COGNIFYZ

IMPORTING LIBRARIES

In [3]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

READING THE DATA

In [9]:
```
df=pd.read_csv('interndata.csv')
df.head(5)
```

Out[9]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Cer P Ma
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City	Lit Ma
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Sh Man C
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	M Man
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	M Man

5 rows × 21 columns

Data Cleaning

In [14]:
```
df.isnull()
```

Out[14]:

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

9551 rows × 21 columns

In [16]:
```
df.isnull().sum()
```

Out[16]:
```
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
```

here we do not have any NaN value

In [19]:
```
df.describe()
```

Out[19]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	P
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	

STARTING OF LEVEL 1

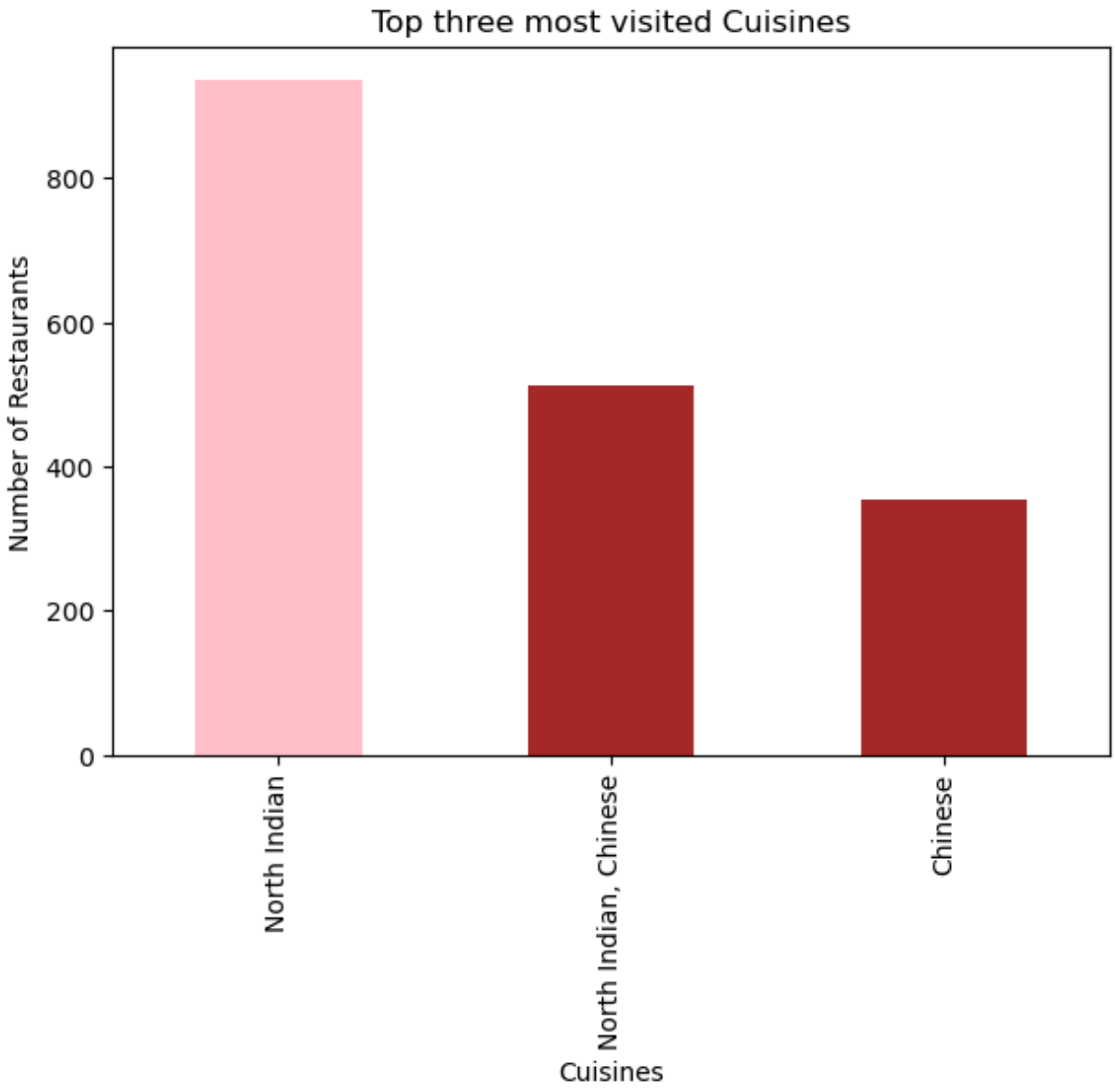
1.Determine the top three most common cuisines in Dataset.

In [36]:
```
a =df['Cuisines'].value_counts()
top_three=a.head(3)
count_cuisines=a
```

In [38]:
```
print("The top three Cuisines:")
for i in range(len(top_three.index)):
    print(top_three.index[i])
```

```
The top three Cuisines:
North Indian
North Indian, Chinese
Chinese
```

In [113…
```
plt.figure(figsize=(7,5))
color=['pink','brown','brown']
top_three.plot(kind='bar',color=color)
plt.title('Top three most visited Cuisines')
plt.xlabel('Cuisines')
plt.ylabel('Number of Restaurants')
plt.show()
```



Percentage of top three cusines

In [60]:
```
tot_res=len(df)
tot_res
per= (top_three.values/tot_res)*100
data=dict(zip(top_three.index,per))
data
data
```

Out[60]:
```
{'North Indian': 9.800020940215685,
 'North Indian, Chinese': 5.350225107318606,
 'Chinese': 3.706418176107214}
```

In [62]:
```
a=pd.DataFrame(data.items(),columns = ['cuisine','percentage'])
a
```

Out[62]:

	cuisine	percentage
0	North Indian	9.800021
1	North Indian, Chinese	5.350225
2	Chinese	3.706418

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

In [72]:
```
df['City'].value_counts()
city=df['City'].value_counts()
city.index[0]
```

Out[72]: 'New Delhi'

average rating

In [81]:
```
mean=df['Aggregate rating'].mean()
average_ratings = df.groupby(['City','Restaurant Name'])['Aggregate rating']
average_ratings
```

Out[81]:

	City	Restaurant Name	Aggregate rating
0	Abu Dhabi	Applebee's	4.0
1	Abu Dhabi	Bait El Khetyar	4.0
2	Abu Dhabi	Cho Gao - Crowne Plaza Abu Dhabi	4.4
3	Abu Dhabi	Denny's	4.6
4	Abu Dhabi	Famous Dave's Barbecue	4.6
...	...	...	...
7969	��stanbul	Leman K�_lt�_r	3.7
7970	��stanbul	Namlı Gurme	4.1
7971	��stanbul	Starbucks	4.9
7972	��stanbul	Valonia	4.2
7973	��stanbul	Walter's Coffee Roastery	4.0

7974 rows × 3 columns

In [85]:
```
average_ratings = df.groupby('City')['Aggregate rating'].mean().reset_index(
average_ratings
```

Out[85]:

	City	Aggregate rating
0	Abu Dhabi	4.300000
1	Agra	3.965000
2	Ahmedabad	4.161905
3	Albany	3.555000
4	Allahabad	3.395000
...	...	...
136	Weirton	3.900000
137	Wellington City	4.250000
138	Winchester Bay	3.200000
139	Yorkton	3.300000
140	��stanbul	4.292857

141 rows × 2 columns

highest average rating

In [90]:
```
average_ratings= df.groupby('City')['Aggregate rating'].mean().reset_index()
average_ratings_city=average_ratings.sort_values(by='Aggregate rating',ascer
average_ratings_city
```

Out[90]:

	City	Aggregate rating
56	Inner City	4.900000
107	Quezon City	4.800000
73	Makati City	4.650000
95	Pasig City	4.633333
75	Mandaluyong City	4.625000
...	...	...
88	New Delhi	2.438845
83	Montville	2.400000
78	Mc Millan	2.400000
89	Noida	2.036204
43	Faridabad	1.866932

141 rows × 2 columns

In [92]:
```
average_ratings_city.iloc[0,0]
```

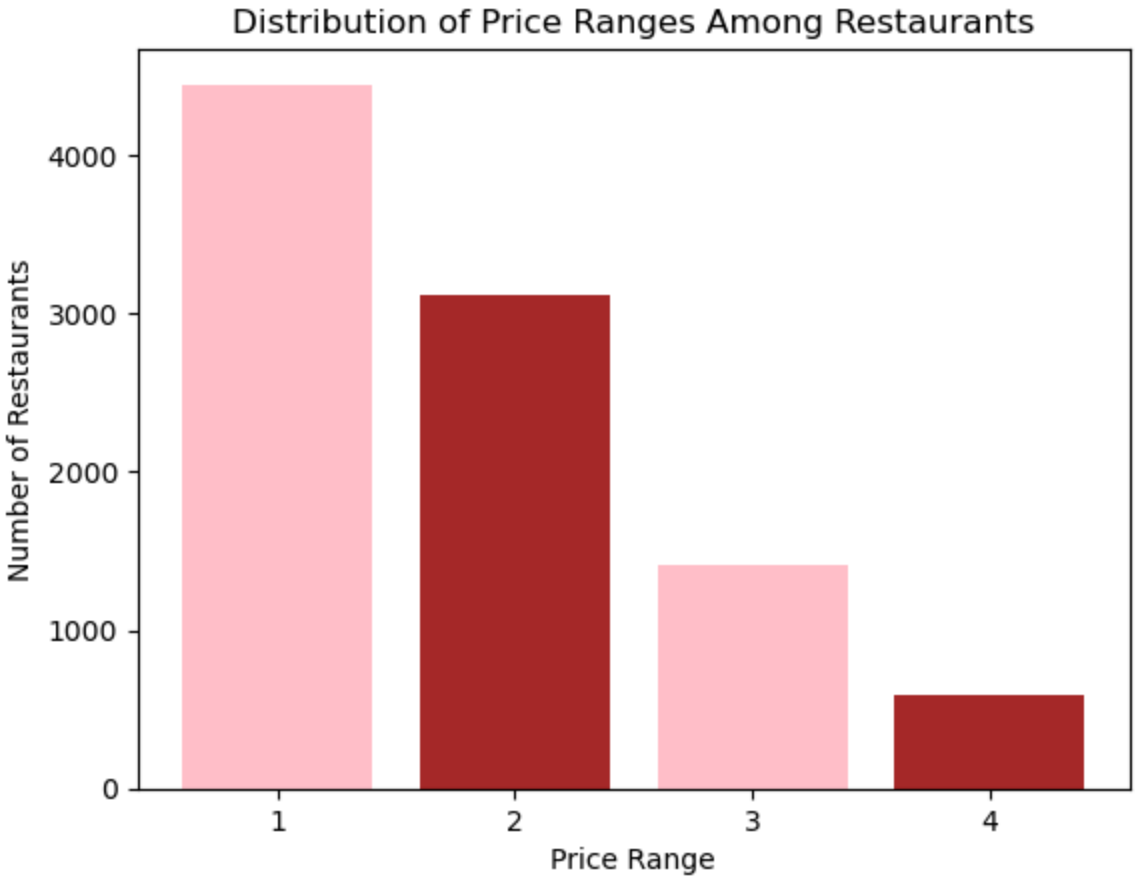
Out[92]: 'Inner City'

3. Creating a histogram or bar chart to visualize the distribution of price ranges among the restaurants.

In [97]:
```
df['Price range'].value_counts()
```

Out[97]:
```
Price range
1    4444
2    3113
3    1408
4     586
Name: count, dtype: int64
```

In [109…
```
price_counts = df['Price range'].value_counts()
plt.bar(['1','2','3','4'], list(price_counts.values), color=['pink', 'brown'
plt.xlabel('Price Range')
plt.ylabel('Number of Restaurants')
plt.title('Distribution of Price Ranges Among Restaurants')
plt.show()
```



calculating the percentage of restraunt

In [116…
```
total=len(df)
total
```

Out[116…
```
9551
```

In [120…
```
count = df['Price range'].value_counts().values
count
```

Out[120…
```
array([4444, 3113, 1408,  586], dtype=int64)
```

In [128…
```
percentage=round(df['Price range'].value_counts()/total*100,2)
percentage
```

Out[128…
```
Price range
1    46.53
2    32.59
3    14.74
4     6.14
Name: count, dtype: float64
```

In [132…
```
df1=pd.DataFrame({'Price range':df['Price range'].value_counts().index,
                  'Count':count,
                  'Percentage':percentage})
df1
```

Out[132…

Price range	Price range	Count	Percentage
1	1	4444	46.53
2	2	3113	32.59
3	3	1408	14.74
4	4	586	6.14

4.the percentage of restaurants that offer online delivery.

In [139…
```
df[df['Has Online delivery']=='Yes']['Restaurant Name'].value_counts()
```

Out[139…
```
Restaurant Name
Subway                  47
McDonald's              30
Pizza Hut Delivery      18
Chaayos                 15
KFC                     15
                        ..
Al Zaitoon               1
The Barley House         1
Bemisaal                 1
Urban Kabab              1
Saffron Mantra           1
Name: count, Length: 1728, dtype: int64
```

In [141…
```
leng=len(df[df['Has Online delivery']=='Yes'])
leng
```

Out[141…
```
2451
```

In [149…
```
round(df[df['Has Online delivery']=='Yes']['Restaurant Name'].value_counts()
```

Out[149…
```
Restaurant Name
Subway                  1.92
McDonald's              1.22
Pizza Hut Delivery      0.73
Chaayos                 0.61
KFC                     0.61
                        ...
Al Zaitoon              0.04
The Barley House        0.04
Bemisaal                0.04
Urban Kabab             0.04
Saffron Mantra          0.04
Name: count, Length: 1728, dtype: float64
```

In [151…
```
total=len(df)
total
```

Out[151…
```
9551
```

In [153…
```
percentage=(leng/total)*100
percentage
```

Out[153…
```
25.662234321013504
```

Comparing the average ratings of restaurants with and without online delivery.

In [158…
```
online_data=df[df['Has Online delivery']=='Yes']
offline_data=df[df['Has Online delivery']=='No']
```

In [160…
```
average_ratings= online_data.groupby('Restaurant Name')['Aggregate rating'].
```

```
average_ratings
```

Out[160…

	Restaurant Name	Aggregate rating
0	#OFF Campus	3.70
1	#Urban Caf��	3.30
2	18 Degrees Resto Lounge	3.60
3	19 Flavours Biryani	4.10
4	2 Bros Kitchen	3.20
...	...	...
1723	Zucca Pizzeria	3.50
1724	bu��no	3.75
1725	hug!	3.50
1726	iGNiTE	3.00
1727	iKitchen	0.00

1728 rows × 2 columns

average rating of restaurant with and without online delivery

In [165…

```
df.groupby('Has Online delivery')['Aggregate rating'].mean().round(2).reset_
```

Out[165…

	Has Online delivery	Aggregate rating
0	No	2.47
1	Yes	3.25

In [ ]:

```

```