Zomato Exploratory Data Analysis

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

```
import warnings
warnings.filterwarnings('ignore')
```

```
from google.colab import files
uploaded = files.upload()
```

Choose files zomato.csv

- **zomato.csv**(text/csv) - 2257316 bytes, last modified: 07/04/2023 - 100% done

```
Saving zomato.csv to zomato.csv
```

```
data_zomato = pd.read_csv("zomato.csv", encoding="latin-1")
data_zomato.head(2)
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude
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
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

2 rows × 21 columns

```
data_zomato.shape
```

```
(9551, 21)
```

```
from google.colab import files
uploaded = files.upload()
```

```
Choose files Country-Code.xlsx
```

- **Country-Code.xlsx**(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 8783 bytes, la

```
Saving Country-Code.xlsx to Country-Code.xlsx
```

```
data_country = pd.read_excel("Country-Code.xlsx")
data_country.head(5)
```

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

```
data_country.shape
```

```
(15, 2)
```

Merging Both the tables on Country Code

```
df = pd.merge(data_zomato, data_country, on= 'Country Code')
df.head(5)
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	L
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...	1
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...	1
2	6300002	Heat - Edsa	162	Mandaluyong	Edsa Shangri-La, 1 Garden	Edsa Shangri-La, Ortigas	Edsa Shangri-La, Ortigas	1

Now let see the columns of the data -- Datatypes of the each columns and basics stasticts of all numerical columns

```
df.columns
```

```
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')
```

4	6314302	Sambo Kojin	162	Mandaluyong	Atrium	Ortigas	Ortigas,	1

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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: 1.7+ MB
```

```
df.describe()
```

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	P
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	
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	

Understanding the data

1. Checking for any missing values

```
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
Rating text             0
Votes                   0
Country                 0
dtype: int64
```

1.1 This willgive the name of all columns having null values

```
[column for column in df.columns if df[column].isnull().sum()>0]
```

```
['Cuisines']
```

1.2 Visualisation

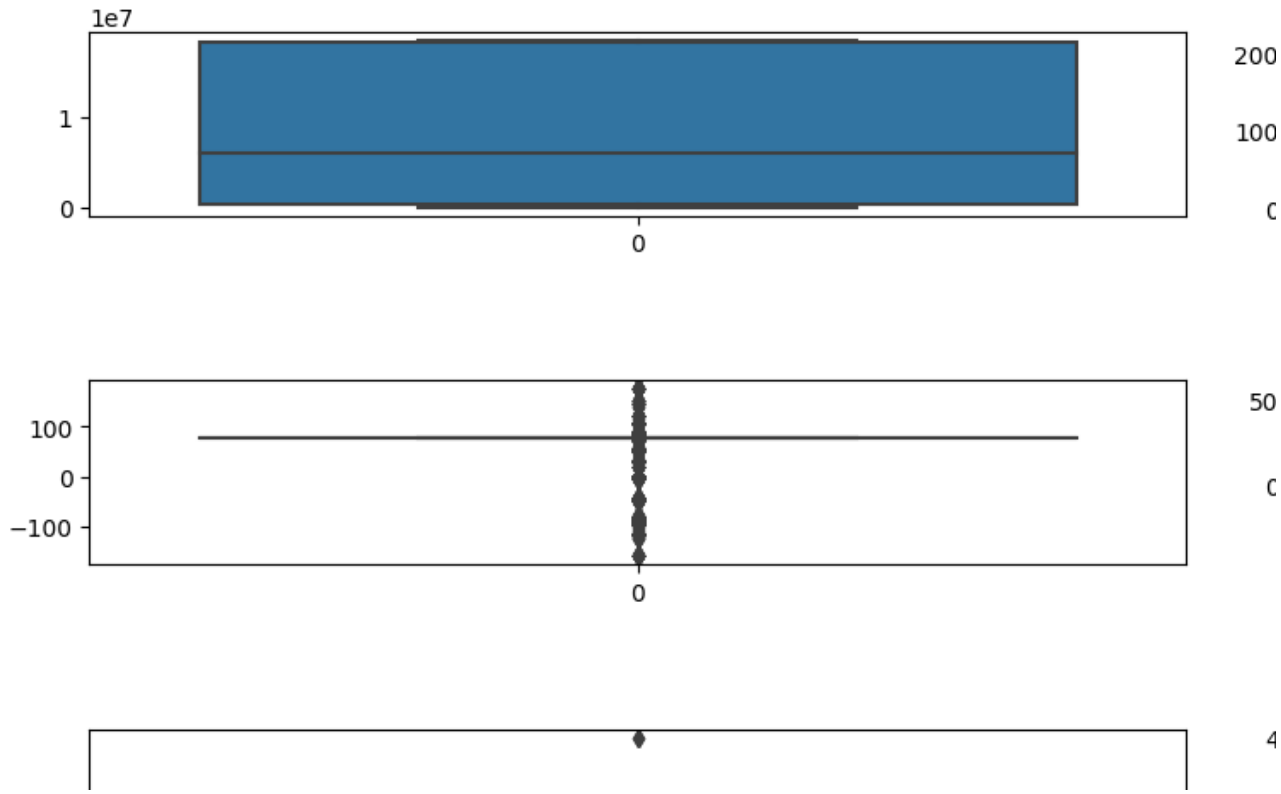
```
matplotlib.rcParams['figure.figsize'] = (14, 6)
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
plt.show()

# matplotlib.rcPrams['figure.figsize'] = (14,6)
# sns.heatmap(df.isnull(),yticklabels=False, cbar= False,cmap='Viridis')
# plt.show()
```

Observation: Only Column Cuisines have missing values

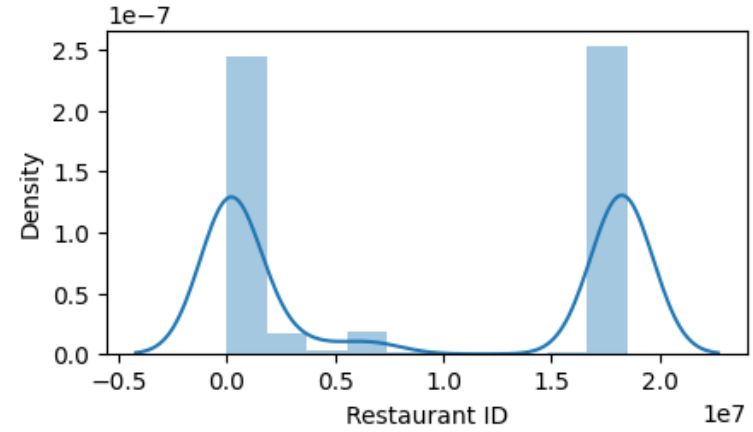
2.*Checking for outliers*

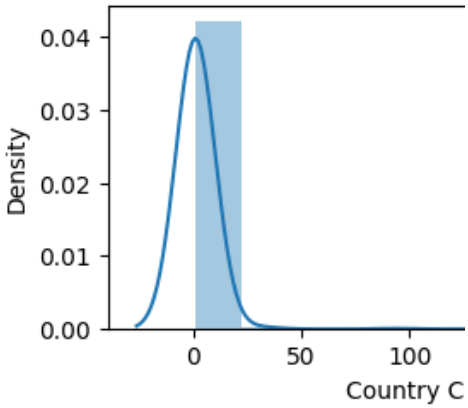
2.1. Visulation -- box plot

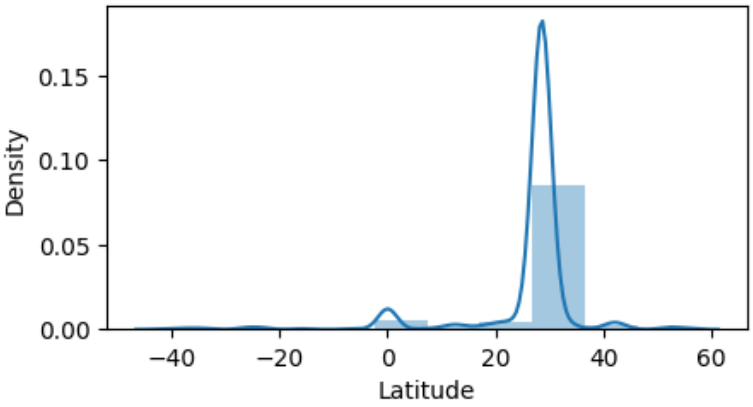
```
numerical_features = [feature for feature in df.columns if df[feature].dtype==int or d


matplotlib.rcParams['figure.figsize'] = (15, 8)
for i in range(8):
  plt.subplot(4,2,i+1)
  sns.boxplot(df[numerical_features[i]])
  plt.subplots_adjust(left=0.1,
                      bottom=0.1,
                      right=1,
                      top=1,
                      wspace=0.1,
                      hspace=0.9)
plt.show()
```

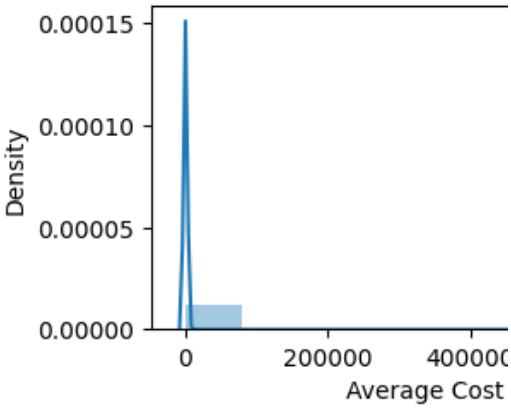


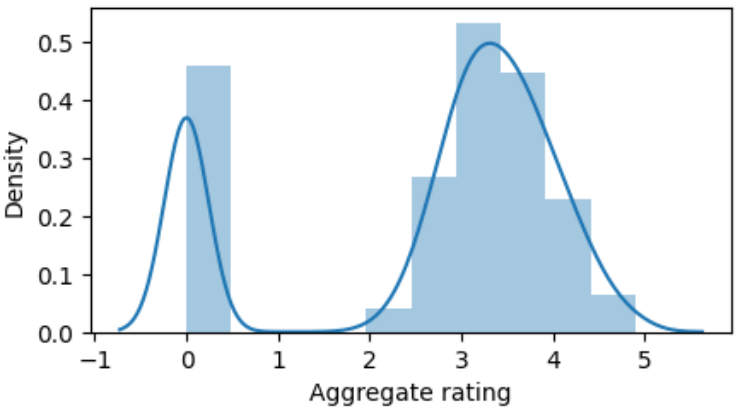

```
matplotlib.rcParams['figure.figsize'] = (15, 8)
for i in range(8):
    plt.subplot(3,3,i+1)
    plt.subplots_adjust(left=0.1,
                        bottom=0.1,
                        right=1,
                        top=1,
                        wspace=0.3,
                        hspace=0.4)
    sns.distplot(df[numerical_features[i]], bins=10)
plt.show()
# matplotlib.rcParams['figure.figsize'] = (8, 4)
df.hist(color='k',
       bins=30,
       figsize=(15,10))
plt.show()
```

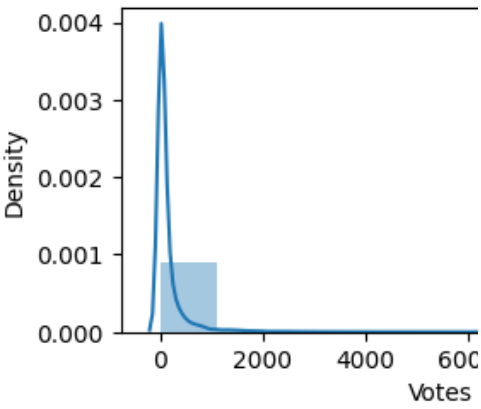

1e−7
Density
Restaurant ID
1e7


Density
Country C

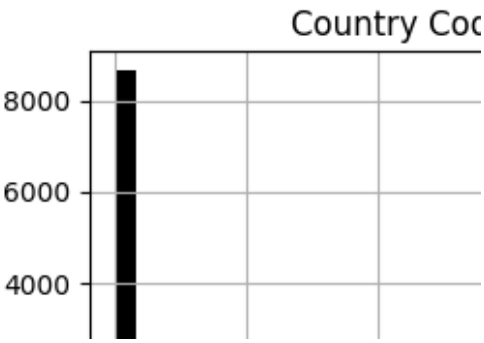

Density
Latitude


Density
Average Cost


Density
Aggregate rating


Density
Votes


Restaurant ID

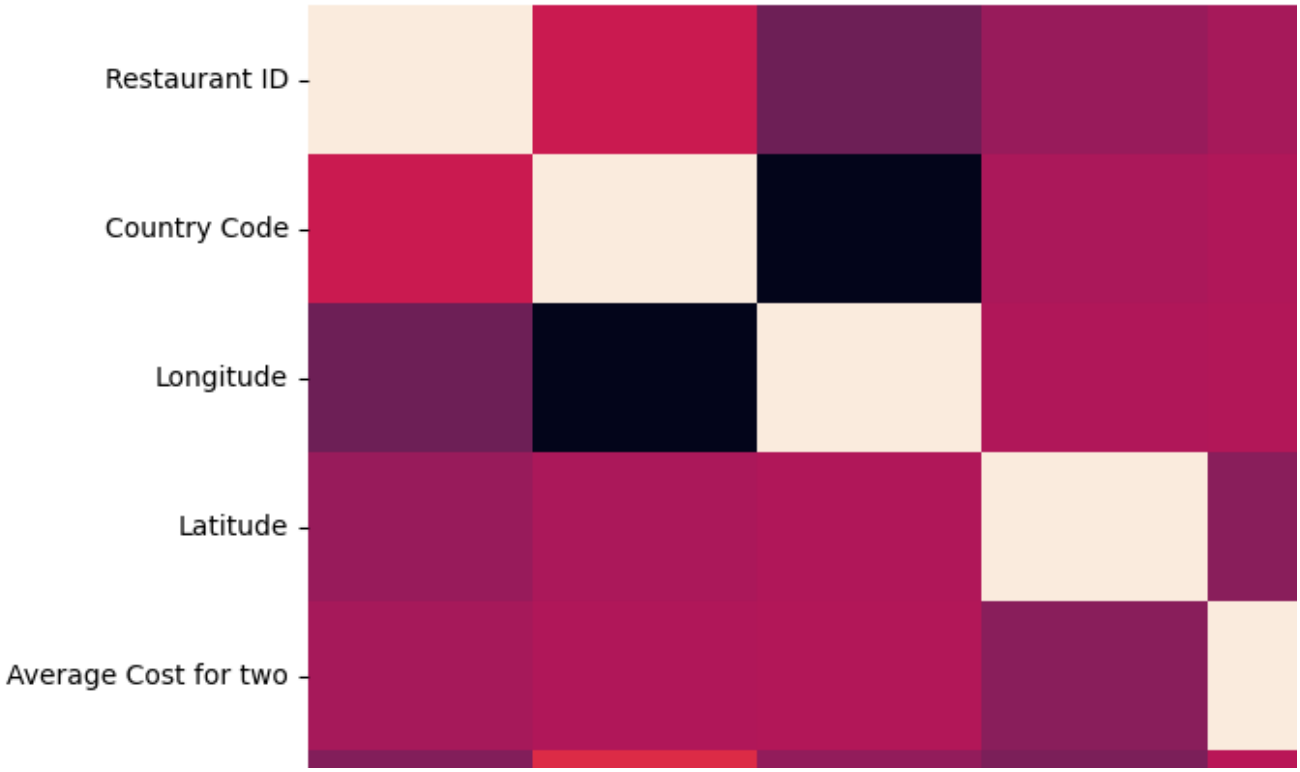

Country Coc

```
df.corr()
```

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost
Restaurant ID	1.000000	0.148471	-0.226081	-0.052081	
Country Code	0.148471	1.000000	-0.698299	0.019792	
Longitude	-0.226081	-0.698299	1.000000	0.043207	
Latitude	-0.052081	0.019792	0.043207	1.000000	
Average Cost for two	-0.001693	0.043225	0.045891	-0.111088	
Price range	-0.134540	0.243327	-0.078939	-0.166688	
Aggregate rating	-0.326212	0.282189	-0.116818	0.000516	
Votes	-0.147023	0.154530	-0.085101	-0.022962	

3.2. Visualisation

```
matplotlib.rcParams['figure.figsize'] = (15, 8)
sns.heatmap(df.corr())
plt.show()
```

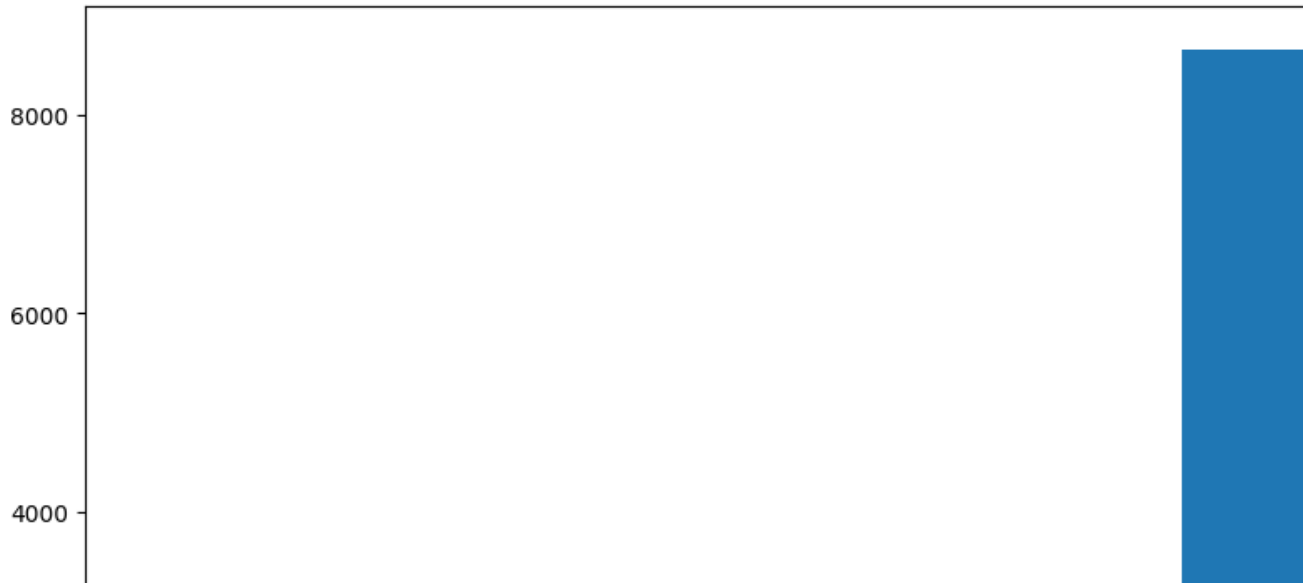


Data Analysis -- Answering Questions

1. Which country have the highest transaction?

```
country_names=df.Country.value_counts().index
country_val=df.Country.value_counts().values
```

```
matplotlib.rcParams['figure.figsize'] = (18,7)
plt.hist(df['Country'],bins=15)
plt.show()
```



Observation : India have maximum Zomato Transaction followed by USA and then United Kingdoms

2. Rating review

```
ratings=df.groupby(['Aggregate rating',
                    'Rating color','Rating text']).size().reset_index().rename(columns

ratings.head()
```

	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

Observation : When Rating is between 4.5 to 4.9 ---> Excellent When Rating is between 4.0 to 4.4 ---> Very good when Rating is between 3.5 to 3.9 ---> Good when Rating is between 3.0 to 2.9 ---> Average when Rating is between 2.0 to 2.4 ---> Poor

```
matplotlib.rcParams['figure.figsize'] = (15, 6)
sns.barplot(x="Aggregate rating",y="Rating Count",data=ratings)
plt.show()
```


```
temp=['Pink' if color=='White' else 'Black' if color=='Dark Green' else color for colo
df['Rating color'][:]=temp
```

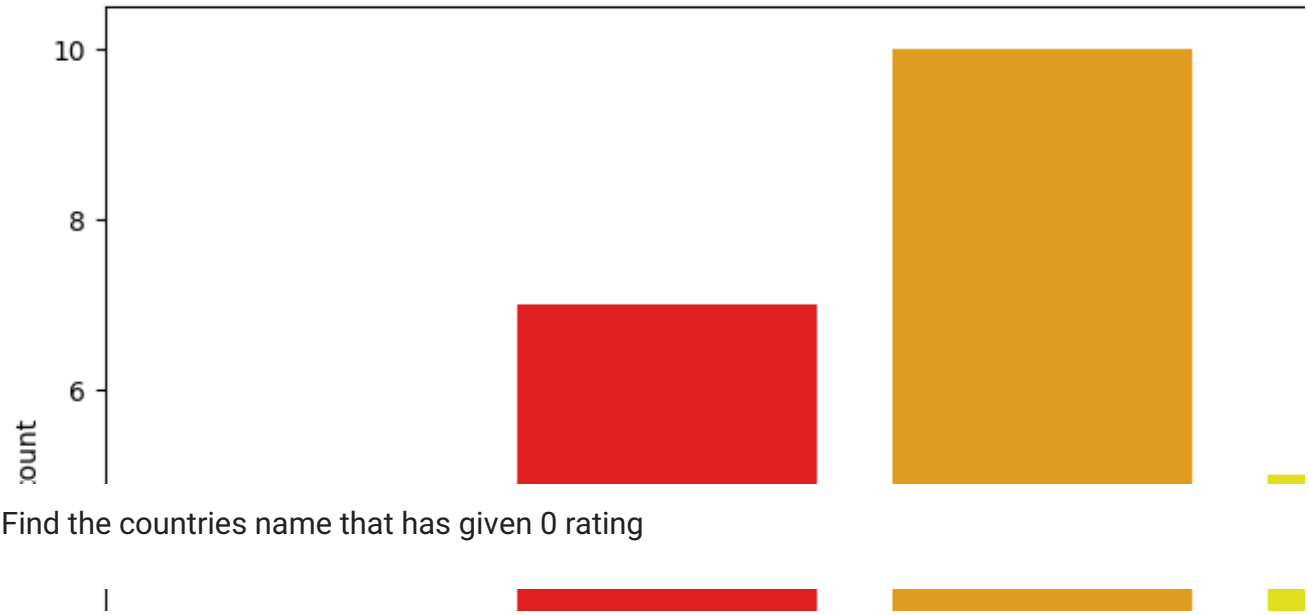
```
sns.barplot(x="Aggregate rating",y="Rating Count",hue='Rating color',
            data=ratings,palette=['pink','red','orange','yellow','green','black'])
plt.show()
```

Observation :

```
Not Rated count is very high -- BLUE LINE
Maximum number of rating are between 2.5 to 3.4
```

```
# Count plot
sns.countplot(x="Rating color",data=ratings,palette=['pink','red','orange','yellow','g
```

```
<Axes: xlabel='Rating color', ylabel='count'>
```



3. Find the countries name that has given 0 rating

```
df[df['Rating color']=='Pink'].groupby('Country').size().reset_index()
```

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

```
df.groupby(['Aggregate rating','Country']).size().reset_index().head()
```

	Aggregate rating	Country	0
0	0.0	Brazil	5
1	0.0	India	2139
2	0.0	United Kingdom	1
3	0.0	United States	3
4	1.8	India	1

Observations Maximum number of 0 ratings are from Indian customers

4. find out which currency is used by which country?

```
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

5. Which Countries do have online deliveries option?

```
df[df['Has Online delivery'] =="Yes"].Country.value_counts()
```

```
India    2423
UAE        28
Name: Country, dtype: int64
```

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

	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

Observations : Online deliveries are available in India and UAE

6	No	Phillipines	22

6. Which cities have the highest transactions?

8	No	Singapore	20

```
df.City.value_counts().index
```

```
Index(['New Delhi', 'Gurgaon', 'Noida', 'Faridabad', 'Ghaziabad',
       'Bhubaneshwar', 'Guwahati', 'Amritsar', 'Lucknow', 'Ahmedabad',
       ...
       'Mayfield', 'Macedon', 'Lorn', 'Lakes Entrance', 'Inverloch',
       'Huskisson', 'Panchkula', 'Forrest', 'Flaxton', 'Chatham-Kent'],
      dtype='object', length=141)
```

14	No	United States	434

```
city_values=df.City.value_counts().values
city_labels=df.City.value_counts().index
```

16	Yes	UAE	28

```
plt.pie(city_values[:5],labels=city_labels[:5],autopct='%1.2f%%')
plt.show()
```

Observation : New Delhi has the highest number of transactions

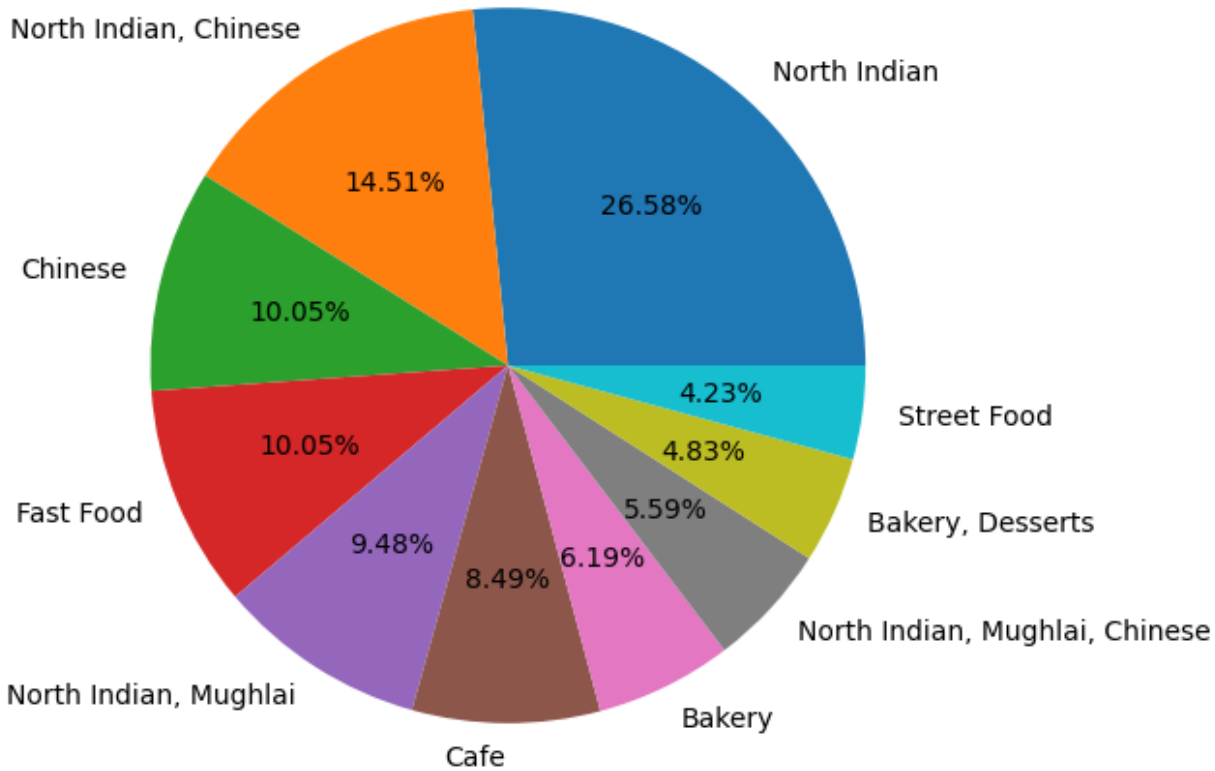
7. Which cuisine have highest sale?

```
df.columns
```

```
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')
```

```
cuisine_count=df.Cuisines.value_counts().values
cuisine_label=df.Cuisines.value_counts().index
```

```
plt.pie(cuisine_count[:10],labels=cuisine_label[:10],autopct='%1.2f%%')
plt.show()
```




```
df[['Cuisines']].groupby(['Cuisines']).size().sort_values(ascending=False)[:10]
```

```
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
dtype: int64
```

Observation : North Indian has the highest sales.

✓ 0s completed at 1:42 AM