

| | |
|----------------------------------|---|
| Exp No: 5 Date: | EDA – Data Visualization with Matplotlib |
|----------------------------------|---|

Aim

The Python code aims to perform exploratory data analysis (EDA) by applying preprocessing steps and creating visualizations with Matplotlib. This helps to identify trends, compare group statistics, and observe data distributions using line charts, bar charts, and histograms.

Problem Statement

Raw datasets often contain large amounts of information that are not immediately meaningful. Without proper preprocessing and exploratory data analysis (EDA). Visualization techniques such as line charts, bar charts, and histograms help in summarizing the data and gaining insights.

Algorithm

Step 1: Import pandas for data handling, matplotlib for visualization, and sklearn scalers for preprocessing.

Step 2: Read the StudentsPerformance.csv dataset into a Pandas DataFrame.

Step 3: Display the first few rows of the dataset using df.head() to understand its structure.

Step 4: Group data by reading score and plot average math scores.

Step 5: Plot separate lines for categories (e.g., gender) for comparison.

Step 6: Group data by gender and calculate the average writing score.

Step 7: Plot a bar chart to compare gender-based averages.

Step 8: Plot the distribution of math scores to observe frequency patterns.

Step 9: Apply StandardScaler or MinMaxScaler for feature normalization if needed for further analysis.

Step 10: Analyze visualizations to identify trends, relationships, and score distributions.

Sample Code

```
import pandas as pd

from sklearn.preprocessing import StandardScaler, MinMaxScaler
```

```
import matplotlib.pyplot as plt
```

```
# Step 1: Load dataset
```

```
df = pd.read_csv('StudentsPerformance.csv')
```

```
df.head()
```

```
:
```

| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|--------|----------------|-----------------------------|--------------|-------------------------|------------|---------------|---------------|
| 0 | female | group B | bachelor's degree | standard | none | 72 | 72 | 74 |
| 1 | female | group C | some college | standard | completed | 69 | 90 | 88 |
| 2 | female | group B | master's degree | standard | none | 90 | 95 | 93 |
| 3 | male | group A | associate's degree | free/reduced | none | 47 | 57 | 44 |
| 4 | male | group C | some college | standard | none | 76 | 78 | 75 |

```
# Step 2: Line Chart - Average math score across reading score levels by gender
```

```
for gender in df["gender"].unique():
```

```
    avg_scores = df[df["gender"] == gender].groupby("reading score")["math score"].mean()
```

```
    plt.plot(avg_scores.index, avg_scores.values, marker='o', label=gender)
```

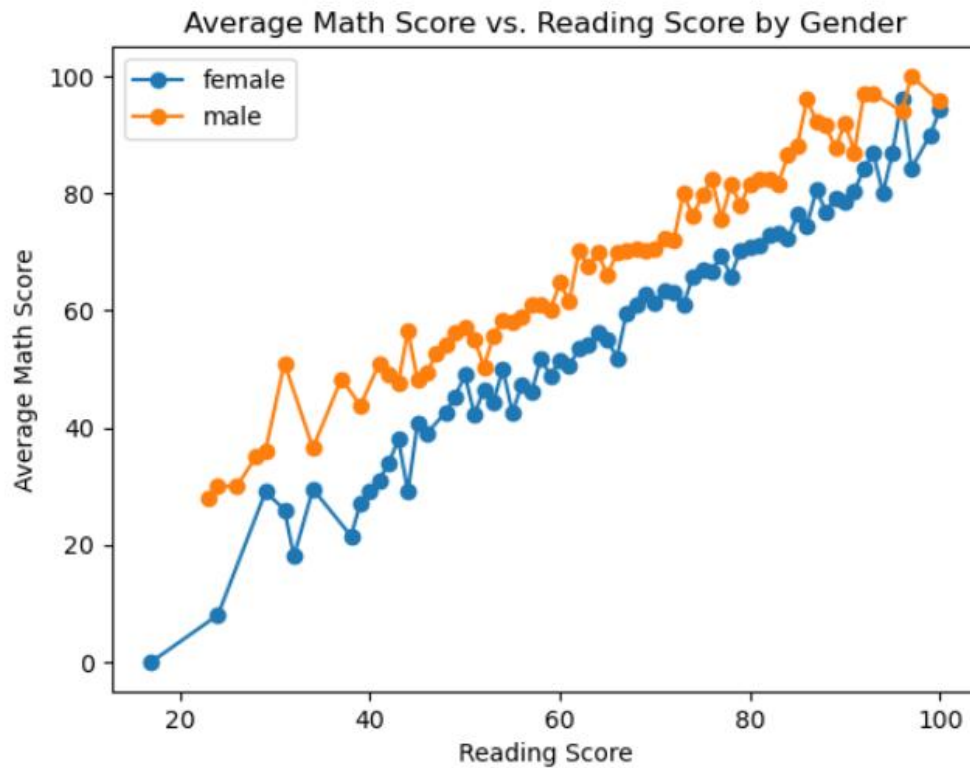
```
plt.title("Average Math Score vs. Reading Score by Gender")
```

```
plt.xlabel("Reading Score")
```

```
plt.ylabel("Average Math Score")
```

```
plt.legend()
```

```
plt.show()
```



Step 3: Bar Chart - Average writing score by gender

```
avg_writing = df.groupby("gender")["writing score"].mean()
```

```
plt.bar(avg_writing.index, avg_writing.values,
        color=['skyblue', 'orange'], edgecolor='black')
```

Add values on top of bars

```
for i, val in enumerate(avg_writing.values):
```

```
    plt.text(i, val + 0.5, round(val, 1), ha='center', fontsize=10)
```

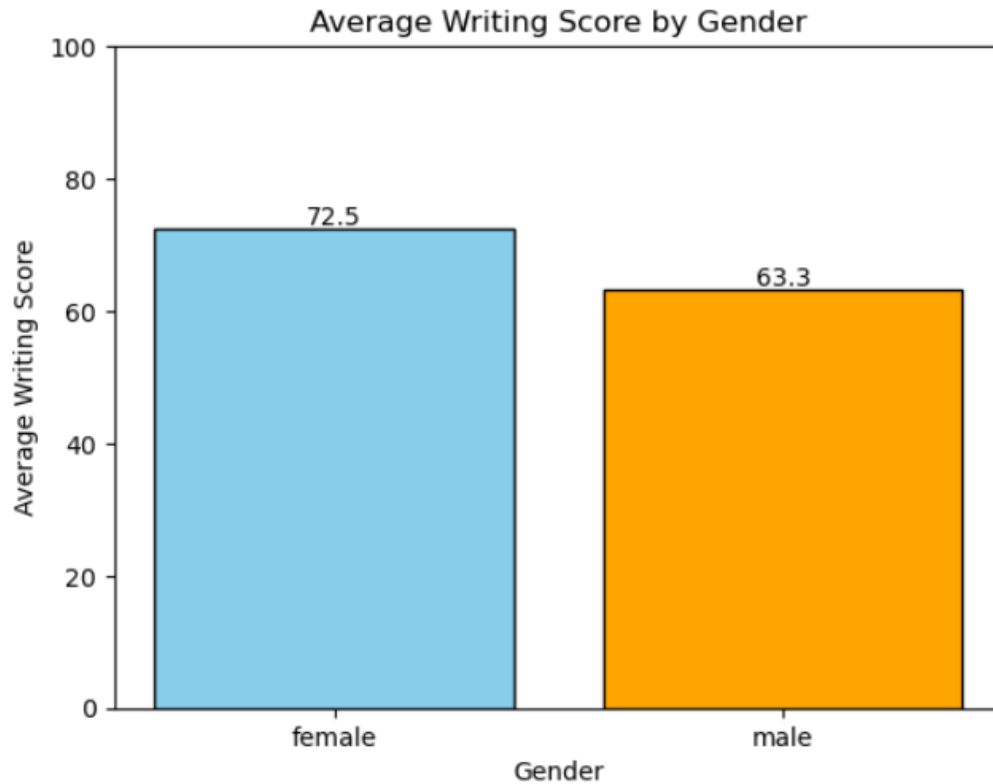
```
plt.title("Average Writing Score by Gender")
```

```
plt.xlabel("Gender")
```

```
plt.ylabel("Average Writing Score")
```

```
plt.ylim(0, 100) # keep y-axis within score range
```

```
plt.show()
```



Step 4: Histogram - Distribution of math scores

```
plt.hist(df["math score"], bins=20, edgecolor='black', color='skyblue')
```

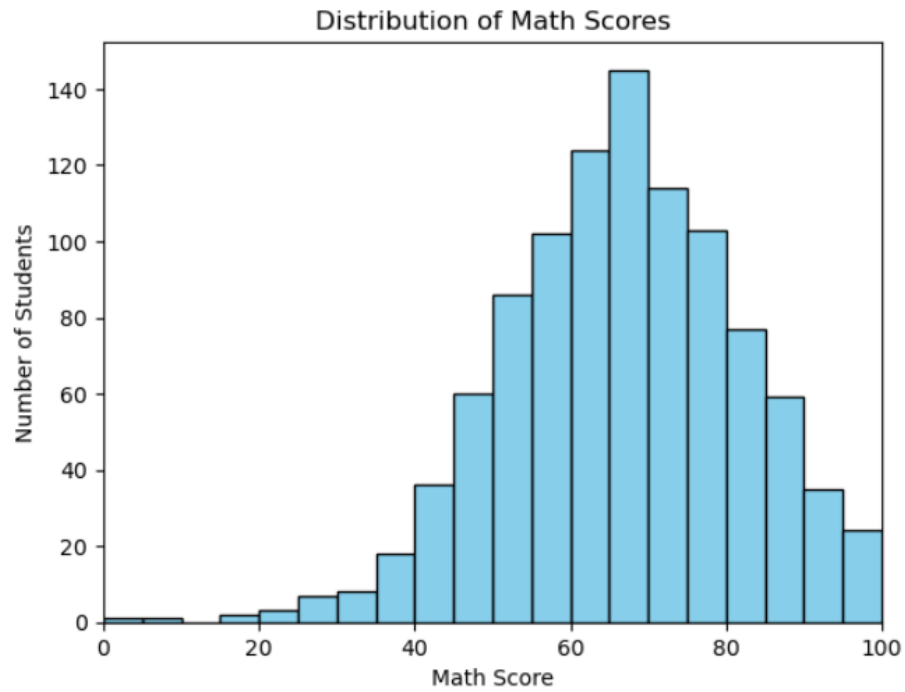
```
plt.title("Distribution of Math Scores")
```

```
plt.xlabel("Math Score")
```

```
plt.ylabel("Number of Students")
```

```
plt.xlim(0, 100) # since scores are between 0–100
```

```
plt.show()
```



Result

Thus the EDA with data visualization with matplotlib was done using line, bar, and histogram charts.

| | |
|----------------------------------|--|
| Exp No: 6 Date: | Data Visualization Using Power BI |
|----------------------------------|--|

Aim:

To learn the Power BI interface and develop skills in connecting to various data sources (Excel, CSV, SQL databases), creating basic visualizations (bar charts, line charts, pie charts), using calculated columns and measures, and building interactive dashboards.

Procedure:

Step 1: Launch Power BI Desktop

- Open Power BI Desktop. Familiarize yourself with the interface like Explore Ribbon (Home, Insert, Modeling, View), Fields Pane (contains tables and columns), Visualizations Pane, Report Canvas

Step 2: Connect to Data Sources

- Home → Get Data.
- Choose the data source type:
- Excel: Browse and select an Excel file, select sheets, and click Load.
- CSV: Browse and select the CSV file, preview, and click Load.
- SQL Database: Enter server name, database, credentials, select tables, and click Load.
- Ensure the data appears in the Fields Pane.

Step 3: Create Basic Visualizations

- Select a visualization type from the Visualizations Pane:
- Bar Chart: Drag a categorical field to the Axis and a numerical field to Values.
- Line Chart: Drag a time/date field to Axis and numerical field to Values.
- Pie Chart: Drag a categorical field to Legend and a numerical field to Values.

- Format charts using the Format options (colors, labels, titles).

Step 4: Create Calculated Columns and Measures

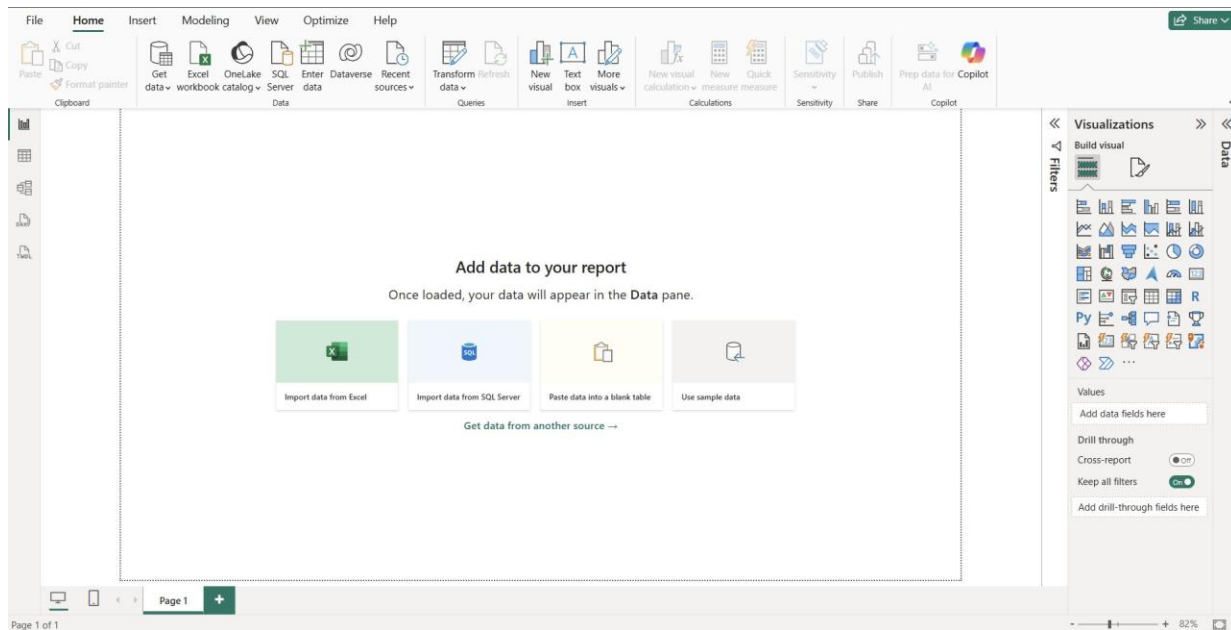
- Calculated Column:
 - Go to Modeling → New Column.
 - Enter DAX formula, e.g., $\text{TotalPrice} = \text{Quantity} * \text{UnitPrice}$.
- Measure:
 - Go to Modeling → New Measure.
 - Enter DAX formula, e.g., $\text{TotalSales} = \text{SUM}(\text{Sales}[\text{TotalPrice}])$.
 - Use these new fields in your visualizations.

Step 5: Build Dashboards

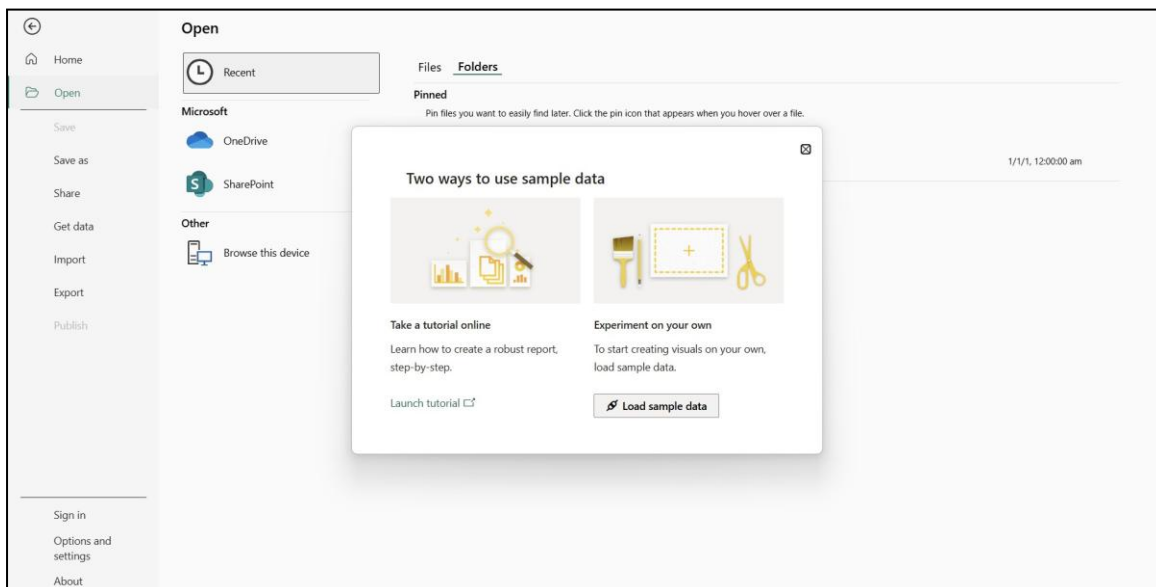
- Arrange multiple visualizations on a single Report Canvas.
- Add slicers to filter data dynamically (e.g., by region or date).
- Customize layout, colors, and titles for readability.
- Save the report: File → Save As.

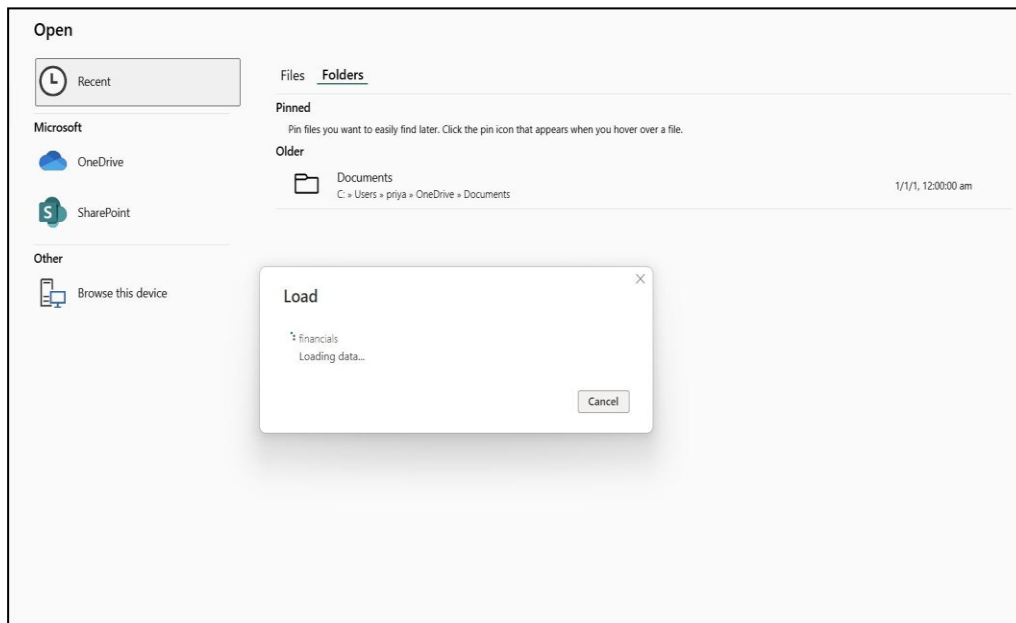
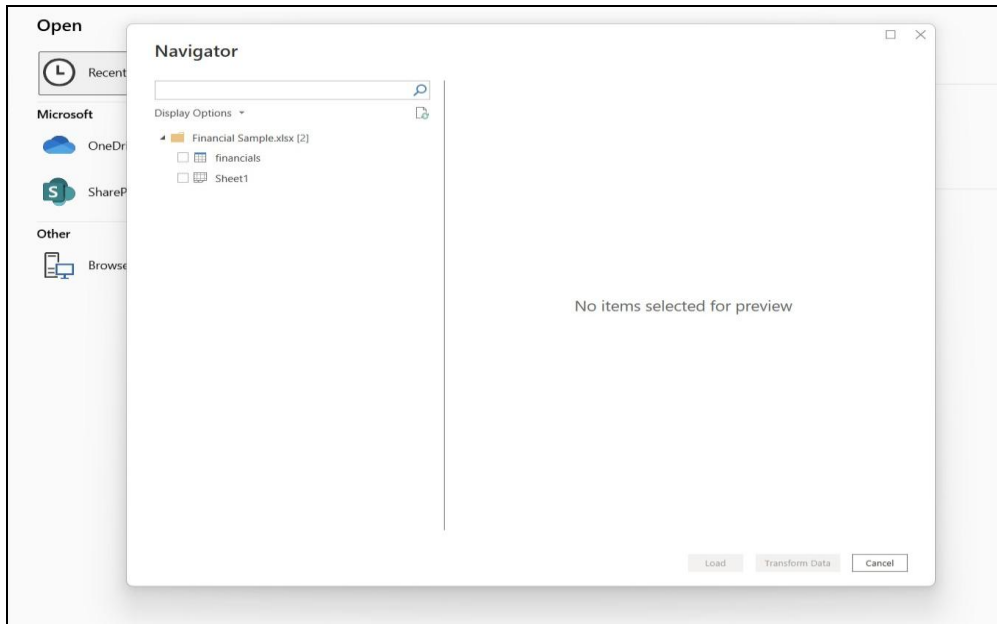
Execution Steps

6.1 Learning the Power BI Interface

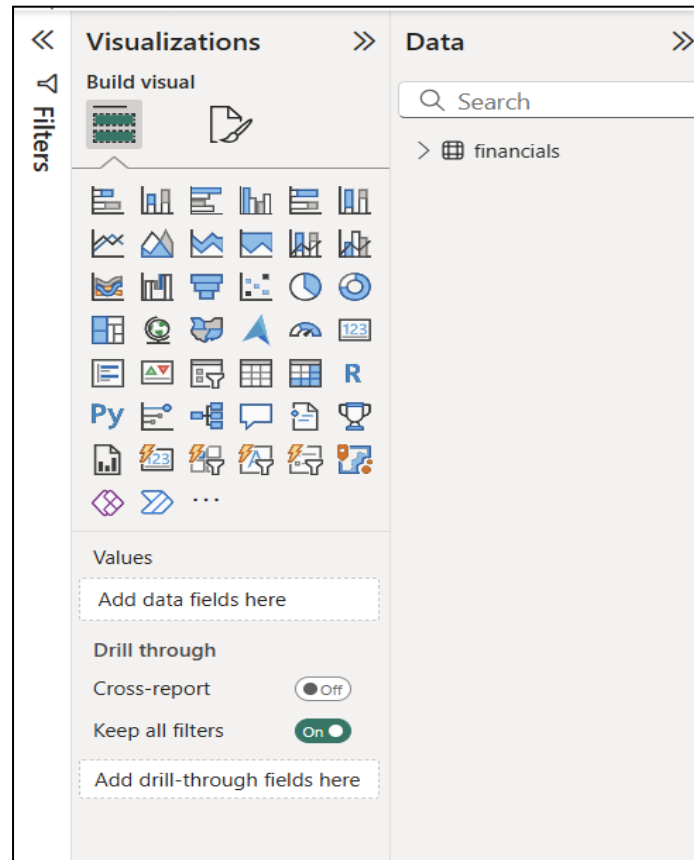


6.2 Connecting to various data sources (Excel, CSV, SQL databases)

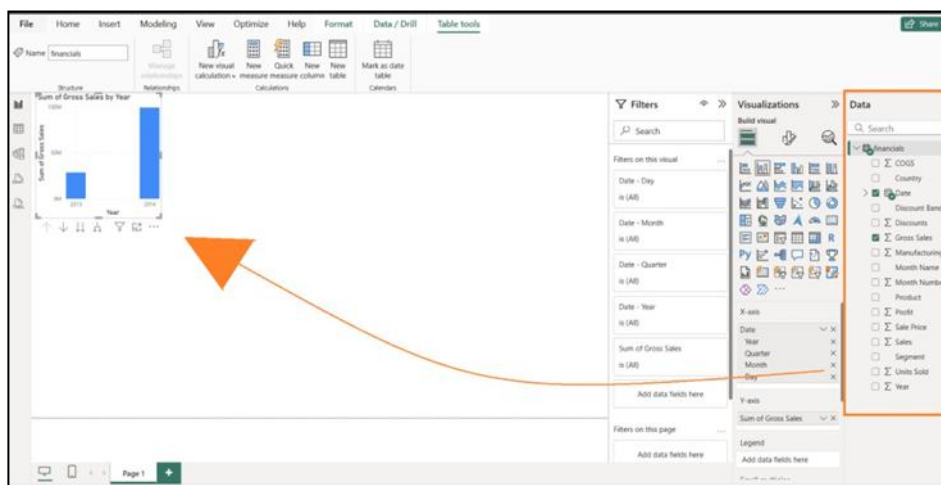




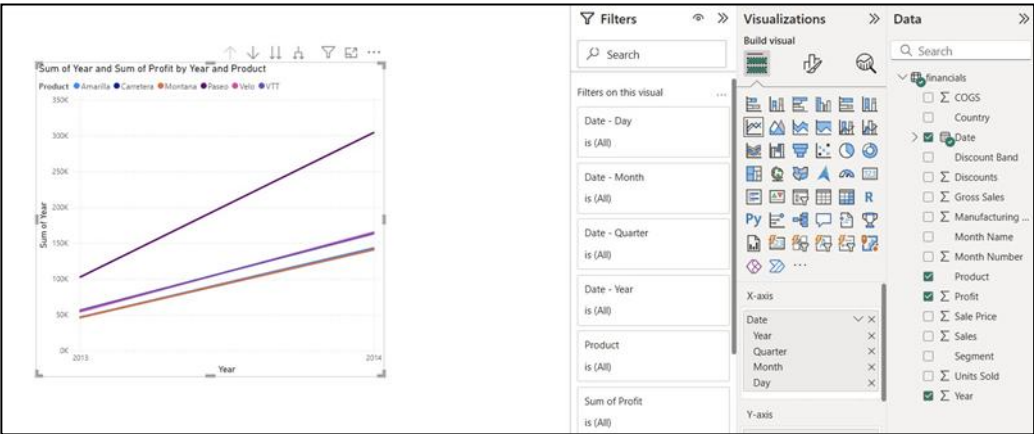
6.3 Creating basic visualizations: bar charts, line charts, pie charts



Bar charts

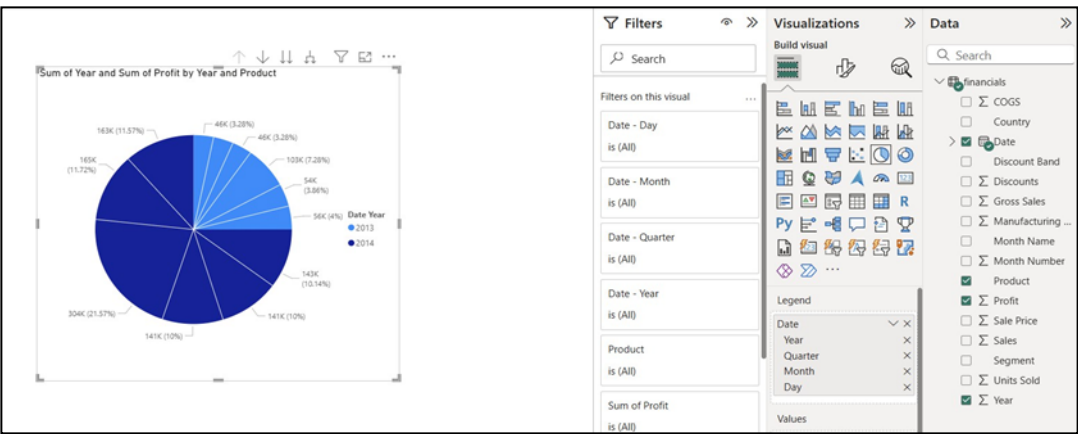


Line chart

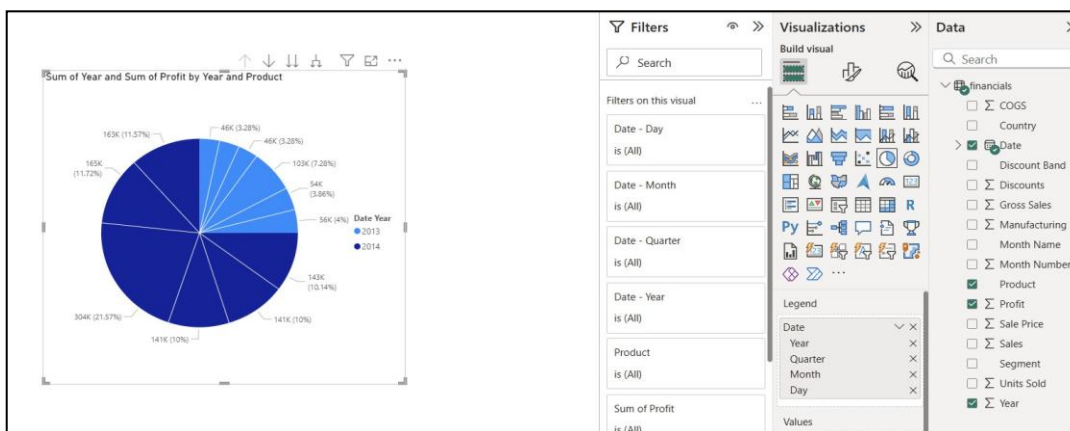
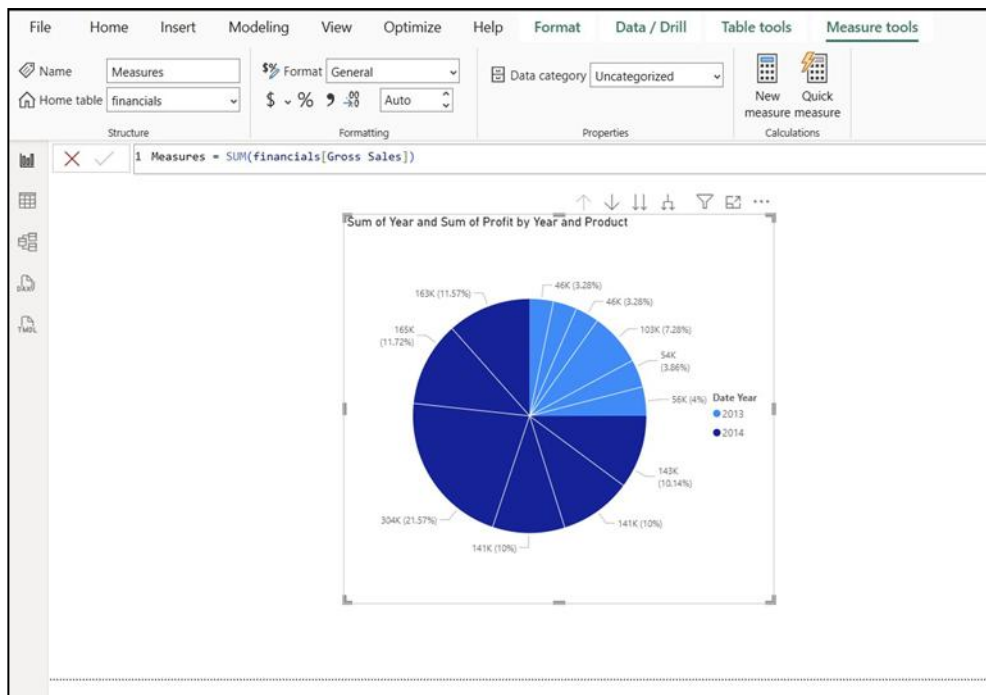


Pie chart

6.4 Creating Calculated Columns and Measures



Go to Modelling, select new measure, upload the formula.



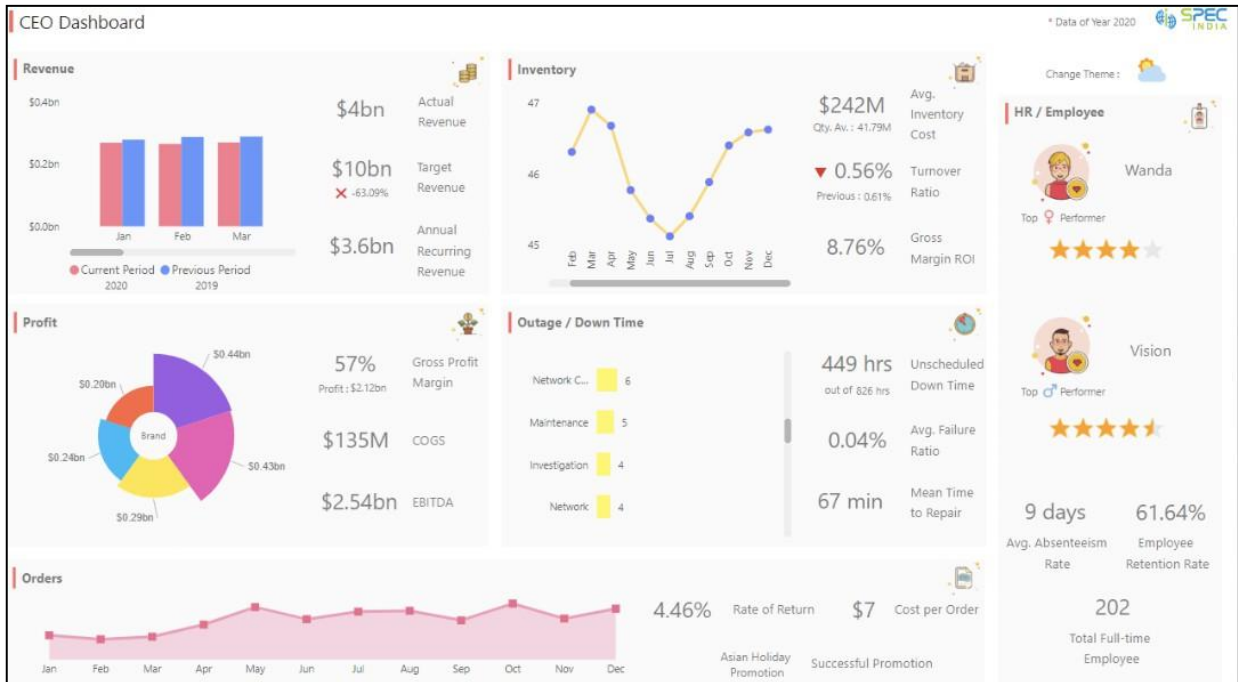
6.5 Building Dashboards

Create a report in Power BI Desktop by following the above mentioned steps and save it.

Publish this to Power BI Service: In Power BI Desktop → Home → Publish. Next, sign in → Select My Workspace (or a shared workspace). Create a dashboard in power BI Service.

Arrange, customize and save the final result.

Sample Power BI Dashboard:



Result:

This experiment provided hands-on experience with Power BI, including connecting to different data sources, creating basic visualizations, performing calculations with DAX, and building interactive dashboards. The process improves data analysis and visualization skills for business intelligence applications.