**ESSENTIAL OF DATA SCIENCE**

**PRACTICE SESSION ACTIVITY**

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

*CS3-71 CHETAN BALIRAM GORE*

*ROLL NO.:-CS3-71*

*PRN:-202401040350*

*DIVISION:-CS3(BATCH-CS34*

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

*Significance of Basic Graphs in Python Data Visualization:-*

**1. Bar Graph**

* **Purpose in Python:**  
  Created using plt.bar() in Matplotlib or sns.barplot() in Seaborn.
* **Significance:**  
  Shows comparison between different categorical values. It highlights the differences and similarities across categories quickly.

**2. Line Graph**

* **Purpose in Python:**  
  Created using plt.plot() in Matplotlib.
* **Significance:**  
  Best for showing trends over time (time series data) such as stock prices, temperature changes, or sensor readings.

**3. Pie Chart**

* **Purpose in Python:**  
  Created using plt.pie() in Matplotlib.
* **Significance:**  
  Used to visualize the percentage or proportional data for small datasets, making parts of a whole easy to understand.

**4. Histogram**

* **Purpose in Python:**  
  Created using plt.hist() in Matplotlib or sns.histplot() in Seaborn.
* **Significance:**  
  Displays the frequency distribution of continuous numerical data. Important for understanding the shape (normal, skewed) of data distribution.

**5. Scatter Plot**

* **Purpose in Python:**  
  Created using plt.scatter() in Matplotlib or sns.scatterplot() in Seaborn.
* **Significance:**  
  Shows the relationship between two numerical variables. Useful for identifying correlations, patterns, and outliers.

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

* **Purpose in Python:**  
  Created using sns.boxplot() in Seaborn.
* **Significance:**  
  Summarizes data distribution using minimum, first quartile, median, third quartile, and maximum. Essential for detecting outliers and comparing groups.

**7. Area Chart**

* **Purpose in Python:**  
  Created using plt.fill\_between() in Matplotlib or by modifying plt.plot().
* **Significance:**  
  Displays cumulative quantities over time, helping in visualizing part-to-whole relationships when dealing with continuous data.

**8. Heatmap**

* **Purpose in Python:**  
  Created using sns.heatmap() in Seaborn.
* **Significance:**  
  Displays matrix-like data (such as correlation matrices) with color coding, making it easier to spot patterns and relationships at a glance.

**9. Bubble Chart**

* **Purpose in Python:**  
  Created using plt.scatter() with varying s (size) parameter in Matplotlib.
* **Significance:**  
  Adds a third dimension (bubble size) to scatter plots, allowing multi-variable analysis visually.

**10. Treemap**

* **Purpose in Python:**  
  Created using squarify.plot() from the Squarify library.
* **Significance:**  
  Visualizes hierarchical data as nested rectangles. Useful for showing proportions between categories and subcategories in a compact form.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CONCLUSION**

In Data Visualization, choosing the right type of graph ensures that the data story is clear, accurate, and impactful. Basic graphs like bar plots, line plots, histograms, and scatter plots are foundational for effective data analysis and machine learning workflows.

\*\*\*