Open in Colab

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

%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

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

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

Out[ ]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency
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)
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)
					Edsa Shangri-	Edsa Shangri-	Edsa Shangri-			Seafood		

2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	La, 1 Garden Way, Ortigas, Mandal...	La, Ortigas, Mandaluyong City	La, Ortigas, Mandaluyong City, Ma...	121.056831	14.581404	Asian, Filipino, Indian	...	Botswana Pula(P)
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)
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)

5 rows × 21 columns

In [ ]:
```
df.sample(5)
```

Out[ ]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Cur
6992	7603	Stuti Restaurant	1	New Delhi	6, Satyaniketan, New Delhi	Satyaniketan	Satyaniketan, New Delhi	77.168962	28.587512	North Indian	...	Rupe
4902	8517	Apni Rasoi	1	New Delhi	A2, Ganesh Nagar, Laxmi Nagar, New Delhi	Laxmi Nagar	Laxmi Nagar, New Delhi	77.277690	28.630719	North Indian	...	Rupe

141	17294850	Thai Kitchen	216	Augusta	4357 Washington Road, Evans, GA 30809	Evans	Evans, Augusta	-82.132800	33.540600	Thai	...	Dc
743	58882	Big Brewsky	1	Bangalore	Behind MK Retail, Before WIPRO Corporate Offic...	Sarjapur Road	Sarjapur Road, Bangalore	77.683237	12.913041	Finger Food, North Indian, Italian, Continenta...	...	Rupe
3581	18381663	Khan Chacha	1	New Delhi	Lower Ground Floor, FCŒ_8, Epicuria, Nehru Pla...	Epicuria Food Mall, Nehru Place	Epicuria Food Mall, Nehru Place, New Delhi	77.251426	28.551456	North Indian, Mughlai	...	Rupe

5 rows × 21 columns

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

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

In [ ]:
```
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
```

```
 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
```

Observation

1. The Dataset contains some null value in the cuisines coloumn.
2. There are total 13 categorical varibales in the datasets .

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

Out[ ]:

	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

. Have a look on Avg,Min,Max values of Prices , Rating and votes

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

Out[ ]:
```
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 [ ]:
```
#Remove the duplicates
df.duplicated().sum()
```

Out[ ]:
```
0
```

In [ ]:
```
df.dropna(inplace=True)
```

In [ ]:
```
# Correlation between required features
df[["Average Cost for two","Price range","Aggregate rating","Votes"]].corr()
```

Out[ ]:

	Average Cost for two	Price range	Aggregate rating	Votes
Average Cost for two	1.000000	0.075111	0.051864	0.067833

Price range	0.075111	1.000000	0.438356	0.309474
Aggregate rating	0.051864	0.438356	1.000000	0.313474
Votes	0.067833	0.309474	0.313474	1.000000

In [ ]:
```
df.iloc[0: 1]
```

Out[ ]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking	de
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	

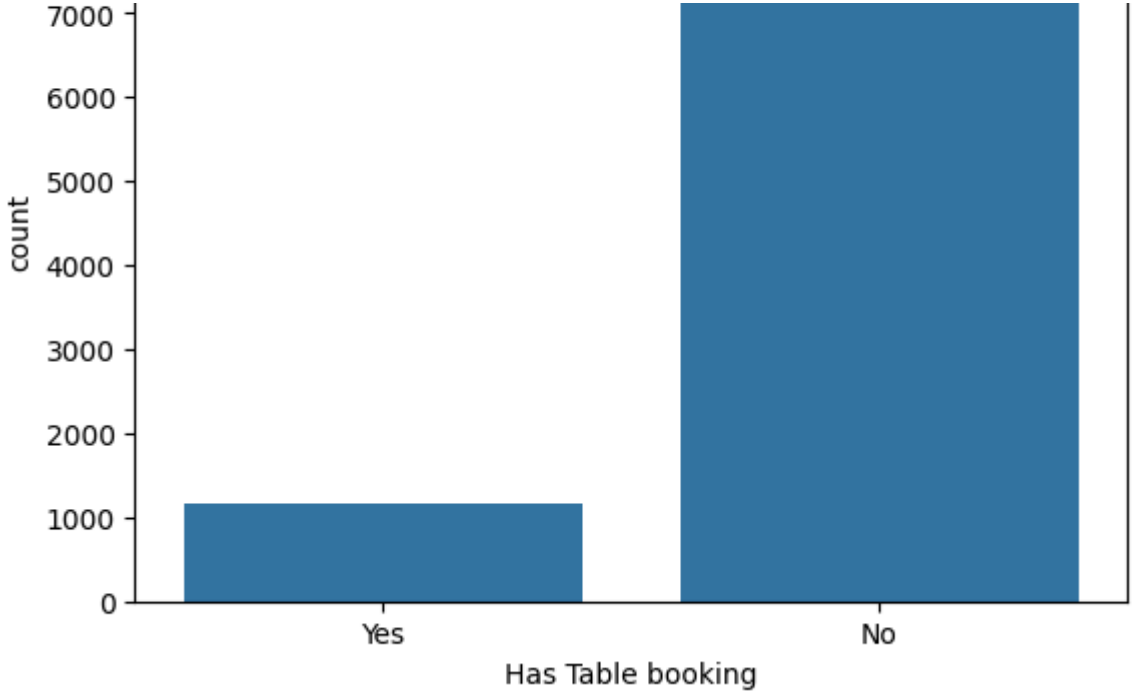
1 rows × 21 columns

```
Warning: Total number of columns (21) exceeds max_columns (20) limiting to first (20) columns.
Warning: Total number of columns (21) exceeds max_columns (20) limiting to first (20) columns.
```

Observation ,countplot for Table booking and not ,or customer online Delivery and not

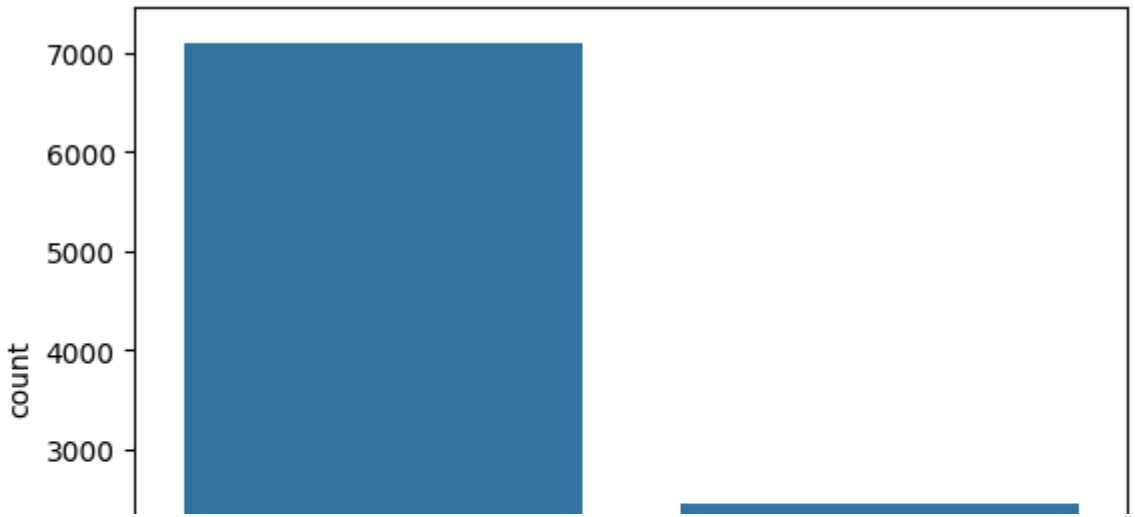
In [ ]:
```
# Analysing costumer has table bookings or not
sns.countplot(x='Has Table booking', data=df)
```

Out[ ]:
```
<Axes: xlabel='Has Table booking', ylabel='count'>
```



In [ ]:
```
# Analysing the customer  has online delivery and not
sns.countplot(x='Has Online delivery', data=df)
```

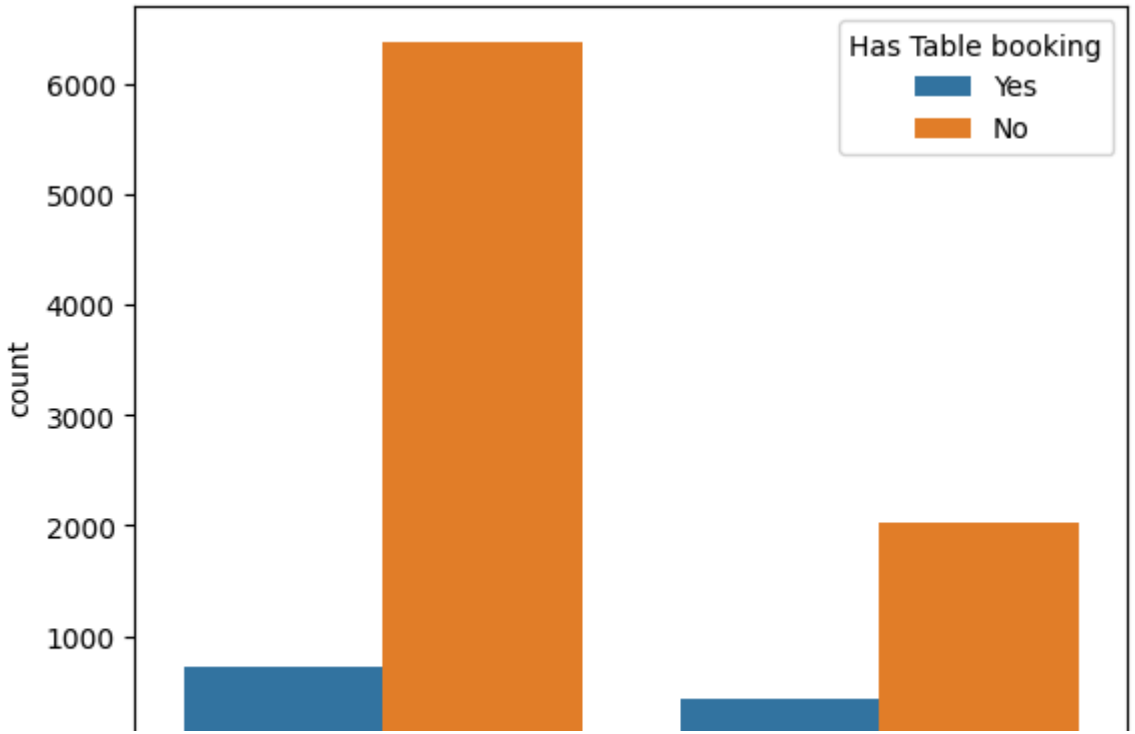
Out[ ]:
```
<Axes: xlabel='Has Online delivery', ylabel='count'>
```



Some of the customer has online delivery

In [ ]:
```
#Analysing the no of delivery in present

sns.countplot(x='Has Online delivery', hue='Has Table booking', data=df)
```

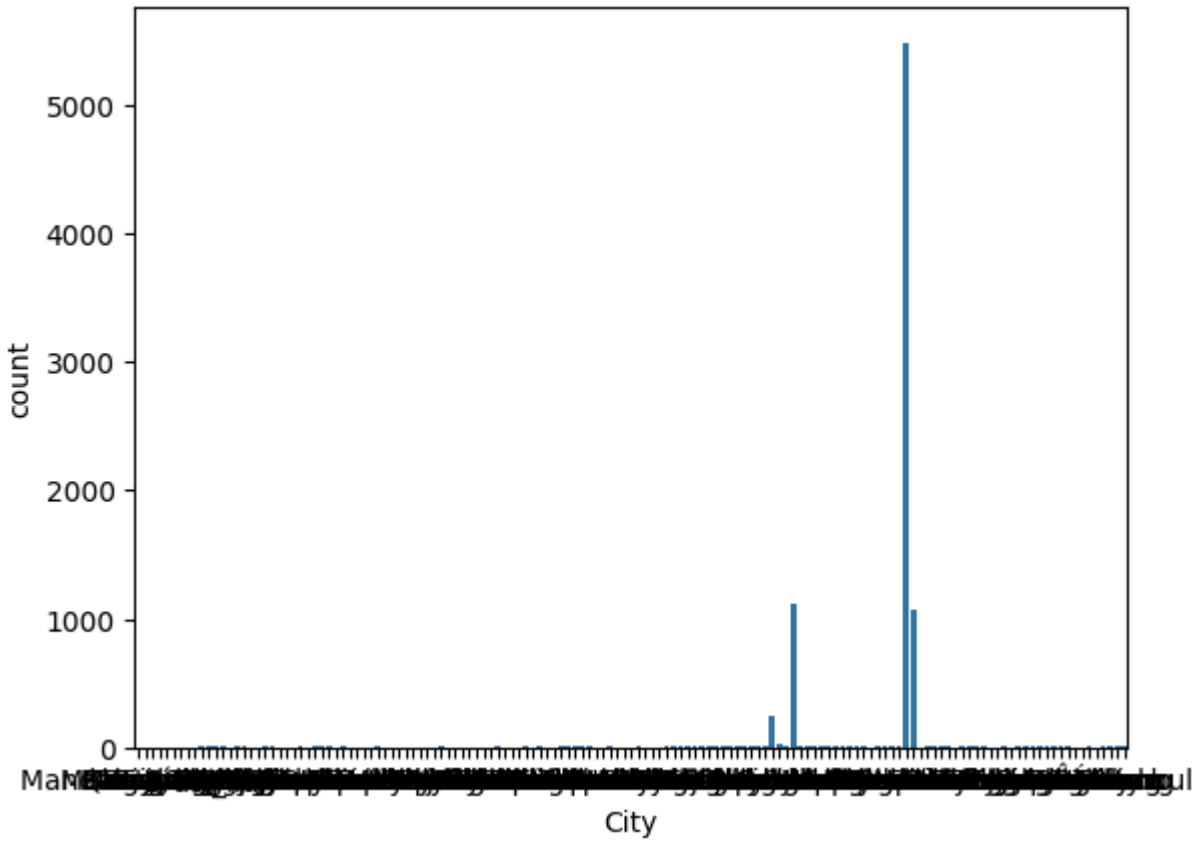
Out[ ]: <Axes: xlabel='Has Online delivery', ylabel='count'>



Observation : Very less deliveries

In [ ]:
```
sns.countplot(x='City', data=df)
```

Out[ ]: <Axes: xlabel='City', ylabel='count'>



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

In [ ]:
```
df_country
```

Out[ ]:

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

In [ ]:
```
final_df
```

Out[ ]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	b
					Third Floor,	Century City	Century City					

0	6317637	Le Petit Souffle	162	Makati City	Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Mall, Poblacion, Makati City, Mak...	121.027535	14.565443	French, Japanese, Desserts	...
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	...
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	...
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	...
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	...
...	...	...	...	...	...	...	...	...	...	...	...
9546	5915730	NamlÛ± Gurme	208	ÛÁstanbul	Kemanke□ô Karamustafa Pa□ôa Mahallesi, RÛ±htÛ±...	Karakí_y	Karakí_y, ÛÁstanbul	28.977392	41.022793	Turkish	...
9547	5908749	Ceviz AÛôacÛ±	208	ÛÁstanbul	Ko□ôuyolu Mahallesi, Muhittin íistí_ndaÛô Cadd...	Ko□ôuyolu	Ko□ôuyolu, ÛÁstanbul	29.041297	41.009847	World Cuisine, Patisserie, Cafe	...

9548	5915807	Huqqa	208	ÛÁstanbul	Kuruí_e□ôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_e□ôme	Kuruí_e□ôme, ÛÁstanbul	29.034640	41.055817	Italian, World Cuisine	...
9549	5916112	A□ô□ôk Kahve	208	ÛÁstanbul	Kuruí_e□ôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_e□ôme	Kuruí_e□ôme, ÛÁstanbul	29.036019	41.057979	Restaurant Cafe	...
9550	5927402	Walter's Coffee Roastery	208	ÛÁstanbul	CafeaÛôa Mahallesi, BademaltÛ± Sokak, No 21/B,...	Moda	Moda, ÛÁstanbul	29.026016	40.984776	Cafe	...

9551 rows × 22 columns

Display the datatype

In [ ]:
```
final_df.dtypes
```

Out[ ]:
```
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
```

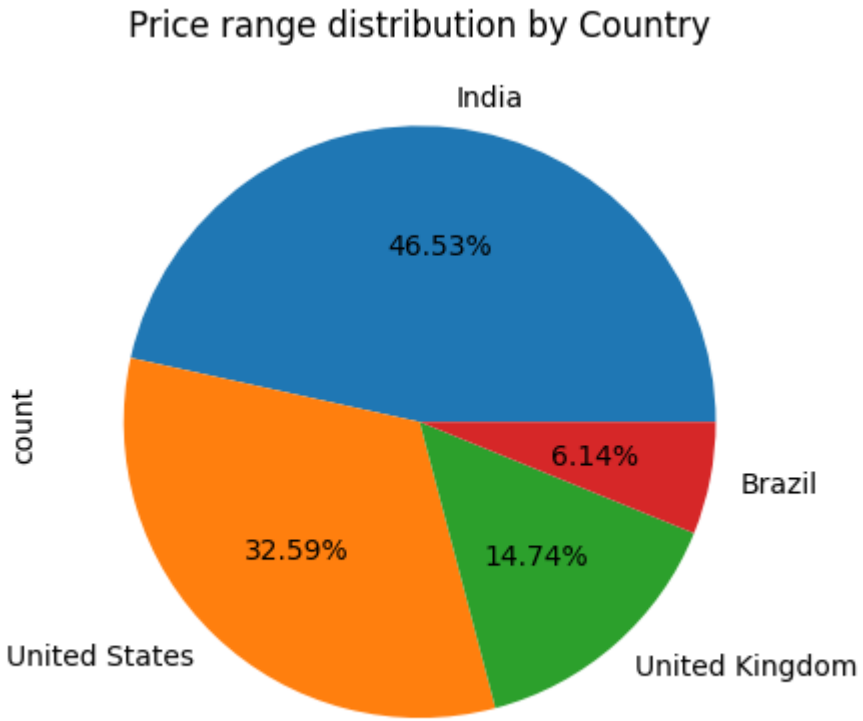
Top 3 Country with Maximum number of order

In [ ]:
```
final_df.Country.value_counts()
```

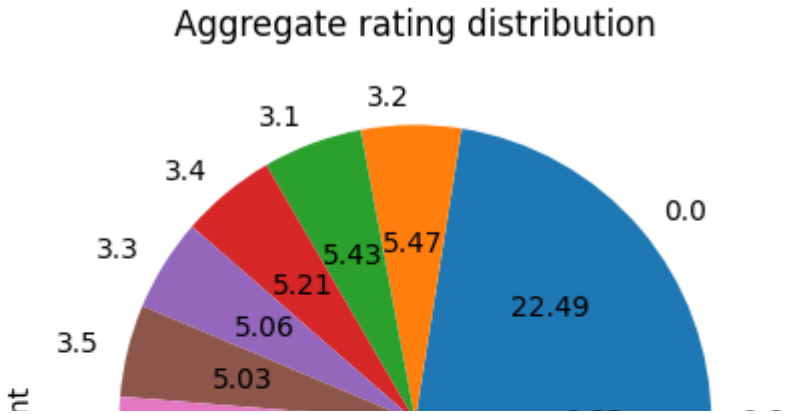
Out[ ]:
```
Country
India             8652
United States      434
United Kingdom      80
Brazil              60
UAE                 60
South Africa        60
New Zealand         40
Turkey              34
Australia           24
Phillipines         22
Indonesia           21
Singapore           20
Qatar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

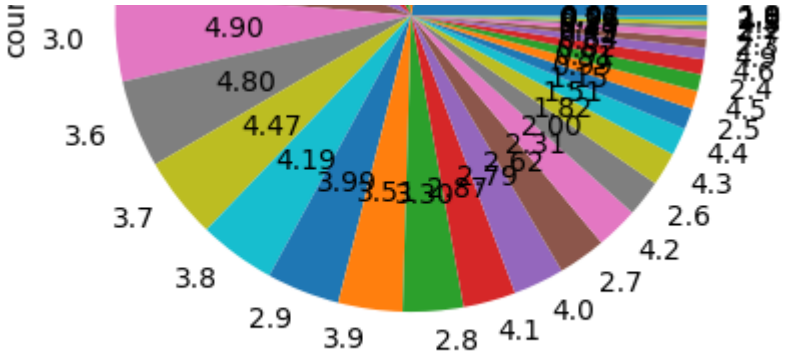
In [ ]:
```
#final_df['Price range'].value_counts().plot(kind='pie', autopct='%.2f%%')

# Assuming final_df is your DataFrame containing 'Price range' and 'Country' columns

final_df['Price range'].value_counts().plot(kind='pie', autopct='%.2f%%', labels=final_df.Country.value_counts().index)
plt.title('Price range distribution by Country')
```

Out[ ]:
```
Text(0.5, 1.0, 'Price range distribution by Country')
```



In [ ]:
```
final_df['Aggregate rating'].value_counts().plot(kind='pie',autopct='%.2f')
plt.title('Aggregate rating distribution')
plt.show()
```





Note that 22.51% costumers given 0 ratings

Display the Ratings related column in the datafram format

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

Out[ ]:

	Aggregate rating	Rating color	Rating text	0
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

INFERENCE

INFERENCE

Rating color is WHITE when it is NOT RATED Rating color is RED when it is POOR Rating color is ORANGE when it is AVERAGE Rating color is YELLOW when it is GOOD Rating color is GREEN when it is VERY GOOD Rating color is DARK GREEN when it is EXCELLENT

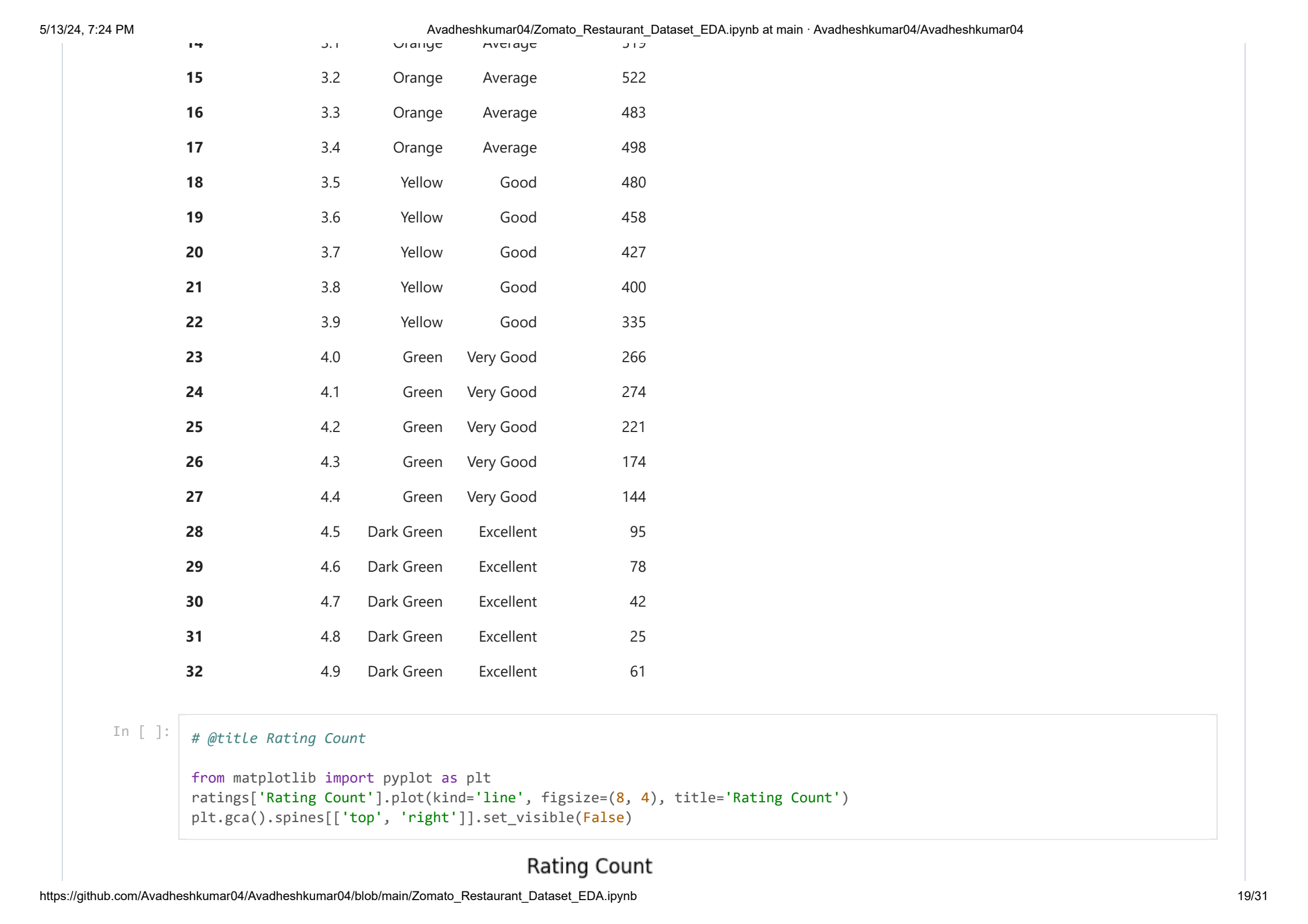
In [ ]:
```
#ratings = final_df.groupby(['Aggregate rating','Rating color','Rating text']).size().reset_index().rename(columns={0:'Rati
#ratings


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

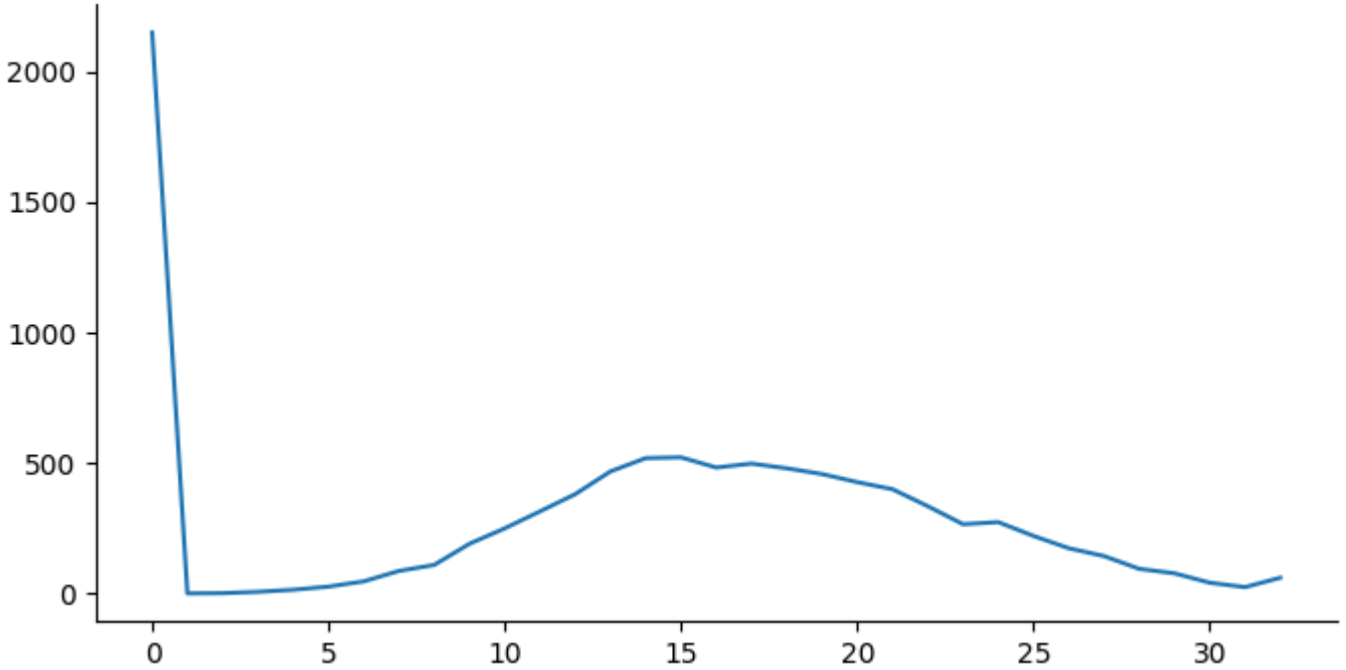
Out[ ]:

	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

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 [ ]:
```
# @title Rating Count

from matplotlib import pyplot as plt
ratings['Rating Count'].plot(kind='line', figsize=(8, 4), title='Rating Count')
plt.gca().spines[['top', 'right']].set_visible(False)
```





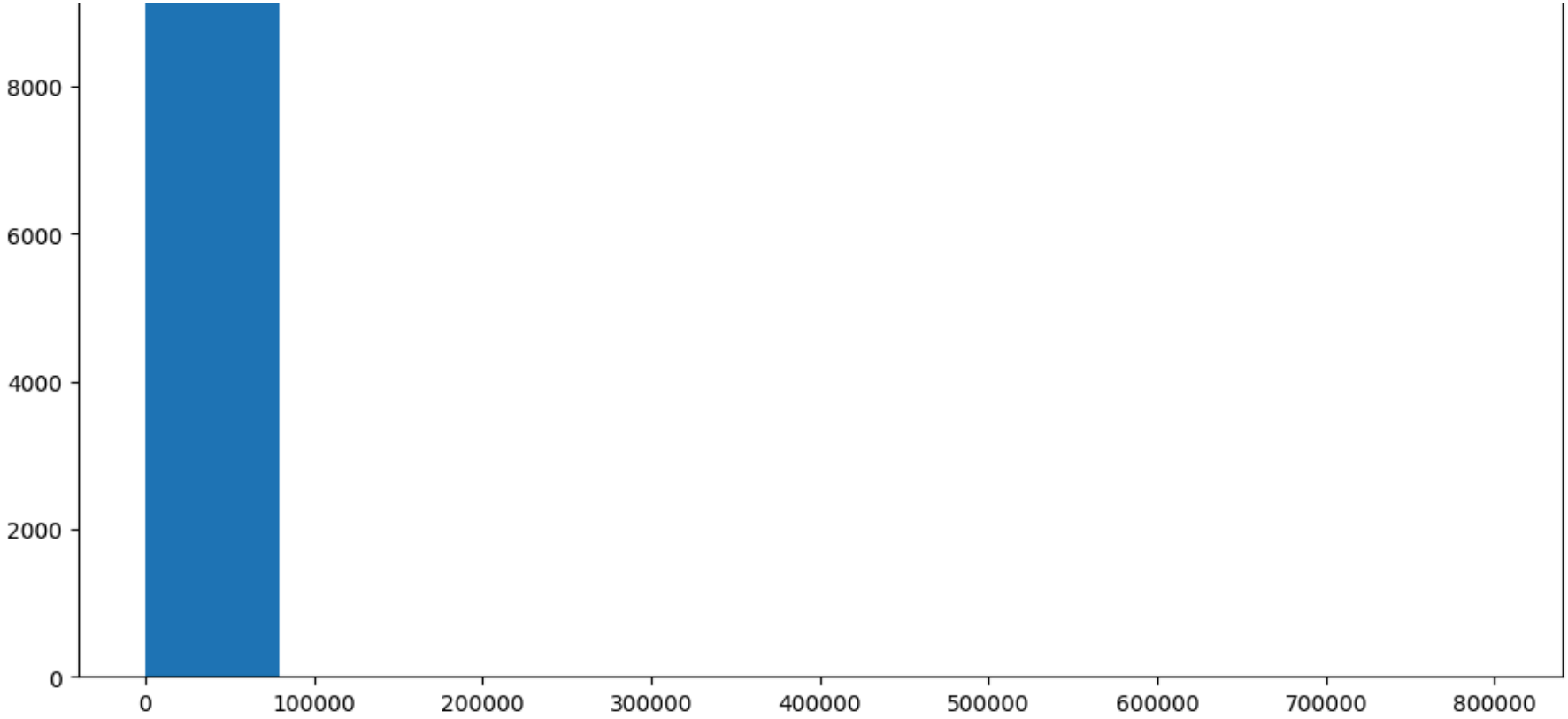
INFERENCE--> Around 2k people have given zero ratings. ratings (4.5-4.9)=excellent. 4.0 to 4.4=v.good. 3.5 to 3.9=good. 2.5 to 3.4 = average and so on .

In [ ]:

Histogram

In [ ]:
```
plt.hist(final_df['Average Cost for two'])
```

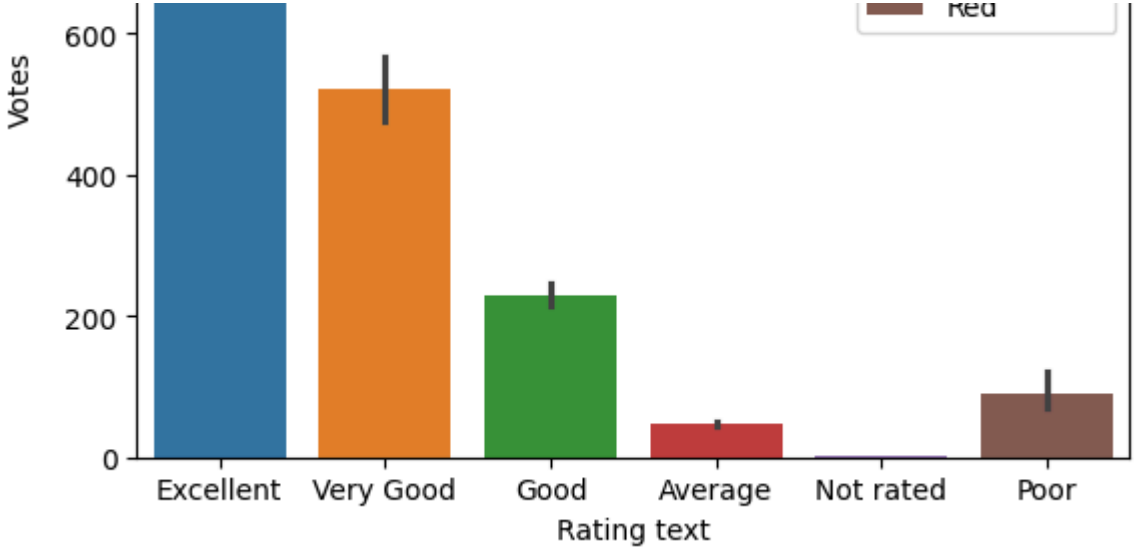
Out[ ]:
```
(array([9.531e+03, 4.000e+00, 7.000e+00, 4.000e+00, 1.000e+00, 1.000e+00,
        1.000e+00, 0.000e+00, 0.000e+00, 2.000e+00]),
 array([     0.,  80000., 160000., 240000., 320000., 400000., 480000.,
        560000., 640000., 720000., 800000.]),
 <BarContainer object of 10 artists>)
```



Bar Plot

In [ ]:
```
sns.barplot(x=final_df["Rating text"],y=df["Votes"],hue =df["Rating color"])
```

Out[ ]:
```
<Axes: xlabel='Rating text', ylabel='Votes'>
```



In [ ]:
```
hue =final_df["Rating color"]
hue
```

Out[ ]:
```
0       Dark Green
1       Dark Green
2            Green
3       Dark Green
4       Dark Green
           ...
9546         Green
9547         Green
9548        Yellow
9549         Green
9550         Green
Name: Rating color, Length: 9551, dtype: object
```

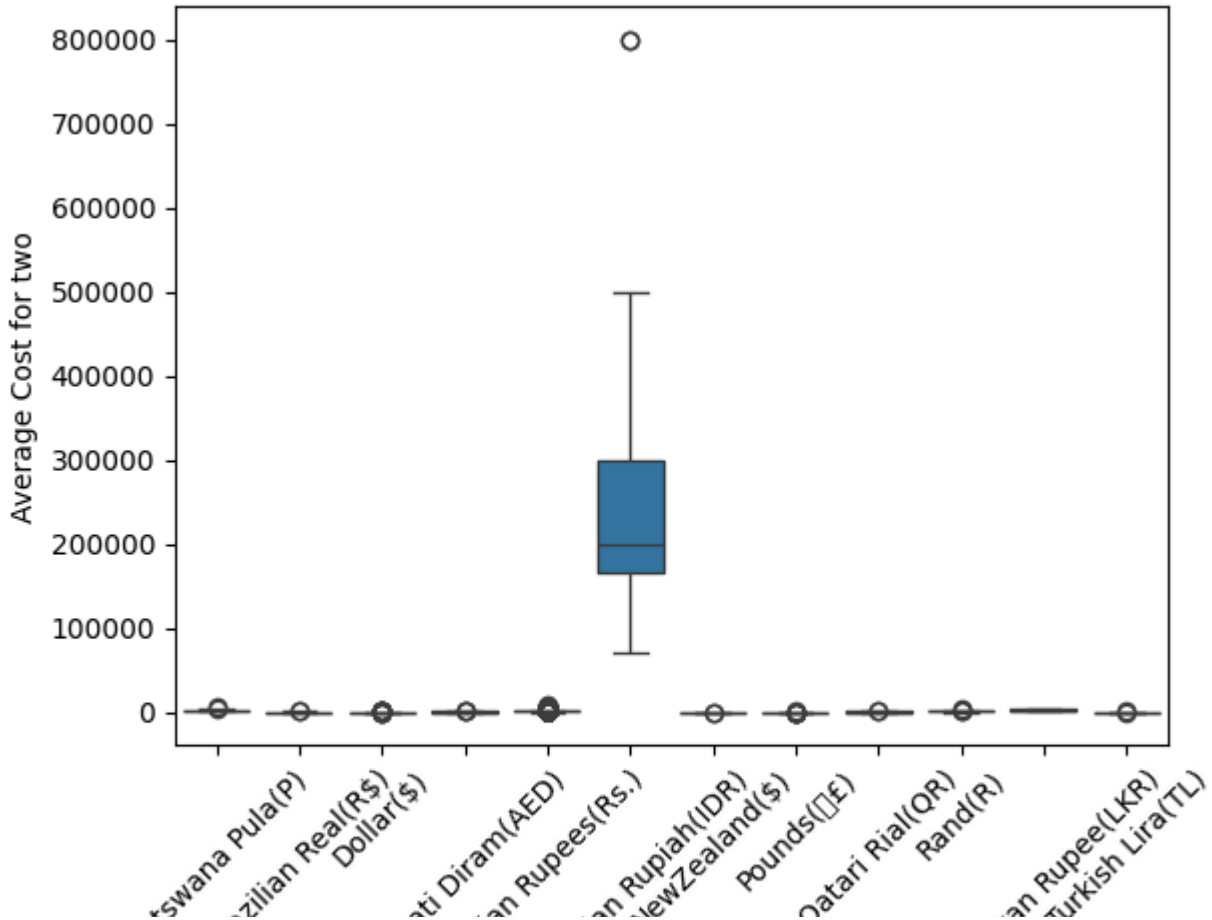
New Section

box plot

In [ ]:
```
sns.boxplot(x=final_df['Currency'], y=final_df['Average Cost for two'])
```

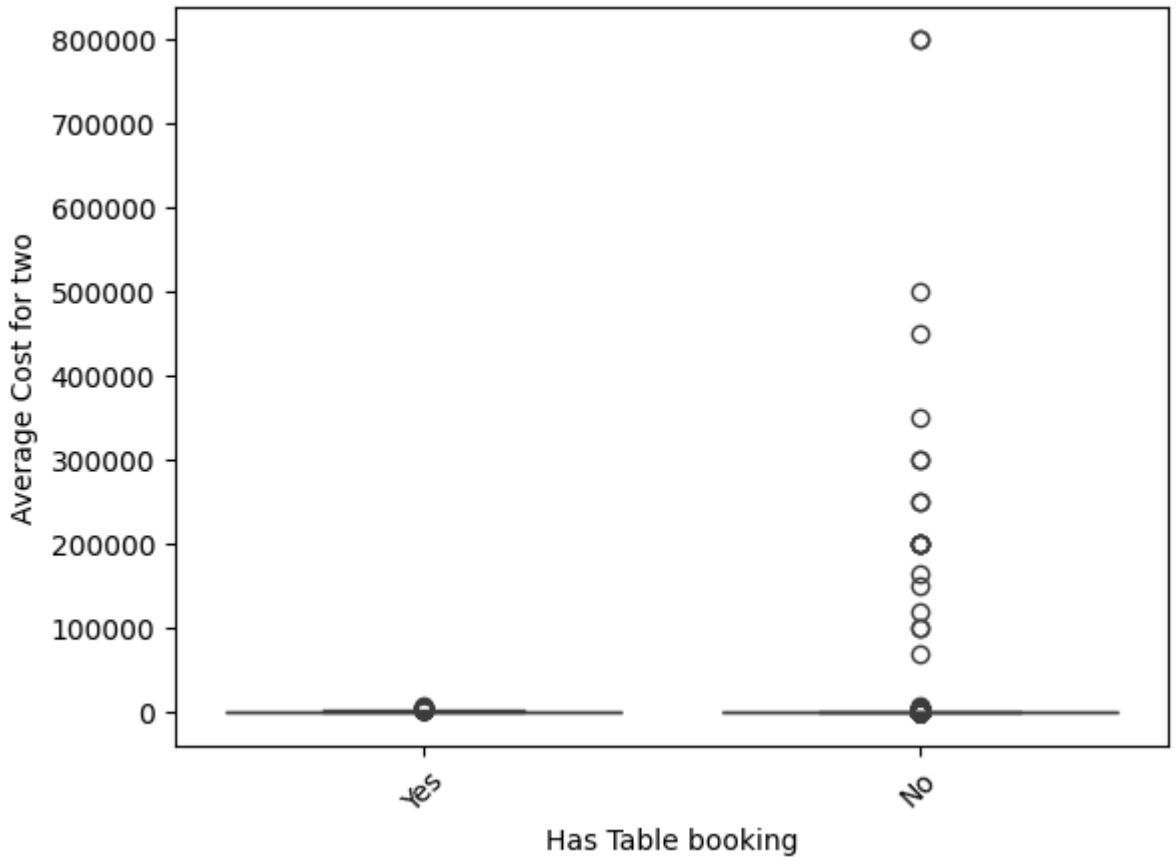
```
plt.xticks(rotation=45)
```

Out[ ]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11],
 [Text(0, 0, 'Botswana Pula(P)'),
  Text(1, 0, 'Brazilian Real(R$)'),
  Text(2, 0, 'Dollar($)'),
  Text(3, 0, 'Emirati Diram(AED)'),
  Text(4, 0, 'Indian Rupees(Rs.)'),
  Text(5, 0, 'Indonesian Rupiah(IDR)'),
  Text(6, 0, 'NewZealand($)'),
  Text(7, 0, 'Pounds(\x8c£)'),
  Text(8, 0, 'Qatari Rial(QR)'),
  Text(9, 0, 'Rand(R)'),
  Text(10, 0, 'Sri Lankan Rupee(LKR)'),
  Text(11, 0, 'Turkish Lira(TL)')])
```



In [ ]:
```
sns.boxplot(x=final_df['Has Table booking'], y=final_df['Average Cost for two'])
plt.xticks(rotation=45)
```

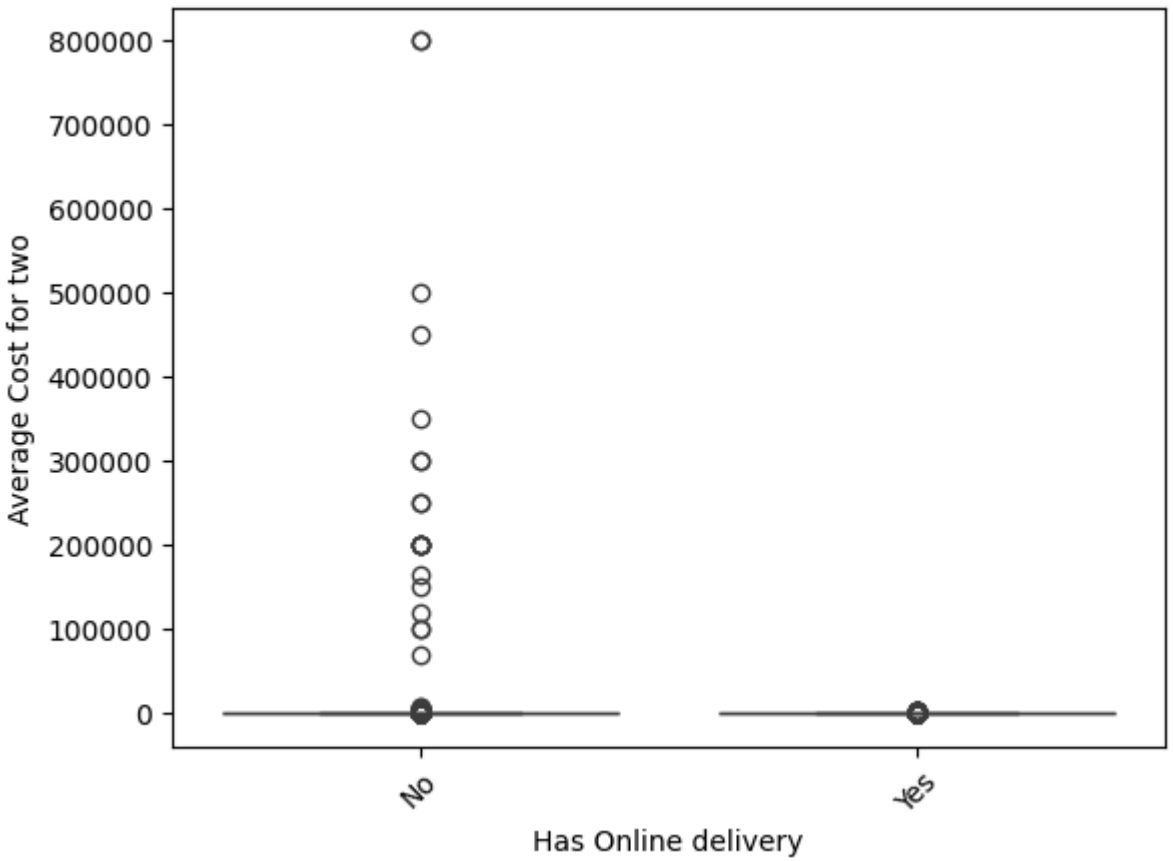
Out[ ]: ([0, 1], [Text(0, 0, 'Yes'), Text(1, 0, 'No')])



In [ ]:
```
sns.boxplot(x=final_df['Has Online delivery'], y=final_df['Average Cost for two'])
plt.xticks(rotation=45)
```

Out[ ]: ([0, 1], [Text(0, 0, 'No'), Text(1, 0, 'Yes')])

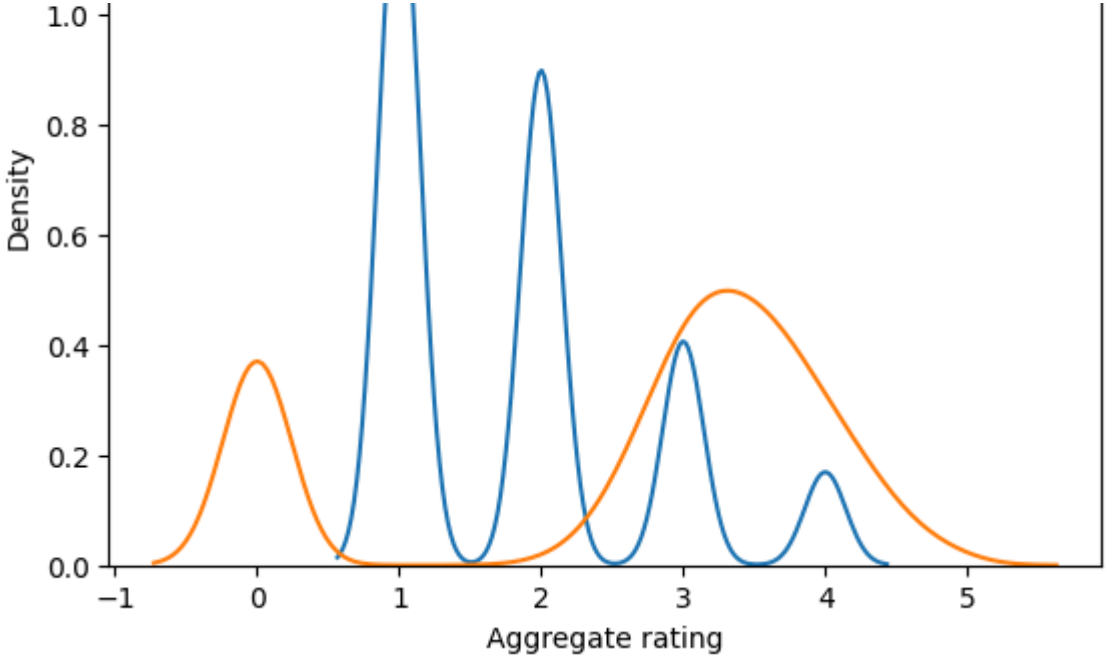
Out[ ]: ([0, 1], [Text(0, 0, 'No'), Text(1, 0, 'Yes')])



Distplot

In [ ]:
```
sns.distplot(final_df['Price range'],hist= False)
sns.distplot(final_df['Aggregate rating'],hist= False)
```

Out[ ]: <Axes: xlabel='Aggregate rating', ylabel='Density'>



Observation :-> Amazing informatios by this Probability Density Function Price range "1" has given no ratings Price range "3 and 4" has given ratings between "2.0 to 4.9

In [ ]:
```
pd.crosstab(final_df['Is delivering now'],final_df['Has Online delivery'])
```

Out[ ]:

Has Online delivery	No	Yes
Is delivering now		
No	7100	2417
Yes	0	34

HeatMap

In [ ]:
```
sns.heatmap(pd.crosstab(final_df['Is delivering now'],final_df['Has Online delivery']))
```

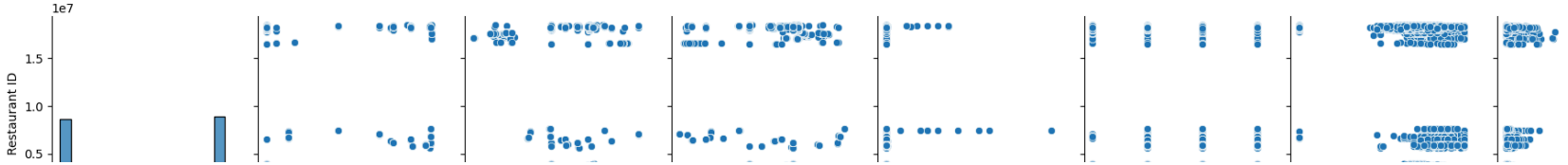
Out[ ]: <Axes: xlabel='Has Online delivery', ylabel='Is delivering now'>

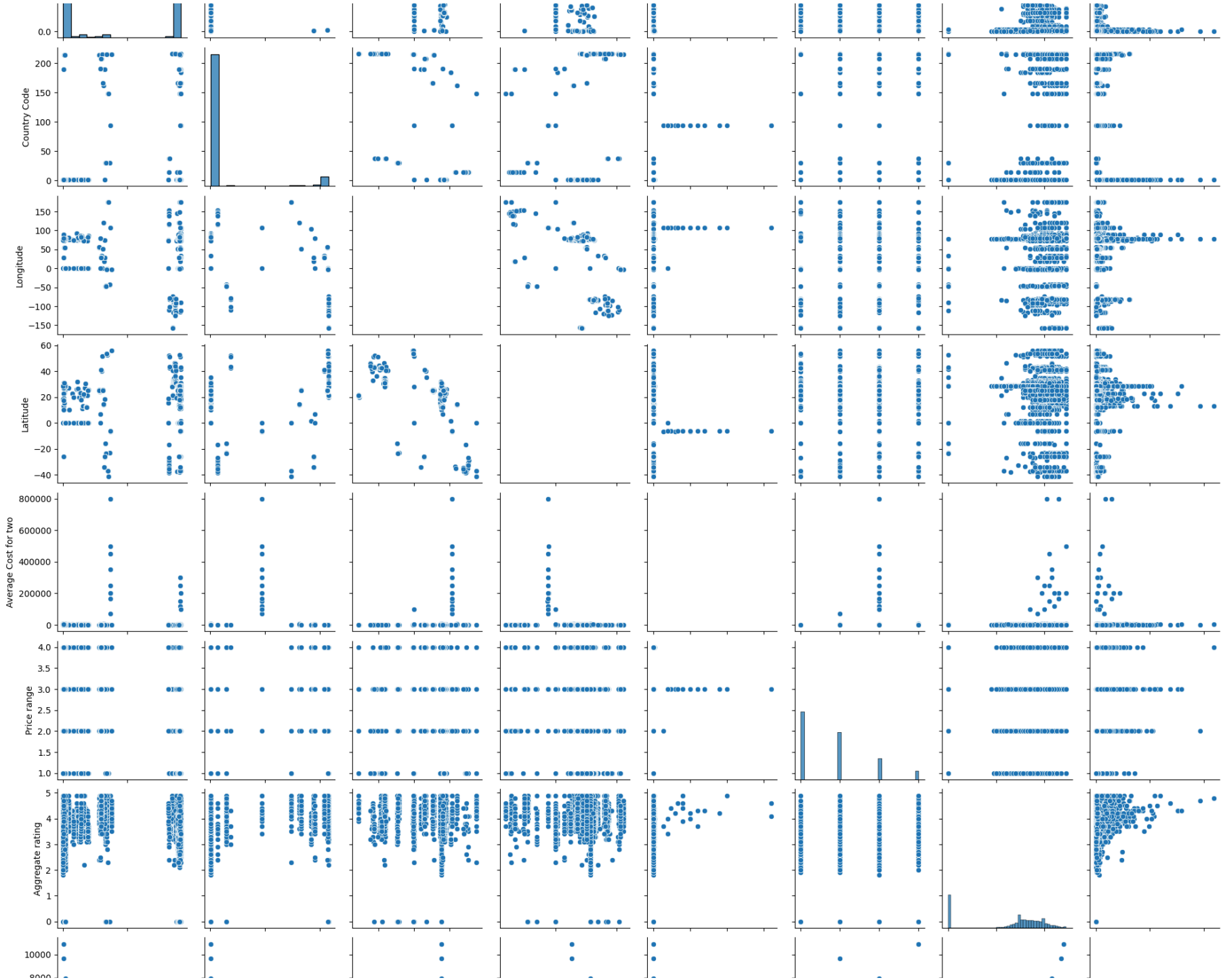
Observation:- this heatmap gives a better understanding , less no. of delivering the order recive

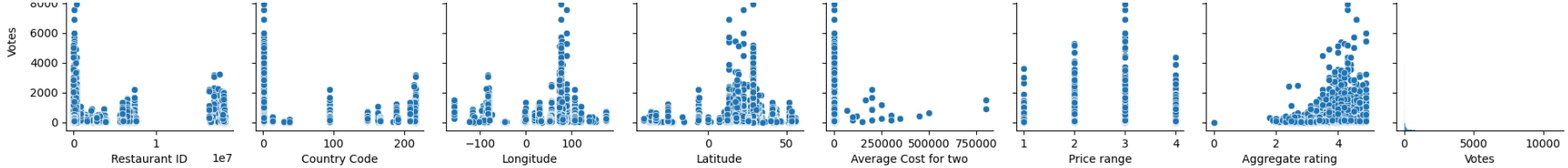
Pairplot

In [ ]:
```
sns.pairplot(final_df)
```

Out[ ]: <seaborn.axisgrid.PairGrid at 0x7db1c45c05b0>




Country Code
Longitude
Latitude
Average Cost for two
Price range
Aggregate rating

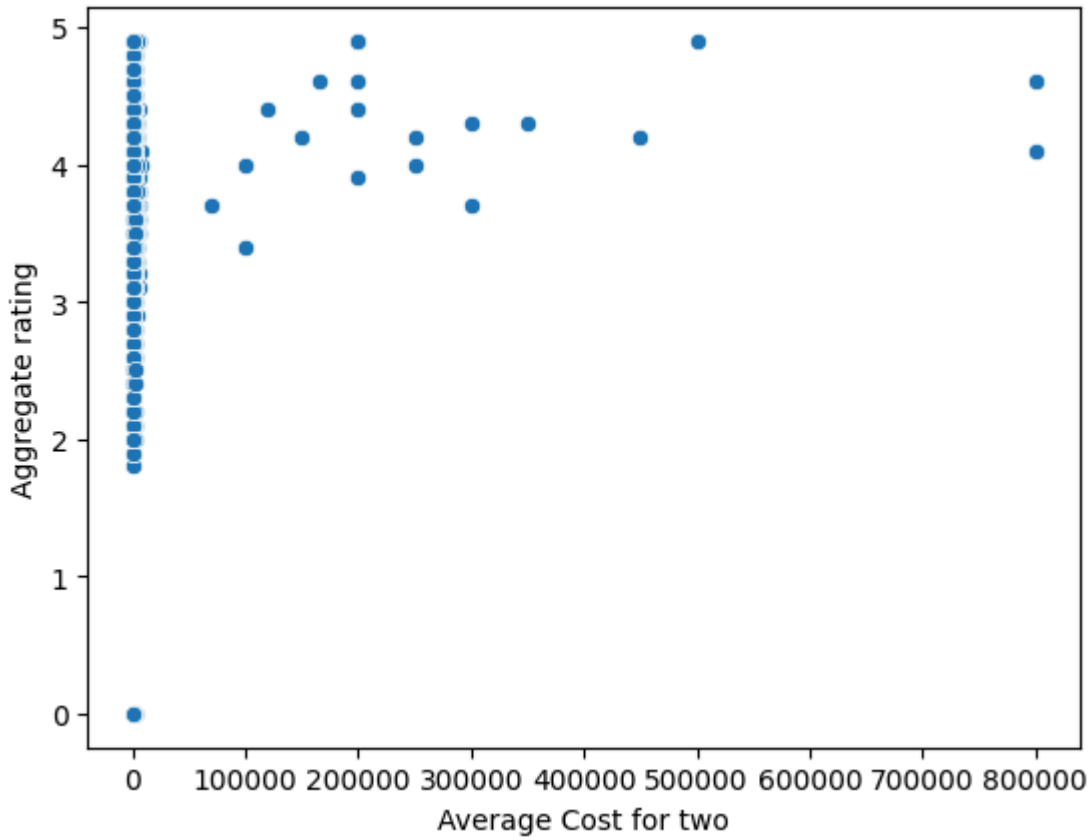


Since pairplot is not required here. But just to give a Bird Eye of Scatter plots

scatterplot

In [ ]:
```
sns.scatterplot(x=final_df['Average Cost for two'],y=final_df['Aggregate rating'])
```

Out[ ]: <Axes: xlabel='Average Cost for two', ylabel='Aggregate rating'>



In [ ]:
```
sns.lineplot(y=final_df['Votes'],x=final_df['Has Online delivery'])
```

Out[ ]: <Axes: xlabel='Has Online delivery', ylabel='Votes'>

In this notebook, we've delved into Exploratory Data Analysis (EDA), aiming to thoroughly understand our dataset's different features, whether numerical or categorical. To achieve this, we've employed a variety of visualization techniques like count plots, histograms, box plots, and scatter plots, among others.

EDA serves as a vital initial step in comprehending the inherent patterns and connections within our data. Through visualization, we're able to glean significant insights, facilitating informed decision-making. It enables us to pinpoint trends, anomalies, correlations, and other critical aspects of our dataset, laying the groundwork for further analysis and interpretation.

In [ ]: