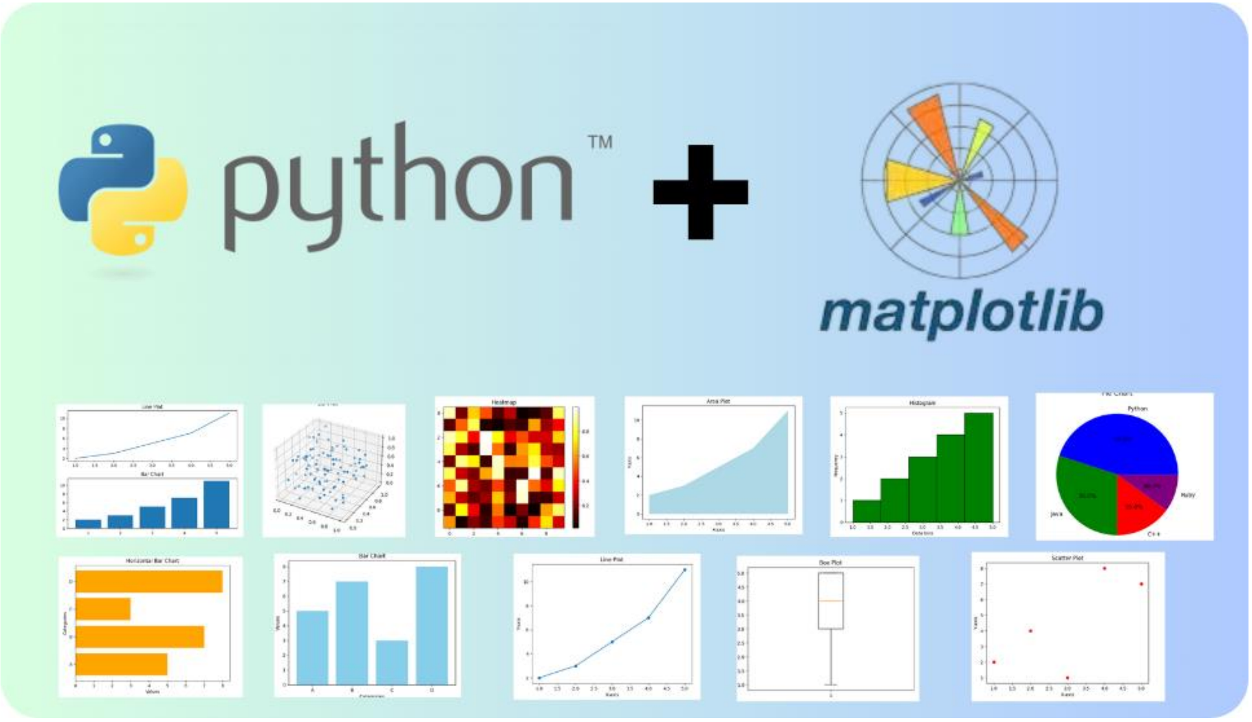

Matplotlib is a powerful and widely-used Python library for creating visualizations that help in data analysis.

1. Line Plot

Used for showing trends or continuous data.

```
import matplotlib.pyplot as plt

# Sample data: x-axis and y-axis values
x = [1, 2, 3, 4, 5]
y = [10, 15, 13, 17, 20]

# Plotting a line graph
plt.plot(x, y)

# Adding title and labels
plt.title("Line Plot")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
```

```
# Adding a grid for better readability
plt.grid(True)

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

---

2. Bar Chart

Used for comparing quantities across categories.

```
# Categories (x-axis) and their corresponding values (y-axis)
categories = ['A', 'B', 'C', 'D']
values = [23, 45, 56, 78]

# Creating a vertical bar chart
plt.bar(categories, values, color='skyblue')

# Adding title and axis labels
plt.title("Bar Chart")
plt.xlabel("Category")
plt.ylabel("Value")

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

---

3. Histogram

Used to understand the distribution of data.

```
import numpy as np

# Generate 1000 random values from a normal distribution
data = np.random.randn(1000)
```

```
# Plot a histogram with 30 bins
plt.hist(data, bins=30, color='orange', edgecolor='black')

# Add title and labels
plt.title("Histogram")
plt.xlabel("Value")
plt.ylabel("Frequency")

# Show plot
plt.show()
```

---

4. Scatter Plot

Shows the relationship between two numerical variables.

```
# Generate 50 random values
x = np.random.rand(50)

# Add noise to simulate a realistic relationship
y = x + np.random.normal(0, 0.1, 50)

# Create scatter plot
plt.scatter(x, y, color='green')

# Add title and labels
plt.title("Scatter Plot")
plt.xlabel("X Value")
plt.ylabel("Y Value")

# Display plot
```

```
plt.show()
```

---

5. Box Plot

Useful for detecting outliers and understanding spread.

```
# Create two sets of 100 random data points
data = [np.random.randn(100), np.random.randn(100)]

# Create boxplot
plt.boxplot(data)

# Set title and category names for x-axis
plt.title("Box Plot")
plt.xticks([1, 2], ['Group 1', 'Group 2'])

# Show plot
plt.show()
```

---

6. Pie Chart

Visualizes proportions among categories.

```
# Define category labels and their sizes (proportions)
labels = ['A', 'B', 'C', 'D']
sizes = [25, 35, 20, 20]

# Create pie chart
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)

# Add title and make the chart circular
plt.title("Pie Chart")
plt.axis('equal')  # Equal aspect ratio ensures circle shape
```

```
# Display pie chart
plt.show()
```

---

7. Heatmap (with Seaborn)

Shows correlations or density over a grid.

```
import seaborn as sns
import pandas as pd

# Create a 5x5 DataFrame with random values
df = pd.DataFrame(np.random.rand(5, 5), columns=list('ABCDE'))

# Create a heatmap from the DataFrame
sns.heatmap(df, annot=True, cmap="YlGnBu")

# Title for the plot
plt.title("Heatmap")

# Show heatmap
plt.show()
```
