Data Visualization with Matplotlib and Seaborn

Data visualization is essential for understanding and communicating insights from data. Python provides two powerful libraries for visualization: **Matplotlib** (for creating static, interactive, and animated plots) and **Seaborn** (a statistical data visualization library built on top of Matplotlib).

---

1. Importing Libraries

python

Copy code

```
import matplotlib.pyplot as plt
import seaborn as sns
```

---

2. Matplotlib Basics

Creating a Simple Plot

python

Copy code

```
x = [1, 2, 3, 4, 5]
y = [10, 20, 25, 30, 35]

plt.plot(x, y, marker='o', linestyle='-', color='blue',
label='Line')
plt.title('Simple Line Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.legend()
plt.grid(True)
plt.show()
```

Bar Plot

python

Copy code

```
categories = ['A', 'B', 'C', 'D']
values = [5, 7, 3, 8]

plt.bar(categories, values, color='skyblue')
plt.title('Bar Plot')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.show()
```

Histogram

python

Copy code

```
data = [1, 1, 2, 3, 3, 3, 4, 5, 5, 5, 6, 7, 8]

plt.hist(data, bins=5, color='purple', alpha=0.7, edgecolor='black')
plt.title('Histogram')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```

Scatter Plot

python

Copy code

```
x = [1, 2, 3, 4, 5]
y = [2, 4, 1, 3, 7]

plt.scatter(x, y, color='red', marker='x')
plt.title('Scatter Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```

---

3. Seaborn Basics

Seaborn simplifies statistical data visualization and provides better aesthetics.

Bar Plot

python

Copy code

```
data = {'Category': ['A', 'B', 'C', 'D'], 'Values': [10, 15, 7, 10]}
df = pd.DataFrame(data)
```

```
sns.barplot(x='Category', y='Values', data=df, palette='viridis')
plt.title('Seaborn Bar Plot')
plt.show()
```

Histogram / KDE Plot

Histogram:
python
Copy code

```
sns.histplot(data=data, bins=5, kde=False, color='green')
plt.title('Seaborn Histogram')
plt.show()
```

-

KDE (Kernel Density Estimate):
python
Copy code

```
sns.kdeplot(data=data, color='blue', shade=True)
plt.title('KDE Plot')
plt.show()
```

-

Scatter Plot

python
Copy code

```
tips = sns.load_dataset('tips')  # Sample dataset
sns.scatterplot(x='total_bill', y='tip', data=tips, hue='time',
style='time', size='size', palette='coolwarm')
plt.title('Seaborn Scatter Plot')
plt.show()
```

Box Plot

python
Copy code

```
sns.boxplot(x='day', y='total_bill', data=tips, palette='Set2')
plt.title('Box Plot')
plt.show()
```

Violin Plot

python

Copy code

```
sns.violinplot(x='day', y='total_bill', data=tips, palette='muted',
split=True)
plt.title('Violin Plot')
plt.show()
```

Heatmap

python

Copy code

```
flights = sns.load_dataset('flights')
pivot_table = flights.pivot('month', 'year', 'passengers')

sns.heatmap(pivot_table, annot=True, fmt="d", cmap='YlGnBu')
plt.title('Heatmap')
plt.show()
```

Pair Plot

python

Copy code

```
sns.pairplot(tips, hue='time', palette='coolwarm')
plt.title('Pair Plot')
plt.show()
```

---

4. Customizing Plots

Titles and Labels:

python

Copy code

```
plt.title('Custom Title')
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')
```

-

Grid:

python

Copy code

```
plt.grid(True)
```

-

Figure Size:
python
Copy code

```
plt.figure(figsize=(10, 6))
```

-

Save the Plot:
python
Copy code

```
plt.savefig('plot.png')
```

-

---

5. Combining Matplotlib and Seaborn

You can use Seaborn for styling and Matplotlib for customization:

python
Copy code

```
sns.set_style('whitegrid')
plt.figure(figsize=(8, 6))

sns.barplot(x='day', y='total_bill', data=tips, palette='coolwarm')
plt.title('Bar Plot with Combined Customization', fontsize=14)
plt.xlabel('Day of the Week', fontsize=12)
plt.ylabel('Total Bill ($)', fontsize=12)

plt.show()
```