

Importing necessary Python libraries.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Step 2: Creating the data frame.

```
dataframe = pd.read_csv("Zomato_data .csv")
print(dataframe.head())
```

|   | name                  | online_order | book_table | rate  | votes | \ |
|---|-----------------------|--------------|------------|-------|-------|---|
| 0 | Jalsa                 | Yes          | Yes        | 4.1/5 | 775   |   |
| 1 | Spice Elephant        | Yes          | No         | 4.1/5 | 787   |   |
| 2 | San Churro Cafe       | Yes          | No         | 3.8/5 | 918   |   |
| 3 | Addhuri Udipi Bhojana | No           | No         | 3.7/5 | 88    |   |
| 4 | Grand Village         | No           | No         | 3.8/5 | 166   |   |

|   | approx_cost(for two people) | listed_in(type) |
|---|-----------------------------|-----------------|
| 0 | 800                         | Buffet          |
| 1 | 800                         | Buffet          |
| 2 | 800                         | Buffet          |
| 3 | 300                         | Buffet          |
| 4 | 600                         | Buffet          |

Step 3: Data Cleaning and Preparation Before moving further we need to clean and process the data.

```
def handleRate(value):
    value=str(value).split('/')
    value=value[0];
    return float(value)

dataframe['rate']=dataframe['rate'].apply(handleRate)
print(dataframe.head())
```

|   | name                  | online_order | book_table | rate | votes | \ |
|---|-----------------------|--------------|------------|------|-------|---|
| 0 | Jalsa                 | Yes          | Yes        | 4.1  | 775   |   |
| 1 | Spice Elephant        | Yes          | No         | 4.1  | 787   |   |
| 2 | San Churro Cafe       | Yes          | No         | 3.8  | 918   |   |
| 3 | Addhuri Udipi Bhojana | No           | No         | 3.7  | 88    |   |
| 4 | Grand Village         | No           | No         | 3.8  | 166   |   |

|   | approx_cost(for two people) | listed_in(type) |
|---|-----------------------------|-----------------|
| 0 | 800                         | Buffet          |
| 1 | 800                         | Buffet          |
| 2 | 800                         | Buffet          |

|   |     |        |
|---|-----|--------|
| 3 | 300 | Buffet |
| 4 | 600 | Buffet |

1. Getting summary of the dataframe use df.info().

```
dataframe.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148 entries, 0 to 147
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   name                                  148 non-null    object
1   online_order                         148 non-null    object
2   book_table                           148 non-null    object
3   rate                                 148 non-null    float64
4   votes                                148 non-null    int64
5   approx_cost(for two people)          148 non-null    int64
6   listed_in(type)                       148 non-null    object
dtypes: float64(1), int64(2), object(4)
memory usage: 8.2+ KB
```

1. Checking for missing or null values to identify any data gaps.

```
print(dataframe.isnull().sum())

name                                0
online_order                        0
book_table                          0
rate                                0
votes                                0
approx_cost(for two people)          0
listed_in(type)                      0
dtype: int64
```

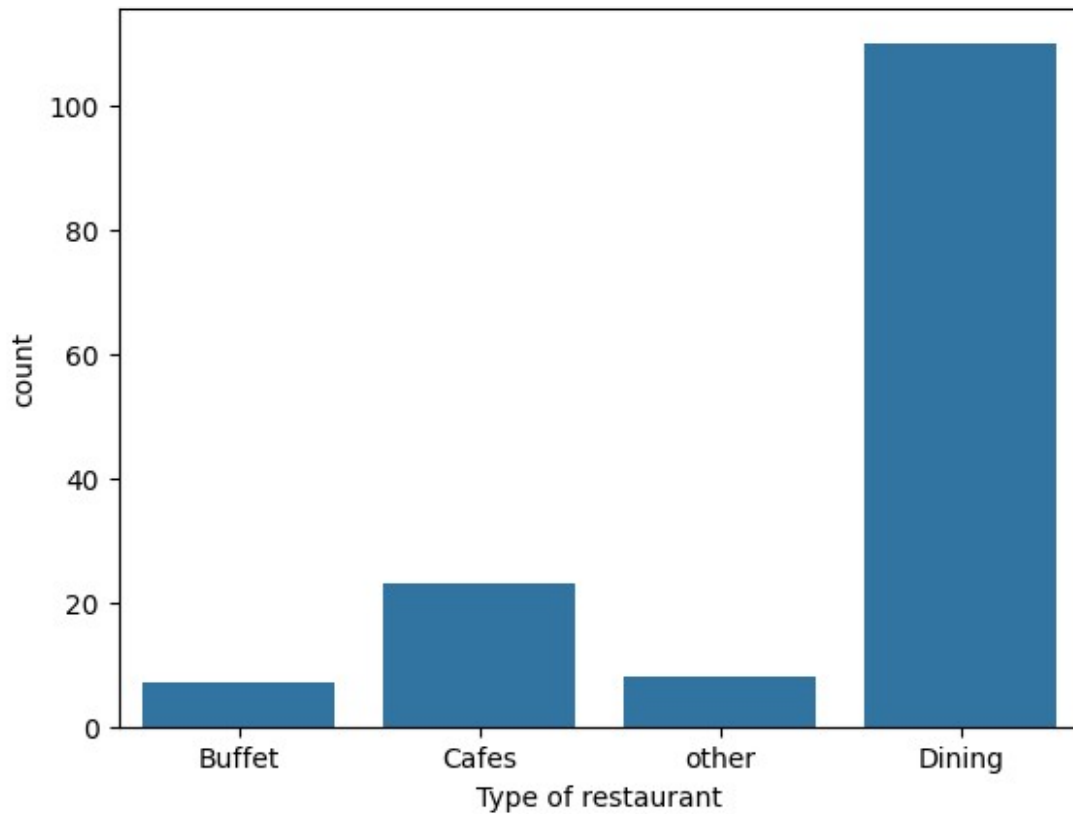
There is no NULL value in dataframe.

#### Step 4: Exploring Restaurant Types

1. Let's see the listed\_in (type) column to identify popular restaurant categories.

```
sns.countplot(x=dataframe['listed_in(type)'])
plt.xlabel("Type of restaurant")

Text(0.5, 0, 'Type of restaurant')
```



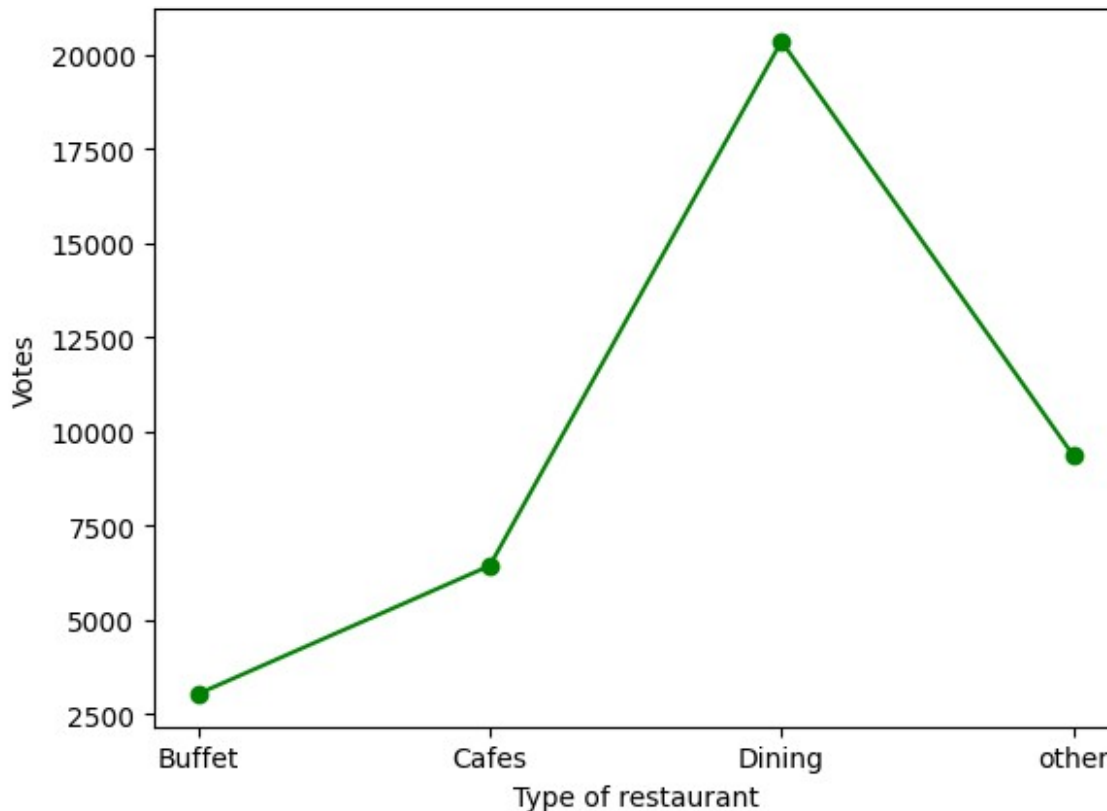
The majority of the restaurants fall into the dining category.

## 2.Votes by Restaurant Type

Here we get the count of votes for each category.

```
grouped_data = dataframe.groupby('listed_in(type)')['votes'].sum()
result = pd.DataFrame({'votes': grouped_data})
plt.plot(result, c='green', marker='o')
plt.xlabel('Type of restaurant')
plt.ylabel('Votes')
```

```
Text(0, 0.5, 'Votes')
```



Dining restaurants are preferred by a larger number of individuals.

Step 5: Identify the Most Voted Restaurant Find the restaurant with the highest number of votes

```
max_votes = dataframe['votes'].max()
restaurant_with_max_votes = dataframe.loc[dataframe['votes'] ==
max_votes, 'name']

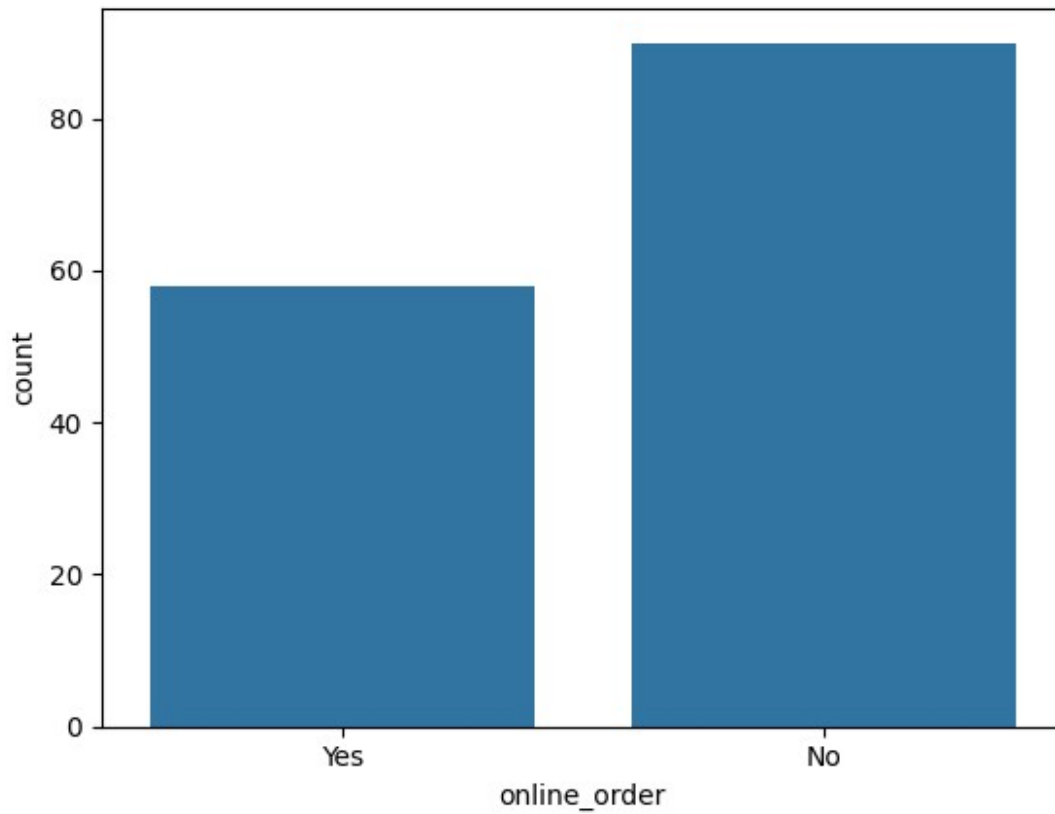
print('Restaurant(s) with the maximum votes:')
print(restaurant_with_max_votes)
```

```
Restaurant(s) with the maximum votes:
38    Empire Restaurant
Name: name, dtype: object
```

Step 6: Online Order Availability Exploring the online\_order column to see how many restaurants accept online orders.

```
sns.countplot(x=dataframe['online_order'])

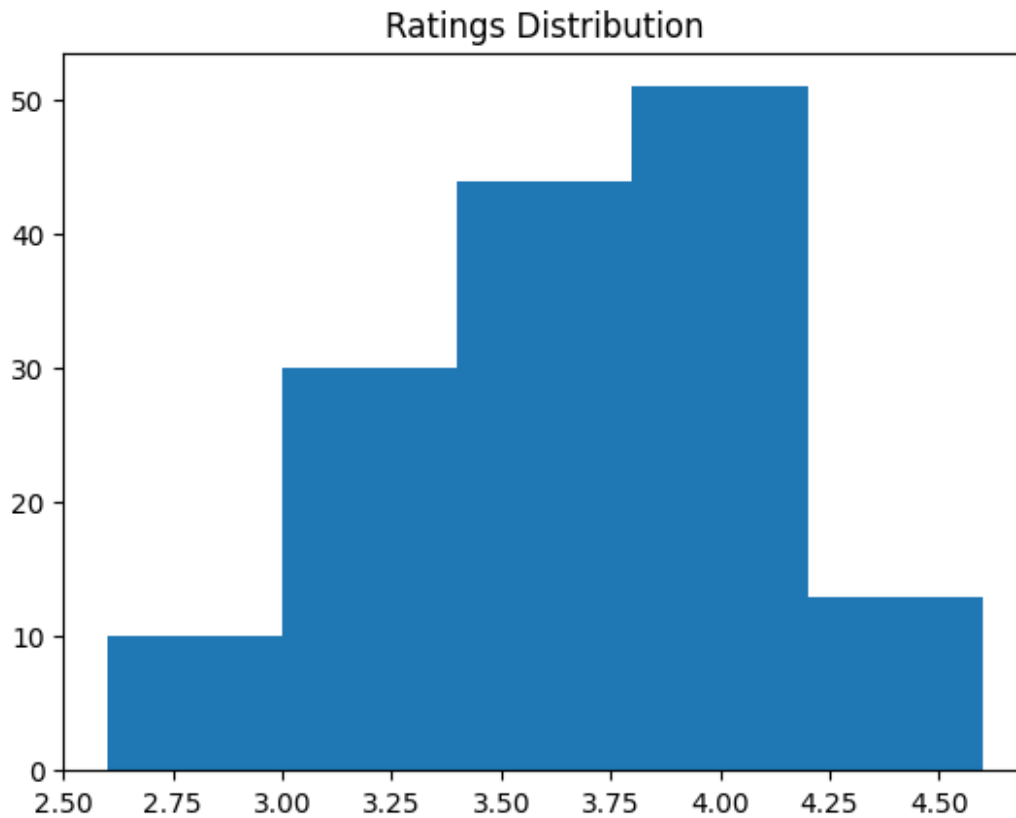
<Axes: xlabel='online_order', ylabel='count'>
```



This suggests that a majority of the restaurants do not accept online orders.

Step 7: Analyze Ratings Checking the distribution of ratings from the rate column.

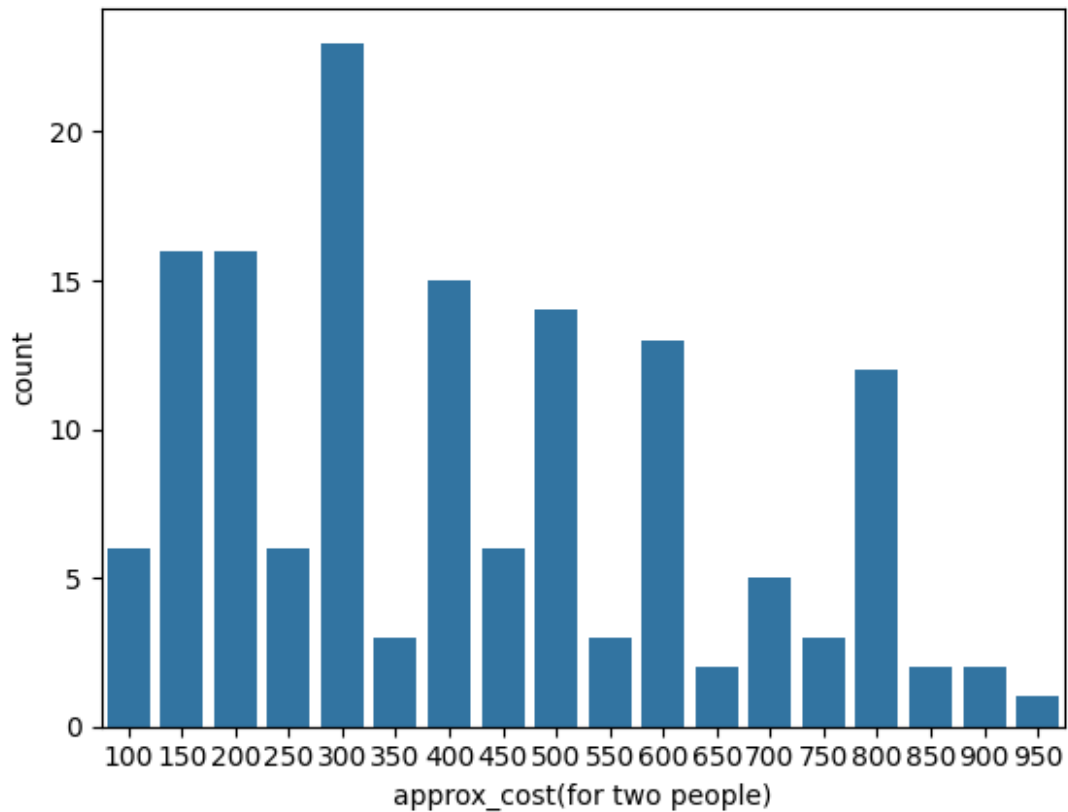
```
plt.hist(dataframe['rate'], bins=5)
plt.title('Ratings Distribution')
plt.show()
```



The majority of restaurants received ratings ranging from 3.5 to 4.

Step 8: Approximate Cost for Couples Analyze the `approx_cost(for two people)` column to find the preferred price range.

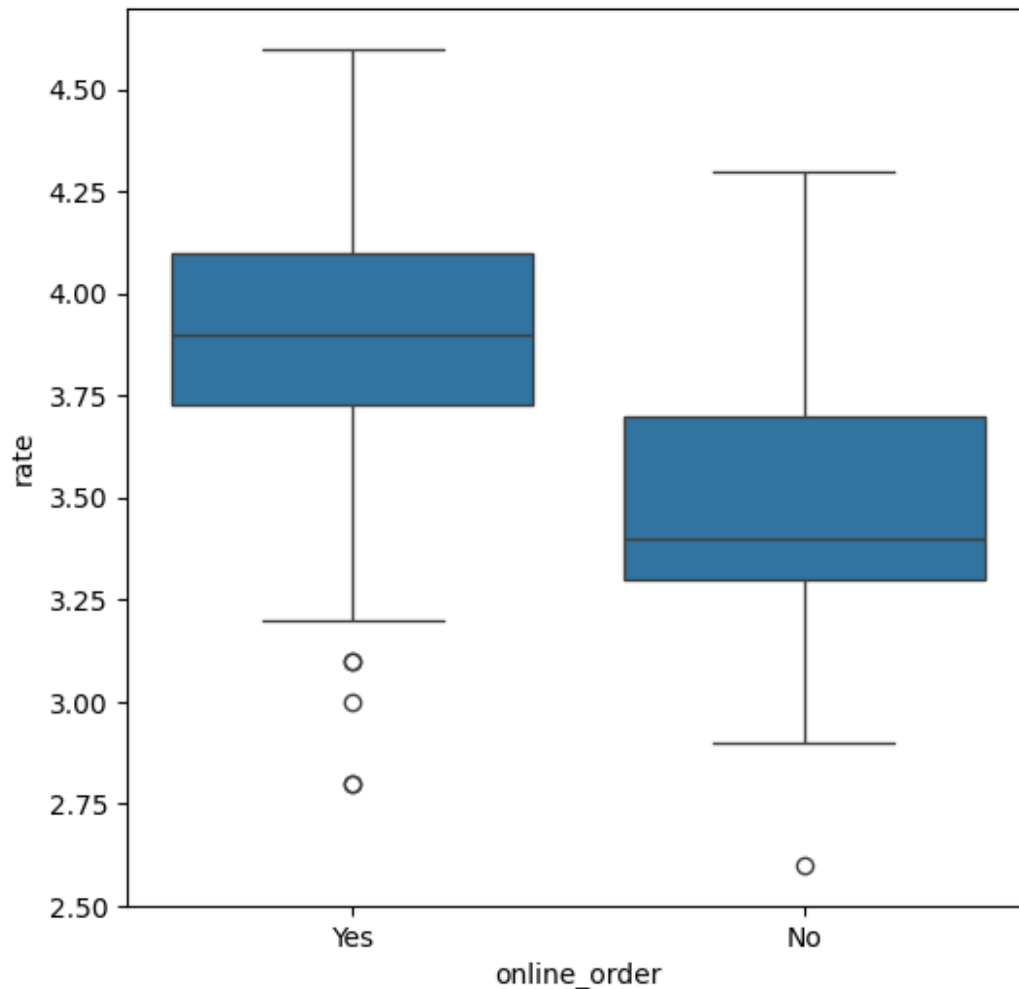
```
couple_data=dataframe['approx_cost(for two people)']  
sns.countplot(x=couple_data)  
  
<Axes: xlabel='approx_cost(for two people)', ylabel='count'>
```



The majority of couples prefer restaurants with an approximate cost of 300 rupees.

Step 9: Ratings Comparison - Online vs Offline Orders Compare ratings between restaurants that accept online orders and those that don't.

```
plt.figure(figsize = (6,6))
sns.boxplot(x = 'online_order', y = 'rate', data = dataframe)
<Axes: xlabel='online_order', ylabel='rate'>
```



Offline orders received lower ratings in comparison to online orders which obtained excellent ratings.

Step 10: Order Mode Preferences by Restaurant Type Find the relationship between order mode (online\_order) and restaurant type (listed\_in(type)).

```
pivot_table = dataframe.pivot_table(index='listed_in(type)',  
columns='online_order', aggfunc='size', fill_value=0)  
sns.heatmap(pivot_table, annot=True, cmap='YlGnBu', fmt='d')  
plt.title('Heatmap')  
plt.xlabel('Online Order')  
plt.ylabel('Listed In (Type)')  
plt.show()
```



