Sub Plots

Certainly! Subplots are a way to display multiple plots within the same figure, allowing for the simultaneous visualization of different aspects of the data or multiple datasets. Each subplot is like a smaller, individual plot within the larger figure.

Key Features of Subplots:

1. **Grid Layout:**

   - Subplots are arranged in a grid, and you can specify the number of rows and columns.

2. **Axes:**

   - Each subplot has its own set of axes, which can be customized independently.

3. **Common Figure:**

   - All subplots share the same figure, making it easy to compare different aspects of the data.

4**. Examples:**

1. **Time Series Comparison:**

   - Suppose you have multiple time series datasets (e.g., stock prices for different companies). You can create subplots to compare their trends.

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

# Generate sample data
time = np.arange(0, 10, 0.1)
data1 = np.sin(time)
data2 = np.cos(time)

# Create subplots
```

Sub Plots

plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)

plt.plot(time, data1, label='Company A')

plt.title('Stock Price - Company A')

plt.subplot(1, 2, 2)

plt.plot(time, data2, label='Company B', color='orange')

plt.title('Stock Price - Company B')

plt.tight_layout()

plt.show()

```

This example creates two subplots side by side, each showing the stock prices of different companies over time.

2. **Data Comparison:**

- If you have multiple datasets you want to compare (e.g., sales figures for different products), subplots can help visualize them together.

*```python*

*import matplotlib.pyplot as plt*

*import numpy as np*

*# Generate sample data*

*products = ['Product A', 'Product B', 'Product C']*

*sales1 = [20, 35, 15]*

*sales2 = [25, 30, 20]*

*# Create subplots*
```

Sub Plots

```
plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)
plt.bar(products, sales1, color='blue')
plt.title('Sales - Scenario 1')

plt.subplot(1, 2, 2)
plt.bar(products, sales2, color='green')
plt.title('Sales - Scenario 2')

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

This example creates two subplots comparing sales figures for different products in two different scenarios.

3. **Multi-Variable Analysis:**

- Subplots are useful for visualizing relationships between multiple variables.

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

# Generate sample data
x = np.linspace(0, 10, 100)
y1 = x + np.random.normal(0, 1, 100)
y2 = 2 * x + np.random.normal(0, 2, 100)

# Create subplots
```

Sub Plots

plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)

plt.scatter(x, y1, color='red')

plt.title('Scatter Plot - Variable 1')

plt.subplot(1, 2, 2)

plt.scatter(x, y2, color='blue')

plt.title('Scatter Plot - Variable 2')

plt.tight_layout()

plt.show()

```

This example creates two subplots with scatter plots, each representing the relationship between the independent variable `x` and a different dependent variable (`y1` and `y2`).

**Summary:**

Subplots provide a powerful way to organize and compare multiple visualizations within a single figure, making it easier to understand complex relationships or analyse different aspects of the data simultaneously.
```