

Identifying and Recommending Best Restaurants

Project 1

DESCRIPTION

Data Analysis is the process of creating a story using the data for easy and effective communication. It mostly utilizes visualization methods like plots, charts, and tables to convey what the data holds beyond the formal modeling or hypothesis testing task.

Domain: Marketing

Read the information given below and also refer to the data dictionary provided separately in an excel file to build your understanding.

Problem Statement A restaurant consolidator is looking to revamp its B-to-C portal using intelligent automation tech. It is in search of different matrix to identify and recommend restaurants. To make sure an effective model can be achieved it is important to understand the behaviour of the data in hand.

Approach:

1. Data Preliminary analysis:

Perform preliminary data inspection and report the findings as the structure of the data, missing values, duplicates cleaning variable names etc. Based on the findings from the previous questions identify duplicates and remove them.

1. Prepare a preliminary report of the given data by answering following questions. Expressing the results using graphs and plot will make it more appealing.

Explore the geographical distribution of the restaurants, finding out the cities with maximum / minimum number of restaurants. Explore how ratings are distributed overall. Restaurant franchise is a thriving venture. So, it becomes very important to explore the franchise with most national presence. What is the ratio between restaurants that allow table booking vs that do not allow table booking? What is the percentage of restaurants providing online delivery? Is there a difference in no. of votes for the restaurants that deliver and the restaurant that don't? What are the top 10 cuisines served across cities? What is the maximum and minimum no. of cuisines that a restaurant serves? Also, what is the relationship between No. of cuisines served and Ratings Discuss the cost vs the other variables. Explain the factors in the data that may have an effect on ratings e.g. No. of cuisines, cost, delivery option etc. All the information gathered here will lead to a better understanding of the data and allow for a better implementation of ML models.

Project Task: Week 1

Importing, Understanding, and Inspecting Data :

Perform preliminary data inspection and report the findings as the structure of the data, missing values, duplicates, etc.

Based on the findings from the previous questions, identify duplicates and remove them

Performing EDA:

Explore the geographical distribution of the restaurants and identify the cities with the maximum and minimum number of restaurants

Restaurant franchising is a thriving venture. So, it is very important to explore the franchise with most national presence

Find out the ratio between restaurants that allow table booking vs. those that do not allow table booking

Find out the percentage of restaurants providing online delivery

Calculate the difference in number of votes for the restaurants that deliver and the restaurants that do not deliver

Project Task: Week 2

Performing EDA:

What are the top 10 cuisines served across cities?

What is the maximum and minimum number of cuisines that a restaurant serves? Also, which is the most served cuisine across the restaurant for each city?

What is the distribution cost across the restaurants?

How ratings are distributed among the various factors?

Explain the factors in the data that may have an effect on ratings. For example, number of cuisines, cost, delivery option, etc.

Dashboarding:

Visualize the variables using Tableau to help user explore the data and create a better understanding of the restaurants to identify the "star" restaurant

Demonstrate the variables associated with each other and factors to build a dashboard

```
In [3]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [12]: cc=pd.read_excel("Country-Code.xlsx")
```

```
In [13]: data=pd.read_excel("data.xlsx")

In [14]: variable=pd.read_excel("variable description.xlsx")

In [15]: cc.head()
```

Out[15]:

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

```
In [9]: data.head()
```

Out[9]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude
0	7402935	Skye	94	Jakarta	Menara BCA, Lantai 56, Jl. MH. Thamrin, Thamri...	Grand Indonesia Mall, Thamrin	Grand Indonesia Mall, Thamrin, Jakarta	106.821999	-6.196778
1	7410290	Satoo - Hotel Shangri-La	94	Jakarta	Hotel Shangri-La, Jl. Jend. Sudirman	Hotel Shangri-La, Sudirman	Hotel Shangri-La, Sudirman, Jakarta	106.818961	-6.203292
2	7420899	Sushi Masa	94	Jakarta	Jl. Tuna Raya No. 5, Penjaringan	Penjaringan	Penjaringan, Jakarta	106.800144	-6.101298
3	7421967	3 Wise Monkeys	94	Jakarta	Jl. Suryo No. 26, Senopati, Jakarta	Senopati	Senopati, Jakarta	106.813400	-6.235241
4	7422489	Avec Moi Restaurant and Bar	94	Jakarta	Gedung PIC, Jl. Teluk Betung 43, Thamrin, Jakarta	Thamrin	Thamrin, Jakarta	106.821023	-6.196270

```
In [10]: variable.head()
```

Out[10]:

	Variable	Description
0	Restaurant ID	Identification Number
1	Restaurant Name	Name Of the Restaurant
2	Country Code	Country code
3	City	City Name of the Restaurant
4	Address	Detailed address of the restaurant

In [11]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 19 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Restaurant ID         9551 non-null   int64
1   Restaurant Name       9550 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  Price range           9551 non-null   int64
15  Aggregate rating      9551 non-null   float64
16  Rating color          9551 non-null   object
17  Rating text           9551 non-null   object
18  Votes                 9551 non-null   int64
dtypes: float64(3), int64(5), object(11)
memory usage: 1.4+ MB
```

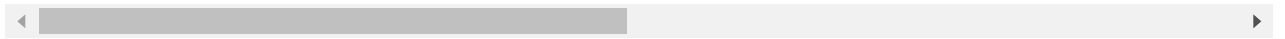
In [16]:

```
df_rest = pd.merge(data,cc,on='Country Code',how='left')
df_rest.head()
```

Out[16]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude
0	7402935	Skye	94	Jakarta	Menara BCA, Lantai 56, Jl. MH. Thamrin, Thamrin...	Grand Indonesia Mall, Thamrin	Grand Indonesia Mall, Thamrin, Jakarta	106.821999	-6.196778
1	7410290	Satoo - Hotel Shangri-La	94	Jakarta	Hotel Shangri-La, Jl. Jend. Sudirman	Hotel Shangri-La, Sudirman	Hotel Shangri-La, Sudirman, Jakarta	106.818961	-6.203292

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude
2	7420899	Sushi Masa	94	Jakarta	Jl. Tuna Raya No. 5, Penjaringan	Penjaringan	Penjaringan, Jakarta	106.800144	-6.101298
3	7421967	3 Wise Monkeys	94	Jakarta	Jl. Suryo No. 26, Senopati, Jakarta	Senopati	Senopati, Jakarta	106.813400	-6.235241
4	7422489	Avec Moi Restaurant and Bar	94	Jakarta	Gedung PIC, Jl. Teluk Betung 43, Thamrin, Jakarta	Thamrin	Thamrin, Jakarta	106.821023	-6.196270



```
In [17]: df_rest.columns = df_rest.columns.str.replace(' ', '_')
df_rest.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', 'Price_range', 'Aggregate_rating',
        'Rating_color', 'Rating_text', 'Votes', 'Country'],
        dtype='object')
```

```
In [18]: df_rest.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9551 entries, 0 to 9550
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Restaurant_ID                        9551 non-null   int64
1   Restaurant_Name                      9550 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  Price_range                          9551 non-null   int64
15  Aggregate_rating                     9551 non-null   float64
16  Rating_color                         9551 non-null   object
17  Rating_text                          9551 non-null   object
18  Votes                               9551 non-null   int64
19  Country                             9551 non-null   object
```

dtypes: float64(3), int64(5), object(12)
memory usage: 1.5+ MB

In [19]: `df_rest.isnull().sum() #total number of null entries per column`

Out[19]:

Restaurant_ID	0
Restaurant_Name	1
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
Price_range	0
Aggregate_rating	0
Rating_color	0
Rating_text	0
Votes	0
Country	0

dtype: int64

In [20]: `df_rest[df_rest['Restaurant_Name'].isnull()]`

Out[20]:

	Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_Verbo
1646	113702	NaN	1	Ahmedabad	Opposite Sindhu Bhawan, Bodakdev, Ahmedabad	Bodakdev	Bodakdev Ahmeda

In [21]:

```
#Since the restaurant name is missing, we dropped the record and reset the index.  
df_rest.dropna(axis=0,subset=['Restaurant_Name'],inplace=True)  
df_rest.reset_index(drop=True,inplace=True)  
df_rest[df_rest['Cuisines'].isnull()]
```

Out[21]:

	Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_Verbo
9082	17374552	Corkscrew Cafe	216	Gainesville	51 W Main St, Dahlonega, GA 30533	Dahlonega	Dahlonega Gainesvi
9085	17501439	Dovetail	216	Macon	543 Cherry St, Macon, GA 31201	Macon	Macon, Mac

	Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_Verbo
9093	17059060	Hillstone	216	Orlando	215 South Orlando Avenue, Winter Park, FL 32789	Winter Park	Winter Pa Orlan
9405	17284158	Jimmie's Hot Dogs	216	Albany	204 S Jackson St, Albany, GA 31701	Albany	Albany, Alba
9493	17142698	Leonard's Bakery	216	Rest of Hawaii	933 Kapahulu Ave, Honolulu, HI 96816	Kaimuki	Kaimuki, Rest Haw
9503	17616465	Tybee Island Social Club	216	Savannah	1311 Butler Ave, Tybee Island, GA 31328	Tybee Island	Tybee Islar Savann
9532	17284105	Cookie Shoppe	216	Albany	115 N Jackson St, Albany, GA 31701	Albany	Albany, Alba
9534	17284211	Pearly's Famous Country Cookng	216	Albany	814 N Slappeg Blvd, Albany, GA 31701	Albany	Albany, Alba
9538	17606621	HI Lite Bar & Lounge	216	Miller	109 N Broadway Ave, Miller, SD 57362	Miller	Miller, Mil

In [22]:

```
#Since there were only 9 records without cuisines, we have replace the null values with  
df_rest['Cuisines'].fillna('Others',inplace=True)
```

In [23]:

```
df_rest.isnull().sum()  
df_rest.info()  
  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 9550 entries, 0 to 9549  
Data columns (total 20 columns):  
#      Column              Non-Null Count  Dtype  
---  -  
0     Restaurant_ID          9550 non-null  int64  
1     Restaurant_Name        9550 non-null  object  
2     Country_Code           9550 non-null  int64  
3     City                   9550 non-null  object  
4     Address                9550 non-null  object  
5     Locality               9550 non-null  object
```

```

6  Locality_Verbose      9550 non-null object
7  Longitude             9550 non-null float64
8  Latitude              9550 non-null float64
9  Cuisines               9550 non-null object
10 Average_Cost_for_two  9550 non-null int64
11 Currency              9550 non-null object
12 Has_Table_booking     9550 non-null object
13 Has_Online_delivery   9550 non-null object
14 Price_range           9550 non-null int64
15 Aggregate_rating      9550 non-null float64
16 Rating_color          9550 non-null object
17 Rating_text           9550 non-null object
18 Votes                 9550 non-null int64
19 Country               9550 non-null object
dtypes: float64(3), int64(5), object(12)
memory usage: 1.5+ MB

```

```

In [24]: cntry_dist = df_rest.groupby(['Country_Code', 'Country']).agg( Count = ('Restaurant_ID',
cntry_dist.sort_values(by='Count', ascending=False)
#We observe that India has then highest number of restaurants with 8651 restaurants and

```

Out[24]:

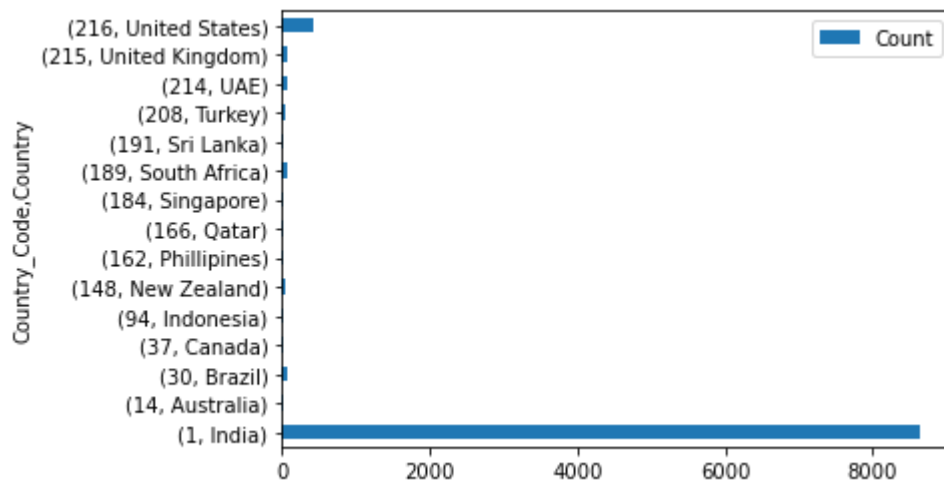
		Count
Country_Code	Country	
1	India	8651
216	United States	434
215	United Kingdom	80
30	Brazil	60
189	South Africa	60
214	UAE	60
148	New Zealand	40
208	Turkey	34
14	Australia	24
162	Phillipines	22
94	Indonesia	21
166	Qatar	20
184	Singapore	20
191	Sri Lanka	20
37	Canada	4

```

In [25]: cntry_dist.plot(kind='barh')

```

Out[25]: <AxesSubplot:ylabel='Country_Code, Country'>



```
In [26]: city_dist = df_rest.groupby(['Country', 'City']).agg(Count = ('Restaurant_ID', 'count'))
city_dist.describe()
#city with max restaurant has count = 5473
#city with min restaurant has count = 1
```

```
Out[26]:
```

	Count
count	141.000000
mean	67.730496
std	476.723952
min	1.000000
25%	1.000000
50%	20.000000
75%	20.000000
max	5473.000000

```
In [27]: city_dist.sort_values(by='Count', ascending=False)
# we see that new Delhi has the maximum restaurant with 5473
# we observe that multiple cities have only one restaurant.
```

```
Out[27]:
```

Country	City	Count
India	New Delhi	5473
	Gurgaon	1118
	Noida	1080
	Faridabad	251
	Ghaziabad	25

	Panchkula	1

		Count
Country	City	
Australia	Balingup	1
Indonesia	Bandung	1
Phillipines	Quezon City	1
United States	Winchester Bay	1

141 rows × 1 columns

In [28]:

```
min_cnt_rest = city_dist[city_dist['Count']==1]
min_cnt_rest.info()
min_cnt_rest
#There are 46 cities in 7 different countries with 1 restaurants

<class 'pandas.core.frame.DataFrame'>
MultiIndex: 46 entries, ('Australia', 'Armidale') to ('United States', 'Winchester Bay')
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Count    46 non-null       int64
dtypes: int64(1)
memory usage: 1.8+ KB
```

Out[28]:

		Count
Country	City	
Australia	Armidale	1
	Balingup	1
	Beechworth	1
	Dicky Beach	1
	East Ballina	1
	Flaxton	1
	Forrest	1
	Huskisson	1
	Inverloch	1
	Lakes Entrance	1
	Lorn	1
	Macedon	1
	Mayfield	1
	Middleton Beach	1
	Montville	1
	Palm Cove	1

		Count
Country	City	
	Paynesville	1
	Penola	1
	Phillip Island	1
	Tanunda	1
	Trentham East	1
	Victor Harbor	1
	Chatham-Kent	1
Canada	Consort	1
	Vineland Station	1
	Yorkton	1
India	Mohali	1
	Panchkula	1
Indonesia	Bandung	1
Phillipines	Quezon City	1
	Tagaytay City	1
South Africa	Randburg	1
United States	Clatskanie	1
	Cochrane	1
	Fernley	1
	Lakeview	1
	Lincoln	1
	Mc Millan	1
	Miller	1
	Monroe	1
	Ojo Caliente	1
	Potrero	1
	Princeton	1
	Vernonia	1
	Weirton	1
	Winchester Bay	1

```
In [29]: max_rate = df_rest.sort_values(by='Aggregate_rating',ascending=False).groupby(['Country
#highest rating restaurants
```

```
min_rate = df_rest.sort_values(by='Aggregate_rating',ascending=False).groupby(['Country', 'City'])\n#Lowest rating restaurants
```

```
In [30]: df_max=max_rate[['Country','City','Restaurant_Name','Aggregate_rating']] #new dataframe\n\n        df_min=min_rate[['Country','City','Restaurant_Name','Aggregate_rating']] #new dataframe\n\n        rating_rest=df_max.merge(df_min,left_on='City',right_on='City',how='inner') #merge into
```

```
In [31]: rating_rest
```

Out[31]:

	Country_x	City	Restaurant_Name_x	Aggregate_rating_x	Country_y	Restaurant_Name_y	Aggregate_rating_y
0	Australia	Armidale	Whitebull Hotel	3.5	Australia	Whitebull Hotel	3.5
1	Australia	Balingup	Taste of Balingup	3.2	Australia	Taste of Balingup	3.2
2	Australia	Beechworth	Bridge Road Brewers	4.6	Australia	Bridge Road Brewers	4.6
3	Australia	Dicky Beach	The Giggling Goat	3.6	Australia	The Giggling Goat	3.6
4	Australia	East Ballina	The Belle General	4.1	Australia	The Belle General	4.1
...
136	United States	Valdosta	Smok'n Pig B-B-Q	4.1	United States	El Toreo Mexican Restaurant	4.1
137	United States	Vernonia	Blue House Cafe	4.3	United States	Blue House Cafe	4.3
138	United States	Waterloo	Four Queens Dairy Cream	3.9	United States	Masala Grill & Coffee House	3.9
139	United States	Weirton	Theo Yianni's Authentic Greek Restaurant	3.9	United States	Theo Yianni's Authentic Greek Restaurant	3.9
140	United States	Winchester Bay	Fishpatrick's Crabby Cafe	3.2	United States	Fishpatrick's Crabby Cafe	3.2

141 rows × 7 columns



```
In [32]: rating_rest.drop(columns='Country_y',axis=1,inplace=True)\n        rating_rest.columns = ['Country','City','Highest Rated Restaurant','Rating Max','Lowest Rated Restaurant','Rating Min']\n        rating_rest
```

Out[32]:

	Country	City	Highest Rated Restaurant	Rating Max	Lowest Rated Restaurant	Rating Min
0	Australia	Armidale	Whitebull Hotel	3.5	Whitebull Hotel	3.5
1	Australia	Balingup	Taste of Balingup	3.2	Taste of Balingup	3.2
2	Australia	Beechworth	Bridge Road Brewers	4.6	Bridge Road Brewers	4.6

	Country	City	Highest Rated Restaurant	Rating Max	Lowest Rated Restaurant	Rating Min
3	Australia	Dicky Beach	The Giggling Goat	3.6	The Giggling Goat	3.6
4	Australia	East Ballina	The Belle General	4.1	The Belle General	4.1
...
136	United States	Valdosta	Smok'n Pig B-B-Q	4.1	El Toreo Mexican Restaurant	3.1
137	United States	Vernonia	Blue House Cafe	4.3	Blue House Cafe	4.3
138	United States	Waterloo	Four Queens Dairy Cream	3.9	Masala Grill & Coffee House	3.2
139	United States	Weirton	Theo Yianni's Authentic Greek Restaurant	3.9	Theo Yianni's Authentic Greek Restaurant	3.9
140	United States	Winchester Bay	Fishpatrick's Crabby Cafe	3.2	Fishpatrick's Crabby Cafe	3.2

141 rows × 6 columns

In [33]:

```
#since India and USA has the most number of restaurants,
#we will try to see the distribution of restaurants ratings for these two countries.

from plotly.offline import download_plotlyjs, init_notebook_mode, iplot
from plotly.graph_objs import *
init_notebook_mode()
import plotly.graph_objs as go #importing plotly or graphs
```

In [34]:

```
rating_rest_city_india=rating_rest[rating_rest['Country']=='India'] #storing the datafr
rating_rest_city_india #In India
city=rating_rest_city_india['City'].tolist()#converting the series to list
rate_max=rating_rest_city_india['Rating Max'].tolist()#converting the series to list
rate_min=rating_rest_city_india['Rating Min'].tolist()#converting the series to list
rest_name_high=rating_rest_city_india['Highest Rated Restaurant'].tolist()#converting t
rest_name_low=rating_rest_city_india['Lowest Rated Restaurant'].tolist()
```

In [35]:

```
stack0 = go.Bar( # GroupBarChart 1 (Highest Rated Resturant)
    x=city,#x axis label
    y=rate_max,# y axis label
    text=rest_name_high,# the value of the restaurant
    name='Highest Rated Restaurant',
    marker=dict(
        color='rgb(76,153,0)', #color of the bar graph's marker
        line=dict(
            color='rgb(76,153,0)', #color of the bar graph's line
            width=1.5, #width of the bar graph
        )
    ),
    opacity=1.0
```

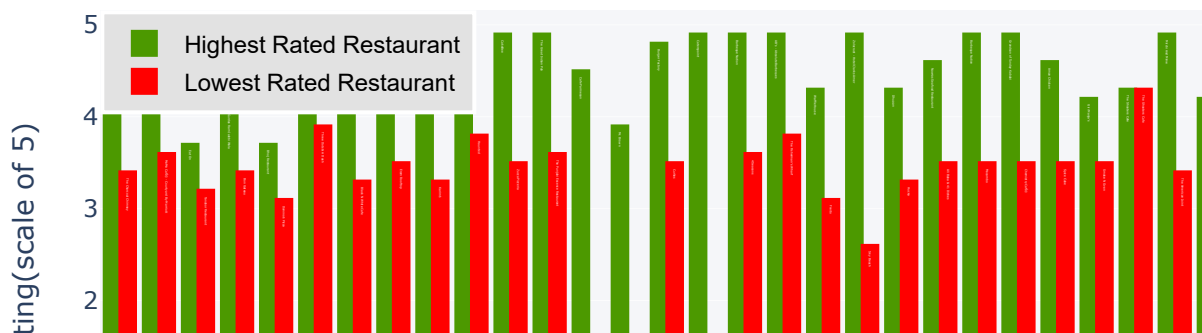
```

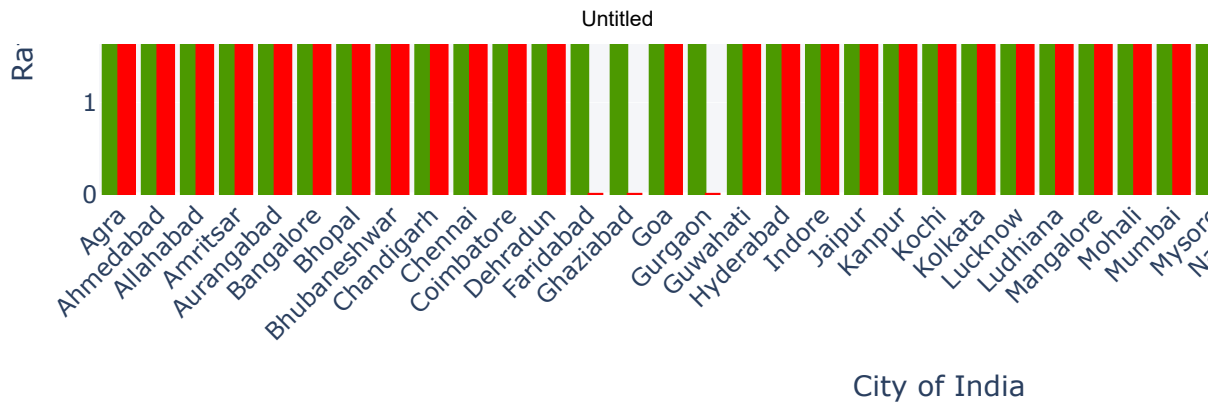
)
stack1 = go.Bar( # GroupBarChart 2 (Lowest Rated Restaurant)
    x=city,
    y=rate_min,
    text=rest_name_low,
    name='Lowest Rated Restaurant',
    marker=dict(
        color='rgb(255,0,0)',#color of the bar graph's marker
        line=dict(
            color='rgb(255,0,0)',#color of the bar graph's line
            width=1.5, #width of the bar graph
        )
    ),
    opacity=1.0
)

data = [stack0,stack1]
layout = go.Layout(
    legend=dict( #the layout of the graph( beautification)
        x=0,
        y=1,
        traceorder='normal',
        font=dict(
            family='sans-serif',
            size=12,
            color='#000'
        ),
        bgcolor='#E2E2E2',
        bordercolor='#FFFFFF',
        borderwidth=2
    ),
    autosize=False,
    width=1000, # size of the graph
    height=450,
    barmode='group',
    title="Graph 1.1: Restaurants rating of India <br>\
<i>hover with cursor to see restaurant's name</i>", #title of the graph
    plot_bgcolor='rgba(245, 246, 249, 1)',
    xaxis=dict(tickangle=-45,title= 'City of India'), #making the graphs label inclined
    yaxis= {'title': 'Rating(scale of 5)'} #label of y-axis
)
fig = go.Figure(data=data, layout=layout) #plotting the graph
iplot(fig, filename='style-barbar')

```

Graph 1.1: Restaurants rating of India
hover with cursor to see restaurant's name





```
In [36]: #perform the same steps as above for Country='United States'
rating_rest_city_usa = rating_rest[rating_rest['Country']=='United States']
rating_rest_city_usa
cityu = rating_rest_city_usa['City'].tolist()
rate_maxu = rating_rest_city_usa['Rating Max'].tolist()
rate_minu = rating_rest_city_usa['Rating Min'].tolist()
rest_name_highu = rating_rest_city_usa['Highest Rated Restaurant'].tolist()
rest_name_lowu = rating_rest_city_usa['Lowest Rated Restaurant'].tolist()
```

```
In [37]: stack0 = go.Bar( # GroupBarChart 1 (Highest Rated Resturant)
    x=cityu,#x axis Label
    y=rate_maxu,# y axis Label
    text=rest_name_highu,# the value of the restaurant
    name='Highest Rated Restaurant',
    marker=dict(
        color='rgb(76,153,0)', #color of the bar graph's marker
        line=dict(
            color='rgb(76,153,0)', #color of the bar graph's Line
            width=1.5, #width of the bar graph
        )
    ),
    opacity=1.0
)
stack1 = go.Bar( # GroupBarChart 2 (Lowest Rated Resturant)
    x=cityu,
    y=rate_minu,
    text=rest_name_lowu,
    name='Lowest Rated Restaurant',
    marker=dict(
        color='rgb(255,0,0)',#color of the bar graph's marker
        line=dict(
            color='rgb(255,0,0)',#color of the bar graph's Line
            width=1.5, #width of the bar graph
        )
    ),
    opacity=1.0
)

data = [stack0,stack1]
layout = go.Layout(
    legend=dict( #the layout of the graph( beautification)
        x=0,
        y=1,
        traceorder='normal',
```

```

font=dict(
    family='sans-serif',
    size=12,
    color='#000'
),
bgcolor='#E2E2E2',
bordercolor='#FFFFFF',
borderwidth=2
),
autosize=False,
width=1000, # size of the graph
height=450,
barmode='group',
title="Graph 1.1: Restaurants rating of USA <br>\
<i>hover with cursor to see restaurant's name</i>", #title of the graph
plot_bgcolor='rgba(245, 246, 249, 1)',
xaxis=dict(tickangle=-45,title= 'City of USA'), #making the graphs label inclined a
yaxis= {'title': 'Rating(scale of 5)'} #label of y-axis
)
fig = go.Figure(data=data, layout=layout) #plotting the graph
iplot(fig, filename='style-barbar')

```

Graph 1.1: Restaurants rating of USA
hover with cursor to see restaurant's name



```

In [38]: df_rest1 = df_rest.copy()
df_rest1.columns

```

```

Out[38]: 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', 'Price_range', 'Aggregate_rating',

```



```
'Rating_color', 'Rating_text', 'Votes', 'Country'],
dtype='object')
```

```
In [39]: dummy = ['Has_Table_booking', 'Has_Online_delivery']
df_rest1 = pd.get_dummies(df_rest1, columns=dummy, drop_first=True)
df_rest1.head()
# 0 indicates 'NO'
# 1 indicates 'YES'
```

Out[39]:

	Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_Verbose	L
0	7402935	Skye	94	Jakarta	Menara BCA, Lantai 56, Jl. MH. Thamrin, Thamri...	Grand Indonesia Mall, Thamrin	Grand Indonesia Mall, Thamrin, Jakarta	10
1	7410290	Satoo - Hotel Shangri-La	94	Jakarta	Hotel Shangri-La, Jl. Jend. Sudirman	Hotel Shangri-La, Sudirman	Hotel Shangri-La, Sudirman, Jakarta	10
2	7420899	Sushi Masa	94	Jakarta	Jl. Tuna Raya No. 5, Penjaringan	Penjaringan	Penjaringan, Jakarta	10
3	7421967	3 Wise Monkeys	94	Jakarta	Jl. Suryo No. 26, Senopati, Jakarta	Senopati	Senopati, Jakarta	10
4	7422489	Avec Moi Restaurant and Bar	94	Jakarta	Gedung PIC, Jl. Teluk Betung 43, Thamrin, Jakarta	Thamrin	Thamrin, Jakarta	10

```
In [40]: #Ration between restaurants allowing table booking and those which dont
table_booking = df_rest1[df_rest1['Has_Table_booking_Yes']==1]['Restaurant_ID'].count()
table_nbooking =df_rest1[df_rest1['Has_Table_booking_Yes']==0]['Restaurant_ID'].count()
print('Ratio between restaurants that allow table booking vs. those that do not allow t
round((table_booking/table_nbooking),2))
```

Ratio between restaurants that allow table booking vs. those that do not allow table booking: 0.14

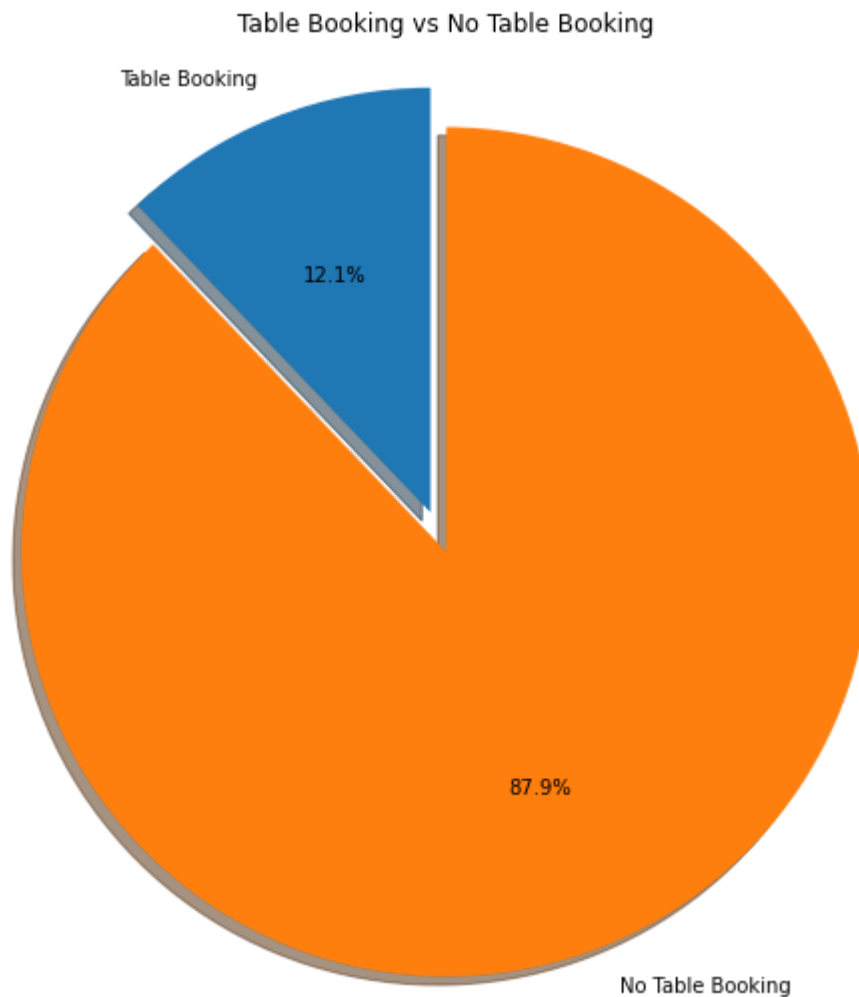
```
In [41]: print(table_booking,table_nbooking)

1158 8392
```

```
In [42]: #Pie chart to show percentage of restaurants which allow table booking and those which
labels = 'Table Booking', 'No Table Booking'
sizes = [table_booking,table_nbooking]
explode = (0.1, 0) # only "explode" the 2nd slice (i.e. 'Hogs')
```

```
fig1, ax1 = plt.subplots(figsize=(9,9))
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%', shadow=True, startangle=90)
ax1.set_title("Table Booking vs No Table Booking")
ax1.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.

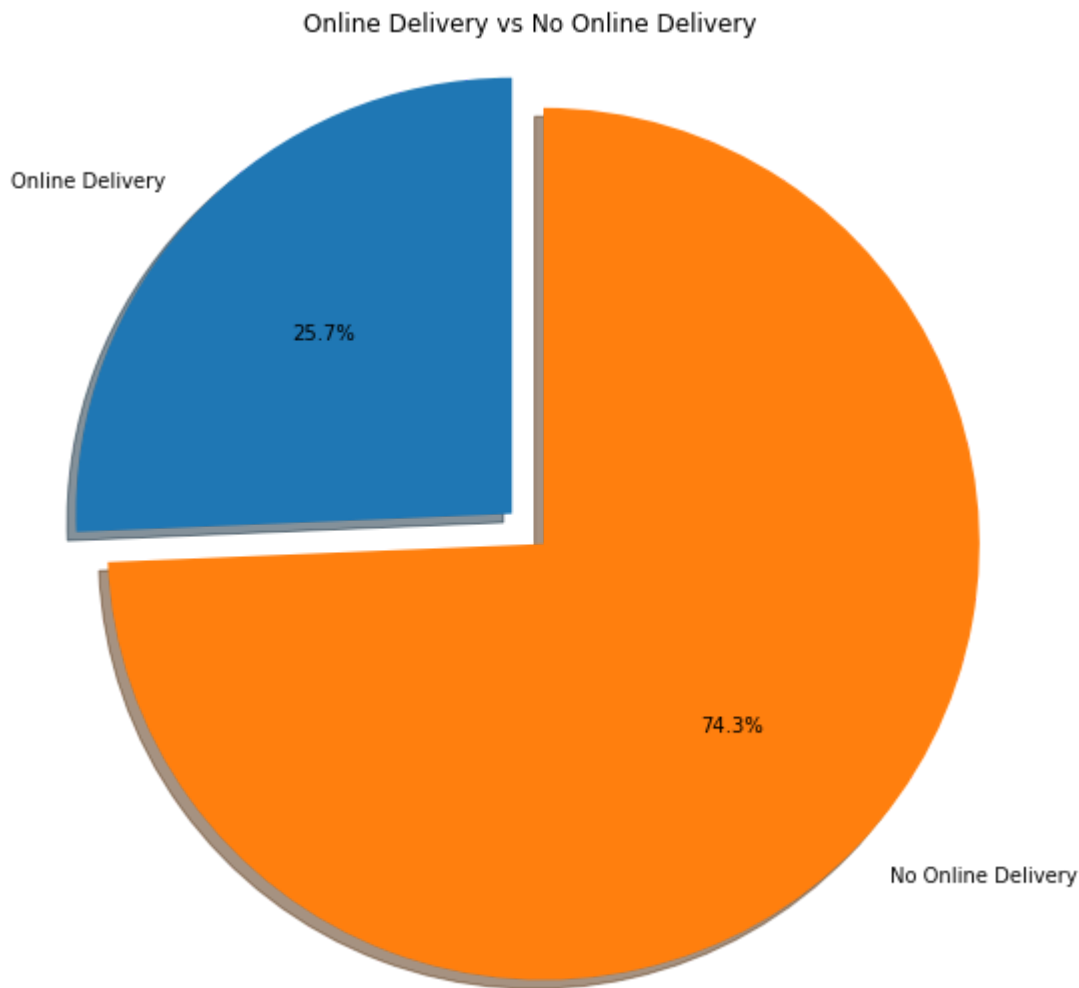
plt.show()
```



```
In [43]: #Percentage of restaurant that has online delivery
rest_od = df_rest1[df_rest1['Has_Online_delivery_Yes'] == 1]['Restaurant_ID'].count()
rest_nod = df_rest1[df_rest1['Has_Online_delivery_Yes'] == 0]['Restaurant_ID'].count()
print('Percentage of restaurants providing online delivery : {} %'.format((round(rest_o
```

Percentage of restaurants providing online delivery : 25.7 %

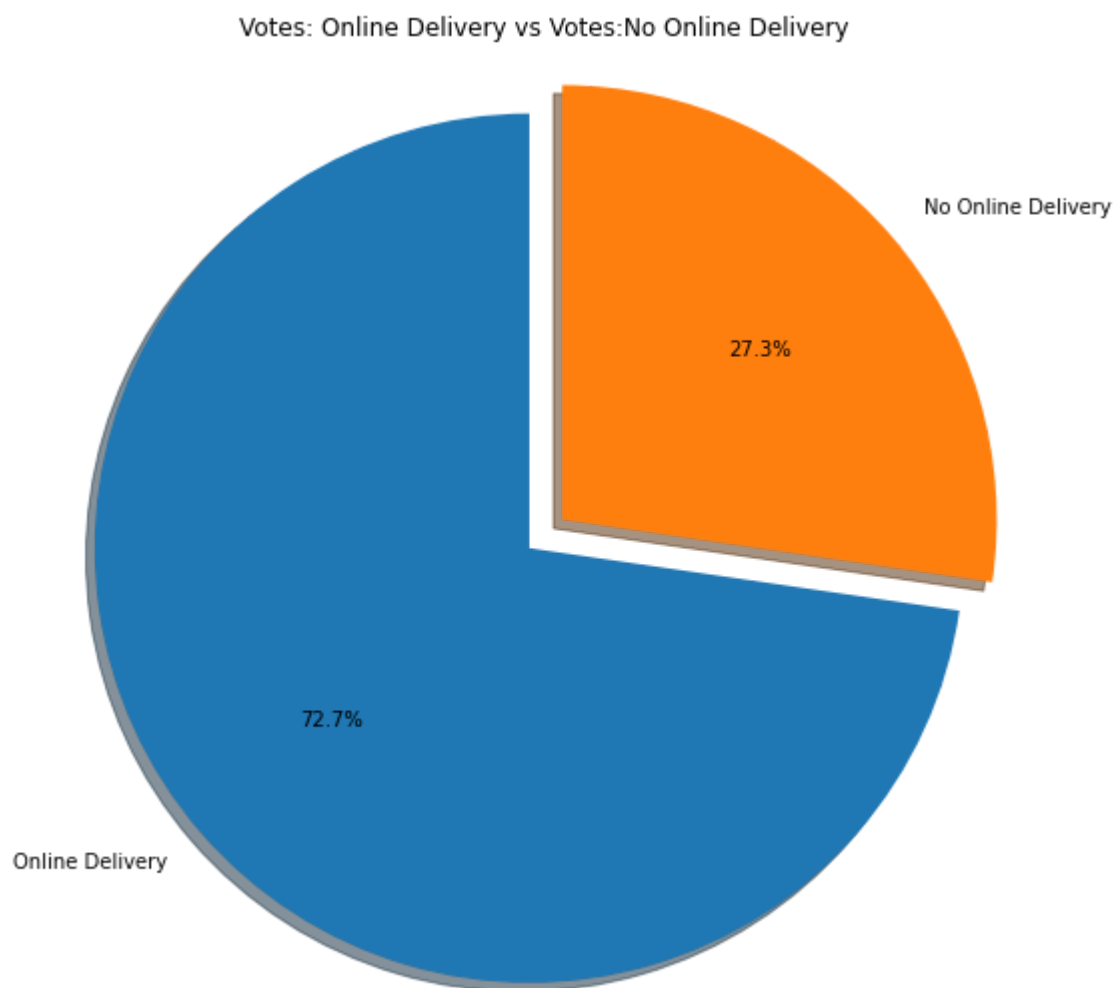
```
In [44]: #pie chart to show percentages of restaurants allowing online delivery vs those which d
labels = 'Online Delivery','No Online Delivery'
size = [rest_od,rest_nod]
explode = (0.1,0)
fig1,ax1 = plt.subplots(figsize=(9,9))
ax1.pie(size,explode=explode,labels=labels,autopct='%1.1f%%', shadow=True,startangle=90)
ax1.set_title("Online Delivery vs No Online Delivery")
ax1.axis('equal')
plt.show()
```



```
In [45]: rest_deliver = df_rest1[df_rest1['Has_Table_booking_Yes'] == 1]['Votes'].sum()
rest_ndeliver = df_rest1[df_rest1['Has_Table_booking_Yes'] == 0]['Votes'].sum()
print('Difference in number of votes for restaurants that deliver and dont deliver: ',a
```

Difference in number of votes for restaurants that deliver and dont deliver: 680082

```
In [46]: labels = 'Online Delivery','No Online Delivery'
size = [rest_ndeliver,rest_deliver]
explode = (0,0.1)
fig1,ax1 = plt.subplots(figsize=(9,9))
ax1.pie(size,explode=explode,labels=labels,autopct='%1.1f%%',shadow=True,startangle=90)
ax1.set_title("Votes: Online Delivery vs Votes:No Online Delivery")
ax1.axis('equal')
plt.show()
#out of the total votes about 27.3% votes were given to restaurants that dont have onli
#out of the total votes about 72.7% votes were given to restaurants that dont have onli
#This clearly shows that restaurants that have online delivery are more likely to get a
```



```
In [47]: df_rest.columns
cuisines = df_rest['Cuisines'].apply(lambda x: pd.Series(x.split(',')))
cuisines.columns = ['Cuisine_1', 'Cuisine_2', 'Cuisine_3', 'Cuisine_4', 'Cuisine_5', 'Cuisine_6', 'Cuisine_7', 'Cuisine_8']
cuisines.tail()
```

```
Out[47]:
```

	Cuisine_1	Cuisine_2	Cuisine_3	Cuisine_4	Cuisine_5	Cuisine_6	Cuisine_7	Cuisine_8
9545	Chinese	North Indian	Fast Food	NaN	NaN	NaN	NaN	NaN
9546	Indian	Chinese	Continental	NaN	NaN	NaN	NaN	NaN
9547	Cafe	Continental	Desserts	Ice Cream	Italian	Beverages	NaN	NaN
9548	Street Food	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9549	Chinese	North Indian	NaN	NaN	NaN	NaN	NaN	NaN

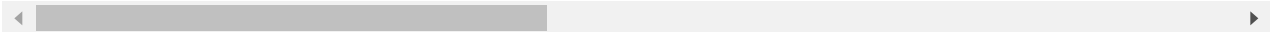
```
In [48]: df_cuisines = pd.concat([df_rest, cuisines], axis=1)
df_cuisines.head()
```

```
Out[48]:
```

Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_Verbose	L
---------------	-----------------	--------------	------	---------	----------	------------------	---

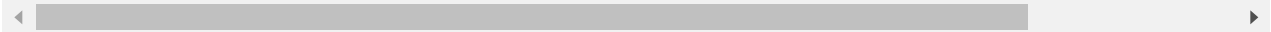
	Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_Verbose	L
0	7402935	Skye	94	Jakarta	Menara BCA, Lantai 56, Jl. MH. Thamrin, Thamri...	Grand Indonesia Mall, Thamrin	Grand Indonesia Mall, Thamrin, Jakarta	10
1	7410290	Satoo - Hotel Shangri-La	94	Jakarta	Hotel Shangri-La, Jl. Jend. Sudirman	Hotel Shangri-La, Sudirman	Hotel Shangri-La, Sudirman, Jakarta	10
2	7420899	Sushi Masa	94	Jakarta	Jl. Tuna Raya No. 5, Penjaringan	Penjaringan	Penjaringan, Jakarta	10
3	7421967	3 Wise Monkeys	94	Jakarta	Jl. Suryo No. 26, Senopati, Jakarta	Senopati	Senopati, Jakarta	10
4	7422489	Avec Moi Restaurant and Bar	94	Jakarta	Gedung PIC, Jl. Teluk Betung 43, Thamrin, Jakarta	Thamrin	Thamrin, Jakarta	10

5 rows × 28 columns



```
In [49]: cuisine_loc = pd.DataFrame(df_cuisines[['Country','City','Locality_Verbose','Cuisine_1','Cuisine_4','Cuisine_5','Cuisine_6','Cuisine_7'])
cuisine_loc_stack=pd.DataFrame(cuisine_loc.stack()) #stacking the columns
cuisine_loc.head()
```

	Country	City	Locality_Verbose	Cuisine_1	Cuisine_2	Cuisine_3	Cuisine_4	Cuisine_5	Cuisine_6
0	Indonesia	Jakarta	Grand Indonesia Mall, Thamrin, Jakarta	Italian	Continental	NaN	NaN	NaN	NaN
1	Indonesia	Jakarta	Hotel Shangri-La, Sudirman, Jakarta	Asian	Indonesian	Western	NaN	NaN	NaN
2	Indonesia	Jakarta	Penjaringan, Jakarta	Sushi	Japanese	NaN	NaN	NaN	NaN
3	Indonesia	Jakarta	Senopati, Jakarta	Japanese	NaN	NaN	NaN	NaN	NaN
4	Indonesia	Jakarta	Thamrin, Jakarta	French	Western	NaN	NaN	NaN	NaN



```
In [50]: keys = [c for c in cuisine_loc if c.startswith('Cuisine')]
a=pd.melt(cuisine_loc, id_vars='Locality_Verbose', value_vars=keys, value_name='Cuisine') #melting the stack into one row
max_rate=pd.DataFrame(a.groupby(by=['Locality_Verbose','variable','Cuisines']).size().r
```

```
#find the highest restuarant in the city
max_rate
del max_rate['variable']
max_rate.columns=['Locality_Verbose','Cuisines','Count']
max_rate.head()
```

Out[50]:

	Locality_Verbose	Cuisines	Count
0	ILD Trade Centre Mall, Sohna Road, Gurgaon	Cafe	1
1	ILD Trade Centre Mall, Sohna Road, Gurgaon	North Indian	1
2	ILD Trade Centre Mall, Sohna Road, Gurgaon	Beverages	1
3	ILD Trade Centre Mall, Sohna Road, Gurgaon	Mughlai	1
4	12th Square Building, Banjara Hills, Hyderabad	Mughlai	1

In [51]:

```
#find the highest restuarant in the city
loc=max_rate.sort_values('Count', ascending=False).groupby(by=['Locality_Verbose'],as_i
loc.head()
```

Out[51]:

	Locality_Verbose	Cuisines	Count
0	ILD Trade Centre Mall, Sohna Road, Gurgaon	Cafe	1
1	12th Square Building, Banjara Hills, Hyderabad	Mughlai	1
2	A Hotel, Gurdev Nagar, Ludhiana	Chinese	1
3	ARSS Mall, Paschim Vihar, New Delhi	North Indian	1
4	Aaya Nagar, New Delhi	Cuisine Varies	1

In [52]:

```
rating_res=loc.merge(df_rest,left_on='Locality_Verbose',right_on='Locality_Verbose',how
#inner join to merge the two dataframe
df=pd.DataFrame(rating_res[['Country','City','Locality_Verbose','Cuisines_x','Count']])
#making a dataframe of rating restaurant
country=rating_res.sort_values('Count', ascending=False).groupby(by=['Country'],as_inde
#grouping the data by country code
con=pd.DataFrame(country[['Country','City','Locality','Cuisines_x','Count']])
con.columns=['Country','City','Locality','Cuisines','Number of restaurants in the count
#renaming the columns
con1=con.sort_values('Number of restaurants in the country', ascending=False)
#sorting the restaurants on the basis of the number of restaurants in the country
con1[:10]
final_con=con1.drop(con1.index[[7,10]])
```

In [53]:

final_con

Out[53]:

	Country	City	Locality	Cuisines	Number of restaurants in the country
3	India	New Delhi	Connaught Place	North Indian	48

	Country	City	Locality	Cuisines	Number of restaurants in the country
14	United States	Dubuque	Dubuque	American	9
5	New Zealand	Wellington City	Te Aro	Cafe	5
1	Brazil	Rio de Janeiro	Ipanema	Brazilian	3
6	Phillipines	Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City	Japanese	2
8	Singapore	Singapore	Marina Centre, Downtown Core	Seafood	2
9	South Africa	Cape Town	Green Point	Grill	2
11	Turkey	Ankara	Gazi Osman PaÅa	World Cuisine	2
12	UAE	Abu Dhabi	Abu Dhabi Mall, Tourist Club Area (Al Zahiyah)	American	2
0	Australia	Victor Harbor	Victor Harbor	Coffee and Tea	1
2	Canada	Vineland Station	Vineland Station	Italian	1
4	Indonesia	Jakarta	Tebet	Western	1
7	Qatar	Doha	The Westin Doha Hotel & Spa, Fereej Bin Mahmoud	Thai	1

In [54]:

```

loc_list=final_con['City'] #converting the series to dataframe
a_list=loc_list.tolist()

cui_list=final_con['Cuisines']# converting the series to dataframe
b_list=cui_list.tolist()

count_list=final_con['Number of restaurants in the country']# converting the series to
c_list=count_list.tolist()
trace0 = go.Bar(# BarChart 1 (Popular cuisines of the country)
    x=b_list, #x axis label
    y=c_list, # y axis label
    text=loc_list, # location of the cuisine
    name='Popular Cuisine',
    marker=dict(
        color=[ 'rgb(255,69,0)',
                 'rgb(255,140,0)',
                 'rgb(165,42,42)',
                 'rgb(220,20,60)',
                 'rgb(255,0,0)',
                 'rgb(255,99,71)',
                 'rgb(255,127,80)',
                 'rgb(205,92,92)',
                 'rgb(240,128,128)',
                 'rgb(233,150,122)'],
    )

```

```

        'rgb(250,128,114)',
        'rgb(255,160,122)'],
    line=dict(
        color='rgb(255,0,0)',#color of the bar graph's line
        width=1.5, #width of the bar graph
    ),
),
opacity=1.0
)
data = [trace0]
layout = go.Layout(

    legend=dict( #the layout of the graph( beautification)
        x=0,
        y=1,
        traceorder='normal',
        font=dict(
            family='sans-serif',
            size=12,
            color='#000'
        ),
        bgcolor='#E2E2E2',
        bordercolor='#FFFFFF',
        borderwidth=20,
    ),
    autosize=False,
    width=1000, # size of the graph
    height=450,
    margin=Margin(r=20, l=300,
                  b=75, t=125),
    title="Graph 2.1 : Most popular cuisines in the World<br>\
    <i>hover with cursor to see location in the country where they are most popular </i>\
    plot_bgcolor='rgba(245, 246, 249, 1)',
    xaxis=dict(tickangle=-45,title= '<br>Cuisine<br>',mirror=True,showticklabels=True),
    #making the graphs label inclined at 45 deg
    yaxis= {'title': 'Number of restaurants offering<br> cuisine in the location'},#Lab
)
fig = go.Figure(data=data, layout=layout)#plotting the graph
iplot(fig)

```

F:\pyhton\lib\site-packages\plotly\graph_objs_deprecations.py:405: DeprecationWarning:

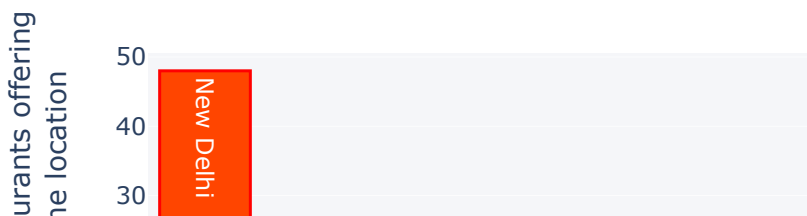
plotly.graph_objs.Margin is deprecated.

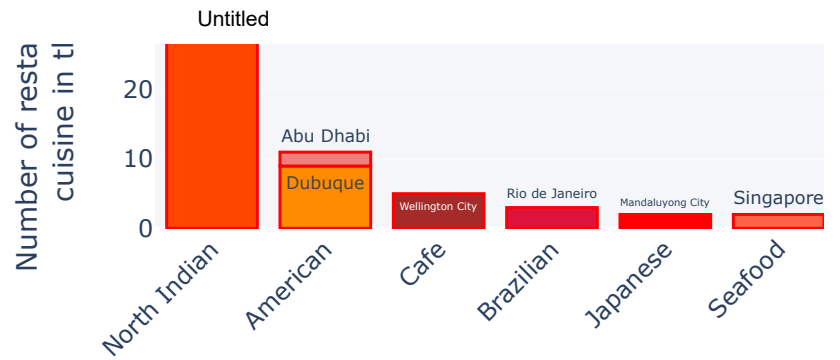
Please replace it with one of the following more specific types

- plotly.graph_objs.layout.Margin

Graph 2.1 : Most popular cuisines in the World

hover with cursor to see location in the country where they are m





Cu

```
In [55]: rest_cuisine = pd.DataFrame(df_cuisines[['Restaurant_Name','City','Cuisine_1','Cuisine_2','Cuisine_3','Cuisine_4','Cuisine_5','Cuisine_6','Cuisine_7','Cuisine_8'])
rest_cuisine_stack=pd.DataFrame(rest_cuisine.stack()) #stacking the columns
rest_cuisine.head()
```

```
Out[55]:
```

	Restaurant_Name	City	Cuisine_1	Cuisine_2	Cuisine_3	Cuisine_4	Cuisine_5	Cuisine_6	Cuisine_7
0	Skye	Jakarta	Italian	Continental	NaN	NaN	NaN	NaN	NaN
1	Satoo - Hotel Shangri-La	Jakarta	Asian	Indonesian	Western	NaN	NaN	NaN	NaN
2	Sushi Masa	Jakarta	Sushi	Japanese	NaN	NaN	NaN	NaN	NaN
3	3 Wise Monkeys	Jakarta	Japanese	NaN	NaN	NaN	NaN	NaN	NaN
4	Avec Moi Restaurant and Bar	Jakarta	French	Western	NaN	NaN	NaN	NaN	NaN

```
In [56]: keys1 = [c for c in rest_cuisine if c.startswith('Cuisine')]
b=pd.melt(rest_cuisine, id_vars='Restaurant_Name', value_vars=keys, value_name='Cuisine') #melting the stack into one row
max_rate1=pd.DataFrame(b.groupby(by=['Restaurant_Name','variable','Cuisines']).size().reset_index()) #find the highest restaurant in the city
max_rate1
del max_rate1['variable']
max_rate1.columns=['Restaurant_Name','Cuisines','Count']
max_rate1.head(20)
```

```
Out[56]:
```

	Restaurant_Name	Cuisines	Count
0	12212	Fast Food	1
1	Let's Burrp	Chinese	1
2	Let's Burrp	North Indian	1
3	#45	Cafe	1
4	#Dilliwaala6	North Indian	1

	Restaurant_Name	Cuisines	Count
5	#InstaFreeze	Ice Cream	1
6	#OFF Campus	Cafe	1
7	#OFF Campus	Continental	1
8	#OFF Campus	Italian	1
9	#OFF Campus	Fast Food	1
10	#Urban Cafè	North Indian	1
11	#Urban Cafè	Chinese	1
12	#Urban Cafè	Italian	1
13	#hashtag	Cafe	1
14	'Ohana	Hawaiian	1
15	10 Downing Street	North Indian	2
16	10 Downing Street	Chinese	2
17	10 To 10 In Delhi	Indian	1
18	10 To 10 In Delhi	Cafe	1
19	11th Avenue Cafe Bistro	Cafe	1

```
In [57]: max_rate1.sort_values('Count',ascending=False)
#Cafe Coffee Day has the max number of cuisines and The Least number of cuisines in a r
```

Out[57]:

	Restaurant_Name	Cuisines	Count
2479	Cafe Coffee Day	Cafe	83
4596	Domino's Pizza	Pizza	79
4597	Domino's Pizza	Fast Food	78
12984	Subway	Salad	63
12985	Subway	Healthy Food	63
...
5568	Gabbar's Bar & Kitchen	Chinese	1
5569	Gabbar's Bar & Kitchen	Mexican	1
5570	Gabbar's Bar & Kitchen	Italian	1
5571	Gaga Manjero	World Cuisine	1
15963	İzgarasöğüt Sofrası	Izgara	1

15964 rows × 3 columns

```
In [58]: Restaurant_Name Cuisines Count
2479 Cafe Coffee Day Cafe 83
```

```
4596 Domino's Pizza Pizza 79
4597 Domino's Pizza Fast Food 78
12984 Subway Salad 63
12985 Subway Healthy Food 63
...
5568 Gabbar's Bar & Kitchen Chinese 1
5569 Gabbar's Bar & Kitchen Mexican 1
5570 Gabbar's Bar & Kitchen Italian 1
5571 Gaga Manjero World Cuisine 1
15963 İaukura€Üa Sofras€± Izgara 1
15964 rows x 3 columns

rating = df_rest1[['Restaurant_ID','Restaurant_Name','Country','City','Aggregate_rating
rating = rating.merge(max_rate1,left_on='Restaurant_Name',right_on='Restaurant_Name',ho
rating
```

File "C:\Users\MYPC\AppData\Local\Temp\ipykernel_1600\677093144.py", line 1

```
Restaurant_Name    Cuisines    Count
              ^
SyntaxError: invalid syntax
```

```
In [59]: rating = df_rest1[['Restaurant_ID','Restaurant_Name','Country','City','Aggregate_rating
```

```
In [60]: rating = rating.merge(max_rate1,left_on='Restaurant_Name',right_on='Restaurant_Name',ho
rating
```

Out[60]:

	Restaurant_ID	Restaurant_Name	Country	City	Aggregate_rating	Average_Cost_for_two	Vc
0	7402935	Skye	Indonesia	Jakarta	4.1	800000	1
1	7402935	Skye	Indonesia	Jakarta	4.1	800000	1
2	7410290	Satoo - Hotel Shangri-La	Indonesia	Jakarta	4.6	800000	
3	7410290	Satoo - Hotel Shangri-La	Indonesia	Jakarta	4.6	800000	
4	7410290	Satoo - Hotel Shangri-La	Indonesia	Jakarta	4.6	800000	
...
23810	18312106	UrbanCrave	India	Kanpur	3.9	0	
23811	18312106	UrbanCrave	India	Kanpur	3.9	0	
23812	3900245	Deena Chat Bhandar	India	Varanasi	3.8	0	
23813	18246202	VNS Live Studio	India	Varanasi	3.5	0	
23814	18246202	VNS Live Studio	India	Varanasi	3.5	0	

23815 rows x 12 columns

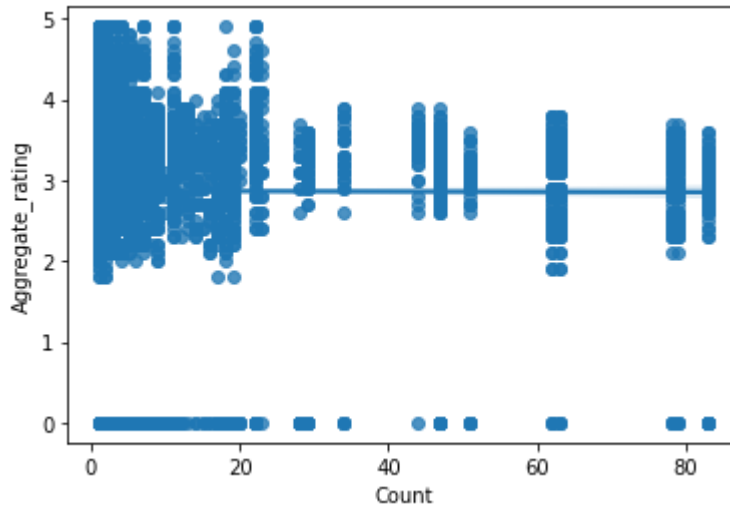


```
In [61]:
```

```
sns.regplot(x='Count',y='Aggregate_rating',data=rating)
rating[["Count", "Aggregate_rating"]].corr()
#Number of cuisines is not a good factor to decide the rating of a restaurant
```

Out[61]:

	Count	Aggregate_rating
Count	1.000000	-0.001642
Aggregate_rating	-0.001642	1.000000

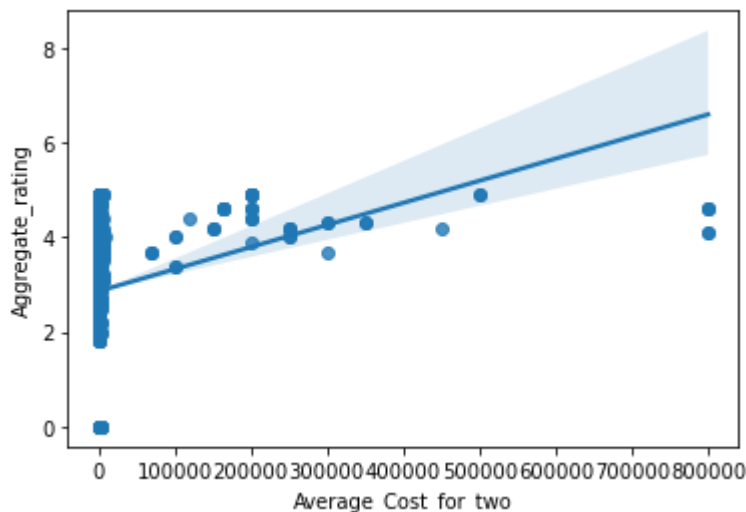


In [62]:

```
sns.regplot(x='Average_Cost_for_two',y='Aggregate_rating',data=rating)
rating[["Average_Cost_for_two", "Aggregate_rating"]].corr()
#Average cost for two is a weak positive factor to decide the rating of a restaurant
```

Out[62]:

	Average_Cost_for_two	Aggregate_rating
Average_Cost_for_two	1.00000	0.05011
Aggregate_rating	0.05011	1.00000

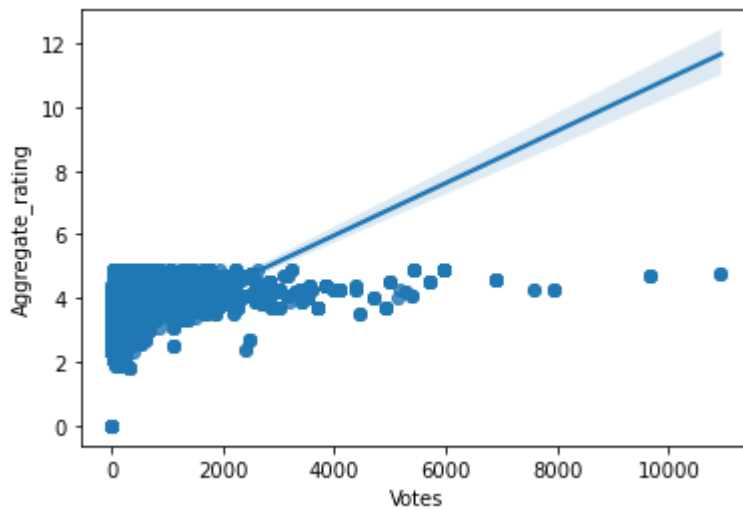


In [63]:

```
sns.regplot(x='Votes',y='Aggregate_rating',data=rating)
rating[["Votes", 'Aggregate_rating']].corr()
##Average cost for two can be a factor to decide the rating of a restaurant
```

Out[63]:

	Votes	Aggregate_rating
Votes	1.000000	0.318667
Aggregate_rating	0.318667	1.000000

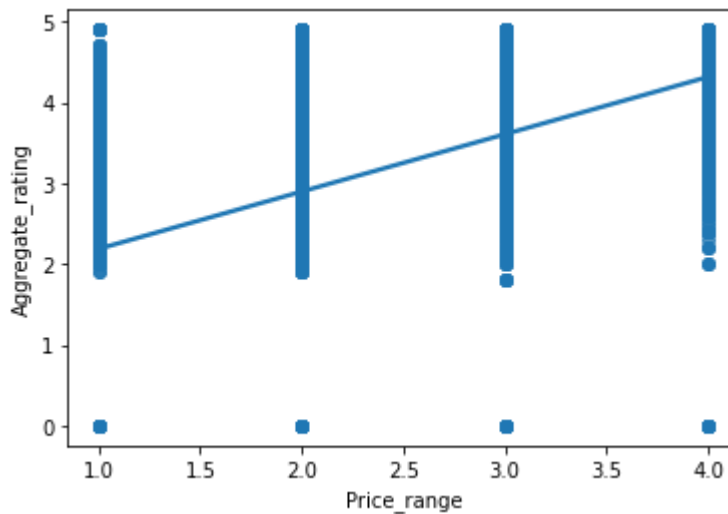


In [64]:

```
abc = df_rest1[df_rest1['Has_Online_delivery_Yes'] == 1]['Aggregate_rating'].mean()
xyz = df_rest1[df_rest1['Has_Online_delivery_Yes'] == 0]['Aggregate_rating'].mean()
sns.regplot(x='Price_range',y='Aggregate_rating',data=rating)
rating[['Price_range','Aggregate_rating']].corr()
##Price range can be a factor to decide the rating of a restaurant
```

Out[64]:

	Price_range	Aggregate_rating
Price_range	1.000000	0.462983
Aggregate_rating	0.462983	1.000000



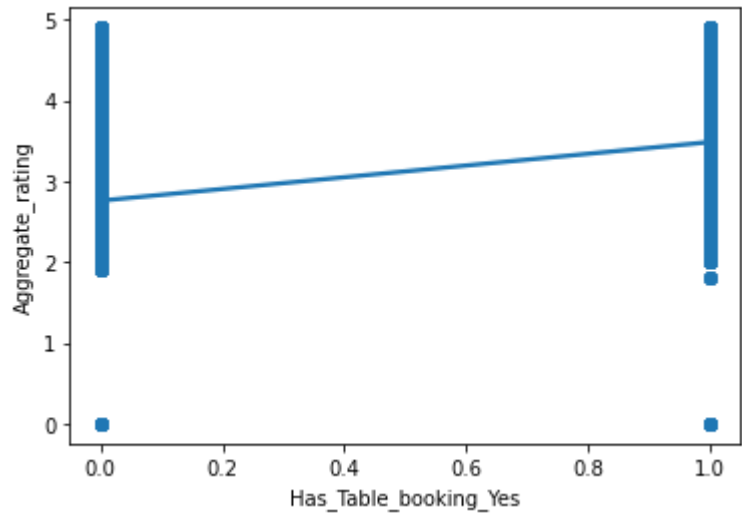
In [65]:

```
sns.regplot(x='Has_Table_booking_Yes',y='Aggregate_rating',data=rating)
rating[['Has_Table_booking_Yes','Aggregate_rating']].corr()
##Table booking can be a factor to decide the rating of a restaurant
```

Out[65]:

	Has_Table_booking_Yes	Aggregate_rating
--	-----------------------	------------------

	Has_Table_booking_Yes	Aggregate_rating
Has_Table_booking_Yes	1.000000	0.181843
Aggregate_rating	0.181843	1.000000



In []: