# Python Visualization Libraries Guide: Matplotlib and Seaborn

## 1. Library Overview

### Matplotlib

Matplotlib is one of the most widely used 2D plotting libraries in Python. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits.

**Key Features:**

* Fine-grained control over plot elements
* Supports a wide range of plot types
* Suitable for static and publication-quality visualizations

**Use Cases:**

* Data exploration and analysis
* Creating plots for academic publications
* Customizing plot appearance extensively

### Seaborn

Seaborn is built on top of Matplotlib and provides a high-level interface for drawing attractive and informative statistical graphics.

**Key Features:**

* Built-in themes for styling Matplotlib graphics
* Simplifies complex visualizations with less code
* Built-in support for Pandas DataFrames

**Use Cases:**

* Statistical data visualization
* Heatmaps, categorical plots, and complex relational plots
* Quick exploratory data analysis

## 2. Graph Types

### Matplotlib

#### a. Line Plot

**Description:** Represents data as a series of points connected by straight lines. **Use Case:** Time series, trend analysis.

import matplotlib.pyplot as plt  
x = [1, 2, 3, 4]  
y = [10, 20, 25, 30]  
plt.plot(x, y)  
plt.title('Line Plot')  
plt.xlabel('X-axis')  
plt.ylabel('Y-axis')  
plt.show()

#### b. Bar Chart

**Description:** Displays categorical data with rectangular bars. **Use Case:** Comparing quantities across categories.

categories = ['A', 'B', 'C']  
values = [10, 15, 7]  
plt.bar(categories, values)  
plt.title('Bar Chart')  
plt.show()

#### c. Histogram

**Description:** Shows frequency distribution of numeric data. **Use Case:** Analyzing data distribution.

data = [1, 2, 2, 3, 3, 3, 4, 4, 5]  
plt.hist(data, bins=5)  
plt.title('Histogram')  
plt.show()

#### d. Scatter Plot

**Description:** Displays values for two variables using Cartesian coordinates. **Use Case:** Identifying relationships between variables.

x = [1, 2, 3, 4, 5]  
y = [5, 4, 3, 2, 1]  
plt.scatter(x, y)  
plt.title('Scatter Plot')  
plt.show()

#### e. Pie Chart

**Description:** Represents proportions of a whole as slices. **Use Case:** Showing percentage breakdown.

labels = ['A', 'B', 'C']  
sizes = [40, 35, 25]  
plt.pie(sizes, labels=labels, autopct='%1.1f%%')  
plt.title('Pie Chart')  
plt.show()

### Seaborn

#### a. Line Plot

**Description:** Enhanced line plots using built-in support for data frames. **Use Case:** Time-series with confidence intervals.

import seaborn as sns  
import pandas as pd  
  
data = pd.DataFrame({"x": [1, 2, 3, 4], "y": [1, 4, 9, 16]})  
sns.lineplot(x="x", y="y", data=data)

#### b. Bar Plot

**Description:** Statistical summary of data with error bars. **Use Case:** Category comparison with confidence intervals.

sns.barplot(x=["A", "B", "C"], y=[10, 20, 15])

#### c. Histogram (displot)

**Description:** Univariate distribution of observations. **Use Case:** Viewing distribution shape.

data = [1, 2, 2, 3, 4, 5, 6]  
sns.displot(data, bins=5)

#### d. Scatter Plot

**Description:** Relationship between two variables with styling options. **Use Case:** Visualizing correlations.

df = pd.DataFrame({'x': [1,2,3,4], 'y':[4,3,2,1]})  
sns.scatterplot(x='x', y='y', data=df)

#### e. Box Plot

**Description:** Displays distribution and outliers. **Use Case:** Comparing groups.

sns.boxplot(data=[1, 2, 5, 7, 8, 8, 9])

#### f. Heatmap

**Description:** Matrix of values with color encoding. **Use Case:** Correlation matrix.

import numpy as np  
corr = np.corrcoef([[1, 2, 3], [4, 5, 6]])  
sns.heatmap(corr, annot=True)

## 3. Comparison: Matplotlib vs Seaborn

| Feature | Matplotlib | Seaborn |
| --- | --- | --- |
| Ease of Use | Requires more code | Higher-level API, easier syntax |
| Customization | Highly customizable | Limited compared to Matplotlib |
| Interactivity | Basic (can be improved with toolkits) | Limited, mostly static |
| Data Handling | Manual (works with lists, arrays) | Works seamlessly with Pandas |
| Built-in Themes | Few | Several appealing themes |
| Performance | Good for small to large datasets | Good for medium datasets |