*** *SECTION-2* ***

*** *ZOMATO DATA SET* ***

*** *Exploratory Data Analysis (EDA)* ***

Step 1: Import necessary libraries

- Pandas
- Matplotlib
- Seaborn

In [221…

```
  # Pandas is a powerful, open-source library in Python for data manipulation an

import pandas as pd

  # Matplotlib is a popular, open-source plotting library for Python, providing

import matplotlib.pyplot as plt

  # Seaborn is a Python data visualization library built on top of Matplotlib, p

import seaborn as sns
```

Step 2: Load Zomato data & Basic Information

In [224…

```
zomato_data = pd.read_csv('zomato_data.csv')   # Replace 'zomato_data.csv' with
print('\nLoading Zomato Data Set:\n')
zomato_data
```

```
Loading Zomato Data Set:
```

Out[224…

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Cen Pc Mal
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City	Littl Mal
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Edsa La, Mand C
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	M Mand N
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	M Mand N
...	...	...	...	...	...	...	
9546	5915730	NamlÛ± Gurme	208	ÛÁstanbul	Kemanke□ô Karamustafa Pa□ôa Mahallesi, RÛ±htÛ±...	Karakí_y	I ÛA
9547	5908749	Ceviz AÛôacÛ±	208	ÛÁstanbul	Ko□ôuyolu Mahallesi, Muhittin íìstí_ndaÛô Cadd...	Ko□ôuyolu	Ko□ ÛA
9548	5915807	Huqqa	208	ÛÁstanbul	Kuruí_e□ôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_e□ôme	Kuruí_ ÛA
9549	5916112	A□ô□ôk Kahve	208	ÛÁstanbul	Kuruí_e□ôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_e□ôme	Kuruí_ ÛA

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	'
9550	5927402	Walter's Coffee Roastery	208	ÛÁstanbul	CafeaÛôa Mahallesi, BademaltÛ± Sokak, No 21/B,...	Moda	ÛÁ

9551 rows × 21 columns

In [226…

```
print('\nBasic Information About Zomato Data Set:\n')
zomato_data.info()
   # Provides information about dataset shape, column data types, and missing va
```

```
Basic Information About Zomato Data Set:

<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
```

Step 3: Identify & Counting of missing values

In [229…

```
# Create a Boolean DataFrame with isnull()
missing_bool = zomato_data.isnull()
missing_bool
```

Out[229…

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	L
0	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
9546	False	False	False	False	False	False	False	False	
9547	False	False	False	False	False	False	False	False	
9548	False	False	False	False	False	False	False	False	
9549	False	False	False	False	False	False	False	False	
9550	False	False	False	False	False	False	False	False	

9551 rows × 21 columns

In [231…

```
# Count missing values for each column using sum()
missing_counts = missing_bool.sum()
missing_counts
```

Out[231…

```
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
```

Step 4: Summary Statistics For Numerical Columns

In [234…

```python
# Get summary statistics
summary_stats = zomato_data.describe()
summary_stats
```

Out[234…

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.00000
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.00000
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.00000
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.00000
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.00000
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.00000

In [ ]:

```python
# 1. Restaurant ID:
      # Range: IDs range from 53 to 18,500,650 (min to max), showing a wide vari
                # No insights can be directly drawn here other than confirming t
# 2. Country Code:
      # Mean: 18.37 indicates that the dataset predominantly consists of restaur
      # Min:    25%, 50%, and 75%: Country code is 1 (likely India) for at least
      # Max:    216 indicates the dataset includes global restaurants.
# 3. Longitude & Latitude:
      # Range:  Longitude ranges from -157.95 to 174.83 and Latitude from -41.33
      # Mean Latitude: ~25.85 and Longitude: ~64.13 suggest that most restaurant
# 4. Average Cost for Two:
      # Mean: ₹1,199 suggests that meals are moderately priced, but...
      # Standard Deviation (std): 16,121 shows high variability—some restaurants
      # Min: ₹0 indicates free meals (possibly promotional or missing data).
             # 75% Quartile: 75% of the restaurants have prices below ₹700, sugg
      # Max: ₹800,000 indicates extreme outliers, probably representing luxury d
# 5. Price Range (1-4):
      # Mean: ~1.8 suggests that most restaurants fall between cheap to moderate
               #  75% Quartile: 75% of restaurants are within the price range of
      # Max of 4: Some high-end restaurants exist, but they are a minority.
# 6. Aggregate Rating:
      # Mean: 2.67 is on the lower side, suggesting that restaurants may general
      # Median (50%): 3.2 shows that half the restaurants score above this ratin
      # Max: 4.9 is near-perfect, meaning at least a few restaurants have excell
# 7. Votes:
      # Mean: ~157 votes suggest that most restaurants attract moderate engageme
# Standard Deviation: 430 shows high variability—some restaurants are far more p
      # Quartile: 75% of restaurants have 131 votes or fewer, indicating that th
      # Max: One restaurant has 10,934 votes, suggesting it is extremely popular

# Key Insights:
# Geographic Concentration:
     # Most restaurants are likely located in India (Country Code = 1). However,

# Pricing Distribution:
    # The majority of restaurants offer meals for less than ₹700, indicating a f
```

```
# Customer Engagement:
    # While the average number of votes is 157, some restaurants receive thousan

# Ratings:
     # The overall average rating of 2.67 is relatively low, possibly indicating
# Price Range and Ratings Correlation:

      # With most restaurants having a price range of 1-2, it seems affordable a

# Outliers and Data Quality Issues:
      # Extreme values like a cost of ₹800,000 for two people or a restaurant wi
```

Step 5: Identify & Iterating of Categorical Columns

In [237…

```
# Identify Categorical Columns:
categorical_columns = zomato_data.select_dtypes(include=['object', 'category']).
categorical_columns
```

Out[237…

```
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 [239…

```
for col in categorical_columns:
    print(f"{col}: {zomato_data[col].nunique()} unique values")
    print(zomato_data[col].unique()[:3], "\n")  # Display first 5 unique values
```

```
Restaurant Name: 7446 unique values
['Le Petit Souffle' 'Izakaya Kikufuji' 'Heat - Edsa Shangri-La']

City: 141 unique values
['Makati City' 'Mandaluyong City' 'Pasay City']

Address: 8918 unique values
['Third Floor, Century City Mall, Kalayaan Avenue, Poblacion, Makati City'
 'Little Tokyo, 2277 Chino Roces Avenue, Legaspi Village, Makati City'
 'Edsa Shangri-La, 1 Garden Way, Ortigas, Mandaluyong City']

Locality: 1208 unique values
['Century City Mall, Poblacion, Makati City'
 'Little Tokyo, Legaspi Village, Makati City'
 'Edsa Shangri-La, Ortigas, Mandaluyong City']

Locality Verbose: 1265 unique values
['Century City Mall, Poblacion, Makati City, Makati City'
 'Little Tokyo, Legaspi Village, Makati City, Makati City'
 'Edsa Shangri-La, Ortigas, Mandaluyong City, Mandaluyong City']

Cuisines: 1825 unique values
['French, Japanese, Desserts' 'Japanese'
 'Seafood, Asian, Filipino, Indian']

Currency: 12 unique values
['Botswana Pula(P)' 'Brazilian Real(R$)' 'Dollar($)']

Has Table booking: 2 unique values
['Yes' 'No']

Has Online delivery: 2 unique values
['No' 'Yes']

Is delivering now: 2 unique values
['No' 'Yes']

Switch to order menu: 1 unique values
['No']

Rating color: 6 unique values
['Dark Green' 'Green' 'Yellow']

Rating text: 6 unique values
['Excellent' 'Very Good' 'Good']
```

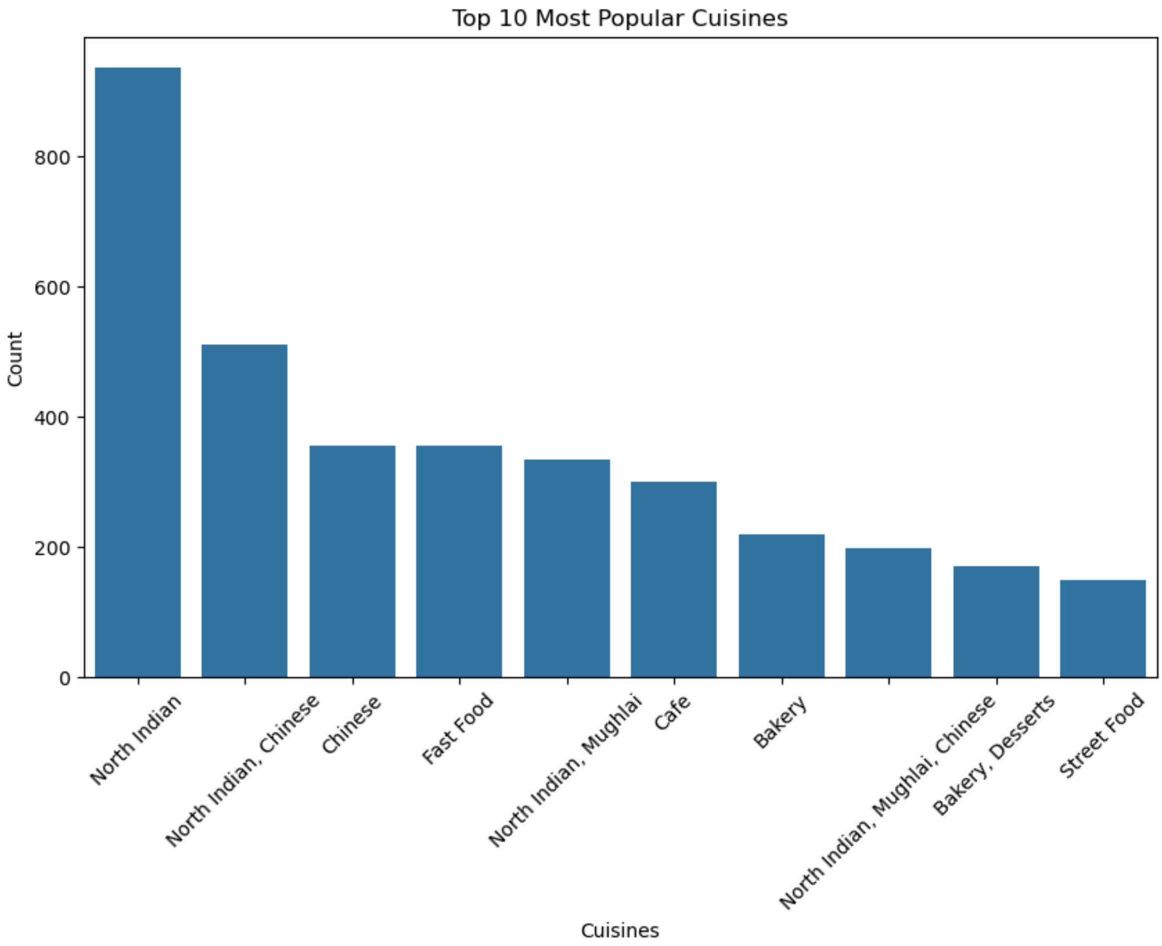
Step 6: Seaborn Plot to Visualizing the Distribution of Cuisine Types.

In [242…

```python
# Assuming 'zomato_data' is our DataFrame with a column named 'Cuisines'
cuisine_counts = zomato_data['Cuisines'].value_counts()

# Visualize the top 10 most frequent and popular cuisines
plt.figure(figsize=(10, 6))
sns.barplot(x=cuisine_counts.index[:10], y=cuisine_counts.values[:10])
plt.xlabel('Cuisines')
plt.ylabel('Count')
```

```python
plt.title('Top 10 Most Popular Cuisines')
plt.xticks(rotation=45)
plt.show()
# Observations
   # Most Popular Cuisines: The cuisines with the tallest bars are the most freq
   # Comparative Popularity: This bar plot also allows a quick comparison, showi
   # Trends: The types of cuisines in the top ranks can suggest culinary trends
   # This plot helps you quickly identify which cuisines are most popular and in
```



Step 7: Distribution of Restaurant Ratings using a Seaborn plot.

In [253…

```python
# Assuming 'zomato_data' is your DataFrame with a column named 'Aggregate rating
rating= zomato_data['Aggregate rating']
sns.histplot(rating , bins=100 ,kde=True)  # Adjust the number of bins as needed
plt.title('Distribution of Restaurant Ratings')
plt.xlabel('Aggregate rating')
plt.ylabel('Count')
plt.show()
# Observations
  # Spread of Ratings: The histogram reveals how ratings are spread across diffe
  # Frequency Peaks: Peaks indicate where ratings are most frequent. For instanc
  # Skewness: If the plot skews toward higher or lower ratings, this suggests a
  # The KDE curve in this plot adds a smooth line, showing the underlying distri
```

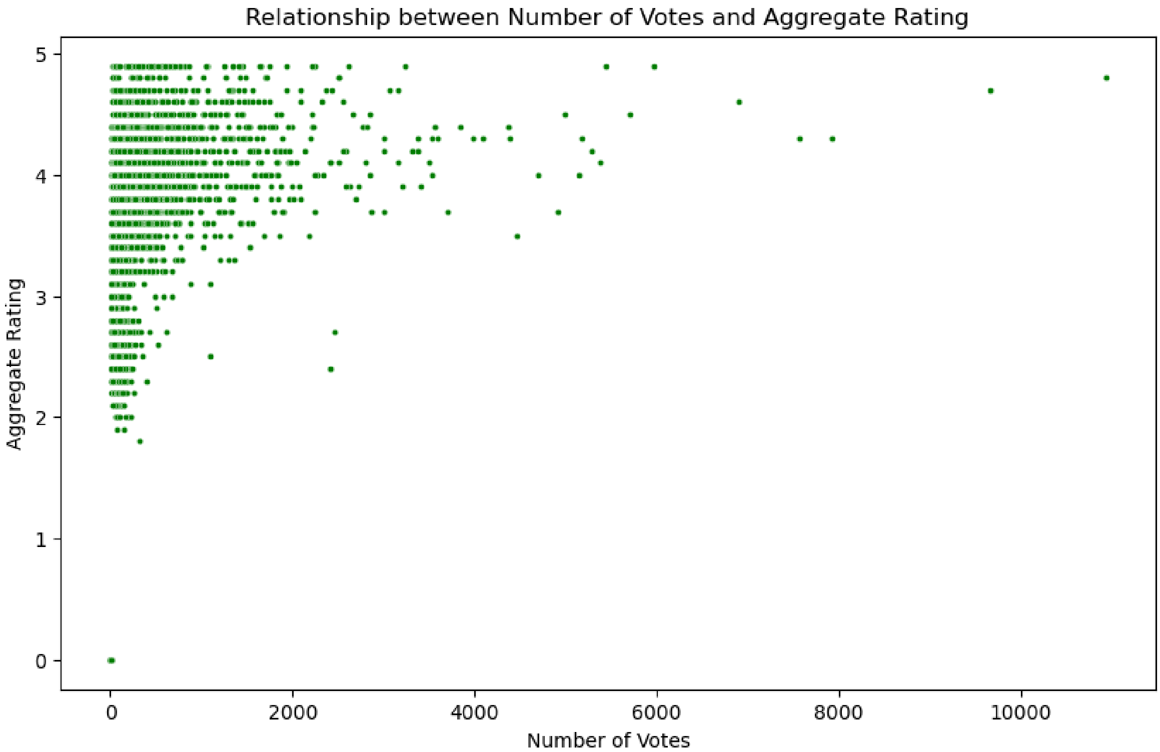
Step 8: Relationship Between Two Numerical Columns

In [247…

```
df = pd.DataFrame(zomato_data)
# Plotting the scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Votes', y='Aggregate rating', data=df, color='g', s=10, edgec
plt.xlabel("Number of Votes")
plt.ylabel("Aggregate Rating")
plt.title("Relationship between Number of Votes and Aggregate Rating")
plt.show()

# Observations
  # Positive or Negative Correlation: If data points form an upward trend from l
  # Cluster Patterns: Clusters of points around certain values may reveal segmen
  # Outliers: Points far from the main cluster may indicate unique cases, such a

# Interpretation
  # A scatter plot like this provides a visual representation of any relationshi
```


Relationship between Number of Votes and Aggregate Rating
Aggregate Rating
Number of Votes
0
1
2
3
4
5
0
2000
4000
6000
8000
10000