

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

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

| Out[2]: | 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 |
|---------|---------------|------------------------|--------------|------------------|--|--|---|-------------|-----------|----------------------------------|-----|------------------|-------------------|---------------------|-------------------|----------------------|
| 0 | 6317637 | Le Petit Souffle | 162 | Makati City | Third Floor, Century City Mall, Kalyayan Avenue... | Century City Mall, Poblacion, Makati City | Century City Mall, Poblacion, Makati City... | 121.027535 | 14.565443 | French, Japanese, Desserts | ... | Botswana Pula(P) | Yes | No | No | No |
| 1 | 6304287 | Itakaya Ikujay | 162 | Makati City | Little Tokyo, Chino Roces Avenue, Ortigas, Legaspi... | Little Tokyo, Legaspi Village, Makati City | Little Tokyo, Legaspi Village, Makati City... | 121.014101 | 14.553708 | Japanese | ... | Botswana Pula(P) | Yes | No | No | No |
| 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... | 121.056831 | 14.581404 | Seafood, Asian, Filipino, Indian | ... | Botswana Pula(P) | Yes | No | No | No |
| 3 | 6318506 | Ooma | 162 | Mandaluyong City | Third Floor, Mega Fashion Hub, SM Megamall, Ortigas... | SM Megamall, Ortigas, Mandaluyong City | SM Megamall, Ortigas, Mandaluyong City... | 121.056475 | 14.585318 | Japanese, Sushi | ... | Botswana Pula(P) | No | No | No | No |
| 4 | 6314302 | Sambo Kojin | 162 | Mandaluyong City | Third Floor, Mega Fashion Hub, SM Megamall, Ortigas... | SM Megamall, Ortigas, Mandaluyong City | SM Megamall, Ortigas, Mandaluyong City... | 121.0567508 | 14.581455 | Japanese, Korean | ... | Botswana Pula(P) | Yes | No | No | No |

5 rows × 16 columns

```
In [3]: df.columns
Out[3]: 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 [4]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 16 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 object
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 int64
20 Votes 9551 non-null int64
dtypes: float64(3), int64(5), object(13)
memory usage: 1.5+ MB
```

```
In [5]: df.describe()
Out[5]:
```

| | Restaurant ID | Country Code | Longitude | Latitude | Average Cost for two | Price range | Aggregate rating | Votes |
|-------|-----------------|--------------|-------------|-------------|----------------------|-------------|------------------|--------------|
| count | 9551.000000e+03 | 9551.000000 | 9551.000000 | 9551.000000 | 9551.000000 | 9551.000000 | 9551.000000 | 9551.000000 |
| mean | 9.9511200e+06 | 18.366016 | 64.126574 | 25.854381 | 1199.210763 | 1.804837 | 2.666370 | 156.909748 |
| std | 8.701521e+06 | 56.750546 | 41.467088 | 11.007935 | 16121.183073 | 0.905609 | 1.516378 | 430.169145 |
| min | 5.300000e+01 | 1.000000 | -157.948486 | -41.330428 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 3.019625e+05 | 1.000000 | 77.061343 | 26.478713 | 250.000000 | 1.000000 | 2.500000 | 5.000000 |
| 50% | 6.004089e+06 | 1.000000 | 77.191964 | 26.570469 | 400.000000 | 2.000000 | 3.200000 | 31.000000 |
| 75% | 1.835229e+07 | 1.000000 | 77.282006 | 26.642758 | 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 |

we will find

- 1.missing values
- 2.explore about the numerical variable
- 3.explore about categorical variables
- 4.finding relationship between variable

```
In [6]: df.isnull().sum()
Out[6]: Restaurant ID      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 [7]: [features for features in df.columns if df[features].isnull().sum()>0]
```

```
Out[7]: ['Cuisines']
```

```
In [8]: sns.heatmap(df.isnull(),yticklabels=False,cmap='viridis')#dont write
```

```
Out[8]: <axes.Subplot>
```



```
In [9]: df_country=pd.read_excel('Country-Code.xlsx')
df_country.head()
Out[9]:
```

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

```
In [10]: final_df=pd.merge(df,df_country,on='Country Code',how='left')
```

```
In [11]: 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 |
|---------------|-----------------|--------------|------------------|--|--|---|------------|-----------|----------------------------------|-----|-------------------|---------------------|-------------------|----------------------|
| 0 | 6317637 | 162 | Makati City | Third Floor, Century City Mall, Kalyayan Avenue... | Century City Mall, Poblacion, Makati City | Century City Mall, Poblacion, Makati City... | 121.027535 | 14.565443 | French, Japanese, Desserts | ... | Yes | No | No | N |
| 1 | 6304287 | 162 | Makati City | Little Tokyo, Chino Roces Avenue, Legaspi... | Little Tokyo, Legaspi Village, Makati City | Little Tokyo, Legaspi Village, Makati City... | 121.014101 | 14.553708 | Japanese | ... | Yes | No | No | N |
| 2 | 6300002 | 162 | Mandaluyong City | Edsa Shangri-La 1 Garden Way, Ortigas, Mandal... | Edsa Shangri-La, Ortigas, Mandaluyong City | Edsa Shangri-La, Ortigas, Mandaluyong City... | 121.056831 | 14.581404 | Seafood, Asian, Filipino, Indian | ... | Yes | No | No | N |

3 rows × 22 columns

```
In [12]: # to check datatypes
final_df.dtypes
```

```
Out[12]: Restaurant ID      int64
Restaurant Name      object
Country Code      int64
City      object
Address      object
Locality      object
Locality Verbose      object
Longitude      float64
Latitude      object
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 [13]: final_df.columns
Out[13]: 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 [14]: country_names=final_df.Country.value_counts().index
```

```
In [15]: country_val=final_df.Country.value_counts().values
```

pie chart

top 3 countries that uses zomato

```
In [16]: plt.pie(country_val[3],labels=country_names[3],autopct='%1.2F%%')
```

```
Out[16]: ([matplotlib.patches.Wedge at 0x19590b3e028],
[matplotlib.patches.Wedge at 0x19590b3e078],
[matplotlib.patches.Wedge at 0x19590b3fc69]),
[Text(1.0820742760952103, 0.19278674827830725, 'India'),
Text(1.87728173583856, -0.22486257134123397, 'United States'),
Text(1.6995865158282035, -0.03015783794312079, 'United Kingdom')],
[Text(0.590713238233761, 0.18515648015183668, '94.39%'),
Text(0.5876680286539102, -0.12111396518517871, '4.73%'),
Text(0.5997744626358918, -0.01644972978715676, '0.87%')])
```



observation:zomato maximum transaction are from India second position:usa third position:uk

```
In [17]: final_df.columns
Out[17]: 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 [18]: ratings=final_df.groupby(['Aggregate rating','Rating color','Rating text']).size().reset_index().rename(columns={0:'Rating Count'})
```

```
Out[19]:
```

| | 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 | 191 |
| 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 | 482 |
| 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 |

observation

1. when ratings are between 4.5 to 4.9 ----> excellent
2. when ratings are between 4.0 to 4.4 ----> very good
3. when ratings are between 3.5 to 3.9 ----> good
4. when ratings are between 3.0 to 3.4 ----> average
5. when ratings are between 2.5 to 2.9 ----> average
6. when ratings are between 2.0 to 2.4 ----> poor

```
In [20]: ratings.head()
Out[20]:
```

| | 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 [21]: import matplotlib
matplotlib.rcParams['figure.figsize']=(12,6)
```

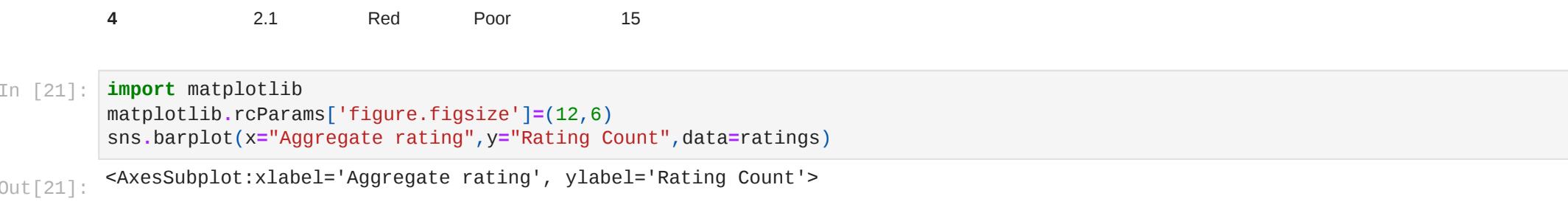
```
sns.barplot(x='Aggregate rating',y='Rating Count',data=ratings)
```

```
Out[21]: <AxesSubplot: xlabel='Aggregate rating', ylabel='Rating Count'>
```



```
In [22]: sns.barplot(x='Aggregate rating',y='Rating Count',hue='Rating color',data=ratings,palette=['blue','red','orange','yellow','green','green'])
```

```
Out[22]: <AxesSubplot: xlabel='Aggregate rating', ylabel='Rating Count'>
```

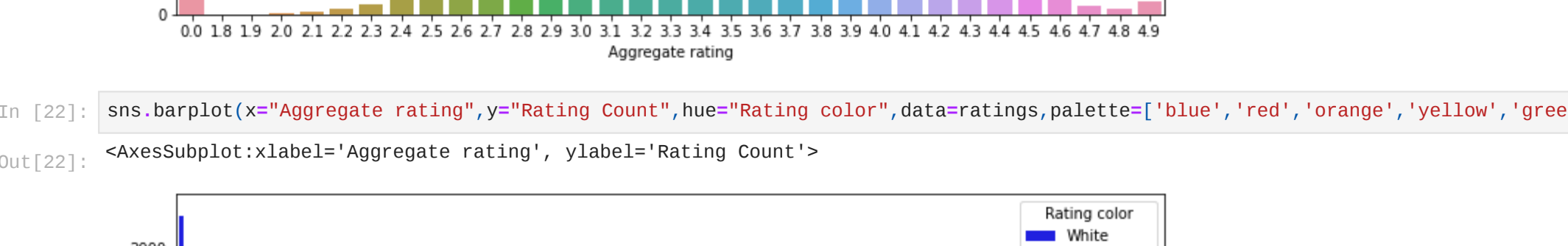


observation:

1. Not rated count is very high
2. Maximum ratings are between 2.5 to 3.4

```
In [23]: ## count plot
sns.countplot(x='Rating color',data=ratings,palette=['blue','red','orange','yellow','green','green'])
```

```
Out[23]: <AxesSubplot: xlabel='Rating color', ylabel='count'>
```



```
In [24]: ratings
Out[24]:
```

| | 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 | 191 |
| 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 | 482 |
| 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 [25]: final_df[final_df['Rating color']=='White'].groupby('Country').size().reset_index()
```

```
Out[25]:
```

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

```
In [26]: final_df.groupby(['Aggregate rating','Country']).size().reset_index().head(5) # alternatively
```

```
Out[26]:
```

| | 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 |

observation:

1. Maximum numbers of zero ratings are from Indian Customers

```
In [30]: # find out which currency is used by which country?
```

```
In [31]: final_df.columns
Out[31]: 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 [32]: final_df[['Country','Currency']].groupby(['Country','Currency']).size().reset_index()
```

```
Out[32]:
```

| | 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 | New Zealand(\$) | 40 |
| 6 | Philippines | 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(TRY) | 34 |
| 12 | UAE | Emirate Diram(AED) | 80 |
| 13 | United Kingdom | Pounds(£) | 60 |
| 14 | United States | Dollar(\$) | 434 |

```
In [33]: # which country do have online delivery?
```

```
In [34]: final_df[final_df['Has Online delivery']=='Yes'].Country.value_counts()
```

```
Out[34]: India      2423
UAE      28
Name: Country, dtype: int64
```

```
In [35]: final_df[['Country','Has Online delivery']].groupby(['Country','Has Online delivery']).size().reset_index()
```

```
Out[35]:
```

| | Country | Has Online delivery | 0 |
|----|----------------|---------------------|------|
| 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 | Philippines | 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 | Yes | 28 |
| 14 | United Kingdom | No | 434 |
| 15 | United States | No | 80 |

observation:

1. Online deliveries are available in India and UAE

```
In [36]: ## create a pie chart for top 5 cities distribution
```

```
Out[36]:
```

| | 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 |
|---|---------------|------------------|--------------|-------------|--|---|--|------------|-----------|----------------------------|-----|------------------|-------------------|---------------------|-------------------|----------------------|
| 0 | 6317637 | Le Petit Souffle | 162 | Makati City | Third Floor, Century City Mall, Kalyayan Avenue... | Century City Mall, Poblacion, Makati City | Century City Mall, Poblacion, Makati City... | 121.027535 | 14.565443 | French, Japanese, Desserts | ... | Botswana Pula(P) | Yes | No | No | No |
| 1 | 630428 | | | | | | | | | | | | | | | |