EXPERIMENT 6

EDA – Quantitative And Quantitative Analysis

Aim:

To understand the importance of EDA – Quantitative And Quantitative Analysis.

Algorithm:

1. Import the required libraries — seaborn, pandas, numpy, and matplotlib.

2. Load the built-in “tips” dataset using sns.load\_dataset('tips').

3. Define a function show\_plot() to create and display plots easily.

4. Use box plots to visualize the distribution of total\_bill and tip.

5. Use count plots to analyze the frequency of categorical variables like day, sex, and time.

6. Create pie and bar charts to show the gender-wise (sex) distribution.

7. Use histograms with and without KDE to study the distribution of total\_bill.

8. Use joint plots (reg, hex) and pair plots to visualize relationships between numerical variables.

9. Plot a heatmap to show the correlation between numerical features in the dataset.

Program:

import seaborn as sns

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

tips = sns.load\_dataset('tips')

def show\_plot(plot\_func):

plt.figure()

plot\_func()

plt.show()

show\_plot(lambda: sns.boxplot(x=tips.total\_bill))

show\_plot(lambda: sns.boxplot(x=tips.tip))

show\_plot(lambda: sns.countplot(x=tips.day))

show\_plot(lambda: sns.countplot(x=tips.sex))

show\_plot(lambda: tips.sex.value\_counts().plot(kind='pie', autopct='%1.1f%%'))

show\_plot(lambda: tips.sex.value\_counts().plot(kind='bar'))

show\_plot(lambda: sns.countplot(x='time', data=tips))

show\_plot(lambda: sns.countplot(x='day', data=tips))

show\_plot(lambda: sns.histplot(x=tips.total\_bill, kde=True))

show\_plot(lambda: sns.histplot(x=tips.total\_bill, kde=False))

sns.jointplot(x=tips.tip, y=tips.total\_bill, kind='reg')

plt.show()

sns.jointplot(x=tips.tip, y=tips.total\_bill, kind='hex')

plt.show()

sns.pairplot(tips)

plt.show()

sns.pairplot(tips, hue='time')

plt.show()

sns.pairplot(tips, hue='day')

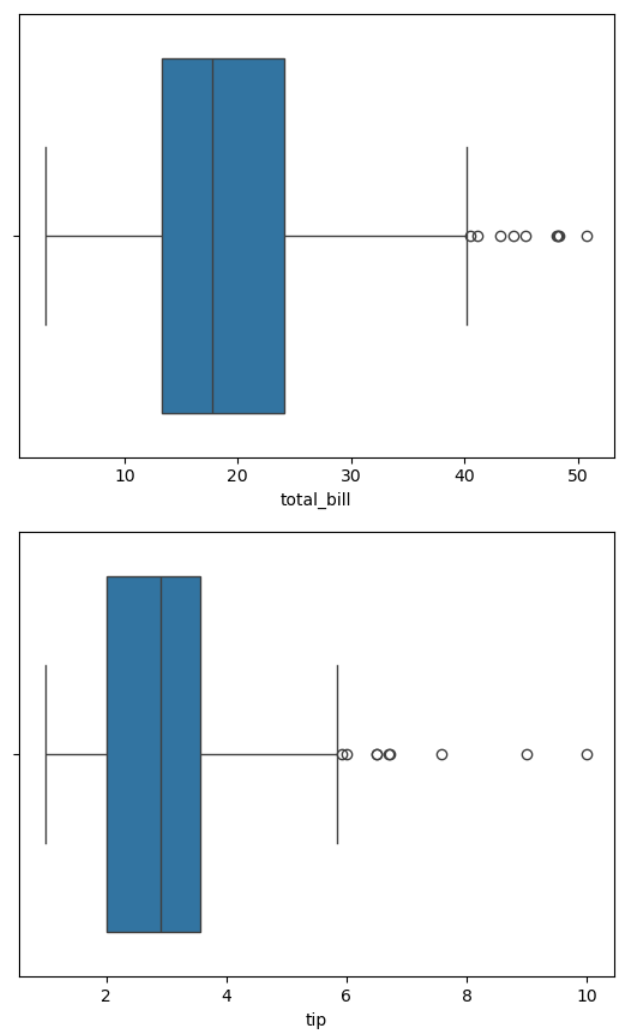
plt.show()

plt.figure(figsize=(8,6))

sns.heatmap(tips.corr(numeric\_only=True), annot=True, cmap="coolwarm")

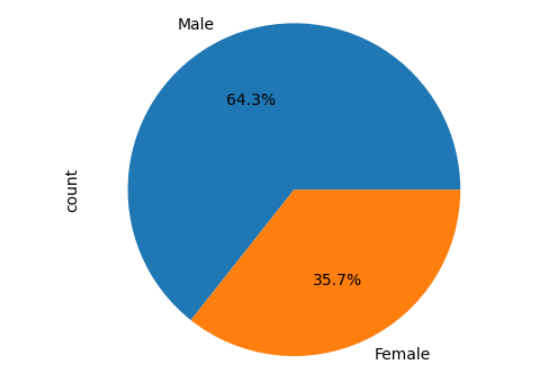
plt.show()

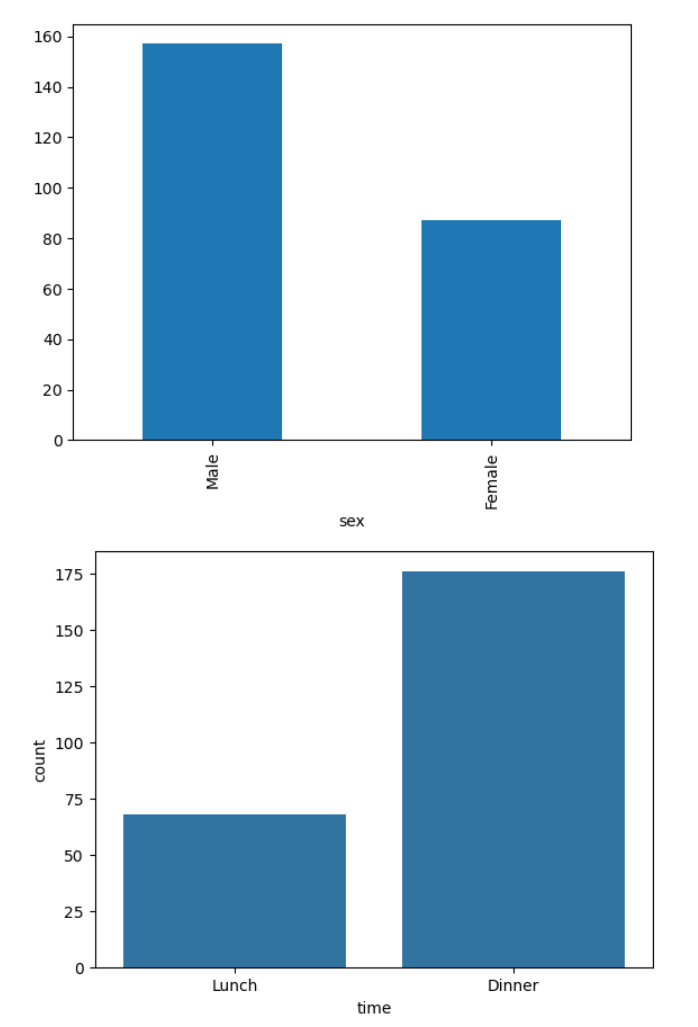
Output:

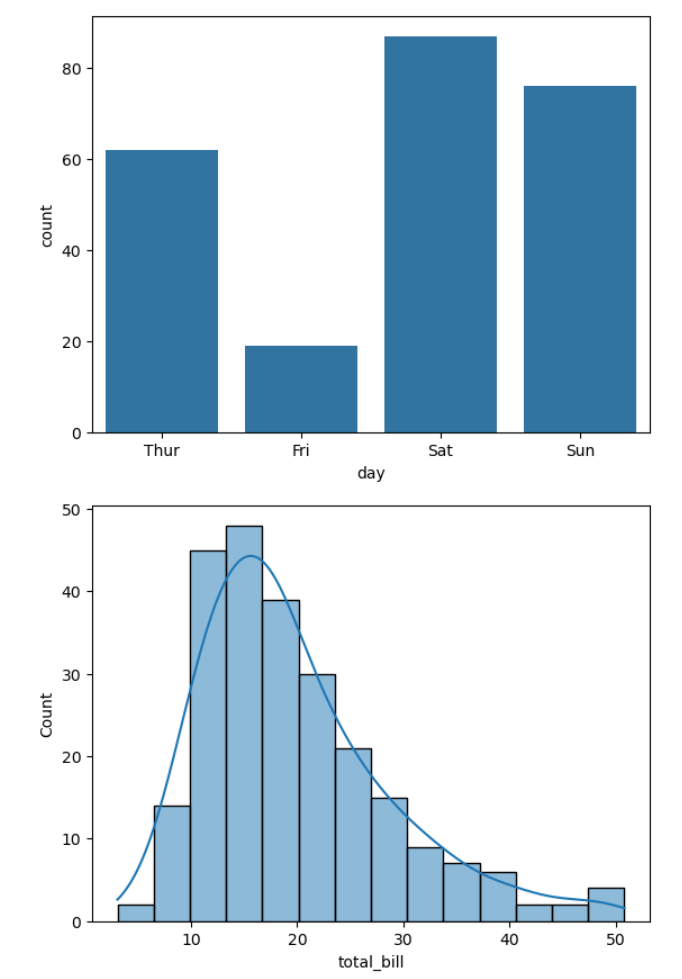


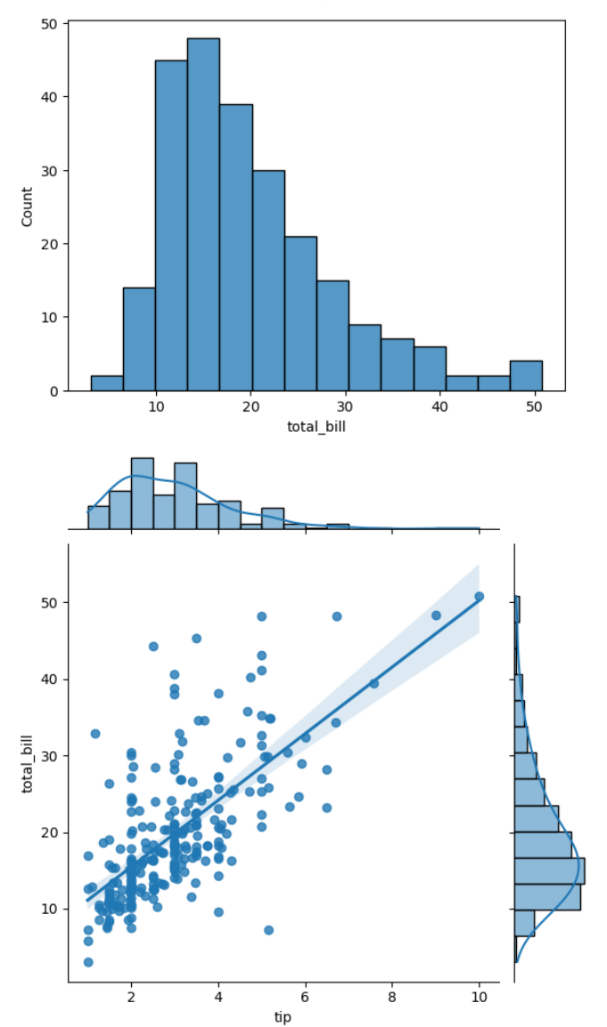
A graph of different sizes and numbers

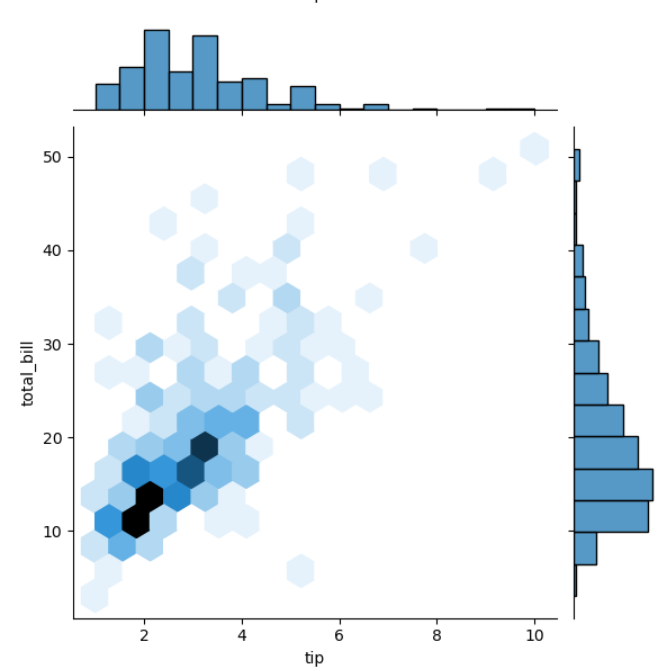
AI-generated content may be incorrect.

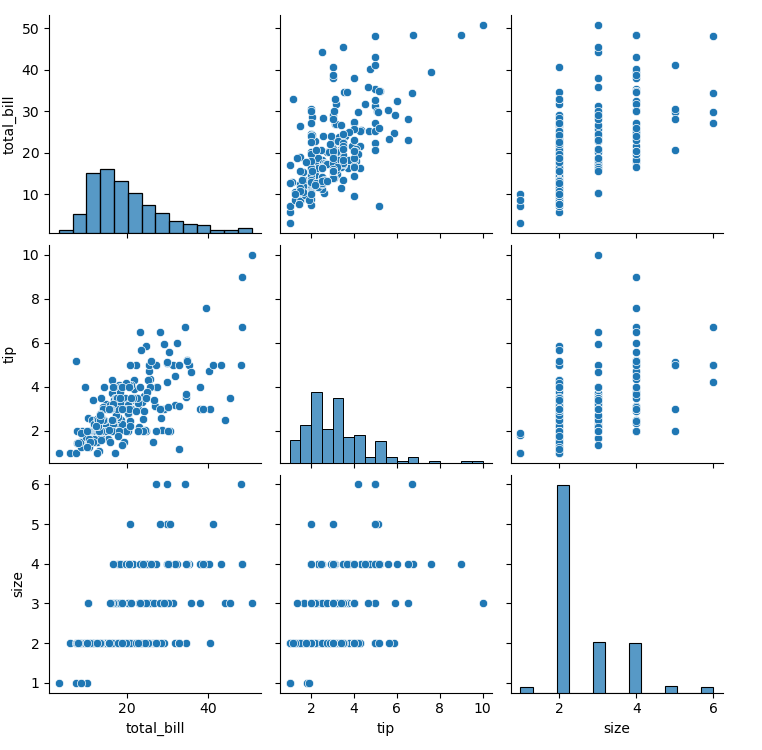


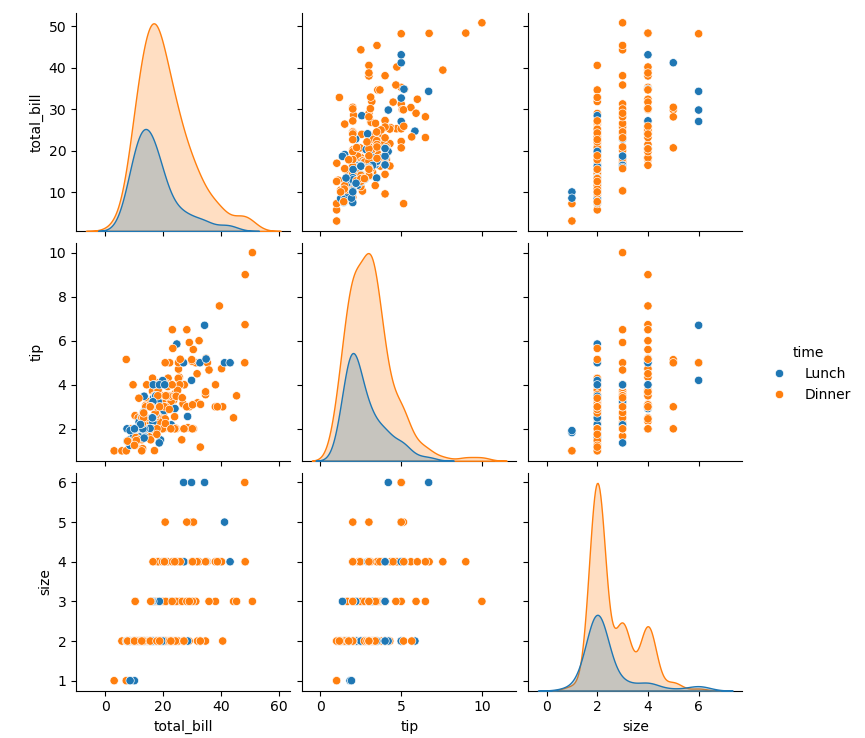


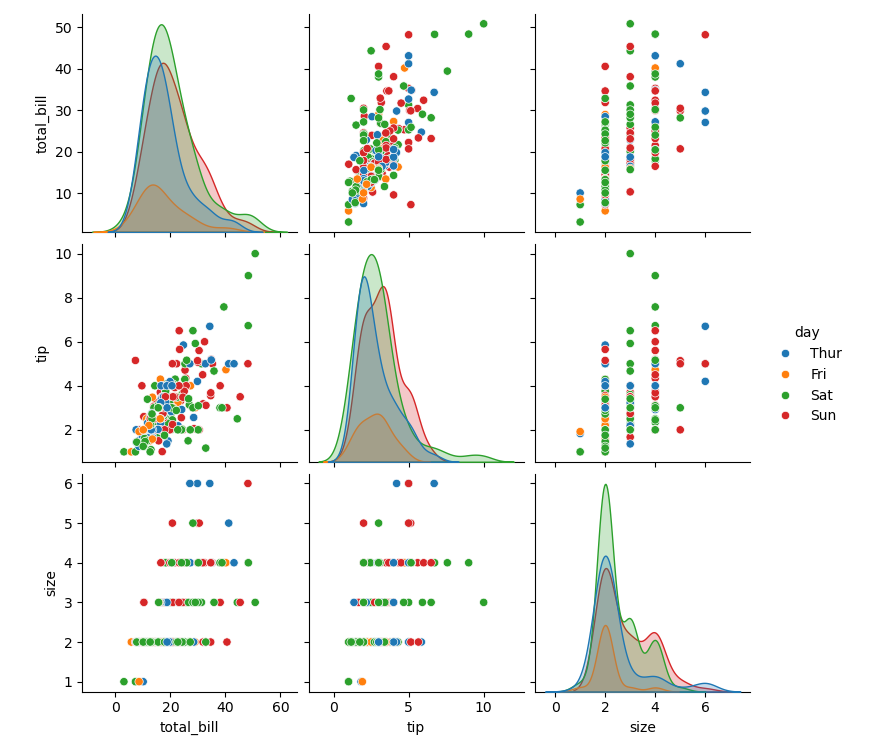


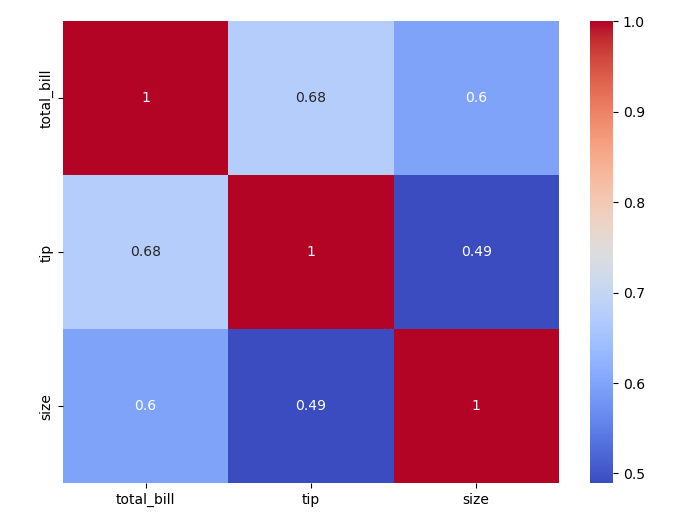












Result:

Hence a python program for EDA is written and executed successfully.