

```
In [103.]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib inline
```

```
In [102.]: df= pd.read_csv('zomato.csv',encoding='latin-1')
df.head()
```

Out[102.]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range	Aggregate rating	Rating color	Rating text	Votes
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenue, Makati City	Century City Mall, Poblacon, Makati City	Century City Mall, Poblacon, Makati City	121.027535	14.565443	French, Japanese, Desserts	...	Botswana Pula(P)	Yes	No	No	No	3	4.8	Dark Green	Excellent	314
1	6304287	Izakaya Kikufu	162	Makati City	Little Tokyo, 2277 Chino Avenue, Legaspi Village, Makati City	Little Tokyo, Legaspi Village, Makati City	Little Tokyo, Legaspi Village, Makati City	121.014101	14.553708	Japanese	...	Botswana Pula(P)	Yes	No	No	No	3	4.5	Dark Green	Excellent	591
2	6300002	Heat-Edsa Shangri-La	162	Mandakyuyong City	Shangri-La, 1 Garden Road, 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	No	No	4	4.4	Green	Very Good	270
3	6318506	Ooma	162	Mandakyuyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.056475	14.585318	Japanese, Sushi	...	Botswana Pula(P)	No	No	No	No	4	4.9	Dark Green	Excellent	365
4	6312002	Sanbo Kojin	162	Mandakyuyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.057506	14.584450	Japanese, Korean	...	Botswana Pula(P)	Yes	No	No	No	4	4.8	Dark Green	Excellent	229

5 rows × 21 columns

```
In [103.]: 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 [104.]: df.describe()
```

Out[104.]:

	Restaurant ID	Country Code	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
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.666370
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.516378
min	5.300000e+01	1.000000	-157.848486	-11.330428	0.000000	1.000000	0.000000
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.500000
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.200000
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.700000
max	1.850065e+07	216.000000	174.832089	55.876980	800000.000000	4.000000	4.900000

## In Data Analysis What we can do

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

```
In [105.]: df.isnull().sum()
```

Out[105.]:

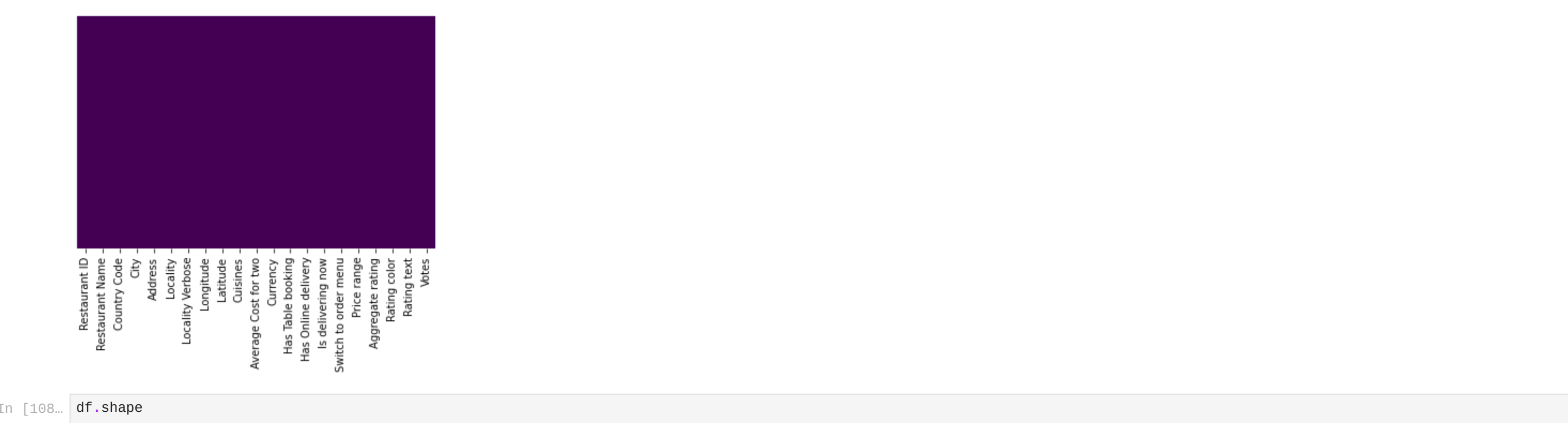
Restaurant ID	0
Restaurant Name	0
Country Code	0
City	0
Address	0
Locality	0
Locality Verbose	0
Longitude	0
Latitude	0
Cuisines	0
Average Cost for two	0
Currency	0
Has Table booking	0
Has Online delivery	0
Is delivering now	0
Switch to order menu	0
Price range	0
Aggregate rating	0
Rating color	0
Rating text	0
Votes	0
dtype: int64	

```
In [106.]: [features for features in df.columns if df[features].isnull().sum()!=0]
```

```
Out[106.]: ['Cuisines']
```

```
In [107.]: sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
Out[107.]: <AxesSubplot:~>
```



```
In [108.]: df.shape
```

```
Out[108.]: (9551, 21)
```

```
In [109.]: df.country=pd.read_excel('Country-code.xlsx')
df.country.head(10)
```

Out[109.]:

Country Code	Country
0	1 India
1	14 Australia
2	30 Brazil
3	37 Canada
4	94 Indonesia
5	148 New Zealand
6	162 Philippines
7	166 Qatar
8	184 Singapore
9	189 South Africa

```
In [110.]: final_df=pd.merge(df,df.country,one='Country Code',how='left')
final_df.head(3)
```

	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	Aggregate rating	Rating color	Rating text	Votes	Country
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenue, Makati City	Century City Mall, Poblacon, Makati City	Century City Mall, Poblacon, Makati City	121.027535	14.565443	French, Japanese, Desserts	Yes	No	No	No	3	4.8	Dark Green	Excellent	314	Philippines
1	6304287	Izakaya Kikufu	162	Makati City	Little Tokyo, 2277 Chino Road, Legaspi Village, Makati City	Little Tokyo, Legaspi Village, Makati City	Little Tokyo, Legaspi Village, Makati City	121.014101	14.553708	Japanese	Yes	No	No	No	3	4.5	Dark Green	Excellent	591	Philippines
2	6300002	Heat-Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Road, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...	121.056831	14.581404	Seafood, Asian, Filipino, Indian	Yes	No	No	No	4	4.4	Green	Very Good	270	Philippines
3 rows x 22 columns																				

```
In [111]: df_country.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
```

3 rows × 22 columns

```
In [111.]: df.country.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 2 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Country Code        15 non-null     int64
1   Country             15 non-null     object
dtypes: int64(1), object(1)
memory usage: 368.8+ bytes
```

```
In [112.]: ## to check data type
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 [113.]: country_name=final_df.Country.value_counts().index
```

```
In [114.]: country_values=final_df.Country.value_counts().values
```

```
In [115.]: ##Pie Chart of 5 top countries that use zomato
plt.pie(country_values[:3],labels=country_name[:3],autopct='%1.2f%%')
```



```
In [116.]: final_df.columns
```

```
Out[116.]: 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 [117.]: ratings=final_df.groupby(['Aggregate rating','Rating color','Rating text']).size().reset_index().rename(columns={'0':'Rating count'})
```

```
In [118.]: ratings
```

Out[118.]:

	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	490
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

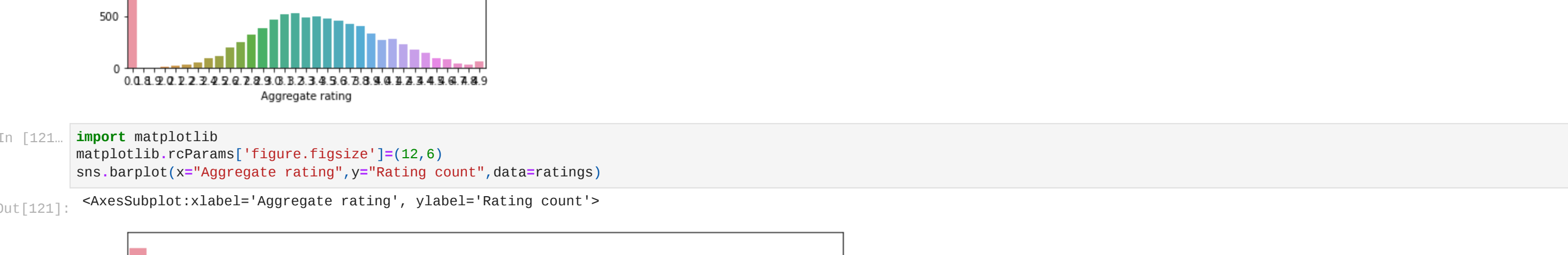
```
In [119.]: ratings.head()
```

Out[119.]:

	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 [120.]: sns.barplot(x='Aggregate rating',y='Rating count',data=ratings)
```

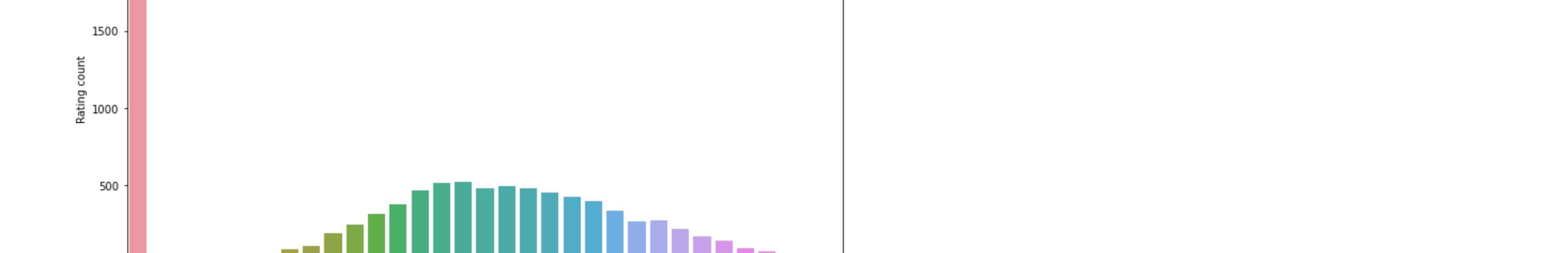
```
Out[120.]: <AxesSubplot:~>
```



```
In [121.]: import matplotlib
matplotlib.rcParams['figure.figsize']=(12,6)
```

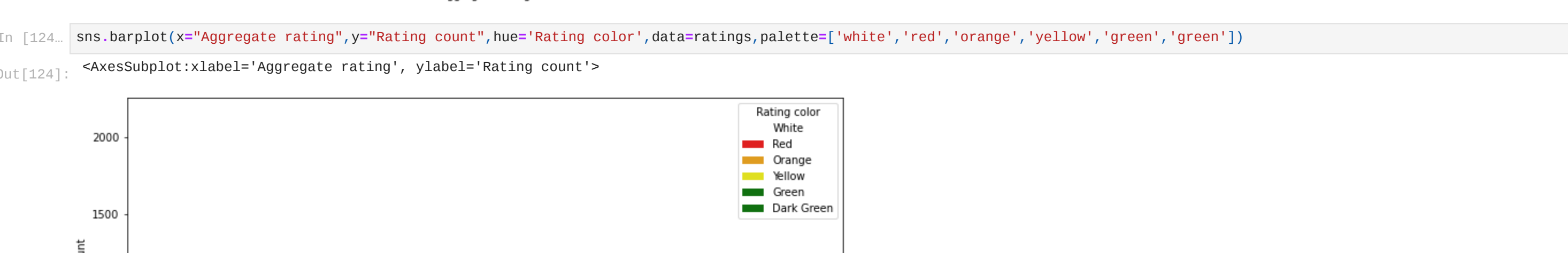
```
sns.barplot(x='Aggregate rating',y='Rating count',data=ratings)
```

```
Out[121.]: <AxesSubplot:~>
```



```
In [124.]: sns.barplot(x='Aggregate rating',y='Rating count',hue='Rating color',data=ratings,palette=['white','red','orange','yellow','green','green'])
```

```
Out[124.]: <AxesSubplot:~>
```



```
In [127.]: final_df[final_df['Rating color']=='white'].groupby('Country').size().reset_index()
```

```
##find out the country name who have most 0 rating
```

Out[127.]:

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

```
In [140.]: ##find out what country use what currency
final_df[['Country','Currency']].groupby(['Country','Currency']).size().reset_index()
```

```

In [140]: df.groupby(['Country', 'Currency']).size().reset_index()
Out[140]:

```

	Country	Currency	0
0	Australia	Dollar(\$)	24

```
In [137.]: ## find out the relation between country and Currency
final_df.groupby(['Country','Aggregate rating']).size().reset_index()
```

Out[137.]:

Country	Aggregate rating	0
0	Australia	2.4 1
1	Australia	2.8 1
2	Australia	3.2 1
3	Australia	3.4 1
...	...	...
217	United States	4.5 24
218	United States	4.6 18
219	United States	4.7 9
220	United States	4.8 3
221	United States	4.9 14

222 rows × 3 columns

```
In [148.]: ## find out the country who has online delivery option
final_df[final_df['Has online delivery']=='Yes'].groupby('Country').size().reset_index()
```

Out[148.]:

Country	0
0	India 2423
1	UAE 28

```
In [149.]: final_df.groupby(['Has Online delivery','Country']).size().reset_index()
```

Out[149.]:

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	Philippines 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

```
In [150.]: city_index=final_df.City.value_counts().index
city_value=final_df.City.value_counts().values
```

```
Out[150.]: plt.pie(city_value[:5], labels=city_index[:5], autopct='%1.2f%%')
```

