# Introduction to Matplotlib

| | |
|---|---|
| **Course Code**: CPE 031 | **Program**: Computer Engineering |
| **Course Title**: Visualization and Data Analysis | **Date Performed**: October 22, 2024 |
| **Section**: CPE21S4 | **Date Submitted**: October 23, 2024 |
| **Name**: Bona, Andrei Nycole So | **Instructor**: Engr. Ma. Rizette Sayo |

**Intended Learning Outcomes (ILO)**:

By the end of this laboratory session, learners will be able to:

1. Utilize Matplotlib's pyplot interface to create a variety of visualizations, including line plots, scatter plots, histograms, and box plots, demonstrating an understanding of the library's syntax and functionality.

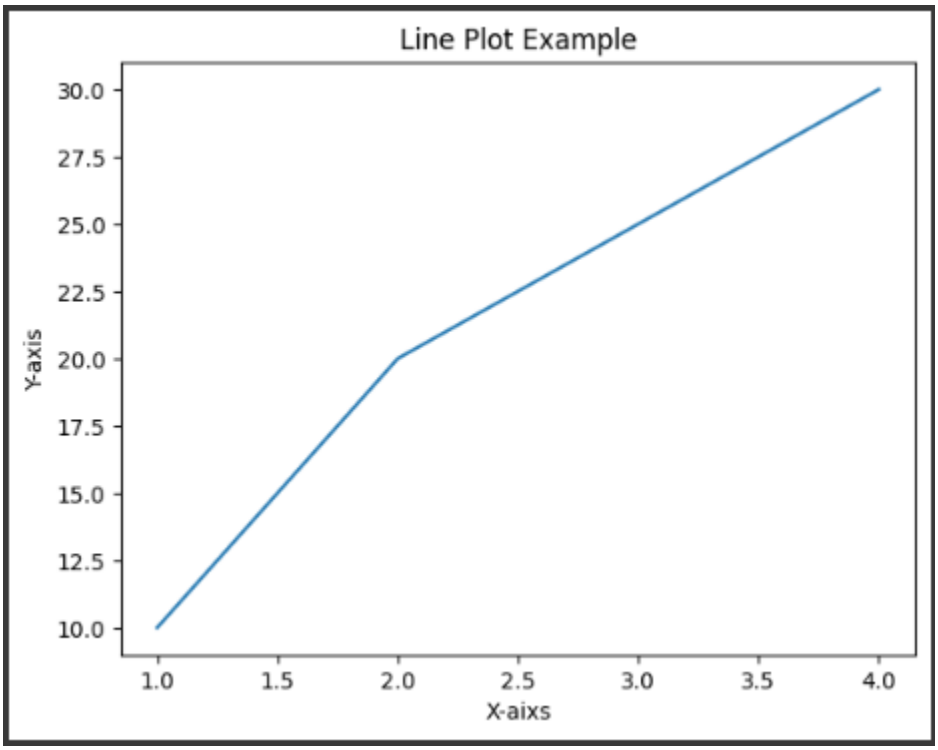
2. Customize visual elements such as titles, labels, and legends to enhance the clarity and aesthetics of their plots, applying best practices in data visualization.

3. Analyze and interpret visual data representations to extract meaningful insights, effectively communicating findings through well-structured graphical presentations.

**Part 1**: Perform the following codes, and understand the difference between line plot, scatter plot, histogram, bar chart, box plot, and pie chart using matplotlib’s pyplot sub-module. **(Provide a screenshot of your output.)**

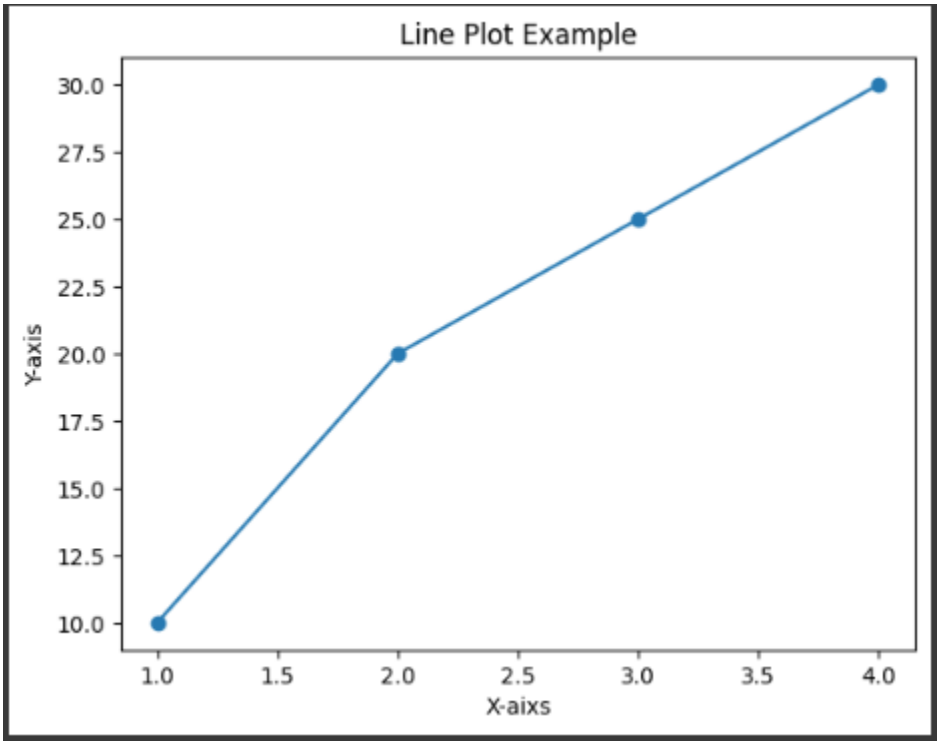
1. Line Plot

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
plt.plot(x, y)
plt.title("Line Plot Example")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.show()
```



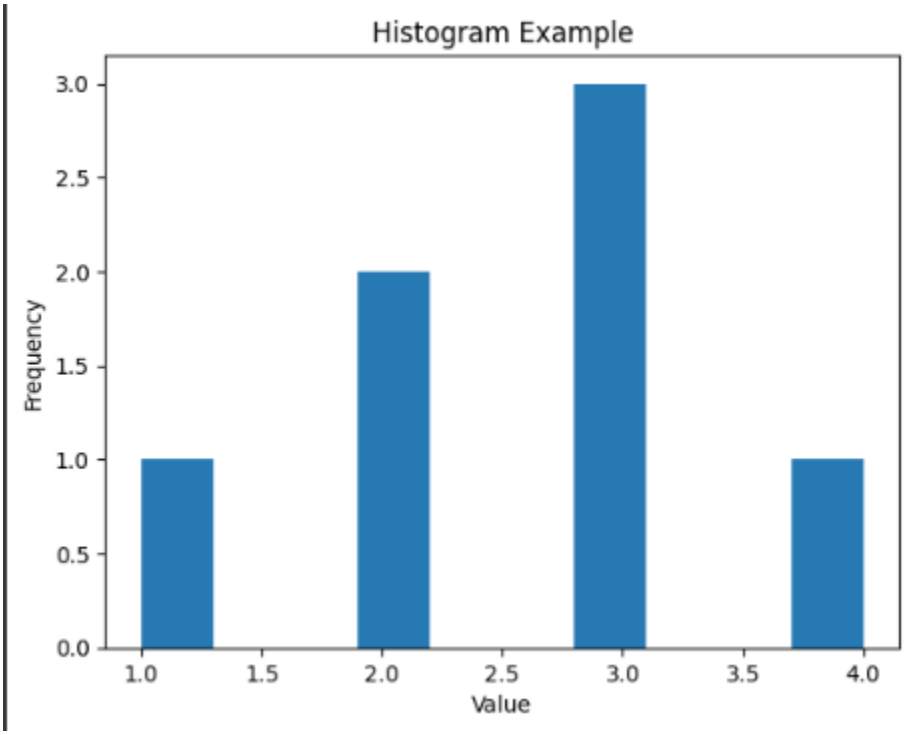
2. Scatter Plot

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
plt.scatter(x, y)
plt.title("Scatter Plot Example")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.show()
```



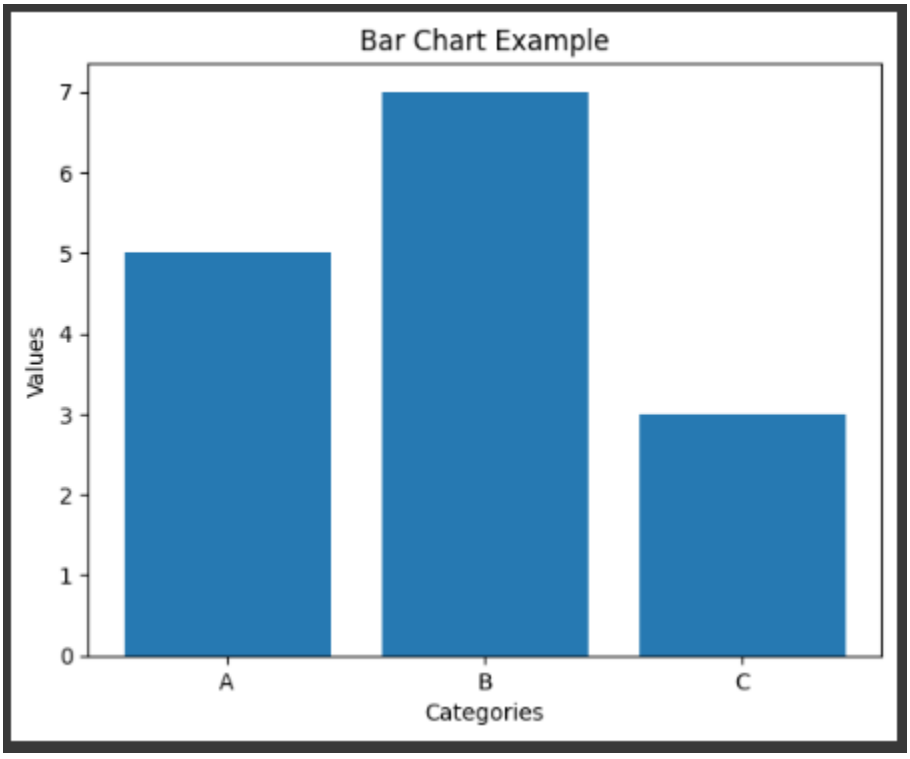
3. Histogram

```
import matplotlib.pyplot as plt

data = [1, 2, 2, 3, 3, 3, 4]
plt.hist(data)
plt.title("Histogram Example")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.show()
```



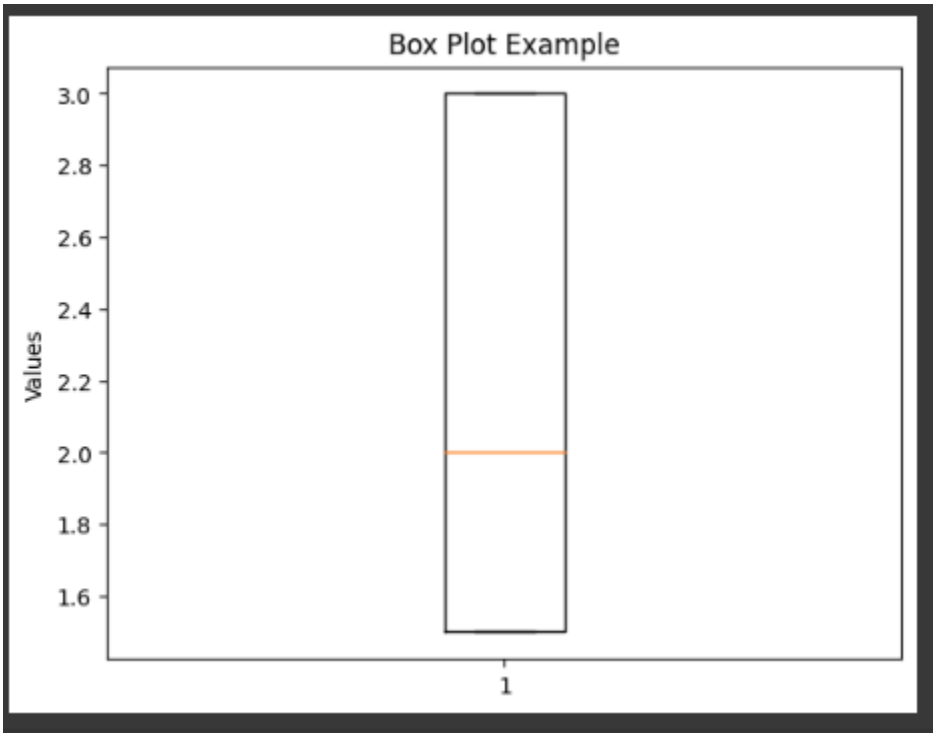
4. Bar Chart

```
import matplotlib.pyplot as plt

categories = ['A', 'B', 'C']
values = [5, 7, 3]
plt.bar(categories, values)
plt.title("Bar Chart Example")
plt.xlabel("Categories")
plt.ylabel("Values")
plt.show()
```



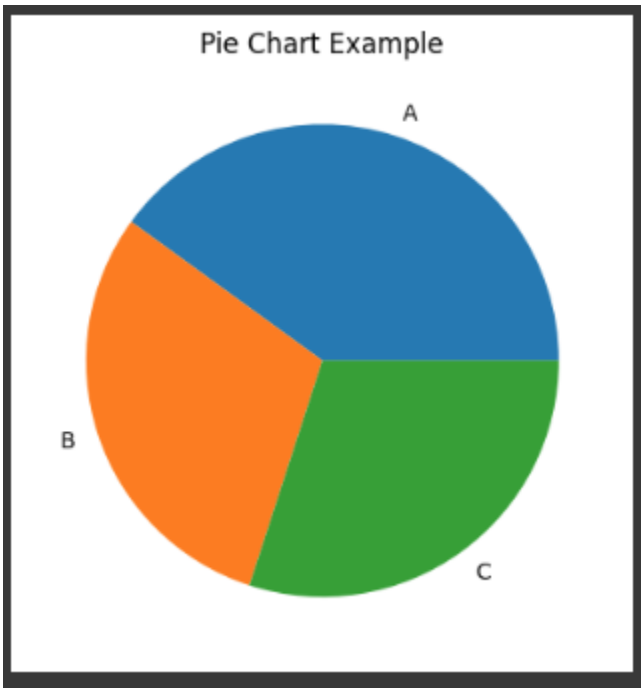
5. Box plot

```
import matplotlib.pyplot as plt

data = [[1.5]*10 + [2]*10 + [3]*10]

plt.boxplot(data)

plt.title("Box Plot Example")
plt.ylabel("Values")
plt.show()
```



6. Pie chart

```
import matplotlib.pyplot as plt

labels = ['A', 'B', 'C']
sizes = [40, 30, 30]

plt.pie(sizes, labels=labels)
plt.title("Pie Chart Example")
plt.show()
```




In matplotlib's pyplot, each plot has a specific purpose. A line plot connects points with lines, making it great for showing how things change over time. A scatter plot shows individual dots and is useful for spotting relationships between two variables. A histogram groups data into ranges to show how frequently values appear, while a bar chart compares different categories using rectangular bars. A box plot is handy for showing the data spread, including the median and any outliers. Finally, a pie chart breaks things into slices to show how much each part contributes to the whole.

**Part 2: Refer to the instructions below.**

1. **Find a dataset for this activity**: Please visit Kaggle and look for a new dataset that would allow you to perform visualization and analysis using matplotlib.

2. **Creating a dataframe from your CSV file:** Once you have successfully loaded your dataset, you need to create a dataframe from your uploaded CSV file

3. **Import the matplotlib.pyplot**

4. **Based on your chosen dataset, you will develop three questions that you will answer using pyplot visualizations. This means that you will need to produce at least three pyplot visualizations. You are also required to make certain customizations on your data vizes.**

5. Provide observations for each of your data viz, then **produce one insight not longer than five sentences given your three observations**. Your output shall follow this outline:
    a. Introduction (Describe your dataset)
    b. Questions
    c. Visualization and Observation
    d. Insight

6. Your grade  will depend on the quality of the question, difficulty/complexity of the visualization, and value-add of the insight that you will generate.

**Introduction**

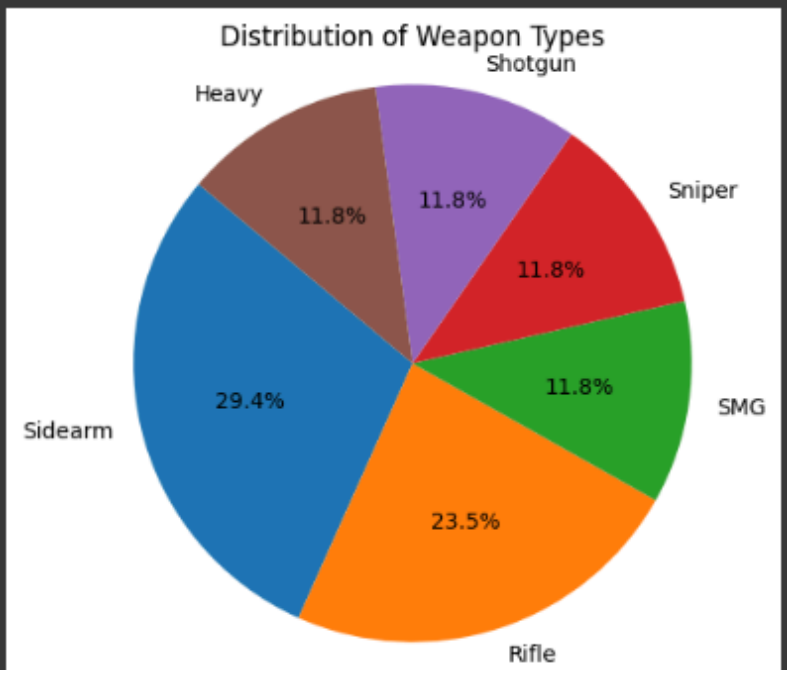
The dataset contains information on the weapons in the first-person shooter game Valorant, detailing both their descriptions and damage statistics. It includes 15 columns and 17 rows, each labeled accordingly. The columns provide data on various attributes such as weapon Name, Type, Price, Fire Rate, Wall Penetration, Magazine Capacity, and damage dealt to different body parts (head, body, legs) across three range categories: close, medium, and long. The damage values are further broken down for each range, making it easy to analyze the effectiveness of each weapon.

**Questions**

1. What percentage of the total distribution do different types of weapons represent?
2. What is the distribution of wall penetration?
3. What are the differences between head, body, and leg damage at close range in the dataset, and how do their distributions compare in terms of frequency?

1. What percentage of the total distribution do different types of weapons represent?

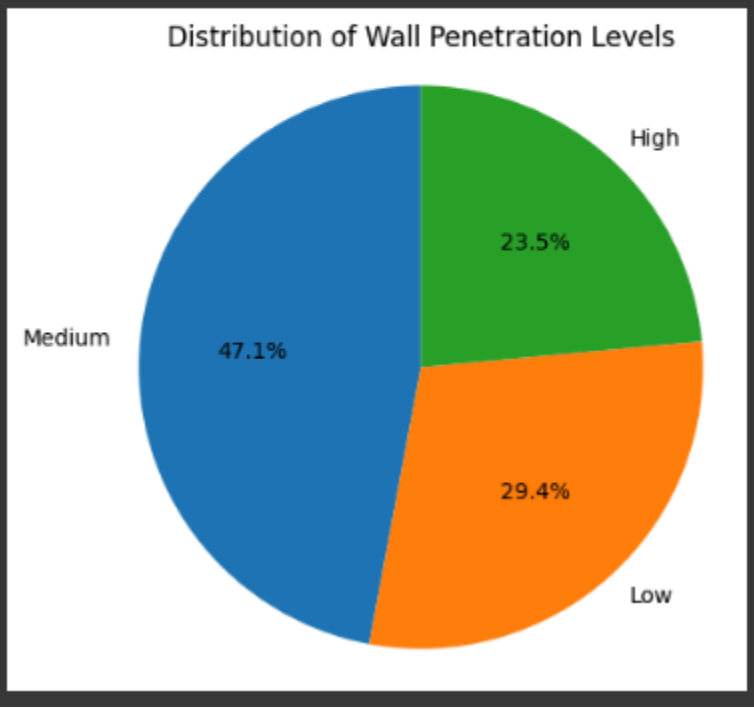
```
# Count the occurrences of each weapon type
weapon_counts = df['Weapon Type'].value_counts()

# Create a pie chart
plt.figure(figsize=(5, 5))
plt.pie(weapon_counts, labels=weapon_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Weapon Types')
plt.axis('equal')  # Equal aspect ratio ensures the pie chart is circular.
plt.show()
```




Observation:
Based on the pie chart, the Sidearm has the largest portion, indicating it's the most common weapon type in the dataset. Rifle would also have a significant share, while Heavy, Sniper, and SMG have the least and equal percentages.

2. What is the distribution of wall penetration?

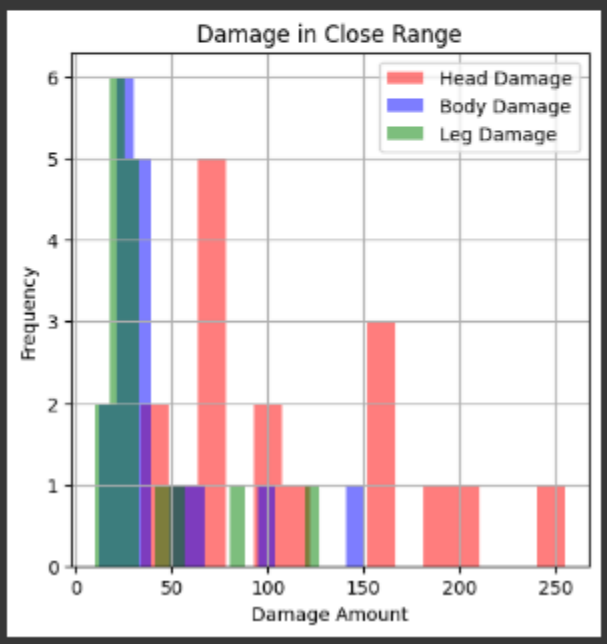
```
counts = df['Wall Penetration'].value_counts()
plt.figure(figsize=(5, 5))
plt.pie(counts, labels=counts.index, autopct='%1.1f%%', startangle=90)
plt.title('Distribution of Wall Penetration Levels')
plt.axis('equal')  # Equal aspect ratio ensures that pie chart is circular.
plt.show()
```




Observation:
Based on the pie chart, the Medium penetration level occupies the largest portion, indicating that it is the most prevalent wall penetration level in the game. This is followed by Low, while High penetration is the least represented.

3. What are the differences between head, body, and leg damage at close range in the dataset, and how do their distributions compare in terms of frequency?

```
plt.figure(figsize=(5, 5))
plt.hist(df['HDMG_0'], bins=15, alpha=0.5, label='Head Damage', color='red')
plt.hist(df['BDMG_0'], bins=15, alpha=0.5, label='Body Damage', color='blue')
plt.hist(df['LDMG_0'], bins=15, alpha=0.5, label='Leg Damage', color='green')

# Add titles and labels
plt.title('Damage in Close Range')
plt.xlabel('Damage Amount')
plt.ylabel('Frequency')
plt.legend(loc='upper right')
plt.grid(True)

# Show the plot
plt.show()
```




Observation:
Head damage (HDMG_0) stands out with a wider range and higher peaks, suggesting it's often more severe. On the other hand, body damage (BDMG_0) tends to cluster around lower values, indicating it's generally less intense. Leg damage (LDMG_0) shows a similar pattern, with most instances at lower levels but a few spikes in severity.

**Insight:**
The analysis shows that Sidearms are the most prominent weapon in Valorant, which speaks to their importance in the game. Medium wall penetration is also the most frequent, indicating players often balance cover with damage. Headshots deal way more damage than body or leg shots, highlighting why aiming for the head is crucial. These insights can help players make smarter choices about weapons and where to aim.