Matplotlib Cheat Sheet (Theory Only)

---

1. Introduction to Matplotlib

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

Works with **NumPy & Pandas** for scientific plotting.

Provides support for **2D & 3D plots, histograms, and custom visualization**.

Importing Matplotlib:

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

---

2. Basic Functions of Matplotlib

Creating a Figure and Axis:

```
fig, ax = plt.subplots()  # Creates a figure and axis
ax.plot([1, 2, 3], [4, 5, 6])  # Plots a simple line
plt.show()
```

Adding Labels and Titles:

```
ax.set_title("Sample Plot")
ax.set_xlabel("X-Axis Label")
ax.set_ylabel("Y-Axis Label")
```

Displaying a Plot:

```
plt.show()
```

---

3. Simple Line Plot

Plotting a Line Graph:

```
x = np.linspace(0, 10, 100)
y = np.sin(x)

plt.plot(x, y, label="Sine Wave", color="blue", linestyle="--")
plt.legend()  # Show legend
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.title("Simple Line Plot")
plt.show()
```

Customizing Line Styles:

- **Dashed:** linestyle="--"
- **Dotted:** linestyle=":"
- **Dash-dot:** linestyle="-."
- **Solid:** linestyle="-"

---

4. Scatter Plot

Creating a Scatter Plot:

```
x = np.random.rand(50)
y = np.random.rand(50)

plt.scatter(x, y, color="red", marker="o")
plt.xlabel("X Data")
plt.ylabel("Y Data")
plt.title("Scatter Plot Example")
plt.show()
```

Customizing Markers in Scatter Plots:

- Circle: marker="o"
- Square: marker="s"
- Triangle: marker="^"

---

5. Density and Contour Plots

Density Plot (2D Histogram):

```
x = np.random.randn(1000)
y = np.random.randn(1000)

plt.hexbin(x, y, gridsize=30, cmap="Blues")
plt.colorbar(label="Density")
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.title("Density Plot")
plt.show()
```

Contour Plot (Contours of a Function):

```
x = np.linspace(-3, 3, 100)
y = np.linspace(-3, 3, 100)
X, Y = np.meshgrid(x, y)
Z = np.sin(np.sqrt(X**2 + Y**2))

plt.contourf(X, Y, Z, cmap="coolwarm")
plt.colorbar()
plt.title("Contour Plot")
plt.show()
```

---

6. Histograms, Binning & Density

Creating a Histogram:

```
data = np.random.randn(1000)

plt.hist(data, bins=30, color="green", edgecolor="black")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.title("Histogram Example")
plt.show()
```

Customizing Bins:

```
plt.hist(data, bins=[-3, -2, -1, 0, 1, 2, 3], density=True)
```

---

7. Customizing Plot Legends

Adding a Legend to a Plot:

```
plt.plot(x, np.sin(x), label="Sine")
plt.plot(x, np.cos(x), label="Cosine")
plt.legend(loc="upper right")
plt.show()
```

Legend Positioning:

- loc="upper left"
- loc="upper right"
- loc="lower left"
- loc="lower right"

---

8. Colour Bars in Matplotlib

Adding a Colour Bar:

```
plt.scatter(x, y, c=x + y, cmap="viridis")
plt.colorbar(label="Color Intensity")
plt.show()
```

Setting Color Map (cmap) Options:

- "viridis"
- "plasma"
- "coolwarm"
- "Blues"

---

9. Three-Dimensional Plotting in Matplotlib

Creating a 3D Plot:

```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")

X = np.linspace(-5, 5, 100)
Y = np.linspace(-5, 5, 100)
X, Y = np.meshgrid(X, Y)
Z = np.sin(np.sqrt(X**2 + Y**2))

ax.plot_surface(X, Y, Z, cmap="viridis")
plt.show()
```

3D Scatter Plot:

```
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")

x = np.random.rand(100)
y = np.random.rand(100)
z = np.random.rand(100)

ax.scatter(x, y, z, c=z, cmap="coolwarm")
plt.show()
```

3D Wireframe Plot:

```
ax.plot_wireframe(X, Y, Z, color="black")
plt.show()
```

Key Takeaways

Matplotlib provides powerful visualization tools for **2D & 3D plots**.

Line & Scatter Plots → Used for **basic data visualization**.

Density & Contour Plots → Represent **intensity variations** in 2D data.

Histograms → Used for **data distribution analysis**.

Customizing Plots → Legends, color bars, and labels improve readability.

3D Plots → Visualize **complex datasets** using plot_surface() and scatter3D().

This **Matplotlib Cheat Sheet** covers **line plots, scatter plots, histograms, contour plots, 3D plotting, and customization techniques**. Let me know if you need further explanations!