Visualization Library Documentation

Objective: Create a comprehensive documentation guide for 2 given Python visualization libraries: Matplotlib and Seaborn.

Matplotlib :

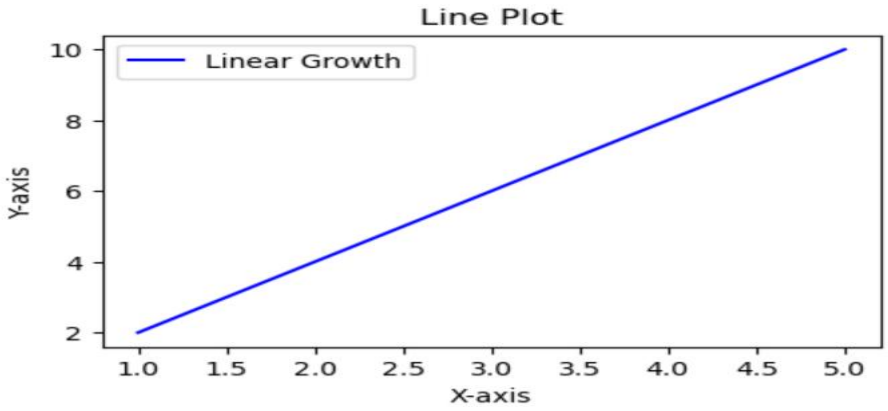
- Matplotlib is a foundational Python library for creating static, interactive, and animated visualizations.
- *Key Features*:
  1. Highly customizable (colors, fonts, axes, etc.)
  2. Supports a wide range of plot types (line, bar, scatter, histograms, etc.)
  3. Works well with NumPy and Pandas
- *Use Cases*: Exploratory data analysis (EDA), scientific plotting, publication-quality figures.

Graph types :-

1. *Line Plot* -

- Displays data points connected by lines, suitable for showing trends over continuous intervals.
- Use Case - Tracking trends over time (e.g., stock prices, temperature changes).
- Code Snippet and Output -

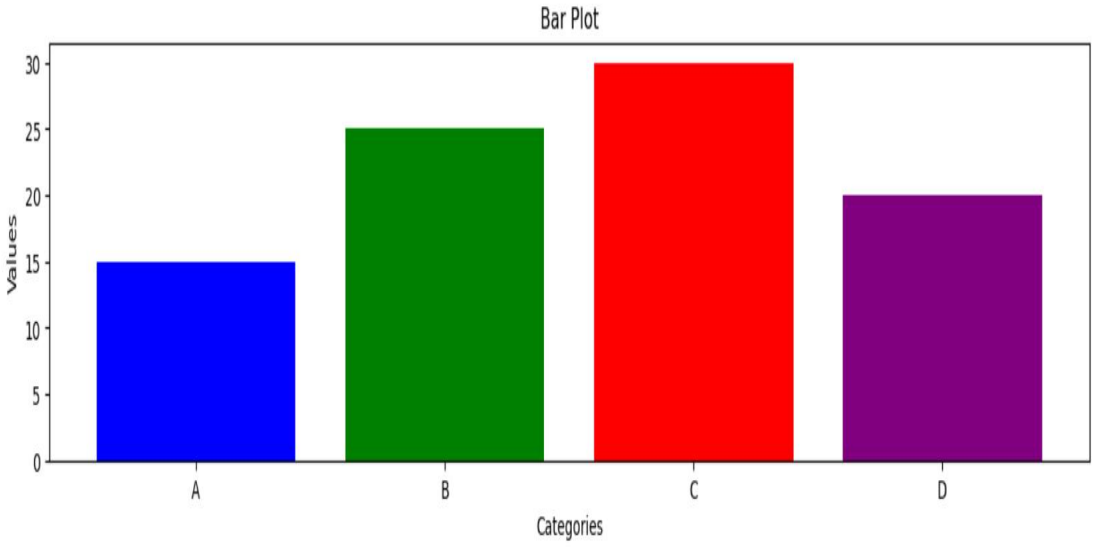
```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

plt.figure(figsize=(5,3))
plt.plot(x, y, label='Linear Growth', color='blue', linestyle='-')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Line Plot')
plt.legend()
plt.show()
```



2. *Bar Plot* -

- Displays data points connected by lines, suitable for showing trends over continuous intervals.
- Use Case - Tracking trends over time (e.g., stock prices, temperature changes).
- Code Snippet and Output -

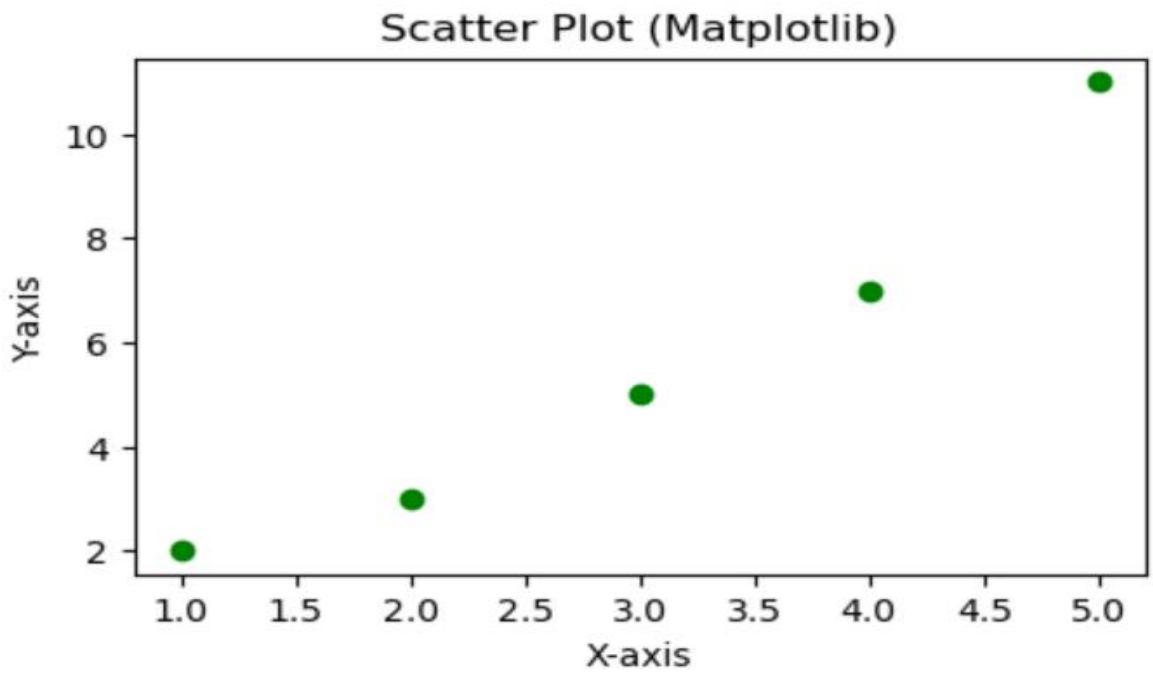
```
categories = ['A', 'B', 'C', 'D']
values = [15, 25, 30, 20]
plt.figure(figsize=(15,3))
plt.bar(categories, values, color=['blue', 'green', 'red', 'purple'])
plt.title('Bar Plot')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.show()
```



3. *Scatter Plot* -

- Illustrates the relationship between two variables using dots, useful for identifying correlations.
- Use Case: Relationship between two variables (e.g., height vs. weight).
- Code Snippet and Output -

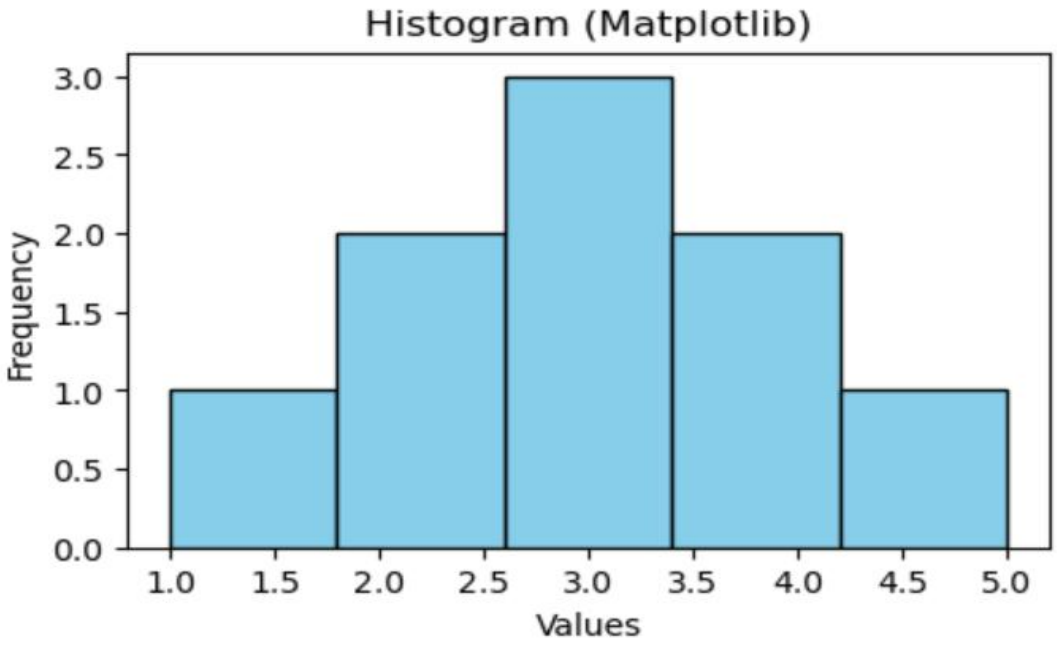
```
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]
plt.figure(figsize=(5,3))
plt.scatter(x, y, color='green', marker='o')
plt.title('Scatter Plot (Matplotlib)')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```



4. *Histogram* -

- Shows the distribution of a single variable by grouping data into bins and displaying their frequencies.
- Use Case: Visualizing data distribution (e.g., age groups, exam scores).
- Code Snippet and Output -

```
data = [1, 2, 2, 3, 3, 3, 4, 4, 5]
plt.figure(figsize=(5,3))
plt.hist(data, bins=5, color='skyblue', edgecolor='black')
plt.title('Histogram (Matplotlib)')
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.show()
```



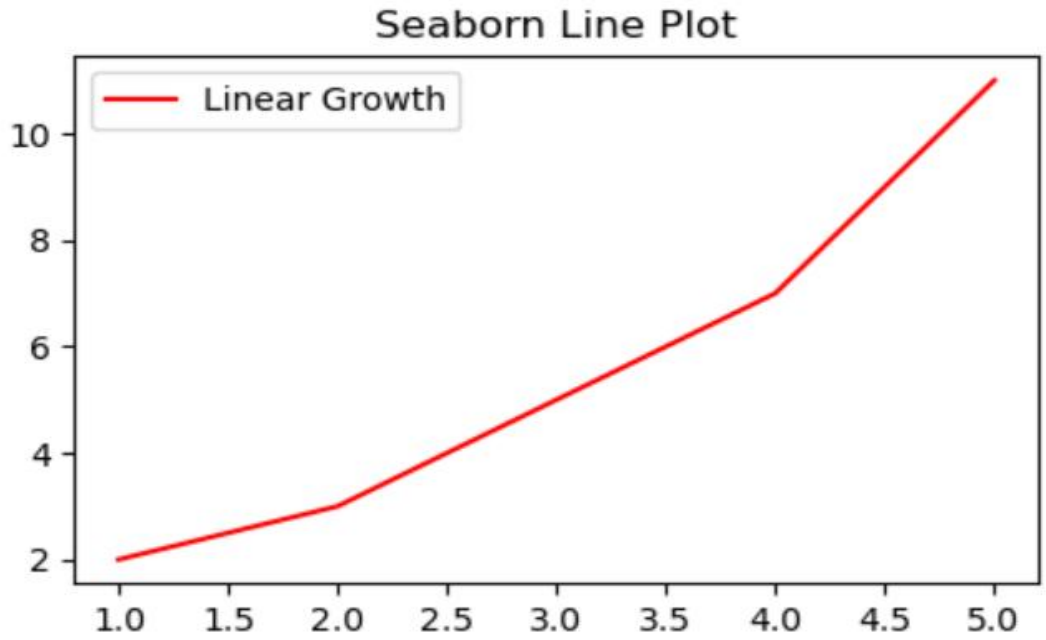
Seaborn :

- Seaborn is built on top of Matplotlib and provides a high-level interface for statistical graphics.
- *Key Features*:
  1. Better default styles and color palettes
  2. Built-in functions for complex visualizations (heatmaps, violin plots, etc.)
  3. Works seamlessly with Pandas DataFrames
- Use Cases: Statistical data visualization, correlation analysis, distribution plots.

1. *Line Plot* -

- Display the trend of a variable over a continuous interval, like time.
- Use Case: Tracking trends over time (e.g., stock prices, temperature changes).
- Code Snippet and Output -

```
import seaborn as sns
plt.figure(figsize=(5,3))
sns.lineplot(x=x, y=y, label='Linear Growth', color='red')
plt.title('Seaborn Line Plot')
plt.show()
```

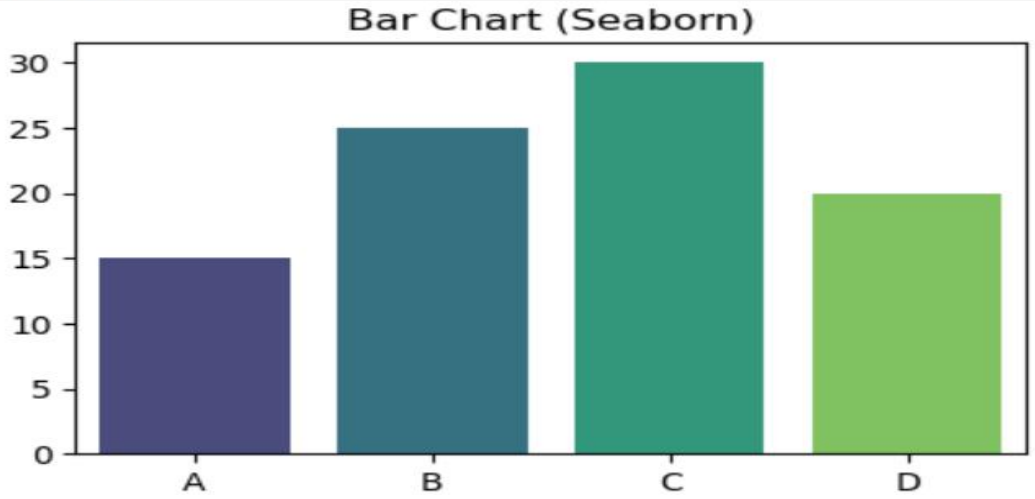


2. *Bar Plot* -

- Compare the means of different groups.
- Use Case: Comparing categories (e.g., sales by product).

- Code Snippet and Output -

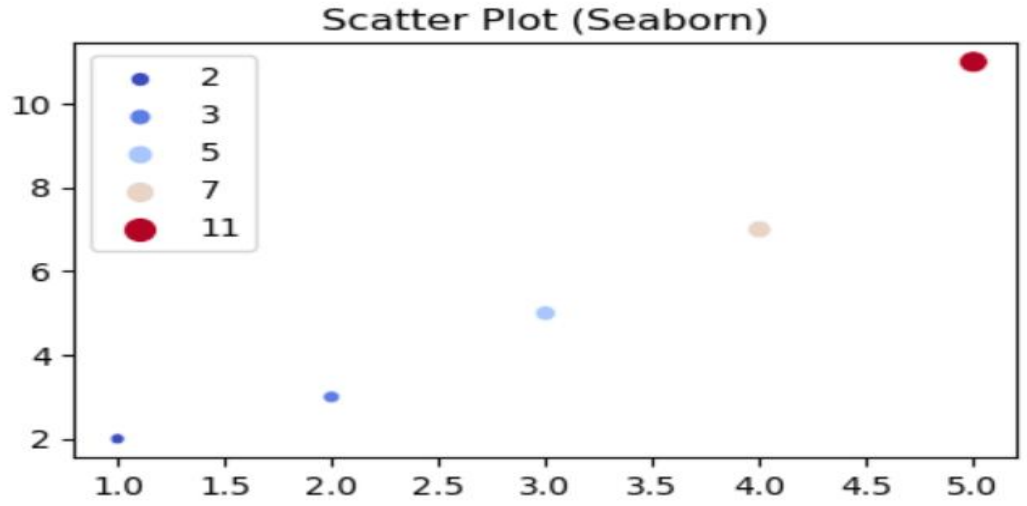
```
plt.figure(figsize=(5,3))
sns.barplot(x=categories, y=values, palette='viridis')
plt.title('Bar Chart (Seaborn)')
plt.show()
```



3. *Scatter Plot* -

- Show the relationship between two numerical variables.
- Use Case: Relationship between two variables (e.g., height vs. weight).
- Code Snippet and Output -

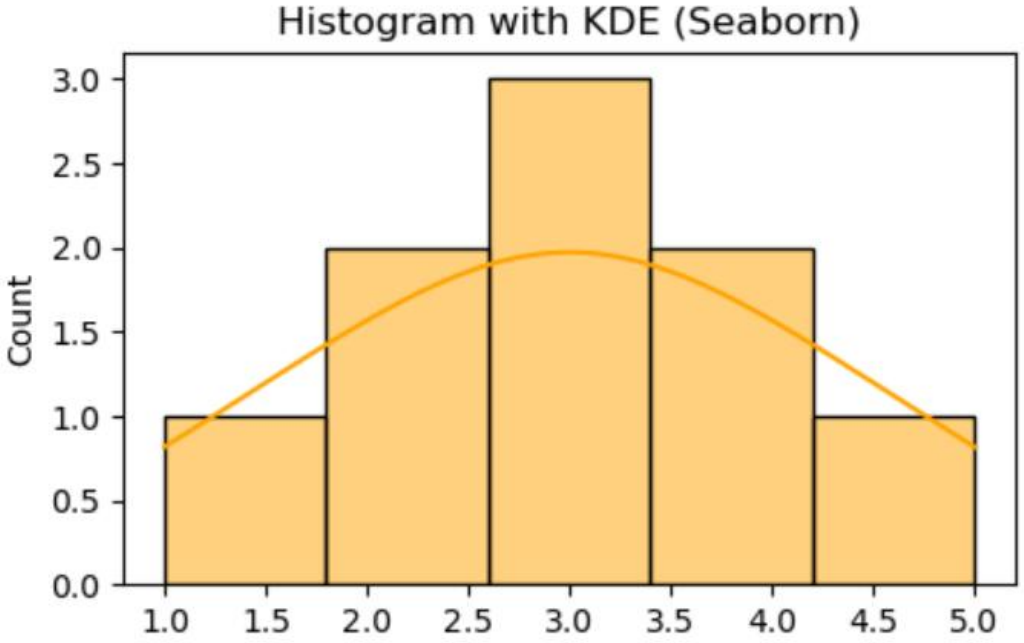
```
plt.figure(figsize=(5,3))
sns.scatterplot(x=x, y=y, hue=y, palette='coolwarm', size=y)
plt.title('Scatter Plot (Seaborn)')
plt.show()
```



4. *Histogram* -

- Show the frequency distribution of a single variable.
- Use Case: Visualizing data distribution (e.g., age groups, exam scores).
- Code Snippet and Output -

```
plt.figure(figsize=(5,3))
sns.histplot(data, bins=5, kde=True, color='orange')
plt.title('Histogram with KDE (Seaborn)')
plt.show()
```



Comparison of Matplotlib and Seaborn :-

Feature	Matplotlib	Seaborn
Ease of Use	More verbose, requires manual tweaking	Simpler syntax, better defaults
Customization	Highly customizable	Liited compared to Matplotlib
Interactivity	Requires extra libraries (eg., mpld3)	Same as Matplotlib
Best For	Fine-tuned, publication-quality plots	Quick statistical visualizations

When to Use Which?

- **Use Matplotlib** if they need full control over every element of the plot.
- **Use Seaborn** for quick, elegant statistical plots with minimal code.