Visualization Library Documentation

Library Covered

Matplotib

Seaborn

1. Introduction

Visualization is a way representing. Data visualization is representing of data in a graphical or pictorial format for better understanding of data. Visualiztion gives a good idea of data and the trends in it.

In python, data visualization has multiple libraries like matplotlib, seaborn,plotly etc.

Let us dive deeper into matplotlib and seaborn.

2. Matplotlib

Overview

Matplotlib introduced in 2002 by john hunter. It is for plotting 2D plots which is built on numpy.

Matplotlib is a low-level, highly customizable Python library used for creating static, animated, and interactive visualizations. It is widely used for basic plotting and forms the foundation for many other visualization libraries.

Key Features: - Highly customizable plots - Supports many plot types - Works well with NumPy and Pandas

Use Cases: - Academic research - Data analysis reports - Basic chart creation

2.2 Types of Graphs in Matplotlib

a) Line Plot

Description: Display data points connected by straight lines.

Use Case: Trend analysis over time.

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]

y = [2, 4, 6, 8, 10]

plt.plot(x, y)

plt.xlabel("X-axis")

plt.ylabel("Y-axis")

plt.title("Simple Line Plot")

plt.show()
```

b) Scatter Plot

Description: Shows relationship between two variables.

Use Case: Correlation analysis

```
plt . scatter(x,y)

plt . title(‘Scatter Plot’)

plt . show()
```

c) Bar Chart

Desciption: Represents categorical data using bars.

Use Case: Comparing quantities.

```
Categories = [ ‘A’, ‘B’, ‘C’ ]

Values = [5,7,3]

plt. bar(categories, values)

plt.title(‘Bar Chart’)

plt.show()
```

d) Histogram

Description: Display frequency distribution of data.

Use Case: Understanding data distribution.

```
Data = [1, 2, 2, 3, 3 ,3, 4, 4, 5]

plt.hist(data, bins=5)

plt.title(‘ Histogram’)

plt.Show()
```

3. Seaborn

3.1 Overview

Seaborn is a high-level data visualization library built on top of Matplotlib. It provides attractive and informative statistical graphics with minimal code.

Key Features: - Built-in themes - Statistical plots - Works directly with Pandas DataFrames

Use Cases: - Exploratory Data Analysis (EDA) - Statistical visualization

3.2 Types of Graphs in Seaborn

a) Line Plot

Description: Similar to Matplotlib line plot but visually enhanced.

Use Case: Performance Analysis

```
Import seaborn as sns
Import pandas as pd

Data = pd.DataFrame( { ‘x’ : [ 1, 2, 3, 4], ‘y’ : [10, 20, 25, 30] } )
sns. lineplot( x= ‘x’ , y= ‘y’ , data=data)
```

b) Scatter Plot

Description: Displays relationship with enhanced styling.

Use Case: Study hours vs Exam marks

```
import seaborn as sns
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5, 6]
y = [10, 12, 14, 13, 15, 18]
sns.scatterplot(x=x, y=y)
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Simple Scatter Plot")
plt.show()
```

c) Bar Chart

Description: Shows categorical data with confidence intervals.

Use Case: Sales by product

sns . barplot (x= ‘x’ , y= ‘y’ , data= data)

d) Histogram

Description: Display relationship with enhanced styling.

Use case: Marks distribution

sns.histplot (data [‘y’] , kde=True)

4. Comparison: Matplotlib vs Seaborn

Matplotlib

1.Ease of Use

- Low-level plotting library that provides full control over every plot element.
- Requires more code for creating complex or visually appealing plots.
- Steeper learning curve for beginners due to detailed syntax.

2. Customization Options

- Highly customizable: every component (axes, labels, ticks, legends) can be modified.
- Suitable for creating publication-quality and highly specific visualizations.
- Ideal when fine-grained control is required.

3. Interactivity

- Supports basic interactivity (zoom, pan, save) in GUI backends.
- Limited built-in interactive features.
- Not ideal for interactive dashboards.

4. Performance with Large Datasets

- Can handle large datasets but may become slow with millions of points.
- Performance depends on plot type and rendering backend.
- Suitable for moderate to large datasets with optimization.

Seaborn

1. Ease of Use

- High-level library built on top of Matplotlib.
- Simplifies plotting with concise, readable code.
- Automatically applies attractive styles and color palettes.
- Easier for beginners and faster for exploratory data analysis.

2. Customization Options

- Provides limited customization compared to Matplotlib.
- Focuses on aesthetics and statistical plots rather than deep control.
- Custom changes often require switching to Matplotlib functions.

3. Interactivity

- Inherits interactivity from Matplotlib.
- No additional interactive functionality of its own.
- Mainly designed for static statistical plots.

4. Performance with Large Datasets

- Generally slower than Matplotlib for large datasets.
- Performs additional statistical computations (aggregation, estimation).
- Better suited for summarized or sampled data rather than raw large datasets.

5. Conclusion

Use Matplotlib when you need full control, custom plots, and better performance.

Use Seaborn when you want quick, beautiful statistical visualizations with minimal code.