Matplotlib: Complete Guide

Matplotlib is a powerful Python library for creating static, animated, and interactive visualizations. It is widely used in data analysis, machine learning, and scientific computing.

Installation

If you haven't installed Matplotlib yet, you can do so using:

```
pip install matplotlib
```

1. Basic Plot with Matplotlib

```
import matplotlib.pyplot as plt

# Sample data
x = [1, 2, 3, 4, 5]
y = [10, 15, 7, 18, 9]

# Create a basic plot
plt.plot(x, y, marker='o', linestyle='-', color='b', label='Data')

# Labels and title
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Simple Line Plot')

# Show legend
plt.legend()

# Display the plot
plt.show()
```

2. Key Components of Matplotlib

Matplotlib has two key objects:

- **Figure**: The overall container for all plots.
- **Axes**: A specific plotting area inside a figure.

Creating a Figure and Axes

```
fig, ax = plt.subplots()  # Creates a figure with one axes
ax.plot(x, y)  # Plot data on the axes
plt.show()
```

3. Different Types of Plots

Matplotlib supports multiple types of plots:

1. Line Plot

```
plt.plot(x, y, linestyle='--', marker='o', color='r')
plt.show()
```

2. Scatter Plot

```
plt.scatter(x, y, color='g', marker='s')  # 's' for square markers
plt.show()
```

3. Bar Chart

```
plt.bar(x, y, color='c')
plt.show()
```

4. Histogram

```
import numpy as np

data = np.random.randn(1000)  # Generate random data
plt.hist(data, bins=30, color='m', edgecolor='black')
plt.show()
```

5. Pie Chart

```
labels = ['A', 'B', 'C', 'D']
sizes = [30, 20, 35, 15]
colors = ['gold', 'blue', 'red', 'green']

plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Equal aspect ratio ensures the pie chart is circular
plt.show()
```

6. Box Plot (Whisker Plot)

```
data = [np.random.randn(100) for _ in range(4)]
plt.boxplot(data)
plt.show()
```

---

4. Customization in Matplotlib

Matplotlib allows extensive customization.

1. Adding Labels & Titles

```
plt.plot(x, y)
plt.xlabel('X Axis')
plt.ylabel('Y Axis')
plt.title('Customized Plot')
plt.show()
```

2. Changing Line Styles, Colors, and Markers

```
plt.plot(x, y, linestyle='--', marker='o', color='r', linewidth=2, markersize=8)
plt.show()
```

3. Adding Grid

```
plt.plot(x, y)
plt.grid(True)  # Adds grid
plt.show()
```

4. Adding a Legend

```
plt.plot(x, y, label='Line 1', color='blue')
plt.plot(x, [i * 2 for i in y], label='Line 2', color='green')
plt.legend(loc='upper left')
plt.show()
```

---

5. Multiple Plots (Subplots)

1. Multiple Plots in One Figure

```
fig, axs = plt.subplots(2, 2)  # Creates a 2x2 grid of plots

axs[0, 0].plot(x, y, color='r')  # Top-left
axs[0, 1].bar(x, y, color='g')   # Top-right
axs[1, 0].scatter(x, y, color='b') # Bottom-left
axs[1, 1].hist(np.random.randn(100), bins=20, color='m')  # Bottom-right

plt.show()
```

2. Shared X and Y Axes

```
fig, axs = plt.subplots(2, sharex=True, sharey=True)
```

```
axs[0].plot(x, y, color='r')
axs[1].bar(x, y, color='b')

plt.show()
```

---

6. Advanced Features

1. Annotations

```
plt.plot(x, y)
plt.annotate('Peak', xy=(4, 18), xytext=(3, 15),
             arrowprops=dict(facecolor='black', shrink=0.05))
plt.show()
```

2. Setting Axis Limits

```
plt.plot(x, y)
plt.xlim(0, 6)
plt.ylim(0, 20)
plt.show()
```

3. Logarithmic Scale

```
plt.plot(x, y)
plt.yscale('log')  # Change y-axis to log scale
plt.show()
```

---

7. Saving Plots

You can save plots as images using:

```
plt.plot(x, y)
plt.savefig('plot.png', dpi=300, bbox_inches='tight')  # Save with high resolution
```

---

8. Interactive Mode

Matplotlib supports interactive plotting using `%matplotlib inline` in Jupyter Notebook:

```
%matplotlib inline  # For inline plotting in Jupyter Notebook
```

You can also use:

```
plt.ion()  # Turn on interactive mode
plt.ioff() # Turn off interactive mode
```

---

9. Matplotlib with Pandas

Matplotlib integrates well with Pandas DataFrames.

```
import pandas as pd

# Sample DataFrame
data = {'X': [1, 2, 3, 4, 5], 'Y': [10, 15, 7, 18, 9]}
df = pd.DataFrame(data)

df.plot(x='X', y='Y', kind='line', marker='o', color='r')
plt.show()
```

Summary

Feature	Description
Basic Plot	`plt.plot(x, y)`
Scatter Plot	`plt.scatter(x, y)`
Bar Chart	`plt.bar(x, y)`
Histogram	`plt.hist(data, bins=30)`
Pie Chart	`plt.pie(sizes, labels=labels, autopct='%1.1f%%')`
Box Plot	`plt.boxplot(data)`
Grid	`plt.grid(True)`
Legend	`plt.legend(loc='upper left')`
Annotations	`plt.annotate('Text', xy=(x, y))`
Multiple Subplots	`fig, axs = plt.subplots(2, 2)`
Save Plot	`plt.savefig('plot.png', dpi=300)`