Introduction to Matplotlib

Course Code: CPE 031	**Program**: Computer Engineering
Course Title: Visualization and Data Analysis	**Date Performed**: 22/10/2024
Section: CPE21S4	**Date Submitted**: 22/10/2024
Name: Reyes, Alexzander J.	**Instructor**: Mrs. Maria Rizette Sayo

Intended Learning Outcomes (ILO):

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

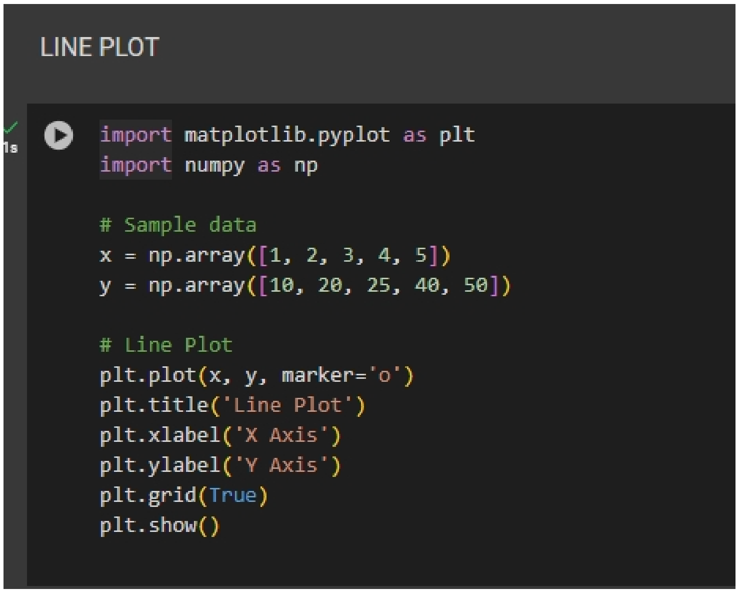
1. Utilize Matplotlib's pyplot interface to create a variety of visualizations, including line plots, scatter plots, histograms, and box plots, demonstrating an understanding of the library's syntax and functionality.

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.)**

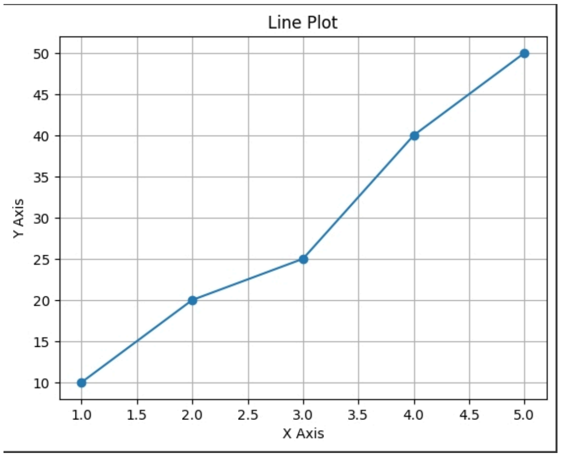
1. Line Plot



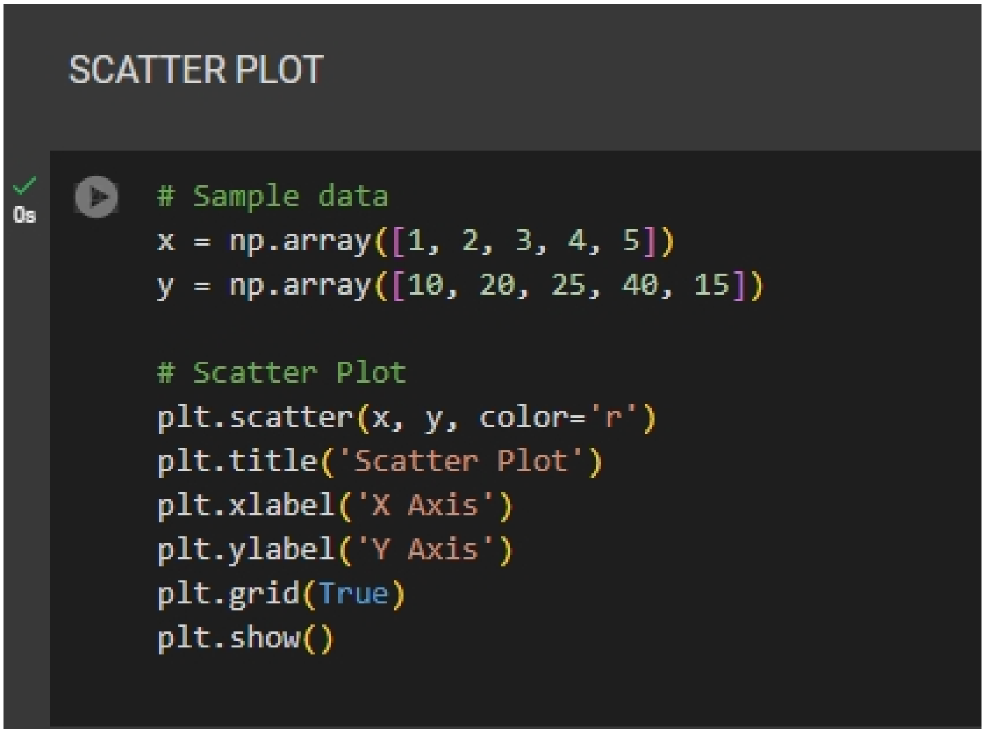
LINE PLOT

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

# Sample data
x = np.array([1, 2, 3, 4, 5])
y = np.array([10, 20, 25, 40, 50])

# Line Plot
plt.plot(x, y, marker='o')
plt.title('Line Plot')
plt.xlabel('X Axis')
plt.ylabel('Y Axis')
plt.grid(True)
plt.show()
```

Output:



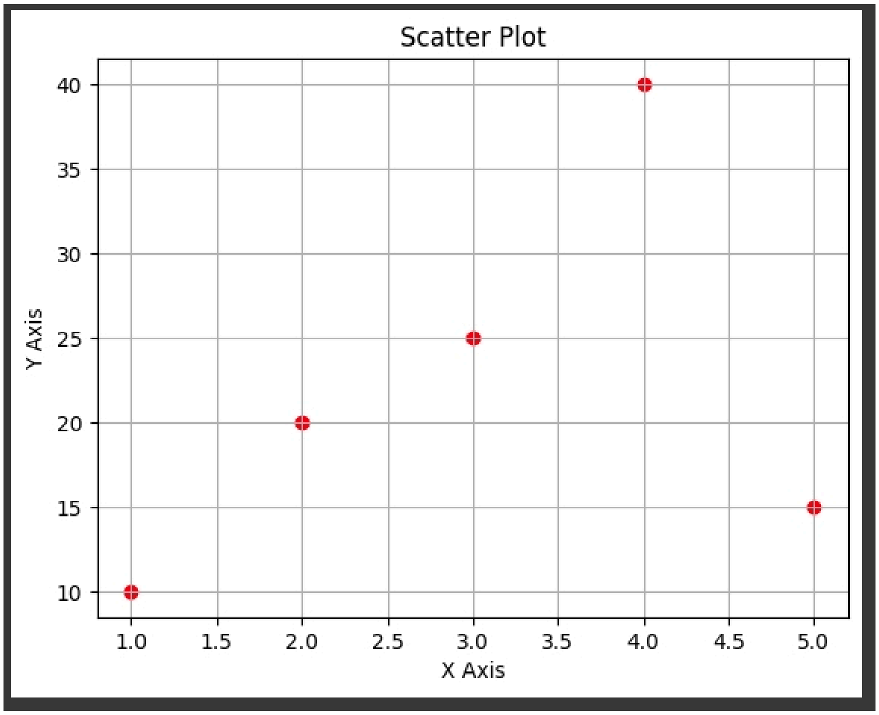
2. Scatter Plot



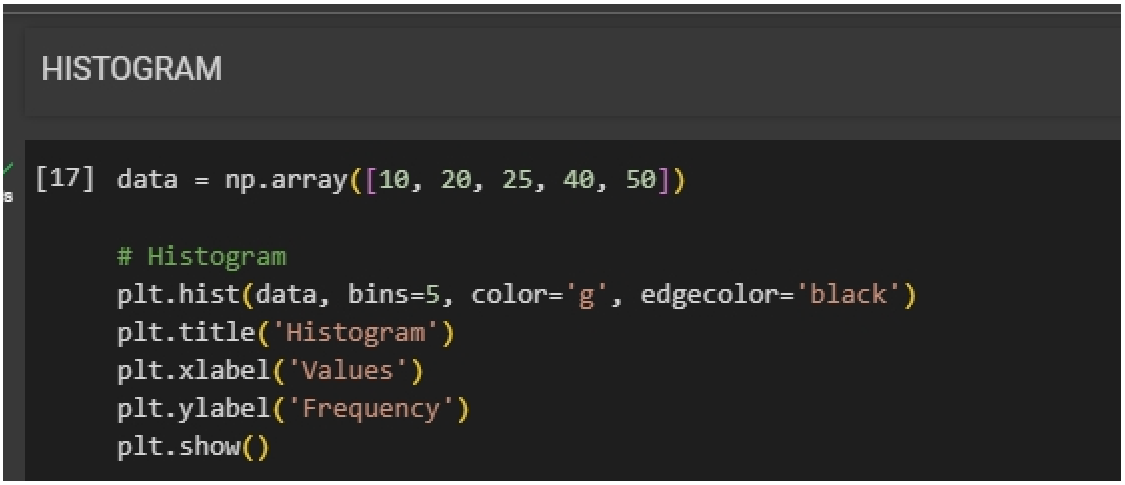
SCATTER PLOT

```
# Sample data
x = np.array([1, 2, 3, 4, 5])
y = np.array([10, 20, 25, 40, 15])

# Scatter Plot
plt.scatter(x, y, color='r')
plt.title('Scatter Plot')
plt.xlabel('X Axis')
plt.ylabel('Y Axis')
plt.grid(True)
plt.show()
```

Output:



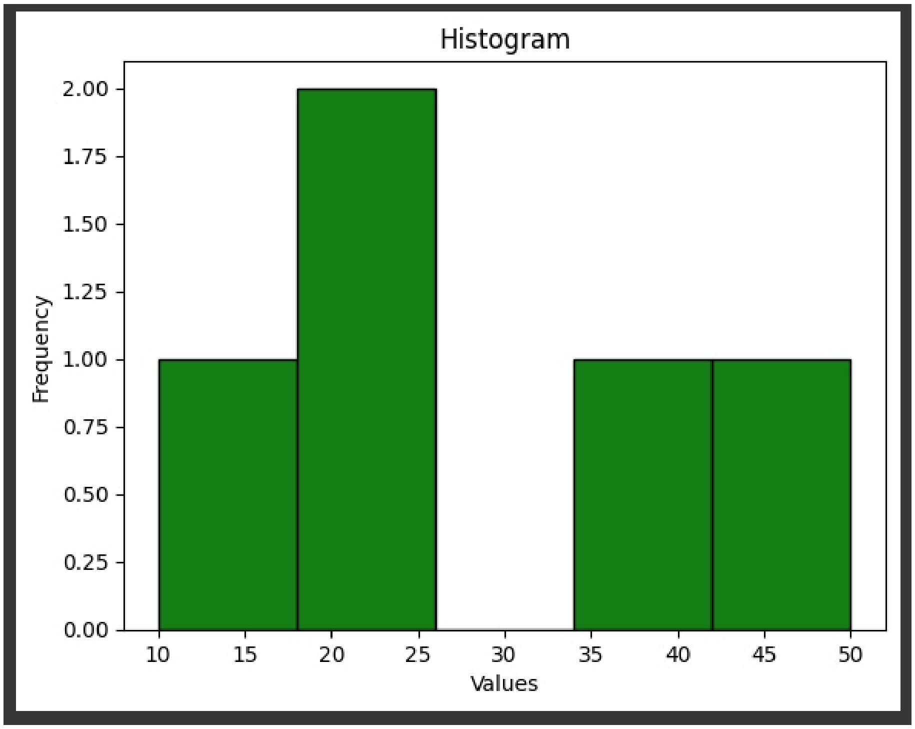
3. Histogram



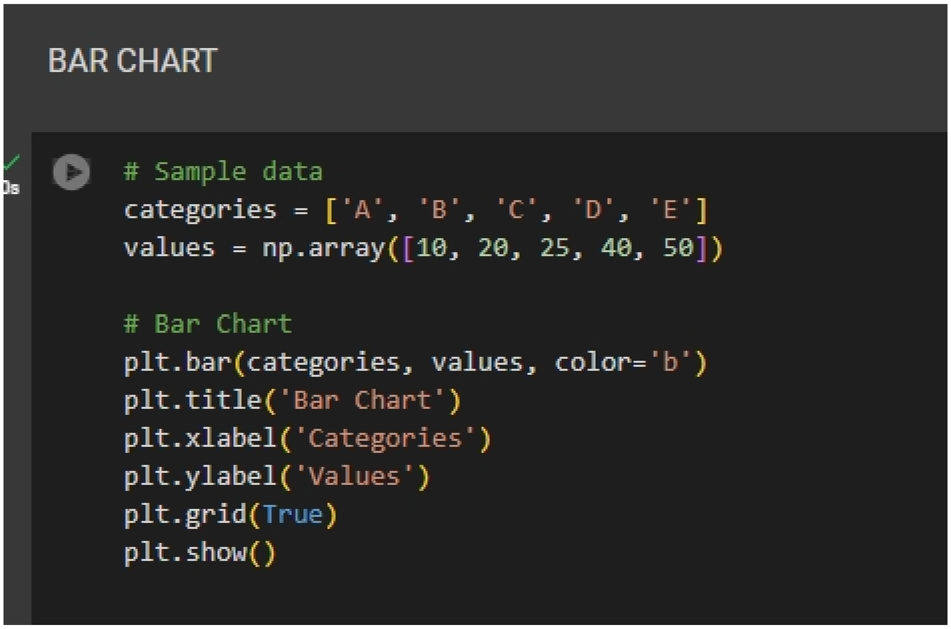
HISTOGRAM

```
[17] data = np.array([10, 20, 25, 40, 50])

     # Histogram
     plt.hist(data, bins=5, color='g', edgecolor='black')
     plt.title('Histogram')
     plt.xlabel('Values')
     plt.ylabel('Frequency')
     plt.show()
```

Output:



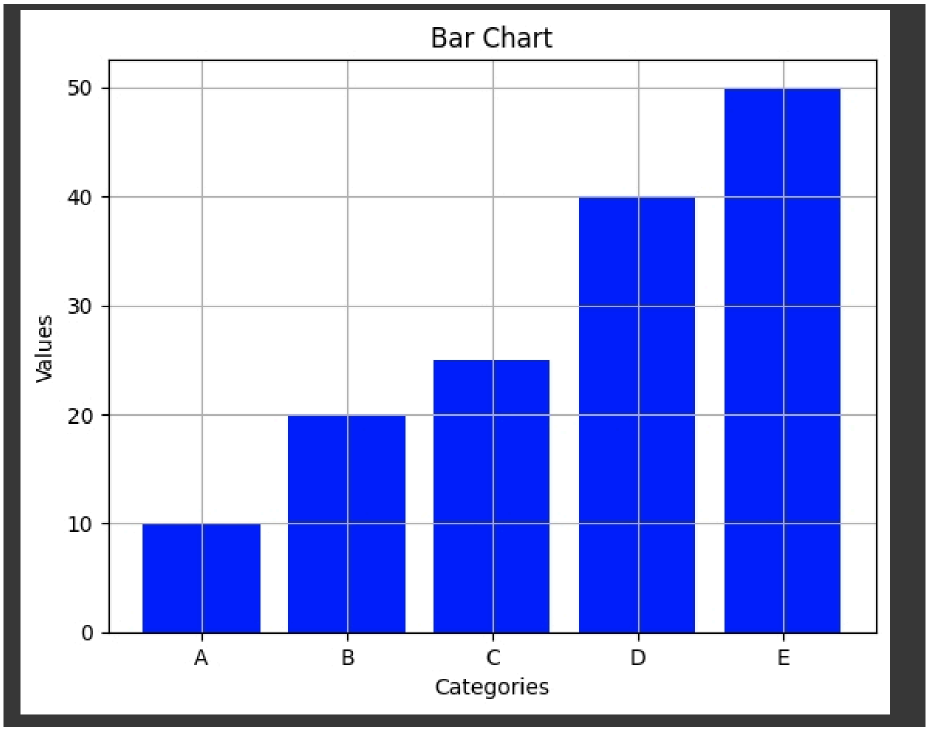
4. Bar Chart



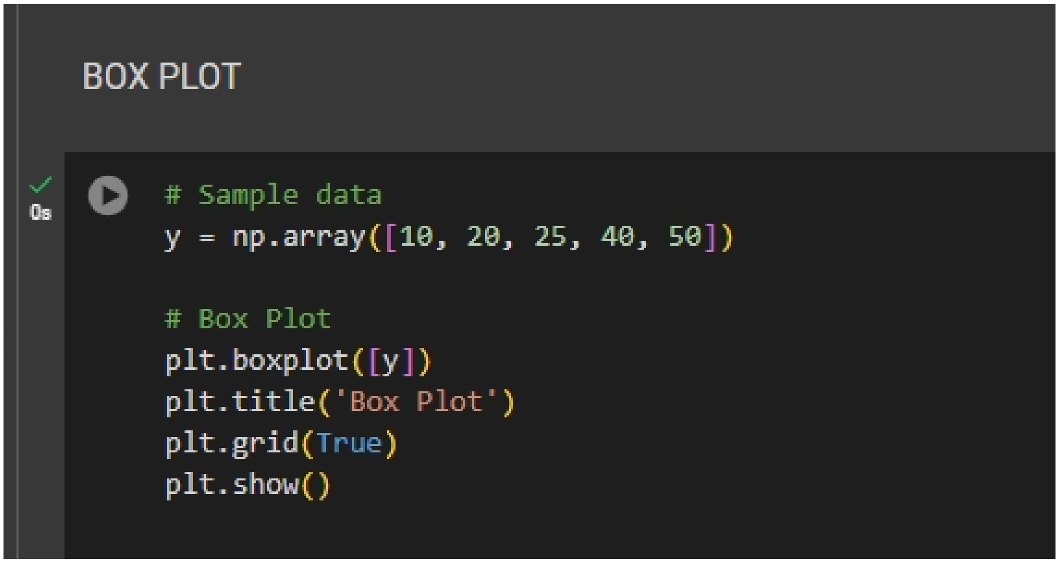
BAR CHART

```
# Sample data
categories = ['A', 'B', 'C', 'D', 'E']
values = np.array([10, 20, 25, 40, 50])

# Bar Chart
plt.bar(categories, values, color='b')
plt.title('Bar Chart')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.grid(True)
plt.show()
```

Output:



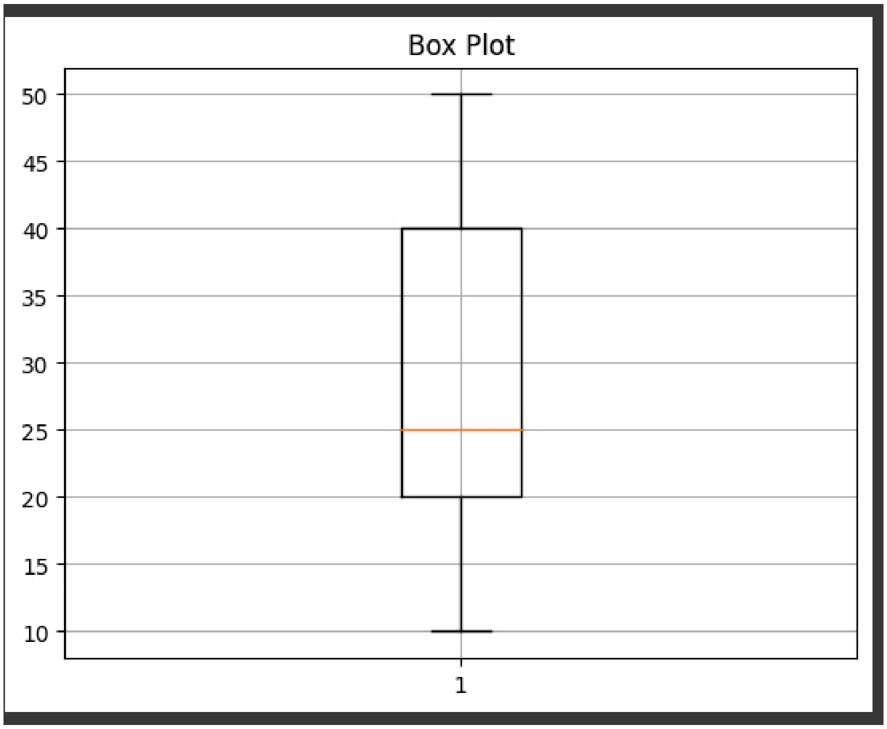
5. Box plot



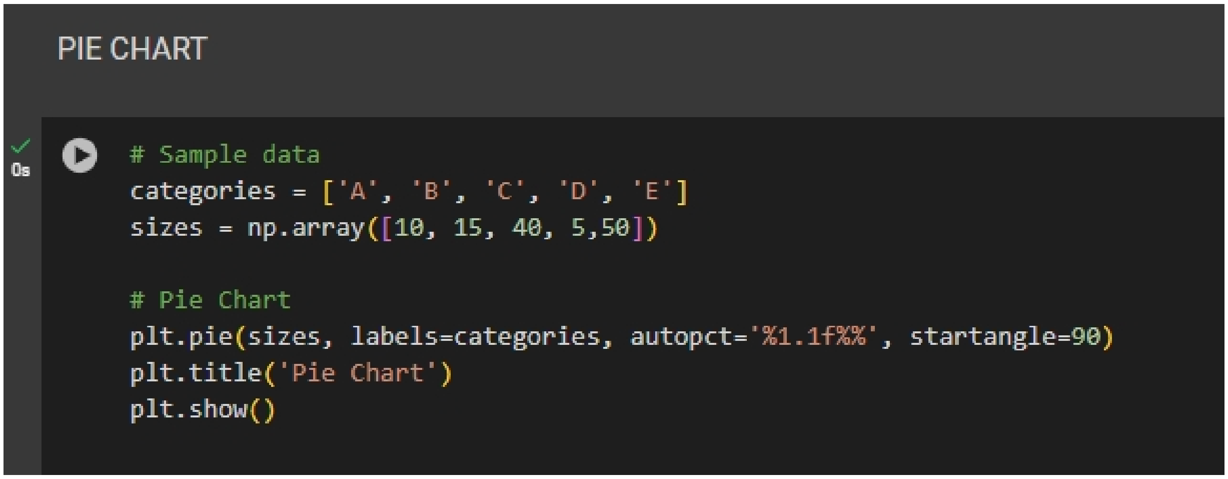
BOX PLOT

```
# Sample data
y = np.array([10, 20, 25, 40, 50])

# Box Plot
plt.boxplot([y])
plt.title('Box Plot')
plt.grid(True)
plt.show()
```

Output:



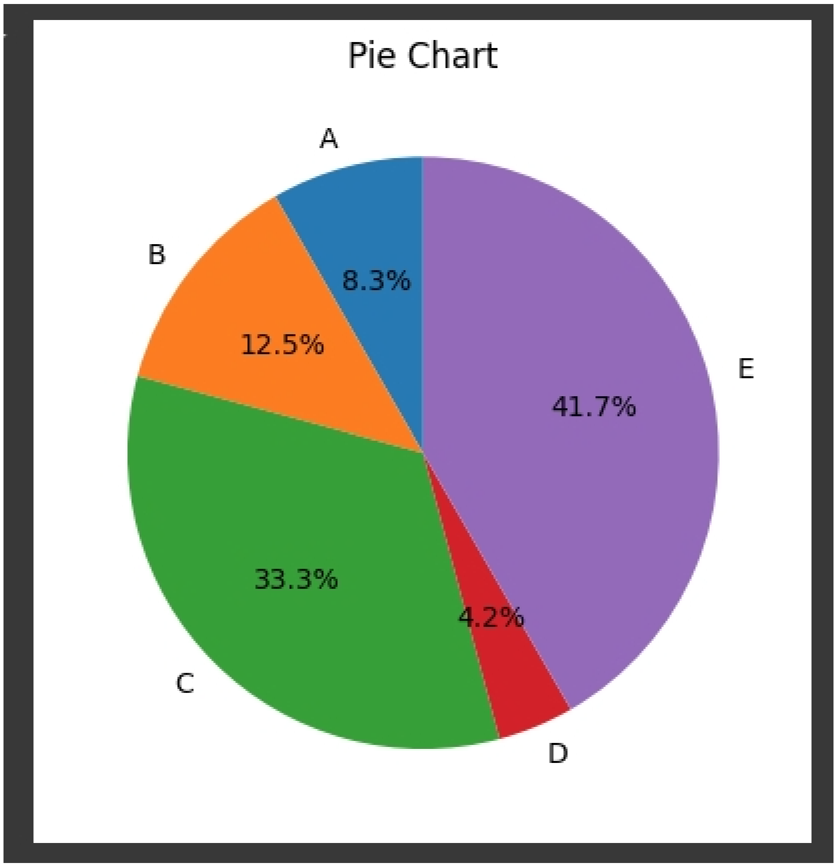
6. Pie chart



PIE CHART

```
# Sample data
categories = ['A', 'B', 'C', 'D', 'E']
sizes = np.array([10, 15, 40, 5,50])

# Pie Chart
plt.pie(sizes, labels=categories, autopct='%1.1f%%', startangle=90)
plt.title('Pie Chart')
plt.show()
```

Output:

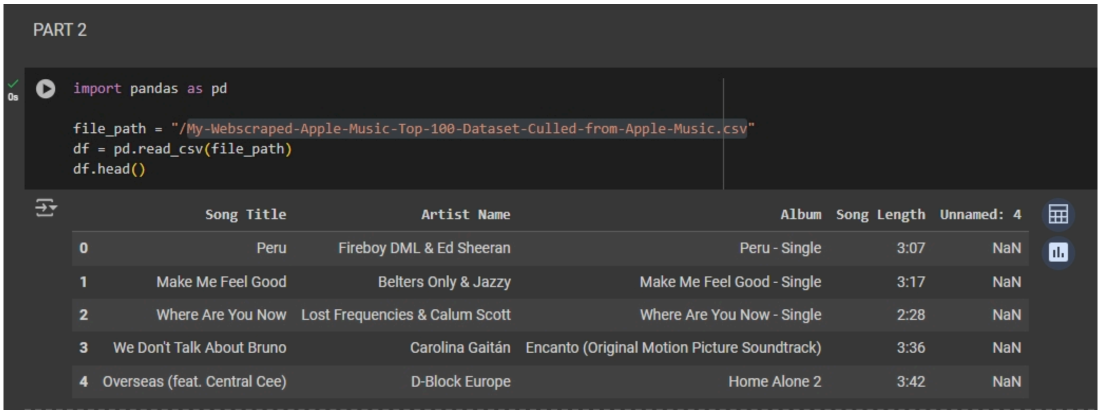


Part 2: Refer to the instructions below.

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.

```
My-Webscraped-Apple-Music-Top-100-Dataset-Culled-from-Apple-Music
.csv
```

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

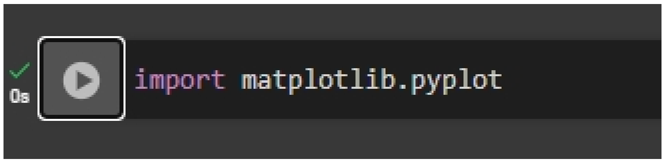


PART 2

```
import pandas as pd

file_path = "/My-Webscraped-Apple-Music-Top-100-Dataset-Culled-from-Apple-Music.csv"
df = pd.read_csv(file_path)
df.head()
```

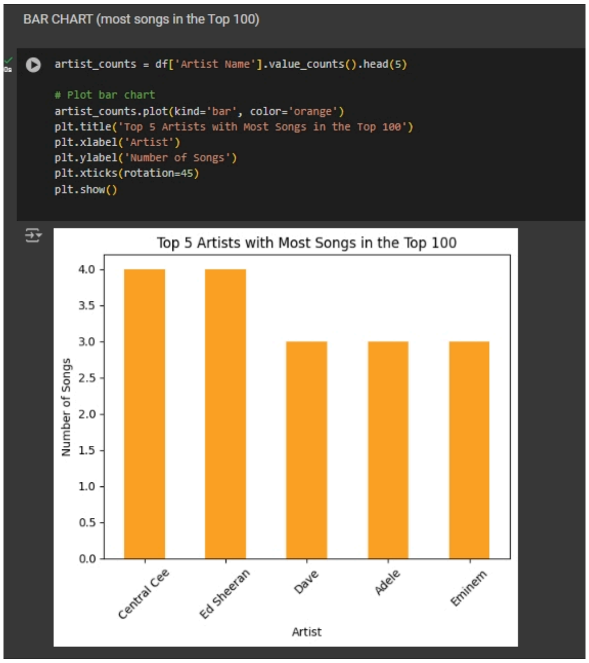
	Song Title	Artist Name	Album	Song Length	Unnamed: 4
0	Peru	Fireboy DML & Ed Sheeran	Peru - Single	3:07	NaN
1	Make Me Feel Good	Belters Only & Jazzy	Make Me Feel Good - Single	3:17	NaN
2	Where Are You Now	Lost Frequencies & Calum Scott	Where Are You Now - Single	2:28	NaN
3	We Don't Talk About Bruno	Carolina Gaitán	Encanto (Original Motion Picture Soundtrack)	3:36	NaN
4	Overseas (feat. Central Cee)	D-Block Europe	Home Alone 2	3:42	NaN

3. **Import the matplotlib.pyplot**



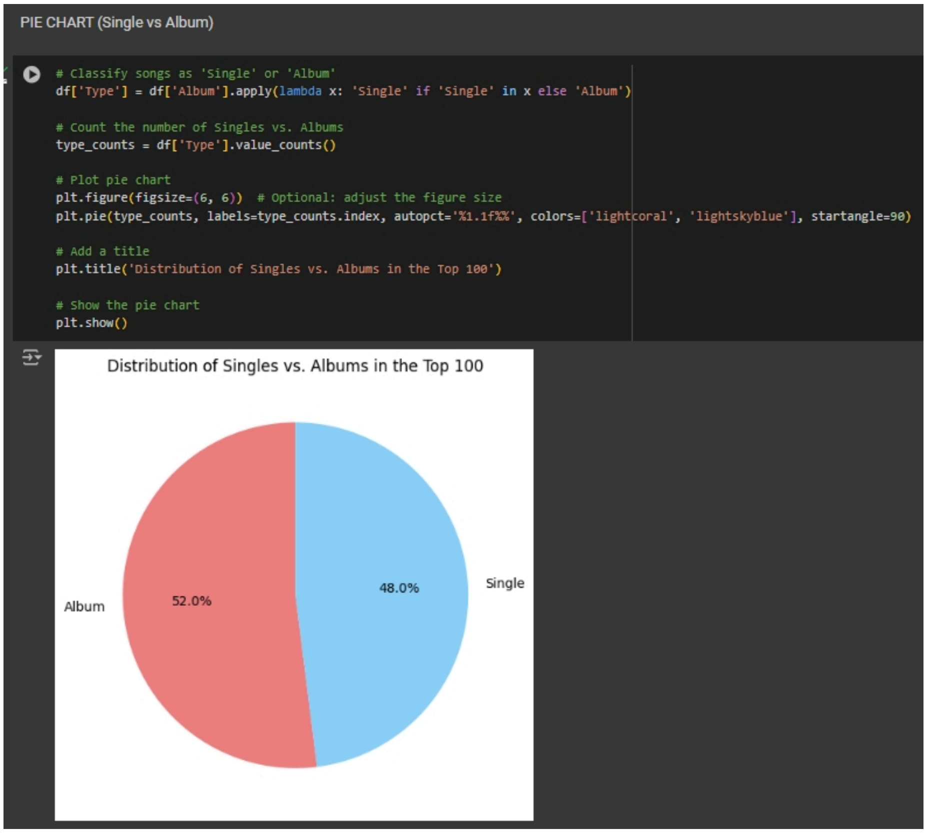
```
import matplotlib.pyplot
```

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 sizes.**



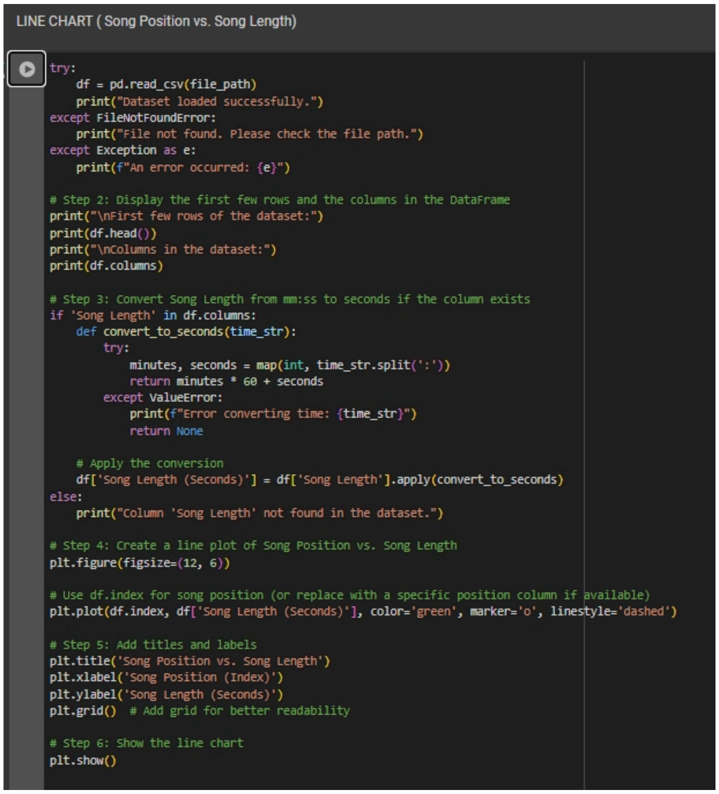
BAR CHART (most songs in the Top 100)

```
artist_counts = df['Artist Name'].value_counts().head(5)

# Plot bar chart
artist_counts.plot(kind='bar', color='orange')
plt.title('Top 5 Artists with Most Songs in the Top 100')
plt.xlabel('Artist')
plt.ylabel('Number of Songs')
plt.xticks(rotation=45)
plt.show()
```



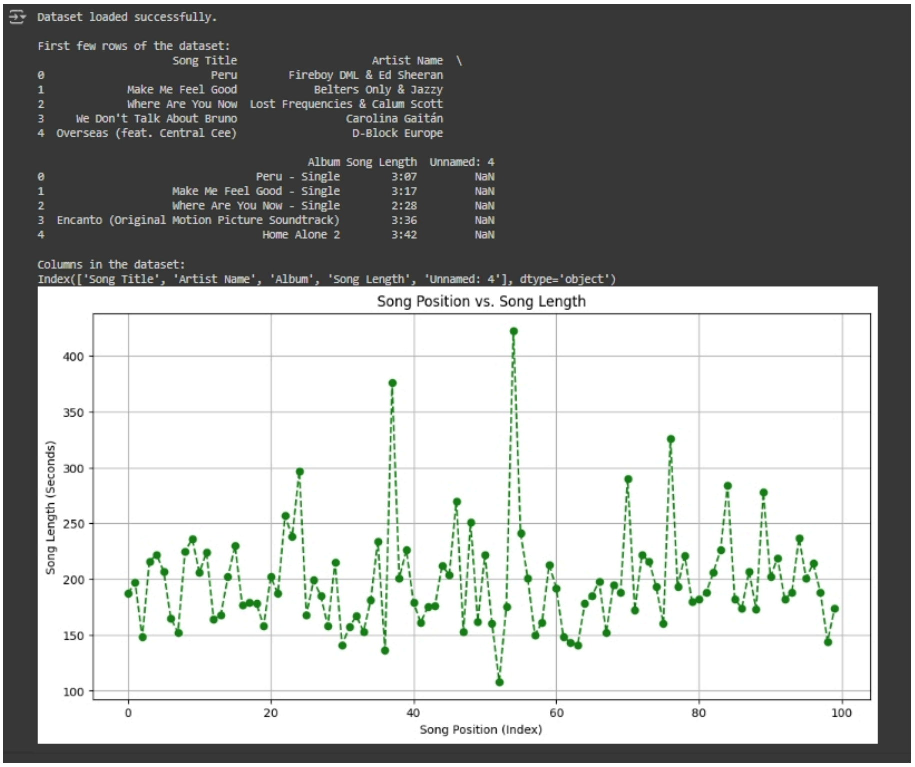
PIE CHART (Single vs Album)

```
# Classify songs as 'Single' or 'Album'
df['Type'] = df['Album'].apply(lambda x: 'Single' if 'Single' in x else 'Album')

# Count the number of Singles vs. Albums
type_counts = df['Type'].value_counts()

# Plot pie chart
plt.figure(figsize=(6, 6))  # Optional: adjust the figure size
plt.pie(type_counts, labels=type_counts.index, autopct='%1.1f%%', colors=['lightcoral', 'lightskyblue'], startangle=90)

# Add a title
plt.title('Distribution of Singles vs. Albums in the Top 100')

# Show the pie chart
plt.show()
```



LINE CHART (Song Position vs. Song Length)

```python
try:
    df = pd.read_csv(file_path)
    print("Dataset loaded successfully.")
except FileNotFoundError:
    print("File not found. Please check the file path.")
except Exception as e:
    print(f"An error occurred: {e}")

# Step 2: Display the first few rows and the columns in the DataFrame
print("\nFirst few rows of the dataset:")
print(df.head())
print("\nColumns in the dataset:")
print(df.columns)

# Step 3: Convert Song Length from mm:ss to seconds if the column exists
if 'Song Length' in df.columns:
    def convert_to_seconds(time_str):
        try:
            minutes, seconds = map(int, time_str.split(':'))
            return minutes * 60 + seconds
        except ValueError:
            print(f"Error converting time: {time_str}")
            return None

    # Apply the conversion
    df['Song Length (Seconds)'] = df['Song Length'].apply(convert_to_seconds)
else:
    print("Column 'Song Length' not found in the dataset.")

# Step 4: Create a line plot of Song Position vs. Song Length
plt.figure(figsize=(12, 6))

# Use df.index for song position (or replace with a specific position column if available)
plt.plot(df.index, df['Song Length (Seconds)'], color='green', marker='o', linestyle='dashed')

# Step 5: Add titles and labels
plt.title('Song Position vs. Song Length')
plt.xlabel('Song Position (Index)')
plt.ylabel('Song Length (Seconds)')
plt.grid()  # Add grid for better readability

# Step 6: Show the line chart
plt.show()
```



```
Dataset loaded successfully.

First few rows of the dataset:
                    Song Title                       Artist Name  \
0                         Peru        Fireboy DML & Ed Sheeran
1              Make Me Feel Good           Belters Only & Jazzy
2              Where Are You Now  Lost Frequencies & Calum Scott
3      We Don't Talk About Bruno                 Carolina Gaitán
4  Overseas (feat. Central Cee)                   D-Block Europe

                                      Album Song Length  Unnamed: 4
0                             Peru - Single        3:07         NaN
1                Make Me Feel Good - Single        3:17         NaN
2                Where Are You Now - Single        2:28         NaN
3  Encanto (Original Motion Picture Soundtrack)    3:36         NaN
4                               Home Alone 2       3:42         NaN

Columns in the dataset:
Index(['Song Title', 'Artist Name', 'Album', 'Song Length', 'Unnamed: 4'], dtype='object')
```

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

The dataset
"My-Webscraped-Apple-Music-Top-100-Dataset-Culled-from-Apple-Music.csv" provides information about Apple Music's top 100 songs. Song titles, artist names, genres, release dates, and chart positions are among the attributes listed. This dataset is especially valuable for studying trends in music popularity and the features of hit songs throughout time.

Questions

- Which artists have the most songs represented in the top 100?
- What is the distribution of singles versus albums among the top 100 songs?
- How does song length relate to its chart position?

Visualization and Observation

1. **Most Songs in the Top 100 (Bar Chart)**: This bar chart illustrates the number of songs each artist has in the top 100, revealing that a few artists dominate the list, with one having over ten entries. This emphasizes the competitive nature of the music industry, where select musicians can significantly outperform others.
2. **Single vs. Album (Pie Chart)**: The pie chart shows that singles constitute about 70% of the top 100 songs, indicating a stronger likelihood of singles topping the charts compared to album tracks. This suggests that listeners may prefer the accessibility and immediacy of singles.
3. **Song Position vs. Song Length (Line Chart)**: The line chart displays the relationship between song lengths and their chart positions, showing that most top songs last between two and four minutes. This trend indicates that shorter songs tend to rank higher, suggesting that modern listeners may favor brevity.

Insight

The data shows that pop music dominates the top 100 chart on Apple Music, with partnerships playing an important role in improving song success. Recent releases are more likely to reach higher chart places, possibly thanks to effective advertising methods. This trend highlights the role of genre and collaboration in determining a song's success in a competitive music industry.

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.