Exploratry Data Analysis of Zomato

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
import matplotlib
matplotlib.rcParams['figure.figsize']=(12,6)
```

In [2]:
```
df=pd.read_csv("zomato.csv",encoding='latin-1')
```

In [3]:
```
df.head()
```

Out[3]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitu
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...	121.027
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...	121.014
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...	121.056
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...	121.056
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...	121.057

5 rows × 21 columns

In [4]:
```
df.shape
```

Out[4]:
```
(9551, 21)
```

In [5]:
```
df.columns
```

Out[5]:
```
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 [6]:
```
len(df.columns)
```

Out[6]:
```
21
```

In [7]:
```
df.shape[0]
```

Out[7]:
```
9551
```

In [8]:
```
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 [9]:
```
df.dtypes
```

Out[9]:
```
Restaurant ID           int64
Restaurant Name        object
Country Code            int64
City                   object
Address                object
Locality               object
Locality Verbose       object
Longitude             float64
Latitude              float64
Cuisines               object
Average Cost for two    int64
Currency               object
Has Table booking      object
Has Online delivery    object
Is delivering now      object
Switch to order menu   object
Price range             int64
Aggregate rating      float64
Rating color           object
Rating text            object
Votes                   int64
dtype: object
```

In [10]:
```
df.describe()
```

Out[10]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggreg rat
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.666
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.516
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.500
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.200
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.700
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900

In data Analysis what we can do is

1. Find Missing value
2. Explore about the numerical variables
3. Explore about the categorical variables
4. Finding the relationship between features

In [11]:
```
df.isnull().sum()
```

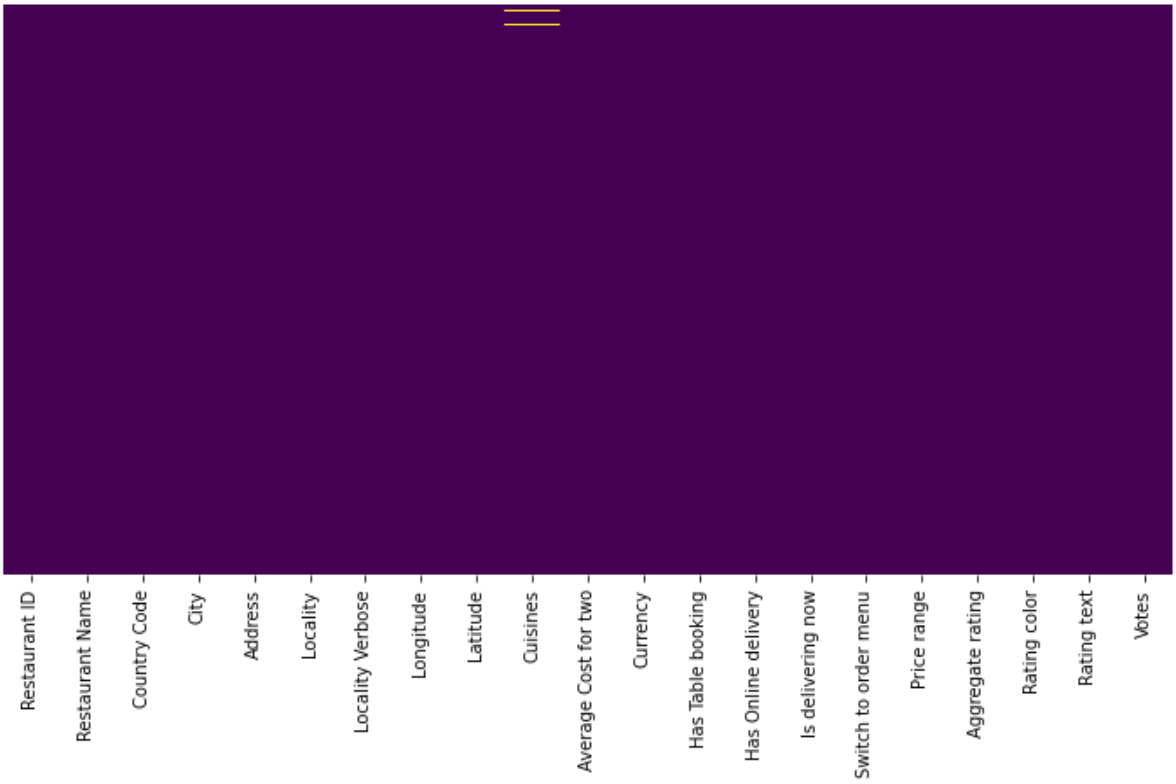
Out[11]:
```
Restaurant ID         0
Restaurant Name       0
Country Code          0
City                  0
Address               0
Locality              0
Locality Verbose      0
Longitude             0
Latitude              0
Cuisines              9
Average Cost for two  0
Currency              0
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 [12]:
```
[feature for feature in df.columns if df[feature].isnull().sum()>0]
```

Out[12]:
```
['Cuisines']
```

In [13]:
```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[13]:
```
<AxesSubplot:>
```



In [14]:
```
df_country=pd.read_excel("Country-Code.xlsx")
```

In [15]:
```
df_country
```

Out[15]:

	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

In [16]:
```
final_df=pd.merge(df,df_country,on='Country Code',how='left')
```

In [17]:
```
final_df.head(5)
```

Out[17]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitu
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...	121.027
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...	121.014
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...	121.056
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...	121.056
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...	121.057

5 rows × 22 columns

In [18]:
```
## To check data types
final_df.dtypes
```

Out[18]:
```
Restaurant ID             int64
Restaurant Name          object
Country Code              int64
City                     object
Address                  object
Locality                 object
Locality Verbose         object
Longitude               float64
Latitude                float64
Cuisines                 object
Average Cost for two      int64
Currency                 object
Has Table booking        object
Has Online delivery      object
Is delivering now        object
Switch to order menu     object
Price range               int64
Aggregate rating        float64
Rating color             object
Rating text              object
Votes                     int64
Country                  object
dtype: object
```

In [19]:
```
final_df.columns
```

Out[19]:
```
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 [20]:
```
final_df.Country.value_counts()
```

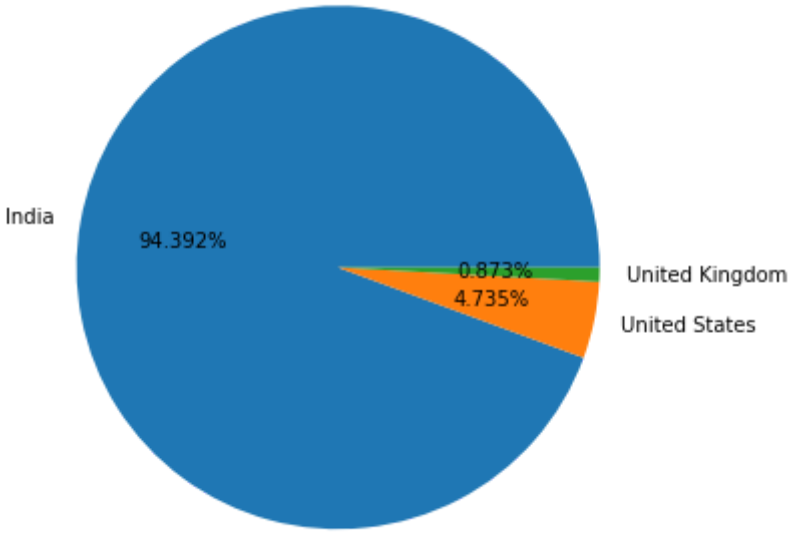
Out[20]:
```
India             8652
United States      434
United Kingdom      80
Brazil              60
UAE                 60
South Africa        60
New Zealand         40
Turkey              34
Australia           24
Phillipines         22
Indonesia           21
Singapore           20
Qatar               20
Sri Lanka           20
Canada               4
Name: Country, dtype: int64
```

In [21]:
```
country_name=final_df.Country.value_counts().index
```

In [22]:
```
country_value=final_df.Country.value_counts().values
```

In [23]:
```
## Pie Chart -Top 3 country using Zomato
plt.pie(country_value[:3],labels=country_name[:3],autopct='%1.3f%%')
```

Out[23]:
```
([<matplotlib.patches.Wedge at 0x212956e3370>,
  <matplotlib.patches.Wedge at 0x212956e3a90>,
  <matplotlib.patches.Wedge at 0x212956f10d0>],
 [Text(-1.0829742700952103, 0.19278674827836725, 'India'),
  Text(1.077281715838356, -0.22240527134123297, 'United States'),
  Text(1.0995865153823035, -0.03015783794312073, 'United Kingdom')],
 [Text(-0.590713238233751, 0.10515640815183668, '94.392%'),
  Text(0.5876082086391032, -0.12131196618612707, '4.735%'),
  Text(0.5997744629358018, -0.01644972978715676, '0.873%')])
```



Observation: Zomato maximum recorfs or transaction are from india, USA and Unilted State

In [24]:
```
final_df.columns
```

Out[24]:
```
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 [25]:
```
final_df.groupby(['Aggregate rating','Rating color','Rating text']).size()
```

Out[25]:

Aggregate rating	Rating color	Rating text	
0.0	White	Not rated	2148
1.8	Red	Poor	1
1.9	Red	Poor	2
2.0	Red	Poor	7
2.1	Red	Poor	15
2.2	Red	Poor	27
2.3	Red	Poor	47
2.4	Red	Poor	87
2.5	Orange	Average	110
2.6	Orange	Average	191
2.7	Orange	Average	250
2.8	Orange	Average	315
2.9	Orange	Average	381
3.0	Orange	Average	468
3.1	Orange	Average	519
3.2	Orange	Average	522
3.3	Orange	Average	483
3.4	Orange	Average	498
3.5	Yellow	Good	480
3.6	Yellow	Good	458
3.7	Yellow	Good	427
3.8	Yellow	Good	400
3.9	Yellow	Good	335
4.0	Green	Very Good	266
4.1	Green	Very Good	274
4.2	Green	Very Good	221
4.3	Green	Very Good	174
4.4	Green	Very Good	144
4.5	Dark Green	Excellent	95
4.6	Dark Green	Excellent	78
4.7	Dark Green	Excellent	42
4.8	Dark Green	Excellent	25
4.9	Dark Green	Excellent	61

dtype: int64

In [26]:
```
Ratings=final_df.groupby(['Aggregate rating','Rating color','Rating text']).size()
```

In [27]:
```
Ratings
```

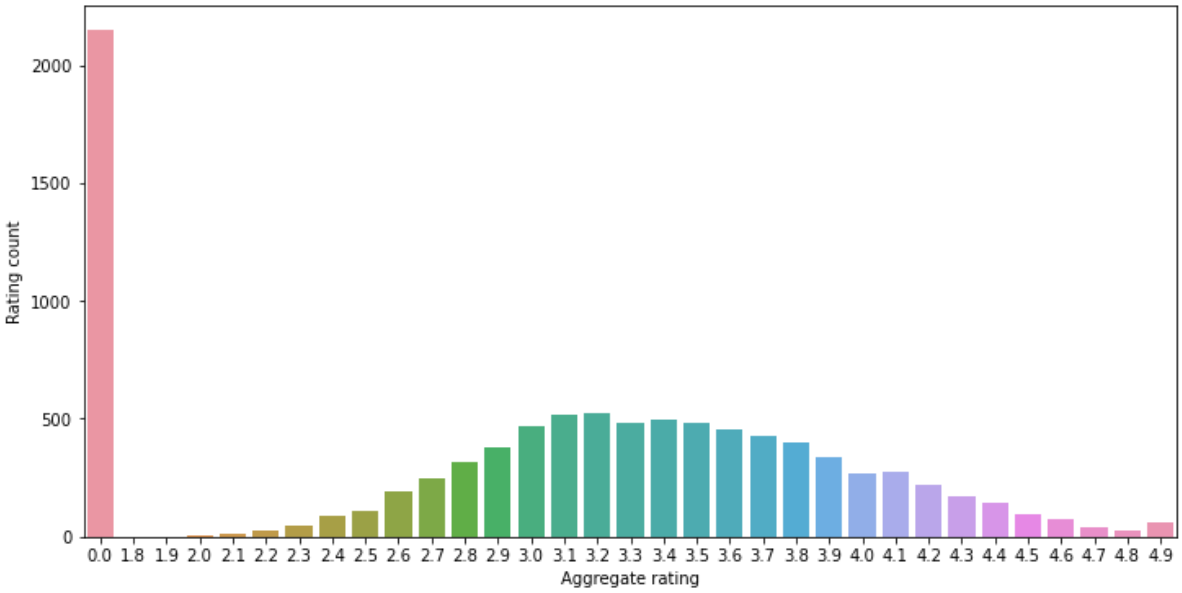
Out[27]:

	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
20	3.7	Yellow	Good	427
21	3.8	Yellow	Good	400
22	3.9	Yellow	Good	335
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	144
28	4.5	Dark Green	Excellent	95
29	4.6	Dark Green	Excellent	78
30	4.7	Dark Green	Excellent	42
31	4.8	Dark Green	Excellent	25
32	4.9	Dark Green	Excellent	61

In [28]:
```
## Observations
```

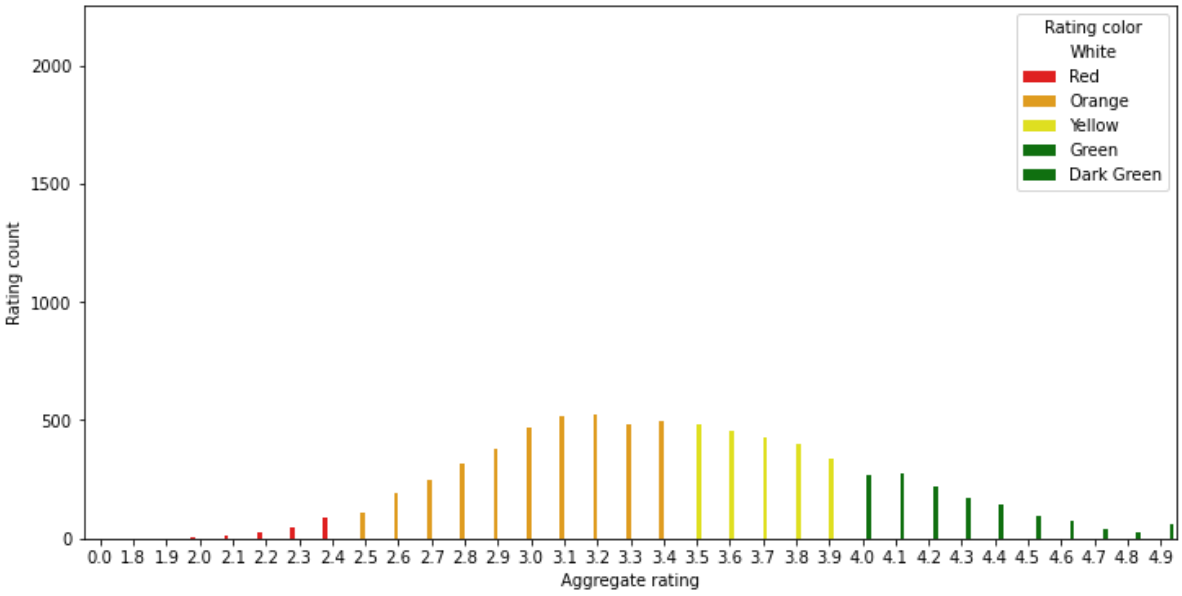
In [29]:
```
sns.barplot(x='Aggregate rating',y='Rating count',data=Ratings)
```

Out[29]: <AxesSubplot:xlabel='Aggregate rating', ylabel='Rating count'>



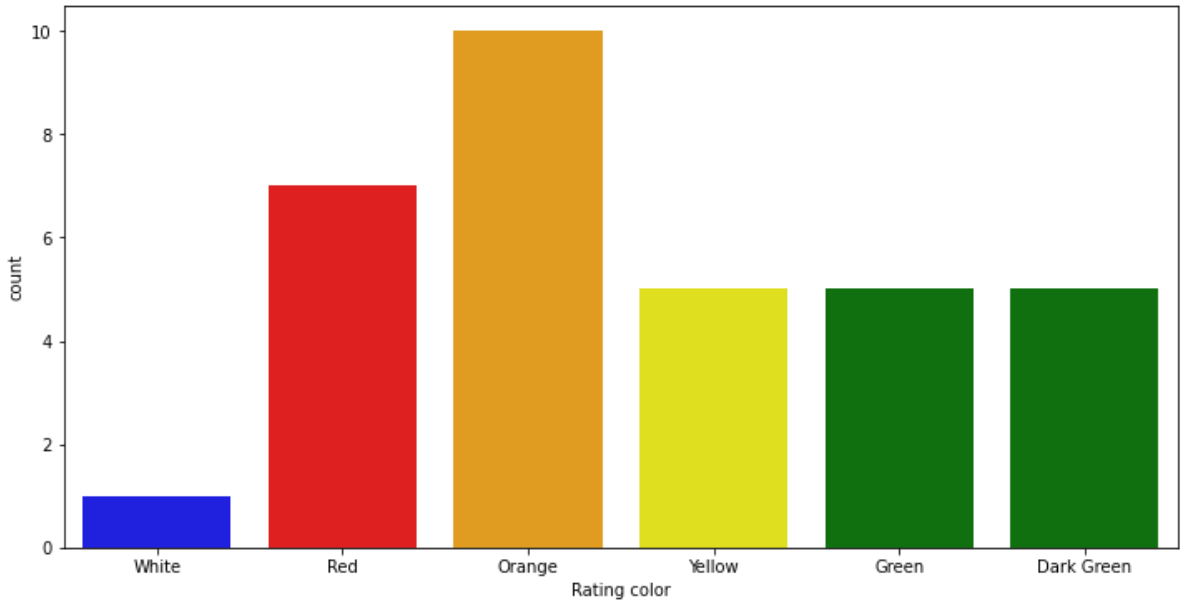
In [30]:
```
sns.barplot(x='Aggregate rating',y='Rating count',hue='Rating color', data=Ratings
```

Out[30]: <AxesSubplot:xlabel='Aggregate rating', ylabel='Rating count'>



In [31]:
```
sns.countplot(x='Rating color',data=Ratings,palette=['blue','red','orange','yellow
```

Out[31]: <AxesSubplot:xlabel='Rating color', ylabel='count'>



Find the countries name that has given 0 rating

In [32]:
```
final_df[final_df['Aggregate rating']==0].groupby('Country').size().reset_index().r
```

Out[32]:

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

In [33]:
```
final_df[final_df['Aggregate rating']==0].groupby(['Aggregate rating','Country']).s
```

Out[33]:

	Aggregate rating	Country	No. Time
0	0.0	Brazil	5
1	0.0	India	2139
2	0.0	United Kingdom	1
3	0.0	United States	3

In [34]:
```
final_df.columns
```

Out[34]:
```
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 [35]:
```
final_df.groupby(['Country','Currency']).size().reset_index().rename(columns={0:'N
```

Out[35]:

	Country	Currency	No. Time
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

In [36]:
```
final_df[final_df['Has Online delivery']=='Yes'].groupby(['Country','Has Online de
```

Out[36]:

	Country	Has Online delivery	No. Time
0	India	Yes	2423
1	UAE	Yes	28

In [37]:
```
final_df.groupby(['Country','Has Online delivery']).size().reset_index().rename(co
```

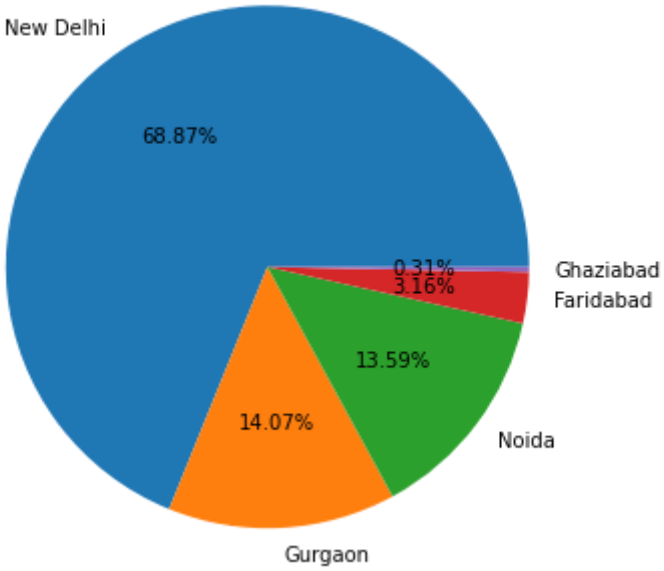
Out[37]:

	Country	Has Online delivery	No. Time
0	Australia	No	24
1	Brazil	No	60
2	Canada	No	4
3	India	No	6229
4	India	Yes	2423
5	Indonesia	No	21
6	New Zealand	No	40
7	Phillipines	No	22
8	Qatar	No	20
9	Singapore	No	20
10	South Africa	No	60
11	Sri Lanka	No	20
12	Turkey	No	34
13	UAE	No	32
14	UAE	Yes	28
15	United Kingdom	No	80
16	United States	No	434

In [38]:
```
City_name=final_df.City.value_counts().index
City_Value=final_df.City.value_counts().values
```

In [39]:
```
plt.pie(City_Value[:5],labels=City_name[:5],autopct='%1.2f%%')
```

Out[39]:
```
([<matplotlib.patches.Wedge at 0x212973ea340>,
  <matplotlib.patches.Wedge at 0x212973eaa60>,
  <matplotlib.patches.Wedge at 0x212973c81f0>,
  <matplotlib.patches.Wedge at 0x212973f5850>,
  <matplotlib.patches.Wedge at 0x212973f5fa0>],
 [Text(-0.6145352824185932, 0.9123301960708633, 'New Delhi'),
  Text(0.0623675251198054, -1.0982305276263407, 'Gurgaon'),
  Text(0.8789045225625368, -0.6614581167535246, 'Noida'),
  Text(1.0922218418223437, -0.13058119407559224, 'Faridabad'),
  Text(1.099946280005612, -0.010871113182029924, 'Ghaziabad')],
 [Text(-0.3352010631374145, 0.497634652402289, '68.87%'),
  Text(0.0340186500653484, -0.5990348332507311, '14.07%'),
  Text(0.47940246685229276, -0.36079533641101336, '13.59%'),
  Text(0.5957573682667329, -0.07122610585941394, '3.16%'),
  Text(0.5999706981848791, -0.005929698099289049, '0.31%')])
```



Assigment

Final the top 10 cuisines

In [40]:
```
final_df.columns
```

Out[40]:
```
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 [41]:
```
final_df.Cuisines.value_counts().head(10)
```

Out[41]:
```
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: Cuisines, dtype: int64
```

In [ ]: