**Analysis of Restaurant Tips Dataset**

**Prepared By** : SHAILESH K R

**Submitted To** : M MUVENDIRAN (Infosys Springboard Internship Mentor)

**Subject** : A Report on Restaurant Tips Dataset.

Introduction

This report provides a detailed analysis of the "Tips" dataset, which contains information about restaurant bills, including total bill amounts, tips given, and associated attributes like gender, smoking status, and time of visit. The analysis aims to uncover patterns and correlations, helping to understand tipping behavior.

Data Exploration

1. import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

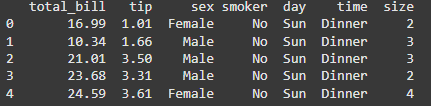
The code imports key libraries for data analysis and visualization. **Seaborn** and **Matplotlib** handle advanced and basic plotting, respectively. **Pandas** is used for efficient data manipulation, while **NumPy** supports numerical computations and array operations. Together, they enable comprehensive data exploration and visualization.

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

The code loads the **"tips" dataset**, a built-in dataset in Seaborn. It contains information about restaurant bills, including total bill amounts, tips, and customer attributes like gender, smoking status, and time of visit. This dataset is commonly used for testing and visualization purposes.

3) print(tips.head())

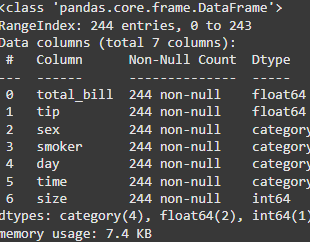
**Output :**



The code displays the first five rows of the **"tips" dataset** using the head() function. This provides a quick preview of the data structure, including columns like total\_bill, tip, sex, smoker, day, time, and size, helping to understand the dataset's content.

4) tips.info()

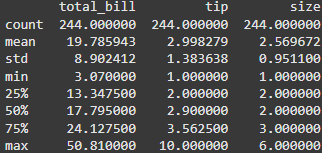
**Output:**



The tips.info() function provides a concise summary of the dataset. It shows the total number of rows, columns, data types for each column, and the count of non-null values, helping to identify missing data and understand the dataset's structure.

5) print(tips.describe())

**Output:**



The tips.describe() function generates summary statistics for the numerical columns in the dataset. It provides metrics like count, mean, standard deviation, minimum, and maximum values, as well as quartiles, giving a quick statistical overview of the data.

Data Visualisations and Insights

6) np.random.seed(42)

restaurants = ['Dominos', 'KFC', 'Mc Donalds', 'Burger King']

tips['restaurant'] = np.random.choice(restaurants, size=len(tips))

The code sets a random seed using np.random.seed(42) to ensure reproducibility of results. It then creates a new column, 'restaurant', in the tips dataset by randomly assigning one of the four restaurant names (Dominos, KFC, McDonald's, Burger King) to each row, using np.random.choice() based on the dataset's length. This simulates the restaurant chain each tip is associated with.

7) visit\_counts = tips['restaurant'].value\_counts()

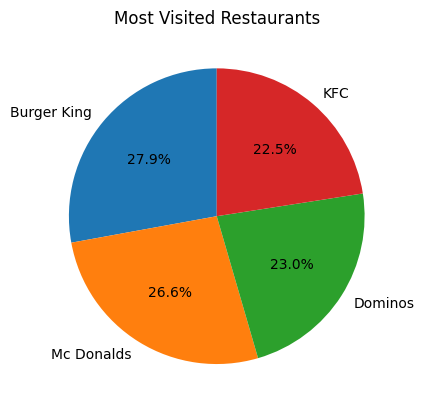
plt.pie(visit\_counts, labels=visit\_counts.index, autopct='%1.1f%%',

startangle=90)

plt.title('Most Visited Restaurants')

plt.show()

**Output:**



The code calculates the frequency of visits to each restaurant using value\_counts(). It then creates a pie chart using plt.pie() to visually represent the proportion of visits for each restaurant, displaying the percentage share. The chart is labeled and titled "Most Visited Restaurants".

8) tips\_sum = tips.groupby('restaurant')['tip'].sum()

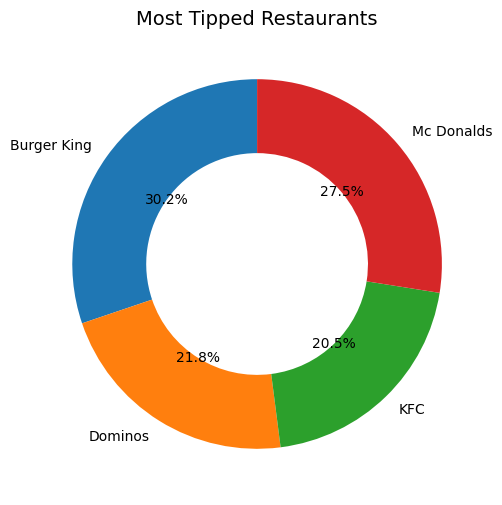
plt.figure(figsize=(8, 6))

plt.pie(tips\_sum, labels=tips\_sum.index, autopct='%1.1f%%', startangle=90, wedgeprops=dict(width=0.4))

plt.title('Most Tipped Restaurants', fontsize=14)

plt.show()

**Output:**



The code groups the tips dataset by restaurant and calculates the total tips (sum()) for each restaurant. It then generates a donut-shaped pie chart to display the percentage of total tips for each restaurant, with a custom width for the wedges. The chart is titled "Most Tipped Restaurants".

9) angles = np.linspace(0, 2 \* np.pi, len(tips\_sum), endpoint=False)

values = tips\_sum.tolist()

angles = np.append(angles, angles[0])

values.append(values[0])

fig = plt.figure()

ax = fig.add\_subplot(111, polar=True)

ax.plot(angles, values, 'o-', label='Tips')

ax.fill(angles, values, alpha=0.3)

ax.set\_xticks(angles[:-1])

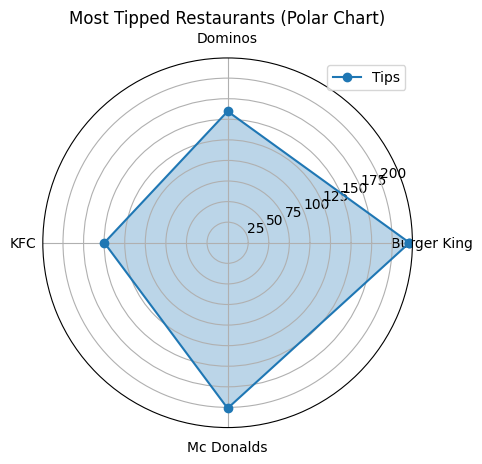
ax.set\_xticklabels(tips\_sum.index)

plt.title('Most Tipped Restaurants (Polar Chart)')

plt.legend()

plt.show()

**Output:**



The code creates a polar plot to visualize the total tips for each restaurant. It computes the angles for the chart using np.linspace() and appends the first value to close the loop. The plot is then drawn using ax.plot() and filled with ax.fill(), displaying the data in a circular format, with restaurant names as labels. The chart is titled "Most Tipped Restaurants (Polar Chart)".

10) gender\_tips = tips.groupby('sex')['tip'].sum()

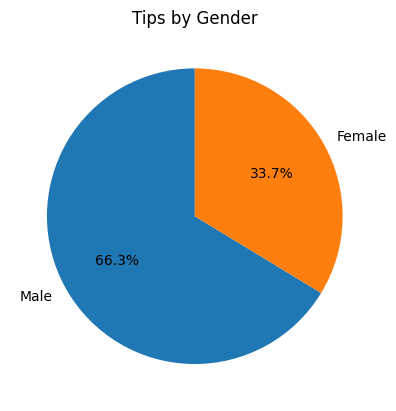
plt.pie(gender\_tips, labels=gender\_tips.index, autopct='%1.1f%%',

startangle=90)

plt.title('Tips by Gender')

plt.show()

**Output:**



The code groups the tips dataset by gender and calculates the total tips for each gender. It then creates a pie chart to visualize the percentage of tips given by males and females, with the chart labeled and titled "Tips by Gender".

11) plt.figure(figsize=(8, 5))

sns.histplot(tips['total\_bill'], kde=True, bins=20, color='blue')

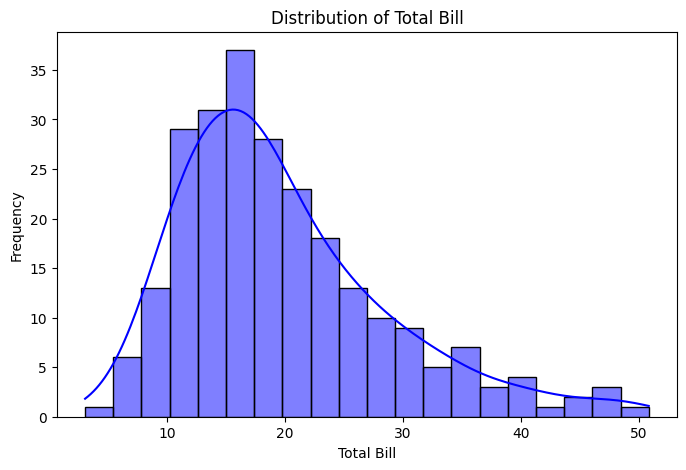
plt.title('Distribution of Total Bill')

plt.xlabel('Total Bill')

plt.ylabel('Frequency')

plt.show()

**Output:**



The code generates a histogram with a KDE (Kernel Density Estimate) overlay to visualize the distribution of the total\_bill column. It uses 20 bins and sets the color to blue. The chart is labeled with a title "Distribution of Total Bill" and axis labels for better clarity.

12) tips['tip\_percentage'] = (tips['tip'] / tips['total\_bill']) \* 100

avg\_tip\_percentage = tips.groupby('day')

['tip\_percentage'].mean().reset\_index()

plt.figure(figsize=(8, 5))

sns.barplot(x='day', y='tip\_percentage', data=avg\_tip\_percentage, pale

tte='pastel')

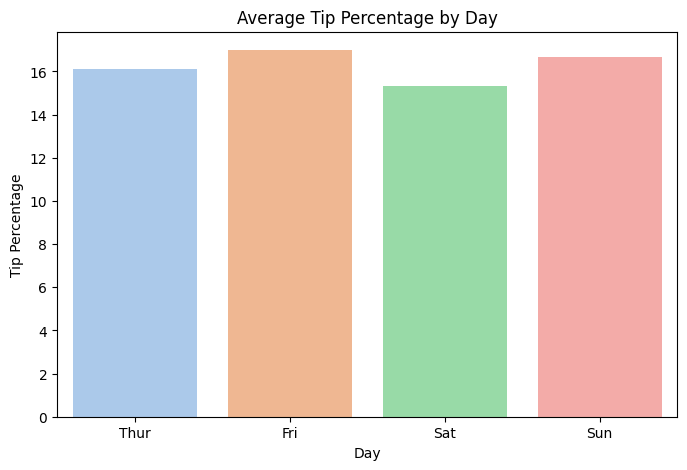
plt.title('Average Tip Percentage by Day')

plt.xlabel('Day')

plt.ylabel('Tip Percentage')

plt.show()

**Output:**



The code calculates the tip percentage for each entry by dividing the tip by the total\_bill and multiplying by 100. It then calculates the average tip percentage for each day of the week and visualizes it using a bar plot. The chart is titled "Average Tip Percentage by Day" with labeled axes for clarity.

Conclusion

This analysis provided insights into the dataset, including the distribution of tips, the relationship between total bill and tip amounts, and the impact of attributes like gender and smoking status on tipping. The visualizations and statistics enable better understanding and potential applications in customer behavior analysis.

Colab Link  
Access the notebook for this analysis   
https://colab.research.google.com/drive/10mPITEJkNThI5g0yEP6KT7PzirVWMUnB#scrollTo=vB0NIgGk4AC7