

SHADOWFOX DATA SCIENCE INTERNSHIP:

NAME: BOAKYE BISMARCK

EMAIL: boakyebismark813@gmail.com

WHATSAPP: +233 554 771 134

Python Visualization Guide: Matplotlib and Seaborn.

1. Library Overview

Matplotlib

Matplotlib is one of the most widely used plotting libraries in Python. It offers a wide range of plotting options and highly customizable plots. Matplotlib is typically used for creating static, publication-quality figures in a variety of formats.

Key Features:

- Extensive documentation and a large community
- Highly customizable
- Integration with other libraries such as Pandas and NumPy

Seaborn

Seaborn is built on top of Matplotlib and provides a high-level interface for drawing attractive statistical graphics. It comes with several built-in themes and color palettes to make more aesthetically pleasing plots.

Key Features:

- Built on top of Matplotlib
- Simplified syntax for complex visualizations
- Integrated with Pandas for data handling

2. Graph Types

Matplotlib

Line Plot

Description: A line plot is used to display information as a series of data points connected by straight line segments. It's useful for time series data.

Use Case: Tracking stock prices over time.

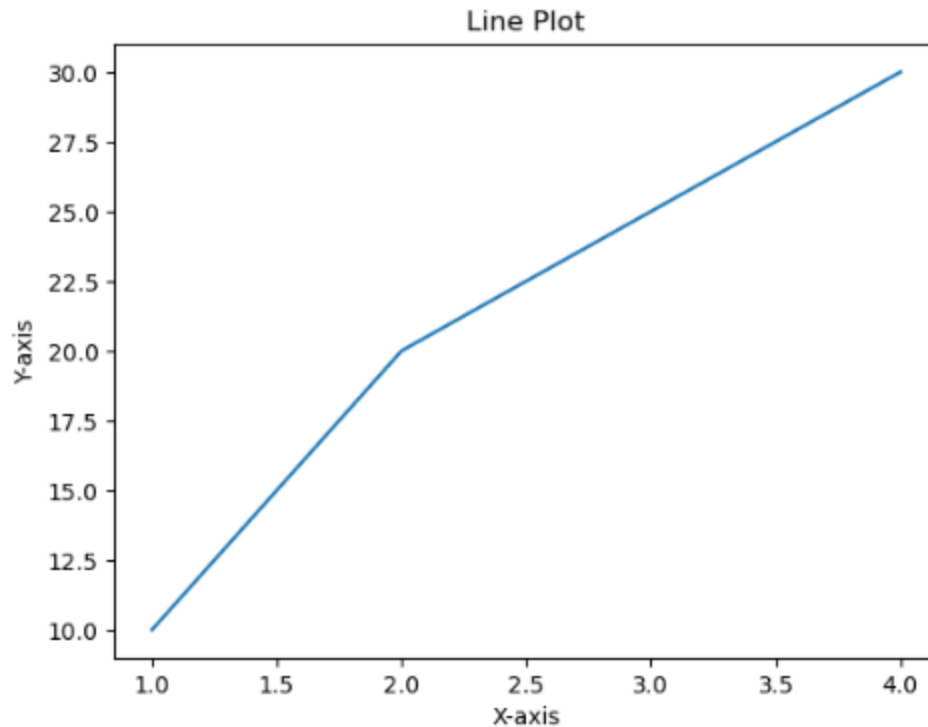
Code Example:

```
[10]: import matplotlib.pyplot as plt
```

```
[11]: X = [1 , 2 , 3, 4]
```

```
[12]: Y = [10 ,20 , 25 , 30 ]
```

```
[14]: plt.plot(X , Y)
      plt.xlabel('X-axis')
      plt.ylabel('Y-axis')
      plt.title('Line Plot')
      plt.show()
```



Scatter Plot

Description: A scatter plot displays data points as a collection of points. It's used to observe relationships between variables.

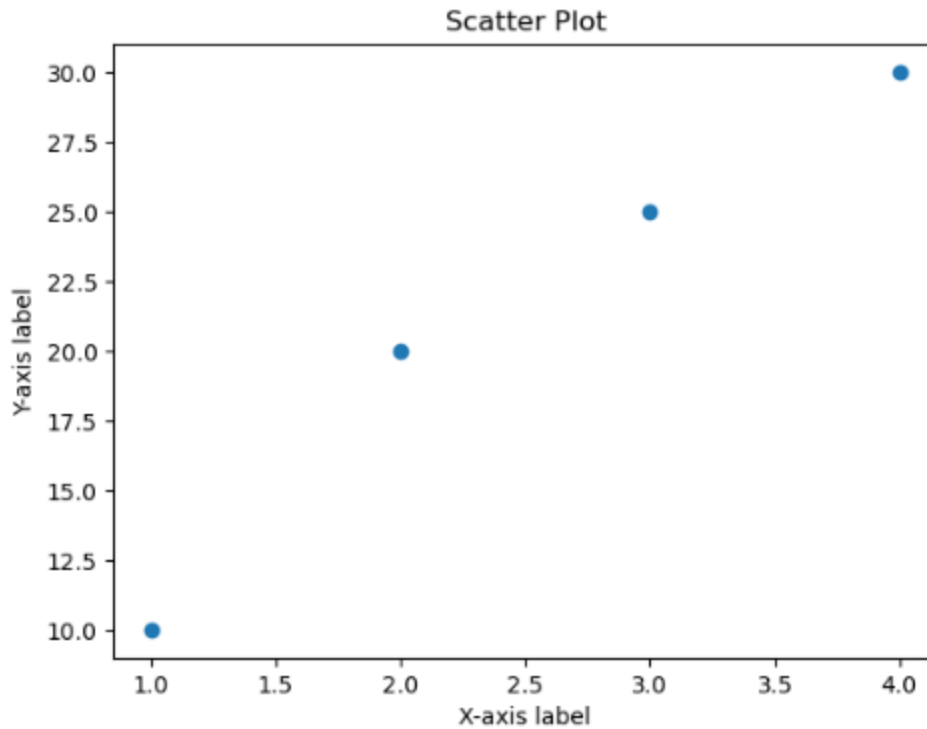
Use Case: Examining the relationship between height and weight.

Code Example:

```
[28]: x = [1, 2, 3, 4]
```

```
[29]: y = [10, 20, 25, 30]
```

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



Bar Chart:

Description: A bar chart presents categorical data with rectangular bars. It's useful for comparing different categories.

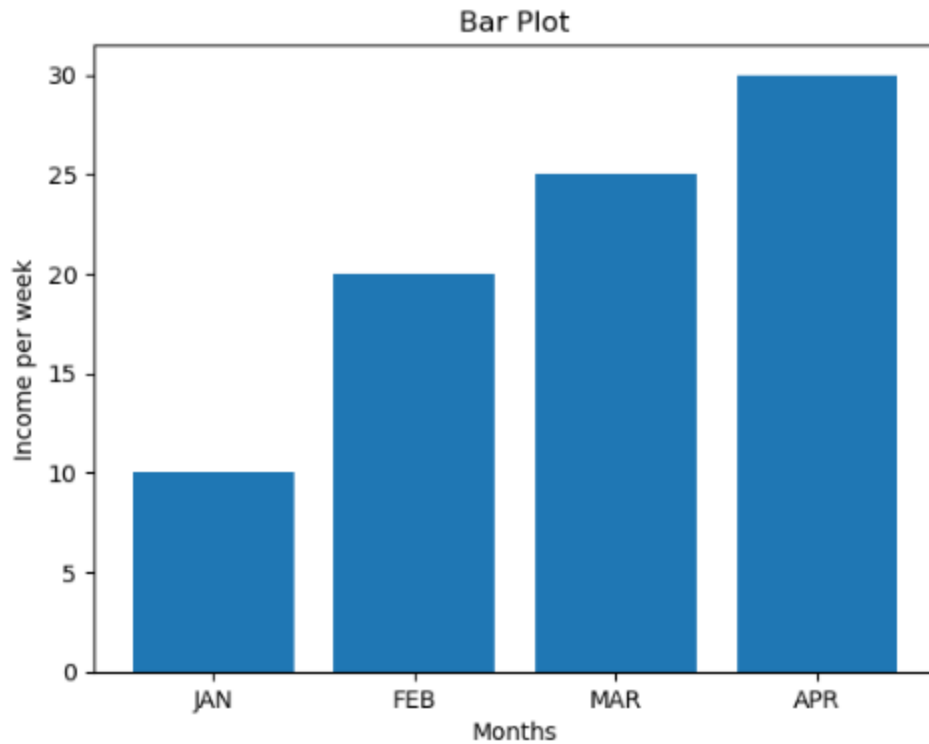
Use Case: Comparing sales across different regions.

Code Example:

```
[15]: MONTHS = ['JAN', 'FEB', 'MAR', 'APR']
```

```
[17]: Income = [10, 20, 25, 30]
```

```
[18]: plt.bar(MONTHS, Income)
plt.xlabel('Months')
plt.ylabel('Income per week')
plt.title('Bar Plot')
plt.show()
```



Histogram:

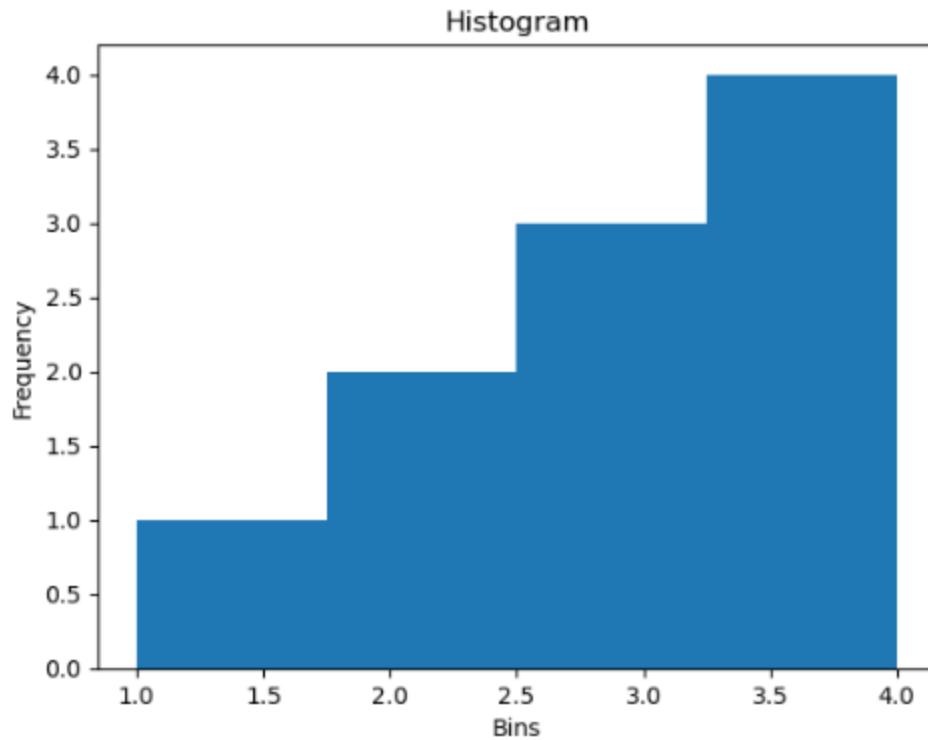
Description: A histogram shows the distribution of a dataset. It's useful for understanding the distribution of continuous data.

Use Case: Analyzing the distribution of exam scores.

Code Example:

```
[22]: data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
```

```
[26]: plt.hist(data, bins=4)
plt.xlabel('Bins')
plt.ylabel('Frequency')
plt.title('Histogram')
plt.show()
```



Seaborn

Line Plot

Description: Similar to Matplotlib, but with additional features and a simpler syntax.

Use Case: Tracking stock prices over time.

Code Example:

```
[48]: import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

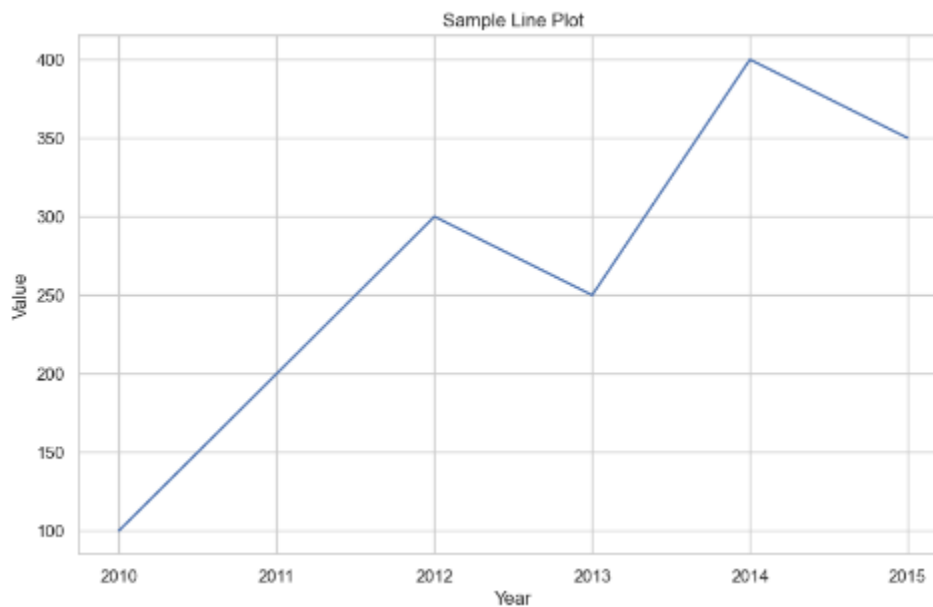
data = {
    'Year':[2010 , 2011 , 2012 ,2013 ,2014 ,2015] ,
    'Value':[100 , 200 , 300 , 250 , 400 , 350]
}
```

```
[50]: df = pd.DataFrame(data)
```

```
[52]: sns.set(style="whitegrid")
plt.figure(figsize=(10 , 6))
line_plot = sns.lineplot(x = 'Year' , y = 'Value' , data = df)

line_plot.set_title(" Sample Line Plot ")
line_plot.set_xlabel('Year')
line_plot.set_ylabel('Value')

plt.show()
```



Scatter Plot

Description: Seaborn scatter plots offer additional functionalities like hue, style, and size differentiation.

Use Case: Examining the relationship between height and weight with gender differentiation.

Code Example:

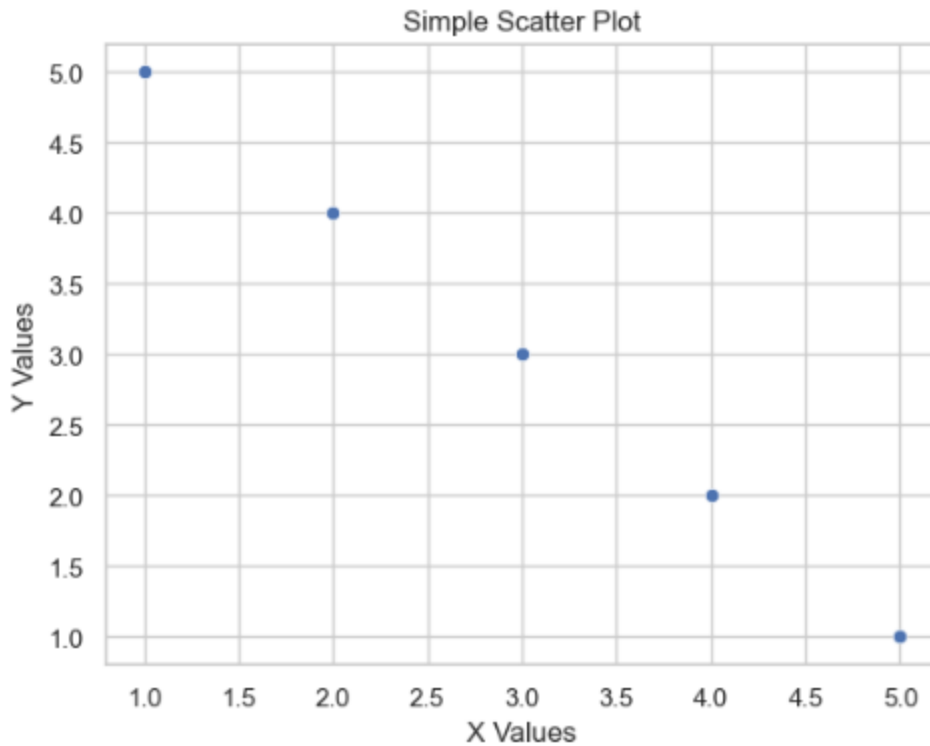
```
[59]: # Import necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Create a simple dataset
x = [1, 2, 3, 4, 5]
y = [5, 4, 3, 2, 1]

# Create a scatter plot
sns.scatterplot(x=x, y=y)

# Add titles and labels
plt.title('Simple Scatter Plot')
plt.xlabel('X Values')
plt.ylabel('Y Values')

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

Bar Chart

Description: Seaborn bar charts offer enhanced aesthetics and functionality.

Use Case: Comparing sales across different regions.

Code Example:

```
[54]: import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Sample data
data = {
    'Category': ['A', 'B', 'C', 'D', 'E'],
    'Value': [23, 45, 56, 78, 213]
}

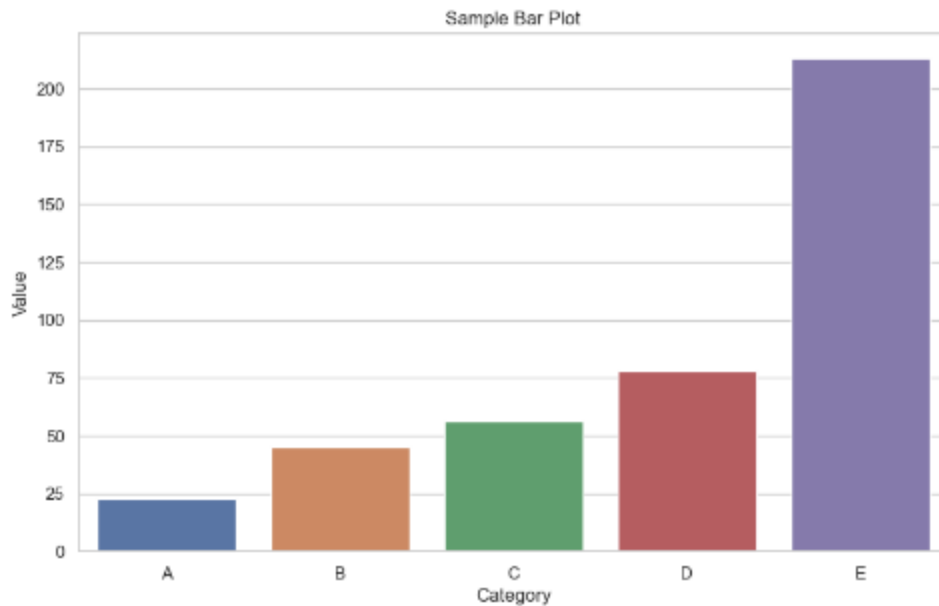
# Create a DataFrame
df = pd.DataFrame(data)

# Initialize the Seaborn style
sns.set(style="whitegrid")

# Create the bar plot
plt.figure(figsize=(10, 6))
bar_plot = sns.barplot(x='Category', y='Value', data=df)

# Customize the plot
bar_plot.set_title('Sample Bar Plot')
bar_plot.set_xlabel('Category')
bar_plot.set_ylabel('Value')

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



Histogram

Description: Seaborn histograms provide additional customization options.

Use Case: Analyzing the distribution of exam scores.

Code Example:

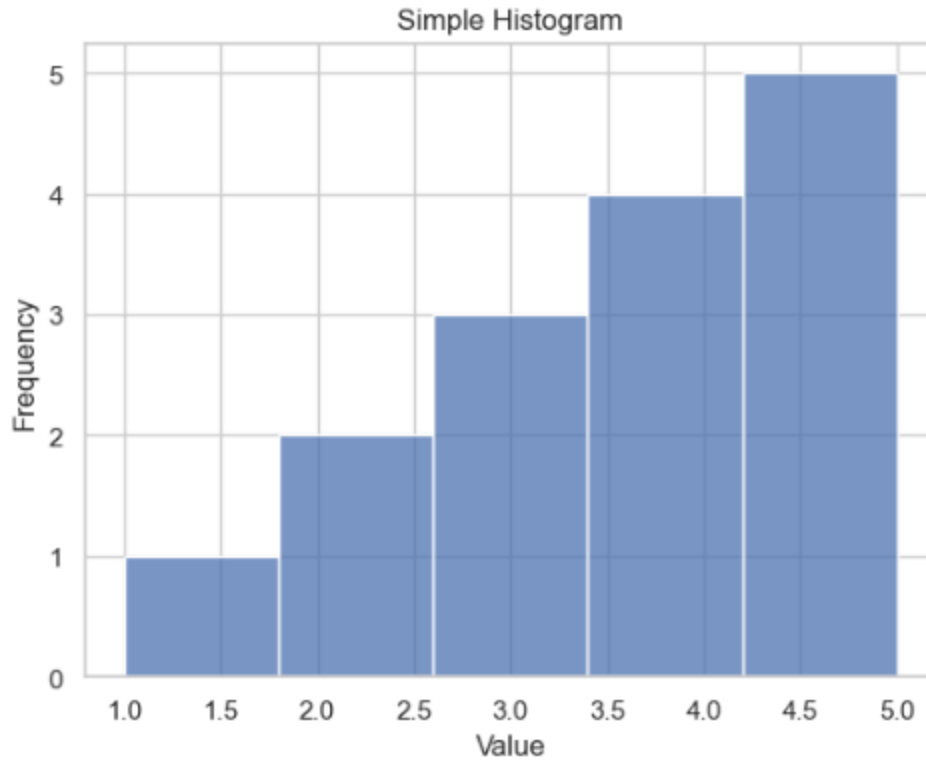
```
[56]: # Import necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Create a simple dataset
data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5]

# Create a histogram
sns.histplot(data, bins=5, kde=False)

# Add titles and labels
plt.title('Simple Histogram')
plt.xlabel('Value')
plt.ylabel('Frequency')

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



3. Comparison

Matplotlib

Strengths:

- Highly customizable
- Extensive documentation and examples
- Integration with other libraries (Pandas, NumPy)

Weaknesses:

- More complex syntax for advanced plots
- Less aesthetically pleasing by default compared to Seaborn

Seaborn

Strengths:

- Built-in themes and color palettes for better aesthetics
- Simplified syntax for complex visualizations
- Excellent integration with Pandas

Weaknesses:

- Limited customization compared to Matplotlib
- Dependent on Matplotlib for underlying functionality

Summary

- **Ease of Use:** Seaborn is generally easier to use for creating attractive and complex visualizations with less code.
- **Customization Options:** Matplotlib offers more customization options, making it suitable for detailed and highly specific visualizations.
- **Interactivity:** Neither library is designed for high interactivity; for interactive plots, Plotly or Bokeh would be more suitable.
- **Performance with Large Datasets:** Both libraries handle large datasets well, but performance might vary depending on the complexity of the plots.