# Visualization Library Documentation

**Objective:** Create a comprehensive documentation guide for Matplotlib and Pandas visualization libraries.

## Matplotlib Overview

Matplotlib is a widely used 2D plotting library in Python. It provides fine-grained control over every element of a plot and is excellent for static, publication-quality visualizations. Typical use cases include data exploration, scientific visualization, and academic reports.

### *Matplotlib Graph Types*

#### *Line Plot*

Used for showing trends over time or continuous data.

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4],[1,4,9,16])
plt.xlabel('x')
plt.ylabel('y')
plt.title('Line Plot')
plt.show()
```

#### *Scatter Plot*

Displays relationships between two numerical variables.

```
plt.scatter([5,7,8,7],[99,86,87,88])
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Scatter Plot')
plt.show()
```

#### *Bar Chart*

Compares quantities across categories.

```
plt.bar(['A','B','C'], [10,20,15])
plt.title('Bar Chart')
plt.show()
```

#### *Histogram*

Shows distribution of numerical data.

```
import numpy as np
x = np.random.randn(1000)
plt.hist(x, bins=30)
plt.title('Histogram')
plt.show()
```

#### *Pie Chart*

Displays proportions of categories.

```
sizes = [15, 30, 45, 10]
labels = ['A','B','C','D']
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.title('Pie Chart')
plt.show()
```

## Pandas Visualization Overview

Pandas is primarily a data manipulation library, but it comes with built-in visualization capabilities based on Matplotlib. It allows users to quickly generate plots directly from DataFrames and Series. Typical use cases include exploratory data analysis (EDA) and quick visualization of tabular data.

## *Pandas Graph Types*

### *Line Plot*

Quickly plot time series or continuous data from DataFrames.

```
import pandas as pd
s = pd.Series([1,3,2,4])
s.plot(kind='line', title='Pandas Line Plot')
```

### *Bar Plot*

Compare values across categories directly from DataFrame.

```
df = pd.DataFrame({'A':[3,2,1],'B':[1,2,3]})
df.plot(kind='bar', title='Pandas Bar Plot')
```

### *Histogram*

Plot distribution of numerical data.

```
df['A'].plot(kind='hist', bins=5, title='Pandas Histogram')
```

### *Box Plot*

Shows spread and outliers in data.

```
df.plot(kind='box', title='Pandas Box Plot')
```

### *Area Plot*

Visualize cumulative values over time or categories.

```
df.plot(kind='area', alpha=0.5, title='Pandas Area Plot')
```

# Comparison of Matplotlib and Pandas

- **Ease of Use:** Pandas is easier for quick plots directly from DataFrames, while Matplotlib requires more setup but offers more control.
- **Customization:** Matplotlib provides full customization; Pandas is limited but sufficient for exploratory analysis.
- **Interactivity:** Both are static by default. For interactivity, external libraries (Plotly, Bokeh) are preferred.
- **Performance:** Both handle medium-sized datasets well, but for very large data, specialized libraries may be faster.

# Resources

- Matplotlib Quick Start: https://matplotlib.org/stable/users/explain/quick_start.html#quick-start
- Pandas User Guide: https://pandas.pydata.org/docs/user_guide/index.html