MIT World Peace University

Data Science for Cybersecurity and Forensics
Third Year B. Tech, Semester 6

Data Visualisation and Analysis on a Dataset

Assignment 4

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 10

April 16, 2024

Contents

1 Aim

To perform data visualisation and analysis on a dataset using Python.

2 Objectives

To Plot the following:

1. Figures in matplotlib
2. Scatter plots in matplotlib and Seaborn
3. Scatter plots in Bokeh
4. Preparation of line plot data
5. Line plots in matplotlib, Seaborn, and Bokeh

3 Theory

3.1 Scatter Plots in Matplotlib and Seaborn

Scatter plots are used to visualize the relationship between two continuous variables. They display individual data points as dots on a two-dimensional plane, with one variable represented on the x-axis and the other on the y-axis. Scatter plots are useful for identifying patterns, correlations, and outliers in the data.

```python
    import matplotlib.pyplot as plt

    x = [1, 2, 3, 4, 5]
    y = [2, 3, 5, 7, 11]

    plt.scatter(x, y)
    plt.xlabel('X-axis')
    plt.ylabel('Y-axis')
    plt.title('Scatter Plot')
    plt.show()
```

Listing 1: "Scatter plot using Matplotlib"

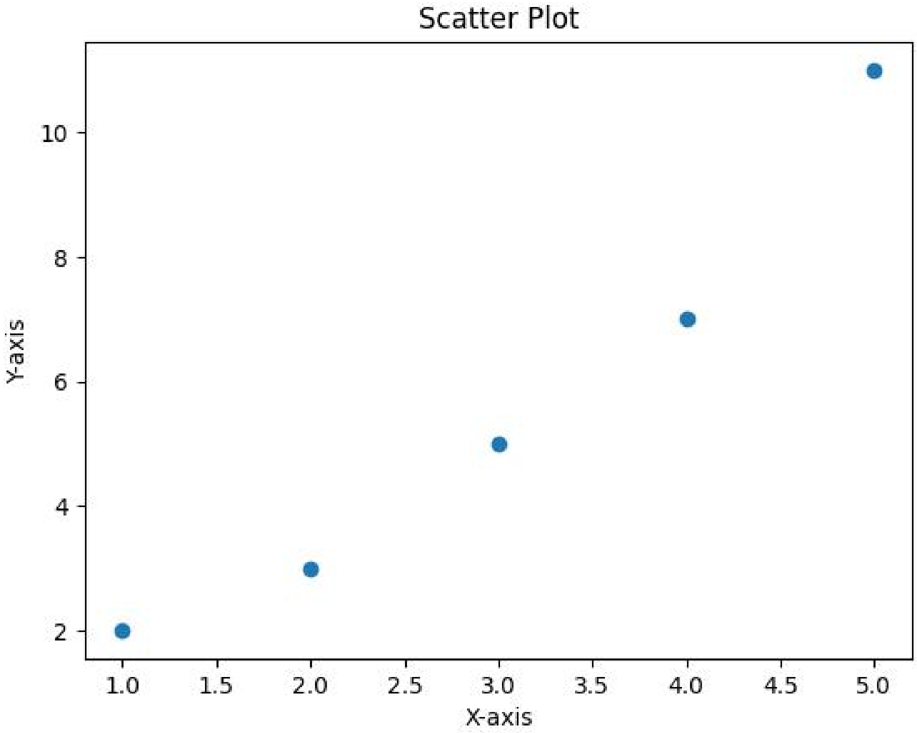

Figure 1: Scatter Plot

```python
import seaborn as sns
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

sns.scatterplot(x=x, y=y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot')
plt.show()
```

Listing 2: Scatter plot using Seaborn

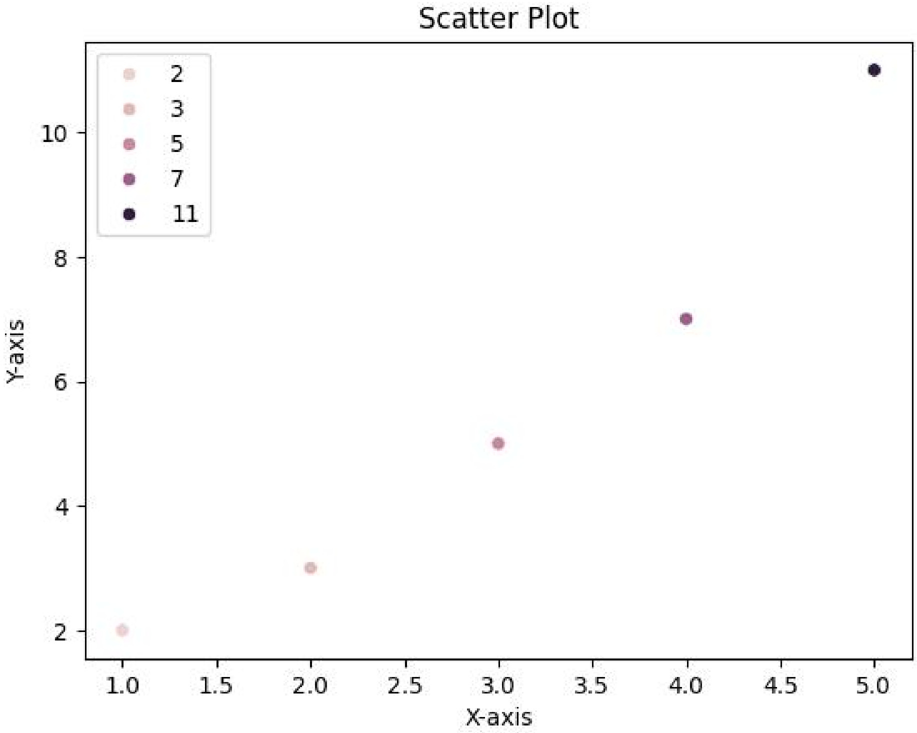

Figure 2: Scatter Plot in Seaborn

Scatter Plots in Bokeh

Bokeh is a Python library for creating interactive visualizations. Scatter plots in Bokeh provide similar functionality to Matplotlib and Seaborn but offer interactivity such as hover tooltips and zooming capabilities.

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

    x = [1, 2, 3, 4, 5]
    y = [2, 3, 5, 7, 11]

    output_file("scatter_plot.html")

    p = figure(title="Scatter Plot", x_axis_label='X-axis', y_axis_label='Y-axis')
    p.circle(x, y, size=10, color="navy", alpha=0.5)

    show(p)
```

Listing 3: Scatter plot using Bokeh

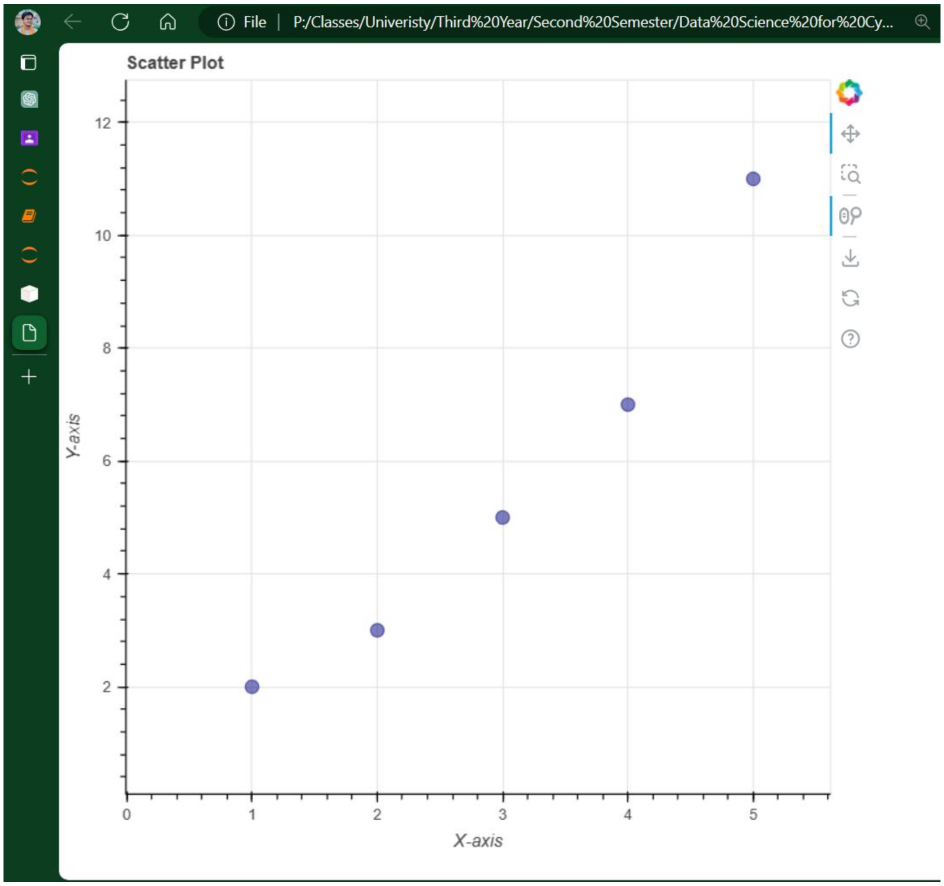

Figure 3: Scatter Plot using Bokeh

Preparation of Line Plot Data

Line plots are used to visualize trends and patterns in data over time or other ordered categories. To prepare data for a line plot, you typically have one variable representing the x-axis (e.g., time) and another representing the y-axis (e.g., numerical values).

Line plots are created similarly in Matplotlib, Seaborn, and Bokeh. They visualize the relationship between two continuous variables by connecting data points with straight lines.

```python
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

plt.plot(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Line Plot')
plt.show()
```

Listing 4: Line plot using Matplotlib

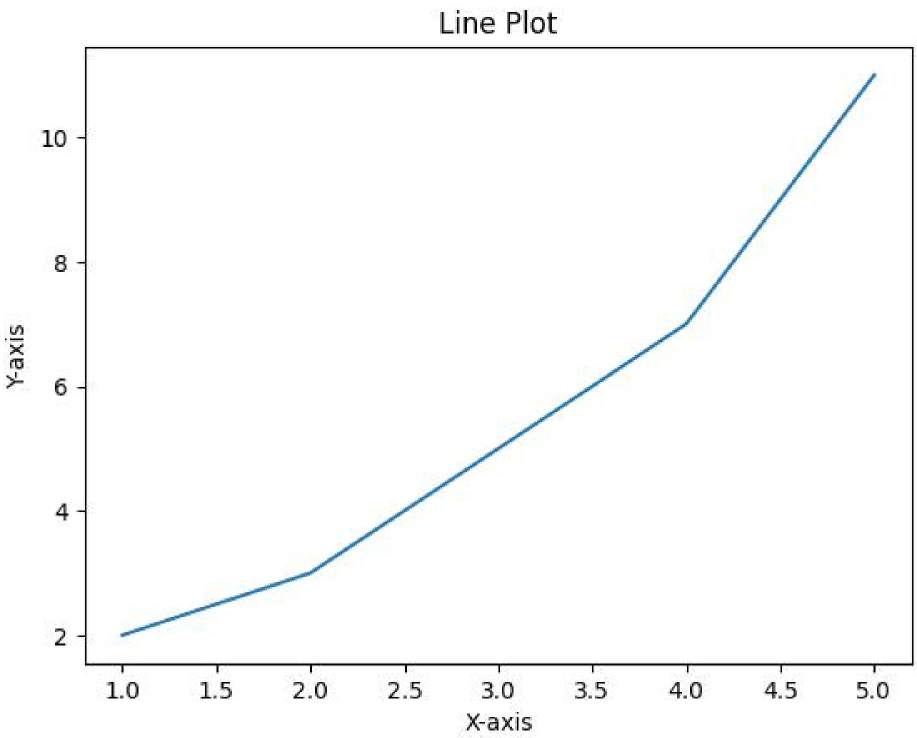

Figure 4: Matplotlib Line Plot

```python
import seaborn as sns
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

sns.lineplot(x=x, y=y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Line Plot')
plt.show()
```

Listing 5: Line plot using Seaborn

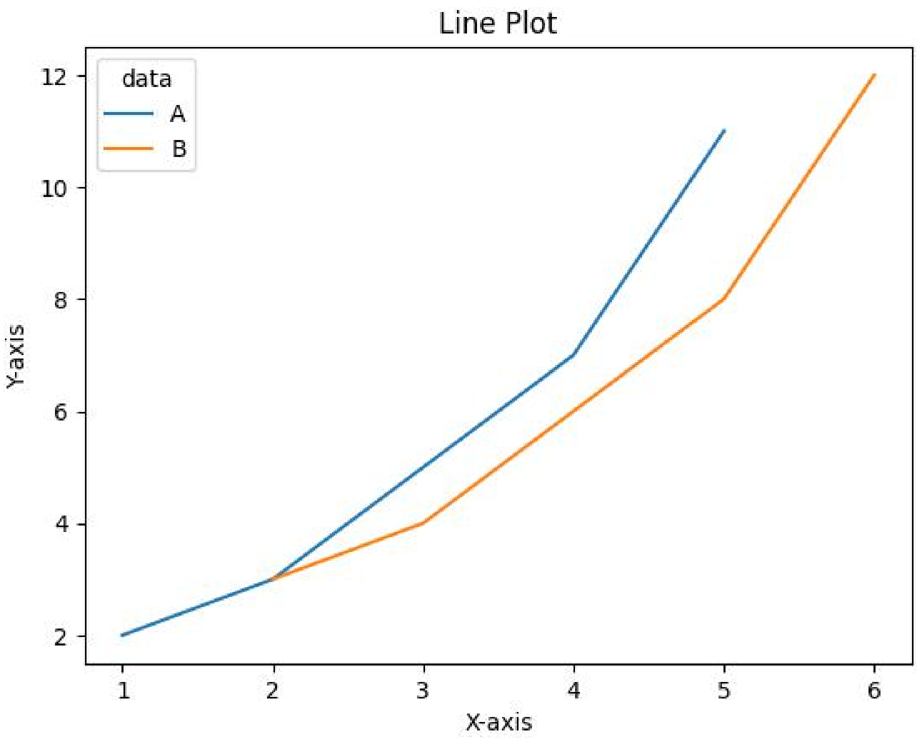

Figure 5: Seaborn Plot by using different hue

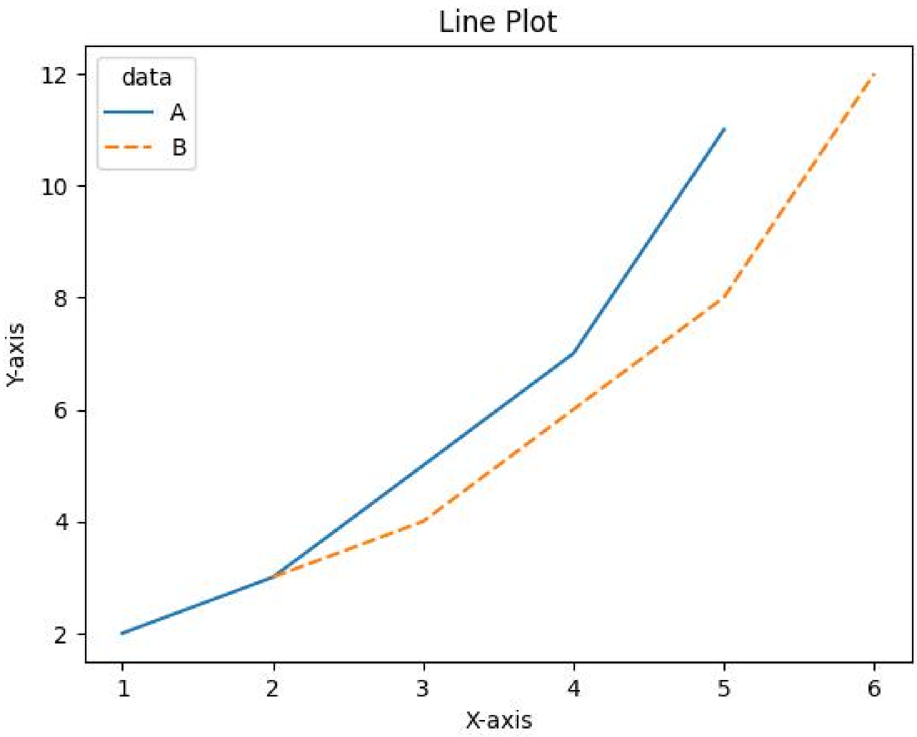

Figure 6: Seaborn line plot using different hue and style.

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

x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

output_file("line_plot.html")

p = figure(title="Line Plot", x_axis_label='X-axis', y_axis_label='Y-axis')
p.line(x, y, line_width=2)

show(p)
```

Listing 6: Line plot using Bokeh

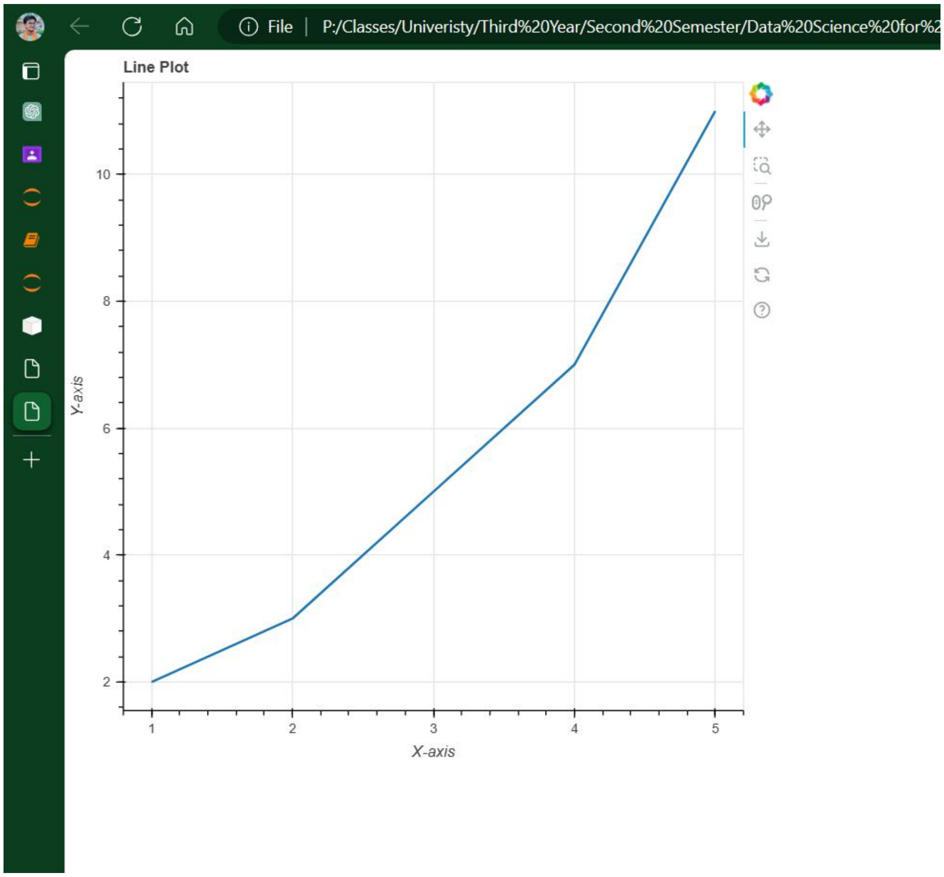
File
P:/Classes/Univeristy/Third%20Year/Second%20Semester/Data%20Science%20for%2
Line Plot
10
8
6
4
2
Y-axis
1
2
3
4
5
X-axis

Figure 7: Bokeh Line Plot

4 Platform

Operating System: Windows 11
IDEs or Text Editors Used: Visual Studio Code
Compilers or Interpreters: Python 3.10.1

5 Requirements

```
python==3.10.1
matplotlib==3.8.3
numpy==1.26.4
pandas==2.2.2
seaborn==0.13.2
bokeh==3.4.1
```

6 FAQs

6.1 Question 1

1. **What is data visualization in Data science?**
   Data visualization in data science refers to the graphical representation of data to visually communicate insights, patterns, and trends. It involves creating visualizations such as charts, graphs, maps, and dashboards to help analysts and stakeholders understand complex datasets more easily and make informed decisions.

6.2 Question 2

1. **Why do we need data visualizations? Explain properly.**
   Data visualizations are essential in data science for several reasons:

   - **Enhanced Understanding**: Visual representations make it easier to interpret and understand complex datasets, enabling analysts to uncover insights and patterns that may not be apparent from raw data alone.
   - **Communication**: Visualizations facilitate communication of findings and insights to stakeholders, making it easier for non-technical audiences to grasp key concepts and make data-driven decisions.
   - **Identification of Trends**: Visualizations allow analysts to identify trends, correlations, and outliers in the data quickly, enabling them to detect patterns and relationships that may inform future strategies or actions.
   - **Decision Making**: Visualizations empower decision-makers to assess data-driven insights visually, enabling them to make informed decisions based on a clear understanding of the data and its implications.
   - **Exploratory Analysis**: Visualizations support exploratory data analysis by enabling analysts to interact with the data dynamically, exploring different perspectives and hypotheses to gain deeper insights.

6.3 Question 3

1. **Describe the different data visualization approaches in Data science using Python?**
   In data science, various data visualization approaches can be implemented using Python libraries such as Matplotlib, Seaborn, Plotly, and Pandas. These approaches include:

   - **Static Visualizations**: Creating static visualizations such as line plots, bar charts, histograms, scatter plots, and box plots using Matplotlib and Seaborn libraries. These visualizations are suitable for presenting static snapshots of data insights.
   - **Interactive Visualizations**: Generating interactive visualizations such as interactive plots, heatmaps, and geographic maps using Plotly library. Interactive visualizations allow users to explore data dynamically, zoom in/out, and interact with data points for deeper insights.
   - **Statistical Visualizations**: Generating statistical visualizations such as probability density plots, violin plots, and pair plots using Seaborn library. Statistical visualizations provide insights into data distributions, correlations, and statistical relationships.

- **Dashboarding**: Creating interactive dashboards and reports using libraries such as Plotly Dash or Bokeh. Dashboards integrate multiple visualizations and controls into a single interface, enabling users to interactively explore and analyze data.

These approaches enable data scientists and analysts to create a wide range of visualizations tailored to specific data exploration and communication needs, enhancing the understanding and impact of data-driven insights.

7 Conclusion

In this assignment, we explored data visualization and analysis techniques using Python libraries such as Matplotlib, Seaborn, and Bokeh. We learned how to create scatter plots and line plots to visualize relationships between variables and trends in data. By leveraging these visualization tools, data scientists can gain deeper insights into complex datasets, communicate findings effectively, and make informed decisions based on data-driven insights.