

Task Level (Beginner):

Python Visualization Libraries Documentation Guide

Selected Libraries: Matplotlib and Plotly

1. Library Overview

Matplotlib

Matplotlib is one of the most widely used Python libraries for 2D plotting. It provides fine-grained control over every aspect of a figure and is ideal for producing static plots for publications.

Features:

- Customizable plots with axes, labels, legends, etc.
 - Integration with NumPy and Pandas
 - Suitable for static and publication-quality visualizations
- Typical Use Cases:**
- Exploratory data analysis
 - Generating visualizations for reports or research
 - Creating static, high-quality images for print

Plotly

Plotly is a high-level, interactive graphing library. It is especially useful for creating dashboards and webbased visualizations.

Features:

- Interactivity: Zoom, hover, and filter in real time
- Built-in support for web deployment
- Integration with Dash for dashboard creation

Typical Use Cases:

- Interactive dashboards

- Visual data storytelling
- Real-time monitoring and analytics

2. Graph Types

Matplotlib Graph Types

Line Plot

Description: Displays data as a series of points connected by lines. **Use Case:** Time series, trends over time.

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [10, 20, 25, 30, 40]
plt.plot(x, y)
plt.title("Line Plot")
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

Bar Chart

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

```
categories = ['A', 'B', 'C']
values = [5, 7, 3]
plt.bar(categories, values)
plt.title("Bar Chart")
plt.show()
```

Scatter Plot

Description: Shows relationship between two numerical variables. **Use Case:** Correlation, clustering.

```
import numpy as np
x = np.random.rand(50)
y = np.random.rand(50)
plt.scatter(x, y)
plt.title("Scatter Plot")
plt.show()
```

Histogram

Description: Shows frequency distribution of a dataset. **Use Case:** Distribution analysis.

```
data = np.random.randn(1000)
plt.hist(data, bins=30)
plt.title("Histogram")
plt.show()
```

Plotly Graph Types

Line Plot

Description: Interactive line chart with tooltips. **Use Case:** Time-series data with user interaction.

```
import plotly.express as px
import pandas as pd
df = pd.DataFrame({"x": [1, 2, 3, 4, 5], "y": [10, 20, 25, 30, 40]})
fig = px.line(df, x="x", y="y", title="Line Plot")
fig.show()
```

Bar Chart

Description: Interactive categorical comparison. **Use Case:** Web-based dashboards.

```
df = pd.DataFrame({"Category": ["A", "B", "C"], "Values": [5, 7, 3]})
fig = px.bar(df, x="Category", y="Values", title="Bar Chart")
fig.show()
```

Scatter Plot

Description: Interactive scatter plot with hover info. **Use Case:** Exploring correlations.

```
import numpy as np
import pandas as pd
x = np.random.rand(50)
y = np.random.rand(50)
df = pd.DataFrame({'x': x, 'y': y})
fig = px.scatter(df, x="x", y="y", title="Scatter Plot")
fig.show()
```

Histogram

Description: Interactive frequency distribution. **Use Case:** Understand data spread.

```
data = np.random.randn(1000)
df = pd.DataFrame({'data': data})
fig = px.histogram(df, x='data', nbins=30, title="Histogram")
fig.show()
```

3. Comparison

Feature	Matplotlib	Plotly
Ease of Use	Moderate (steeper learning curve)	Easy (higher-level abstraction)
Customization	High (manual configuration)	Moderate (pre-configured styles)
Interactivity	None (static images)	High (zoom, pan, tooltips)
Performance	Good for small to medium datasets	Better for interactive dashboards
Ideal Use Cases	Research, Reports, Publications	Web dashboards, Data Exploration