Cognifyz_Internship

April 22, 2024

1 Level 1 - Task 1: Top Cuisines

Importing the Required libraries

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sn
import matplotlib.pyplot as mp
import warnings
warnings.filterwarnings('ignore')
from collections import Counter
```

[2]:
```
data = pd.read_csv(r'C:\Users\DELL\Desktop\Internship\Dataset.csv')
```

[3]:
```
data.head(10)
```

[3]:
```
   Restaurant ID                      Restaurant Name  Country Code  \
0        6317637                     Le Petit Souffle           162
1        6304287                     Izakaya Kikufuji           162
2        6300002               Heat - Edsa Shangri-La           162
3        6318506                                 Ooma           162
4        6314302                          Sambo Kojin           162
5       18189371                        Din Tai Fung           162
6        6300781                          Buffet 101           162
7        6301290                              Vikings           162
8        6300010  Spiral - Sofitel Philippine Plaza Manila           162
9        6314987                             Locavore           162

               City                                            Address  \
0       Makati City  Third Floor, Century City Mall, Kalayaan Avenu…
1       Makati City  Little Tokyo, 2277 Chino Roces Avenue, Legaspi…
2  Mandaluyong City  Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal…
3  Mandaluyong City  Third Floor, Mega Fashion Hall, SM Megamall, O…
4  Mandaluyong City  Third Floor, Mega Atrium, SM Megamall, Ortigas…
5  Mandaluyong City  Ground Floor, Mega Fashion Hall, SM Megamall, …
6        Pasay City  Building K, SM By The Bay, Sunset Boulevard, M…
7        Pasay City  Building B, By The Bay, Seaside Boulevard, Mal…
8        Pasay City  Plaza Level, Sofitel Philippine Plaza Manila, …
```

```
9       Pasig City  Brixton Technology Center, 10 Brixton Street, …

                                       Locality  \
0       Century City Mall, Poblacion, Makati City
1      Little Tokyo, Legaspi Village, Makati City
2      Edsa Shangri-La, Ortigas, Mandaluyong City
3          SM Megamall, Ortigas, Mandaluyong City
4          SM Megamall, Ortigas, Mandaluyong City
5          SM Megamall, Ortigas, Mandaluyong City
6  SM by the Bay, Mall of Asia Complex, Pasay City
7  SM by the Bay, Mall of Asia Complex, Pasay City
8     Sofitel Philippine Plaza Manila, Pasay City
9                                       Kapitolyo

                                 Locality Verbose   Longitude   Latitude  \
0  Century City Mall, Poblacion, Makati City, Mak…  121.027535  14.565443
1  Little Tokyo, Legaspi Village, Makati City, Ma…  121.014101  14.553708
2  Edsa Shangri-La, Ortigas, Mandaluyong City, Ma…  121.056831  14.581404
3  SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.056475  14.585318
4  SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.057508  14.584450
5  SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.056314  14.583764
6  SM by the Bay, Mall of Asia Complex, Pasay Cit…  120.979667  14.531333
7  SM by the Bay, Mall of Asia Complex, Pasay Cit…  120.979333  14.540000
8  Sofitel Philippine Plaza Manila, Pasay City, P…  120.980090  14.552990
9                            Kapitolyo, Pasig City  121.056532  14.572041

                           Cuisines  …          Currency  \
0        French, Japanese, Desserts  …  Botswana Pula(P)
1                          Japanese  …  Botswana Pula(P)
2   Seafood, Asian, Filipino, Indian  …  Botswana Pula(P)
3                   Japanese, Sushi  …  Botswana Pula(P)
4                  Japanese, Korean  …  Botswana Pula(P)
5                           Chinese  …  Botswana Pula(P)
6                   Asian, European  …  Botswana Pula(P)
7  Seafood, Filipino, Asian, European  …  Botswana Pula(P)
8            European, Asian, Indian  …  Botswana Pula(P)
9                          Filipino  …  Botswana Pula(P)

  Has Table booking Has Online delivery Is delivering now  \
0               Yes                  No                No
1               Yes                  No                No
2               Yes                  No                No
3                No                  No                No
4               Yes                  No                No
5                No                  No                No
6               Yes                  No                No
7               Yes                  No                No
```

```
8              Yes                    No                No
9              Yes                    No                No

  Switch to order menu Price range  Aggregate rating  Rating color  \
0                   No            3               4.8    Dark Green
1                   No            3               4.5    Dark Green
2                   No            4               4.4         Green
3                   No            4               4.9    Dark Green
4                   No            4               4.8    Dark Green
5                   No            3               4.4         Green
6                   No            4               4.0         Green
7                   No            4               4.2         Green
8                   No            4               4.9    Dark Green
9                   No            3               4.8    Dark Green

  Rating text Votes
0   Excellent   314
1   Excellent   591
2   Very Good   270
3   Excellent   365
4   Excellent   229
5   Very Good   336
6   Very Good   520
7   Very Good   677
8   Excellent   621
9   Excellent   532

[10 rows x 21 columns]
```

[4]:
```
data.shape
```

[4]:
```
(9551, 21)
```

[5]:
```
data.dtypes
```

[5]:
```
Restaurant ID             int64
Restaurant Name          object
Country Code              int64
City                     object
Address                  object
Locality                 object
Locality Verbose         object
Longitude               float64
Latitude                float64
Cuisines                 object
Average Cost for two      int64
Currency                 object
```

```
Has Table booking        object
Has Online delivery      object
Is delivering now        object
Switch to order menu     object
Price range               int64
Aggregate rating        float64
Rating color             object
Rating text              object
Votes                     int64
dtype: object
```

[6]:
```
data.isnull().sum()
```

[6]:
```
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
```

From the above results we can observe that there are total 9 null values in the cuisines data we have to remove those null values.

1.1 Task-1 Objective-1: Determine the top three most common cuisines in the dataset.

[7]:
```
data1=data.dropna()
```

[8]:
```
data1.Cuisines.dropna()
```

[8]:
```
0             French, Japanese, Desserts
1                               Japanese
```

```
2       Seafood, Asian, Filipino, Indian
3                       Japanese, Sushi
4                      Japanese, Korean
                      …
9546                            Turkish
9547     World Cuisine, Patisserie, Cafe
9548              Italian, World Cuisine
9549                     Restaurant Cafe
9550                                Cafe
Name: Cuisines, Length: 9542, dtype: object
```

[9]:
```python
data1.shape
```

[9]:
```
(9542, 21)
```

[10]:
```python
data1.Cuisines.isnull().sum()
```

[10]:
```
0
```

[11]:
```python
total_cuisine_count = data1.Cuisines.value_counts()
total_cuisine_count.sort_values(ascending=False)
total_cuisine_count
```

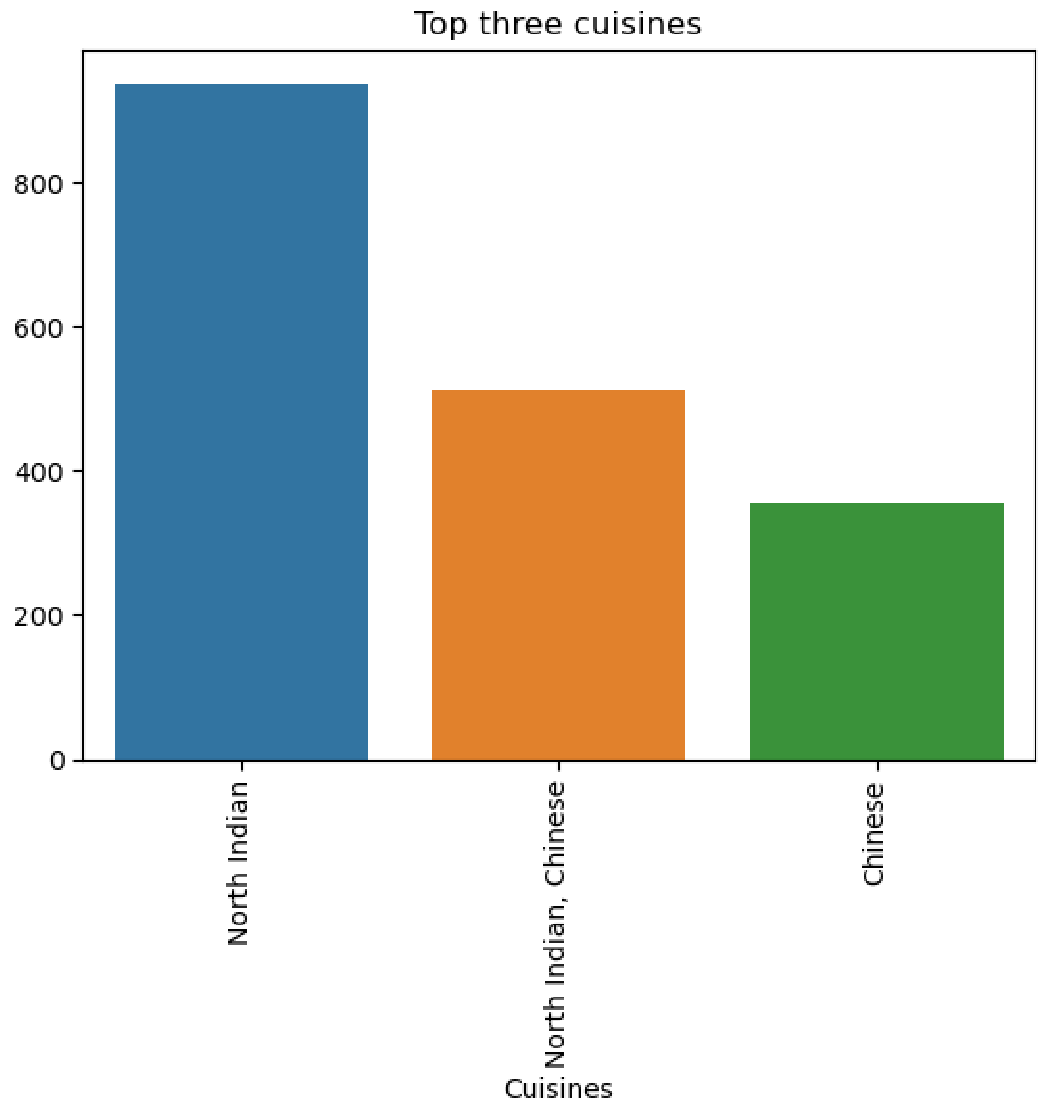
[11]:
```
Cuisines
North Indian                                          936
North Indian, Chinese                                 511
Chinese                                               354
Fast Food                                             354
North Indian, Mughlai                                 334
                                                     …
Bengali, Fast Food                                      1
North Indian, Rajasthani, Asian                         1
Chinese, Thai, Malaysian, Indonesian                    1
Bakery, Desserts, North Indian, Bengali, South Indian   1
Italian, World Cuisine                                  1
Name: count, Length: 1825, dtype: int64
```

[12]:
```python
top_three_cuisines = total_cuisine_count.head(3)
top_three_cuisines
```

[12]:
```
Cuisines
North Indian             936
North Indian, Chinese    511
Chinese                  354
Name: count, dtype: int64
```

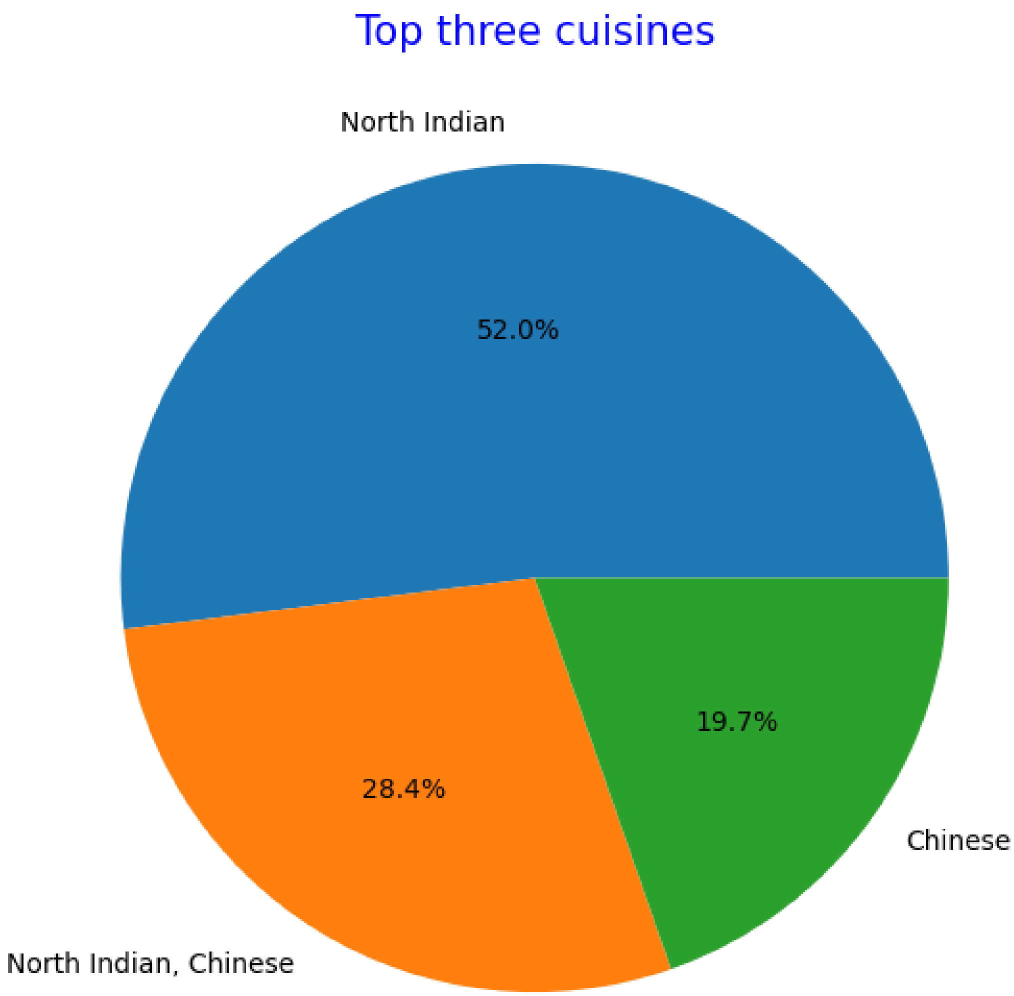
[13]:
```
sn.barplot(x=top_three_cuisines.index,y=top_three_cuisines.values)
mp.xticks(rotation=90)
mp.xlabel("Cuisines")
mp.title('Top three cuisines')
```

[13]: Text(0.5, 1.0, 'Top three cuisines')



[14]:
```
mp.figure(figsize=(7,7))
mp.pie(top_three_cuisines,labels=top_three_cuisines.index,autopct='%1.1f%%')
mp.title('Top three cuisines',fontsize=15,color='blue')
```

[14]: Text(0.5, 1.0, 'Top three cuisines')



1.2 Task-1 Objective 2: Calculate the percentage of restaurants that serve each of the top cuisines

[15]:
```
restaurants_top_cuisines = data.groupby(by='City')['Restaurant Name'].
  ↪value_counts()
restaurants_top_cuisines
```

[15]:
```
City        Restaurant Name
Abu Dhabi   Applebee's                1
            Sofra Istanbul            1
            Via Delhi                 1
            Tikka Tonight             1
            The Cheesecake Factory    1
```

```
                                            ..
 stanbul  Draft Gastro Pub                  1
          Dem Karak _y                      1
          Ceviz A ac                        1
          Baltazar                          1
          Walter's Coffee Roastery          1
Name: count, Length: 7974, dtype: int64
```

[16]:
```
restaurants_top_cuisines.sort_values(ascending=False)
```

[16]:
```
City        Restaurant Name
New Delhi   Cafe Coffee Day                 57
            Domino's Pizza                  55
            Subway                          38
            Green Chick Chop                37
            McDonald's                      33
                                            ..
            Sher E Punjab                    1
            Sher -E- Punjab                  1
            SFC                              1
            Shaolin                          1
 stanbul  Walter's Coffee Roastery          1
Name: count, Length: 7974, dtype: int64
```

[17]:
```
total_count_of_restaurants = data[data['Cuisines'].isin(top_three_cuisines.
 ↪index)]
```

[18]:
```
top_three_cuisines_served_restaurants = total_count_of_restaurants['Restaurant␣
 ↪Name'].nunique()
top_three_cuisines_served_restaurants
```

[18]:
```
1617
```

There are 1617 restaurants that serve the top three cuisines.

[19]:
```
total_number_of_restaurants =data['Restaurant Name'].count()
total_number_of_restaurants
```

[19]:
```
9551
```

[20]:
```
percentage_of_restaurants_that_serve_top_three_cuisines =␣
 ↪top_three_cuisines_served_restaurants/total_number_of_restaurants*100
percentage_of_restaurants_that_serve_top_three_cuisines
```

[20]:
```
16.93016438069312
```

2 Level 1-Task 2: City Analysis

2.1 Task 2 Objective 1: Identify the city with the highest number of restaurants in the dataset.

[21]:
```
data.groupby(by='City')['Restaurant Name'].sum()
```

[21]:
```
City
Abu Dhabi          Denny'sFamous Dave's BarbecuePizza Di RoccoSof…
Agra               JahanpanahRangrezz RestaurantTime2Eat - Mama C…
Ahmedabad          650 - The Global KitchenPatang - The Revolving…
Albany             Austin's BBQ and Oyster BarBJ's Country Buffet…
Allahabad          Aryan Family's DelightBean HereBikanerwalaDews…
                                        …
Weirton                     Theo Yianni's Authentic Greek Restaurant
Wellington City    Maranui CafeFive BoroughsEkim BurgersOmbraThe …
Winchester Bay                                Fishpatrick's Crabby Cafe
Yorkton                                                  Arigato Sushi
 stanbul          J'adore ChocolatierStarbucksValoniaDraft Gastr…
Name: Restaurant Name, Length: 141, dtype: object
```

[22]:
```
city_with_highest_number_of_restaurants = data.groupby(by='City')['Restaurant␣
↪Name'].value_counts()
city_with_highest_number_of_restaurants
```

[22]:
```
City       Restaurant Name
Abu Dhabi  Applebee's                1
           Sofra Istanbul            1
           Via Delhi                 1
           Tikka Tonight             1
           The Cheesecake Factory    1
                                    ..
 stanbul  Draft Gastro Pub          1
           Dem Karak _y              1
           Ceviz A ac               1
           Baltazar                  1
           Walter's Coffee Roastery  1
Name: count, Length: 7974, dtype: int64
```

[23]:
```
city_with_highest_number_of_restaurants.sort_values(ascending=False)
```

[23]:
```
City       Restaurant Name
New Delhi  Cafe Coffee Day           57
           Domino's Pizza            55
           Subway                    38
           Green Chick Chop          37
           McDonald's                33
                                     ..
```

```
            Sher E Punjab                1
            Sher -E- Punjab              1
            SFC                          1
            Shaolin                      1
 stanbul   Walter's Coffee Roastery     1
Name: count, Length: 7974, dtype: int64
```

2.2 Task 2 Objective 2&3: Calculate the average rating for restaurants in each city and the city with the highest average rating

[24]:
```
average_rating_restaurants_in_each_city = data.groupby(by='City')[['Restaurant␣
↪Name','Aggregate rating']].value_counts()
average_rating_restaurants_in_each_city
```

[24]:
```
City        Restaurant Name           Aggregate rating
Abu Dhabi   Applebee's                4.0                 1
            Via Delhi                 4.0                 1
            Tikka Tonight             4.0                 1
            The Cheesecake Factory    4.6                 1
            Tamba                     4.7                 1
                                                         ..
 stanbul   Leman K _lt _r            3.7                 1
            Naml  Gurme              4.1                 1
            Starbucks                 4.9                 1
            Valonia                   4.2                 1
            Walter's Coffee Roastery  4.0                 1
Name: count, Length: 9011, dtype: int64
```

[25]:
```
average_rating_restaurants_in_each_city.sort_values(ascending=False)
```

[25]:
```
City        Restaurant Name                Aggregate rating
New Delhi   Aggarwal Sweets                0.0                 10
Noida       Baskin Robbins                 0.0                  8
New Delhi   Cafe Coffee Day                0.0                  8
            McDonald's                     3.4                  8
            Domino's Pizza                 2.5                  8
                                                               ..
            Street Chaat Chatoron Ka Adda  0.0                  1
            Stabbers                       2.4                  1
            Standard Burfee                3.8                  1
            Standard Corner                3.6                  1
 stanbul   Walter's Coffee Roastery       4.0                  1
Name: count, Length: 9011, dtype: int64
```

[26]:
```
average_rating_restaurants_in_each_city_high = data.
↪groupby(by='City')['Aggregate rating'].mean()
average_rating_restaurants_in_each_city_high.sort_values(ascending=False)
```

[26]:
```
City
Inner City          4.900000
Quezon City         4.800000
Makati City         4.650000
Pasig City          4.633333
Mandaluyong City    4.625000
                      …
New Delhi           2.438845
Montville           2.400000
Mc Millan           2.400000
Noida               2.036204
Faridabad           1.866932
Name: Aggregate rating, Length: 141, dtype: float64
```

3 Level 1 - Task 3: Price Range Distribution

3.1 Task 3 Objective 1: Create a histogram or bar chart to visualize the distribution of price ranges among the restaurants.

[27]:
```
price_ranges_restaurants = data.groupby(by='Restaurant Name')['Price range'].
  ↪mean()
price_ranges_restaurants
```

[27]:
```
Restaurant Name
#45                     2.0
#Dilliwaala6            3.0
#InstaFreeze            1.0
#OFF Campus             2.0
#Urban Caf             2.0
                       …
t Lounge by Dilmah      2.0
tashas                  4.0
wagamama                4.0
{Niche} - Cafe & Bar    3.0
 ukura a Sofras        3.0
Name: Price range, Length: 7446, dtype: float64
```

[28]:
```
price_ranges_restaurants=price_ranges_restaurants.sort_values()
price_ranges_restaurants
```

[28]:
```
Restaurant Name
Laxmi Food Corner         1.0
Grover Burfee & Cakes     1.0
Grover Dhaba              1.0
Grover Eating Point       1.0
Grover Mithaivala         1.0
                        …
```

```
Downtown Grill           4.0
Kinoshita                4.0
Carnival By Tresind      4.0
Draft Gastro Pub         4.0
Restaurant Andre         4.0
Name: Price range, Length: 7446, dtype: float64
```

[29]:

```python
number_of_price_ranges_restaurants = price_ranges_restaurants.value_counts()
number_of_price_ranges_restaurants
```

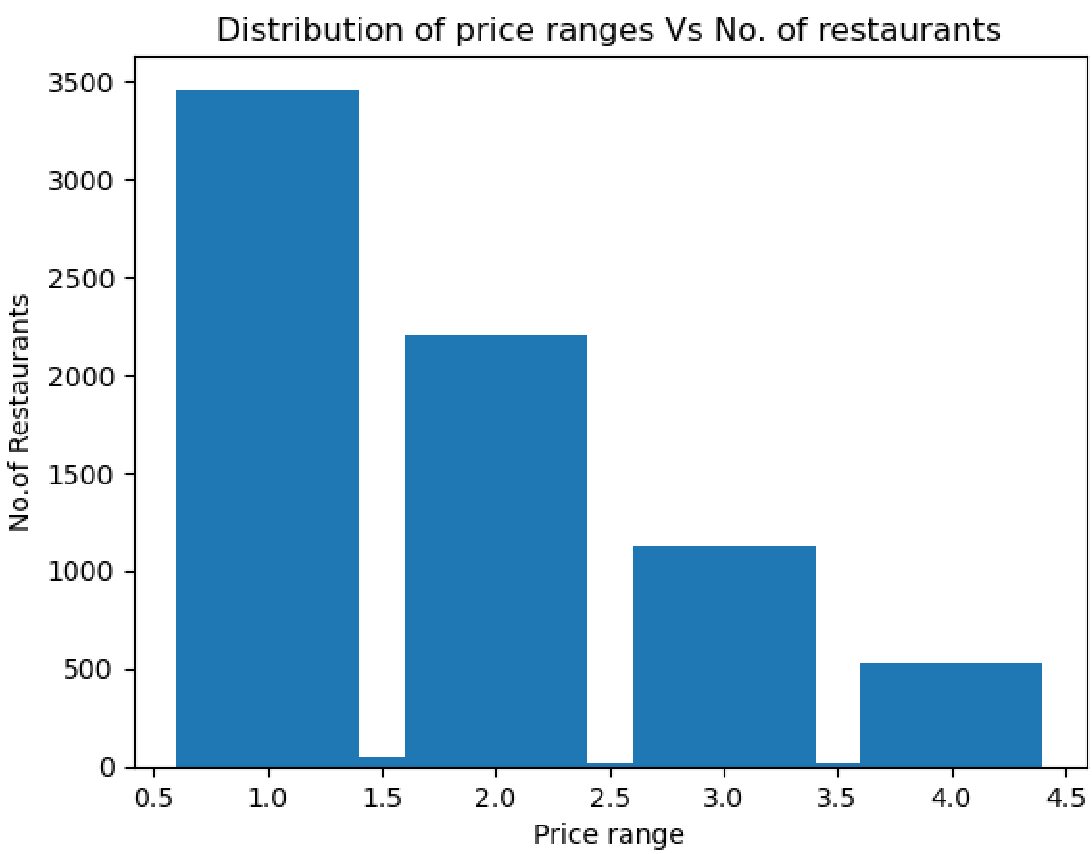
[29]:

```
Price range
1.000000    3453
2.000000    2204
3.000000    1129
4.000000     523
1.500000      44
2.500000      20
3.500000      14
1.333333       8
3.666667       4
2.333333       4
3.333333       3
1.250000       3
2.833333       2
1.200000       2
3.250000       2
1.375000       2
1.750000       2
1.666667       2
3.600000       1
1.222222       1
3.384615       1
2.766667       1
3.200000       1
1.090909       1
2.888889       1
2.857143       1
3.750000       1
2.777778       1
2.600000       1
2.750000       1
1.285714       1
2.571429       1
2.550000       1
2.384615       1
1.083333       1
2.055556       1
```

```
1.125000       1
1.979167       1
1.928571       1
1.875000       1
1.800000       1
1.571429       1
1.157895       1
Name: count, dtype: int64
```

[30]:
```python
mp.bar(number_of_price_ranges_restaurants.
 ↪index,height=number_of_price_ranges_restaurants.values)
mp.xlabel('Price range')
mp.ylabel('No.of Restaurants')
mp.title('Distribution of price ranges Vs No. of restaurants')
```

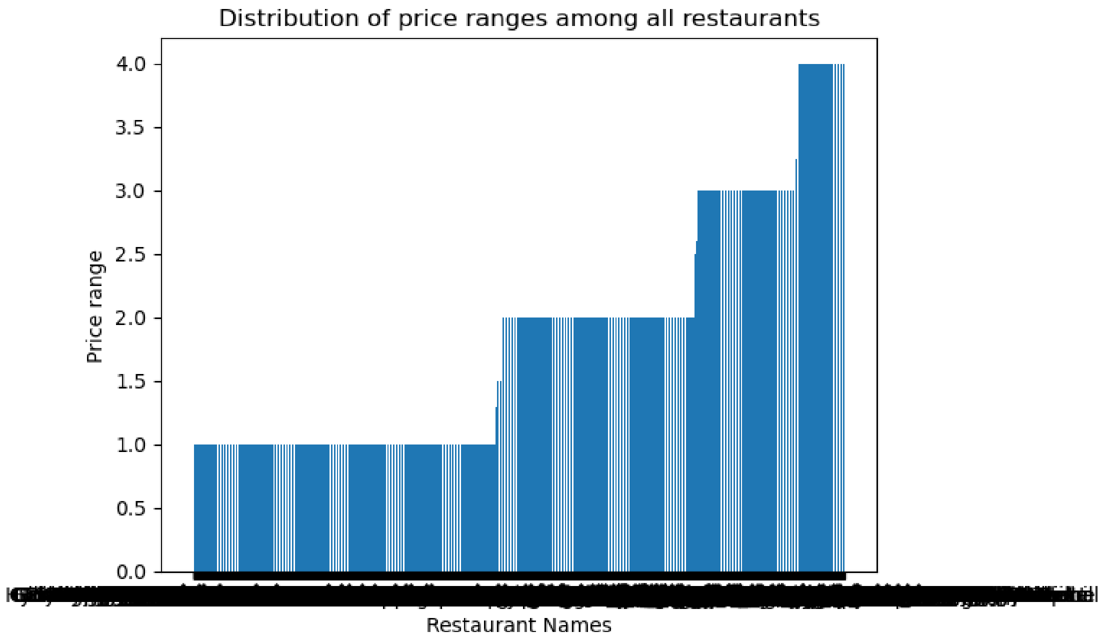
[30]:
```
Text(0.5, 1.0, 'Distribution of price ranges Vs No. of restaurants')
```



[31]:
```python
mp.bar(price_ranges_restaurants.index,height=price_ranges_restaurants.values)
mp.xlabel('Restaurant Names')
mp.ylabel('Price range')
```

```
mp.title('Distribution of price ranges among all restaurants')
```

[31]: Text(0.5, 1.0, 'Distribution of price ranges among all restaurants')



3.2 Task 3 Objective 2: Calculate the percentage of restaurants in each price range category.

[32]:
```
price_range_in_restaurant_name_wise = data.groupby(by='Price␣
↪range')['Restaurant Name'].value_counts()
price_range_in_restaurant_name_wise
```

[32]:
```
Price range  Restaurant Name
1            Cafe Coffee Day                83
             Green Chick Chop               51
             Keventers                      34
             Giani                          29
             Baskin Robbins                 28
                                            ..
4            Larry's China - Taj Vivanta     1
             Lakhori - Haveli Dharampura     1
             Lake House Restaurant           1
             La Piazza - Hyatt Regency       1
             wagamama                        1
Name: count, Length: 7599, dtype: int64
```

[33]:
```
#Total no.of restaurants
total_number_of_restaurants = data['Restaurant Name'].count()
total_number_of_restaurants
```

[33]: 9551

[34]:
```
for i in range (1,5):
    percentage_of_restaurants_in_each_price_range = (data['Price range']==i).
 ↪sum()/total_number_of_restaurants*100
    price_range_in_restaurant_name_wise
    print('Percentage of restaurants in the price range␣
 ↪of',i,percentage_of_restaurants_in_each_price_range)
```

```
Percentage of restaurants in the price range of 1 46.52915925034028
Percentage of restaurants in the price range of 2 32.59344571249084
Percentage of restaurants in the price range of 3 14.741911841691968
Percentage of restaurants in the price range of 4 6.135483195476914
```

4 Level 1 - Task 4: Online delivery

4.1 Task 4 Objective-1: Determine the percentage of restaurants that offer online delivery.

[35]:
```
restaurants_offer_delivery = data.groupby(by='Has Online delivery')['Restaurant␣
 ↪Name'].value_counts()
restaurants_offer_delivery
```

[35]:
```
Has Online delivery  Restaurant Name
No                   Domino's Pizza          79
                     Cafe Coffee Day         78
                     Green Chick Chop        47
                     Keventers               33
                     Barbeque Nation         26
                                             ..
Yes                  Chew - Pan Asian Cafe    1
                     Cherry Fresh             1
                     Cherry Comet             1
                     Chehel Pehel             1
                     iKitchen                 1
Name: count, Length: 7659, dtype: int64
```

[36]:
```
number_of_restaurants_delivery = data.groupby(by='Has Online␣
 ↪delivery')['Restaurant Name'].count()
number_of_restaurants_delivery
```

[36]:
```
Has Online delivery
No     7100
```

```
Yes    2451
Name: Restaurant Name, dtype: int64
```

[37]:
```
percentage_of_restaurants_offer_delivery = number_of_restaurants_delivery.
 ↪values[1]/total_number_of_restaurants*100
percentage_of_restaurants_offer_delivery
```

[37]:
```
25.662234321013504
```

4.2 Task 4 Objective 2: Compare the average ratings of restaurants with and without online delivery.

[38]:
```
average_rating_of_restaurants_with_delivery = data[data['Has Online␣
 ↪delivery']=='Yes']['Aggregate rating'].mean()
average_rating_of_restaurants_with_delivery
```

[38]:
```
3.2488372093023257
```

[39]:
```
average_rating_of_restaurants_without_delivery=data[data['Has Online␣
 ↪delivery']=='No']['Aggregate rating'].mean()
average_rating_of_restaurants_without_delivery
```

[39]:
```
2.465295774647887
```

5 Level 2 - Task 1: Restaurant Ratings

5.1 Task 1 Objective 1: Analyze the distribution of aggregate ratings and determine the most common rating range.

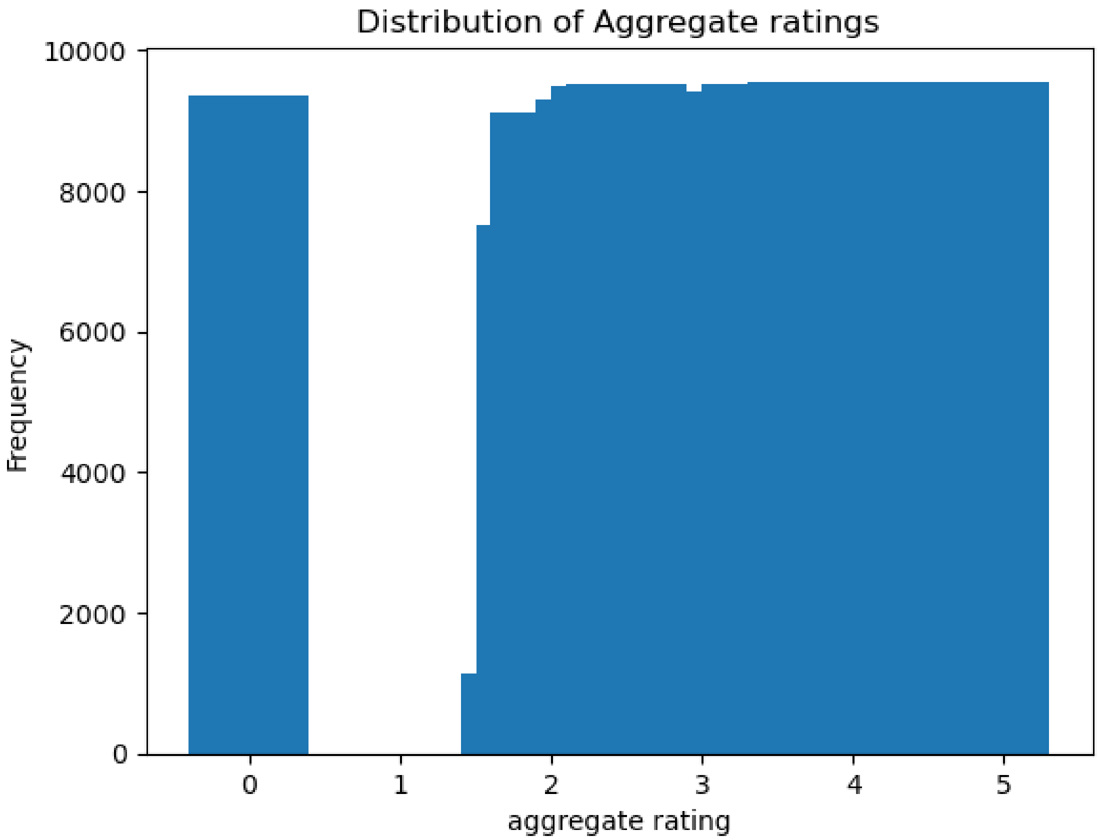
[40]:
```
aggregate_ratings = data['Aggregate rating']
aggregate_ratings.value_counts()
```

[40]:
```
Aggregate rating
0.0    2148
3.2     522
3.1     519
3.4     498
3.3     483
3.5     480
3.0     468
3.6     458
3.7     427
3.8     400
2.9     381
3.9     335
2.8     315
4.1     274
```

```
4.0     266
2.7     250
4.2     221
2.6     191
4.3     174
4.4     144
2.5     110
4.5      95
2.4      87
4.6      78
4.9      61
2.3      47
4.7      42
2.2      27
4.8      25
2.1      15
2.0       7
1.9       2
1.8       1
Name: count, dtype: int64
```

[41]:
```
mp.bar(aggregate_ratings.values,height=aggregate_ratings.index)
mp.xlabel('aggregate rating')
mp.ylabel('Frequency')
mp.title('Distribution of Aggregate ratings')
```

[41]:
```
Text(0.5, 1.0, 'Distribution of Aggregate ratings')
```



5.2 Task 1 Objective 2: Calculate the average number of votes received by restaurants.

[42]:
```
average_number_of_votes_received_by_restaurants = data['Votes'].mean()
average_number_of_votes_received_by_restaurants
```

[42]: 156.909747670401

6 Level 2 - Task 2: Cuisine Combination

6.1 Task 2 Objective 1: Identify the most common combinations of cuisines in the dataset.

[43]:
```
data.dtypes['Cuisines']
```

[43]: dtype('O')

[44]:
```
def identify_common_cuisine_combination(dataset_path):
    cuisine_combinations = data['Cuisines'].str.split(',').explode()
    cuisine_combinations_count = cuisine_combinations.value_counts()
```

```
    return cuisine_combinations_count
dataset_path = 'restaurants.csv'
common_cuisine_combinations = identify_common_cuisine_combination (dataset_path)
print('Most common combinations of Cuisines')
print(common_cuisine_combinations.head(10))
```

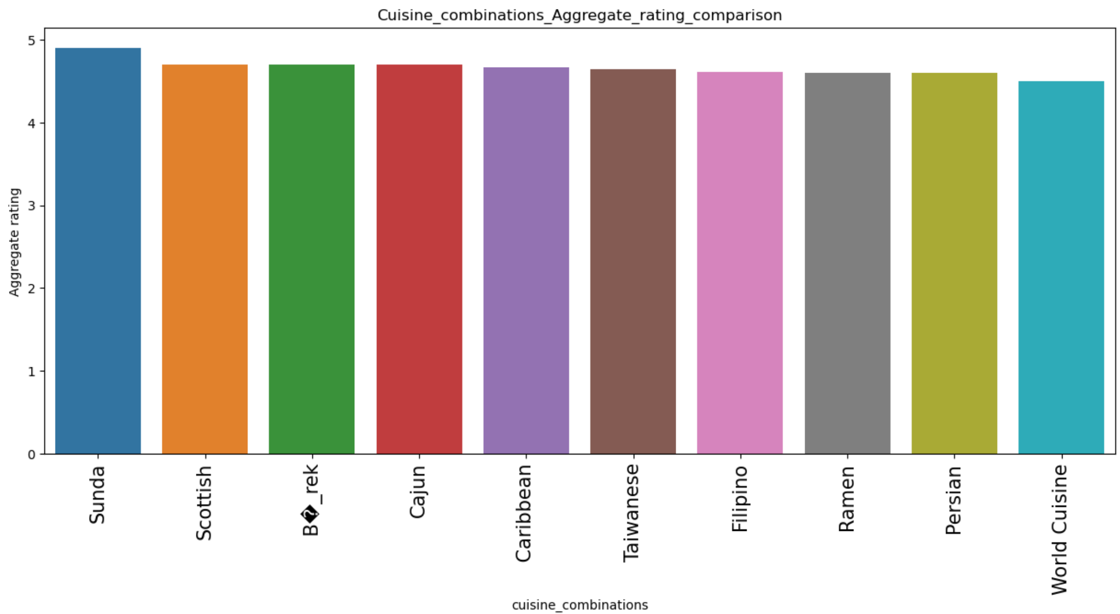
```
Most common combinations of Cuisines
Cuisines
North Indian     2992
 Chinese         1880
 Fast Food       1314
 North Indian     968
Chinese           855
 Mughlai          780
Fast Food         672
Bakery            621
Cafe              617
 Italian          530
Name: count, dtype: int64
```

6.2 Task 2 Objective 2: Determine if certain cuisine combinations tend to have higher ratings.

[45]:
```
def cuisine_combination_ratings(dataset_path):
    data['cuisine_combinations'] = data['Cuisines'].str.split(',')
    data_exploded = data.explode('cuisine_combinations')
    Sorted_ratings = data_exploded.groupby('cuisine_combinations')['Aggregate␣
 ↪rating'].mean().sort_values(ascending=False)
    return Sorted_ratings
datasert_path = 'restaurants.csv'
average_ratings = cuisine_combination_ratings(dataset_path)
Ar = average_ratings.head(10)
print(Ar)
```

```
cuisine_combinations
Sunda            4.900000
Scottish         4.700000
 B _rek         4.700000
Cajun            4.700000
 Caribbean       4.666667
Taiwanese        4.650000
Filipino         4.616667
Ramen            4.600000
Persian          4.600000
World Cuisine    4.500000
Name: Aggregate rating, dtype: float64
```

[46]:
```
mp.figure(figsize=(15,6))
sn.barplot(x=Ar.index, y=Ar.values)
mp.xticks(rotation=90)
mp.ylabel('Aggregate rating')
mp.title('Cuisine_combinations_Aggregate_rating_comparison')
mp.tick_params(axis='x', which='major', labelsize=15)
```



7 Level 2 - Task 3: Geographic Analysis

7.1 Task 3 Objective: Plot the locations of restaurants on a map using longitude and latitude coordinates and identify any patterns or clusters of restaurants in specific areas.

[47]:
```
pip install folium
```

```
Requirement already satisfied: folium in c:\users\dell\anaconda\lib\site-
packages (0.16.0)
Requirement already satisfied: branca>=0.6.0 in c:\users\dell\anaconda\lib\site-
packages (from folium) (0.7.1)
Requirement already satisfied: jinja2>=2.9 in c:\users\dell\anaconda\lib\site-
packages (from folium) (3.1.3)
Requirement already satisfied: numpy in c:\users\dell\anaconda\lib\site-packages
(from folium) (1.26.4)
Requirement already satisfied: requests in c:\users\dell\anaconda\lib\site-
packages (from folium) (2.31.0)
Requirement already satisfied: xyzservices in c:\users\dell\anaconda\lib\site-
packages (from folium) (2022.9.0)
```

```
Requirement already satisfied: MarkupSafe>=2.0 in
c:\users\dell\anaconda\lib\site-packages (from jinja2>=2.9->folium) (2.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
c:\users\dell\anaconda\lib\site-packages (from requests->folium) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell\anaconda\lib\site-
packages (from requests->folium) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
c:\users\dell\anaconda\lib\site-packages (from requests->folium) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\dell\anaconda\lib\site-packages (from requests->folium) (2024.2.2)
Note: you may need to restart the kernel to use updated packages.
```

[48]:
```
restaurant_name = data['Restaurant Name']
latitude = data['Latitude']
longitude= data['Longitude']
```

[49]:
```
import folium
from IPython.display import display
from sklearn.cluster import KMeans
```

[50]:
```
latitude_longitude = data[['Latitude','Longitude']]
number_of_clusters = 10
```

[51]:
```
Kmeans = KMeans(n_clusters=number_of_clusters, random_state=100)
data['Cluster'] = Kmeans.fit_predict(latitude_longitude)
```

[52]:
```
map_center = [latitude.mean(), longitude.mean()]
rest_map = folium.Map(location=map_center, zoom_start=15)
```

[53]:
```
cluster_colour = ['red','blue','green','aqua','black']
```

[54]:
```
for index, row in data.iterrows():
    restaurant_name = row['Restaurant Name']
    latitude = row['Latitude']
    longitude= row['Longitude']
    cuisines = row['Cuisines']
    rating = row['Aggregate rating']
    cluster = row['Cluster']

    popup_text = f"Restaurant: {restaurant_name}\nCuisines: {cuisines}\nRating:␣
  ↪{rating}"
    marker = folium.Marker([latitude, longitude], popup=popup_text)
    marker.add_to(rest_map)
```

[55]:
```
display(rest_map)
```

```
<folium.folium.Map at 0x238fa533810>
```

8 Level 2 - Task 4: Restaurant Chains

8.1 Task 4 Objective 1: Identify if there are any restaurant chains present in the dataset.

[56]:
```
restaurant_chains = data.groupby('Restaurant Name').size().reset_index(name =␣
↪'Chain Count')
restaurant_chains = restaurant_chains[restaurant_chains['Chain Count'] > 1]
restaurant_chains
```

[56]:
```
               Restaurant Name  Chain Count
7            10 Downing Street            2
27          221 B Baker Street            3
44          34 Parkstreet Lane            2
45        34, Chowringhee Lane           12
59              4700BC Popcorn            2
…                          …          …
7383                     Zaika            4
7389         Zaika Kathi Rolls            2
7417                      Zizo            3
7424            Zooby's Kitchen           2
7432                     bu no           2

[734 rows x 2 columns]
```

[57]:
```
restaurant_chains = restaurant_chains.sort_values(by = 'Chain Count', ascending␣
↪= False)
restaurant_chains
```

[57]:
```
       Restaurant Name  Chain Count
1098   Cafe Coffee Day           83
2098    Domino's Pizza           79
6106            Subway           63
2716  Green Chick Chop           51
4077        McDonald's           48
…                  …          …
2770           Gullu's            2
2764             Gulab            2
2746     Grover Sweets            2
2739            Grillz            2
7432             bu no           2

[734 rows x 2 columns]
```

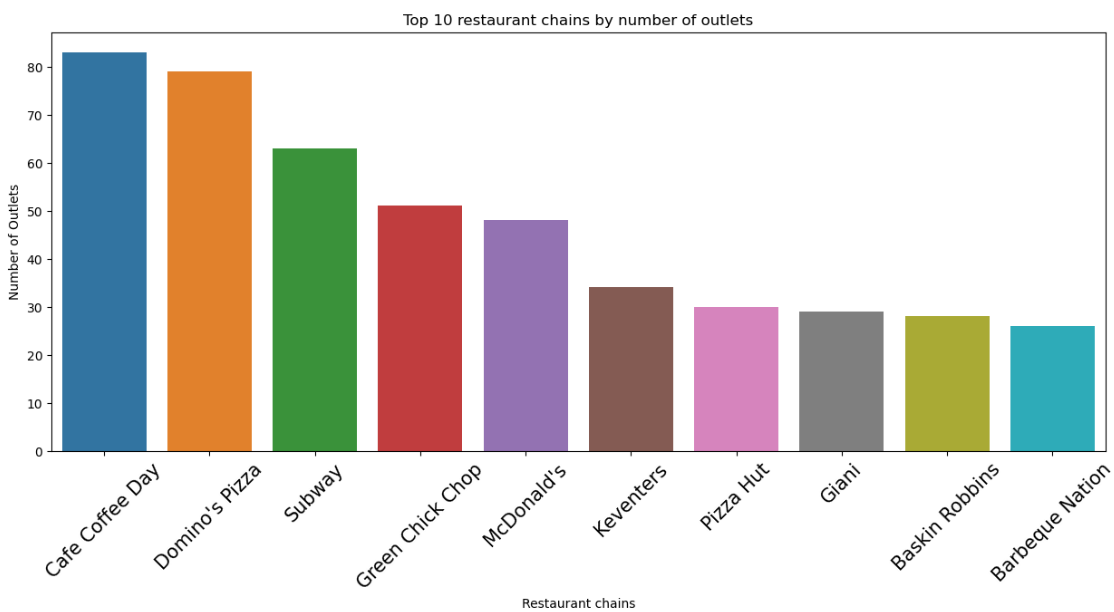
[58]:
```
Top_ten_restaurant_chains = restaurant_chains.head(10)
Top_ten_restaurant_chains
```

[58]:

	Restaurant Name	Chain Count
1098	Cafe Coffee Day	83
2098	Domino's Pizza	79
6106	Subway	63
2716	Green Chick Chop	51
4077	McDonald's	48
3478	Keventers	34
4961	Pizza Hut	30
2619	Giani	29
680	Baskin Robbins	28
663	Barbeque Nation	26

[59]:

```
mp.figure(figsize=(15,6))
sn.barplot(x=Top_ten_restaurant_chains['Restaurant Name'],␣
 ↪y=Top_ten_restaurant_chains['Chain Count'])
mp.xticks(rotation=45)
mp.xlabel('Restaurant chains')
mp.ylabel('Number of Outlets')
mp.title('Top 10 restaurant chains by number of outlets')
mp.tick_params(axis='x', which='major', labelsize=15)
```



8.2 Task 4 Objective 2: Analyze the ratings and popularity of different restaurant chains.

[60]:
```
restaurant_chains_ratings = data.groupby('Restaurant Name')['Aggregate rating'].
↪mean().reset_index(name = 'Average rating')
restaurant_chains_ratings
```

[60]:
```
            Restaurant Name  Average rating
0                       #45             3.6
1              #Dilliwaala6             3.7
2              #InstaFreeze             0.0
3               #OFF Campus             3.7
4              #Urban Caf             3.3
…                       …              …
7441     t Lounge by Dilmah             3.6
7442                 tashas             4.1
7443               wagamama             3.7
7444   {Niche} - Cafe & Bar             4.1
7445      ukura a Sofras             4.4

[7446 rows x 2 columns]
```

[61]:
```
restaurant_chains_votes = data.groupby('Restaurant Name')['Votes'].mean().
↪reset_index(name = 'Total Votes')
restaurant_chains_votes
```

[61]:
```
            Restaurant Name  Total Votes
0                       #45        209.0
1              #Dilliwaala6        124.0
2              #InstaFreeze          2.0
3               #OFF Campus        216.0
4              #Urban Caf        49.0
…                       …           …
7441     t Lounge by Dilmah         34.0
7442                 tashas        374.0
7443               wagamama        131.0
7444   {Niche} - Cafe & Bar        492.0
7445      ukura a Sofras        296.0

[7446 rows x 2 columns]
```

[62]:
```
Analyzed_chains = pd.merge(restaurant_chains_ratings, restaurant_chains_votes,␣
↪on='Restaurant Name')
Analyzed_chains
```

[62]:
```
            Restaurant Name  Average rating  Total Votes
0                       #45             3.6        209.0
1              #Dilliwaala6             3.7        124.0
```

```
2             #InstaFreeze              0.0         2.0
3              #OFF Campus              3.7       216.0
4             #Urban Caf               3.3        49.0
…                      …            …          …
7441     t Lounge by Dilmah              3.6        34.0
7442                 tashas              4.1       374.0
7443               wagamama              3.7       131.0
7444   {Niche} - Cafe & Bar              4.1       492.0
7445      ukura a Sofras               4.4       296.0

[7446 rows x 3 columns]
```

[63]:
```
Sorted_Analyzed_chains = Analyzed_chains.sort_values(by='Average rating',␣
↪ascending=False)
Sorted_Analyzed_chains.head(10)
```

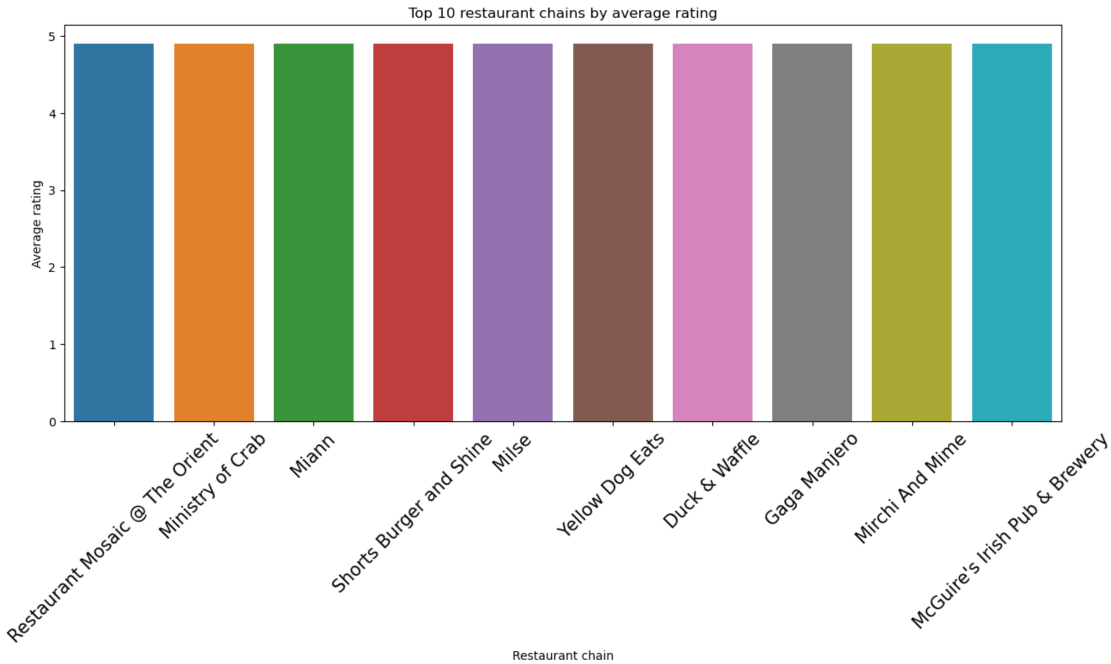
[63]:
```
                      Restaurant Name  Average rating  Total Votes
5322  Restaurant Mosaic @ The Orient             4.9         85.0
4177                 Ministry of Crab             4.9        203.0
4135                            Miann             4.9        281.0
5757          Shorts Burger and Shine             4.9        820.0
4165                            Milse             4.9        754.0
7339                  Yellow Dog Eats             4.9       1252.0
2133                    Duck & Waffle             4.9        706.0
2559                    Gaga Manjero             4.9         95.0
4182                  Mirchi And Mime             4.9       3244.0
4078    McGuire's Irish Pub & Brewery             4.9       2238.0
```

[64]:
```
Sorted_Analyzed_chains.head(10)
```

[64]:
```
                      Restaurant Name  Average rating  Total Votes
5322  Restaurant Mosaic @ The Orient             4.9         85.0
4177                 Ministry of Crab             4.9        203.0
4135                            Miann             4.9        281.0
5757          Shorts Burger and Shine             4.9        820.0
4165                            Milse             4.9        754.0
7339                  Yellow Dog Eats             4.9       1252.0
2133                    Duck & Waffle             4.9        706.0
2559                    Gaga Manjero             4.9         95.0
4182                  Mirchi And Mime             4.9       3244.0
4078    McGuire's Irish Pub & Brewery             4.9       2238.0
```

[65]:
```
mp.figure(figsize=(15,6))
sn.barplot(x=Sorted_Analyzed_chains.head(10)['Restaurant Name'] ,␣
↪y=Sorted_Analyzed_chains.head(10)['Average rating'])
mp.xticks(rotation=45)
mp.xlabel('Restaurant chain')
```

```
mp.ylabel('Average rating')
mp.title('Top 10 restaurant chains by average rating')
mp.tick_params(axis='x', which='major', labelsize=15)
```



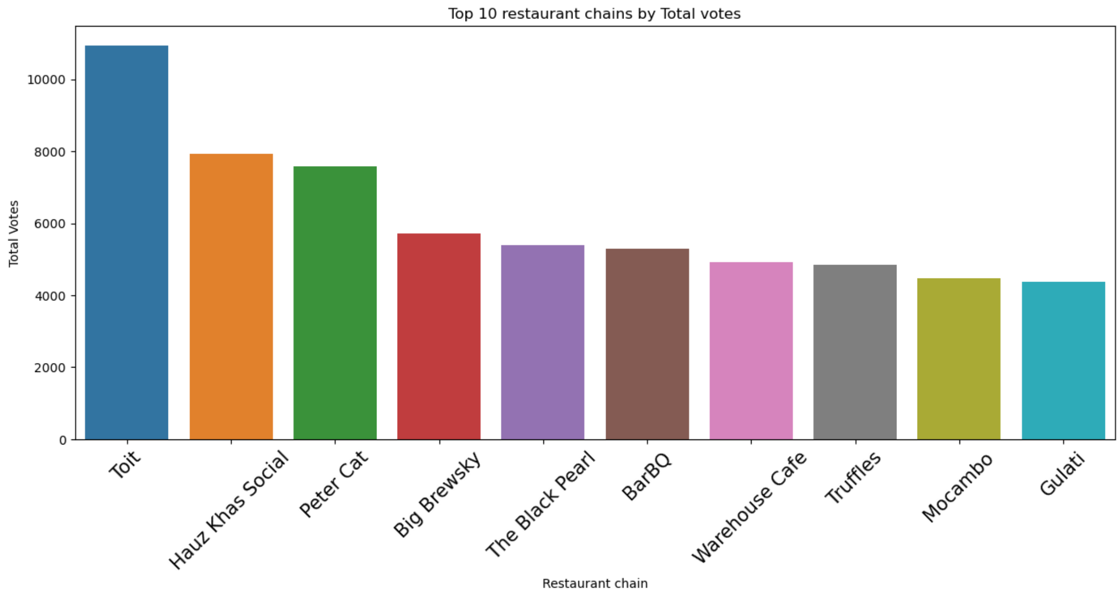
[66]:
```
Sorted_Analyzed_chains_1 = Analyzed_chains.sort_values(by='Total Votes',␣
↪ascending=False)
Sorted_Analyzed_chains_1.head(10)
```

[66]:

	Restaurant Name	Average rating	Total Votes
6943	Toit	4.80	10934.0
2879	Hauz Khas Social	4.30	7931.0
4902	Peter Cat	4.30	7574.0
783	Big Brewsky	4.50	5705.0
6449	The Black Pearl	4.10	5385.0
659	BarBQ	4.20	5288.0
7243	Warehouse Cafe	3.70	4914.0
6988	Truffles	3.95	4841.0
4213	Mocambo	3.50	4464.0
2765	Gulati	4.40	4373.0

[67]:
```
mp.figure(figsize=(15,6))
sn.barplot(x=Sorted_Analyzed_chains_1.head(10)['Restaurant Name'] ,␣
↪y=Sorted_Analyzed_chains_1.head(10)['Total Votes'])
mp.xticks(rotation=45)
mp.xlabel('Restaurant chain')
```

```
mp.ylabel('Total Votes')
mp.title('Top 10 restaurant chains by Total votes')
mp.tick_params(axis='x', which='major', labelsize=15)
```



9 Level 3 - Task 1: Restaurant Reviews

9.1 Task 1 Objective 1: Analyze the text reviews to identify the most common positive and negative keywords.

[68]:
```
reviews = data['Rating text']
reviews
```

[68]:
```
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
```

[69]:
```
reviews.describe()
```

[69]:
```
count          9551
unique            6
top         Average
freq           3737
Name: Rating text, dtype: object
```

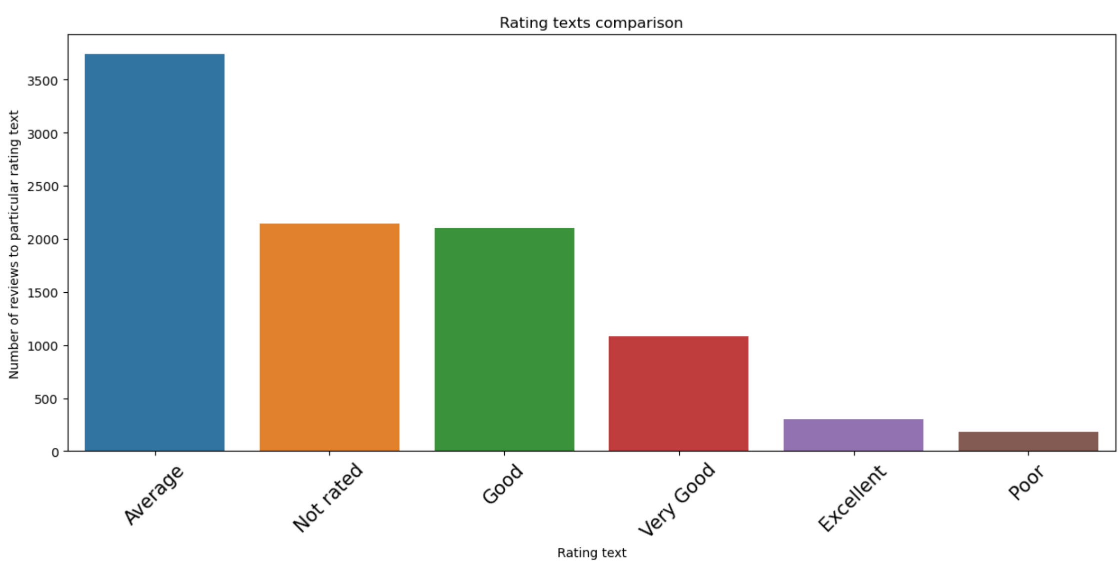
[70]:
```
reviews.value_counts()
```

[70]:
```
Rating text
Average      3737
Not rated    2148
Good         2100
Very Good    1079
Excellent     301
Poor          186
Name: count, dtype: int64
```

[71]:
```
reviews.unique()
```

[71]:
```
array(['Excellent', 'Very Good', 'Good', 'Average', 'Not rated', 'Poor'],
      dtype=object)
```

[72]:
```
mp.figure(figsize=(15,6))
sn.barplot(x=reviews.value_counts().index, y=reviews.value_counts().values)
mp.xticks(rotation=45)
mp.xlabel('Rating text')
mp.ylabel('Number of reviews to particular rating text')
mp.title('Rating texts comparison')
mp.tick_params(axis='x', which='major', labelsize=15)
```



9.2 Task 1 Objective 2: Calculate the average length of reviews and explore if there is a relationship between review length and rating.

[73]:
```
Average_length_of_rating_text = data["Rating text"].apply(lambda x: len(x.
    split()))
Average_length_of_rating_text.value_counts()
```

[73]:
```
Rating text
1    6324
2    3227
Name: count, dtype: int64
```

[74]:
```
Average_length_of_rating_text.mean()
```

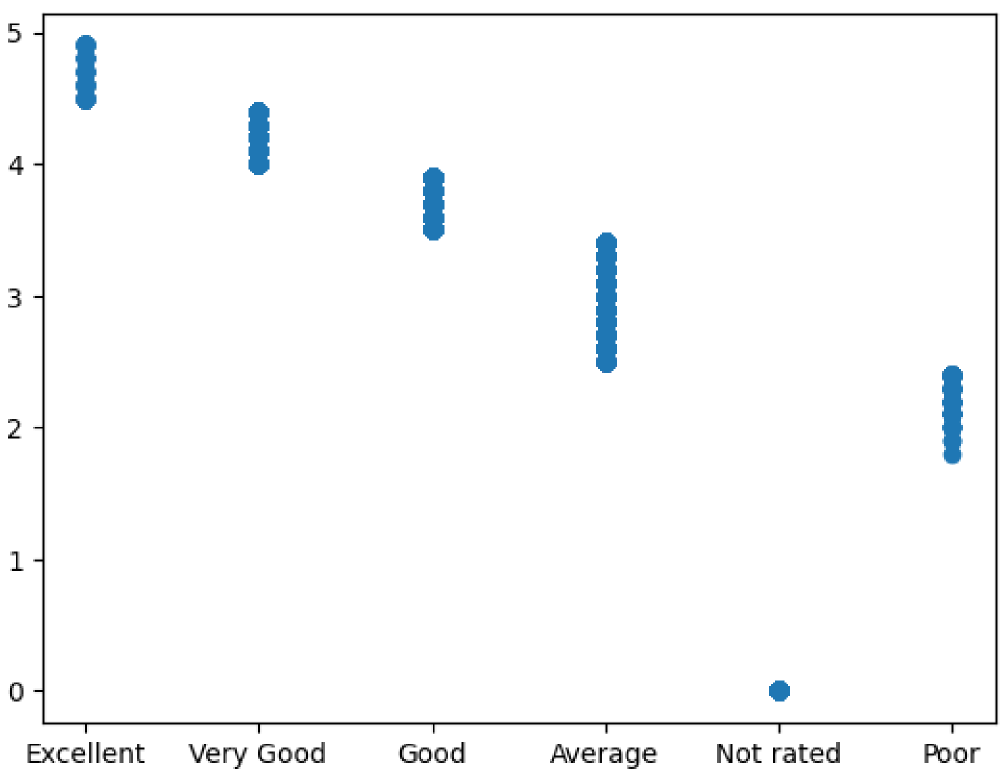
[74]:
```
1.3378703800649148
```

[75]:
```
average_rating = data['Aggregate rating'].mean()
average_rating
```

[75]:
```
2.66637001361114
```

[76]:
```
mp.scatter(data['Rating text'],data['Aggregate rating'])
```

[76]:
```
<matplotlib.collections.PathCollection at 0x23881de2b10>
```



10 Level 3 - Task 2: Votes Analysis

10.1 Task 2 Objective 1: Identify the restaurants with the highest and lowest number of votes.

[77]:
```
Restaurants_with_highest_number_of_votes= data.groupby(by='Restaurant␣
↪Name')['Votes'].mean()
Restaurants_with_highest_number_of_votes.sort_values(ascending=False)
```

[77]:
```
Restaurant Name
Toit                              10934.0
Hauz Khas Social                   7931.0
Peter Cat                          7574.0
Big Brewsky                        5705.0
The Black Pearl                    5385.0
                                    …
Cafe Treat                            0.0
Ralhan Eating Corner                  0.0
Raju Vaishno Amritsari Dhaba          0.0
The Golden Spoon                      0.0
Shree Vinayaga Restaurant             0.0
Name: Votes, Length: 7446, dtype: float64
```

[78]:
```
Restaurant_with_highest_number_of_votes =␣
↪Restaurants_with_highest_number_of_votes.sort_values(ascending=False)
Restaurant_with_highest_number_of_votes.head(1)
```

[78]:
```
Restaurant Name
Toit    10934.0
Name: Votes, dtype: float64
```

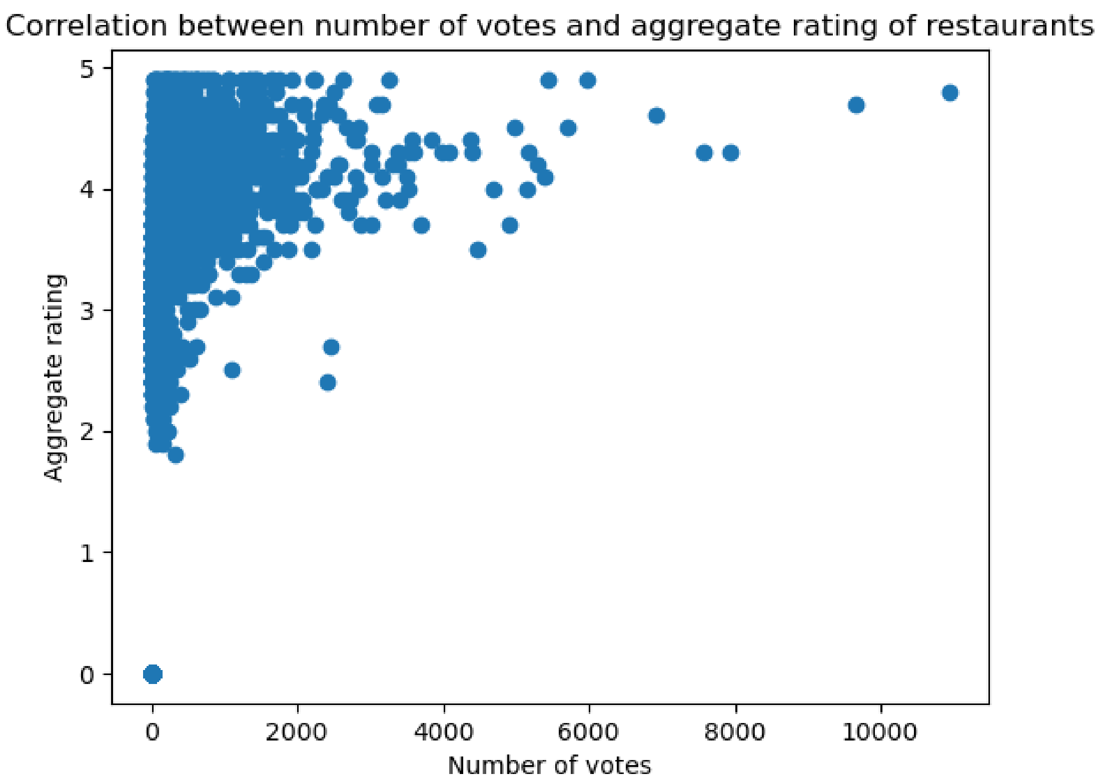
[79]:
```
Restaurant_with_lowest_number_of_votes=Restaurants_with_highest_number_of_votes.
↪sort_values(ascending=False)
Restaurant_with_lowest_number_of_votes.tail(1)
```

[79]:
```
Restaurant Name
Shree Vinayaga Restaurant    0.0
Name: Votes, dtype: float64
```

10.2 Task 2 Objective 2: Analyze if there is a correlation between the number of votes and the rating of a restaurant.

[80]:
```
mp.scatter(data['Votes'],data['Aggregate rating'])
mp.xlabel('Number of votes')
mp.ylabel('Aggregate rating')
mp.title('Correlation between number of votes and aggregate rating of␣
↪restaurants')
```

[80]:
```
Text(0.5, 1.0, 'Correlation between number of votes and aggregate rating of
restaurants')
```



11 Level 3 - Task 3: Price Range vs. Online Delivery and Table Booking

11.1 Task 3 Objective 1: Analyze if there is a relationship between the price range and the availability of online delivery and table booking.

[81]:
```
price_delivery_booking = data.groupby(by='Price range')[['Has Online␣
↪delivery','Has Table booking']]
price_delivery_booking.value_counts().reset_index(name='Values')
```

[81]:

	Price range	Has Online delivery	Has Table booking	Values
0	1	No	No	3743
1	1	Yes	No	700
2	1	Yes	Yes	1
3	2	No	No	1711
4	2	Yes	No	1163
5	2	Yes	Yes	123
6	2	No	Yes	116
7	3	No	No	624
8	3	No	Yes	373
9	3	Yes	Yes	271
10	3	Yes	No	140
11	4	No	No	299
12	4	No	Yes	234
13	4	Yes	Yes	40
14	4	Yes	No	13

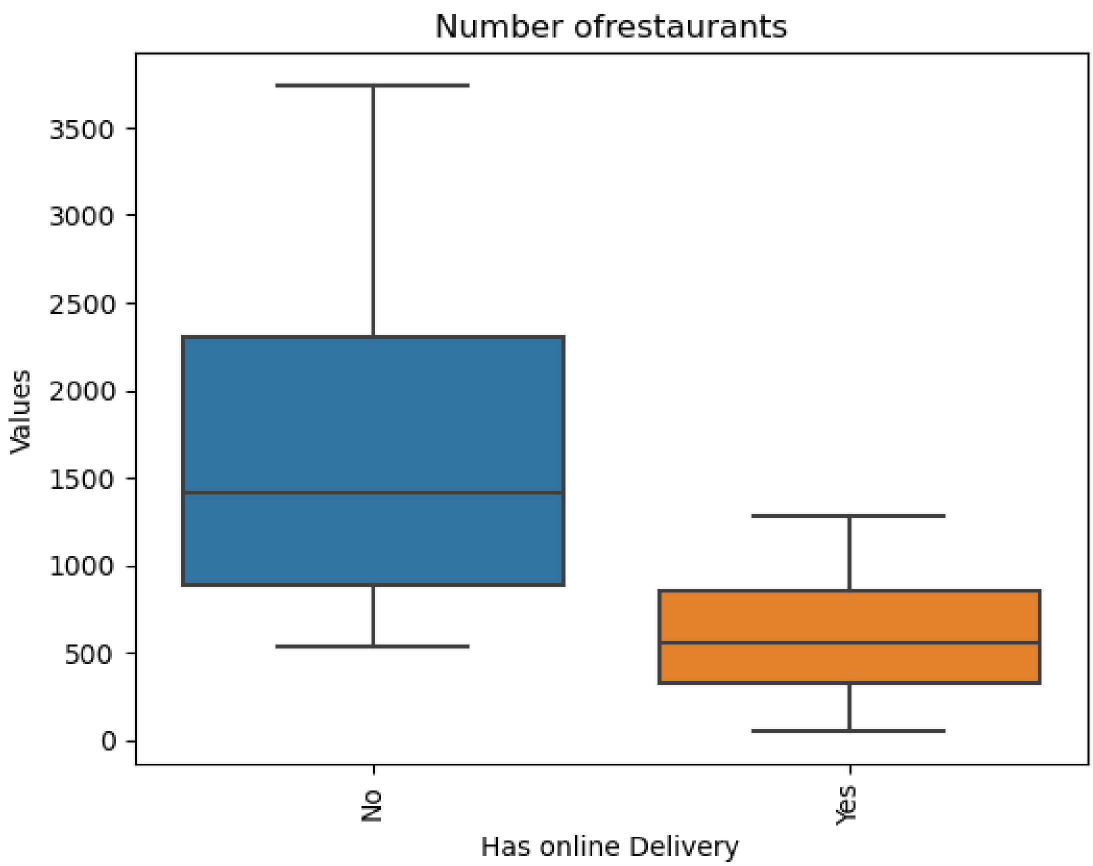
[82]:
```
price_delivery = data.groupby(by='Price range')['Has Online delivery'].
  ↪value_counts().reset_index(name='Values')
price_delivery
```

[82]:

	Price range	Has Online delivery	Values
0	1	No	3743
1	1	Yes	701
2	2	No	1827
3	2	Yes	1286
4	3	No	997
5	3	Yes	411
6	4	No	533
7	4	Yes	53

[83]:
```
sn.boxplot(x=price_delivery['Has Online delivery'], y=price_delivery['Values'])
mp.xticks(rotation=90)
mp.xlabel('Has online Delivery')
mp.title('Number ofrestaurants')
```

[83]: Text(0.5, 1.0, 'Number ofrestaurants')



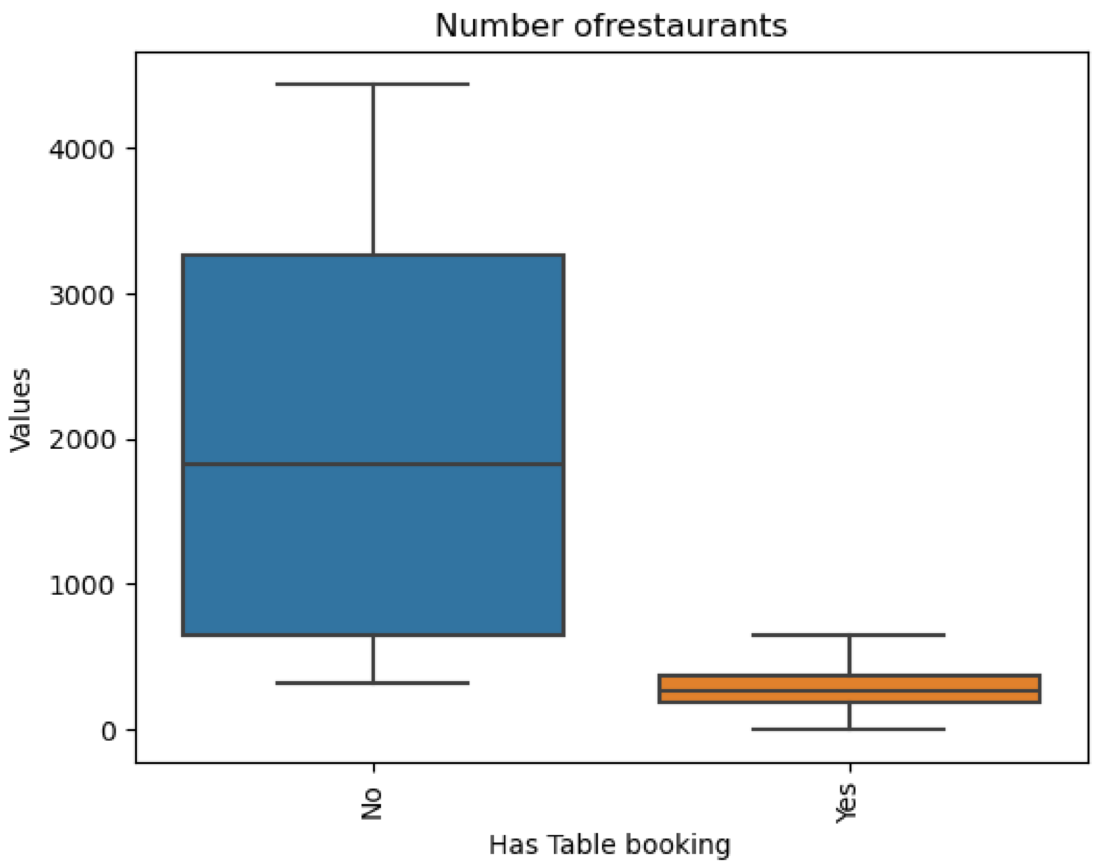
[84]:
```
price_booking = data.groupby(by='Price range')['Has Table booking'].
    ↪value_counts().reset_index(name='Values')
price_booking
```

[84]:

	Price range	Has Table booking	Values
0	1	No	4443
1	1	Yes	1
2	2	No	2874
3	2	Yes	239
4	3	No	764
5	3	Yes	644
6	4	No	312
7	4	Yes	274

[85]:
```
sn.boxplot(x=price_booking['Has Table booking'], y=price_booking['Values'])
mp.xticks(rotation=90)
mp.xlabel('Has Table booking')
mp.title('Number ofrestaurants')
```

[85]: Text(0.5, 1.0, 'Number ofrestaurants')



11.2 Task 3 Objective 2: Determine if higher-priced restaurants are more likely to offer these services.

[86]:
```
price_range_restaurants_with_delivery = data[data['Has Online␣
↪delivery']=='Yes']['Price range'].value_counts()
price_range_restaurants_with_delivery
```

[86]:
```
Price range
2    1286
1     701
3     411
4      53
Name: count, dtype: int64
```

[87]:
```
sn.barplot(x=price_range_restaurants_with_delivery.index,␣
↪y=price_range_restaurants_with_delivery.values)
mp.xlabel('Has Online delivery')
```

```
mp.ylabel('Price range')
mp.title('Price range restaurants offer online delivery')
```

[87]: Text(0.5, 1.0, 'Price range restaurants offer online delivery')

[88]:
```
price_range_restaurants_with_table_booking = data[data['Has Table␣
↪booking']=='Yes']['Price range'].value_counts()
price_range_restaurants_with_table_booking
```

[88]:
```
Price range
3    644
4    274
2    239
1      1
Name: count, dtype: int64
```

[89]:
```
sn.barplot(x=price_range_restaurants_with_table_booking.index,␣
↪y=price_range_restaurants_with_table_booking.values)
mp.xlabel('Has Online Delivery')
mp.ylabel('Price range')
```

```
mp.title('Price range restaurants offer table booking')
```

[89]: Text(0.5, 1.0, 'Price range restaurants offer table booking')