

Practice Session Assignment

Name-Priyam Mohite

Divison - CM

Batch -CM-1

Roll No -23

PRN- 202401030043

**Title: significance of all basic graphs required for
Visulization**

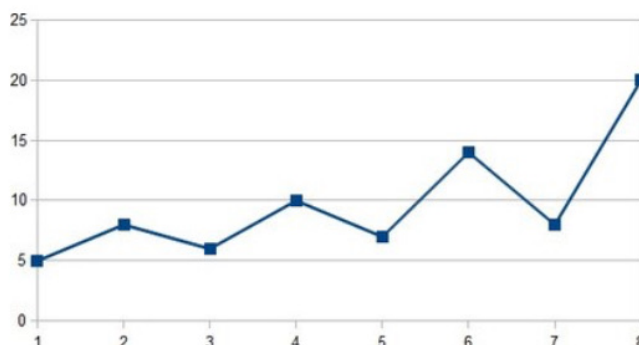
Data visualization plays a crucial role in understanding and interpreting data effectively. Python, with libraries like Matplotlib, Seaborn, and Plotly, offers various types of graphs to help visualize data insights. Here are the basic graphs commonly used and their significance:

1. Line Graph

Significance: Used to visualize trends over time (timeseries data).

Ideal For: Stock prices, weather reports, and performance metrics.

Python Example: `plt.plot(x, y)`

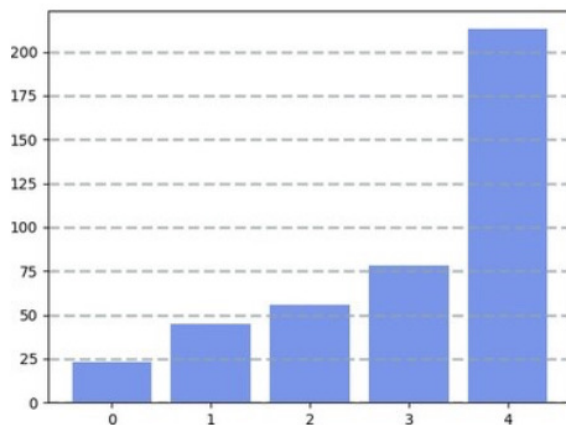


2. Bar Chart

Significance: Useful for comparing quantities of different categories.

Ideal For: Sales reports, survey results, frequency of items.

Python Example: `plt.bar(x, y)`

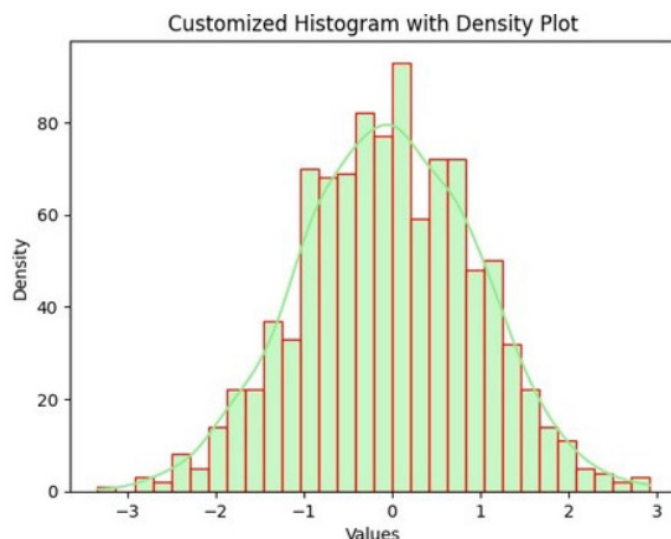


3. Histogram

Significance: Shows the distribution of a single continuous variable.

Ideal For: Understanding frequency distribution, outliers, and skewness.

Python Example: `plt.hist(data, bins=10)`

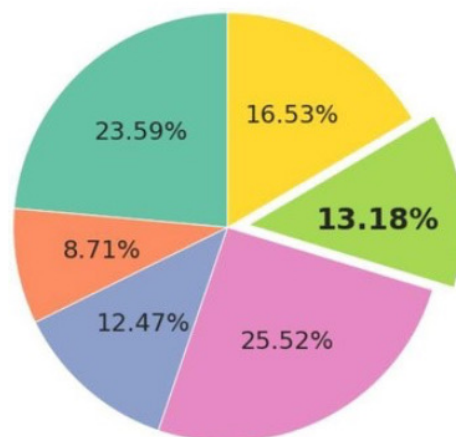


4. Pie Chart

Significance: Represents proportions and percentage distribution of categories.

Ideal For: Market share, budget allocation, composition of elements.

Python Example: `plt.pie(sizes, labels=labels, autopct='%1.1f%%')`

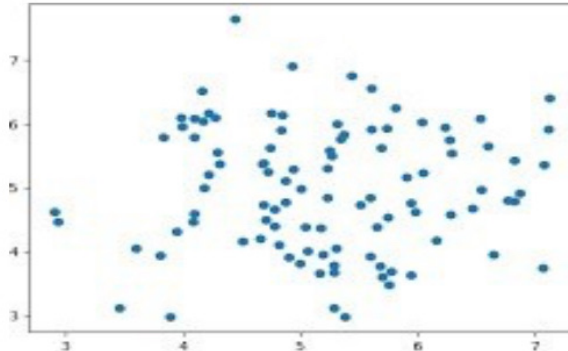


5. Scatter Plot

Significance: Displays the relationship or correlation between two continuous variables.

Ideal For: Regression analysis, identifying patterns and outliers.

Python Example: `plt.scatter(x, y)`

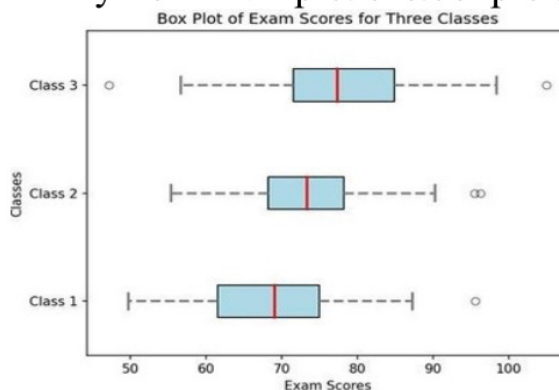


6. Box Plot (Box-and-Whisker Plot)

Significance: Summarizes distribution using median, quartiles, and outliers.

Ideal For: Comparing distributions across categories.

Python Example: `sns.boxplot(x=data)`

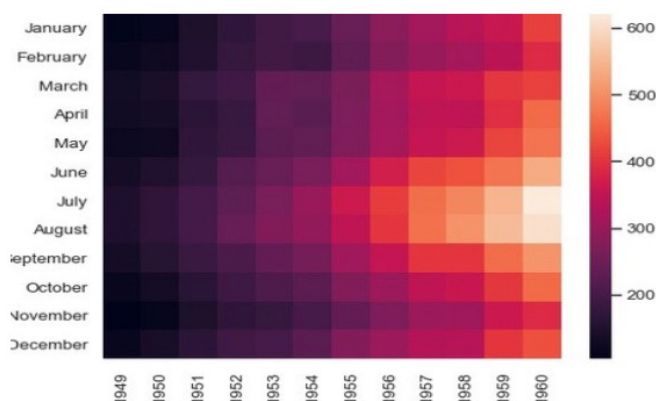


7. Heatmap

Significance: Shows correlation or frequency using color intensity.

Ideal For: Correlation matrices, confusion matrices.

Python Example: `sns.heatmap(data, annot=True)`



8. Area Chart

Significance: Like a line chart, but with filled areas to emphasize volume.

Ideal For: Cumulative data trends over time.

Python Example: `plt.fill_between(x, y)`

