22-11-2024

Training Day – 48

November 22, Friday

- ***Topic:* Creating a Dashboard**
  - Integrated multiple Matplotlib visualizations into one figure.
  - Example: Combined a line chart, bar chart, and pie chart in subplots.

Matplotlib allows you to combine multiple visualizations (such as line charts, bar charts, and pie charts) into a single figure using **subplots**. This is useful when you want to display different types of visualizations side-by-side for comparative purposes or for a more comprehensive view of the data.

1. Using Subplots in Matplotlib

Subplots allow you to arrange multiple plots in a grid layout. You can specify the number of rows and columns in the grid, and then plot different visualizations in each grid cell.

2. Example: Combining Line Chart, Bar Chart, and Pie Chart in Subplots
In this example, we'll create a figure that contains three different plots:
A **line chart** showing a trend over time.
A **bar chart** representing categorical data.
A **pie chart** showing the proportions of categories.
Code Example:

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

# Sample data
x = np.arange(1, 6)
y1 = [2, 4, 6, 8, 10]  # Line chart data
y2 = [5, 3, 6, 2, 7]   # Bar chart data
labels = ['A', 'B', 'C', 'D', 'E']  # Pie chart categories
sizes = [15, 30, 45, 10, 20]  # Pie chart data

# Create a figure with 1 row and 3 columns
fig, axs = plt.subplots(1, 3, figsize=(15, 5))

# Line chart in the first subplot
axs[0].plot(x, y1, marker='o', color='b', label='Trend')
axs[0].set_title('Line Chart')
axs[0].set_xlabel('X Axis')
axs[0].set_ylabel('Y Axis')
axs[0].legend()

# Bar chart in the second subplot
axs[1].bar(x, y2, color='g', label='Values')
axs[1].set_title('Bar Chart')
axs[1].set_xlabel('Categories')
axs[1].set_ylabel('Values')
axs[1].set_xticks(x)
axs[1].set_xticklabels(labels)
```

```
axs[1].legend()

# Pie chart in the third subplot
axs[2].pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)
axs[2].set_title('Pie Chart')

# Adjust layout to prevent overlap
plt.tight_layout()

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