Visualization Guide: Comparing Matplotlib and Seaborn

Visualization Guide: Comparing Matplotlib and Seaborn

1. Library Overview

Matplotlib

Matplotlib is a widely used low-level plotting library for creating static, animated, and interactive visualizations in Python. It provides full control over plot appearance and is considered the foundational library on which others like Seaborn are built.

Key Features:

Highly customizable

Publication-quality plots

Integrates with NumPy and Pandas

Typical Use Cases:

Research and academic publications

Basic data visualization

Custom visual design

Seaborn

Seaborn is a high-level statistical data visualization library built on top of Matplotlib. It provides a more user-friendly interface and adds support for complex visualizations with less code.

Key Features:

- Built-in themes and color palettes

- Simplified syntax

- Integrated with Pandas DataFrames

Typical Use Cases:

- Statistical plots

- Exploratory data analysis

- Visualizing distributions and relationships

Visualization Guide: Comparing Matplotlib and Seaborn

2. Graph Types with Code Example

A.Matplotlib

1. Line Plot

Description: Displays information as a series of data points connected by lines.

Use Case: Visualize trends over time.

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [10, 20, 25, 30, 40]
plt.plot(x, y)
plt.title("Line Plot")
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.show()
```

2. Scatter Plot

Description: Shows relationships between two variables.

Use Case: Visualize correlation.

```
x = [1, 2, 3, 4, 5]
y = [5, 7, 4, 6, 9]
plt.scatter(x, y)
plt.title("Scatter Plot")
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

3. Bar Chart

Description: Represents categorical data with rectangular bars.

Use Case: Compare categories.

```
categories = ['A', 'B', 'C']
values = [10, 24, 36]
plt.bar(categories, values)
plt.title("Bar Chart")
plt.show()
```

4. Histogram

Description: Displays the distribution of a dataset.

Use Case: Analyze frequency of data intervals.

```
import numpy as np
data = np.random.randn(1000)
plt.hist(data, bins=30)
plt.title("Histogram")
plt.show()
```

5. Pie Chart

Description: Shows proportions of a whole.

Use Case: Visualize percentage composition.

```
labels = ['A', 'B', 'C']
sizes = [30, 45, 25]
```

```
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.title("Pie Chart")
plt.show()
```

B. Seaborn

1. Line Plot

Description: Enhanced line plot with automatic confidence intervals.

Use Case: Time series with trend analysis.

```
import seaborn as sns
import pandas as pd
data = pd.DataFrame({'x': [1, 2, 3, 4, 5], 'y': [10, 20, 25, 30, 40]})
sns.lineplot(data=data, x="x", y="y")
plt.title("Line Plot")
plt.show()
```

2. Scatter Plot

Description: Supports grouping and hue-based coloring.

Use Case: Cluster analysis.

```
tips = sns.load_dataset("tips")
sns.scatterplot(data=tips, x="total_bill", y="tip", hue="sex") plt.title("Scatter
Plot")
plt.show()
```

3. Bar Chart

Description: Summary statistics with error bars.

Use Case: Compare averages.

```
sns.barplot(data=tips, x="day", y="total_bill")
plt.title("Bar Chart")
plt.show()
```

4. Histogram (Displot)

Description: Distribution histogram with flexible bin size.

Use Case: Frequency analysis.

```
sns.histplot(tips["total_bill"], bins=20)
plt.title("Histogram")
plt.show()
```

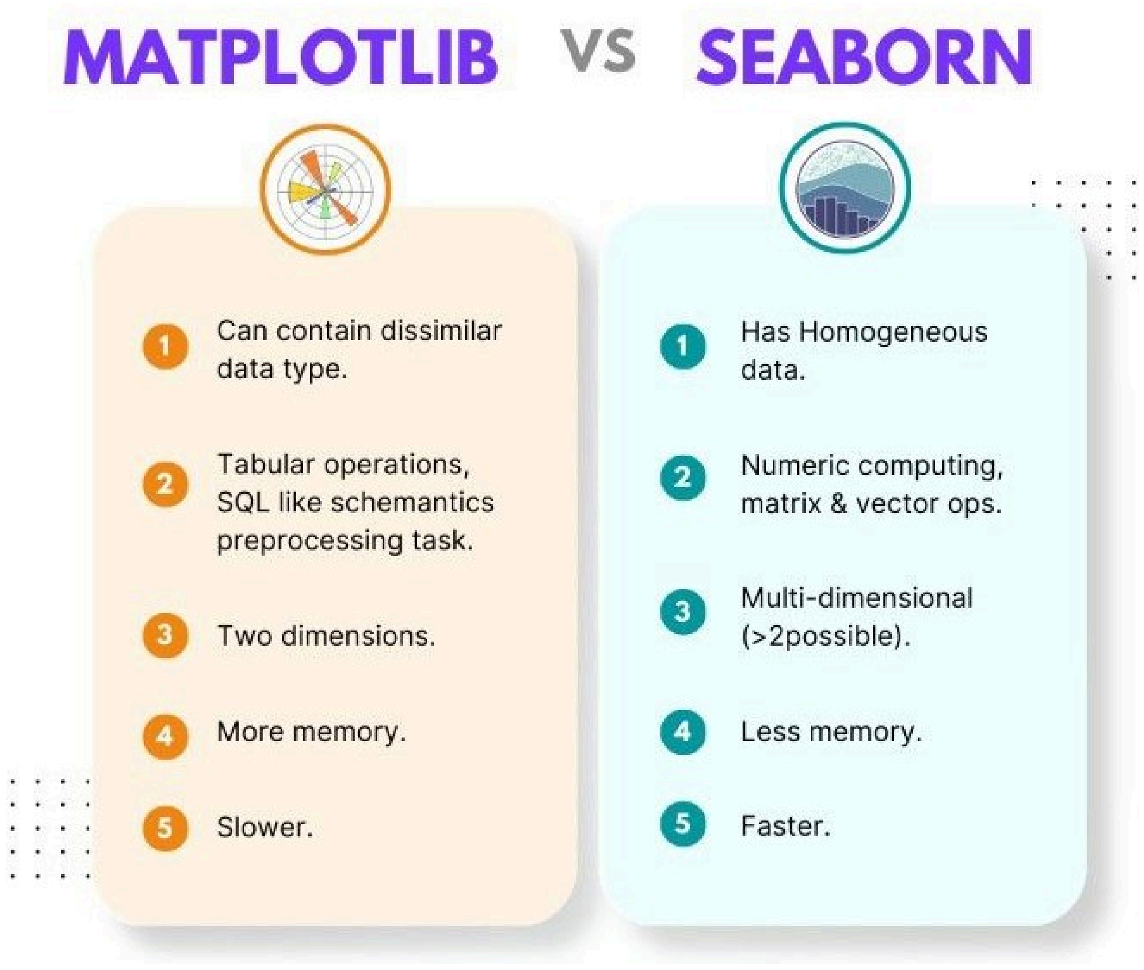
5. Boxplot

Description: Shows distribution summary with quartiles.

Use Case: Outlier detection.

```
sns.boxplot(data=tips, x="day", y="total_bill")
plt.title("Boxplot")
plt.show()
```

3. Comparison: Matplotlib vs. Seaborn



MATPLOTLIB VS SEABORN

MATPLOTLIB	SEABORN
1 Can contain dissimilar data type.	1 Has Homogeneous data.
2 Tabular operations, SQL like schemantics preprocessing task.	2 Numeric computing, matrix & vector ops.
3 Two dimensions.	3 Multi-dimensional (>2possible).
4 More memory.	4 Less memory.
5 Slower.	5 Faster.

Summary:

- Use Matplotlib when you need complete control over every visual element or are creating complex custom visualizations.

- Use Seaborn when performing exploratory data analysis or when you need beautiful, statistical plots with minimal code.

4. Resources

- Matplotlib Quick Start Guide: https://matplotlib.org/stable/users/explain/quick_start.html

- Seaborn Introduction Tutorial: https://seaborn.pydata.org/tutorial/introduction.html