Zomato

Exploratory Data Analysis

(EDA)

Muhammad Tabish Sami
www.bigdaata.com

To import libraries

- pandas
- numpy
- matplotlib
- seaborn

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

To read csv

```
1 df = pd.read_csv('zomato.csv',encoding='ISO-8859-1')
2 df.head()
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking	Has Online delivery	deliv
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	...	Botswana Pula(P)	Yes	No	
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	...	Botswana Pula(P)	Yes	No	
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	...	Botswana Pula(P)	Yes	No	
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	...	Botswana Pula(P)	No	No	
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	...	Botswana Pula(P)	Yes	No	

5 rows × 21 columns

To check columns

```
1 df.columns
```

```
Out[3]: 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')
```

To determine non-null values and data type

```
1 df.info()
```

```
 #   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
```

To find statistical information about numerical columns.

```
1 df.describe()
```

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	Votes
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.666370	156.909748
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.516378	430.169145
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	0.000000
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.500000	5.000000
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.200000	31.000000
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.700000	131.000000
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	10934.000000

To check the size

```
1 df.size
```

```
Out[6]: 200571
```

To check the shape

```
1 df.shape
```

```
(9551, 21)
```

To check the missing values

```
1  df.isnull().sum()
```

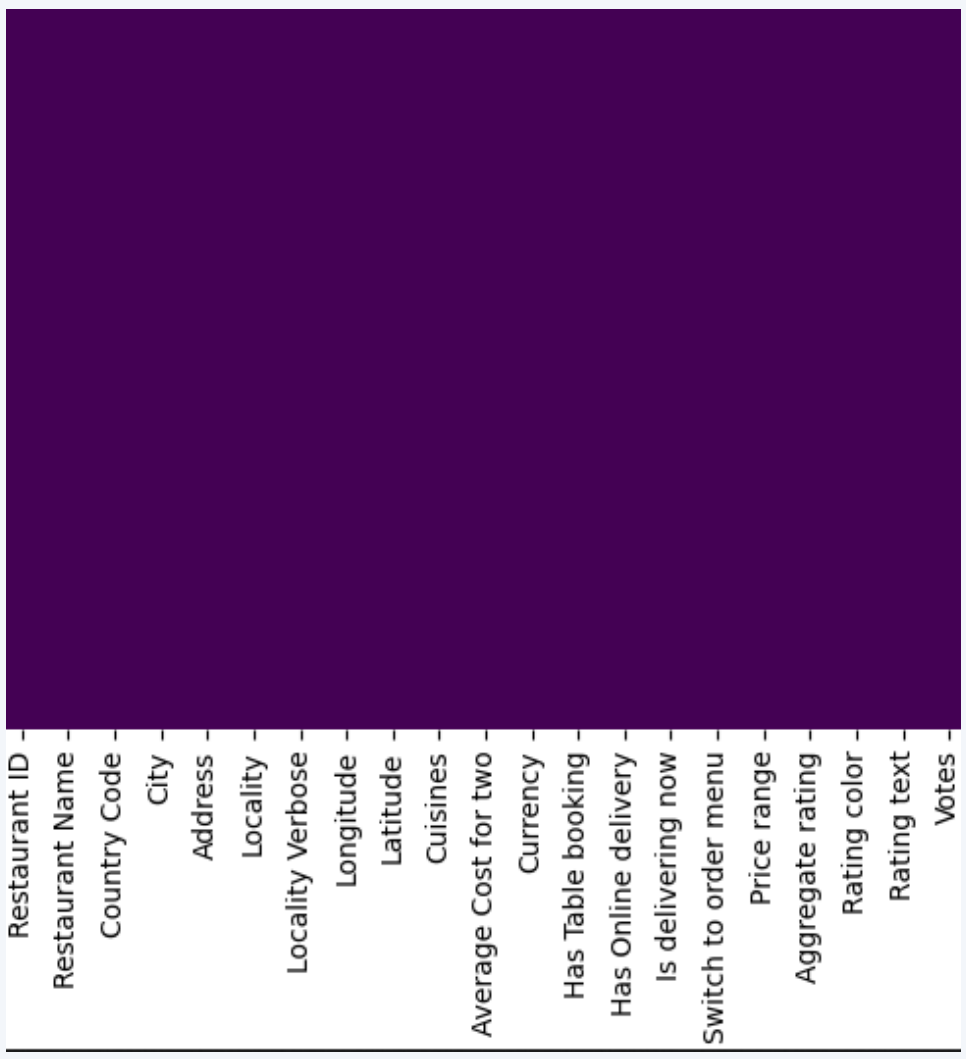
```
Restaurant ID         0
Restaurant Name       0
Country Code          0
City                  0
Address               0
Locality              0
Locality Verbose      0
Longitude             0
Latitude              0
Cuisines              9
Average Cost for two  0
Currency              0
Has Table booking     0
Has Online delivery   0
Is delivering now     0
Switch to order menu  0
Price range           0
Aggregate rating      0
Rating color          0
```

Observation:

Cuisines has missing values: **9.**

To check the missing values profile using heat map

```
1  # With the help of heat map
2  sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```



We have two excel files (i.e., ***zomato.csv & Country-Code.xlsx***).
We have to merge these both files on column = '***Country-Code***'.

To read excel file for 'Country-Code.xlsx'

```
df_country = pd.read_excel('Country-Code.xlsx')
df_country.head()
```

	Country Code	Country
0	1	India
1	14	Australia
2	30	Brazil
3	37	Canada
4	94	Indonesia

To combine both datasets (Column = 'Country Code', Join = 'LEFT')

```
final_df=pd.merge(df,df_country,on='Country Code',how='left')
```

To display top 2 records

```
1  final_df.head(2)
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range
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	...	Yes	No	No	No	3
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	...	Yes	No	No	No	3

2 rows × 22 columns

Question 1.

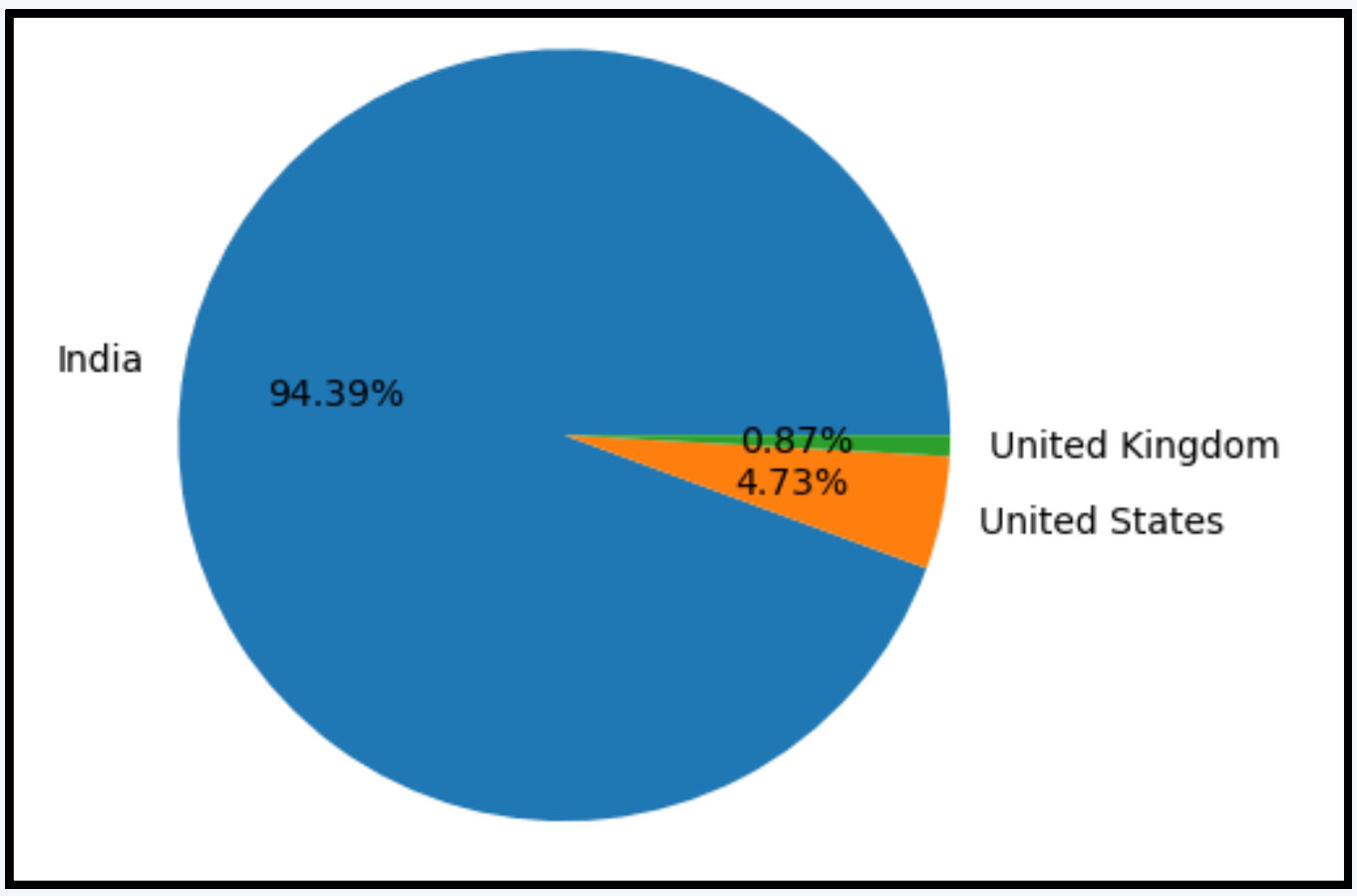
To find out how many particular countries are there?

```
1  # To make labels for 'Country names'
2  country_name = final_df.Country.value_counts().index
```

```
1  # To create an array for values of total country
2  country_values=final_df.Country.value_counts().values
```

```
1  # To create a pie chart showing Top 3 countries
2  plt.pie(country_values[:3],labels=country_name[:3],autopct='%1.2f%%')
```

Pie Chart



Observation:

Zomato has maximum transaction from **India** at **94.39%** followed by **United States** at **4.73%**, and **United Kingdom** having least transactions at **0.87%**.

Question 2 :

To find the ratings for Zomato order.

```
ratings = final_df.groupby(['Aggregate rating','Rating color','Rating text']).size().reset_index().rename(columns={0:'Rating
# reset_index() function will reset the orignal index.
# rename() function will rename the column 0 to 'Rating Count'
```

Rating

Observation:

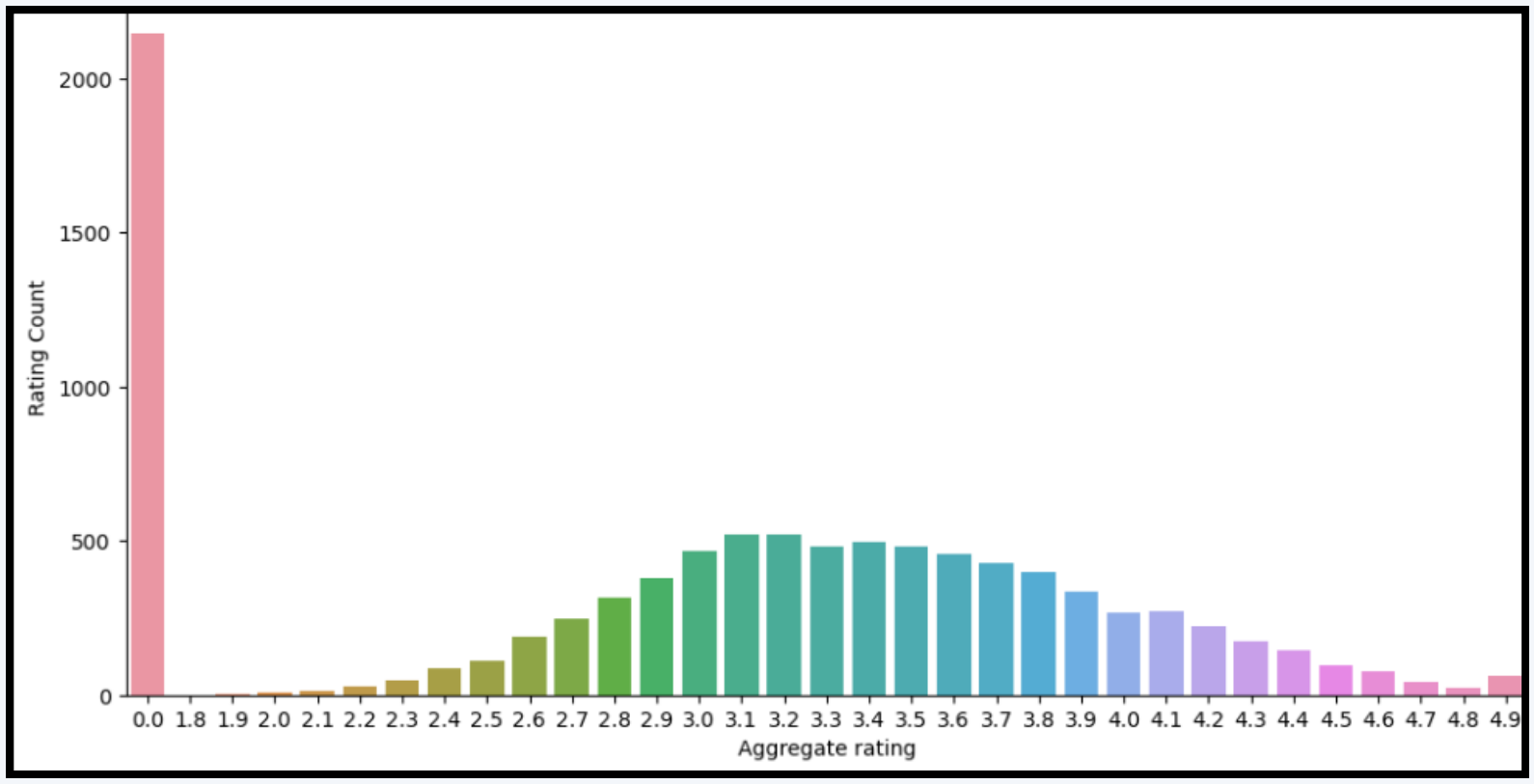
1. When rating is from **4.5-4.9** ---> **Excellent**
2. When rating is from **4.0-4.4** ---> **Very Good**
3. When rating is from **3.5-3.9** ---> **Good**
4. When rating is from **2.5-3.4** ---> **Average**
5. When rating is from **1.8-2.4** ---> **Poor**
6. When rating is **0** ---> **Not Rated**

	Aggregate rating	Rating color	Rating text	Rating 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	498
18	3.5	Yellow	Good	480
19	3.6	Yellow	Good	458

Question 3:

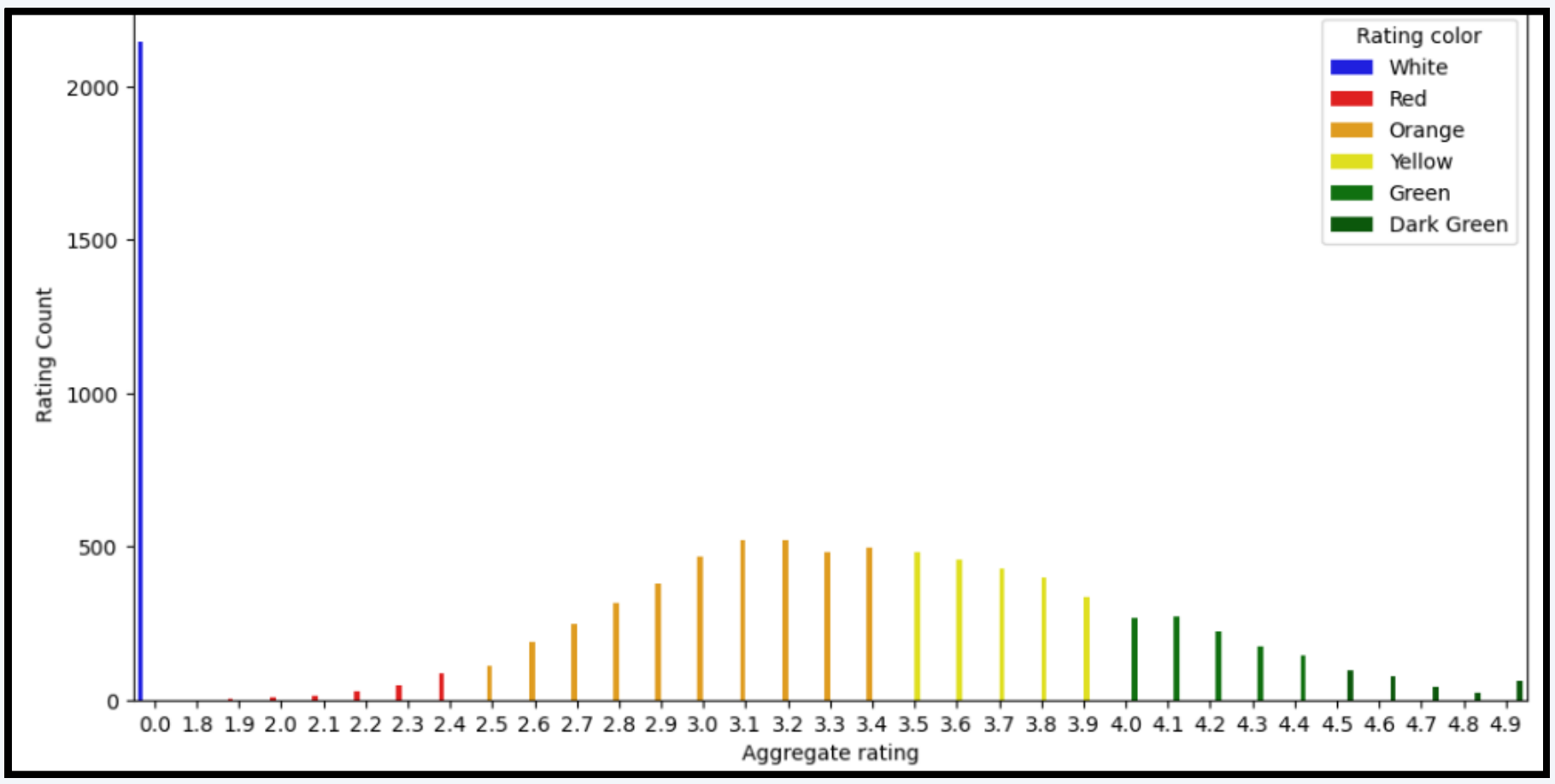
Display Aggregate Rating vs Rating count?

```
matplotlib.rcParams['figure.figsize']=(12,6)
sns.barplot(x='Aggregate rating',y='Rating Count',data=ratings)
```



Give rating specific color coding

```
sns.barplot(x='Aggregate rating',y='Rating Count',hue='Rating color',data=ratings,palette=['blue','red','orange'
```



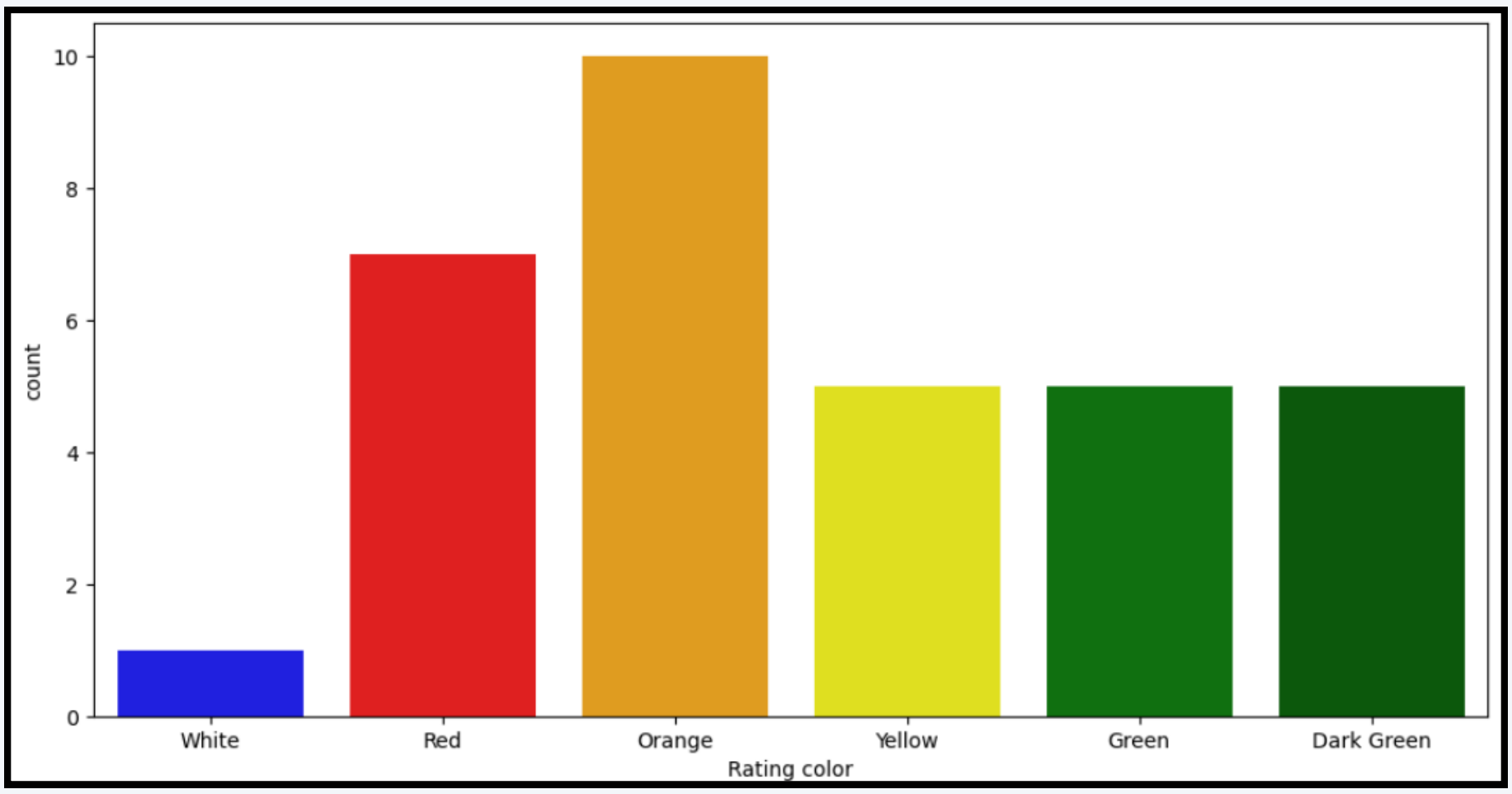
Observation:

1. **Not Rated:** count is very high at **2200**.
2. Maximum Number of ratings are between **2.5 to 3.4**.

Question 4:

Find frequency of color ratings?

```
## Count plot : use for categorical variables
## count is basically telling about the frequency of color for ratings
sns.countplot(x='Rating color',data=ratings,palette=['blue','red','orange','yellow','green','darkgreen'])
```



Question 5:

Find the countries name that has given 0 rating?

```
# We used boolean indexing to filter out 'Aggregate rating' to 0 then group it by 'Country' and used size() to find number o
final_df[final_df['Aggregate rating']==0.0].groupby('Country').size()
```

```
Out[25]: Country
         Brazil                5
         India              2139
         United Kingdom        1
         United States         3
         dtype: int64
```

Observation:

1. Maximum number of **0 rating** is from **Indian customers**.

Question 6:

Find out which currency is used by which country?

```
# To show currency for each country:
final_df[['Country','Currency']].groupby(['Country','Currency']).size().reset_index()
```

	Country	Currency	0
0	Australia	Dollar($)	24
1	Brazil	Brazilian Real(R$)	60
2	Canada	Dollar($)	4
3	India	Indian Rupees(Rs.)	8652
4	Indonesia	Indonesian Rupiah(IDR)	21
5	New Zealand	NewZealand($)	40
6	Phillipines	Botswana Pula(P)	22
7	Qatar	Qatari Rial(QR)	20
8	Singapore	Dollar($)	20
9	South Africa	Rand(R)	60
10	Sri Lanka	Sri Lankan Rupee(LKR)	20
11	Turkey	Turkish Lira(TL)	34
12	UAE	Emirati Diram(AED)	60
13	United Kingdom	Pounds(□£)	80
14	United States	Dollar($)	434

Question 7:

Which countries do have online deliveries option?

```
final_df[['Has Online delivery','Country']].groupby(['Has Online delivery','Country']).size().reset_index()
```

	Has Online delivery	Country	0
0	No	Australia	24
1	No	Brazil	60
2	No	Canada	4
3	No	India	6229
4	No	Indonesia	21
5	No	New Zealand	40
6	No	Phillipines	22
7	No	Qatar	20
8	No	Singapore	20
9	No	South Africa	60
10	No	Sri Lanka	20

Observation:

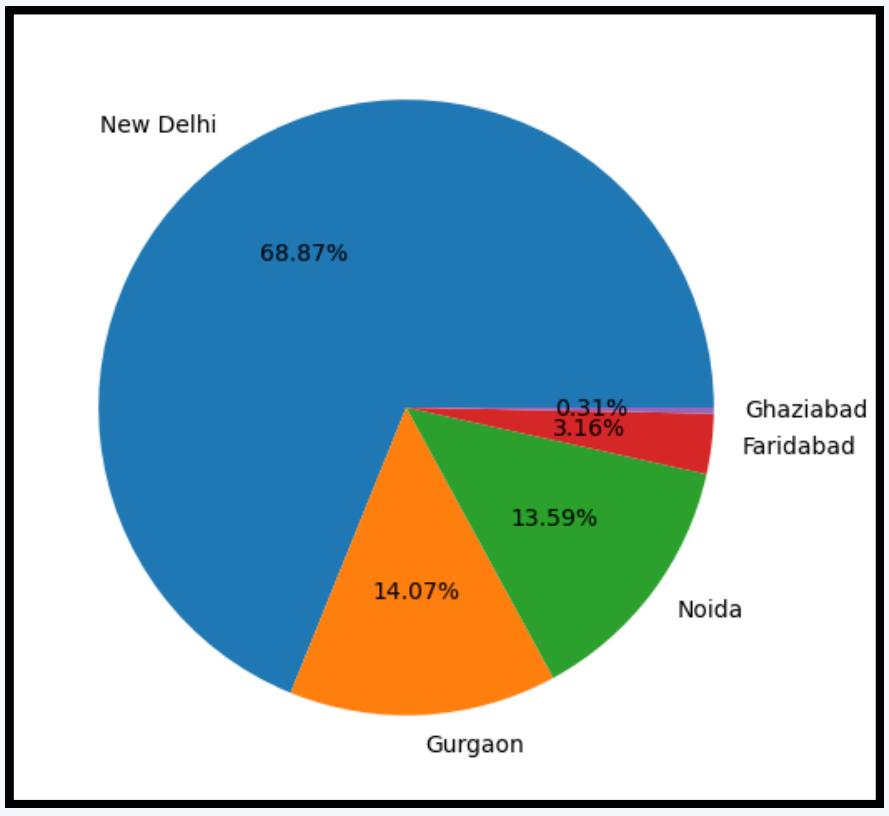
1. Online deliveries are only available in **India** and **UAE**.

Question 8:

Create a pie chart for cities distribution?

```
city_values = final_df.City.value_counts().values #array for values for city
city_labels = final_df.City.value_counts().index #labels for city names
```

```
# To plot a pie chart for city distribution
plt.pie(city_values[:5],labels=city_labels[:5],autopct='%1.2f%%')
```



Observation:

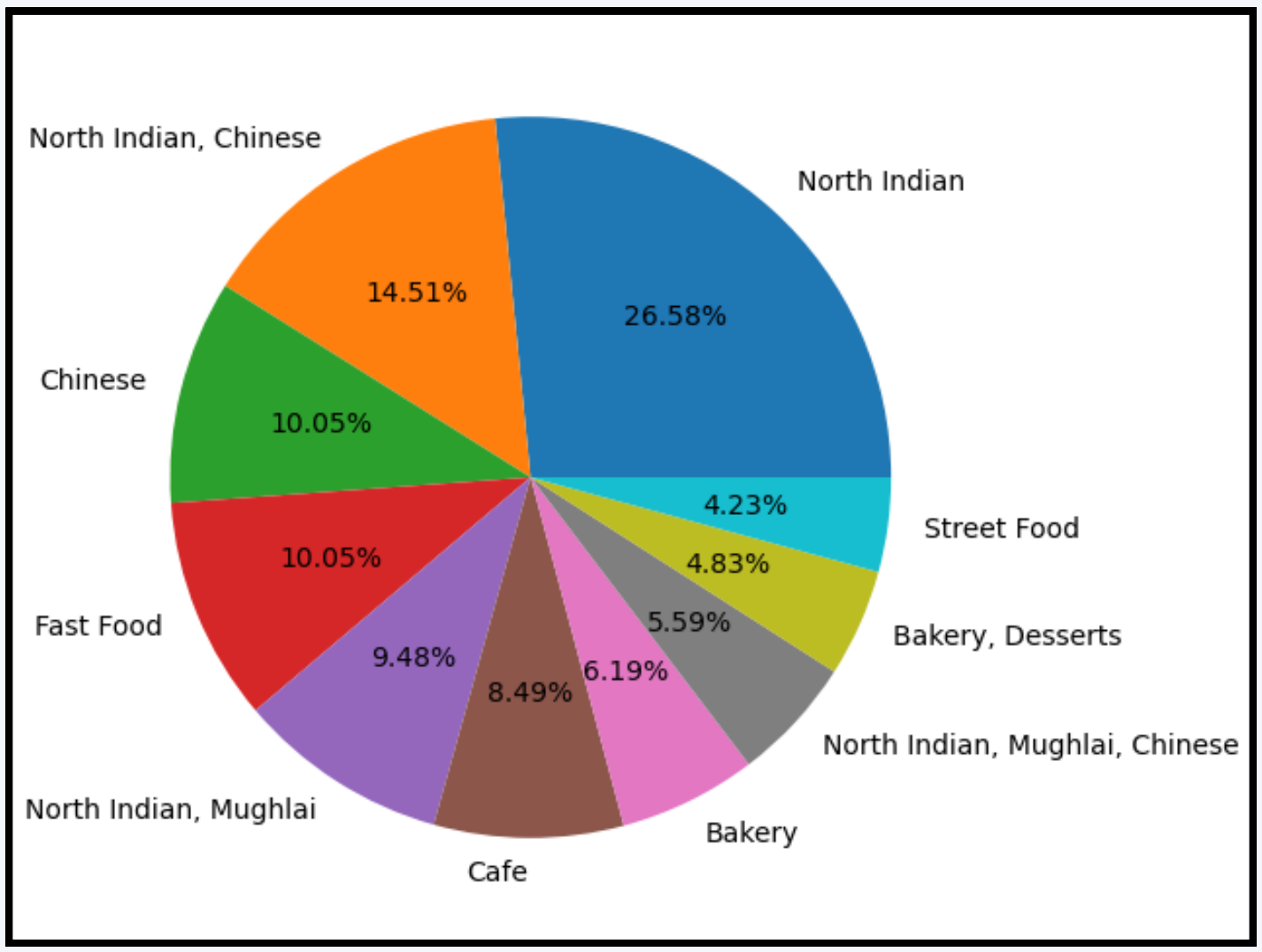
1. **New Delhi** has highest cities distribution at **68.87%**, followed by **Gurgaon** at **14.07%**, and **Ghaziabad** having least at **0.31%.**

Question 8:

Find the top 10 cuisines?

```
cuisines_count = final_df.Cuisines.value_counts().values
cuisines_labels = final_df.Cuisines.value_counts().index
```

```
plt.pie(cuisines_count[:10],labels=cuisines_labels[:10],autopct='%1.2f%%')
```



Observation:

1. **North Indian** is the top cuisine at **26.58%,** followed by **North Indian, Chinese** at **26.58%,** and **Street Food** having at bottom at **4.23%**

ZOMATO

CASE STUDY Completed!