Cognifyz Technologies Task

In [1]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.10/site-
packages (3.6.2)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.10/sit
e-packages (from matplotlib) (0.11.0)
Requirement already satisfied: python-dateutil>=2.7 in /opt/conda/lib/python
3.10/site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.10/si
te-packages (from matplotlib) (9.2.0)
Requirement already satisfied: contourpy>=1.0.1 in /opt/conda/lib/python3.1
0/site-packages (from matplotlib) (1.0.6)
Requirement already satisfied: numpy>=1.19 in /opt/conda/lib/python3.10/site
-packages (from matplotlib) (1.23.5)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.1
0/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.1
0/site-packages (from matplotlib) (4.38.0)
Requirement already satisfied: pyparsing>=2.2.1 in /opt/conda/lib/python3.1
0/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.10/
site-packages (from matplotlib) (22.0)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-pa
ckages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in /opt/conda/lib/python3.10/site-pack
ages (1.5.2)
Requirement already satisfied: python-dateutil>=2.8.1 in /opt/conda/lib/pyth
on3.10/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/sit
e-packages (from pandas) (2022.6)
Requirement already satisfied: numpy>=1.21.0 in /opt/conda/lib/python3.10/si
te-packages (from pandas) (1.23.5)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-pa
ckages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```python
import urllib.request
url = 'https://raw.githubusercontent.com/Himanshu09311/Cognifyz-Technologies
urllib.request.urlretrieve(url, 'Restaurants_dataset.csv')

print("File downloaded successfully")
```

```
File downloaded successfully
```

In [4]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:
```python
Data=pd.read_csv("Restaurants_dataset.csv")
```

In [6]:
```python
Data
```

Out[6]:

| | Restaurant ID | Restaurant Name | Country Code | City | Address | Locality | Locality Ve |
|---|---|---|---|---|---|---|---|
| **0** | 6317637 | Le Petit Souffle | 162 | Makati City | Third Floor, Century City Mall, Kalayaan Avenu... | Century City Mall, Poblacion, Makati City | Centur Mall, Pobla Makat M |
| **1** | 6304287 | Izakaya Kikufuji | 162 | Makati City | Little Tokyo, 2277 Chino Roces Avenue, Legaspi... | Little Tokyo, Legaspi Village, Makati City | Little T Legaspi Vi Makati City, |
| **2** | 6300002 | Heat - Edsa Shangri-La | 162 | Mandaluyong City | Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal... | Edsa Shangri-La, Ortigas, Mandaluyong City | Edsa Shang Or Mandalu City, |
| **3** | 6318506 | Ooma | 162 | Mandaluyong City | Third Floor, Mega Fashion Hall, SM Megamall, O... | SM Megamall, Ortigas, Mandaluyong City | SM Mega Or Mandalu City, Mar |
| **4** | 6314302 | Sambo Kojin | 162 | Mandaluyong City | Third Floor, Mega Atrium, SM Megamall, Ortigas... | SM Megamall, Ortigas, Mandaluyong City | SM Mega Or Mandalu City, Mar |
| **...** | ... | ... | ... | ... | ... | ... | |
| **9546** | 5915730 | Namlı Gurme | 208 | ��stanbul | Kemanke�� Karamustafa Pa��a Mahallesi, Rıhtım ... | Karak�_y | Karak ��st |
| **9547** | 5908749 | Ceviz A��acı | 208 | ��stanbul | Ko��uyolu Mahallesi, Muhittin ��st�_nda�� Cadd... | Ko��uyolu | Ko��u ��st |
| **9548** | 5915807 | Huqqa | 208 | ��stanbul | Kuru�_e��me Mahallesi, Muallim Naci Caddesi, N... | Kuru�_e��me | Kuru�_e�� ��st |
| **9549** | 5916112 | A���k Kahve | 208 | ��stanbul | Kuru�_e��me Mahallesi, Muallim Naci Caddesi, N... | Kuru�_e��me | Kuru�_e�� ��st |
| **9550** | 5927402 | Walter's Coffee Roastery | 208 | ��stanbul | Cafea��a Mahallesi, Bademaltı Sokak, No 21/B, ... | Moda | M ��st |

9551 rows × 21 columns

```
In [7]: Data.head()
```

Out[7]:

| | Restaurant ID | Restaurant Name | Country Code | City | Address | Locality | Locality Verbose | Longitud |
|---|---|---|---|---|---|---|---|---|
| 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... | 121.02753 |
| 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... | 121.01410 |
| 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... | 121.05683 |
| 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... | 121.05647 |
| 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... | 121.05750 |

5 rows × 21 columns

```
In [8]: Data.tail()
```

Out[8]:

| | Restaurant ID | Restaurant Name | Country Code | City | Address | Locality | Locality Verb |
|---|---|---|---|---|---|---|---|
| **9546** | 5915730 | Namlı Gurme | 208 | ��stanbul | Kemanke�� Karamustafa Pa��a Mahallesi, Rıhtım ... | Karak�_y | Karak� ��stan |
| **9547** | 5908749 | Ceviz A��acı | 208 | ��stanbul | Ko��uyolu Mahallesi, Muhittin ��st�_nda�� Cadd... | Ko��uyolu | Ko��uy ��stan |
| **9548** | 5915807 | Huqqa | 208 | ��stanbul | Kuru�_e��me Mahallesi, Muallim Naci Caddesi, N... | Kuru�_e��me | Kuru�_e�� ��stan |
| **9549** | 5916112 | A���k Kahve | 208 | ��stanbul | Kuru�_e��me Mahallesi, Muallim Naci Caddesi, N... | Kuru�_e��me | Kuru�_e�� ��stan |
| **9550** | 5927402 | Walter's Coffee Roastery | 208 | ��stanbul | Cafea��a Mahallesi, Bademaltı Sokak, No 21/B, ... | Moda | Mc ��stan |

5 rows × 21 columns

In [9]:
```
Data.describe()
```

Out[9]:

| | Restaurant ID | Country Code | Longitude | Latitude | Average Cost for two | Price range | Agg |
|---|---|---|---|---|---|---|---|
| **count** | 9.551000e+03 | 9551.000000 | 9551.000000 | 9551.000000 | 9551.000000 | 9551.000000 | 9551.0 |
| **mean** | 9.051128e+06 | 18.365616 | 64.126574 | 25.854381 | 1199.210763 | 1.804837 | 2.6 |
| **std** | 8.791521e+06 | 56.750546 | 41.467058 | 11.007935 | 16121.183073 | 0.905609 | 1.5 |
| **min** | 5.300000e+01 | 1.000000 | -157.948486 | -41.330428 | 0.000000 | 1.000000 | 0.0 |
| **25%** | 3.019625e+05 | 1.000000 | 77.081343 | 28.478713 | 250.000000 | 1.000000 | 2.5 |
| **50%** | 6.004089e+06 | 1.000000 | 77.191964 | 28.570469 | 400.000000 | 2.000000 | 3.2 |
| **75%** | 1.835229e+07 | 1.000000 | 77.282006 | 28.642758 | 700.000000 | 2.000000 | 3.7 |
| **max** | 1.850065e+07 | 216.000000 | 174.832089 | 55.976980 | 800000.000000 | 4.000000 | 4.9 |

In [10]:
```
Data.isnull().sum()
```

Out[10]:
```
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 [11]:
```
Data['Cuisines'].fillna(Data['Cuisines'].mode()[0], inplace=True)
```

In [12]:
```
Data.isnull().sum()
```

Out[12]:
```
Restaurant ID           0
Restaurant Name         0
Country Code            0
City                    0
Address                 0
Locality                0
Locality Verbose        0
Longitude               0
Latitude                0
Cuisines                0
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 [13]:
```
Data.columns
```

```
Out[13]: Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Addres
s',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes'],
      dtype='object')
```

```
In [14]: Data.shape
```

```
Out[14]: (9551, 21)
```

```
In [15]: Data.head()
```

Out[15]:

| | Restaurant ID | Restaurant Name | Country Code | City | Address | Locality | Locality Verbose | Longitude |
|---|---|---|---|---|---|---|---|---|
| **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... | 121.02753 |
| **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... | 121.01410 |
| **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... | 121.05683 |
| **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... | 121.05647 |
| **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... | 121.05750 |

5 rows × 21 columns

# Level 1

Task 1

Task: Top Cuisines Determine the top three most common cuisines in the dataset. Calculate the percentage of restaurants that serve each of the top cuisines.

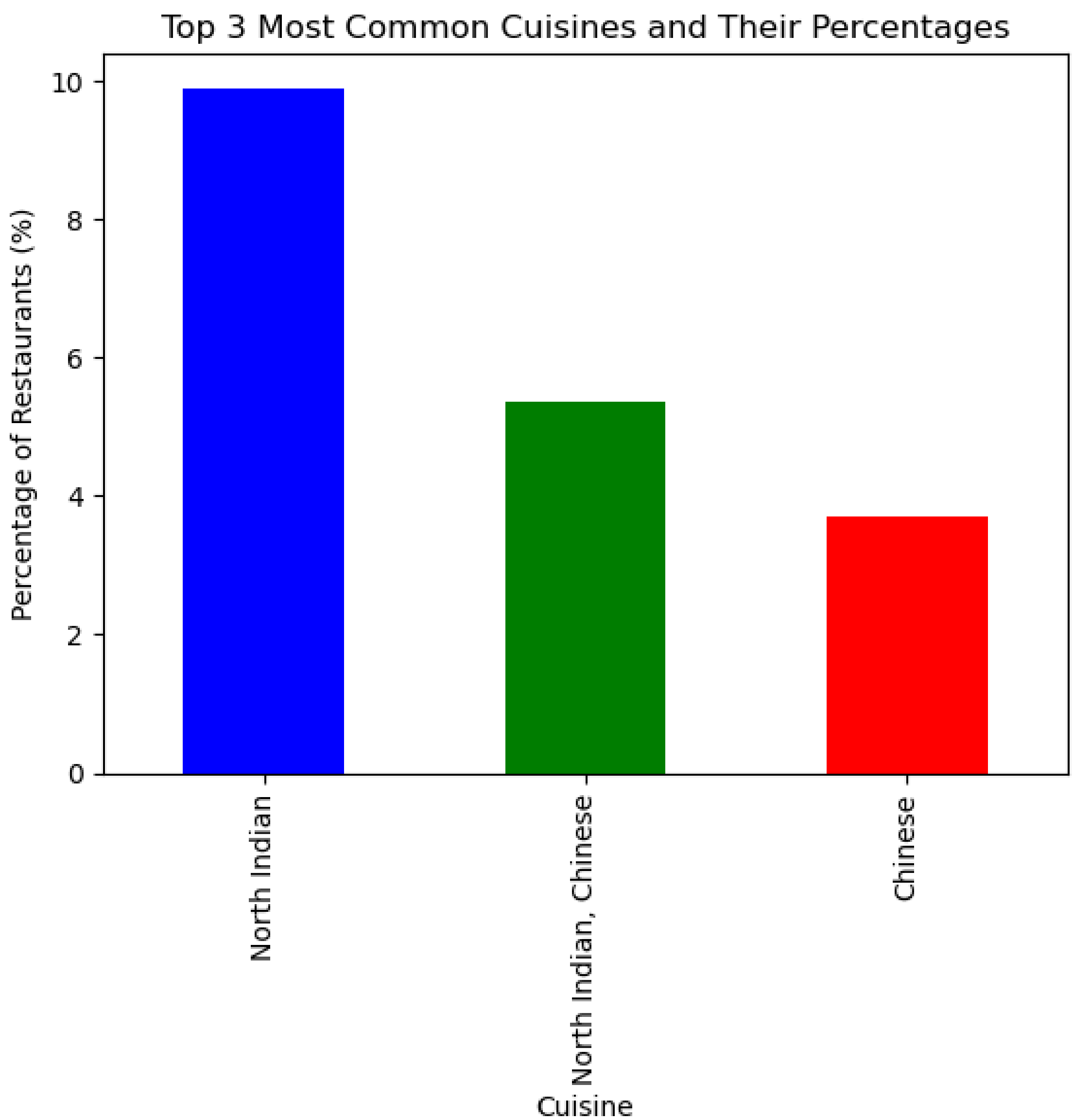
In [16]:

```
Cuisines_counts = Data['Cuisines'].value_counts()

Total_restaurants = len(Data)

Top_Cuisines = Cuisines_counts.head(3)

Top_Cuisines_percentage = (Top_Cuisines / Total_restaurants) * 100

print("Top 3 Cuisines:")
print(Top_Cuisines)
print("\nPercentages:")
print(Top_Cuisines_percentage)
```

```
Top 3 Cuisines:
North Indian             945
North Indian, Chinese    511
Chinese                  354
Name: Cuisines, dtype: int64

Percentages:
North Indian             9.894252
North Indian, Chinese    5.350225
Chinese                  3.706418
Name: Cuisines, dtype: float64
```

In [17]:

```
fig, ax = plt.subplots()
Top_Cuisines_percentage.plot(kind='bar', ax=ax, color=['blue', 'green', 'red

ax.set_ylabel('Percentage of Restaurants (%)')
ax.set_xlabel('Cuisine')
ax.set_title('Top 3 Most Common Cuisines and Their Percentages')

plt.show()
```




# Level 1

Task 2

Task: City Analysis Identify the city with the highest number of restaurants in the dataset. Calculate the average rating for restaurants in each city. Determine the city with the highest average rating.

In [18]:

```
restaurant_counts = Data['City'].value_counts()

city_with_highest_restaurants = restaurant_counts.idxmax()

highest_restaurant_count = restaurant_counts.max()

print(f"The city with the highest number of restaurants is {city_with_highes
```

```
The city with the highest number of restaurants is New Delhi with 5473 resta
urants.
```

In [19]:
```python
average_ratings = Data.groupby('City')['Aggregate rating'].mean()

print("Average ratings for restaurants in each city:", average_ratings)
```

```
Average ratings for restaurants in each city: City
Abu Dhabi          4.300000
Agra               3.965000
Ahmedabad          4.161905
Albany             3.555000
Allahabad          3.395000
                     ...
Weirton            3.900000
Wellington City    4.250000
Winchester Bay     3.200000
Yorkton            3.300000
��stanbul          4.292857
Name: Aggregate rating, Length: 141, dtype: float64
```

In [20]:
```python
average_ratings = Data.groupby('City')['Aggregate rating'].mean()

city_with_highest_rating = average_ratings.idxmax()

highest_average_rating = average_ratings.max()

print(f"The city with the highest average rating is {city_with_highest_ratin
```

```
The city with the highest average rating is Inner City with an average ratin
g of 4.90.
```

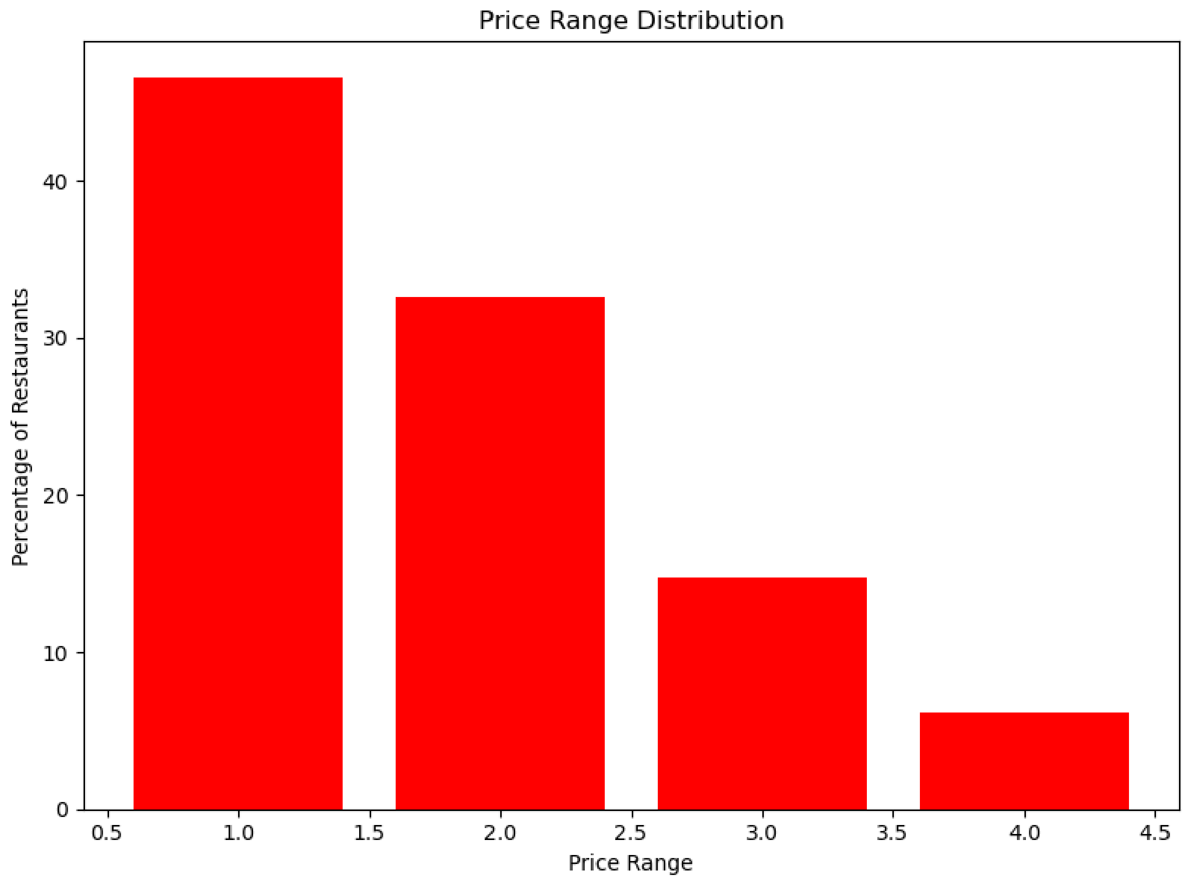
# Level 1

Task 3

Task: Price Range Distribution Create a histogram or bar chart to visualize the distribution of price ranges among the restaurants. Calculate the percentage of restaurants in each price range category.

In [21]:
```python
price_range_counts = Data['Price range'].value_counts()

total_restaurants = len(Data)

percentage_price_range = (price_range_counts / total_restaurants) * 100

plt.figure(figsize=(8, 6))
plt.bar(percentage_price_range.index, percentage_price_range, color='Red')
plt.title('Price Range Distribution')
plt.xlabel('Price Range')
plt.ylabel('Percentage of Restaurants')
plt.xticks(rotation=0)
```

```
plt.tight_layout()
plt.show()

print("Percentage of restaurants in each price range category:")
print(percentage_price_range)
```




```
Percentage of restaurants in each price range category:
1    46.529159
2    32.593446
3    14.741912
4     6.135483
Name: Price range, dtype: float64
```

# Level 1

Task 4

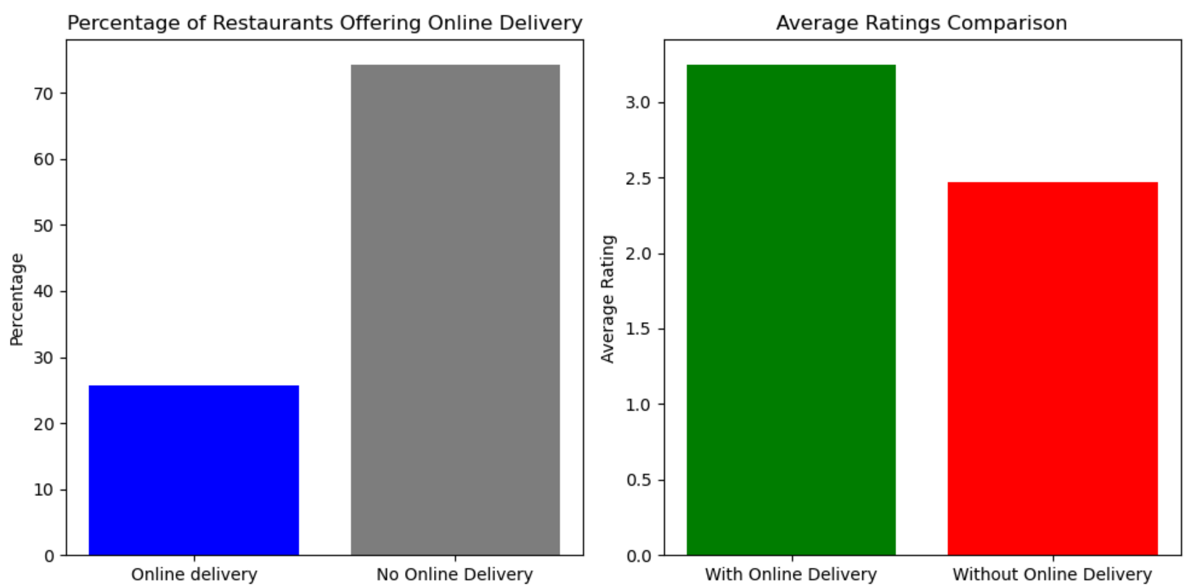
Task: Online Delivery Determine the percentage of restaurants that offer online delivery. Compare the average ratings of restaurants with and without online delivery.

In [22]:
```
total_restaurants = len(Data)
online_delivery_count = Data['Has Online delivery'].value_counts().get('Yes'
online_delivery_percentage = (online_delivery_count / total_restaurants) * 1

average_rating_with_delivery = Data[Data['Has Online delivery'] == 'Yes']['A
average_rating_without_delivery = Data[Data['Has Online delivery'] == 'No']['
```

```
print(f"Percentage of restaurants that offer online delivery: {online_delive
print(f"Average rating of restaurants with online delivery: {average_rating_
print(f"Average rating of restaurants without online delivery: {average_rati
```

```
Percentage of restaurants that offer online delivery: 25.66%
Average rating of restaurants with online delivery: 3.25
Average rating of restaurants without online delivery: 2.47
```

In [23]:
```
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.bar(['Online delivery', 'No Online Delivery'], [online_delivery_percenta
plt.ylabel('Percentage')
plt.title('Percentage of Restaurants Offering Online Delivery')

plt.subplot(1, 2, 2)
plt.bar(['With Online Delivery', 'Without Online Delivery'], [average_rating
plt.ylabel('Average Rating')
plt.title('Average Ratings Comparison')
plt.tight_layout()
plt.show()
```




# Level 2

Task 1

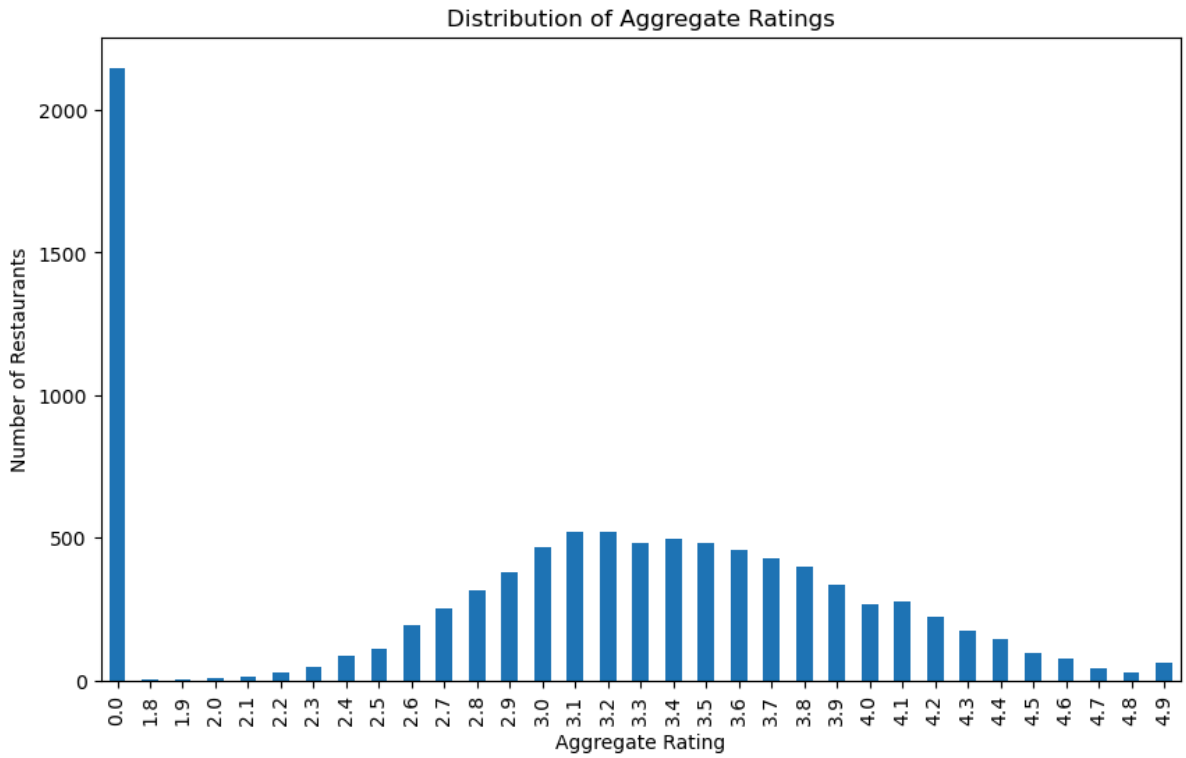
Task: Restaurant Ratings

Analyze the distribution of aggregate ratings and determine the most common rating range. Calculate the average number of votes received by restaurants.

In [24]:
```
rating_counts = Data['Aggregate rating'].value_counts().sort_index()

most_common_rating_range = rating_counts.idxmax()
```

```
most_common_rating_count = rating_counts.max()

average_votes = Data['Votes'].mean()

print(f"The most common rating range is: {most_common_rating_range} with {mo
print(f"The average number of votes received by restaurants is: {average_vot

plt.figure(figsize=(10, 6))
rating_counts.plot(kind='bar')
plt.xlabel('Aggregate Rating')
plt.ylabel('Number of Restaurants')
plt.title('Distribution of Aggregate Ratings')
plt.show()
```

```
The most common rating range is: 0.0 with 2148 occurrences.
The average number of votes received by restaurants is: 156.91
```




# Level 2

Task 2

Task: Cuisine Combination

Identify the most common combinations of cuisines in the dataset. Determine if certain cuisine combinations tend to have higher ratings.

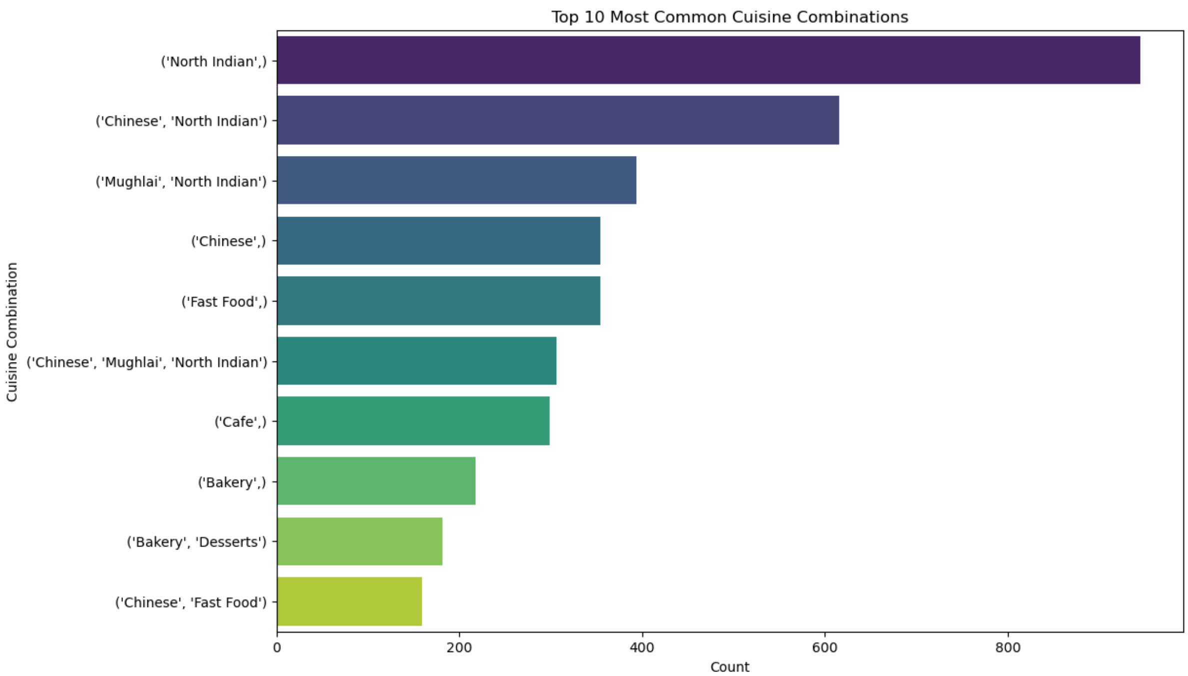
In [25]:
```
pip install counter
```

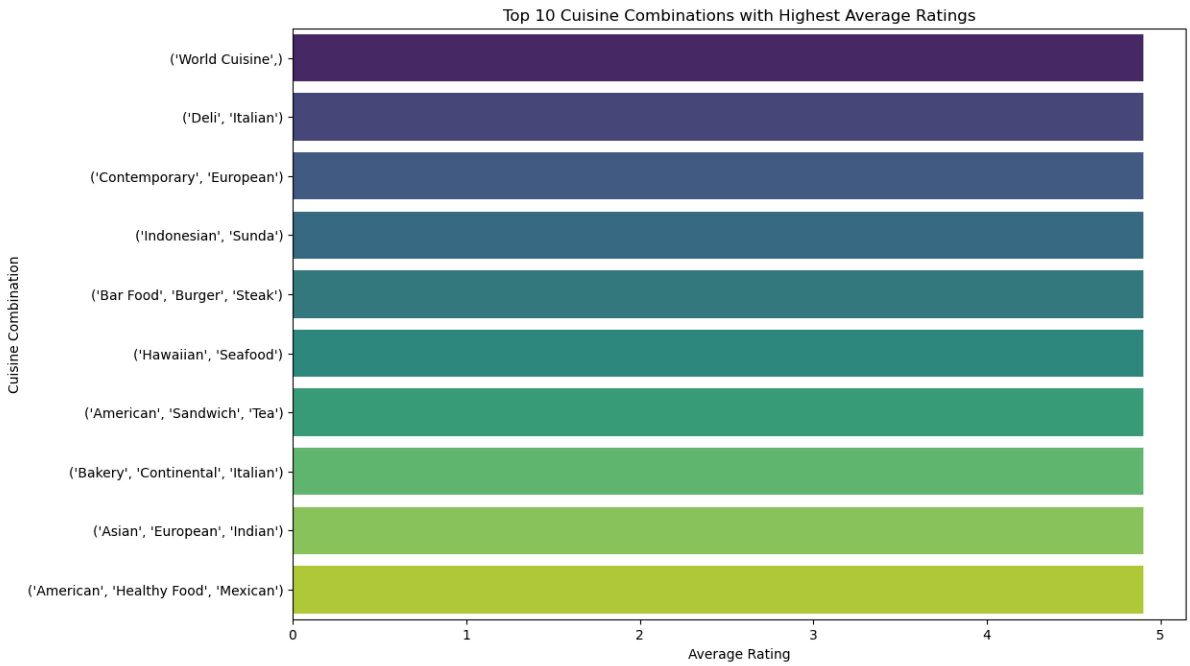
```
Requirement already satisfied: counter in /opt/conda/lib/python3.10/site-pac
kages (1.0.0)
Note: you may need to restart the kernel to use updated packages.
```

In [26]:

```python
from collections import Counter
Data['Cuisines'] = Data['Cuisines'].str.split(', ')

cuisine_combinations = Data['Cuisines'].apply(lambda x: tuple(sorted(x)))
combination_counts = Counter(cuisine_combinations)
most_common_combinations = combination_counts.most_common(10)
print("Most common cuisine combinations:")
for combo, count in most_common_combinations:
    print(f"{combo}: {count}")

Data['cuisine_combination'] = Data['Cuisines'].apply(lambda x: tuple(sorted(
combination_ratings = Data.groupby('cuisine_combination')['Aggregate rating'
combination_ratings.columns = ['cuisine_combination', 'average_rating']

highest_rated_combinations = combination_ratings.sort_values(by='average_rat
print("\nCuisine combinations with the highest average ratings:")
print(highest_rated_combinations)

common_combinations_df = pd.DataFrame(most_common_combinations, columns=['cu
common_combinations_df = common_combinations_df.merge(combination_ratings, o

plt.figure(figsize=(12, 8))
sns.barplot(x='count', y='cuisine_combination', data=common_combinations_df,
plt.title('Top 10 Most Common Cuisine Combinations')
plt.xlabel('Count')
plt.ylabel('Cuisine Combination')
plt.show()

plt.figure(figsize=(12, 8))
sns.barplot(x='average_rating', y='cuisine_combination', data=highest_rated_
plt.title('Top 10 Cuisine Combinations with Highest Average Ratings')
plt.xlabel('Average Rating')
plt.ylabel('Cuisine Combination')
plt.show()
```

```
Most common cuisine combinations:
('North Indian',): 945
('Chinese', 'North Indian'): 616
('Mughlai', 'North Indian'): 394
('Chinese',): 354
('Fast Food',): 354
('Chinese', 'Mughlai', 'North Indian'): 306
('Cafe',): 299
('Bakery',): 218
('Bakery', 'Desserts'): 181
('Chinese', 'Fast Food'): 159

Cuisine combinations with the highest average ratings:
                    cuisine_combination  average_rating
1342                     (World Cuisine,)             4.9
996                        (Deli, Italian)             4.9
908            (Contemporary, European)             4.9
1166                (Indonesian, Sunda)             4.9
412              (Bar Food, Burger, Steak)             4.9
1131                 (Hawaiian, Seafood)             4.9
158            (American, Sandwich, Tea)             4.9
377       (Bakery, Continental, Italian)             4.9
265            (Asian, European, Indian)             4.9
132    (American, Healthy Food, Mexican)             4.9
```






# Level 2

Task 3

Task: Geographic Analysis

Plot the locations of restaurants on a map using longitude and latitude coordinates. Identify any patterns or clusters of restaurants in specific areas.

In [27]:
```
pip install folium
```

```
Requirement already satisfied: folium in /opt/conda/lib/python3.10/site-packages (0.16.0)
Requirement already satisfied: requests in /opt/conda/lib/python3.10/site-packages (from folium) (2.28.1)
Requirement already satisfied: xyzservices in /opt/conda/lib/python3.10/site-packages (from folium) (2022.9.0)
Requirement already satisfied: branca>=0.6.0 in /opt/conda/lib/python3.10/site-packages (from folium) (0.7.2)
Requirement already satisfied: jinja2>=2.9 in /opt/conda/lib/python3.10/site-packages (from folium) (3.1.2)
Requirement already satisfied: numpy in /opt/conda/lib/python3.10/site-packages (from folium) (1.23.5)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.10/site-packages (from jinja2>=2.9->folium) (2.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/lib/python3.10/site-packages (from requests->folium) (1.26.13)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/site-packages (from requests->folium) (2022.12.7)
Requirement already satisfied: charset-normalizer<3,>=2 in /opt/conda/lib/python3.10/site-packages (from requests->folium) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-packages (from requests->folium) (3.4)
Note: you may need to restart the kernel to use updated packages.
```

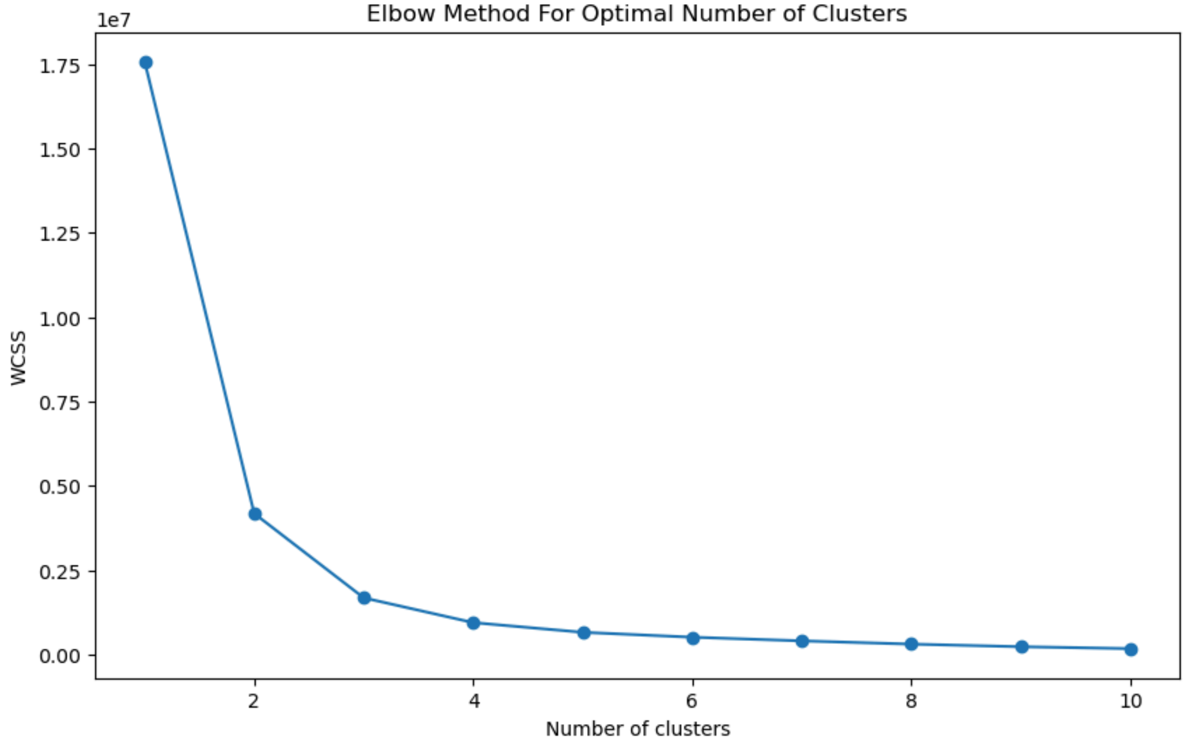
In [28]:

```python
import folium

if 'Latitude' not in Data.columns or 'Longitude' not in Data.columns:
    raise ValueError("The dataset must contain 'latitude' and 'longitude' co

map_center = [Data['Latitude'].mean(), Data['Longitude'].mean()]
mymap = folium.Map(location=map_center, zoom_start=12)

for _, row in Data.iterrows():
    folium.Marker(
        location=[row['Latitude'], row['Longitude']],
        popup=row['name'] if 'name' in row else None ).add_to(mymap)
mymap.save('restaurants_map.html')
# Display the map in a Jupyter Notebook (if you are using one)
#mymap  # Uncomment this line if running in a Jupyter Notebook
```

In [29]:

```python
import pandas as pd
import folium
from sklearn.cluster import KMeans
import seaborn as sns
import matplotlib.pyplot as plt

if 'Latitude' not in Data.columns or 'Longitude' not in Data.columns:
    raise ValueError("The dataset must contain 'latitude' and 'longitude' co

map_center = [Data['Latitude'].mean(), Data['Longitude'].mean()]
mymap = folium.Map(location=map_center, zoom_start=12)

for _, row in Data.iterrows():
    folium.Marker(
        location=[row['Latitude'], row['Longitude']],
        popup=row['name'] if 'name' in row else None).add_to(mymap)
```

```
mymap.save('restaurants_map.html')

coordinates = Data[['Latitude', 'Longitude']]

wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10,
    kmeans.fit(coordinates)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Elbow Method For Optimal Number of Clusters')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

optimal_clusters = 4
kmeans = KMeans(n_clusters=optimal_clusters, init='k-means++', max_iter=300,
Data['cluster'] = kmeans.fit_predict(coordinates)

cluster_map = folium.Map(location=map_center, zoom_start=12)
colors = ['red', 'blue', 'green', 'purple', 'orange', 'darkred', 'lightred',
for i in range(optimal_clusters):
    cluster_data = Data[Data['cluster'] == i]
    for _, row in cluster_data.iterrows():
        folium.CircleMarker(
            location=[row['Latitude'], row['Longitude']],
            radius=5,
            color=colors[i % len(colors)],
            fill=True,
            fill_color=colors[i % len(colors)],
            fill_opacity=0.6,
            popup=row['name'] if 'name' in row else None ).add_to(cluster_ma

cluster_map.save('restaurants_cluster_map.html')

heatmap_data = Data[['Latitude', 'Longitude']]
sns.kdeplot(x=heatmap_data['Longitude'], y=heatmap_data['Latitude'], cmap="v
plt.title('Restaurant Density Heatmap')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.show()
```



Elbow Method For Optimal Number of Clusters
1e7
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
WCSS
2
4
6
8
10
Number of clusters


Restaurant Density Heatmap
60
40
20
0
−20
−40
Latitude
−150
−100
−50
0
50
100
150
200
Longitude


# Level 2

Task 4

Task: Restaurant Chains

Identify if there are any restaurant chains present in the dataset. Analyze the ratings and popularity of different restaurant chains.

In [30]:
```
restaurant_counts = Data['Restaurant Name'].value_counts()
chains = restaurant_counts[restaurant_counts > 1].index.tolist()
chain_df = Data[Data['Restaurant Name'].isin(chains)]
chain_stats = chain_df.groupby('Restaurant Name').agg(
    average_rating=pd.NamedAgg(column='Aggregate rating', aggfunc='mean'),
    popularity=pd.NamedAgg(column='Aggregate rating', aggfunc='count')
).reset_index()
chain_stats = chain_stats.sort_values(by='popularity', ascending=False)
print(chain_stats)

# Save the results to a CSV file
chain_stats.to_csv('chain_stats.csv', index=False)
```

```
      Restaurant Name  average_rating  popularity
122    Cafe Coffee Day        2.419277          83
220     Domino's Pizza        2.740506          79
618             Subway        2.907937          63
281    Green Chick Chop       2.672549          51
408         McDonald's        3.339583          48
..                ...              ...         ...
289           Gullu's         3.000000           2
288             Gulab         2.950000           2
287      Grover Sweets        1.550000           2
286            Grillz         2.350000           2
733            bu��no         3.750000           2

[734 rows x 3 columns]
```

# Level 3

Task 1

Task: Restaurant Reviews

Analyze the text reviews to identify the most common positive and negative keywords.

Calculate the average length of reviews and explore if there is a relationship between review length and rating.

In [31]:
```
pip install nltk
```

```
Requirement already satisfied: nltk in /opt/conda/lib/python3.10/site-packag
es (3.8.1)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.10/site-packag
es (from nltk) (4.64.1)
Requirement already satisfied: joblib in /opt/conda/lib/python3.10/site-pack
ages (from nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in /opt/conda/lib/python3.10/
site-packages (from nltk) (2024.5.15)
Requirement already satisfied: click in /opt/conda/lib/python3.10/site-packa
ges (from nltk) (8.1.3)
Note: you may need to restart the kernel to use updated packages.
```

In [32]:

```
Data['Rating text'].unique
```

Out[32]:

```
<bound method Series.unique of 0       Excellent
1       Excellent
2       Very Good
3       Excellent
4       Excellent
          ...
9546    Very Good
9547    Very Good
9548         Good
9549    Very Good
9550    Very Good
Name: Rating text, Length: 9551, dtype: object>
```

In [33]:

```
import pandas as pd
import string
from collections import Counter
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import nltk

nltk.download('vader_lexicon')
nltk.download('punkt')
nltk.download('stopwords')

def preprocess_text(text):
    text = text.lower()
    text = text.translate(str.maketrans('', '', string.punctuation))
    words = word_tokenize(text)
    words = [word for word in words if word not in stopwords.words('english'
    return words
Data['processed_reviews'] = Data['Rating text'].apply(preprocess_text)

sid = SentimentIntensityAnalyzer()

def get_sentiment_words(words):
    pos_words = []
    neg_words = []
    for word in words:
        if sid.polarity_scores(word)['compound'] > 0:  # Positive word
            pos_words.append(word)
        elif sid.polarity_scores(word)['compound'] < 0:  # Negative word
```

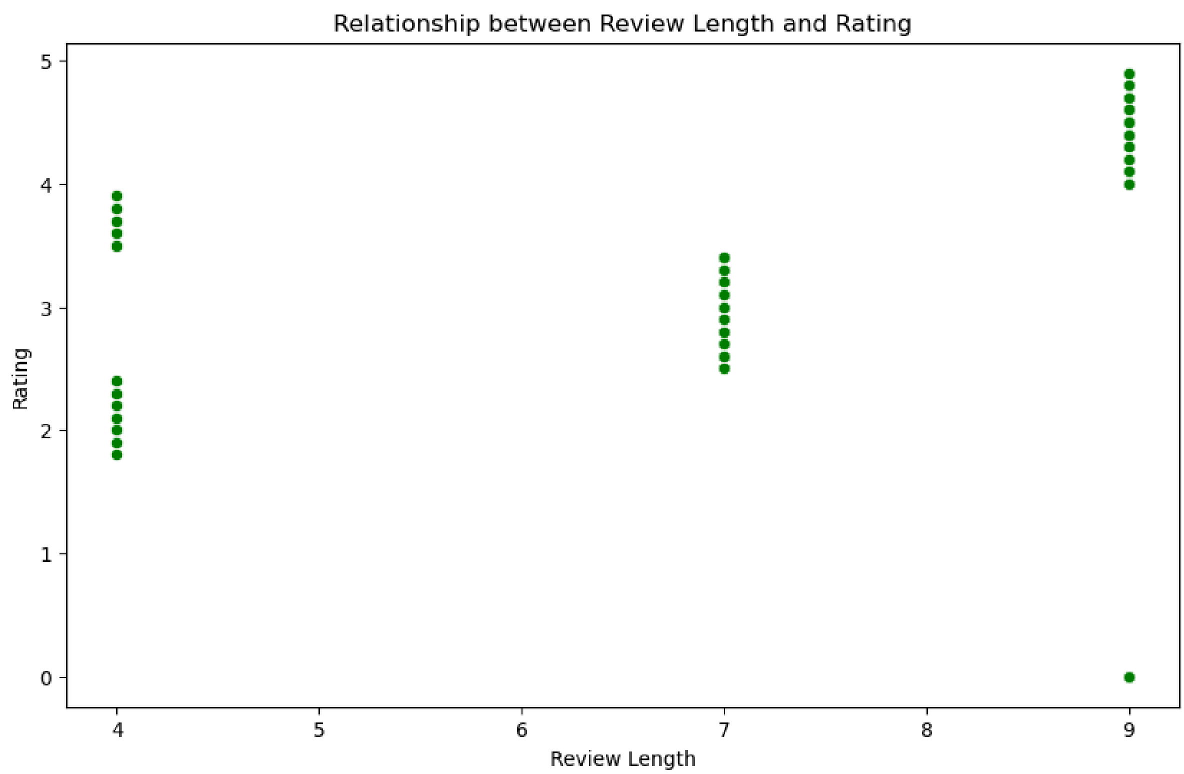
```
            neg_words.append(word)
    return pos_words, neg_words
Data['pos_words'], Data['neg_words'] = zip(*Data['processed_reviews'].apply(

pos_words = Counter([word for words in Data['pos_words'] for word in words])
neg_words = Counter([word for words in Data['neg_words'] for word in words])

most_common_pos = pos_words.most_common(10)
most_common_neg = neg_words.most_common(10)
print("Most common positive words:", most_common_pos)
print("Most common negative words:", most_common_neg)
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     /home/jovyan/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
[nltk_data] Downloading package punkt to /home/jovyan/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /home/jovyan/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Most common positive words: [('good', 3179), ('excellent', 301)]
Most common negative words: [('poor', 186)]
```

In [34]:

```
Data['review_length'] = Data['Rating text'].apply(len)

average_length = Data['review_length'].mean()
print("Average review length:", average_length)

correlation = Data[['review_length', 'Aggregate rating']].corr().iloc[0, 1]
print("Correlation between review length and rating:", correlation)

plt.figure(figsize=(10, 6))
sns.scatterplot(x='review_length', y='Aggregate rating', data=Data,color="Gr
plt.title('Relationship between Review Length and Rating')
plt.xlabel('Review Length')
plt.ylabel('Rating')
plt.show()
```

```
Average review length: 7.020730813527379
Correlation between review length and rating: -0.4788848381349332
```




# Level 3

Task 2

Task: Votes Analysis

Identify the restaurants with the highest and lowest number of votes.

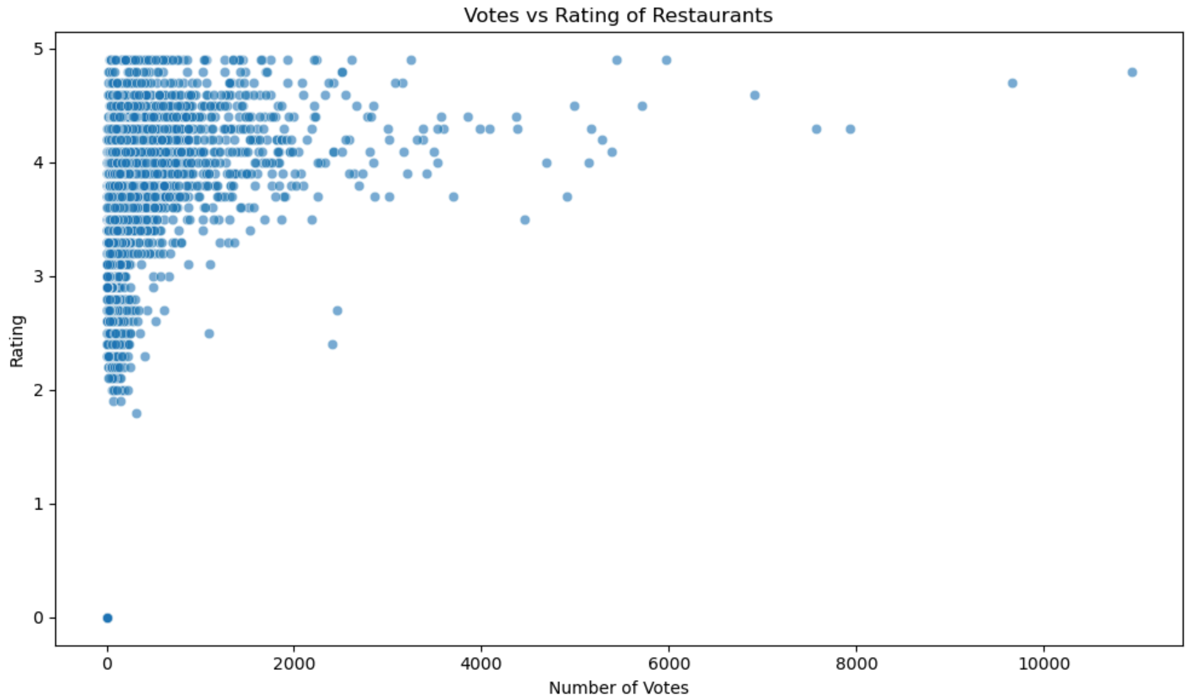
Analyze if there is a correlation between the number of votes and the rating of a restaurant.

In [35]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import pearsonr

df = Data

highest_votes_restaurant = df.loc[df['Votes'].idxmax()]
lowest_votes_restaurant = df.loc[df['Votes'].idxmin()]
print("Restaurant with the highest number of votes:")
print(highest_votes_restaurant)
print("\nRestaurant with the lowest number of votes:")
print(lowest_votes_restaurant)

correlation, p_value = pearsonr(df['Votes'], df['Aggregate rating'])
print(f"\nCorrelation between number of votes and rating: {correlation:.2f}"
print(f"P-value: {p_value:.2f}")
```

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Votes', y='Aggregate rating', data=df, alpha=0.6)
plt.title('Votes vs Rating of Restaurants')
plt.xlabel('Number of Votes')
plt.ylabel('Rating')
plt.tight_layout()
plt.show()
```

```
Restaurant with the highest number of votes:
Restaurant ID                                                        51705
Restaurant Name                                                       Toit
Country Code                                                             1
City                                                             Bangalore
Address                 298, Namma Metro Pillar 62, 100 Feet Road, Ind...
Locality                                                       Indiranagar
Locality Verbose                                    Indiranagar, Bangalore
Longitude                                                        77.640709
Latitude                                                         12.979166
Cuisines                                      [Italian, American, Pizza]
Average Cost for two                                                  2000
Currency                                              Indian Rupees(Rs.)
Has Table booking                                                       No
Has Online delivery                                                     No
Is delivering now                                                       No
Switch to order menu                                                    No
Price range                                                              4
Aggregate rating                                                       4.8
Rating color                                                    Dark Green
Rating text                                                      Excellent
Votes                                                                10934
cuisine_combination                           (American, Italian, Pizza)
cluster                                                                  0
processed_reviews                                              [excellent]
pos_words                                                      [excellent]
neg_words                                                               []
review_length                                                            9
Name: 728, dtype: object

Restaurant with the lowest number of votes:
Restaurant ID                                                      6710645
Restaurant Name                                           Cantinho da Gula
Country Code                                                            30
City                                                             S��o Paulo
Address                 Rua Pedroso Alvarenga, 522, Itaim Bibi, S��o P...
Locality                                                       Itaim Bibi
Locality Verbose                                     Itaim Bibi, S��o Paulo
Longitude                                                       -46.675667
Latitude                                                           -23.581
Cuisines                                                       [Brazilian]
Average Cost for two                                                    55
Currency                                              Brazilian Real(R$)
Has Table booking                                                       No
Has Online delivery                                                     No
Is delivering now                                                       No
Switch to order menu                                                    No
Price range                                                              2
Aggregate rating                                                       0.0
Rating color                                                         White
Rating text                                                      Not rated
Votes                                                                    0
cuisine_combination                                          (Brazilian,)
cluster                                                                  2
processed_reviews                                                  [rated]
pos_words                                                               []
```

```
neg_words                                                          []
review_length                                                       9
Name: 69, dtype: object

Correlation between number of votes and rating: 0.31
P-value: 0.00
```




In [36]:

```
df=Data
highest_votes_restaurant = df.loc[df['Votes'].idxmax()]
lowest_votes_restaurant = df.loc[df['Votes'].idxmin()]

output_df = pd.concat([highest_votes_restaurant, lowest_votes_restaurant], a
output_df.to_csv('highest_lowest_votes_restaurants.csv', index=False)
print("CSV file saved successfully.")
```

```
CSV file saved successfully.
```

# Level 3

Task: Price Range vs. Online Delivery and Table Booking

Analyze if there is a relationship between the price range and the availability of online delivery and table booking.

Determine if higher-priced restaurants are more likely to offer these services.

In [37]:

```
import pandas as pd
from scipy.stats import chi2_contingency
proportion_data = Data.groupby('Price range')[['Has Online delivery', 'Has T
print("Proportion of Restaurants Offering Online Delivery and Table Booking
print(proportion_data)
```

```
higher_priced_delivery = Data['Has Online delivery'].iloc[-1]
higher_priced_booking = Data['Has Table booking'].iloc[-1]
lower_priced_delivery = Data['Has Online delivery'].iloc[0]
lower_priced_booking = Data['Has Table booking'].iloc[0]
print("\nComparison of Higher-Priced Restaurants vs. Lower-Priced Restaurant
print("Online Delivery - Higher-Priced vs. Lower-Priced:", higher_priced_del
print("Table Booking - Higher-Priced vs. Lower-Priced:", higher_priced_booki
```

```
Proportion of Restaurants Offering Online Delivery and Table Booking by Pric
e Range:
Empty DataFrame
Columns: []
Index: [1, 2, 3, 4]

Comparison of Higher-Priced Restaurants vs. Lower-Priced Restaurants:
Online Delivery - Higher-Priced vs. Lower-Priced: No vs. No
Table Booking - Higher-Priced vs. Lower-Priced: No vs. Yes
```

```
/tmp/ipykernel_2460/4131093833.py:3: FutureWarning: The default value of num
eric_only in DataFrameGroupBy.mean is deprecated. In a future version, numer
ic_only will default to False. Either specify numeric_only or select only co
lumns which should be valid for the function.
  proportion_data = Data.groupby('Price range')[['Has Online delivery', 'Has
Table booking']].mean()
```

In [38]:

```
import pandas as pd
from scipy.stats import chi2_contingency
contingency_table = pd.crosstab(Data['Price range'],
                                 [Data['Has Online delivery'], Data['Has Tab
                                 rownames=['Price Range'],
                                 colnames=['Online Delivery', 'Table Booking
print("Contingency Table:")
print(contingency_table)

chi2, p, _, _ = chi2_contingency(contingency_table)

print("\nChi-square test statistic:", chi2)
print("p-value:", p)

if p < 0.05:
    print("There is a significant relationship between price range and the a
else:
    print("There is no significant relationship between price range and the
```

```
Contingency Table:
Online Delivery    No        Yes
Table Booking      No  Yes    No  Yes
Price Range
1                3743    0   700    1
2                1711  116  1163  123
3                 624  373   140  271
4                 299  234    13   40

Chi-square test statistic: 3778.7126357124143
p-value: 0.0
There is a significant relationship between price range and the availability
of online delivery and table booking.
```

```
In [ ]:
```