

# Python Visualization Libraries Guide: Matplotlib & Seaborn

## Library Overview

### Matplotlib

Matplotlib is a foundational Python library for data visualization. It provides extensive control over plots and is well-suited for static, 2D plots in Python.

- **Unique Features:**
  - Highly customizable.
  - Works seamlessly with NumPy and Pandas.
  - Used as a base for other libraries like Seaborn.
- **Typical Use Cases:**
  - Simple to complex 2D plots.
  - Academic and scientific visualization.

### Seaborn

Seaborn is built on top of Matplotlib and is designed to create attractive and informative statistical graphics.

- **Unique Features:**
  - Simplified syntax for complex plots.
  - Built-in themes and color palettes.
  - Deep integration with Pandas data frames.
- **Typical Use Cases:**
  - Statistical visualizations.
  - Exploratory data analysis.

## Graph Types

### Matplotlib Graphs

#### 1. Line Plot

**Description:** Shows data points connected by straight lines, ideal for trends over time.

**Use Case:** Stock prices over time, temperature changes.

```
1  import matplotlib.pyplot as plt # type: ignore
2
3  x = [1, 2, 3, 4]
4  y = [10, 20, 25, 30]
5
6  plt.plot(x, y)
7  plt.title('Simple Line Plot')
8  plt.xlabel('X-axis')
9  plt.ylabel('Y-axis')
10 plt.show()
```

## 2. Scatter Plot

**Description:** Displays individual data points as dots, showing relationship between variables.

**Use Case:** Correlation between height and weight.

```
1  import matplotlib.pyplot as plt # type: ignore
2
3  x = [5, 7, 8, 7, 2, 17]
4  y = [99, 86, 87, 88, 100, 86]
5
6  plt.scatter(x, y)
7  plt.title('Scatter Plot Example')
8  plt.xlabel('X values')
9  plt.ylabel('Y values')
10 plt.show()
```

## 3. Bar Chart

**Description:** Uses rectangular bars to represent categorical data.

**Use Case:** Sales by product category.

```
1  import matplotlib.pyplot as plt
2
3  categories = ['A', 'B', 'C']
4  values = [15, 10, 20]
5
6  plt.bar(categories, values)
7  plt.title('Bar Chart Example')
8  plt.xlabel('Categories')
9  plt.ylabel('Values')
10 plt.show()
```

#### 4. Histogram

**Description:** Shows distribution of numerical data using bins.

**Use Case:** Age distribution in a population.

```
1  import matplotlib.pyplot as plt
2  import numpy as np
3
4  data = np.random.normal(170, 10, 250)
5
6  plt.hist(data, bins=20)
7  plt.title('Histogram Example')
8  plt.xlabel('Height')
9  plt.ylabel('Frequency')
10 plt.show()
```

#### 5. Pie Chart

**Description:** Circular chart divided into slices to show proportions.

**Use Case:** Market share distribution.

```
1 import matplotlib.pyplot as plt
2
3 labels = ['A', 'B', 'C', 'D']
4 sizes = [15, 30, 45, 10]
5
6 plt.pie(sizes, labels=labels, autopct='%1.1f%%')
7 plt.title('Pie Chart Example')
8 plt.show()
```

## Seaborn Graphs

### 1. Distribution Plot (Distplot)

**Description:** Combines histogram with kernel density estimate.

**Use Case:** Visualizing distribution of a dataset.

```
1 import seaborn as sns
2 import numpy as np
3
4 data = np.random.normal(size=1000)
5 sns.displot(data, kde=True)
6 plt.title('Distribution Plot')
7 plt.show()
```

### 2. Box Plot

**Description:** Shows distribution through quartiles with outliers.

**Use Case:** Comparing distributions across categories.

```
1  import seaborn as sns
2  import pandas as pd
3
4  data = pd.DataFrame({
5      'Category': ['A', 'A', 'B', 'B', 'C', 'C'],
6      'Values': [10, 15, 12, 18, 20, 25]
7  })
8
9  sns.boxplot(x='Category', y='Values', data=data)
10 plt.title('Box Plot Example')
11 plt.show()
```

### 3. Violin Plot

**Description:** Combines box plot with kernel density estimation.

**Use Case:** Detailed distribution comparison.

```
1  import seaborn as sns
2  import pandas as pd
3
4  data = pd.DataFrame({
5      'Category': ['A', 'A', 'B', 'B', 'C', 'C'],
6      'Values': [10, 15, 12, 18, 20, 25]
7  })
8
9  sns.violinplot(x='Category', y='Values', data=data)
10 plt.title('Violin Plot Example')
11 plt.show()
```

### 4. Heatmap

**Description:** Matrix representation of data with color coding.

**Use Case:** Correlation matrices, confusion matrices.

```
1 import seaborn as sns
2 import numpy as np
3
4 data = np.random.rand(5, 5)
5 sns.heatmap(data, annot=True)
6 plt.title('Heatmap Example')
7 plt.show()
```

## 5. Pair Plot

**Description:** Matrix of scatter plots for multiple variables.

**Use Case:** Exploring relationships in multivariate data.

```
1 import seaborn as sns
2 iris = sns.load_dataset('iris')
3
4 sns.pairplot(iris, hue='species')
5 plt.title('Pair Plot Example')
6 plt.show()
```

## Comparison

Feature	Matplotlib	Seaborn
Ease of Use	Steeper learning curve	Easier for common plots
Customization	Highly customizable	Limited but good defaults
Interactivity	Basic (with add-ons)	Basic (inherited from mpl)
Performance	Good with large datasets	Good with large datasets
Best For	Precise control, publication	Quick EDA, statistical viz
Style	Basic defaults	Attractive defaults

Feature	Matplotlib	Seaborn
Data Format	Arrays, lists	Pandas Data Frames preferred