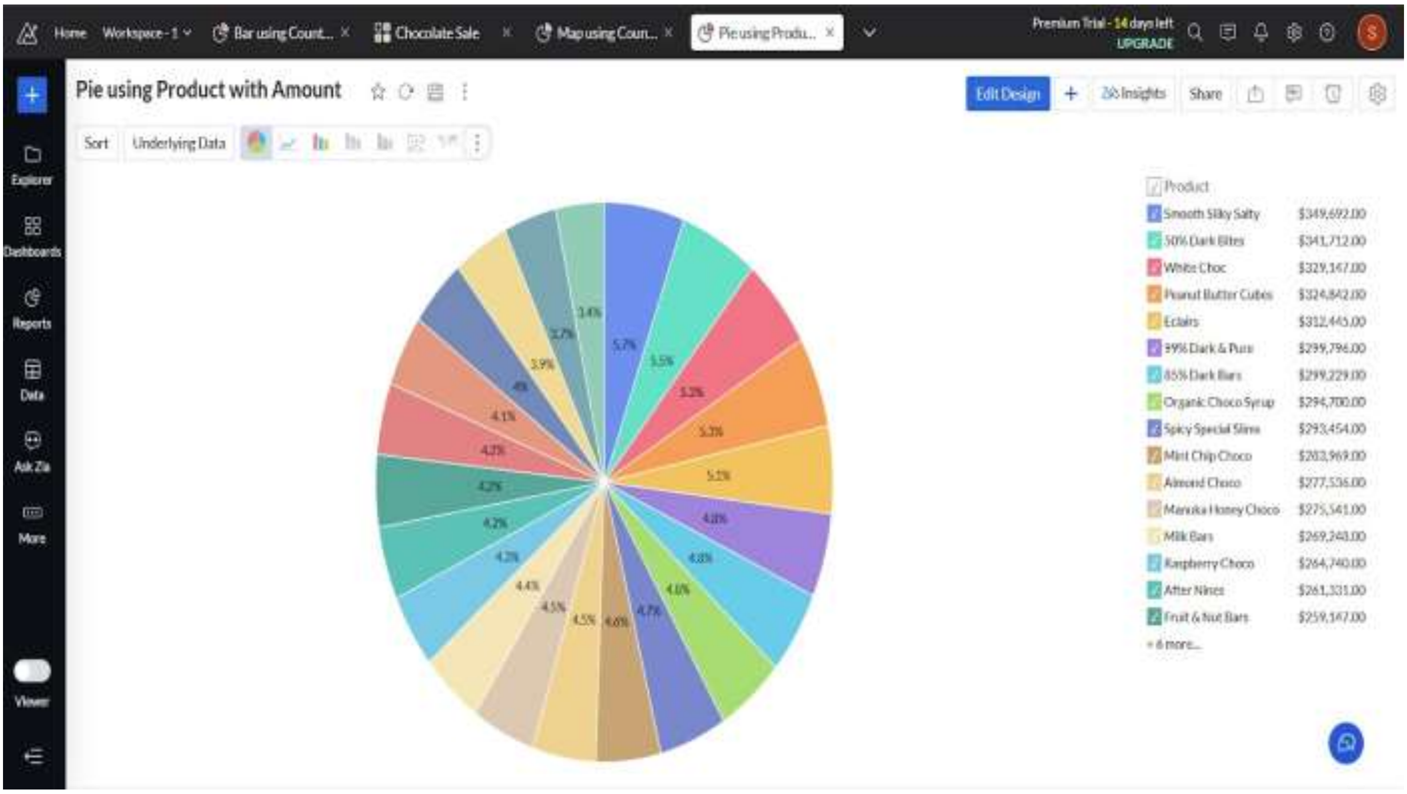
- **WORLD MAP**



- **LINE CHART**

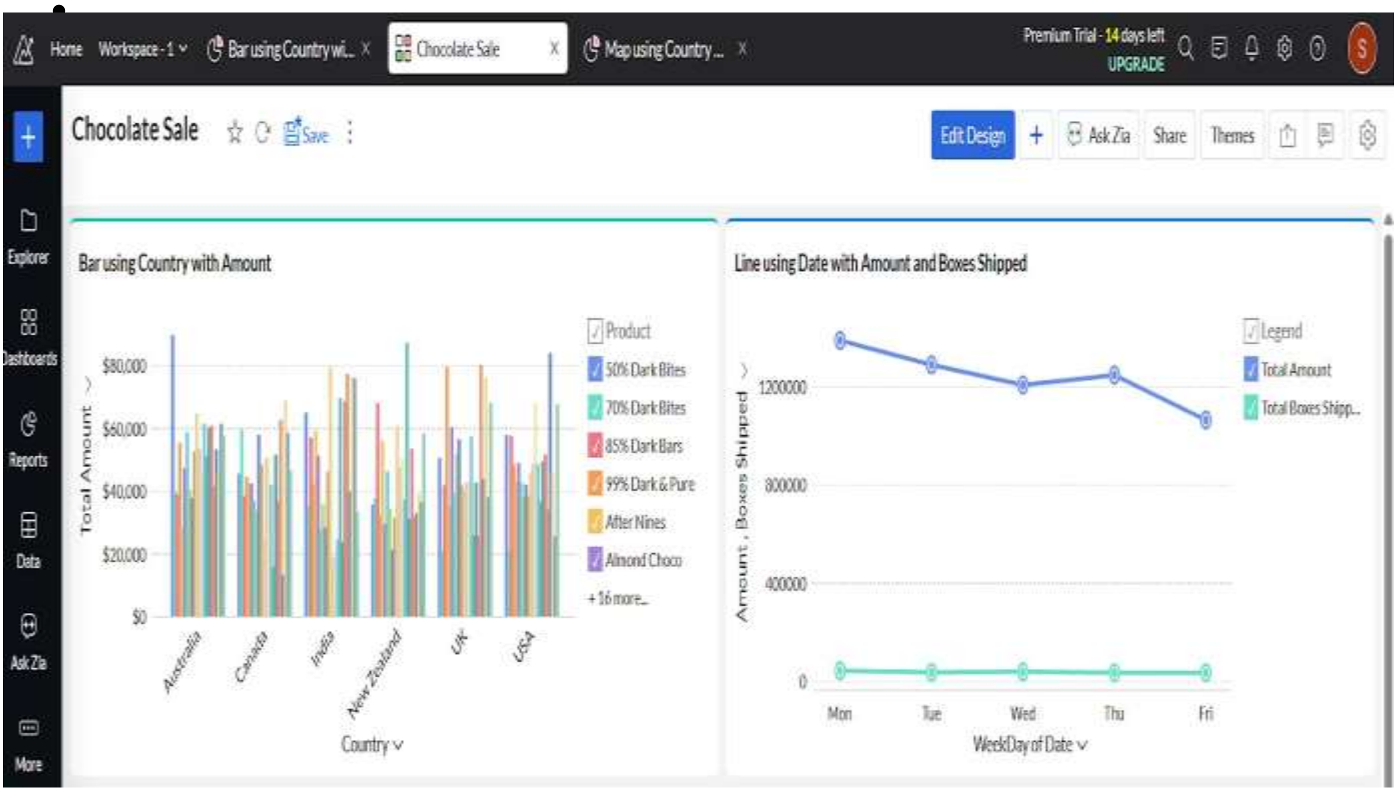


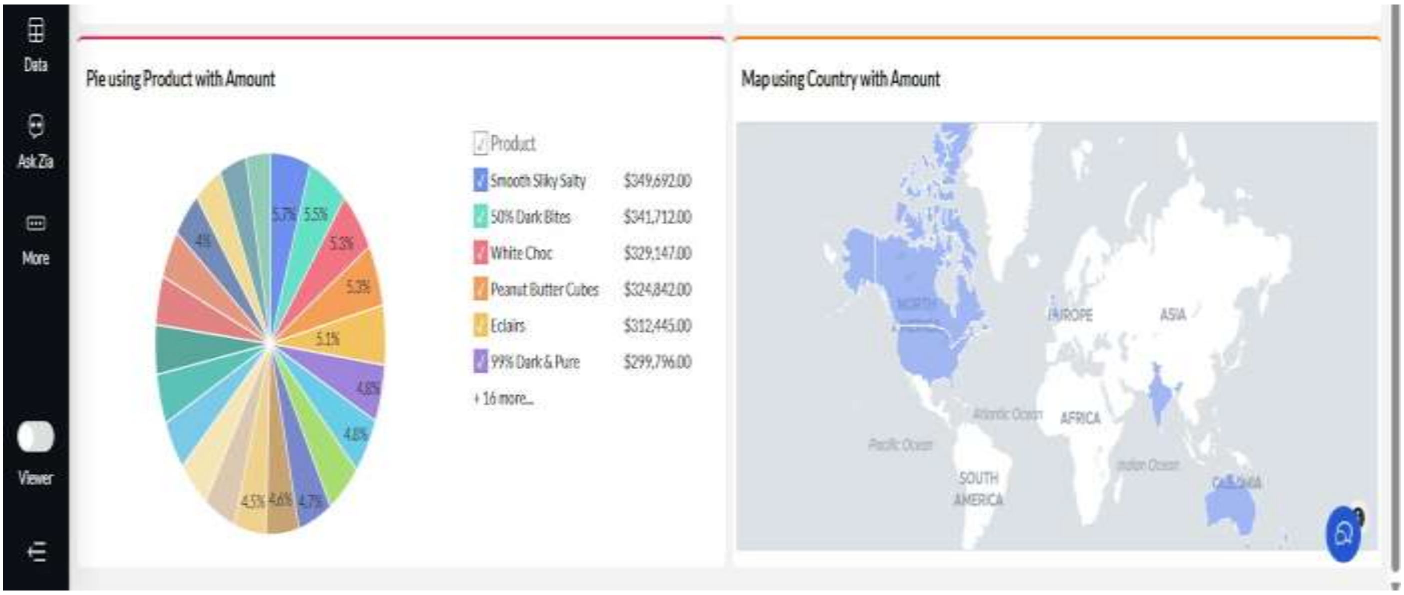
- **PIE CHART**



☑ Product	
☑ Smooth Silky Salty	$349,692.00
☑ 50% Dark Bites	$341,712.00
☑ White Choc	$329,147.00
☑ Peanut Butter Cubes	$324,842.00
☑ Eclairs	$312,445.00
☑ 99% Dark & Pure	$299,796.00
☑ 85% Dark Bars	$299,229.00
☑ Organic Choco Syrup	$294,700.00
☑ Spicy Special Slims	$293,454.00
☑ Mint Chip Choco	$283,969.00
☑ Almond Choco	$277,536.00
☑ Manuka Honey Choco	$275,541.00
☑ Milk Bars	$269,248.00
☑ Raspberry Choco	$264,740.00
☑ After Nines	$261,331.00
☑ Fruit & Nut Bars	$259,147.00
+ 6 more...	

- **INTERACTIVE DASHBOARD**



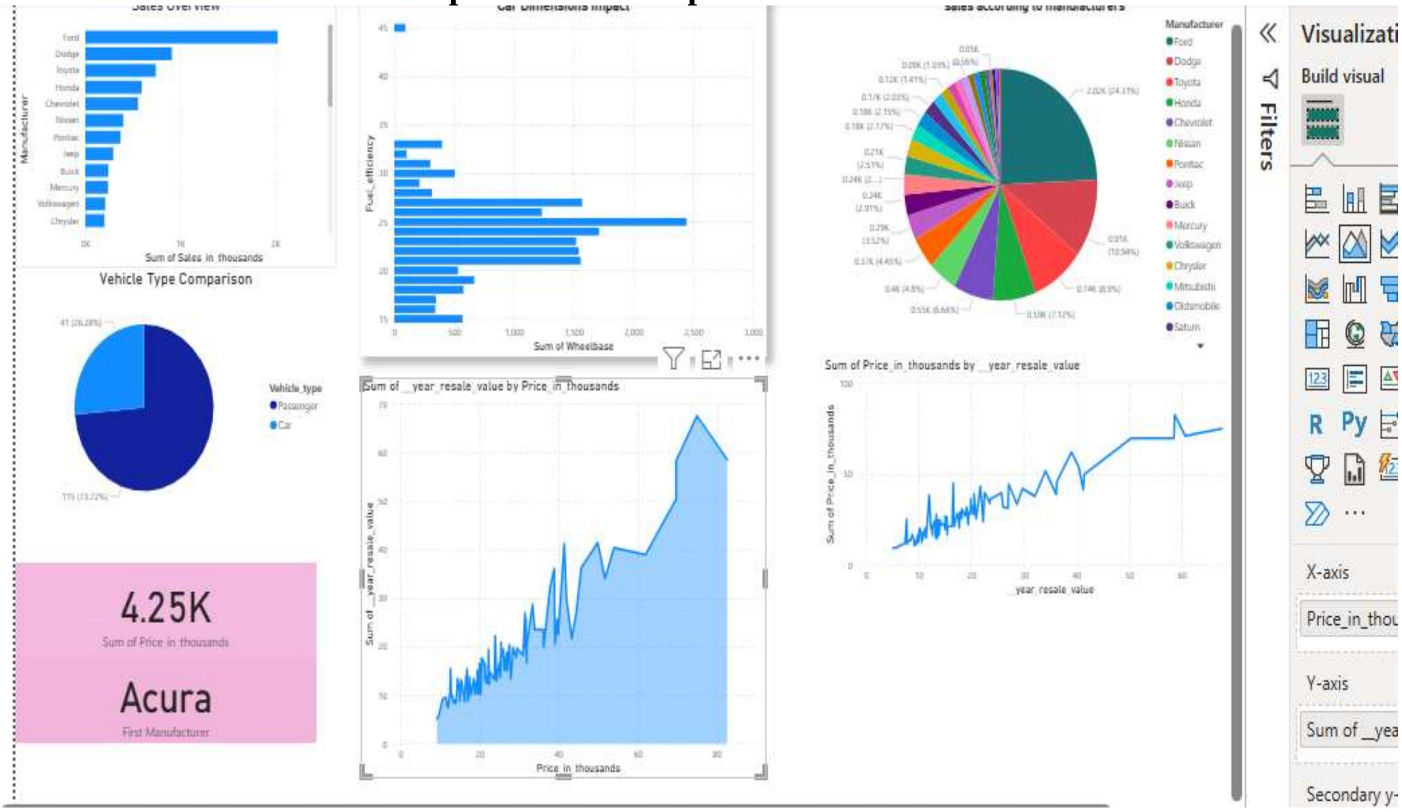


Product	
Smooth Sliky Salty	$349,692.00
50% Dark Bites	$341,712.00
White Choc	$329,147.00
Peanut Butter Cubes	$324,842.00
Eclairs	$312,445.00
99% Dark & Pure	$299,796.00
+ 16 more...	

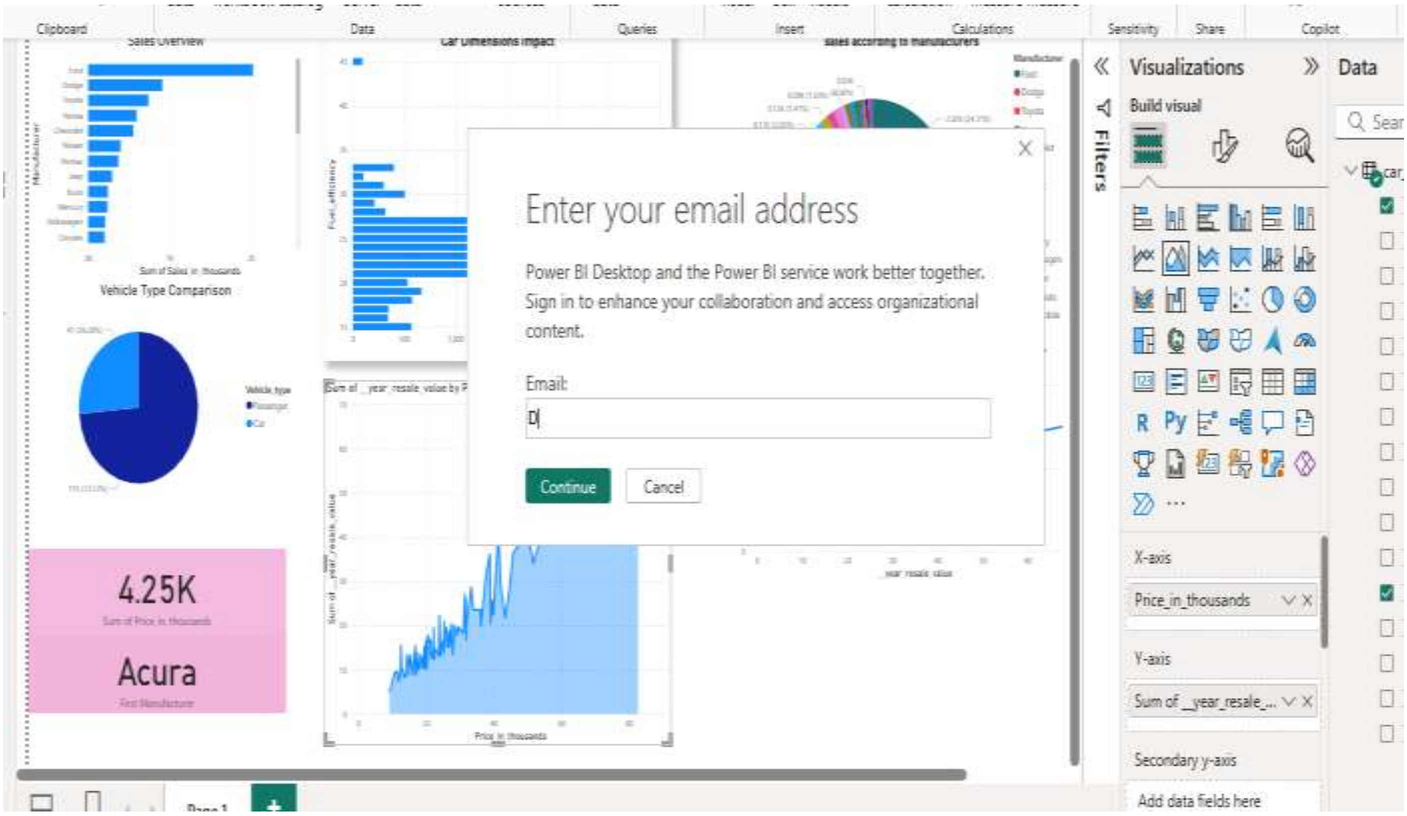
Practical no .11 Publish visualised data on Cloud in PowerBi

Aim: To publish interactive data visualizations created in Power BI Desktop to the cloud (Power BI Service) using an Azure/AWS-hosted Power BI AMI.

- **Create visualization in power Bi desktop**



- **Publish to Power BI Service (Cloud)**

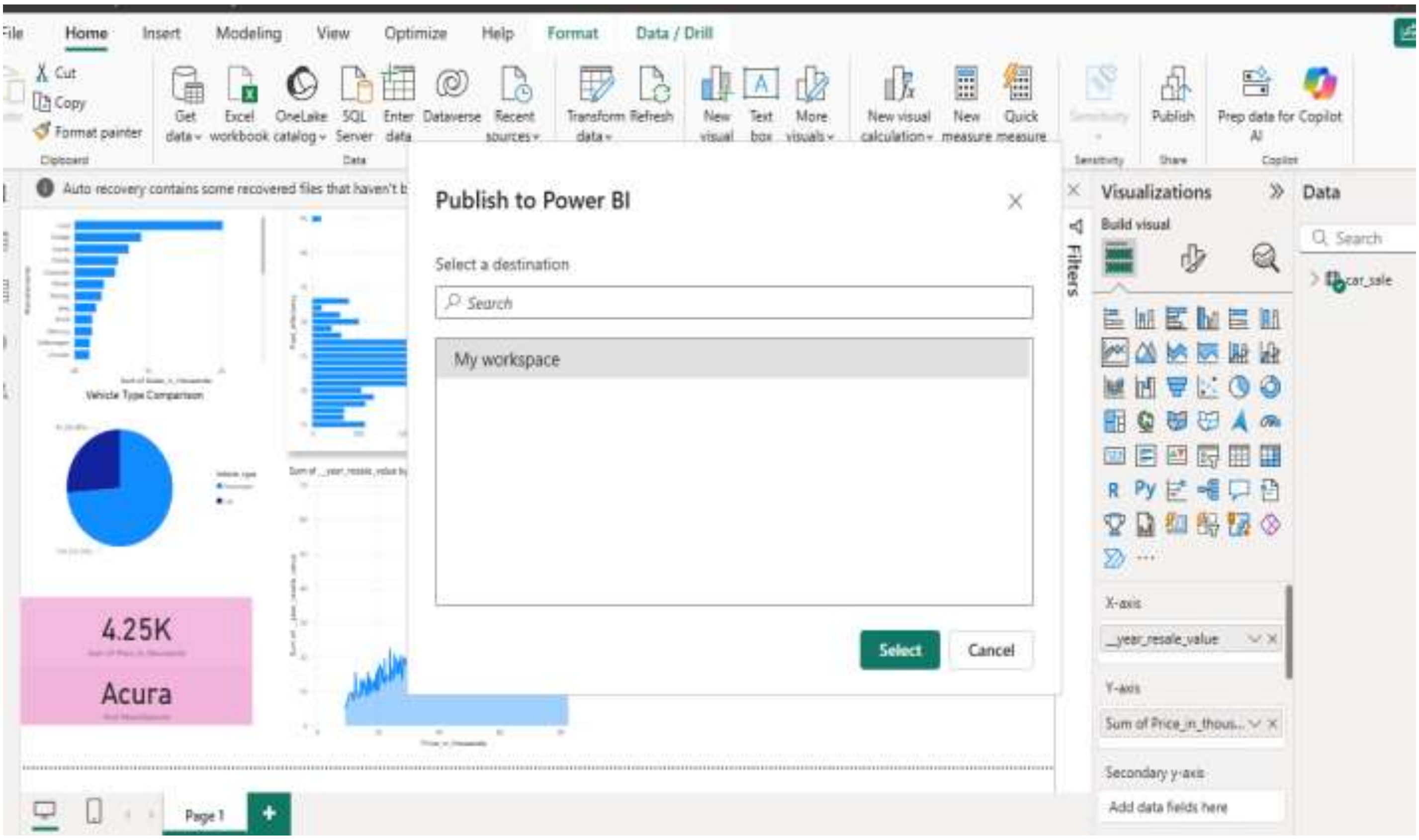


Enter your email address

Power BI Desktop and the Power BI service work better together. Sign in to enhance your collaboration and access organizational content.

Email: D|

Continue Cancel



Auto recovery contains some recovered files that haven't b

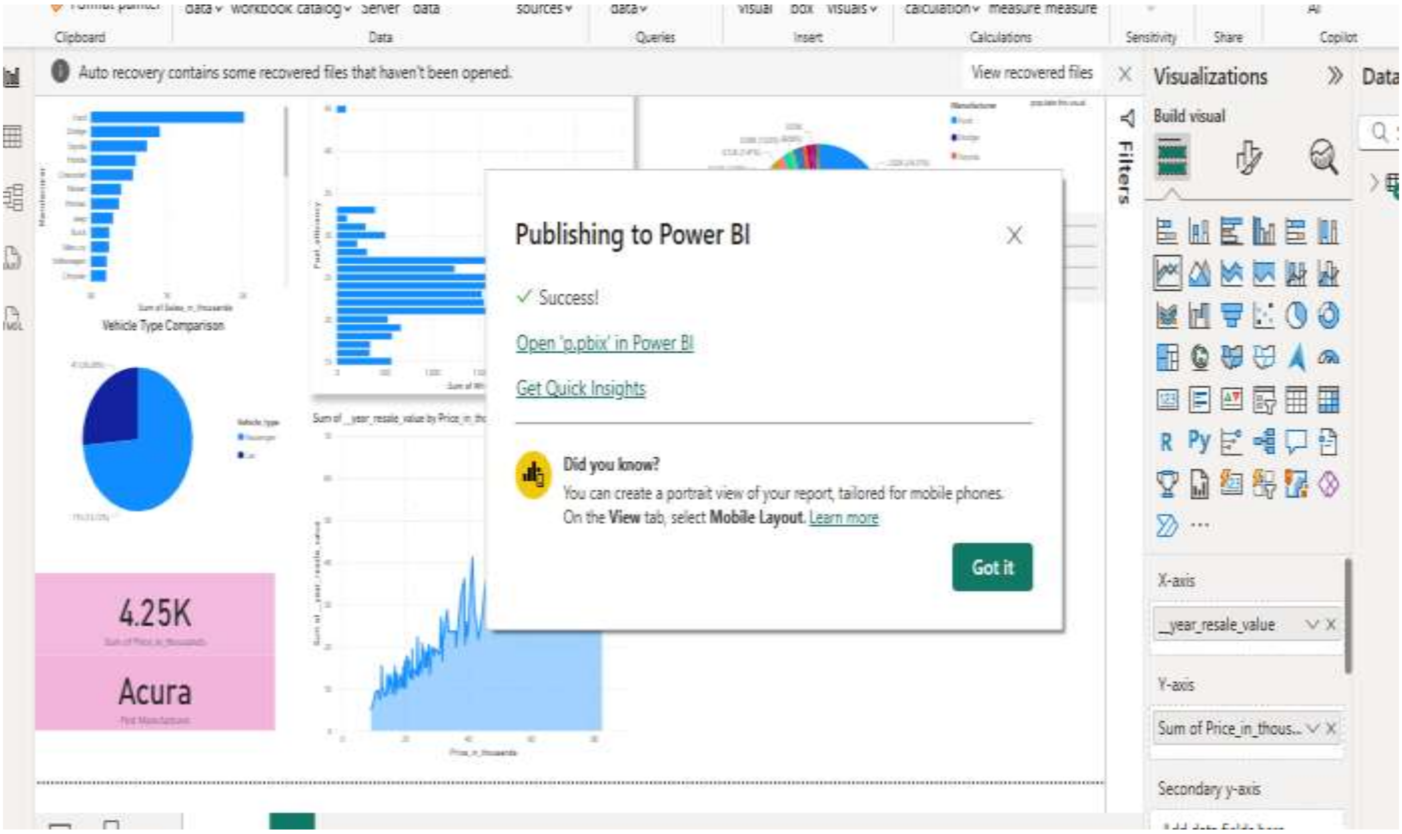
Publish to Power BI

Select a destination

Search

My workspace

- Share & Manage Cloud Reports



Auto recovery contains some recovered files that haven't been opened.

Publishing to Power BI

✓ Success!

Open 'p.pbix' in Power BI

Get Quick Insights

Did you know?

You can create a portrait view of your report, tailored for mobile phones. On the **View** tab, select **Mobile Layout**. Learn more