

Principles of Information Visualization and Matplotlib

1. Principles of Information Visualization

Information visualization is the graphical representation of data to make patterns, trends, and insights easier to understand.

Key Principles: 1. **Accuracy:** Represent data truthfully. 2. **Clarity:** Avoid clutter and make interpretation easy. 3. **Efficiency:** Highlight patterns quickly. 4. **Interactivity:** Allow exploration of data (optional). 5. **Aesthetic:** Use colors, size, and layout effectively.

2. Matplotlib in Python

Matplotlib is a powerful Python library for creating static, animated, and interactive visualizations.

2.1 Matplotlib Architecture

Layers: 1. **Backend Layer:** Handles rendering and output (PNG, PDF, GUI like Tkinter). 2. **Artist Layer:** Everything drawn on the canvas (lines, text, shapes) is an "Artist". 3. **Scripting Layer (pyplot API):** Provides functions like `plot()`, `scatter()` for easy plotting.

Conceptual Flow:

```
User Code → Pyplot API → Artist Layer → Backend → Output (File/GUI)
```

3. Basic Plotting with Matplotlib

3.1 Importing Matplotlib

```
import matplotlib.pyplot as plt
```

3.2 Line Plot

```
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
```

```
plt.plot(x, y, color='blue', marker='o', linestyle='--', label='Line Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Simple Line Plot')
plt.legend()
plt.show()
```

4. Common Plots

4.1 Scatter Plot

```
x = [5, 7, 8, 7, 2]
y = [99, 86, 87, 88, 100]

plt.scatter(x, y, color='red')
plt.title('Scatter Plot Example')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```

4.2 Bar Chart

```
categories = ['A', 'B', 'C', 'D']
values = [10, 24, 36, 40]

plt.bar(categories, values, color='green')
plt.title('Bar Chart Example')
plt.show()
```

5. Charting Fundamentals

5.1 Subplots

```
plt.subplot(1, 2, 1)
plt.plot(x, y, 'r-')
plt.title('Plot 1')

plt.subplot(1, 2, 2)
plt.bar(categories, values)
plt.title('Plot 2')
```

```
plt.tight_layout()
plt.show()
```

5.2 Histograms

```
data = [1,2,2,3,3,3,4,4,5]
plt.hist(data, bins=5, color='orange', edgecolor='black')
plt.title('Histogram Example')
plt.show()
```

5.3 Box Plots

```
data = [7, 8, 5, 6, 9, 10, 2, 5]
plt.boxplot(data)
plt.title('Box Plot Example')
plt.show()
```

5.4 Heatmaps

```
import numpy as np
data = np.random.rand(5,5)
plt.imshow(data, cmap='hot', interpolation='nearest')
plt.colorbar()
plt.title('Heatmap Example')
plt.show()
```

5.5 Animation

```
from matplotlib.animation import FuncAnimation

fig, ax = plt.subplots()
x_data, y_data = [], []
line, = plt.plot([], [], 'r-')

def animate(i):
    x_data.append(i)
    y_data.append(i**0.5)
    line.set_data(x_data, y_data)
    ax.relim()
    ax.autoscale_view()
    return line,
```

```
ani = FuncAnimation(fig, animate, frames=20, interval=200)
plt.show()
```

6. Summary Table of Plots

Plot Type	Purpose	Example Use Case
Line Plot	Show trends over time	Stock prices, sales over months
Scatter Plot	Show relationship between variables	Height vs Weight
Bar Chart	Compare categories	Population of countries
Histogram	Frequency distribution	Exam scores, ages
Box Plot	Distribution & outliers	Salary distribution
Heatmap	Visualize 2D data/matrix	Correlation matrices
Animation	Dynamic, changing data visualization	Real-time sensor data