In [167…
```
#import necessary library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import geopandas as gpd
import contextily as ctx
import warnings
warnings.filterwarnings('ignore')
```

In [168…
```
#load and read dataset
df=pd.read_csv('Dataset.csv')
df
```

Out[168]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines
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...	121.027535	14.565443	French, Japanese, Desserts
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...	121.014101	14.553708	Japanese
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...	121.056831	14.581404	Seafood, Asian, Filipino, Indian
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...	121.056475	14.585318	Japanese, Sushi
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...	121.057508	14.584450	Japanese, Korean
...	...	...	...	...	...	...	...	...	...	...
9546	5915730	Namlı Gurme	208	��stanbul	Kemanke�� Karamustafa Pa��a Mahallesi, Rıhtım ...	Karak�_y	Karak�_y, ��stanbul	28.977392	41.022793	Turkish
9547	5908749	Ceviz A��acı	208	��stanbul	Ko��uyolu Mahallesi, Muhittin ��st�_nda�� Cadd...	Ko��uyolu	Ko��uyolu, ��stanbul	29.041297	41.009847	World Cuisine, Patisserie, Cafe
9548	5915807	Huqqa	208	��stanbul	Kuru�_e��me Mahallesi, Muallim Naci Caddesi, N...	Kuru�_e��me	Kuru�_e��me, ��stanbul	29.034640	41.055817	Italian, World Cuisine
9549	5916112	A���k Kahve	208	��stanbul	Kuru�_e��me Mahallesi, Muallim Naci Caddesi, N...	Kuru�_e��me	Kuru�_e��me, ��stanbul	29.036019	41.057979	Restaurant Cafe
9550	5927402	Walter's Coffee Roastery	208	��stanbul	Cafea��a Mahallesi, Bademaltı Sokak, No 21/B, ...	Moda	Moda, ��stanbul	29.026016	40.984776	Cafe

9551 rows × 21 columns

In [169…
```
#Check no of rows and column
print("Number of rows =",df.shape[0])
print("Number of column =",df.shape[1])
```

```
Number of rows = 9551
Number of column = 21
```

In [170…
```
#Checking infomation of all data
df.info()
```

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

In [171…

```
#Check null value present in our datasets
df.isnull().sum()
```

Out[171]:

```
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 [172…

```
#Drop null value in our datasets
df.dropna(inplace=True)
```

In [173…

```
#Again check null value present in our datasets
pd.isnull(df).sum()
```

Out[173]:

```
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 [174…

```
#descriptive stat
df.describe()
```

Out[174]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	Votes
count	9.542000e+03	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000
mean	9.043301e+06	18.179208	64.274997	25.848532	1200.326137	1.804968	2.665238	156.772060
std	8.791967e+06	56.451600	41.197602	11.010094	16128.743876	0.905563	1.516588	430.203324
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	0.000000
25%	3.019312e+05	1.000000	77.081565	28.478658	250.000000	1.000000	2.500000	5.000000
50%	6.002726e+06	1.000000	77.192031	28.570444	400.000000	2.000000	3.200000	31.000000
75%	1.835260e+07	1.000000	77.282043	28.642711	700.000000	2.000000	3.700000	130.000000
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	10934.000000

In [175…

```
#Number of unique value in our datasets
df.nunique()
```

Out[175]:

```
Restaurant ID           9542
Restaurant Name         7437
Country Code              15
City                     140
Address                 8910
Locality                1206
Locality Verbose        1263
Longitude               8111
Latitude                8668
Cuisines                1825
Average Cost for two     140
Currency                  12
Has Table booking          2
Has Online delivery        2
Is delivering now          2
Switch to order menu       1
Price range                4
Aggregate rating          33
Rating color               6
Rating text                6
Votes                   1012
dtype: int64
```

In [176…

```
#check duplicate value of sum in our datasets
print(df.duplicated().sum())
```

```
0
```

In [177…

```
df.columns
```

Out[177]:

```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       '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')
```

Identify Categorical & Numerical feature present in our datasets.

In [178…

```
#These are our categorical feature
[feature for feature in df.columns if df[feature].dtype=='O']
```

Out[178]:

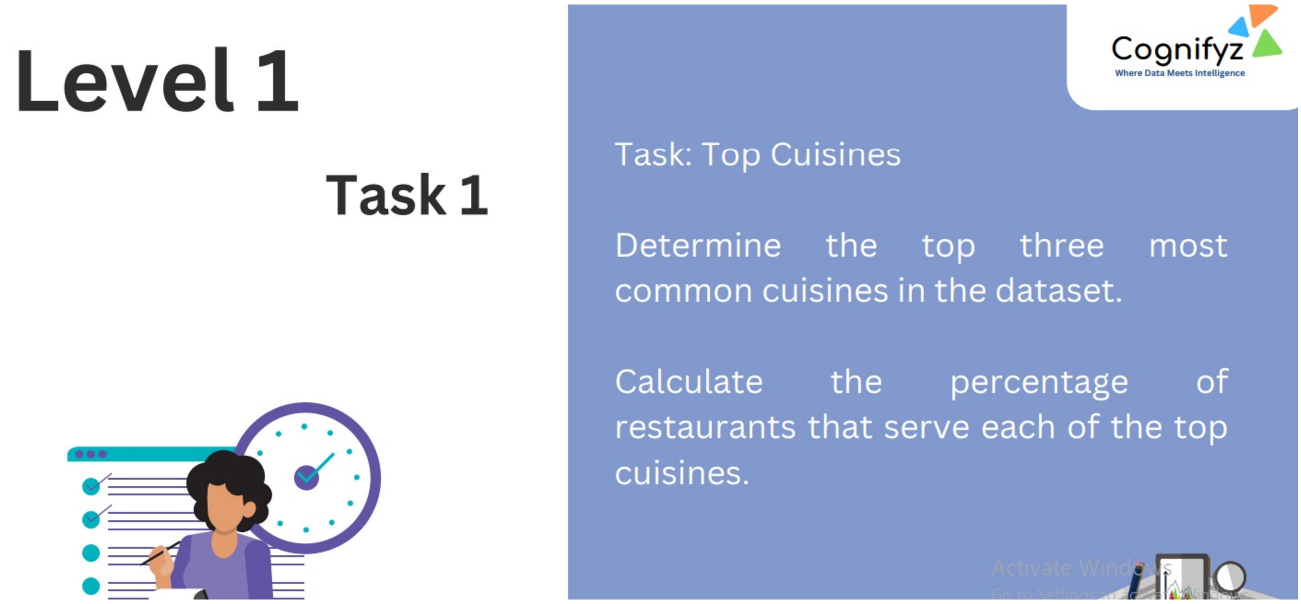
```
['Restaurant Name',
 'City',
 'Address',
 'Locality',
 'Locality Verbose',
 'Cuisines',
 'Currency',
 'Has Table booking',
 'Has Online delivery',
 'Is delivering now',
 'Switch to order menu',
 'Rating color',
 'Rating text']
```

In [179…

```
#These are our Numerical feature
[feature for feature in df.columns if df[feature].dtype!='O']
```

Out[179]:

```
['Restaurant ID',
 'Country Code',
 'Longitude',
 'Latitude',
 'Average Cost for two',
 'Price range',
 'Aggregate rating',
 'Votes']
```



Task: Top Cuisines

1. Determine the top three most common cuisines in the dataset.
2. Calculate the percentage of restaurants that serve each of the top cuisines.

Task

1. Determine the top three most common cuisines in the dataset

In [180…
```
df_cuisines=df['Cuisines'].str.split(', ').explode().value_counts()
```

In [181…
```
top_cuisin=df_cuisines.head(3)
top_cuisin
```

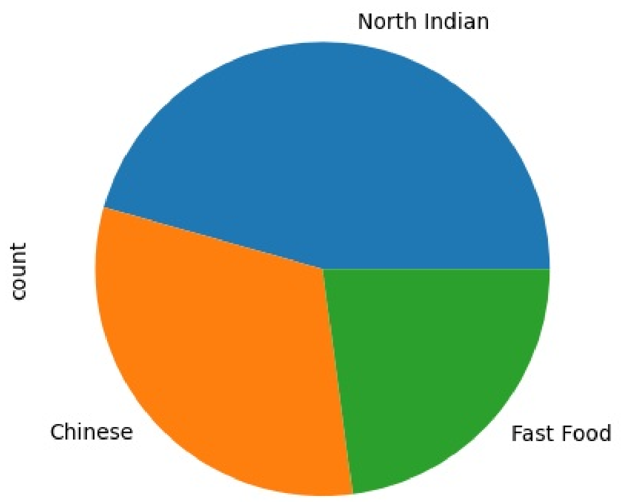
Out[181]:
```
Cuisines
North Indian    3960
Chinese         2735
Fast Food       1986
Name: count, dtype: int64
```

Insights:-

Dominant Cuisines: North Indian cuisine is the most popular in the dataset, followed by Chinese and Fast Food.

In [182…
```
#visualize by using pie chart
top_cuisin.plot(kind='pie')
```

Out[182]:
```
<Axes: ylabel='count'>
```



Task

1. Calculate the percentage of restaurants that serve each of the top cuisines

In [183…

```python
total_restaurant = len(df)
total_percentage=(top_cuisin/total_restaurant)*100
total_percentage
```

Out[183]:

```
Cuisines
North Indian    41.500734
Chinese         28.662754
Fast Food       20.813247
Name: count, dtype: float64
```

Insights:-

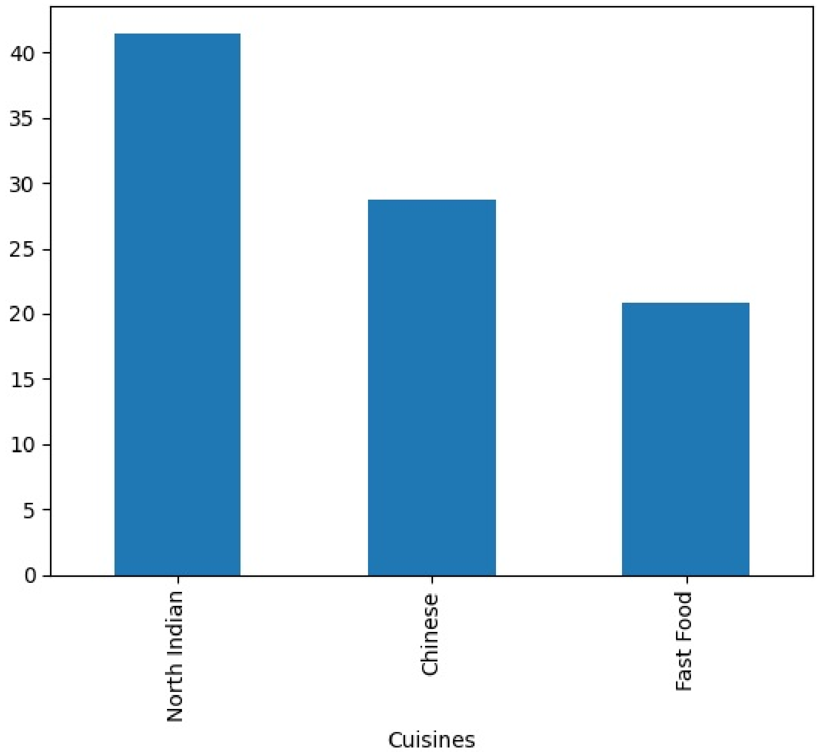
1. Dominance of North Indian Cuisine is a significant portion (41.46%) of restaurants offer North Indian cuisine, indicating its popularity.
2. Chinese and Fast Food cuisines are also widely available, but with lower percentages compared to North Indian.

In [184…

```python
total_percentage.plot(kind='bar')
```

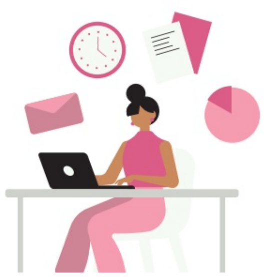
Out[184]:

```
<Axes: xlabel='Cuisines'>
```



Level 1

Task 2



Cognifyz
Where Data Meets Intelligence

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.

Task

1. Identify the city with the highest number of restaurants in the dataset

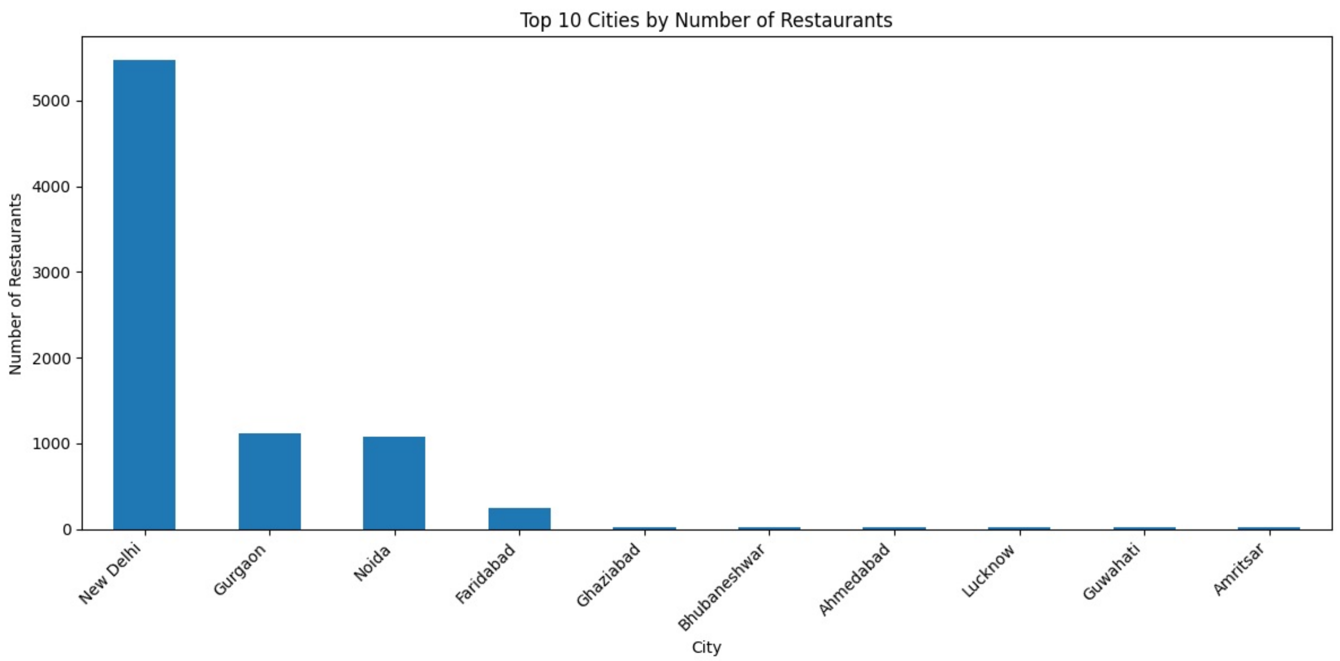
In [185…
```
#count the number of restaurant percity
city_counts = df['City'].value_counts()
city_counts
```

Out[185]:
```
City
New Delhi    5473
Gurgaon      1118
Noida        1080
Faridabad     251
Ghaziabad      25
             ...
Inverloch       1
Mohali          1
Panchkula       1
Bandung         1
Randburg        1
Name: count, Length: 140, dtype: int64
```

In [186…
```
top_city = city_counts.idxmax()
top_city_count = city_counts.max()
print(f"1. The city with the highest number of restaurants is {top_city} with {top_city_count} restaurants.")
```

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

In [187…
```
# Visualize the top 10 cities by number of restaurants
plt.figure(figsize=(12, 6))
city_counts.head(10).plot(kind='bar')
plt.title('Top 10 Cities by Number of Restaurants')
plt.xlabel('City')
plt.ylabel('Number of Restaurants')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Insights:-

1. New Delhi has a significantly higher number of restaurants compared to other cities in the dataset.
2. There's a considerable difference between New Delhi and the other cities in terms of restaurant count.

Task

1. Calculate the average rating for restaurants in each city.

In [188…
```
avg_city_ratting=df.groupby('City')['Aggregate rating'].mean().sort_values(ascending=False)
avg_city_ratting
```

Out[188]:
```
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: 140, dtype: float64
```

In [189…
```python
print("Average ratings for restaurants in each city:")
print(avg_city_ratting.head())
```

```
Average ratings for restaurants in each city:
City
Inner City          4.900000
Quezon City         4.800000
Makati City         4.650000
Pasig City          4.633333
Mandaluyong City    4.625000
Name: Aggregate rating, dtype: float64
```

Insights:

1. City-wise Rating Variation:- There's a significant difference in average ratings across cities, indicating varying restaurant quality standards.
2. Top-rated Cities:- Cities like Inner City, Quezon City, and Makati City have exceptionally high average ratings, suggesting a concentration of high-quality restaurants.
3. Low-rated Cities:- Cities like Noida and Faridabad have relatively lower average ratings, which might indicate areas for improvement in restaurant quality or service.

Task

1. Determine the city with the highest average rating.

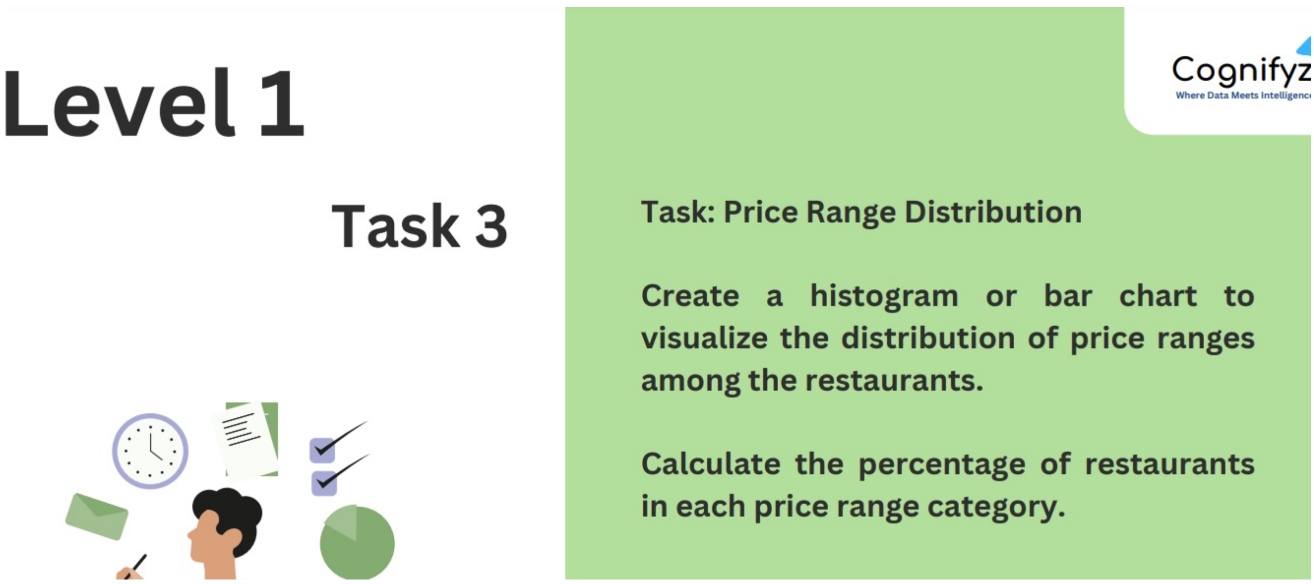
In [190…
```python
hight_city_avg_rating = avg_city_ratting.index[0]
hight_avg_rating = avg_city_ratting.iloc[0]
```

In [191…
```python
print(f"The city with the highest average rating is {hight_city_avg_rating} with an average rating of {hight_av
```

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

Insights:-

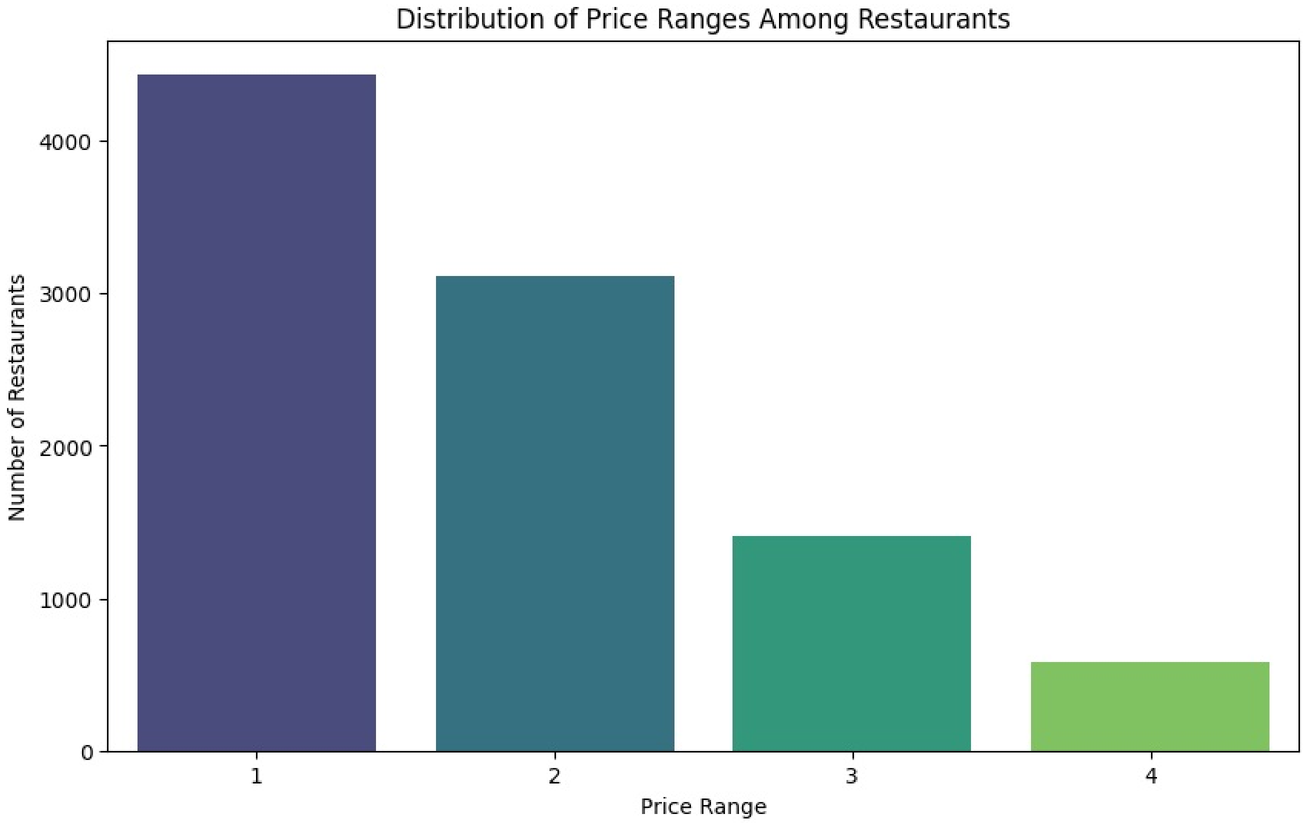
Inner City has the highest average restaurant rating among all cities in the dataset.



Task: Price Range Distribution

1. Create a histogram or bar chart to visualize the distribution of price ranges among the restaurants.

In [192…
```
# Create a bar chart for price range distribution
plt.figure(figsize=(10, 6))
sns.countplot(x='Price range', data=df, palette='viridis')
plt.title('Distribution of Price Ranges Among Restaurants')
plt.xlabel('Price Range')
plt.ylabel('Number of Restaurants')
plt.show()
```



In [193…
```
df['Price range'].value_counts()
```

Out[193]:
```
Price range
1    4438
2    3113
3    1405
4     586
Name: count, dtype: int64
```

Observations:

1. Price Range Distribution: Most restaurants fall in lower price ranges, with fewer in higher tiers.
2. Dominance of Lower Prices: Budget-friendly options dominate the market.

Task

1. Calculate the percentage of restaurants in each price range category

In [194…
```
# Calculate the percentage of restaurants in each price range category
price_range_counts = df['Price range'].value_counts(normalize=True) * 100
```

In [195…
```
# Convert to a DataFrame for better readability
price_range_percentages = price_range_counts.sort_index().reset_index()
price_range_percentages.columns = ['Price Range', 'Percentage']
price_range_percentages
```

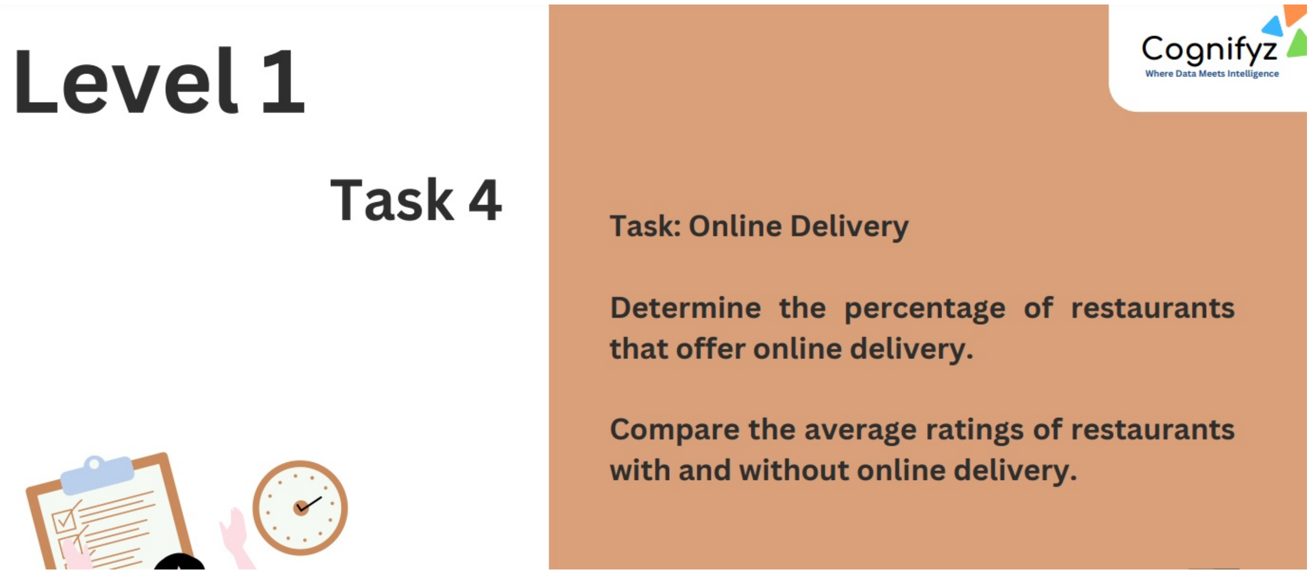
Out[195]:

	Price Range	Percentage
0	1	46.510166
1	2	32.624188
2	3	14.724376
3	4	6.141270

Insight:

Insight:

1. The majority of restaurants fall into price range 1, followed by price range 2.
2. As the price range increases, the percentage of restaurants decreases significantly.



Task

1. Determine the percentage of restaurants that offer online delivery.

In [196…

```
total_count=df[['Restaurant ID','Restaurant Name','Has Online delivery']].groupby(['Restaurant ID','Restaurant 
total_count
```

Out[196]:

	Restaurant ID	Restaurant Name	Has Online delivery	total
0	53	Amber	Yes	1
1	55	Berco's	Yes	1
2	60	Colonel's Kababz	No	1
3	64	Diva - The Italian Restaurant	Yes	1
4	65	Drums of Heaven	Yes	1
...	...	...	...	...
9537	18499493	Zombiez	No	1
9538	18500618	Veg. Darbar	No	1
9539	18500628	Grill & Cafe	No	1
9540	18500639	Chandni Chowk 2 China	No	1
9541	18500652	Mahek By Greenz	No	1

9542 rows × 4 columns

In [197…

```
delivery_online_percentage=(df['Has Online delivery']=="Yes").mean()*100
print(f"Percentage of restaurants offering online delivery: {delivery_online_percentage:.2f}%")
```

```
Percentage of restaurants offering online delivery: 25.69%
```

Ovservation

Percentage of restaurants offering online delivery: 25.66%.This means that about a quarter of the restaurants in the dataset offer online delivery services

Task

1. Compare the average ratings of restaurants with and without online delivery.

In [198…

```
#Compare rating
avg_rating_with_delivery = df[df['Has Online delivery']=="Yes"]['Aggregate rating'].mean()
avg_rating_without_delivery = df[df['Has Online delivery']=="No"]['Aggregate rating'].mean()
print("Average rating of restaurants with online delivery =",avg_rating_with_delivery)
print("Average rating of restaurants without online delivery =",avg_rating_without_delivery)
```

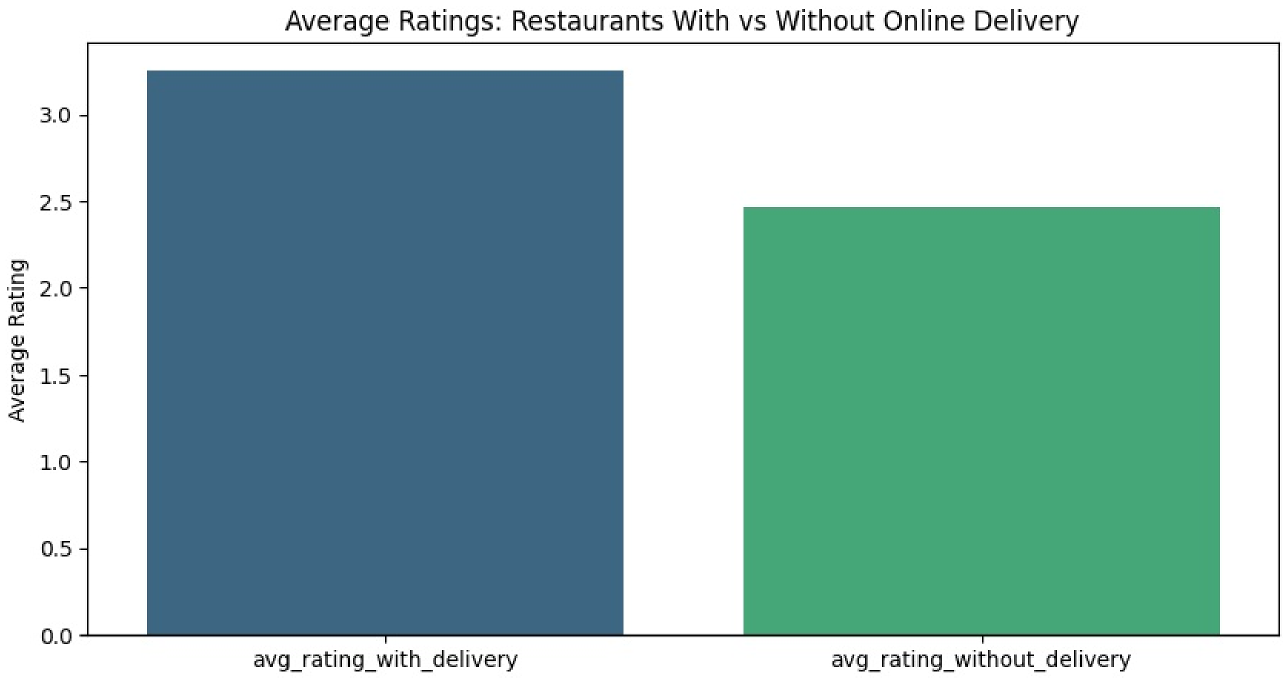
```
Average rating of restaurants with online delivery = 3.2488372093023257
Average rating of restaurants without online delivery = 2.4635171343957127
```

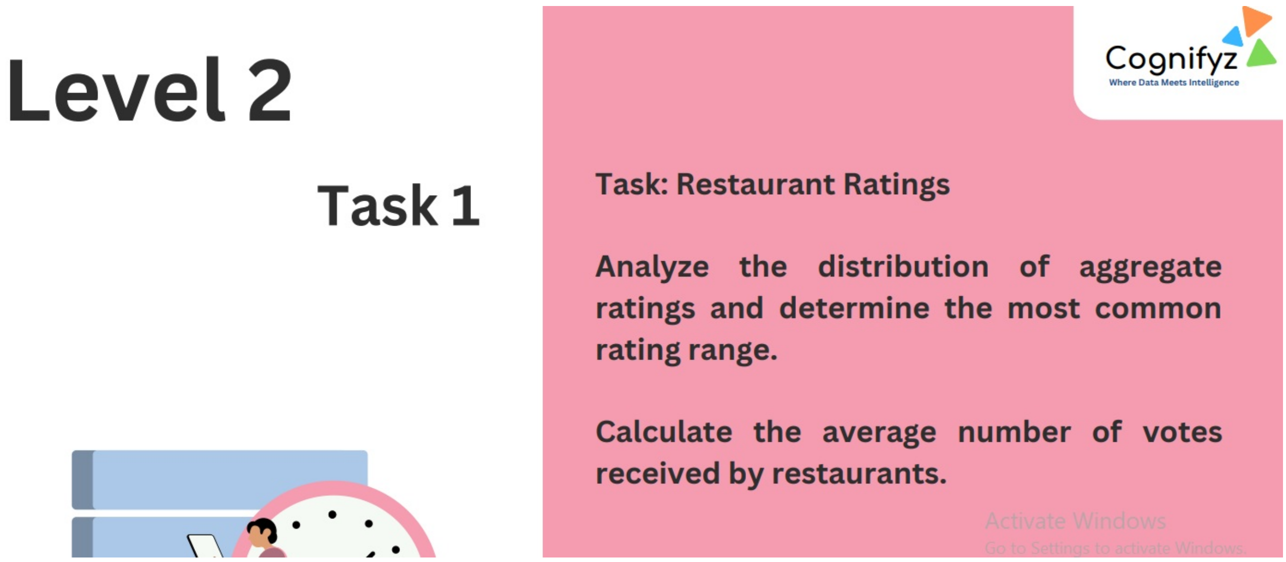
Ovservation

1. Average rating of restaurants with online delivery: 3.24
2. Average rating of restaurants without online delivery: 2.46

In [199…

```
# Create a bar plot to visualize the comparison
plt.figure(figsize=(10,5))
sns.barplot(x=['avg_rating_with_delivery','avg_rating_without_delivery'],y=[avg_rating_with_delivery,avg_rating
plt.title('Average Ratings: Restaurants With vs Without Online Delivery')
plt.ylabel('Average Rating')
plt.show()
```



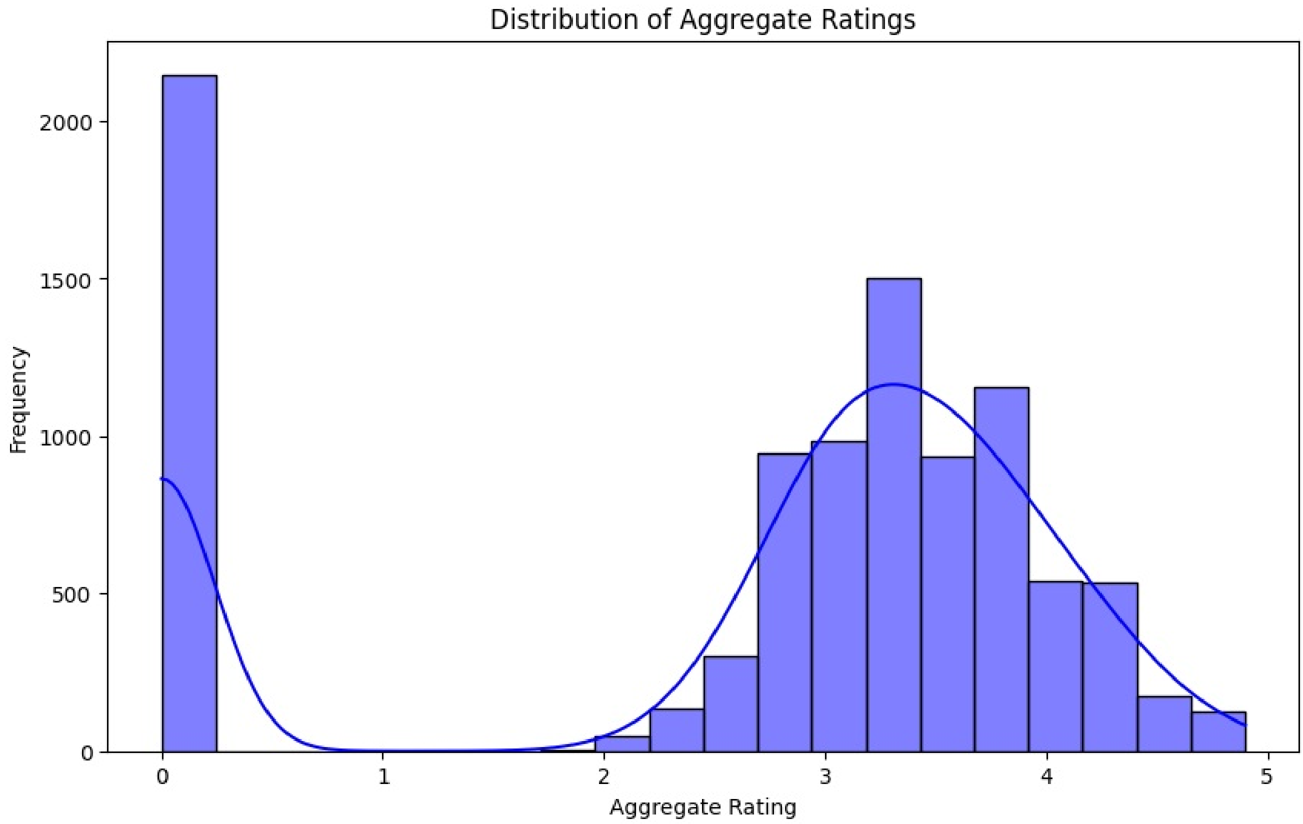


Task

1. Analyze the distribution of aggregate ratings and determine the most common rating range

In [200…

```
# Analyze the distribution of aggregate ratings
plt.figure(figsize=(10, 6))
sns.histplot(df['Aggregate rating'], bins=20, kde=True, color='blue')
plt.title('Distribution of Aggregate Ratings')
plt.xlabel('Aggregate Rating')
plt.ylabel('Frequency')
plt.show()
```



In [201…
```
# Determine the most common rating range
rating_distribution = df['Aggregate rating'].value_counts().sort_index()
rating_distribution
```

Out[201]:
```
Aggregate rating
0.0    2148
1.8       1
1.9       2
2.0       7
2.1      15
2.2      27
2.3      47
2.4      87
2.5     110
2.6     191
2.7     250
2.8     315
2.9     381
3.0     468
3.1     519
3.2     522
3.3     483
3.4     495
3.5     480
3.6     458
3.7     427
3.8     399
3.9     332
4.0     266
4.1     274
4.2     221
4.3     174
4.4     143
4.5      95
4.6      78
4.7      41
4.8      25
4.9      61
Name: count, dtype: int64
```

In [202…
```
most_common_rating=rating_distribution.idxmax()
print(most_common_rating)
```

```
0.0
```

In [203…
```
#This way we also find out the most common rating
most_common_rating = df["Aggregate rating"].mode()[0]
print(most_common_rating)
```

```
0.0
```

Task

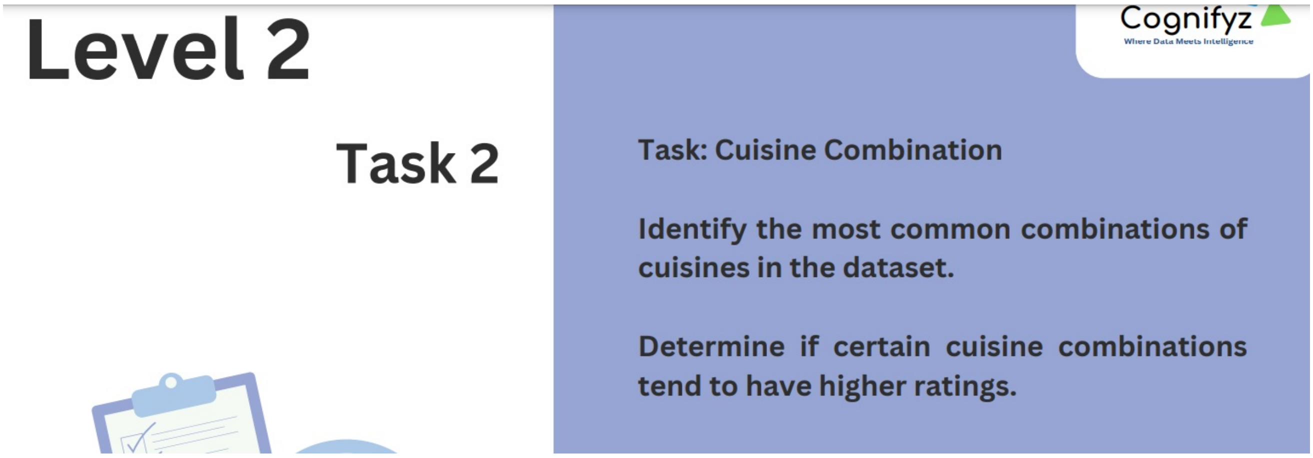
1. Calculate the average number of votes received by restaurants.

In [204…
```
average_rating = df["Votes"].mean()
print(f"Average number of votes received by restaurants: {average_rating:.2f}")
```

```
Average number of votes received by restaurants: 156.77
```

Insight:

The average number of votes received by restaurants is 156.77



Task

1. Identify the most common combinations of cuisines in the dataset

In [205…
```
most_common_combinations=df["Cuisines"].str.split(', ').value_counts()
most_common_combinations.head(10)
```

Out[205]:
```
Cuisines
[North Indian]                      936
[North Indian, Chinese]             511
[Chinese]                           354
[Fast Food]                         354
[North Indian, Mughlai]             334
[Cafe]                              299
[Bakery]                            218
[North Indian, Mughlai, Chinese]    197
[Bakery, Desserts]                  170
[Street Food]                       149
Name: count, dtype: int64
```

Insight

1. North Indian cuisine is the most popular This is evident from its high frequency in the dataset.
2. Diverse cuisine options: While North Indian cuisine dominates, there's a variety of other cuisines available, including Chinese, Fast Food, and various regional options.

Task

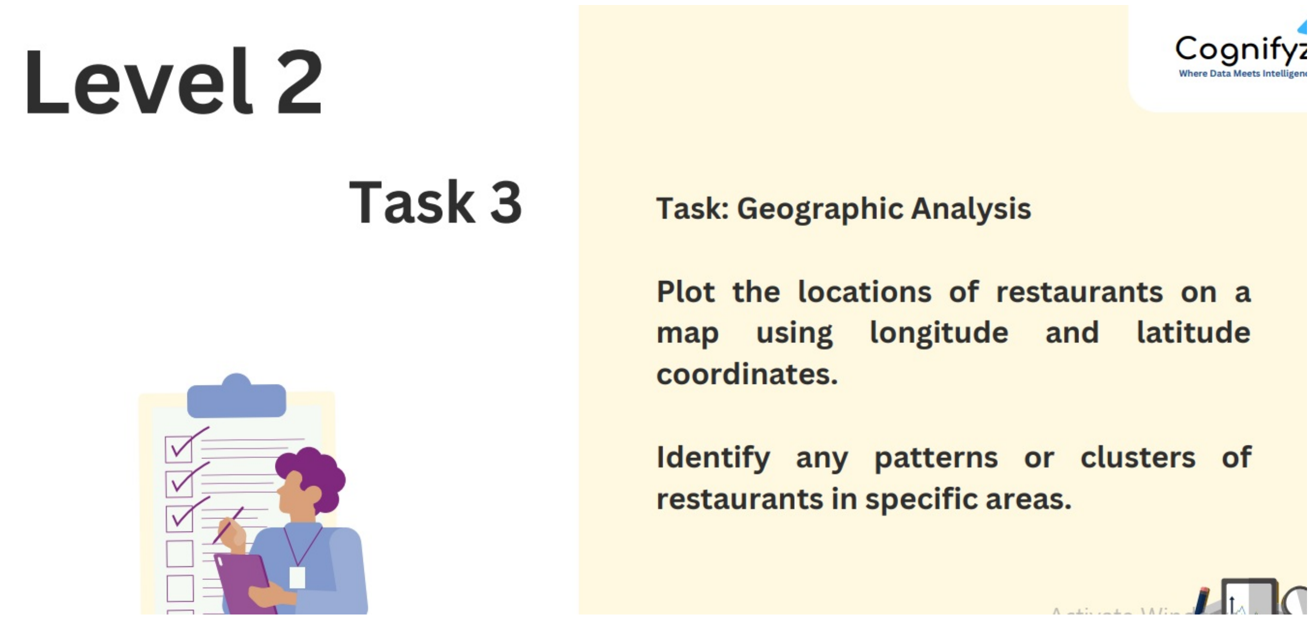
1. Determine if certain cuisine combinations tend to have higher ratings.

In [206…
```
# Split the cuisines into individual entries
df["Cuisines"]=df["Cuisines"].str.split(', ')
# # Explode the dataframe so each cuisine combination is a separate row
to_explode = df.explode("Cuisines")
```

In [207…
```
cuisines_rating = to_explode.groupby("Cuisines")["Aggregate rating"].mean().reset_index()
```

In [208…
```
# Sort by the mean rating in descending order
cuisines_rating = cuisines_rating.sort_values(by='Aggregate rating',ascending=False)
# Display the top 10 cuisines with the highest average ratings
print(cuisines_rating.head())
```

```
     Cuisines  Aggregate rating
130     Sunda          4.900000
26      B�_rek         4.700000
132  Taiwanese         4.650000
112     Ramen          4.500000
43    Dim Sum          4.466667
```



Task

14.Plot the locations of restaurants on a map using longitude and latitude coordinates.

In [209…

```python
# Assuming df is your DataFrame containing the Latitude and Longitude columns
# Create a GeoDataFrame from the DataFrame
geometry = gpd.points_from_xy(df["Longitude"], df["Latitude"])  # Note: Longitude first, then Latitude
gdf = gpd.GeoDataFrame(df, geometry=geometry)
geometry,gdf
```

Out[209]:

```
(<GeometryArray>
 [<POINT (121.028 14.565)>, <POINT (121.014 14.554)>, <POINT (121.057 14.581)>,
  <POINT (121.056 14.585)>, <POINT (121.058 14.584)>, <POINT (121.056 14.584)>,
   <POINT (120.98 14.531)>,  <POINT (120.979 14.54)>,  <POINT (120.98 14.553)>,
  <POINT (121.057 14.572)>,
  ...
   <POINT (29.057 41.105)>,   <POINT (29.023 40.99)>,  <POINT (28.978 41.025)>,
   <POINT (28.978 41.023)>,  <POINT (28.981 41.026)>,  <POINT (28.977 41.023)>,
    <POINT (29.041 41.01)>,  <POINT (29.035 41.056)>,  <POINT (29.036 41.058)>,
   <POINT (29.026 40.985)>]
 Length: 9542, dtype: geometry,
       Restaurant ID           Restaurant Name  Country Code              City  \
 0           6317637          Le Petit Souffle           162       Makati City
 1           6304287          Izakaya Kikufuji           162       Makati City
 2           6300002    Heat - Edsa Shangri-La           162  Mandaluyong City
 3           6318506                      Ooma           162  Mandaluyong City
 4           6314302               Sambo Kojin           162  Mandaluyong City
 ...             ...                       ...           ...               ...
 9546        5915730               Naml\ Gurme           208         ��stanbul
 9547        5908749              Ceviz A��ac\           208         ��stanbul
 9548        5915807                     Huqqa           208         ��stanbul
 9549        5916112               A���k Kahve           208         ��stanbul
 9550        5927402  Walter's Coffee Roastery           208         ��stanbul

                                                 Address  \
 0     Third Floor, Century City Mall, Kalayaan Avenu...
 1     Little Tokyo, 2277 Chino Roces Avenue, Legaspi...
 2     Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...
 3     Third Floor, Mega Fashion Hall, SM Megamall, O...
 4     Third Floor, Mega Atrium, SM Megamall, Ortigas...
 ...                                                 ...
 9546  Kemanke�� Karamustafa Pa��a Mahallesi, R\ht\m ...
 9547  Ko��uyolu Mahallesi, Muhittin ��st�_nda�� Cadd...
 9548  Kuru�_e��me Mahallesi, Muallim Naci Caddesi, N...
 9549  Kuru�_e��me Mahallesi, Muallim Naci Caddesi, N...
 9550  Cafea��a Mahallesi, Bademalt\ Sokak, No 21/B, ...

                                         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
```

```
...                                          ...
9546                                    Karak�_y
9547                                  Ko��uyolu
9548                                Kuru�_e��me
9549                                Kuru�_e��me
9550                                        Moda

                                    Locality Verbose   Longitude  \
0     Century City Mall, Poblacion, Makati City, Mak...  121.027535
1     Little Tokyo, Legaspi Village, Makati City, Ma...  121.014101
2     Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...  121.056831
3     SM Megamall, Ortigas, Mandaluyong City, Mandal...  121.056475
4     SM Megamall, Ortigas, Mandaluyong City, Mandal...  121.057508
...                                                ...         ...
9546                              Karak�_y, ��stanbul   28.977392
9547                             Ko��uyolu, ��stanbul   29.041297
9548                           Kuru�_e��me, ��stanbul   29.034640
9549                           Kuru�_e��me, ��stanbul   29.036019
9550                                  Moda, ��stanbul   29.026016

       Latitude                          Cuisines  ...  Has Table booking  \
0     14.565443        [French, Japanese, Desserts]  ...                Yes
1     14.553708                          [Japanese]  ...                Yes
2     14.581404  [Seafood, Asian, Filipino, Indian]  ...                Yes
3     14.585318                  [Japanese, Sushi]  ...                 No
4     14.584450                 [Japanese, Korean]  ...                Yes
...         ...                                 ...  ...                ...
9546  41.022793                           [Turkish]  ...                 No
9547  41.009847   [World Cuisine, Patisserie, Cafe]  ...                 No
9548  41.055817            [Italian, World Cuisine]  ...                 No
9549  41.057979                   [Restaurant Cafe]  ...                 No
9550  40.984776                              [Cafe]  ...                 No

     Has Online delivery Is delivering now Switch to order menu Price range  \
0                     No                No                   No           3
1                     No                No                   No           3
2                     No                No                   No           4
3                     No                No                   No           4
4                     No                No                   No           4
...                  ...               ...                  ...         ...
9546                  No                No                   No           3
9547                  No                No                   No           3
9548                  No                No                   No           4
9549                  No                No                   No           4
9550                  No                No                   No           2

     Aggregate rating  Rating color  Rating text Votes  \
0                 4.8    Dark Green    Excellent   314
1                 4.5    Dark Green    Excellent   591
2                 4.4         Green    Very Good   270
3                 4.9    Dark Green    Excellent   365
4                 4.8    Dark Green    Excellent   229
...               ...           ...          ...   ...
9546              4.1         Green    Very Good   788
9547              4.2         Green    Very Good  1034
9548              3.7        Yellow         Good   661
9549              4.0         Green    Very Good   901
9550              4.0         Green    Very Good   591

                         geometry
0     POINT (121.02754 14.56544)
1      POINT (121.0141 14.55371)
2      POINT (121.05683 14.5814)
3     POINT (121.05648 14.58532)
4     POINT (121.05751 14.58445)
...                          ...
9546   POINT (28.97739 41.02279)
9547    POINT (29.0413 41.00985)
9548   POINT (29.03464 41.05582)
9549   POINT (29.03602 41.05798)
9550   POINT (29.02602 40.98478)

[9542 rows x 22 columns])
```

In [210…

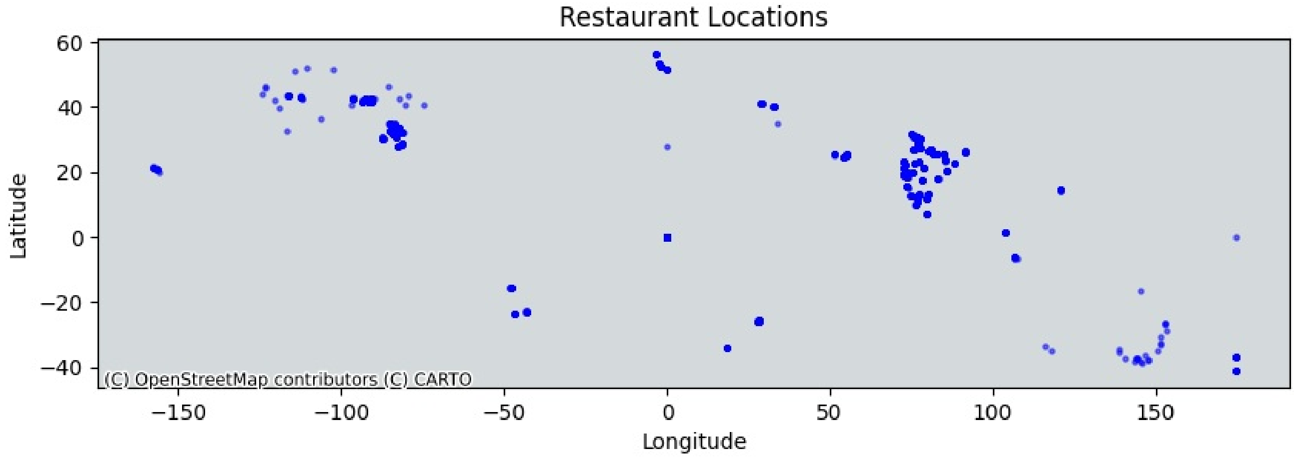
```python
# Set the coordinate reference system to WGS84
gdf.set_crs(epsg=4326, inplace=True)

# Plot the locations
fig, ax = plt.subplots(figsize=(10, 10))
gdf.plot(ax=ax, color='blue', markersize=5, alpha=0.5)

# Add a basemap
gdf = gdf.to_crs(epsg=3857)
ctx.add_basemap(ax, source=ctx.providers.CartoDB.Positron)

# Show the plot
plt.title('Restaurant Locations')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
```

```
plt.show()
```



Task

1. Identify any patterns or clusters of restaurants in specific areas.

In [211…

```
# Count the number of restaurants in each city
city_counts = df['City'].value_counts()
print("Number of restaurants in each city:")
print(city_counts)
```

```
Number of restaurants in each city:
City
New Delhi    5473
Gurgaon      1118
Noida        1080
Faridabad     251
Ghaziabad      25
             ...
Inverloch       1
Mohali          1
Panchkula       1
Bandung         1
Randburg        1
Name: count, Length: 140, dtype: int64
```

In [212…

```
grouped_data = df.groupby(['City', 'Locality']).size().reset_index(name='Restaurant Count')
grouped_data.head()
```

Out[212]:

	City	Locality	Restaurant Count
0	Abu Dhabi	Abu Dhabi Mall, Tourist Club Area (Al Zahiyah)	2
1	Abu Dhabi	Al Dhafrah	2
2	Abu Dhabi	Al Mushrif	1
3	Abu Dhabi	Al Wahda Mall, Al Wahda	2
4	Abu Dhabi	Crowne Plaza Abu Dhabi, Al Markaziya	1

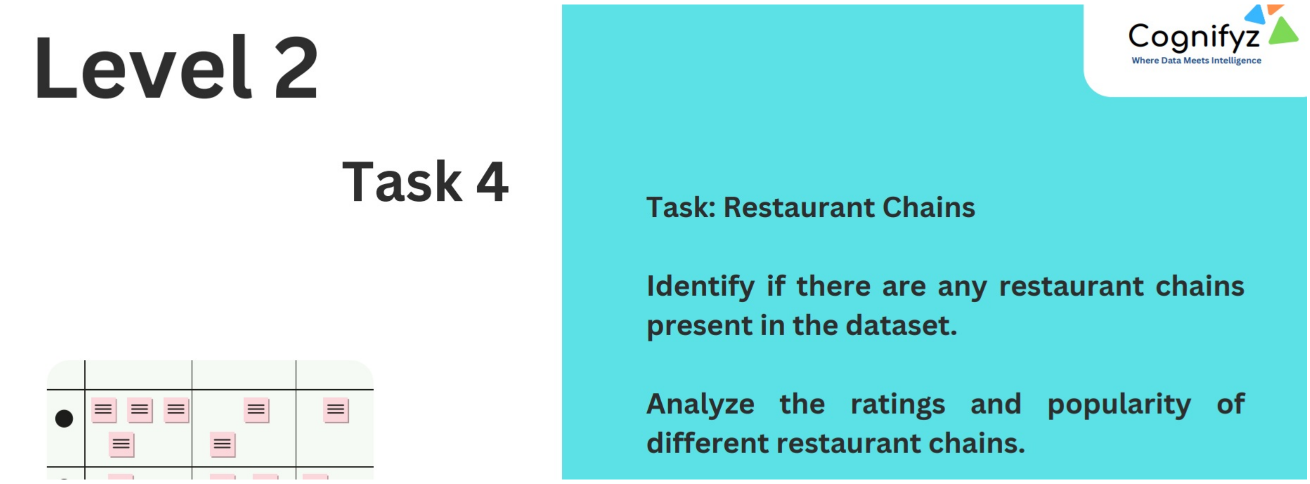
In [213…

```
# Create a scatter plot of restaurant locations
plt.figure(figsize=(12, 8))
sns.scatterplot(data=df, x='Longitude', y='Latitude', hue='City', palette='deep')
plt.title('Restaurant Locations')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend(title='City', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```


−20
−40
−150
−100
−50
0
50
100
150
Longitude
Balingup
Beechworth
Boise
Cedar Rapids/Iowa City
Chatham-Kent
Clatskanie
Cochrane
Columbus
Consort
Dalton
Davenport
Des Moines
Dicky Beach
Dubuque
East Ballina
Fernley
Flaxton
Forrest
Gainesville
Hepburn Springs
Huskisson
Inverloch
Lakes Entrance
Lakeview
Lincoln
Lorn
Macedon
Macon
Mayfield
Mc Millan
Middleton Beach
Monroe
Montville
Ojo Caliente
Orlando
Palm Cove
Paynesville
Penola
Pensacola
Phillip Island
Pocatello
Potrero
Princeton
Rest of Hawaii
Savannah
Singapore
Sioux City
Tampa Bay
Tanunda
Trentham East
Valdosta
Vernonia
Victor Harbor
Vineland Station
Waterloo
Weirton
Winchester Bay
Yorkton
Abu Dhabi
Dubai
Sharjah
Agra
Ahmedabad
Allahabad
Amritsar
Aurangabad
Bangalore
Bhopal
Bhubaneshwar
Chandigarh
Chennai
Coimbatore
Dehradun
Faridabad
Ghaziabad
Goa
Gurgaon
Guwahati
Hyderabad
Indore
Jaipur
Kanpur
Kochi
Kolkata
Lucknow
Ludhiana
Mangalore
Mohali
Mumbai
Mysore
Nagpur
Nashik
New Delhi
Noida
Panchkula
Patna
Puducherry
Pune
Ranchi

Observation

The scatter plot above shows the distribution of restaurants based on their longitude and latitude coordinates, with different colors representing different cities. We can observe some clear clusters of restaurants in specific areas.

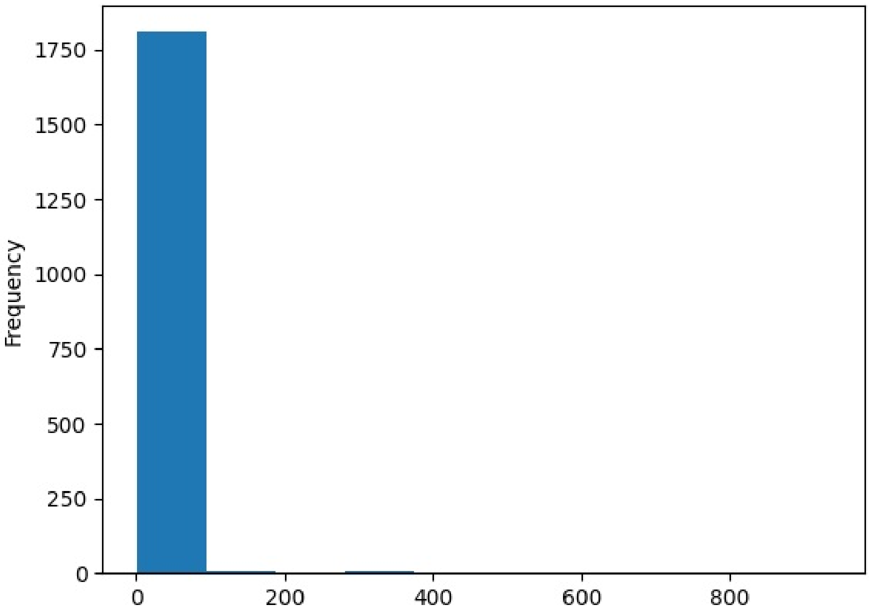


Task

1. Identify if there are any restaurant chains present in the dataset.

In [214…
```
df['Cuisines'].value_counts().plot(kind="hist")
```

Out[214]:
```
<Axes: ylabel='Frequency'>
```



In [215…

```
df.groupby(df['Cuisines']=='Chinese').size()
```

Out[215]:

```
Cuisines
False    9542
dtype: int64
```

In [216…

```
# Identify restaurant chains by finding restaurants with the same name in different locations<<<<<<<julia file>:
restaurant_chains = df.groupby('Restaurant Name').filter(lambda x: len(x) > 1)
restaurant_chains
```

Out[216]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude
5	18189371	Din Tai Fung	162	Mandaluyong City	Ground Floor, Mega Fashion Hall, SM Megamall, ...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.056314	14.583764
10	6309903	Silantro Fil-Mex	162	Pasig City	75 East Capitol Drive, Kapitolyo, Pasig City	Kapitolyo	Kapitolyo, Pasig City	121.057916	14.567689
12	6318433	Silantro Fil-Mex	162	Quezon City	Second Floor, UP Town Center, Katipunan Avenue...	UP Town Center, Diliman, Quezon City	UP Town Center, Diliman, Quezon City, Quezon City	121.075419	14.649503
35	6601589	Coco Bambu	30	Bras�_lia	Bras�_lia Shopping - Piso 2, SCN 5, Bloco A, A...	Bras�_lia Shopping, Asa Norte	Bras�_lia Shopping, Asa Norte, Bras�_lia	-47.889000	-15.786500
39	6600427	Coco Bambu	30	Bras�_lia	SCES, Trecho 2, Conjunto 13/36, Setor de Clube...	Setor De Clubes Esportivos Sul	Setor De Clubes Esportivos Sul, Bras�_lia	-47.868500	-15.819000
...	...	...	...	...	...	...	...	...	...
9517	6001980	Timboo Cafe	208	Ankara	Armada AVM, Kat -1, Eski��ehir Yolu, No 6, Yen...	S�_���_t�_z�_, Yenimahalle	S�_���_t�_z�_, Yenimahalle, Ankara	32.809247	39.913206
9528	6003668	Timboo Cafe	208	Ankara	Kentpark AVM, Kat -1, Mustafa Kemal Mahallesi,...	Kentpark AVM, ��niversiteler, �ankaya	Kentpark AVM, ��niversiteler, �ankaya, Ankara	32.776255	39.908957
9534	6004089	D�_vero��lu	208	Ankara	Maltepe Mahallesi, Gen�_lik Caddesi, No 28, �a...	Maltepe	Maltepe, Ankara	32.842742	39.922536
9535	6000921	D�_vero��lu	208	Ankara	��mitk�_y Mahallesi, 2432. Cadde (8. Cadde), N...	��mitk�_y	��mitk�_y, Ankara	32.701775	39.891564
9538	5901782	Starbucks	208	��stanbul	Bebek Mahallesi, Cevdetpa��a Caddesi, No 30/A,...	Bebek	Bebek, ��stanbul	29.043734	41.077696

2839 rows × 21 columns

In [217…

```
# Display the restaurant chains
final_data=restaurant_chains[['Restaurant Name', 'City', 'Address']].sort_values(by='Restaurant Name').head(20)
```

In [218…

```
final_data
```

Out[218]:

	Restaurant Name	City	Address
751	10 Downing Street	Bhopal	Third Floor, DB City Mall, Maharana Pratap Nag...
2333	10 Downing Street	Indore	Second Floor, Malhar Mega Mall, AB Road, Schem...
8848	221 B Baker Street	Noida	PG 30, TOT Mall, Sector 62, Noida
8498	221 B Baker Street	Noida	21, Jalvayu Vihar Market, Sector 25, Noida
8039	221 B Baker Street	Noida	10, Brahmaputra Shoping Complex, Sector 29, Noida
5547	34 Parkstreet Lane	New Delhi	Shop 7, Mukherjee Tower, Mukherjee Nagar, New ...
3903	34 Parkstreet Lane	New Delhi	DDA Market,Kala Sarai, Hauz Khas, New Delhi
7701	34, Chowringhee Lane	New Delhi	V 3 S Mall, Laxmi Nagar, New Delhi
5444	34, Chowringhee Lane	New Delhi	B-10, Opposite Metro Pillar 21, Model Town 2, ...
6166	34, Chowringhee Lane	New Delhi	Shop 9, Block A2, DDA Market, Paschim Vihar, N...
2691	34, Chowringhee Lane	New Delhi	115, Central Market, Ashok Vihar Phase 1, New ...
4462	34, Chowringhee Lane	New Delhi	UB-101, Kamla Nagar, New Delhi
4355	34, Chowringhee Lane	New Delhi	61-D, Ground Floor, Ber Sarai Market, Opposite...
6720	34, Chowringhee Lane	New Delhi	C8/354, Sector 8, Rohini, New Delhi
6499	34, Chowringhee Lane	New Delhi	Shop 68, Vasant Place Market, Sector 6, R K Pu...
7972	34, Chowringhee Lane	New Delhi	G-37, Ground Floor, Westend Mall, Janakpuri, N...
4093	34, Chowringhee Lane	New Delhi	23/1, Prem Nagar, Jail Road, New Delhi
7622	34, Chowringhee Lane	New Delhi	Shop 2, Plot 57, Under Dwarka Mor Metro Statio...
7000	34, Chowringhee Lane	New Delhi	93, Opposite Venkateswara College, Satyaniketa...
3431	4700BC Popcorn	New Delhi	Ground Floor, DLF Place Mall, Saket, New Delhi

Task

1. Analyze the ratings and popularity of different restaurant chains.

In [219…

```
# Group by 'Restaurant Name' to analyze ratings and popularity
restaurant_analysis = df.groupby('Restaurant Name').agg({'Aggregate rating':'mean','Votes':'sum'}).reset_index(
restaurant_analysis
```

Out[219]:

	Restaurant Name	Aggregate rating	Votes
0	#45	3.6	209
1	#Dilliwaala6	3.7	124
2	#InstaFreeze	0.0	2
3	#OFF Campus	3.7	216
4	#Urban Caf��	3.3	49
...	...	...	...
7432	t Lounge by Dilmah	3.6	34
7433	tashas	4.1	374
7434	wagamama	3.7	131
7435	{Niche} - Cafe & Bar	4.1	492
7436	�ukura��a Sofras\	4.4	296

7437 rows × 3 columns

In [220…

```
# Rename columns for clarity
restaurant_analysis.columns = ['Restaurant Name', 'Average Rating', 'Total Votes']
```

In [221…

```
# Sort by Total Votes to see the most popular restaurants
popular_restaurants = restaurant_analysis.sort_values(by='Total Votes', ascending=False).head(10)
# Sort by Average Rating to see the top-rated restaurants
top_rated_restaurants = restaurant_analysis.sort_values(by='Average Rating', ascending=False).head(10)
```

In [222…

```
popular_restaurants
```

Out[222]:

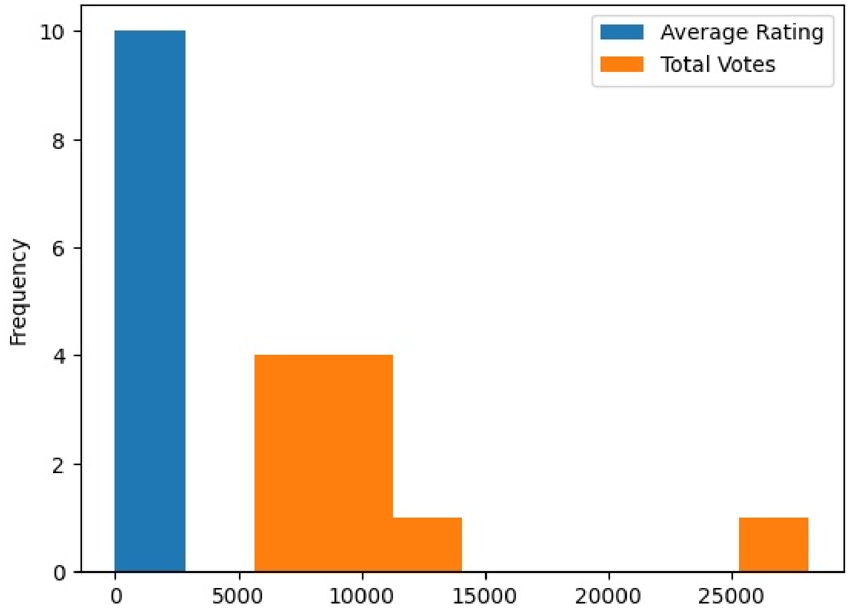
	Restaurant Name	Average Rating	Total Votes
663	Barbeque Nation	4.353846	28142
101	AB's - Absolute Barbecues	4.825000	13400
6935	Toit	4.800000	10934
785	Big Chill	4.475000	10853
2294	Farzi Cafe	4.366667	10098
6980	Truffles	3.950000	9682
1510	Chili's	4.580000	8156
2875	Hauz Khas Social	4.300000	7931
3255	Joey's Pizza	4.250000	7807
4894	Peter Cat	4.300000	7574

In [223…

```
popular_restaurants.plot(kind='hist')
```

Out[223]:

```
<Axes: ylabel='Frequency'>
```



In [224…

```
top_rated_restaurants
```

Out[224]:

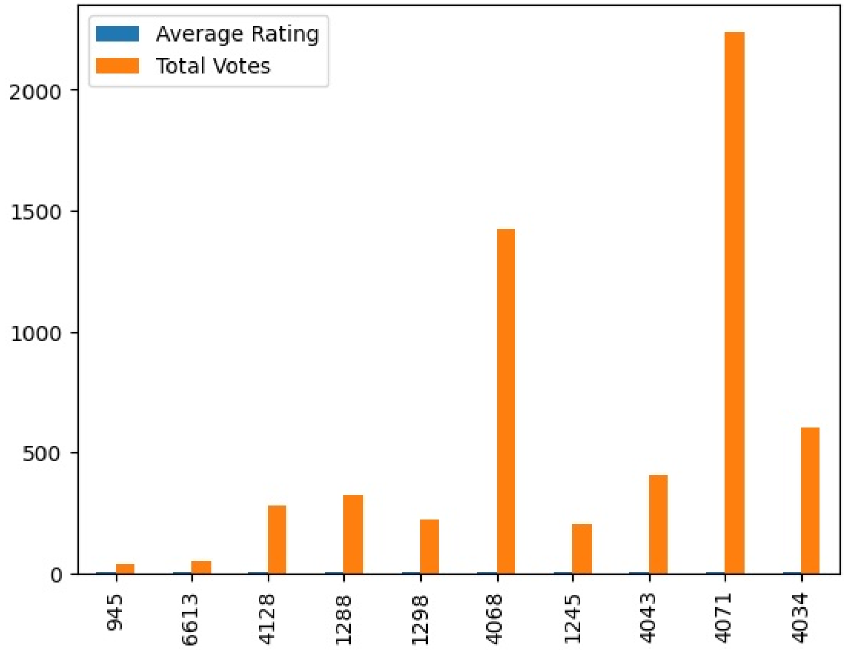
	Restaurant Name	Average Rating	Total Votes
945	Braseiro da G��vea	4.9	40
6613	The Great Indian Pub	4.9	50
4128	Miann	4.9	281
1288	Carnival By Tresind	4.9	322
1298	Caterspoint	4.9	223
4068	Mazzaro's Italian Market	4.9	1424
1245	CakeBee	4.9	200
4043	Masala Library	4.9	408
4071	McGuire's Irish Pub & Brewery	4.9	2238
4034	Marukame Udon	4.9	602

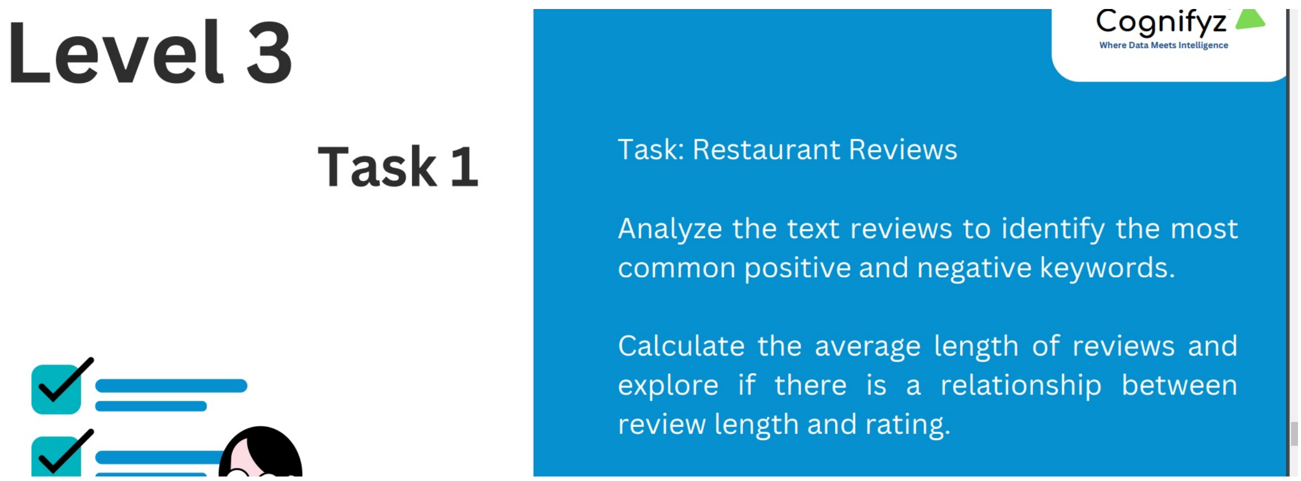
In [225…

```
top_rated_restaurants.plot(kind='bar')
```

Out[225]:

```
<Axes: >
```





Task

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

In [226…

```
from collections import Counter
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import nltk
import warnings
warnings.filterwarnings('ignore')
```

In [227…

```
# Inspect rating distribution
rating_counts = df['Rating text'].value_counts()
print(rating_counts)
```

```
Rating text
Average      3734
Not rated    2148
Good         2096
Very Good    1078
Excellent     300
Poor          186
Name: count, dtype: int64
```

In [228…

```python
# Group by Aggregate rating, Rating color, and Rating text
grouped_ratings = df.groupby(['Aggregate rating', 'Rating color', 'Rating text']).size().reset_index(name='tota
print(grouped_ratings)
```

```
    Aggregate rating Rating color Rating text  total count
0                0.0        White   Not rated         2148
1                1.8          Red        Poor            1
2                1.9          Red        Poor            2
3                2.0          Red        Poor            7
4                2.1          Red        Poor           15
5                2.2          Red        Poor           27
6                2.3          Red        Poor           47
7                2.4          Red        Poor           87
8                2.5       Orange     Average          110
9                2.6       Orange     Average          191
10               2.7       Orange     Average          250
11               2.8       Orange     Average          315
12               2.9       Orange     Average          381
13               3.0       Orange     Average          468
14               3.1       Orange     Average          519
15               3.2       Orange     Average          522
16               3.3       Orange     Average          483
17               3.4       Orange     Average          495
18               3.5       Yellow        Good          480
19               3.6       Yellow        Good          458
20               3.7       Yellow        Good          427
21               3.8       Yellow        Good          399
22               3.9       Yellow        Good          332
23               4.0        Green   Very Good          266
24               4.1        Green   Very Good          274
25               4.2        Green   Very Good          221
26               4.3        Green   Very Good          174
27               4.4        Green   Very Good          143
28               4.5   Dark Green   Excellent           95
29               4.6   Dark Green   Excellent           78
30               4.7   Dark Green   Excellent           41
31               4.8   Dark Green   Excellent           25
32               4.9   Dark Green   Excellent           61
```

In [229…

```python
# Download stopwords
nltk.download('stopwords')
nltk.download('punkt')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\MANISH\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\MANISH\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[229]: True

In [230…

```python
# Function to extract words from text
def extract_words(text):
    # Tokenize text
    tokens = word_tokenize(text.lower())
    # Remove punctuation and stopwords
    tokens = [word for word in tokens if word.isalpha() and word not in stopwords.words('english')]
    return tokens
```

In [231…

```python
# Analyze text reviews and extract keywords
positive_words = Counter()
negative_words = Counter()
```

In [232…

```python
positive_words
```

Out[232]: Counter()

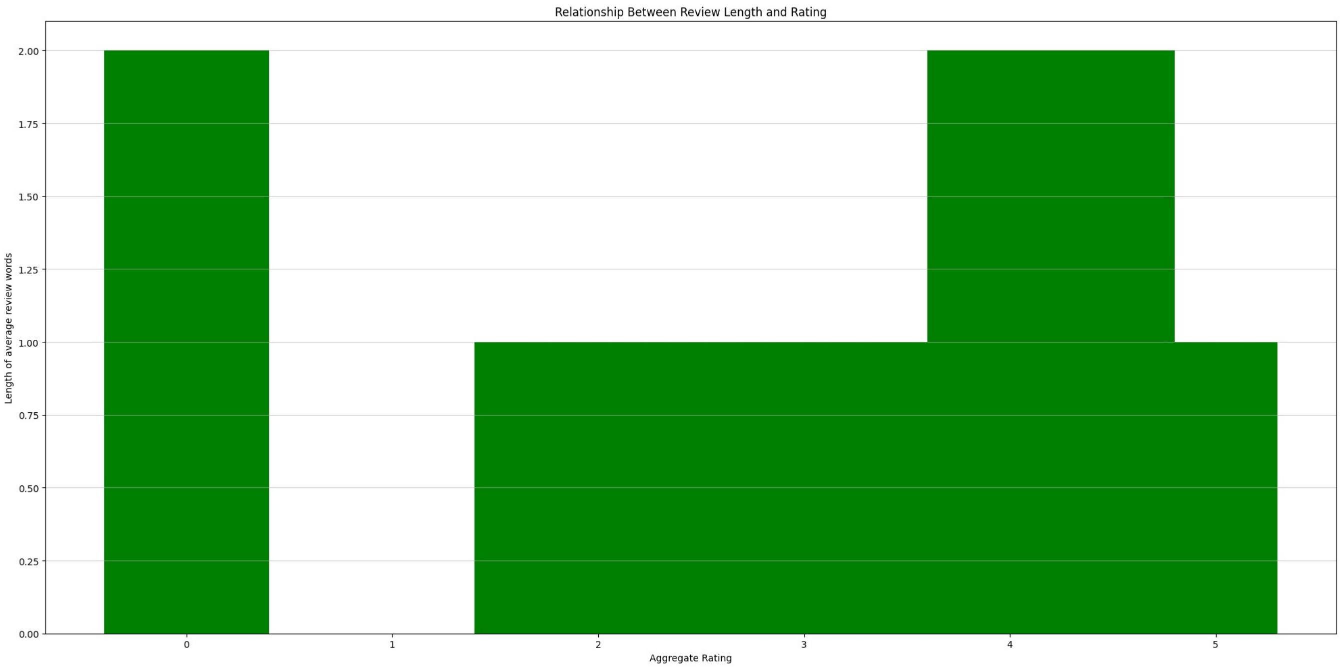
In [233…

```python
negative_words
```

Out[233]: Counter()

Task

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

In [234…
```
df.columns
```
Out[234]:
```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       '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 [235…
```
df['Review length']=df['Rating text'].apply(lambda x:len(x.split()))
```
In [236…
```
avg_Rating_len = df.groupby('Aggregate rating')['Review length'].mean()
avg_Rating_len
```
Out[236]:
```
Aggregate rating
0.0    2.0
1.8    1.0
1.9    1.0
2.0    1.0
2.1    1.0
2.2    1.0
2.3    1.0
2.4    1.0
2.5    1.0
2.6    1.0
2.7    1.0
2.8    1.0
2.9    1.0
3.0    1.0
3.1    1.0
3.2    1.0
3.3    1.0
3.4    1.0
3.5    1.0
3.6    1.0
3.7    1.0
3.8    1.0
3.9    1.0
4.0    2.0
4.1    2.0
4.2    2.0
4.3    2.0
4.4    2.0
4.5    1.0
4.6    1.0
4.7    1.0
4.8    1.0
4.9    1.0
Name: Review length, dtype: float64
```
In [237…
```
plt.figure(figsize=(20,10))
plt.bar(avg_Rating_len.index,avg_Rating_len.values,color='green')
plt.xlabel('Aggregate Rating')
plt.ylabel('Length of average review words')
plt.title("Relationship Between Review Length and Rating")
plt.grid(axis='y',linestyle='-',alpha=0.6)
plt.tight_layout()
plt.show()
```


Level 3

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.

Task

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

In [238…

```
hi_vote=df.groupby('Restaurant Name')["Votes"].sum().reset_index(name="vote count")
```

In [239…

```
sort_votes=hi_vote.sort_values(by='vote count',ascending=False)
top_10 = sort_votes.head(10)
top_10
```

Out[239]:

	Restaurant Name	vote count
663	Barbeque Nation	28142
101	AB's - Absolute Barbecues	13400
6935	Toit	10934
785	Big Chill	10853
2294	Farzi Cafe	10098
6980	Truffles	9682
1510	Chili's	8156
2875	Hauz Khas Social	7931
3255	Joey's Pizza	7807
4894	Peter Cat	7574

Task

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

In [240…

```
# Convert Votes to numeric (in case it is not)
df['Votes'] = pd.to_numeric(df['Votes'], errors='coerce')
```

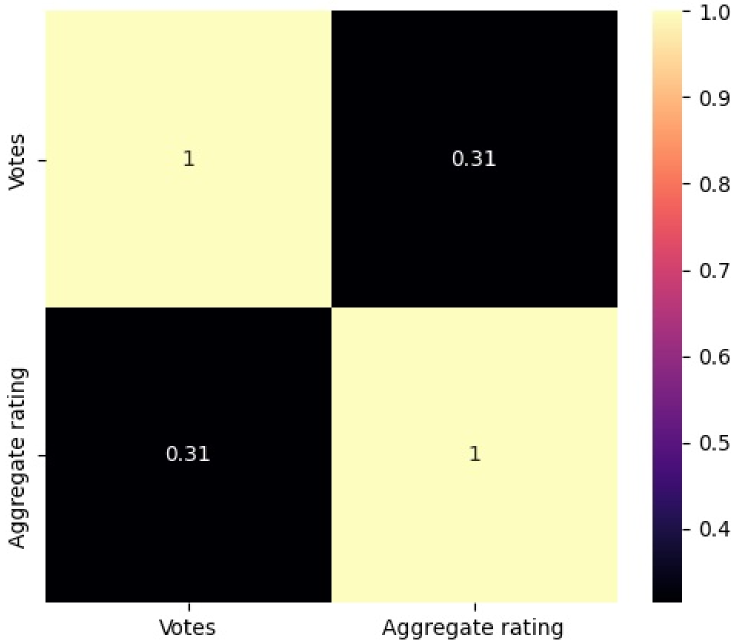
In [241…

```
#calculate the correlation between number of vote and rating
correlation = df[['Votes','Aggregate rating']].corr()
print("Correlation between Number of votes and Rating:", correlation)
```

```
Correlation between Number of votes and Rating:                        Votes  Aggregate rating
Votes             1.000000          0.313474
Aggregate rating  0.313474          1.000000
```

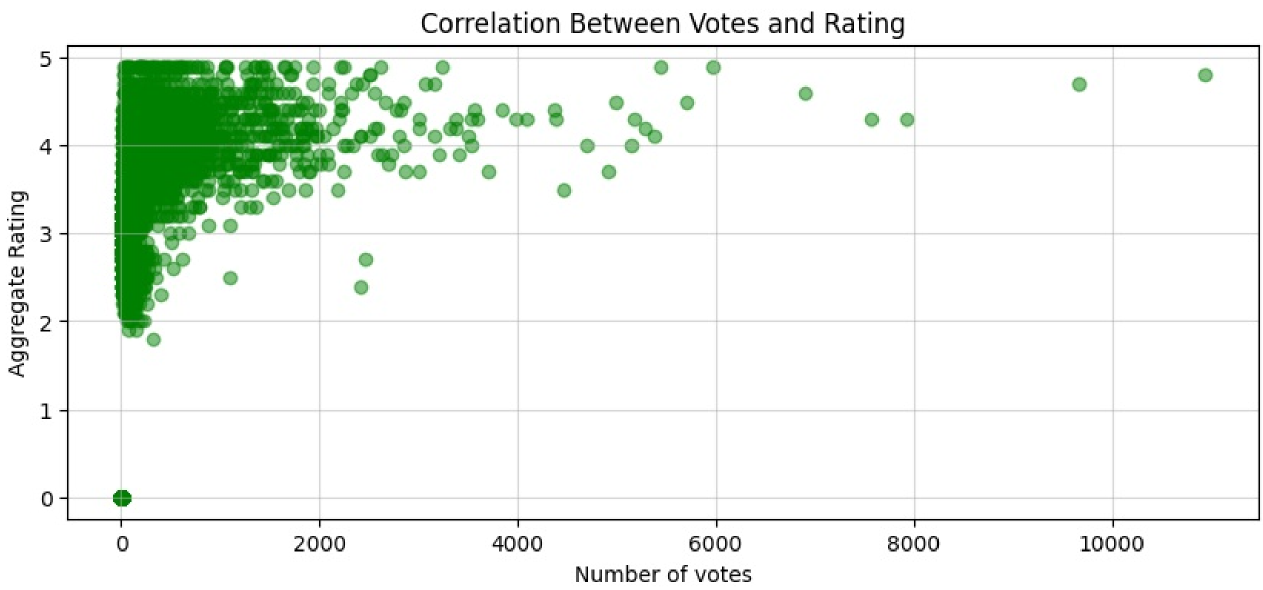
In [242…

```
plt.figure(figsize=(6,5))
sns.heatmap(correlation, cmap='magma', annot=True)
plt.show()
```



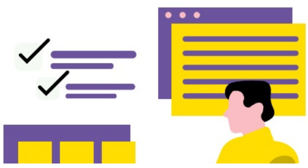
In [243…

```
# top_10.plot(kind='bar')
plt.figure(figsize=(10,4))
plt.scatter(df['Votes'],df['Aggregate rating'],alpha=0.5,color='green')
plt.xlabel("Number of votes")
plt.ylabel("Aggregate Rating")
plt.title("Correlation Between Votes and Rating")
plt.grid(axis='both',linestyle='-',alpha=0.5)
```



Level 3

Task 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 [244…

```
df.columns
```

Out[244]:
```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       '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', 'Review length'],
      dtype='object')
```

Task

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

In [245…
```python
# Convert relevant columns to categorical types for easier analysis
df['Price range'] = df['Price range'].astype('category')
df['Has Table booking'] = df['Has Table booking'].astype('category')
df['Has Online delivery'] = df['Has Online delivery'].astype('category')
# Cross-tabulation of Price range with Table booking and Online delivery
table_booking_vs_price = pd.crosstab(df['Price range'], df['Has Table booking'], normalize='index') * 100
online_delivery_vs_price = pd.crosstab(df['Price range'], df['Has Online delivery'], normalize='index') * 100
print(table_booking_vs_price)
print(online_delivery_vs_price)
```

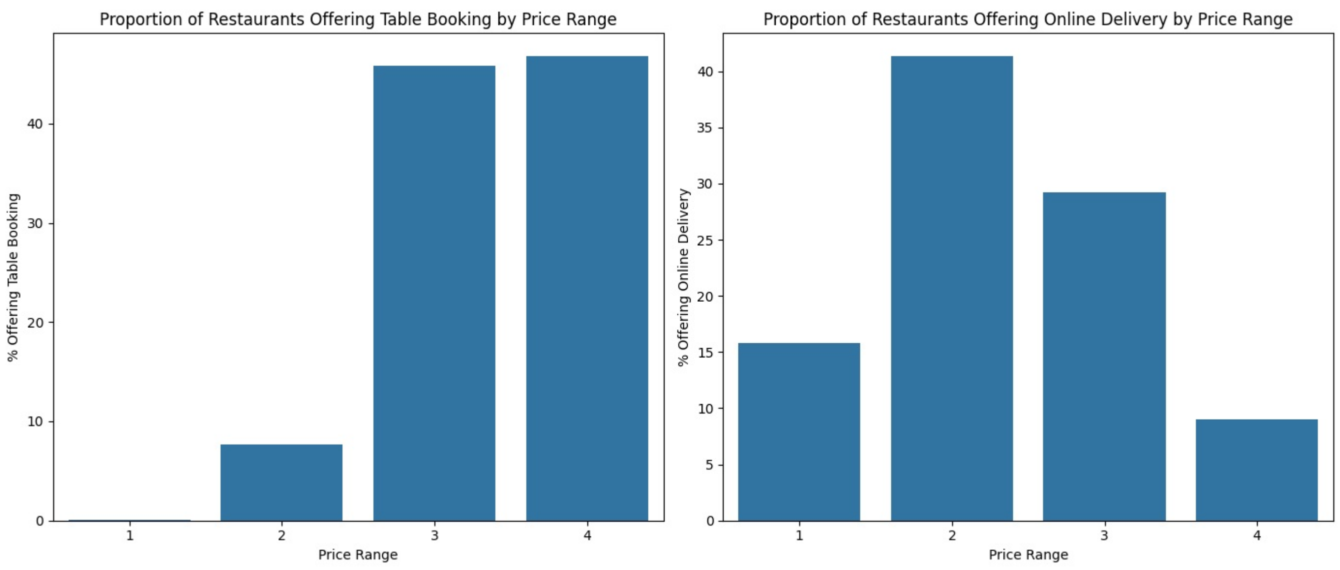
```
Has Table booking         No        Yes
Price range
1                  99.977467   0.022533
2                  92.322518   7.677482
3                  54.163701  45.836299
4                  53.242321  46.757679
Has Online delivery         No        Yes
Price range
1                    84.204597  15.795403
2                    58.689367  41.310633
3                    70.747331  29.252669
4                    90.955631   9.044369
```

Task

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

In [246…
```python
# Calculate the correlation between Price range and service availability
table_booking_correlation = df['Price range'].cat.codes.corr(df['Has Table booking'].cat.codes)
online_delivery_correlation = df['Price range'].cat.codes.corr(df['Has Online delivery'].cat.codes)
print(table_booking_correlation,online_delivery_correlation)
# Plotting the trends
fig, axes = plt.subplots(1, 2, figsize=(14, 6))
# Table booking vs Price range
sns.barplot(x=table_booking_vs_price.index, y=table_booking_vs_price['Yes'], ax=axes[0])
axes[0].set_title('Proportion of Restaurants Offering Table Booking by Price Range')
axes[0].set_ylabel('% Offering Table Booking')
axes[0].set_xlabel('Price Range')
# Online delivery vs Price range
sns.barplot(x=online_delivery_vs_price.index, y=online_delivery_vs_price['Yes'], ax=axes[1])
axes[1].set_title('Proportion of Restaurants Offering Online Delivery by Price Range')
axes[1].set_ylabel('% Offering Online Delivery')
axes[1].set_xlabel('Price Range')
plt.tight_layout()
plt.show()
```

```
0.5021659712421332 0.07788653076795976
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/extensions/Safe.js