In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [7]:
```
df = pd.read_csv('zomato.csv', encoding='latin-1')
df.head()
```

Out[7]:

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

5 rows × 21 columns

In [9]:
```
df.columns
```

Out[9]:
```
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 [11]:
```
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 [15]:
```
df.describe()
```

Out[15]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Pric
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	955
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	

In data analytics what all things we do

1. Missing Values
2. Explore about Numerical Variable
3. Explore about Categorical Variable
4. Finding Relationship between Features

In [30]:
```
df.shape
```

Out[30]:
```
(9551, 21)
```

In [18]:
```
df.isnull().sum()
```

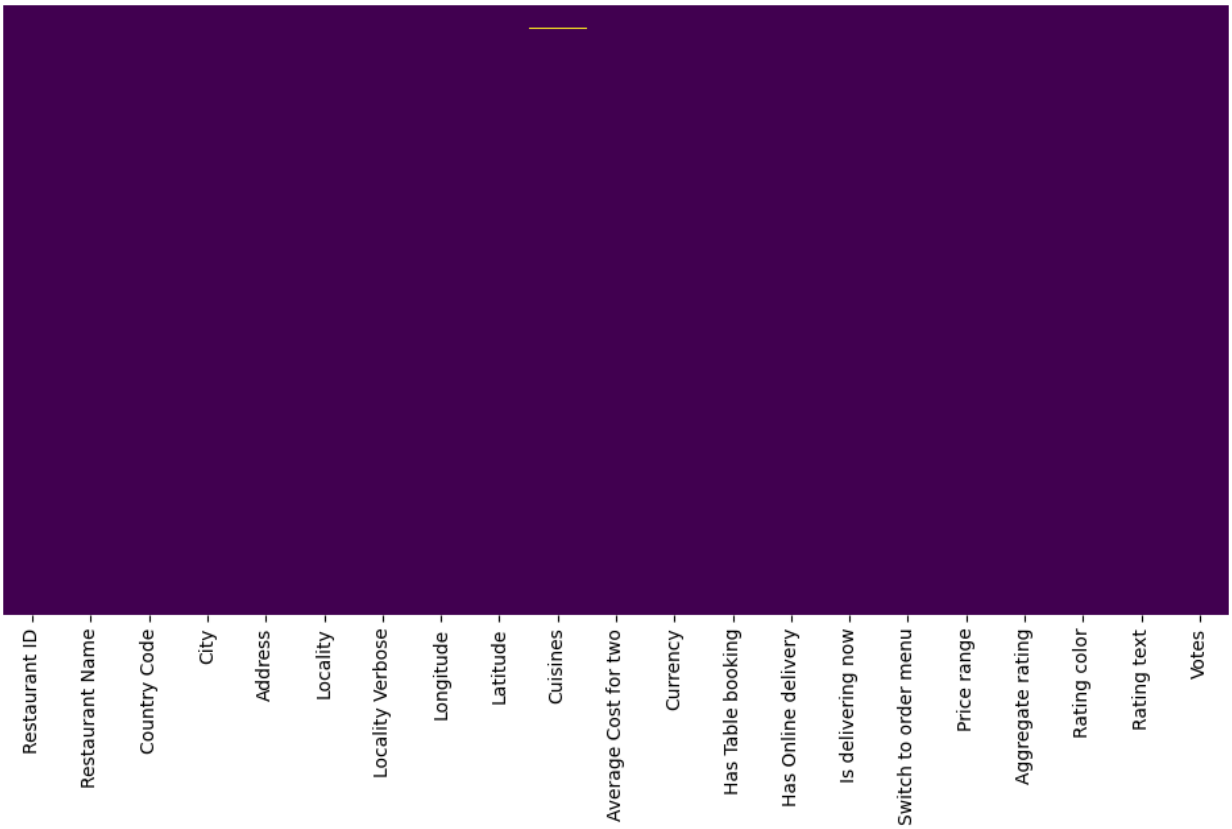
Out[18]:
```
Restaurant ID           0
Restaurant Name         0
Country Code            0
City                    0
Address                 0
Locality                0
Locality Verbose        0
Longitude               0
Latitude                0
Cuisines                9
Average Cost for two    0
Currency                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 [24]:
```
# list comphrension
[features for features in df.columns if df[features].isnull().sum()>0]
```

Out[24]:
```
['Cuisines']
```

In [92]:
```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[92]:
```
<Axes: >
```



In [40]:
```
df_country = pd.read_excel('Country-code.xlsx')
df_country.head()
```

Out[40]:

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

In [42]:
```
df.columns
```

Out[42]:
```
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 [114…
```
final_df = pd.merge(df,df_country,on='Country Code', how='left')
```

In [116…
```
final_df
```

Out[116…

	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
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City
...	...	...	...	...	...	...
9546	5915730	NamlÛ± Gurme	208	ÛÁstanbul	Kemanke◈ô Karamustafa Pa◈ôa Mahallesi, RÛ±htÛ±...	Karakí_y
9547	5908749	Ceviz AÛôacÛ±	208	ÛÁstanbul	Ko◈ôuyolu Mahallesi, Muhittin îstí_ndaÛô Cadd...	Ko◈ôuyolu
9548	5915807	Huqqa	208	ÛÁstanbul	Kuruí_e◈ôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_e◈ôme
9549	5916112	A◈ô◈ôk Kahve	208	ÛÁstanbul	Kuruí_e◈ôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_e◈ôme

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality
9550	5927402	Walter's Coffee Roastery	208	ÛÁstanbul	CafeaÛôa Mahallesi, BademaltÛ± Sokak, No 21/B,...	Moda

9551 rows × 22 columns

In [112…
```
## To check datatypes
final_df.dtypes
```

Out[112…
```
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 [110…
```
final_df.Country.value_counts()
```

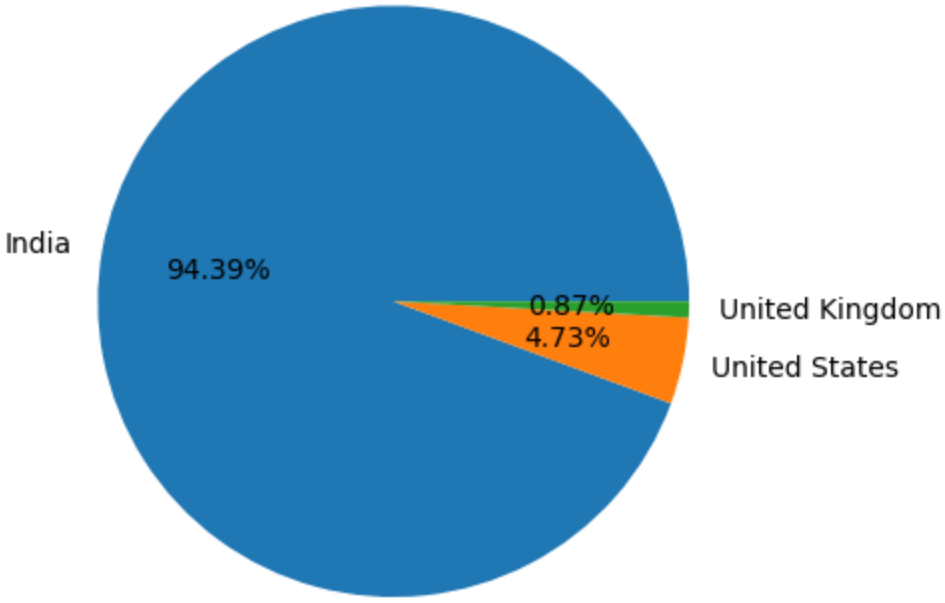
Out[110…
```
Country
India             8652
United States      434
United Kingdom      80
Brazil              60
UAE                 60
South Africa        60
New Zealand         40
Turkey              34
Australia           24
Phillipines         22
Indonesia           21
Singapore           20
Qatar               20
Sri Lanka           20
Canada               4
Name: count, dtype: int64
```

In [56]:
```
country_names =final_df.Country.value_counts().index
```

In [60]:
```
country_val = final_df.Country.value_counts().values
```

In [64]:
```
## Pie Chart  Top 3 countries that uses zomato
plt.pie(country_val[:3],labels=country_names[:3],autopct='%1.2f%%')
```

Out[64]:
```
([<matplotlib.patches.Wedge at 0x22713d0f770>,
  <matplotlib.patches.Wedge at 0x22713a90290>,
  <matplotlib.patches.Wedge at 0x22713d48110>],
 [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, '94.39%'),
  Text(0.5876082086391032, -0.12131196618612707, '4.73%'),
  Text(0.5997744629358018, -0.01644972978715676, '0.87%')])
```



Observation: zomato maximum records or transaction are from India after that US and Then United Kingdoms

In [108…
```
final_df.columns
```

Out[108…
```
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 [75]:
```
ratings = final_df.groupby(['Aggregate rating','Rating color','Rating text']).
```

In [77]:
```
ratings
```

Out[77]:

	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

	Aggregate rating	Rating color	Rating text	Rating Count
31	4.8	Dark Green	Excellent	25
32	4.9	Dark Green	Excellent	61

Obseravtion

1. When rating is between 4.5 to 4.9--> Excellent
2. When rating are between 4.0 to 3.4-->very good
3. when rating is between 3.5 to 3.9-->good
4. when rating is between 2.5 to 3.4-->average
5. when rating is between 1.8 to 2.4-->bad

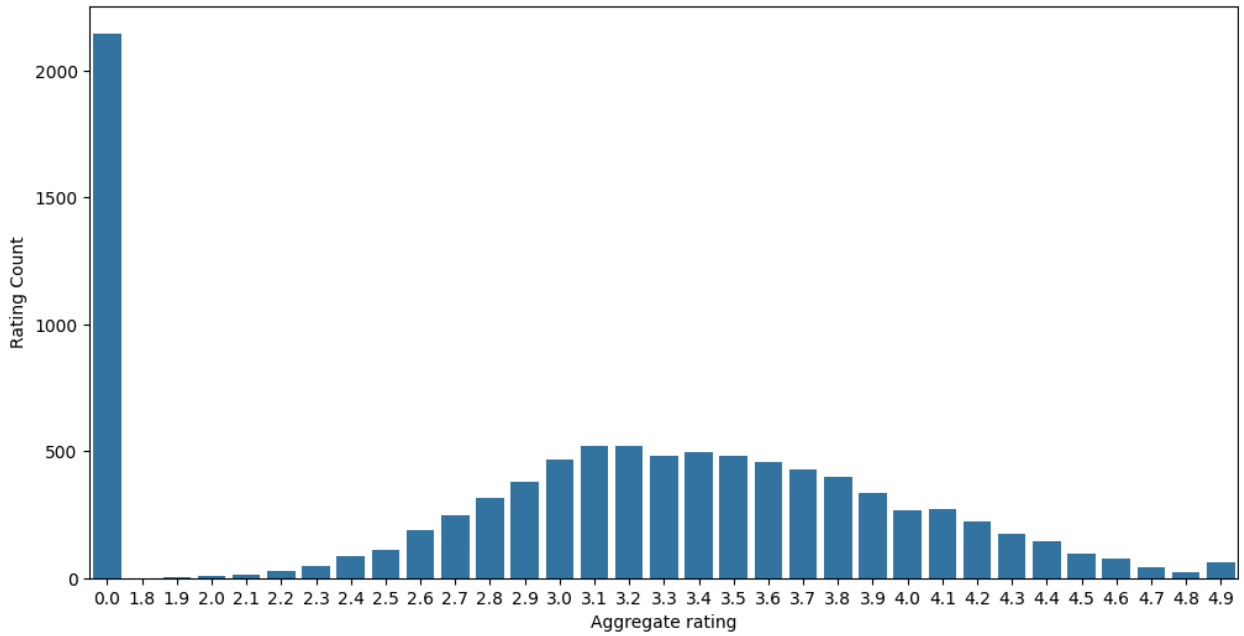
In [80]:
```
ratings.head()
```

Out[80]:

	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

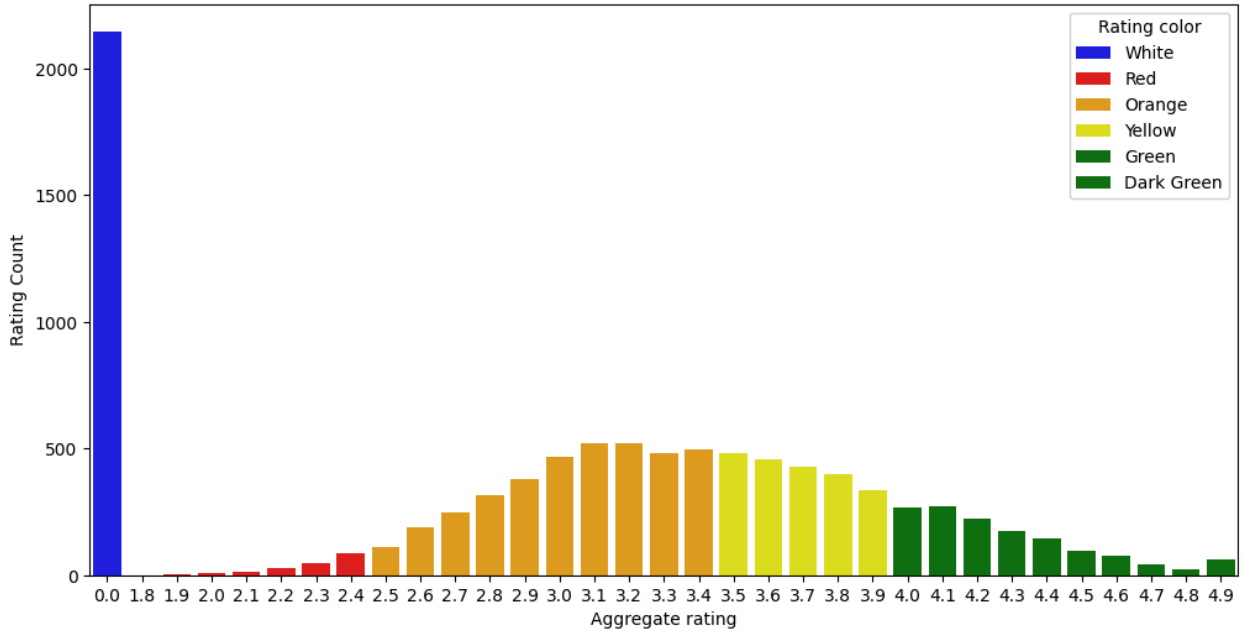
In [90]:
```
import matplotlib
matplotlib.rcParams['figure.figsize'] = (12,6)
sns.barplot(x="Aggregate rating",y="Rating Count",data=ratings)
```

Out[90]:
```
<Axes: xlabel='Aggregate rating', ylabel='Rating Count'>
```



In [98]:
```
sns.barplot(x="Aggregate rating",y="Rating Count",hue = 'Rating color',data=ra
```

Out[98]: <Axes: xlabel='Aggregate rating', ylabel='Rating Count'>



Observation:

1. Not related count is very high
2. maximum number of rating are between 2.5 to 3.4

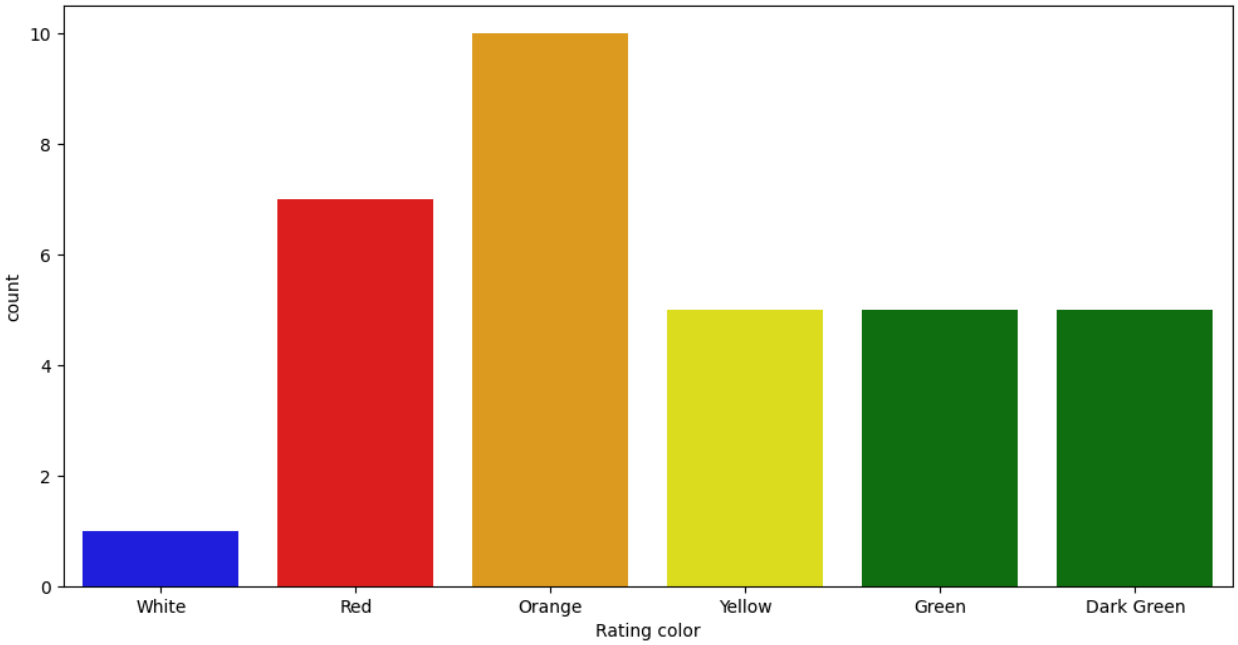
In [102…
```
## Count Plot
sns.countplot(x="Rating color",data=ratings , palette=['blue','red','orange','
```

```
C:\Users\haris\AppData\Local\Temp\ipykernel_17628\2517740499.py:2: FutureWarnin
g:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same e
ffect.

  sns.countplot(x="Rating color",data=ratings , palette=['blue','red','orang
e','yellow','green','green'])
```

Out[102… <Axes: xlabel='Rating color', ylabel='count'>



In [104… ratings

Out[104…

	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

	Aggregate rating	Rating color	Rating text	Rating Count
31	4.8	Dark Green	Excellent	25
32	4.9	Dark Green	Excellent	61

In [124…
```
### FInd the countries name that has given 0 rating
final_df[final_df['Rating color']=='White'].groupby('Country').size().reset_ir
```

Out[124…

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

Observation

1. Maximum number pf 0 ratings are from Indian customers

In [131…
```
## Find out which currency is used bu which country
final_df.columns
```

Out[131…
```
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 [135…
```
final_df[['Country','Currency']].groupby(['Country','Currency']).size().reset_
```

Out[135…

	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($)	20
9	South Africa	Rand(R)	60
10	Sri Lanka	Sri Lankan Rupee(LKR)	20
11	Turkey	Turkish Lira(TL)	34
12	UAE	Emirati Diram(AED)	60
13	United Kingdom	Pounds(Œ£)	80
14	United States	Dollar($)	434

In [147…
```
## WHich country do have online delivery option
final_df[final_df['Has Online delivery'] == "Yes"].Country.value_counts()
```

Out[147…
```
Country
India    2423
UAE        28
Name: count, dtype: int64
```

In [151…
```
final_df[['Has Online delivery','Country']].groupby(['Has Online delivery','Cc
```

Out[151…

	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
11	No	Turkey	34
12	No	UAE	32
13	No	United Kingdom	80
14	No	United States	434
15	Yes	India	2423
16	Yes	UAE	28

Observation

1.Online deliveries are available in Indian and UAE

In [154…
```
##  Create a Pie Chart for top 5 Cities Distribution
final_df.City.value_counts().index
```

Out[154…
```
Index(['New Delhi', 'Gurgaon', 'Noida', 'Faridabad', 'Ghaziabad',
       'Bhubaneshwar', 'Amritsar', 'Ahmedabad', 'Lucknow', 'Guwahati',
       ...
       'Ojo Caliente', 'Montville', 'Monroe', 'Miller', 'Middleton Beach',
       'Panchkula', 'Mc Millan', 'Mayfield', 'Macedon', 'Vineland Station'],
      dtype='object', name='City', length=141)
```

In [162…
```
city_values = final_df.City.value_counts().values
city_labels = final_df.City.value_counts().index
```

In [166…
```
plt.pie(city_values[:5],labels=city_labels[:5],autopct='%1.2f%%')
```

Out[166… ([<matplotlib.patches.Wedge at 0x227171b3320>,
<matplotlib.patches.Wedge at 0x227171b3260>,
<matplotlib.patches.Wedge at 0x227171b3ce0>,
<matplotlib.patches.Wedge at 0x227171e0380>,
<matplotlib.patches.Wedge at 0x227171e09e0>],
[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%')])