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

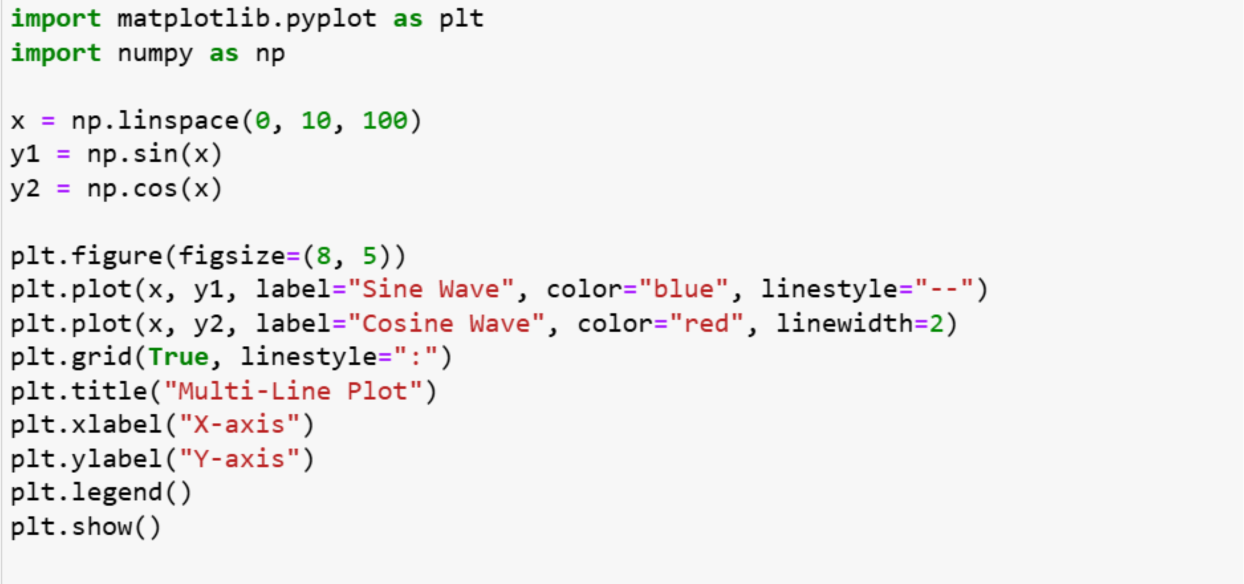
- Visual data storytelling
- Real-time monitoring and analytics

2. Graph Types

Matplotlib Graph Types

Multi-Line Plot with Grid

Description: Plots sine and cosine waves together with grid lines for easy comparison.



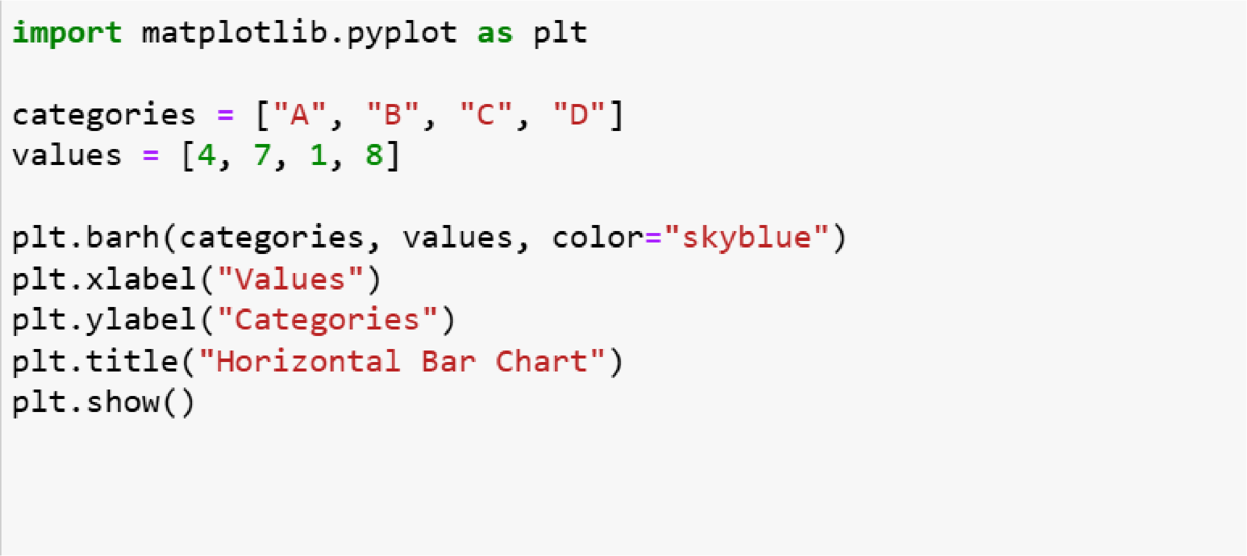
```python
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)

plt.figure(figsize=(8, 5))
plt.plot(x, y1, label="Sine Wave", color="blue", linestyle="--")
plt.plot(x, y2, label="Cosine Wave", color="red", linewidth=2)
plt.grid(True, linestyle=":")
plt.title("Multi-Line Plot")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.legend()
plt.show()
```

Horizontal Bar Chart

Description: Displays categorical values as horizontal bars for better label readability.



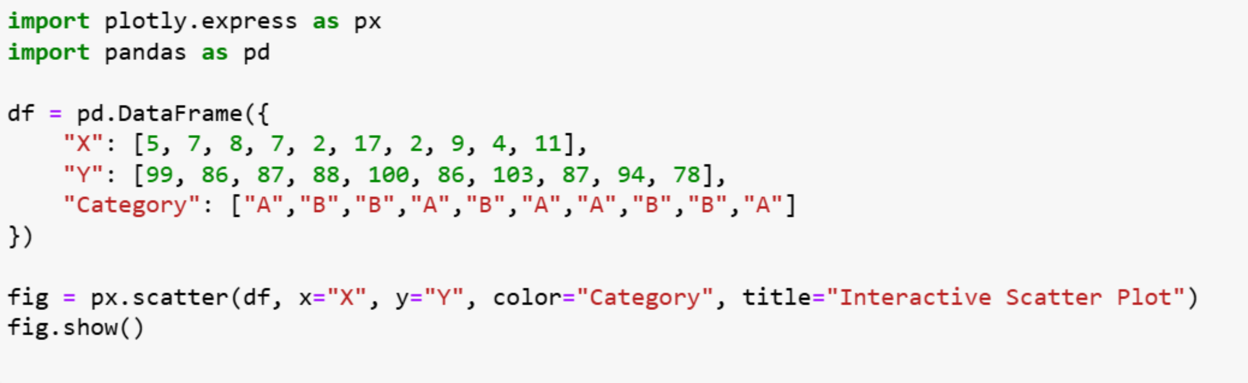
```python
import matplotlib.pyplot as plt

categories = ["A", "B", "C", "D"]
values = [4, 7, 1, 8]

plt.barh(categories, values, color="skyblue")
plt.xlabel("Values")
plt.ylabel("Categories")
plt.title("Horizontal Bar Chart")
plt.show()
```

Interactive Scatter Plot

Description: Shows data points with hover info and color coding by category.



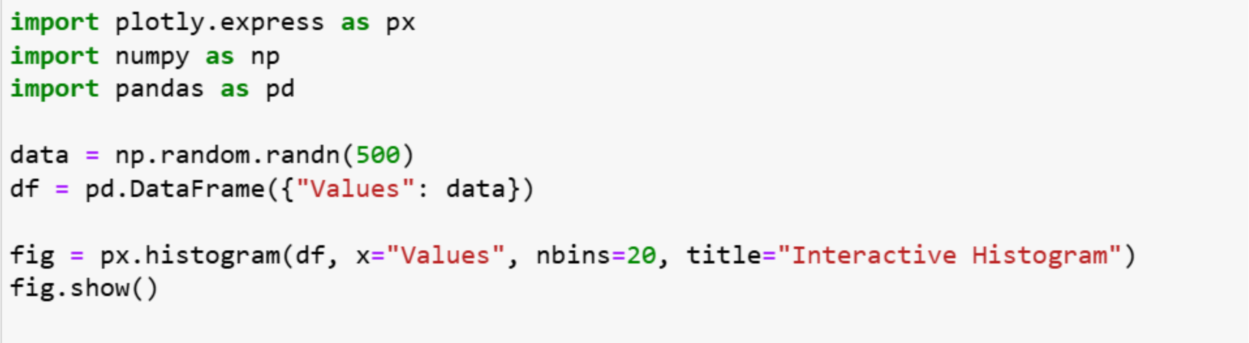
```
import plotly.express as px
import pandas as pd

df = pd.DataFrame({
    "X": [5, 7, 8, 7, 2, 17, 2, 9, 4, 11],
    "Y": [99, 86, 87, 88, 100, 86, 103, 87, 94, 78],
    "Category": ["A","B","B","A","B","A","A","B","B","A"]
})

fig = px.scatter(df, x="X", y="Y", color="Category", title="Interactive Scatter Plot")
fig.show()
```

Interactive Histogram

Description: Visualizes frequency distribution with adjustable bins interactively.



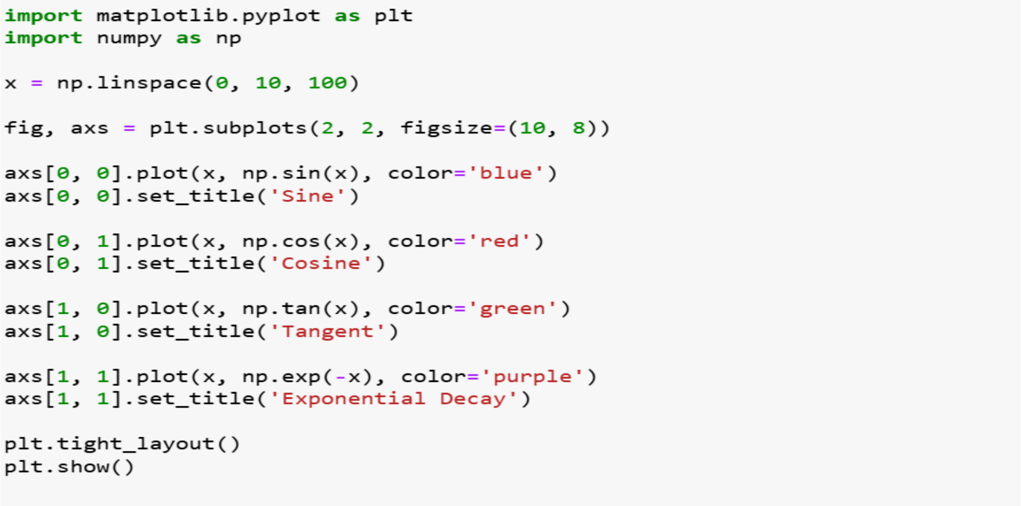
```
import plotly.express as px
import numpy as np
import pandas as pd

data = np.random.randn(500)
df = pd.DataFrame({"Values": data})

fig = px.histogram(df, x="Values", nbins=20, title="Interactive Histogram")
fig.show()
```

Multiple Subplots

Description: Visualizes frequency distribution with adjustable bins interactively.



```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 10, 100)

fig, axs = plt.subplots(2, 2, figsize=(10, 8))

axs[0, 0].plot(x, np.sin(x), color='blue')
axs[0, 0].set_title('Sine')

axs[0, 1].plot(x, np.cos(x), color='red')
axs[0, 1].set_title('Cosine')

axs[1, 0].plot(x, np.tan(x), color='green')
axs[1, 0].set_title('Tangent')

axs[1, 1].plot(x, np.exp(-x), color='purple')
axs[1, 1].set_title('Exponential Decay')

plt.tight_layout()
plt.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