

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

In [5]: df = pd.read_csv(r"C:\Users\WALU\Downloads\Zomato_EDA\Zomato.csv",encoding='latin-1')

In [6]: df.head(5)

Out[6]:
  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, Kalyan Avenue... Century City Mall, Poblacion, Makati City, Mak... 121.027535 14.565443 French, Japanese, Desserts ... Botswana Pula(P) Yes No No No 3 4.8 Dark Green Excellent 314
1 6304287 Izakaya Kikufuji 162 Makati City Little Tokyo, Legaspi Village, Legaspi... Little Tokyo, Legaspi Village, Makati City, Ma... 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 Mandaluyong City Third Floor, Garden Walk, Ortigas, Mandal... 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 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) No No No No 4 4.9 Dark Green Excellent 365
4 6314302 Sambo Kojin 162 Mandaluyong City Third Floor, Mega Fashion Hall, SM Megamall, O... SM Megamall, Ortigas, Mandaluyong City SM Megamall, Ortigas, Mandaluyong City, Mandal... 121.057508 14.584450 Japanese, Korean ... Botswana Pula(P) Yes No No No 4 4.8 Dark Green Excellent 229

5 rows x 21 columns

In [7]: df.shape
Out[7]: (9551, 21)

In [12]: df.columns
Out[12]: 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 [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: 1.8+ MB

In [9]: df.describe()

Out[9]:
  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+03 18.365616 16.126574 25.854381 1199.210763 1.804837 2.666370 156.909748
std 8.791521e+06 56.750546 11.467058 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 131.000000
75% 1.855229e+07 1.000000 77.282006 28.642758 700.000000 2.000000 3.700000 31.000000
max 1.850056e+07 216.000000 174.832089 55.976980 800000.000000 4.000000 4.900000 10934.000000

In [10]: df.isnull().sum()

Out[10]:
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 [15]: df.country=pd.read_excel(r"C:\Users\WALU\Downloads\Zomato_EDA\Country-Code.xlsx")
df.country.head()

Out[15]:
  Country Code Country
0 1 India
1 14 Australia
2 30 Brazil
3 37 Canada
4 84 Indonesia

In [18]: final_df = pd.merge(df,df.country,on='Country Code',how='left')

In [21]: final_df.head(2)

Out[21]:
  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 Country
0 6317637 Le Petit Souffle 162 Makati City Third Floor, Century City Mall, Kalyan Avenue... Century City Mall, Poblacion, Makati City, Mak... 121.027535 14.565443 French, Japanese, Desserts ... Botswana Pula(P) Yes No No No 3 4.8 Dark Green Excellent 314 Philippines
1 6304287 Izakaya Kikufuji 162 Makati City Little Tokyo, Legaspi Village, Legaspi... Little Tokyo, Legaspi Village, Makati City, Ma... 121.014101 14.553708 Japanese ... Botswana Pula(P) Yes No No No 3 4.5 Dark Green Excellent 591 Philippines

2 rows x 22 columns

In [28]: country_names=final_df.Country.value_counts().index
country_val=final_df.Country.value_counts().values
# we pie chart: Top 5 countries that uses zomato
plt.pie(country_val[:5],labels=country_names[:5],autopct='%1.2f%%')

Out[28]: ([<matplotlib.patches.Wedge at 0x24488917358>,
<matplotlib.patches.Wedge at 0x244879ff499>,
<matplotlib.patches.Wedge at 0x244876ed450>],
[Text(1.0829742708952183, 0.19278674827836725, 'India'),
Text(1.077281710838956, -0.2248527134123297, 'United States'),
Text(1.0986061538228295, -0.3395178789320973, 'United Kingdom'),
Text(1.0999462800895612, -0.01087113182029924, 'Ghaziabad'),
Text(0.5076802080391032, -0.1231518653021707, '1.73%')],
[Text(0.5997744629358018, -0.81644972978715676, '0.87%')])

Observation: Zomato maximum records or transaction are from India After that USA and then United Kingdoms

In [22]: ratings=final_df.groupby(['Aggregate rating','Rating color','Rating text']).size().reset_index().rename(columns={0:'Rating Count'})
ratings

Out[22]:
  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 [23]: import matplotlib
sns.set(rcparams={'figure.figsize': (12, 6)})
sns.barplot(x='Aggregate rating',y='Rating Count',data=ratings)

Out[23]: <Axes: xlabel='Aggregate rating', ylabel='Rating Count'>

Observation:
Not Rated count is very high
Maximum number of rating are between 2.5 to 3.4

In [29]: final_df.columns
Out[29]: 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 [30]: countries_with_zero_rating = final_df.loc[final_df['Aggregate rating']==0, 'Country'].unique()

In [31]: countries_with_zero_rating
Out[31]: array(['Brazil', 'United States', 'India', 'United Kingdom'], dtype=object)

In [34]: final_df[final_df['Aggregate rating']==0].groupby('Country').size().reset_index()

Out[34]:
  Country 0
0 Brazil 5
1 India 2139
2 United Kingdom 1
3 United States 3

Observations: Maximum number of 0 ratings are from Indian customers

In [35]: final_df.groupby(['Currency','Country']).size().reset_index()

Out[35]:
  Currency Country 0
0 Botswana Pula(P) Philippines 22
1 Brazilian Real(R$) Brazil 60
2 Dollar($) Australia 24
3 Dollar($) Canada 4
4 Dollar($) Singapore 20
5 Dollar($) United States 434
6 Emirati Dirham(AED) UAE 60
7 Indian Rupee(Rs.) India 2652
8 Indonesian Rupiah(IDR) Indonesia 21
9 NewZealand($) New Zealand 40
10 Pound($£) United Kingdom 80
11 Qatari Rial(QR) Qatar 20
12 Rand(R) South Africa 60
13 Sri Lankan Rupee(LKR) Sri Lanka 2
14 Turkish Lira(TL) Turkey 34

In [42]: final_df[final_df['Has Online delivery']== 'Yes'].groupby('Country').size().reset_index()

Out[42]:
  Country 0
0 India 2423
1 UAE 28

In [46]: final_df.loc[final_df['Has Online delivery']== 'Yes','Country'].unique()
Out[46]: array(['UAE', 'India'], dtype=object)

Observations:
Online Deliveries are available in India and UAE

In [43]: city_values=final_df.City.value_counts().values
city_labels=final_df.City.value_counts().index

In [44]: plt.pie(city_values[:5],labels=city_labels[:5],autopct='%1.2f%%')

Out[44]: ([<matplotlib.patches.Wedge at 0x2448a19ff80>,
<matplotlib.patches.Wedge at 0x244808a8150>,
<matplotlib.patches.Wedge at 0x244841da0690>,
<matplotlib.patches.Wedge at 0x2448c91a350>,
<matplotlib.patches.Wedge at 0x2448bf9dc990>],
[Text(0.6145352821858392, 0.1233915696708653, 'New Delhi'),
Text(0.8623675251198054, -1.09823595276263487, 'Gurgaon'),
Text(0.878945225023268, -0.602481107353246, 'Noida'),
Text(1.092521843822437, -0.1395118407959229, 'Faridabad'),
Text(1.0999462800895612, -0.01087113182029924, 'Ghaziabad')],
[Text(0.325509063574145, -0.497034952402289, '68.97%'),
Text(0.8340186580653484, -0.5990348332507311, '14.07%'),
Text(0.47946246885229276, -0.36979533641101336, '13.59%'),
Text(0.5957727680627325, -0.87253501898941394, '0.10%'),
Text(0.599776081848791, -0.86952658989289649, '0.31%')])

In [47]: final_df['Cuisines'].value_counts().nlargest(10)

Out[47]:
Cuisines
North Indian 896
North Indian, Chinese 511
Chinese 354
Fast Food 354
North Indian, Mughlai 334
Cafe 299
Bakery 218
North Indian, Mughlai, Chinese 197
Bakery, Deserts 178
Street Food 149
Name: count, dtype: int64

Observations: North Indian is the most frequent cuisine

In [55]: top_rated_restaurants = final_df.nlargest(10, 'Aggregate rating')
print(top_rated_restaurants[['Restaurant Name', 'Country', 'Aggregate rating']])

Out[55]:
  Restaurant Name Country Aggregate rating
3 Spiral - SoFiTel Philippine Plaza Manila Philippines 4.9
10 Silantro Fil-Mex Philippines 4.9
99 Coco Bubu Brazil 4.9
49 Braseiro da Gifea Brazil 4.9
58 Garota de Japanea Brazil 4.9
124 Rae's Coastal Cafe United States 4.9
175 Short's Burger and Shine United States 4.9
213 Oldwood Cafe United States 4.9
238 Tantra Asian Bistro United States 4.9

In [57]: final_df[final_df['Aggregate rating']==4.9].groupby('Country').size().reset_index()

Out[57]:
  Country 0
0 Brazil 3
1 India 19
2 Indonesia 4
3 New Zealand 2
4 Philippines 3
5 Qatar 1
6 South Africa 3
7 Sri Lanka 1
8 Turkey 3
9 UAE 4
10 United Kingdom 4
11 United States 14

In [63]: restaurants_in_india = final_df[final_df['Country'] == 'India'][['Restaurant Name', 'Aggregate rating','Votes']]
restaurants_in_india

Out[63]:
  Restaurant Name Aggregate rating Votes
624 Jahangpanah 3.9 140
625 Pangrazz Restaurant 3.5 71
626 Time2Eat - Mama Chicken 3.6 94
627 Chokho Jeeman Marwari Jain Bhojanayla 4.0 87
628 Pinch Of Spice 4.2 177
... ..
9271 D Cabana 3.6 193
9272 Kulturez 3.7 85
9273 Plot 17 4.3 172
9274 Vida - The Park 3.8 74
9275 Flying Spaghetti Monster 4.4 316
8652 rows x 3 columns

In [66]: restaurants_in_india = final_df[final_df['Country'] == 'India'][['Restaurant Name', 'Aggregate rating','Votes']]
restaurants_in_india

Out[66]:
  Restaurant Name Aggregate rating Votes
624 Jahangpanah 3.9 140
625 Pangrazz Restaurant 3.5 71
626 Time2Eat - Mama Chicken 3.6 94
627 Chokho Jeeman Marwari Jain Bhojanayla 4.0 87
628 Pinch Of Spice 4.2 177
9271 D Cabana 3.6 193
9272 Kulturez 3.7 85
9273 Plot 17 4.3 172
9274 Vida - The Park 3.8 74
9275 Flying Spaghetti Monster 4.4 316
8652 rows x 3 columns

In [71]: plt.figure(figsize=(12, 6))
plt.bar(restaurants_in_india.top10['Restaurant Name'], restaurants_in_india.top10['Votes'], color='skyblue')
plt.xlabel('Restaurant Name')
plt.ylabel('Votes')
plt.title('Top Voted Restaurants in India')
plt.xticks(rotation=45, ha='right')
plt.show()

Top Voted Restaurants in India

In [78]: restaurants_in_india.top10 = restaurants_in_india.nlargest(10, 'Votes')
restaurants_in_india.top10

Out[78]:
  Restaurant Name Aggregate rating Votes
725 Tot 4.8 10934
735 Truffles 4.7 9667
3894 Hazz Khas Social 4.3 7531
2412 Peter Cat 4.3 7514
739 AB's - Absolute Barbecues 4.6 6907
2414 Barbeque Nation 4.9 5966
743 Big Brewsky 4.5 5705
2307 AB's - Absolute Barbecues 4.9 5434
736 The Black Pearl 4.1 5385
2411 BarQ 4.2 5288
3110 Saravana Bhavan 4.3 5172
2480 Joey's Pizza 4.0 5145
4638 Big Chili 4.5 4986
3085 Warehouse Cafe 3.7 4914
4178 Karim's 4.0 4689
```


