

Task Level (Beginner):

Python Visualization Guide: Matplotlib & Seaborn

This guide provides a beginner-friendly overview of two of the most essential Python visualization libraries: **Matplotlib** and **Seaborn**.

1. Library Overview

Matplotlib

Matplotlib is the "grandfather" of Python visualization. It is a low-level library that provides total control over every element of a figure, including line styles, font properties, and axes.

- **Unique Features:** Extremely powerful, highly customizable, and serves as the foundation for many other libraries (including Seaborn and Pandas plotting).
- **Typical Use Case:** Creating publication-quality static figures or when precise control over the UI is required.

Different Types of Plots in Matplotlib

Matplotlib offers a wide range of plot types to suit various data visualization needs. Here are some of the most commonly used types of plots in Matplotlib:

- Line Graph
- Bar Chart
- Histogram
- Scatter Plot
- Pie Chart
- 3D Plot

Key Uses of Matplotlib:

- **Basic Plots:** Line plots, bar charts, histograms, scatter plots, etc.
- **Statistical Visualization:** Box plots, error bars, and density plots.

- **Customization:** Control over colors, labels, gridlines, and styles.
- **Subplots & Layouts:** Create multiple plots in a single figure.
- **3D Plotting:** Surface plots and 3D scatter plots using `mpl_toolkits.mplot3d`.
- **Animations & Interactive Plots:** Dynamic visualizations with `FuncAnimation`.
- **Integration:** Works well with Pandas, NumPy and Jupyter Notebooks.

Seaborn

Seaborn is built on top of Matplotlib and integrates closely with Pandas data structures. It provides a high-level interface for drawing attractive and informative statistical graphics.

- **Unique Features:** Comes with beautiful default themes, handles complex statistical aggregations automatically, and requires significantly less code than Matplotlib for complex plots.
- **Typical Use Case:** Exploratory data analysis (EDA) and visualizing statistical relationships.

Different categories of plot in Seaborn

Plots are basically used for visualizing the relationship between variables. Those variables can be either completely numerical or a category like a group, class, or division. Seaborn divides the plot into the below categories –

- **Relational plots:** This plot is used to understand the relation between two variables.
- **Categorical plots:** This plot deals with categorical variables and how they can be visualized.
- **Distribution plots:** This plot is used for examining univariate and bivariate distributions
- **Regression plots:** The regression plots in Seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.

- **Matrix plots:** A matrix plot is an array of scatterplots.
- **Multi-plot grids:** It is a useful approach to draw multiple instances of the same plot on different subsets of the dataset.

2. Graph Types & Examples

A. Matplotlib Examples

1. Line Plot

Description: Shows information as a series of data points connected by straight line segments.
Use Case: Tracking changes over short and long periods of time.

Python Code:

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

plt.plot(x, y, marker='o', color='b', label='Trend')
plt.title("Simple Line Plot")
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.legend()
plt.show()
```

2. Bar Chart

Description: Represents categorical data with rectangular bars with heights proportional to the values. **Use Case:** Comparing quantities across different categories.

Python Code:

```
categories = ['Apples', 'Bananas', 'Cherries']
values = [10, 15, 7]
plt.bar(categories, values, color='orange')
plt.title("Fruit Count")
plt.show()
```

B. Seaborn Examples

1. Histogram (Distplot)

Description: Shows the distribution of a numeric variable.

Use Case: Understanding the spread and skewness of data (e.g., age distribution).

Python Code

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

data = [1, 2, 2, 3, 3, 3, 4, 4, 5, 8, 9]
sns.histplot(data, kde=True, color="skyblue")
plt.title("Distribution of Data with Density Curve")
plt.show()
```

2. Scatter Plot

Description: Displays values for typically two variables for a set of data.

Use Case: Identifying correlations or relationships between two continuous variables.

Python Code

```
tips = sns.load_dataset("tips") # Built-in dataset
sns.scatterplot(data=tips, x="total_bill", y="tip", hue="time")
plt.title("Total Bill vs Tip (Colored by Time)")
plt.show()
```

3. Comparison

Feature	Matplotlib	Seaborn
Ease of Use	Steeper learning curve for complex plots.	Very easy; handles dataframes automatically.
Customization	Infinite; you can change every pixel.	High; but focused on statistical aesthetics.
Interactivity	Primarily static (unless using widgets).	Primarily static (built on Matplotlib).
Default Aesthetics	Basic and functional.	Modern, colorful, and professional.
Performance	Excellent with large datasets.	Slower than Matplotlib for massive datasets.