

In [2]: # 1

```
import pandas as pd

zomato_df = pd.read_excel('zomato.xlsx')
zomato_df
```

Out[2]:

| | Restaurant ID | Restaurant Name | Country Code | City | Address | Locality | Locality Verbose |
|------|---------------|--------------------------|--------------|------------------|---|--|---|
| 0 | 6317637 | Le Petit Souffle | 162 | Makati City | Third Floor, Century City Mall, Kalayaan Avenu... | Century City Mall, Poblacion, Makati City | Century City Mall, Poblacion, Makati City, Mak... |
| 1 | 6304287 | Izakaya Kikufuji | 162 | Makati City | Little Tokyo, 2277 Chino Roces Avenue, Legaspi... | Little Tokyo, Legaspi Village, Makati City | Little Tokyo, Legaspi Village, Makati City, Ma... |
| 2 | 6300002 | Heat - Edsa Shangri-La | 162 | Mandaluyong City | Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal... | Edsa Shangri-La, Ortigas, Mandaluyong City | Edsa Shangri-La, Ortigas, Mandaluyong City, Ma... |
| 3 | 6318506 | Ooma | 162 | Mandaluyong City | Third Floor, Mega Fashion Hall, SM Megamall, O... | SM Megamall, Ortigas, Mandaluyong City | SM Megamall, Ortigas, Mandaluyong City, Mandal... |
| 4 | 6314302 | Sambo Kojin | 162 | Mandaluyong City | Third Floor, Mega Atrium, SM Megamall, Ortigas... | SM Megamall, Ortigas, Mandaluyong City | SM Megamall, Ortigas, Mandaluyong City, Mandal... |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 9546 | 5915730 | Namlı Gurme | 208 | İstanbul | Kemankeş Karamustafa Paşa Mahallesi, Rıhtım... | Karaköy | Karaköy, İstanbul |
| 9547 | 5908749 | Ceviz Aca | 208 | İstanbul | Koşuyolu Mahallesi, Muhittin İstiklal Cadd... | Koşuyolu | Koşuyolu, İstanbul |
| 9548 | 5915807 | Huqqa | 208 | İstanbul | Kuruçeşme Mahallesi, Muallim Naci Caddesi, N... | Kuruçeşme | Kuruçeşme, İstanbul |
| 9549 | 5916112 | Afak Kahve | 208 | İstanbul | Kuruçeşme Mahallesi, Muallim Naci Caddesi, N... | Kuruçeşme | Kuruçeşme, İstanbul |
| 9550 | 5927402 | Walter's Coffee Roastery | 208 | İstanbul | Cafea Mahallesi, Bademaltı Sokak, No 21/B,... | Moda | Moda, İstanbul |

9551 rows × 21 columns

In [3]: *# Display basic information about the dataset*

```
zomato_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 21 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Restaurant ID                        9551 non-null   int64
1   Restaurant Name                      9551 non-null   object
2   Country Code                        9551 non-null   int64
3   City                                9551 non-null   object
4   Address                             9551 non-null   object
5   Locality                            9551 non-null   object
6   Locality Verbose                    9551 non-null   object
7   Longitude                           9551 non-null   float64
8   Latitude                            9551 non-null   float64
9   Cuisines                            9542 non-null   object
10  Average Cost for two                 9551 non-null   int64
11  Currency                            9551 non-null   object
12  Has Table booking                   9551 non-null   object
13  Has Online delivery                 9551 non-null   object
14  Is delivering now                   9551 non-null   object
15  Switch to order menu                9551 non-null   object
16  Price range                         9551 non-null   int64
17  Aggregate rating                    9551 non-null   float64
18  Rating color                        9551 non-null   object
19  Rating text                         9551 non-null   object
20  Votes                              9551 non-null   int64
dtypes: float64(3), int64(5), object(13)
memory usage: 1.5+ MB
```

In [4]: # 2

```
import pandas as pd

# Create a DataFrame of Boolean values indicating missing values
missing_values = zomato_df.isnull()

# Count the number of missing values for each column
missing_counts = missing_values.sum()

# Display the count of missing values for each column
print("Number of missing values for each column:")
print(missing_counts)
```

Number of missing values for each column:

| | |
|----------------------|---|
| Restaurant ID | 0 |
| Restaurant Name | 0 |
| Country Code | 0 |
| City | 0 |
| Address | 0 |
| Locality | 0 |
| Locality Verbose | 0 |
| Longitude | 0 |
| Latitude | 0 |
| Cuisines | 9 |
| Average Cost for two | 0 |
| Currency | 0 |
| Has Table booking | 0 |
| Has Online delivery | 0 |
| Is delivering now | 0 |
| Switch to order menu | 0 |
| Price range | 0 |
| Aggregate rating | 0 |
| Rating color | 0 |
| Rating text | 0 |
| Votes | 0 |

dtype: int64

In [5]: # 3

```
import pandas as pd

# describe() method to obtain summary statistics for numerical columns
summary_stats = zomato_df.describe()

# Display the summary statistics
print(summary_stats)
```

| | Restaurant ID | Country Code | Longitude | Latitude \ |
|-------|---------------|--------------|-------------|-------------|
| count | 9.551000e+03 | 9551.000000 | 9551.000000 | 9551.000000 |
| mean | 9.051128e+06 | 18.365616 | 64.126574 | 25.854381 |
| std | 8.791521e+06 | 56.750546 | 41.467058 | 11.007935 |
| min | 5.300000e+01 | 1.000000 | -157.948486 | -41.330428 |
| 25% | 3.019625e+05 | 1.000000 | 77.081343 | 28.478713 |
| 50% | 6.004089e+06 | 1.000000 | 77.191964 | 28.570469 |
| 75% | 1.835229e+07 | 1.000000 | 77.282006 | 28.642758 |
| max | 1.850065e+07 | 216.000000 | 174.832089 | 55.976980 |

| | Average Cost for two | Price range | Aggregate rating | Votes |
|-------|----------------------|-------------|------------------|--------------|
| count | 9551.000000 | 9551.000000 | 9551.000000 | 9551.000000 |
| mean | 1199.210763 | 1.804837 | 2.666370 | 156.909748 |
| std | 16121.183073 | 0.905609 | 1.516378 | 430.169145 |
| min | 0.000000 | 1.000000 | 0.000000 | 0.000000 |
| 25% | 250.000000 | 1.000000 | 2.500000 | 5.000000 |
| 50% | 400.000000 | 2.000000 | 3.200000 | 31.000000 |
| 75% | 700.000000 | 2.000000 | 3.700000 | 131.000000 |
| max | 800000.000000 | 4.000000 | 4.900000 | 10934.000000 |

In [6]: # 4

```
import pandas as pd

# Identify categorical columns using select_dtypes()
categorical_columns = zomato_df.select_dtypes(include=['object', 'category']).

# Display the list of categorical columns
print("Categorical columns in the Zomato dataset:")
print(categorical_columns)
```

```
Categorical columns in the Zomato dataset:
Index(['Restaurant Name', 'City', 'Address', 'Locality', 'Locality Verbose',
      'Cuisines', 'Currency', 'Has Table booking', 'Has Online delivery',
      'Is delivering now', 'Switch to order menu', 'Rating color',
      'Rating text'],
      dtype='object')
```

In [7]: # 5

```

import seaborn as sns
import matplotlib.pyplot as plt

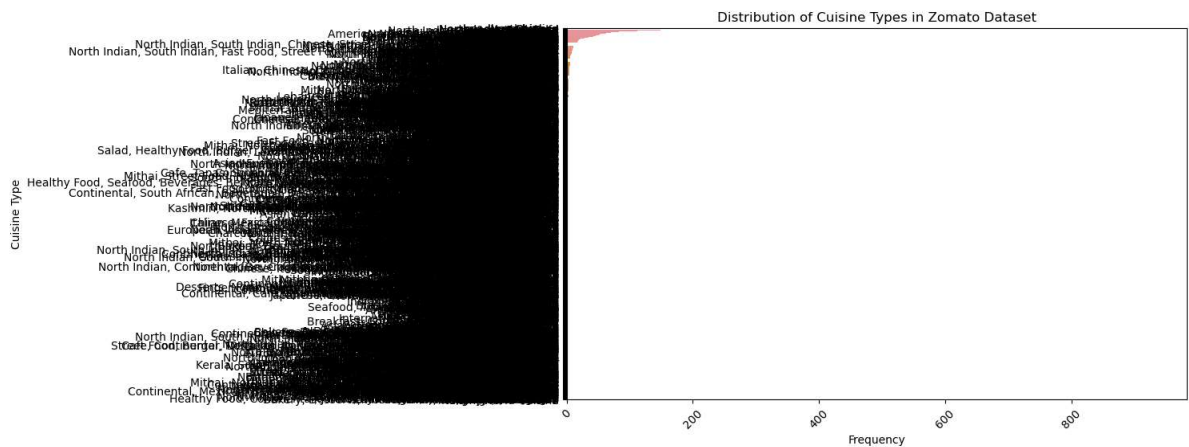
# Create a count plot using Seaborn's countplot() function
plt.figure(figsize=(10, 6)) # Adjust the figure size if needed
sns.countplot(y='Cuisines', data=zomato_df, order=zomato_df['Cuisines'].value_

# Add title and Labels
plt.title('Distribution of Cuisine Types in Zomato Dataset')
plt.xlabel('Frequency')
plt.ylabel('Cuisine Type')

# Rotate the y-axis Labels for better readability
plt.xticks(rotation=45)

# Show the plot
plt.show()

```

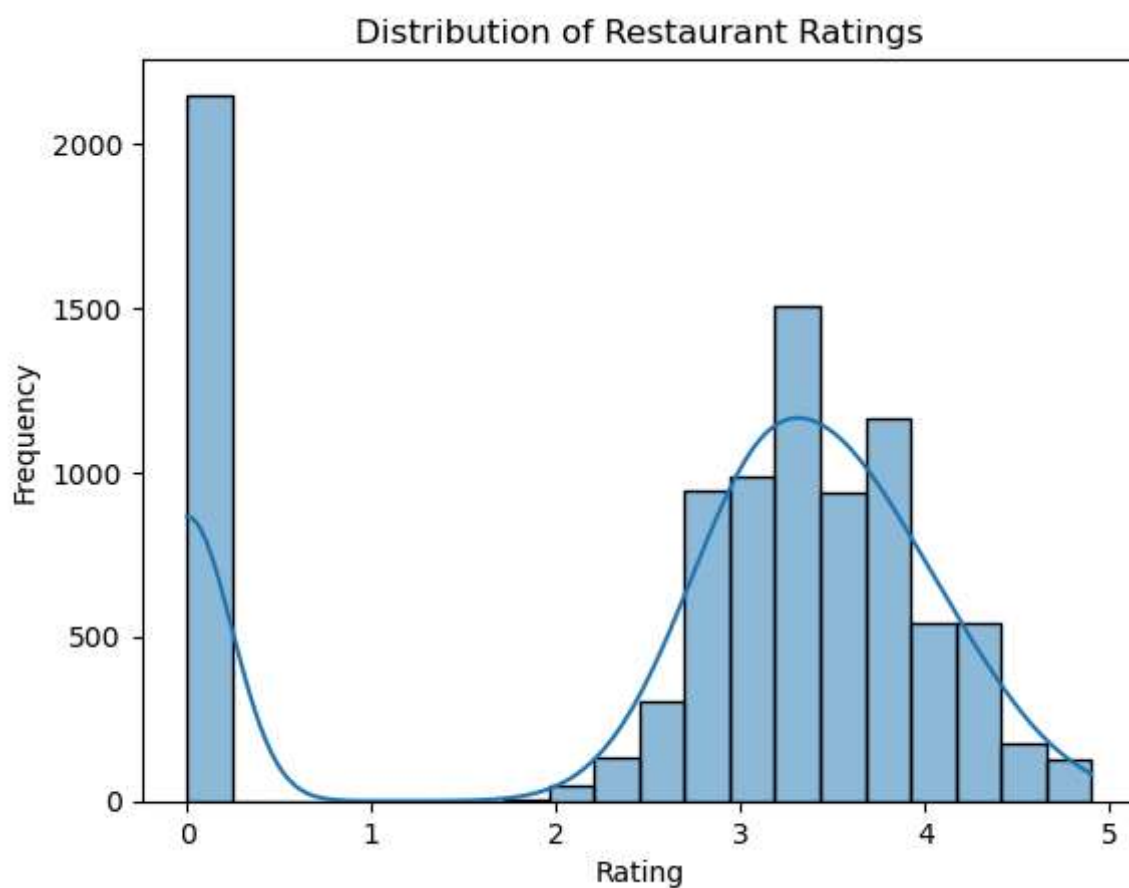


In [9]: # 6

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

zomato_df = pd.read_excel('zomato.xlsx')
zomato_df

sns.histplot(data = zomato_df, x='Aggregate rating', bins=20, kde=True)
plt.title('Distribution of Restaurant Ratings')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()
```



In [10]: # 7

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.scatterplot(data=zomato_df, x='City', y='Price range')
plt.title('Relationship between City and Price range')
plt.xlabel('City')
plt.ylabel('Price range')
plt.show()
```



In [11]: # 8

```

import seaborn as sns
import matplotlib.pyplot as plt

# Let's say 'City' is the column for cities and 'Aggregate rating' is the column for aggregate rating
plt.figure(figsize=(12, 6))
sns.barplot(data=zomato_df, x='City', y='Aggregate rating')
plt.title('Aggregate Rating of Restaurants in Different Cities')
plt.xlabel('City')
plt.ylabel('Aggregate Rating')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

```



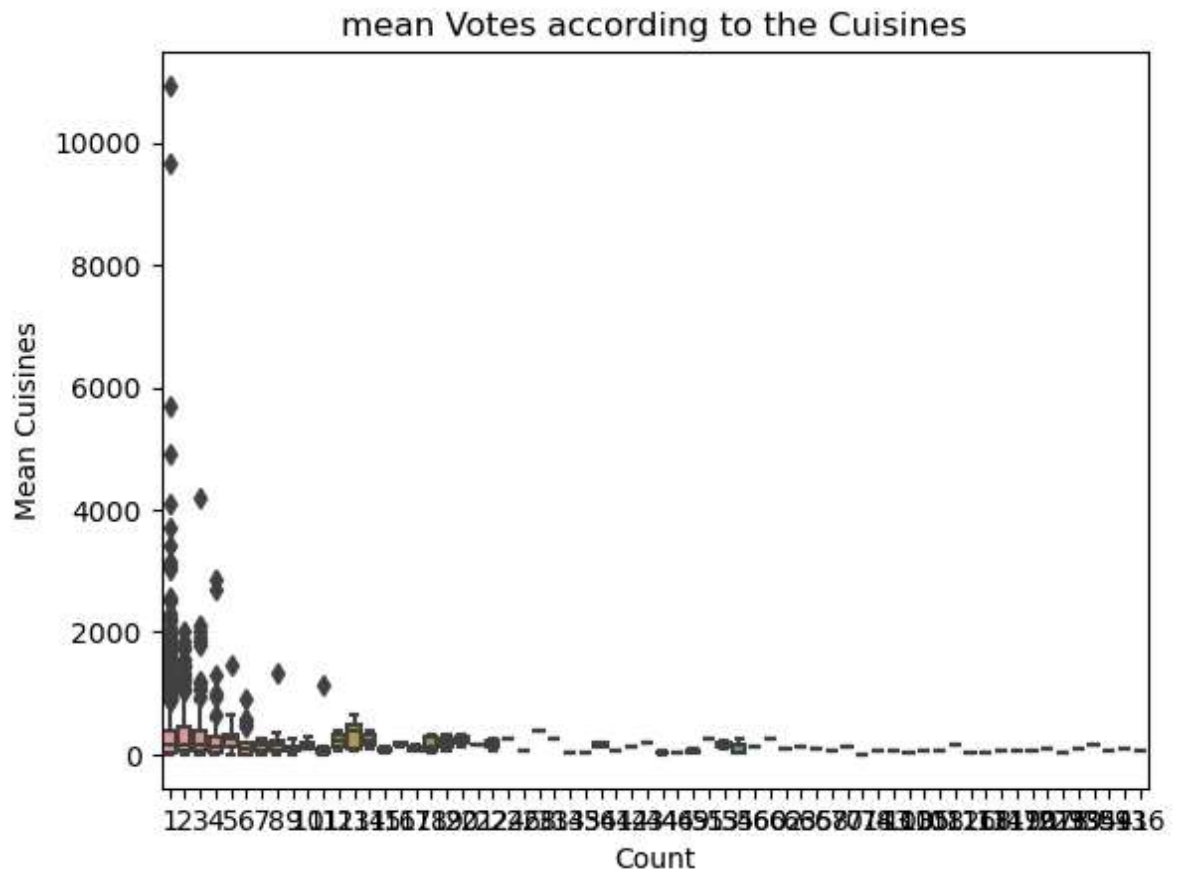
In [12]: # 9

```

df1=zomato_df.groupby('Cuisines')['Votes'].agg(['mean','count']).reset_index()
df1.columns=['Cuisines','mean Votes','count']
print(df1.head(1))
sns.boxplot(x='count',y='mean Votes',data=df1)
plt.title('mean Votes according to the Cuisines')
plt.xlabel('Count')
plt.ylabel('Mean Cuisines')
plt.show()

```

| | Cuisines | mean Votes | count |
|---|----------|------------|-------|
| 0 | Afghani | 9.75 | 4 |



In []: *#10 conclusion:*

From the count plot of cuisine types, it's clear which cuisines are most popular. Regional preferences can be deduced by cross-referencing cuisine types with city Rating Patterns:

The histogram of restaurant ratings might reveal that most restaurants fall within a certain range. If the ratings are heavily skewed towards the higher end, it might suggest that Cost vs. Rating Analysis:

The scatter plot between average cost for two and ratings can show if there's a correlation. A lack of strong correlation might indicate that restaurant pricing and quality are not directly related. City-Wise Performance:

The bar plot of average ratings by city helps to identify which cities have higher ratings. If certain cities have significantly lower ratings, it may point to regional issues or Data Quality and Integrity:

The analysis of missing values reveals columns that might have incomplete data. Handling missing data properly is crucial for maintaining the accuracy of further analysis.