Zomato Data analysis Report

Import the libraries

In [98]:
```
import pandas as pd                                          #for manipulating dataframe
import numpy as np                                           #for performing mathematical calculations
import matplotlib.pyplot as plt                              #for data visualizations i.e. graphs and charts
import seaborn as sns                                        #for advance data visualizations
import plotly.express as px
import plotly.graph_objects as go

pd.pandas.set_option('display.max_columns', None)
```

Read the Dataset

In [206…
```
#dataset1
df = pd.read_csv('zomato.csv',encoding = 'latin-1')
```

In [207…
```
df
```

Out[207…

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	A C
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	NamlÛ± Gurme	208	ÛÁstanbul	Kemankeô Karamustafa Paôa Mahallesi, RÛ±htÛ±...	Karakí_y	Karakí_y, ÛÁstanbul	28.977392	41.022793	Turkish	
9547	5908749	Ceviz AÛôacÛ±	208	ÛÁstanbul	Koôuyolu Mahallesi, Muhittin íistí_ndaÛô Cadd...	Koôuyolu	Koôuyolu, ÛÁstanbul	29.041297	41.009847	World Cuisine, Patisserie, Cafe	
9548	5915807	Huqqa	208	ÛÁstanbul	Kuruí_eôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_eôme	Kuruí_eôme, ÛÁstanbul	29.034640	41.055817	Italian, World Cuisine	
9549	5916112	Aôôk Kahve	208	ÛÁstanbul	Kuruí_eôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_eôme	Kuruí_eôme, ÛÁstanbul	29.036019	41.057979	Restaurant Cafe	
9550	5927402	Walter's Coffee Roastery	208	ÛÁstanbul	CafeaÛôa Mahallesi, BademaltÛ± Sokak, No 21/B,...	Moda	Moda, ÛÁstanbul	29.026016	40.984776	Cafe	

9551 rows × 21 columns

In [32]:
```
df.shape
```

Out[32]: (9551, 21)

In [33]:
```
#dataset2
df1 = pd.read_excel('Country-Code (1).xlsx')
```

In [34]:
```
df1
```

Out[34]:

	Country Code	Country
0	1	India
1	14	Australia
2	30	Brazil
3	37	Canada
4	94	Indonesia
5	148	New Zealand
6	162	Phillipines
7	166	Qatar
8	184	Singapore
9	189	South Africa
10	191	Sri Lanka
11	208	Turkey
12	214	UAE
13	215	United Kingdom
14	216	United States

Merge the both tables

In [35]:
```
data = pd.merge(df,df1, on = 'Country Code', how = 'left')
```

In [36]:
```
data
```

Out[36]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	A C
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	NamlÛ± Gurme	208	ÛÁstanbul	Kemankeô Karamustafa Paôa Mahallesi, RÛ±htÛ±...	Karakí_y	Karakí_y, ÛÁstanbul	28.977392	41.022793	Turkish	
9547	5908749	Ceviz AÛôacÛ±	208	ÛÁstanbul	Koôuyolu Mahallesi, Muhittin íistí_ndaÛô Cadd...	Koôuyolu	Koôuyolu, ÛÁstanbul	29.041297	41.009847	World Cuisine, Patisserie, Cafe	
9548	5915807	Huqqa	208	ÛÁstanbul	Kuruí_eôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_eôme	Kuruí_eôme, ÛÁstanbul	29.034640	41.055817	Italian, World Cuisine	
9549	5916112	Aôôk Kahve	208	ÛÁstanbul	Kuruí_eôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_eôme	Kuruí_eôme, ÛÁstanbul	29.036019	41.057979	Restaurant Cafe	
9550	5927402	Walter's Coffee Roastery	208	ÛÁstanbul	CafeaÛôa Mahallesi, BademaltÛ± Sokak, No 21/B,...	Moda	Moda, ÛÁstanbul	29.026016	40.984776	Cafe	

9551 rows × 22 columns

Data Cleaning

Now, do some Descriptive Analysis

In [37]:
```
#check for missing values
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 22 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  
 21  Country               9551 non-null   object 
dtypes: float64(3), int64(5), object(14)
memory usage: 1.6+ MB
```

In [39]:
```
#check for null values
data.isnull().sum()
```

Out[39]:
```
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
Country                 0
dtype: int64
```

In [44]:
```
# remove the null values in 'cuisines' column
data.dropna(subset = ['Cuisines'], inplace = True)
```

In [47]:
```
#again check if there is any null value
data.isnull().sum()
```

Out[47]:
```
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
Country               0
dtype: int64
```

In [48]:
```
data.describe()
```

Out[48]:

	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 [49]:
```
data.shape
```

Out[49]:
```
(9542, 22)
```

In [52]:
```
#count the countries
counts = data.Country.value_counts()
```

In [53]:
```
counts
```

Out[53]:
```
Country
India             8652
United States      425
United Kingdom      80
Brazil              60
South Africa        60
UAE                 60
New Zealand         40
Turkey              34
Australia           24
Phillipines         22
Indonesia           21
Qatar               20
Singapore           20
Sri Lanka           20
Canada               4
Name: count, dtype: int64
```

In [59]:
```
country_name = counts.index
country_name
```

Out[59]:
```
Index(['India', 'United States', 'United Kingdom', 'Brazil', 'South Africa',
       'UAE', 'New Zealand', 'Turkey', 'Australia', 'Phillipines', 'Indonesia',
       'Qatar', 'Singapore', 'Sri Lanka', 'Canada'],
      dtype='object', name='Country')
```

Data Analysis

que1 : Find the top 5 countries according to counts

In [ ]:
```
#This graph is done with plotly

c = ['pink','yellow','grey','green','red']
ex = [0.0,0.0,0.0,0.4,0.0]
plt.pie(counts[:5],labels = country_name[:5], autopct = '%1.1f%%',colors = c,explode = ex)
plt.legend()
plt.show()
```

In [107…
```
#this chart is more clear

fig = go.Figure(data = [go.Pie(labels = country_name[:5],values = counts[:5])])
fig.update_layout(title_text = "Top 5 Countries which has maximum counts")
```

que2: What are the maximum ratings given by citizens

In [109…
```
data.columns
```

Out[109…
```
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', 'Country'],
      dtype='object')
```

In [116…
```
data.groupby(['Aggregate rating','Rating color','Rating text']).size().reset_index()
```

Out[116…

	Aggregate rating	Rating color	Rating text	0
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 [119…

```
## rename the column "0"
ratings = data.groupby(['Aggregate rating','Rating color','Rating text']).size().reset_index().rename(columns={(
ratings
```

Out[119…

	Aggregate rating	Rating color	Rating text	Counting
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 [ ]:
```
''' So, here we seen in our Analysis that
    1. 0.0 rating is in white color which is not rated
    2. 1.8 - 2.4 has given red color which is Poor
    3. 2.5 - 3.4 has given orange color which is average
    4. 3.5 - 3.9 has given yellow color which is Good
    5. 4.0 - 4.4 has given green color which is very Good
    6. 4.5 - 4.9 has given dark green which is excellent '''
```

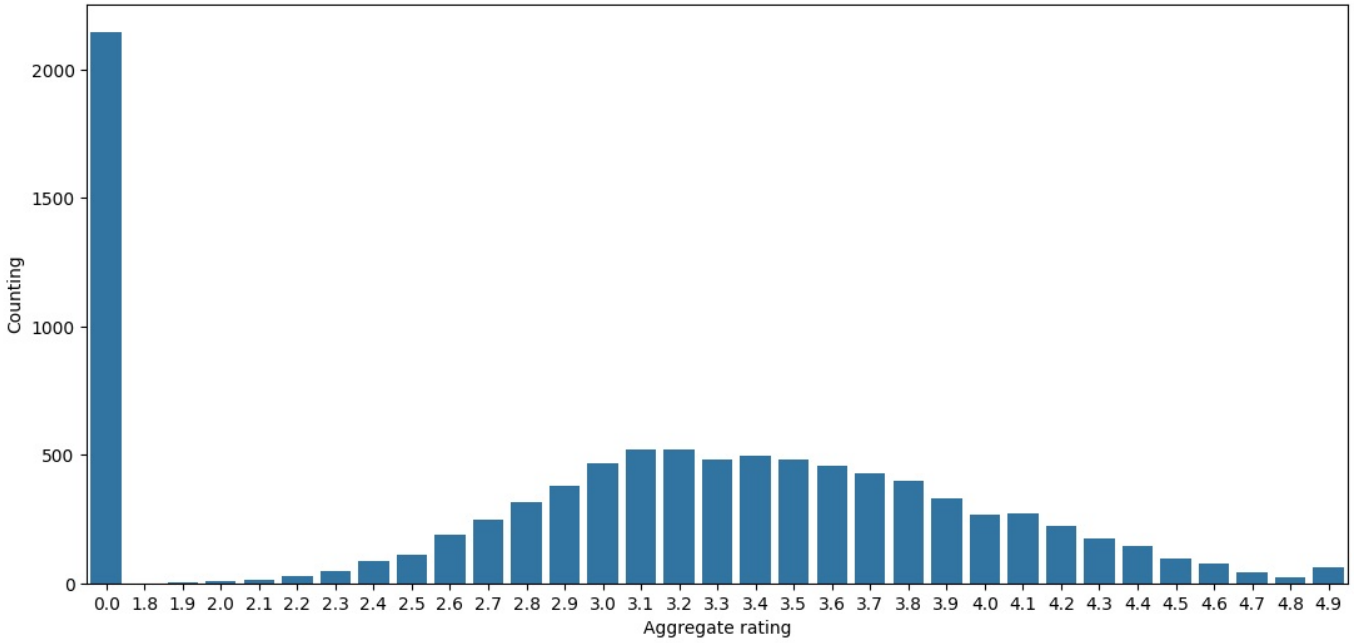
In [120…
```
ratings.describe()
```

Out[120…

	Aggregate rating	Counting
count	33.000000	33.000000
mean	3.248485	289.151515
std	1.092051	379.913980
min	0.000000	1.000000
25%	2.500000	61.000000
50%	3.300000	221.000000
75%	4.100000	427.000000
max	4.900000	2148.000000

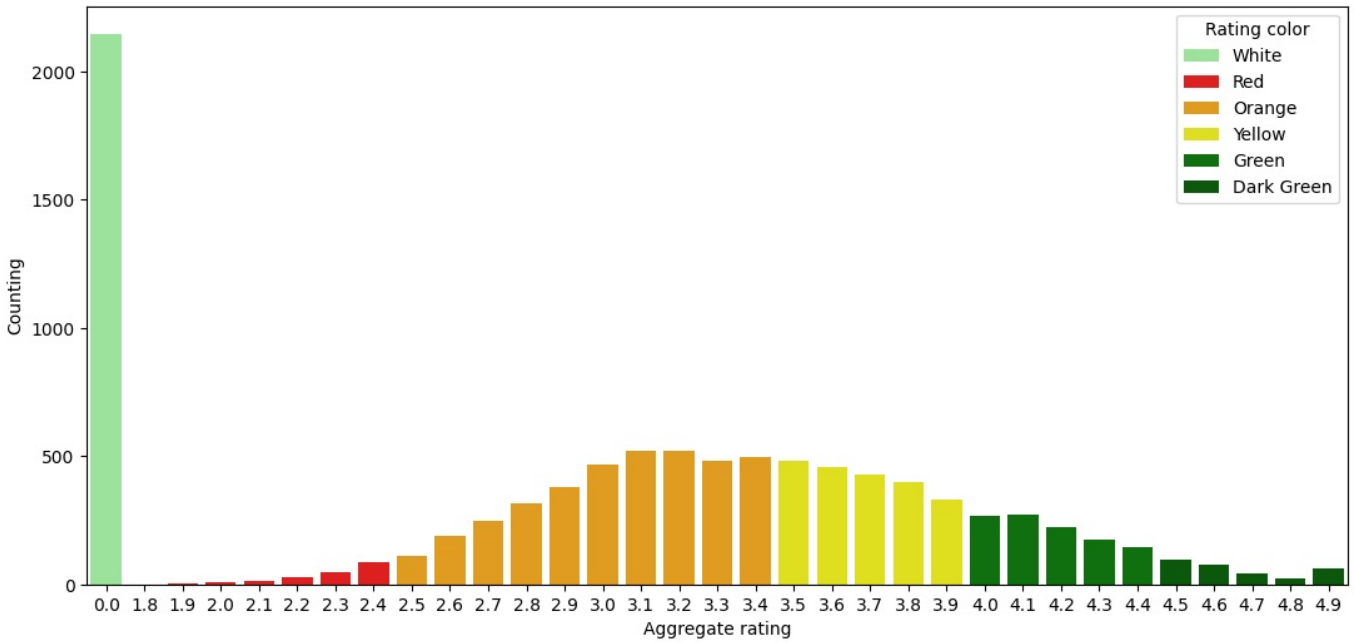
In [131…
```
## draw the chart
plt.figure(figsize=(13,6))
sns.barplot(x = 'Aggregate rating',y = 'Counting',data = ratings)
```

Out[131…
```
<Axes: xlabel='Aggregate rating', ylabel='Counting'>
```



In [134…
```
c = ['lightgreen','red','orange','yellow','green','darkgreen']
plt.figure(figsize=(13,6))
sns.barplot(x = 'Aggregate rating',y = 'Counting',data = ratings,hue = 'Rating color',palette = c)
```

Out[134…
```
<Axes: xlabel='Aggregate rating', ylabel='Counting'>
```



que3: Countries that has given 0 ratings

In [140…
```
data[data['Rating color'] == 'White'].groupby('Country').size().reset_index()
```

Out[140…

	Country	0
0	Brazil	5
1	India	2139
2	United Kingdom	1
3	United States	3

In [141…
```
data[data['Rating color'] == 'White'].groupby('Country').size().reset_index().rename(columns = {0:'No ratings'}
```

Out[141…

	Country	No ratings
0	Brazil	5
1	India	2139
2	United Kingdom	1
3	United States	3

In [ ]:
```
''' Only 4 countries are there who gives '0' ratings
    Indiaa,Brazil,United States,United Kingdom'''
```

que4: Find the Currency which is maximum used

In [157…
```
data.groupby(['Currency','Country']).size().reset_index().rename(columns = {0:'No. of Currencies'}).sort_values
```

Out[157…

	Currency	Country	No. of Currencies
7	Indian Rupees(Rs.)	India	8652
5	Dollar($)	United States	425
10	Pounds(£)	United Kingdom	80
6	Emirati Diram(AED)	UAE	60
12	Rand(R)	South Africa	60
1	Brazilian Real(R$)	Brazil	60
9	NewZealand($)	New Zealand	40
14	Turkish Lira(TL)	Turkey	34
2	Dollar($)	Australia	24
0	Botswana Pula(P)	Phillipines	22
8	Indonesian Rupiah(IDR)	Indonesia	21
11	Qatari Rial(QR)	Qatar	20
4	Dollar($)	Singapore	20
13	Sri Lankan Rupee(LKR)	Sri Lanka	20
3	Dollar($)	Canada	4

In [ ]:
```
''' From this observation we have seen that the maximum currencies are from INDIA i.e 8652 '''
```

Que5: Find the countries who have online options

In [160…
```
data[data['Has Online delivery'] == 'Yes'].groupby('Country').size().reset_index()
```

Out[160…

	Country	0
0	India	2423
1	UAE	28

In [ ]:
```
''' From this observation only Two countries have online options INDIA and UAE'''
```

Que6: Find the top 5 cities distribution

In [170…
```
max_cities = data.City.value_counts().reset_index()
max_cities
```

Out[170…

	City	count
0	New Delhi	5473
1	Gurgaon	1118
2	Noida	1080
3	Faridabad	251
4	Ghaziabad	25
...	...	...
135	Inverloch	1
136	Mohali	1
137	Panchkula	1
138	Bandung	1
139	Randburg	1

140 rows × 2 columns

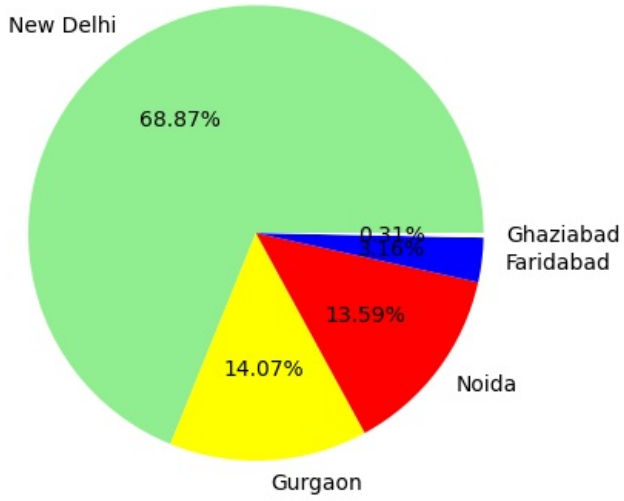
In [179…
```
city_values = data.City.value_counts().values
city_values
```

Out[179…
```
array([5473, 1118, 1080,  251,   25,   21,   21,   21,   21,   21,   20,
         20,   20,   20,   20,   20,   20,   20,   20,   20,   20,   20,
         20,   20,   20,   20,   20,   20,   20,   20,   20,   20,   20,
         20,   20,   20,   20,   20,   20,   20,   20,   20,   20,   20,
         20,   20,   20,   20,   20,   20,   20,   20,   20,   20,   20,
         20,   20,   20,   20,   20,   19,   19,   19,   19,   19,   18,
         18,   17,   16,   14,   11,    6,    4,    4,    3,    3,    2,
          2,    2,    2,    2,    2,    2,    2,    1,    1,    1,    1,
          1,    1,    1,    1,    1,    1,    1,    1,    1,    1,    1,
          1,    1,    1,    1,    1,    1,    1,    1,    1,    1,    1,
          1,    1,    1,    1,    1,    1,    1,    1])
```

In [181…
```
city_labels = data.City.value_counts().index
city_labels
```

Out[181…
```
Index(['New Delhi', 'Gurgaon', 'Noida', 'Faridabad', 'Ghaziabad',
       'Bhubaneshwar', 'Ahmedabad', 'Lucknow', 'Guwahati', 'Amritsar',
       ...
       'Forrest', 'Dicky Beach', 'Flaxton', 'Huskisson', 'Lakes Entrance',
       'Inverloch', 'Mohali', 'Panchkula', 'Bandung', 'Randburg'],
      dtype='object', name='City', length=140)
```

In [192…
```
c = ['lightgreen','yellow','red','blue','white']
plt.pie(city_values[:5],labels = city_labels[:5],autopct = '%1.2f%%',colors = c)
plt.show()
```



Que7: Find the top 10 cuisines served by the restaurants

In [199…
```
data['Cuisines'].value_counts().reset_index()
```

Out[199…

	Cuisines	count
0	North Indian	936
1	North Indian, Chinese	511
2	Chinese	354
3	Fast Food	354
4	North Indian, Mughlai	334
...	...	...
1820	World Cuisine, Patisserie, Cafe	1
1821	Burger, Izgara	1
1822	Desserts, Bí_rek	1
1823	Restaurant Cafe, Turkish, Desserts	1
1824	Restaurant Cafe, Desserts	1

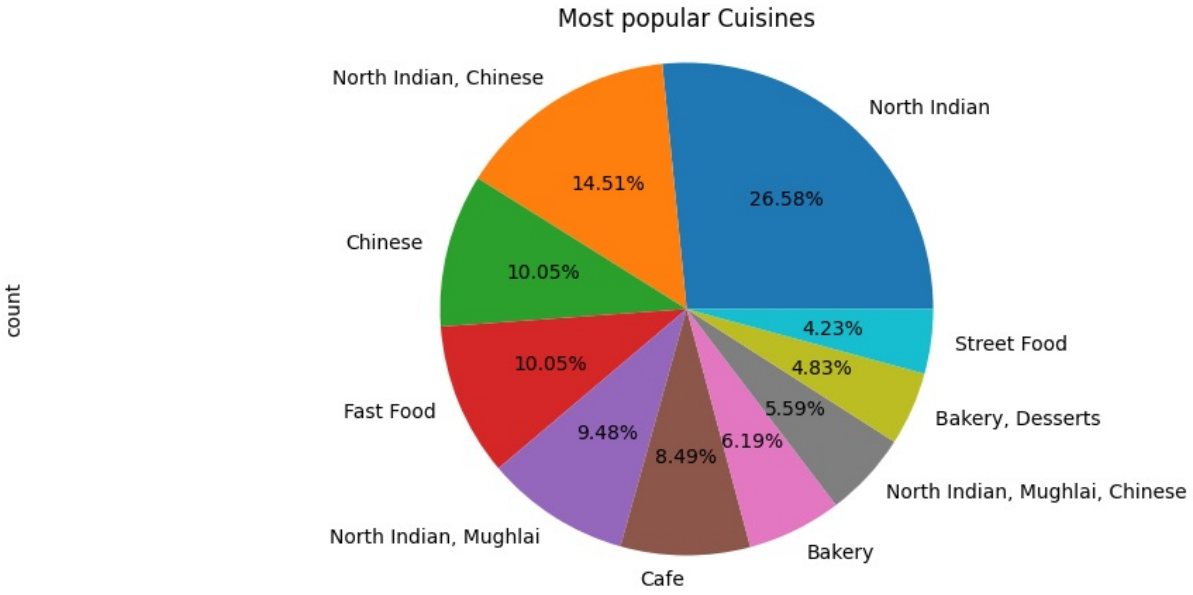
1825 rows × 2 columns

In [213…
```
data['Cuisines'].value_counts().sort_values(ascending = False).head(10)
```

Out[213…
```
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
```

In [232…
```
data['Cuisines'].value_counts().sort_values(ascending = False).head(10).plot(kind = 'pie',figsize = (12,5),autop
plt.axis('equal')
```

Out[232…
```
(np.float64(-1.099999947692328),
 np.float64(1.0999999975091586),
 np.float64(-1.1000000038455147),
 np.float64(1.0999997430473287))
```



In [ ]:
```
''' so, this is our final analysis'''
```