📘 Matplotlib Advanced Plotting Notes

🎨 Colored Scatter Plot

➤ Color Based on Values

You can color each point based on a third variable (like category or value):

```
x = [1, 2, 3, 4]
y = [10, 20, 15, 25]
colors = [100, 200, 300, 400]  # Value-based coloring
plt.scatter(x, y, c=colors, cmap='viridis')
plt.colorbar()  # Adds a color scale
```

📏 Change Plot Size

```
plt.figure(figsize=(8, 5))  # width, height in inches
plt.plot([1, 2, 3], [4, 5, 6])
```

📝 Add Annotations

```
plt.plot([1, 2, 3], [4, 5, 6])
plt.annotate("This point", xy=(2, 5), xytext=(2.2, 5.5), arrowprops=dict(arrowstyle='->'))
```

↕️ Add Horizontal and Vertical Lines

➤ Horizontal Line

```
plt.axhline(y=10, color='red', linestyle='--')
```

➤ Vertical Line

```
plt.axvline(x=2, color='green', linestyle='-.')
```

➤ Line at a Specific Point

```
plt.axhline(y=max(y), label="Max value")
```

```
plt.axvline(x=x[np.argmax(y)], label="X of Max")
```

■ Subplots (Multiple Plots in One Figure)

➤ Basic Subplots

```
fig, axs = plt.subplots(2, 2, figsize=(10, 6))

axs[0, 0].plot([1, 2, 3], [4, 5, 6])

axs[0, 1].bar(["A", "B"], [5, 7])

axs[1, 0].hist([1, 2, 2, 3])

axs[1, 1].pie([10, 20, 70], labels=["A", "B", "C"])
```

🔺 3D Plots (Scatter, Line, Surface)

➤ Enable 3D Plotting

```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()

ax = fig.add_subplot(111, projection='3d')
```

➤ 3D Scatter Plot

```
ax.scatter(x, y, z, c='r', marker='o')
```

➤ 3D Line Plot

```
ax.plot(x, y, z)
```

➤ 3D Surface Plot

```
X, Y = np.meshgrid(np.linspace(-5, 5, 100), np.linspace(-5, 5, 100))

Z = np.sin(np.sqrt(X**2 + Y**2))

ax.plot_surface(X, Y, Z, cmap='viridis')
```

🔁 Contour and Filled Contour Plots

➤ Contour Plot

```
plt.contour(X, Y, Z)
```

➤ Filled Contour Plot

```
plt.contourf(X, Y, Z, cmap='plasma')
plt.colorbar()
```

🌡️ Heatmap

➤ Plot a Heatmap from Matrix

```
data = np.random.rand(5, 5)
plt.imshow(data, cmap='hot', interpolation='nearest')
plt.colorbar()
```

🐼 Pandas Built-in Plotting

➤ Line Plot

```
df.plot()  # Default is line plot
```

➤ Scatter Plot with Customization

```
df.plot.scatter(x='height', y='weight', color='red', marker='^', figsize=(6,4), cmap='coolwarm')
```

➤ Bar Charts

```
df.plot.bar()           # Vertical bars
df.plot.barh()          # Horizontal bars
df.plot.bar(stacked=True)  # Stacked bars
```

➤ Histogram

```
df['age'].plot.hist(bins=10)
```

➤ Pie Chart

```
df['category'].value_counts().plot.pie(autopct='%1.1f%%')
```

📚 Multiple Separate Graphs Together

To display multiple plots without overlapping:

```
plt.figure()
plt.plot([1, 2, 3])
plt.figure()
```

plt.bar([1, 2, 3], [3, 2, 1])

🧩 Pie Chart with MultiIndex DataFrame

You can group multi-index data and then plot a pie chart:

df.groupby(["category", "sub_category"])["value"].sum().unstack().plot.pie(subplots=True, autopct='%1.1f%%')
