Name – Sayyam Jain

PRN – 22311031

Roll No – 282036

Year – SY B

**Assignment No – 3**

**Problem Statement :**

Visualize the data using R/Python by plotting the graphs for assignment no. 1 and 2. Consider

suitable data set. Use Scatter plot, Bar plot, Box plot, Pie chart, Line Chart.

**Objective:**

In order to explore important patterns, distributions, and relationships using a range of graphical representations, exploratory data analysis (EDA) on an appropriate dataset using Python is the aim. In order to meet the requirements of Assignments No. 1 and No. 2, the assignment specifically include creating a bar plot, scatter plot, box plot, pie chart, and line chart. Insights into the dataset's structure, categorical distributions, numerical trends, and possible relationships will be offered by these visualizations, enabling a thorough comprehension for further analysis or modeling.

**Methodology**

The code follows a simple workflow to load and visualize a dataset:

1. Library Import: Import necessary libraries (pandas, numpy, seaborn, matplotlib.pyplot) for data manipulation, numerical operations, and visualization.
2. Data Loading: Load the mushrooms.csv dataset into a pandas DataFrame using pd.read\_csv().
3. Visualization: Create a bar plot to display the distribution of the class column using seaborn.countplot(), customized with labels and a title via matplotlib.pyplot.

**Main Functions**

1. import pandas as pd: Imports the pandas library for data manipulation and loading the CSV file.
2. import numpy as np: Imports numpy for numerical operations (though unused in this code).
3. import seaborn as sns: Imports seaborn for statistical visualization, specifically the bar plot.
4. import matplotlib.pyplot as plt: Imports matplotlib.pyplot for plot customization and display.
5. pd.read\_csv('mushrooms.csv'): Loads the mushroom dataset into a DataFrame (dataset).
6. plt.figure(figsize=(10, 5)): Sets the figure size for the plot (10 inches wide, 5 inches tall).
7. sns.countplot(x='class', data=dataset): Generates a bar plot showing the count of each unique value in the class column.
8. plt.xlabel('Class'), plt.ylabel('Count'), plt.title('Class Distribution'): Adds labels and a title to the plot.
9. plt.show(): Displays the plot.

**Advantages**

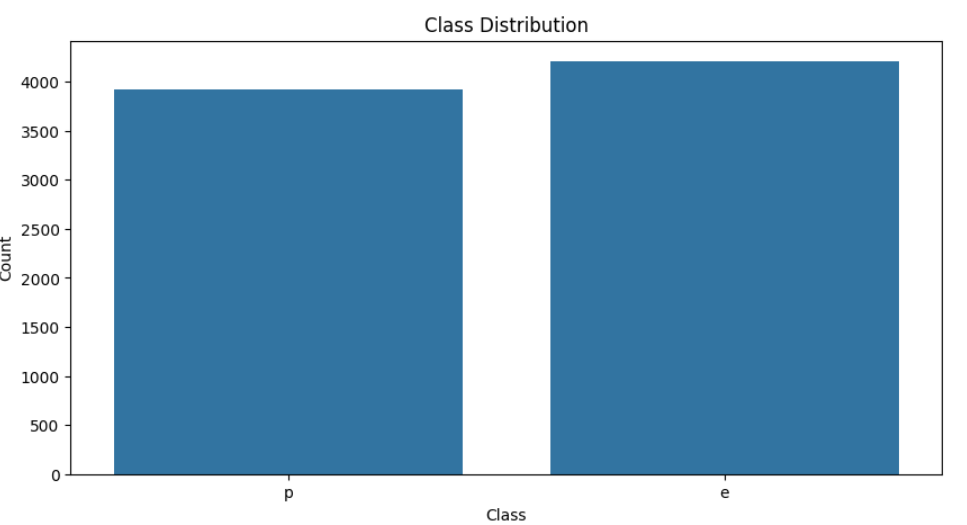
* Pandas: Easily loads data and works well with visualization software.
* NumPy: Supports other libraries and performs numerical operations quickly.
* Seaborn: Uses DataFrames to create straightforward, eye-catching bar graphs for EDA
* Matplotlib.pyplot: Controls display and allows for flexible plot customisation.

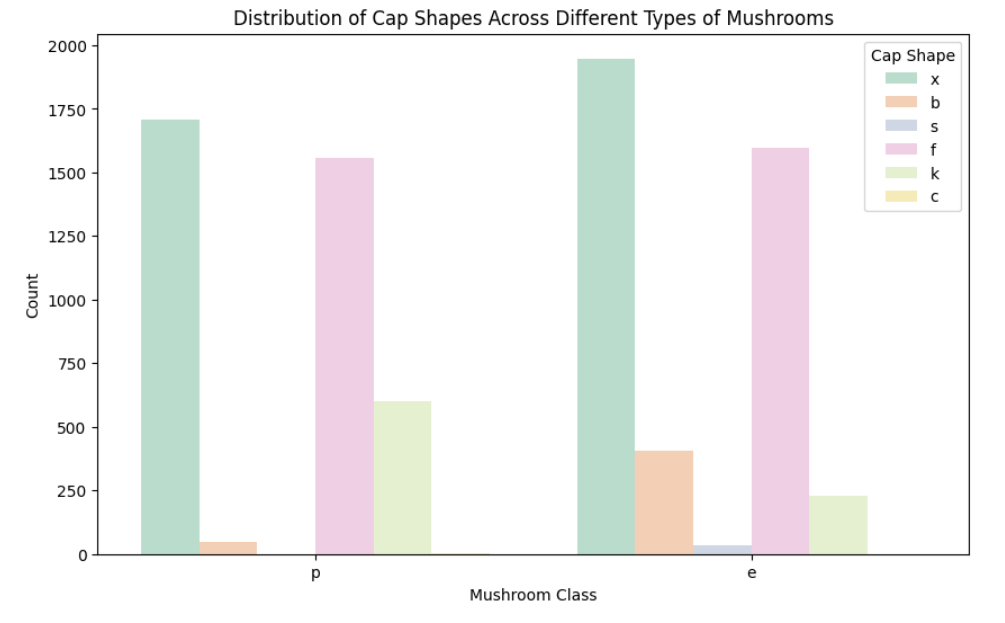
**Disadvantages**

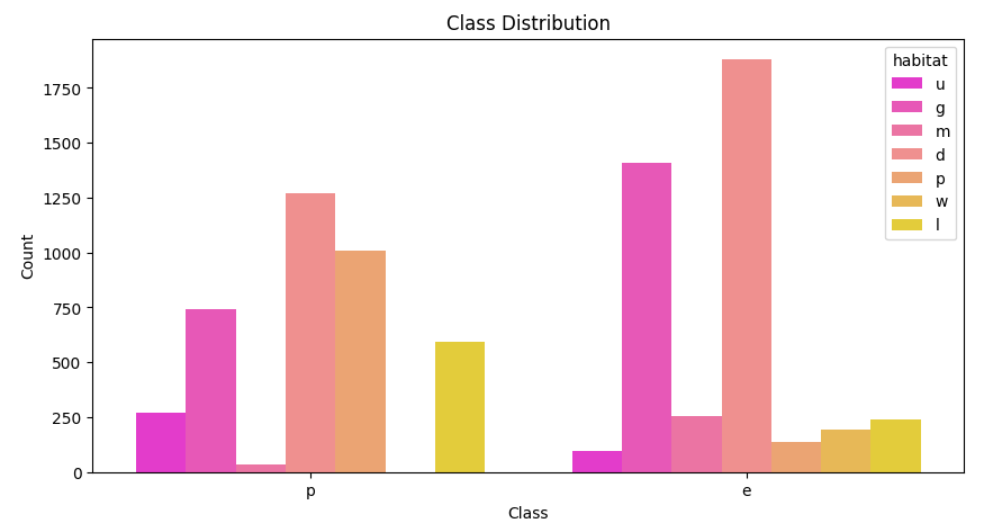
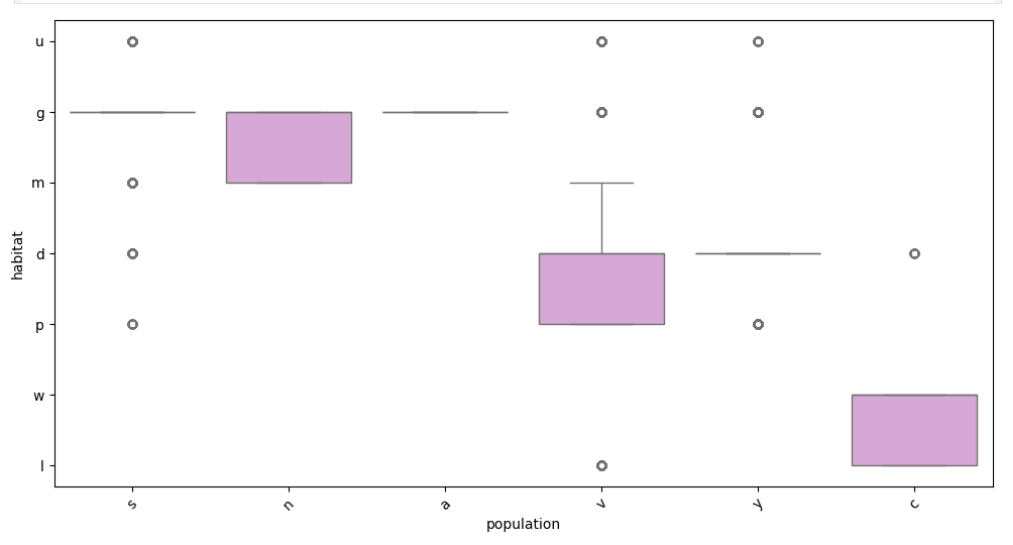
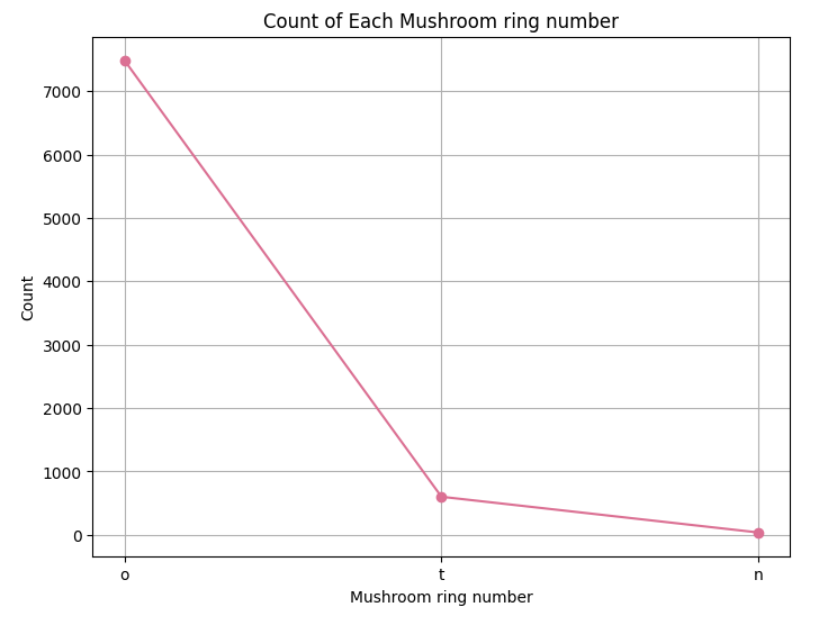
* Pandas: Memory-intensive for huge data, underutilized here.
* NumPy: Low-level for stand-alone activities, not used in this code.
* Seaborn: Depends on Matplotlib and offers little customisation.
* Matplotlib.pyplot: higher learning curve, verbose for simple jobs

**Conclusion**

A `seaborn` bar plot, customized with `matplotlib.pyplot`, is used to show the `class` column distribution after the Jupyter Notebook code loads the `mushrooms.csv` dataset using `pandas`. It's a straightforward EDA assignment that successfully illustrates class balance, but it's only available in one plot with unused `numpy` and no more analysis. Its simplicity and clarity are its strong points; its limited reach and little investigation are its faults.

**Results :**

****

****