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

In [2]:
```
#1. Loading the data file using Pandas
df = pd.read_excel("restaurant_data.xlsx")
df.head()
```

Out[2]:

	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 [3]:
```
cntry_code = pd.read_excel("Country-Code.xlsx")
cntry_code.head()
```

Out[3]:

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

In [4]:
```
#Using merge function to merge the 2 files using Country Code
df_new = pd.merge(df, cntry_code,on='Country Code', how='left')
df_new.head()
```

Out[4]:

	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 [5]:
```
# Addinng profer format for col names and getting column info
df_new.columns = df_new.columns.str.replace(' ', '_')
df_new.columns
```

Out[5]:
```
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 [6]:
```
#Description of columns
df_new.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 [7]:
```
# 2. Check for null values in the data.
# Get the number of null values for each column
df_new.isnull().sum()
```

Out[7]:
```
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 [8]:
```
df_new[df_new['Restaurant_Name'].isnull()]
```

Out[8]:

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

In [9]:
```
#Dropping the row with no restaurant_name and resetting index
df_new.dropna(axis=0, subset=['Restaurant_Name'], inplace=True)
df_new.reset_index(drop=True, inplace=True)
```

In [10]:
```
df_new[df_new['Cuisines'].isnull()]
```

Out[10]:

	Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_Verbo

	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	Dahlonec Gainesvi
9085	17501439	Dovetail	216	Macon	543 Cherry St, Macon, GA 31201	Macon	Macon, Mac
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 Slappey 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 [11]:
```
#Marking records with Cusisines as NULL to 'Others'
df_new['Cuisines'].fillna('Others', inplace=True)
```

In [12]:
```
#checking for nulls and returning col info
df_new.isnull().sum
df_new.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 [13]:
```
# Saving the changes to excel file to use in Tableau for Dashboard creation
#Creating exel writer object
writer = pd.ExcelWriter('restaurant_output.xlsx')

#write dataframe to excel
df_new.to_excel(writer)

#save the excel
writer.save()
```

In [14]:
```
# Performing EDA
# EDA 1. Identifying count of restaurants by countries
ctry_res_count = df_new.groupby(['Country_Code','Country']).agg( Count= ('Restaurant_ID
ctry_res_count.sort_values(by='Count', ascending=False)
```

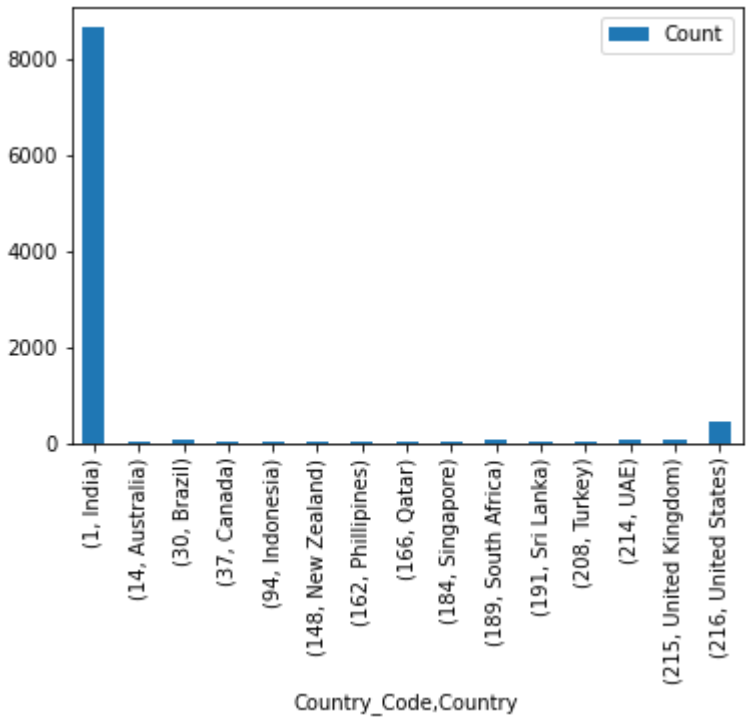
Out[14]:

		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

		Count
Country_Code	**Country**	
94	**Indonesia**	21
166	**Qatar**	20
184	**Singapore**	20
191	**Sri Lanka**	20
37	**Canada**	4

In [14]:
```
ctry_res_count.plot(kind='bar')
```

Out[14]: <AxesSubplot:xlabel='Country_Code,Country'>



In [15]:
```
# getting the restaurant count by city using group by on Country
city_res_count = df_new.groupby(['Country','City']).agg( Count= ('Restaurant_ID','count
city_res_count.sort_values(by='Count', ascending=False)
```

Out[15]:

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

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

141 rows × 1 columns

In [16]:
```
city_res_count.describe()
#The data seems to be skewed towards New Delhi, Gurgaon and Noida. There is minimal dat
# New Delhi has max restaurant has count = 5473
# city with min restaurant has count = 1
```

Out[16]:

	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 [49]:
```
min_res_cnt = city_res_count[city_res_count['Count']==1]
min_res_cnt
#  46 cities in different countries have 1 restaurant
```

Out[49]:

		Count
Country	**City**	
Australia	**Armidale**	1
	Balingup	1
	Beechworth	1
	Dicky Beach	1
	East Ballina	1
	Flaxton	1
	Forrest	1
	Huskisson	1

Country	City	Count
	Inverloch	1
	Lakes Entrance	1
	Lorn	1
	Macedon	1
	Mayfield	1
	Middleton Beach	1
	Montville	1
	Palm Cove	1
	Paynesville	1
	Penola	1
	Phillip Island	1
	Tanunda	1
	Trentham East	1
	Victor Harbor	1
Canada	Chatham-Kent	1
	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

	Count
Country	**City**
	Ojo Caliente 1
	Potrero 1
	Princeton 1
	Vernonia 1
	Weirton 1
	Winchester Bay 1

In [17]:
```
# EDA 2: Explore the franchise with most national presence
# Since majority count of restaurants is from India I am creating a new data set to ver

df_India = df_new.loc[df_new['Country'] == 'India']
df_India.head()
```

Out[17]:

	Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_Verbose
21	2701	Orient Express - Taj Palace Hotel	1	New Delhi	Taj Palace Hotel, Diplomatic Enclave, Chanakya...	The Taj Palace Hotel, Chanakyapuri	The Taj Palace Hotel, Chanakyapuri, New Delhi
22	309548	Tian - Asian Cuisine Studio - ITC Maurya	1	New Delhi	ITC Maurya, Diplomatic Enclave, Chanakyapuri, ...	ITC Maurya, Chanakyapuri	ITC Maurya, Chanakyapuri, New Delhi
23	2742	Bukhara - ITC Maurya	1	New Delhi	ITC Maurya, Chanakyapuri, New Delhi	ITC Maurya, Chanakyapuri	ITC Maurya, Chanakyapuri, New Delhi
25	301523	Nostalgia at 1911 Brasserie - The Imperial	1	New Delhi	The Imperial, Janpath, New Delhi	The Imperial, Janpath	The Imperial, Janpath, New Delhi
26	2724	1911 - The Imperial	1	New Delhi	The Imperial, Janpath, New Delhi	The Imperial, Janpath	The Imperial, Janpath, New Delhi

In [18]:
```
#Identifying restauranct franchise with national presence

res_count_India = df_India.groupby(['Restaurant_Name']).agg( Count= ('Restaurant_ID','c
res_count_India.sort_values(by='Count', ascending=False)

# Cafe Coffee Day has most restaurants in India
```

Out[18]:

	Count

Restaurant_Name	Count
Restaurant_Name	
Cafe Coffee Day	83
Domino's Pizza	79
Subway	63
Green Chick Chop	51
McDonald's	48
...	...
Global Local	1
Glenz Cafe N Bakers	1
Glen's Bakehouse	1
Giri Momos Centre & Chinese Fast Food	1
{Niche} - Cafe & Bar	1

6603 rows × 1 columns

In [19]:
```
# Exploring data for Cafe Coffee Day since it has most national presence
#Average cost for 2 at Cafe Coffee Day restaurant (In Indian rupees)

avg_cost_for_2 = df_India[df_India['Restaurant_Name']=='Cafe Coffee Day']['Average_Cost
avg_cost_for_2
#Average cost for 2 people to eat at Coffe Cafe Day is 450 rupees which is pretty reaso
```

Out[19]:
```
450    83
Name: Average_Cost_for_two, dtype: int64
```

In [20]:
```
#Rating text for top franchise

ratings_for_top_fran = df_India[df_India['Restaurant_Name']=='Cafe Coffee Day']['Rating
ratings_for_top_fran
# Most of the restaurants for the franchise are rated Average, 16 are not rated
```

Out[20]:
```
Average      59
Not rated    16
Good          4
Poor          4
Name: Rating_text, dtype: int64
```

In [21]:
```
#Checking Price ranges for the franchise

price_range_for_fran = df_India[df_India['Restaurant_Name']=='Cafe Coffee Day']['Price_
price_range_for_fran.sort_values(ascending=False).head()

#All the restaurants under this franchise fall in price range 1
```

Out[21]:
```
1    83
Name: Price_range, dtype: int64
```

In [22]:
```
#Aggregate ratings for this franchise
```

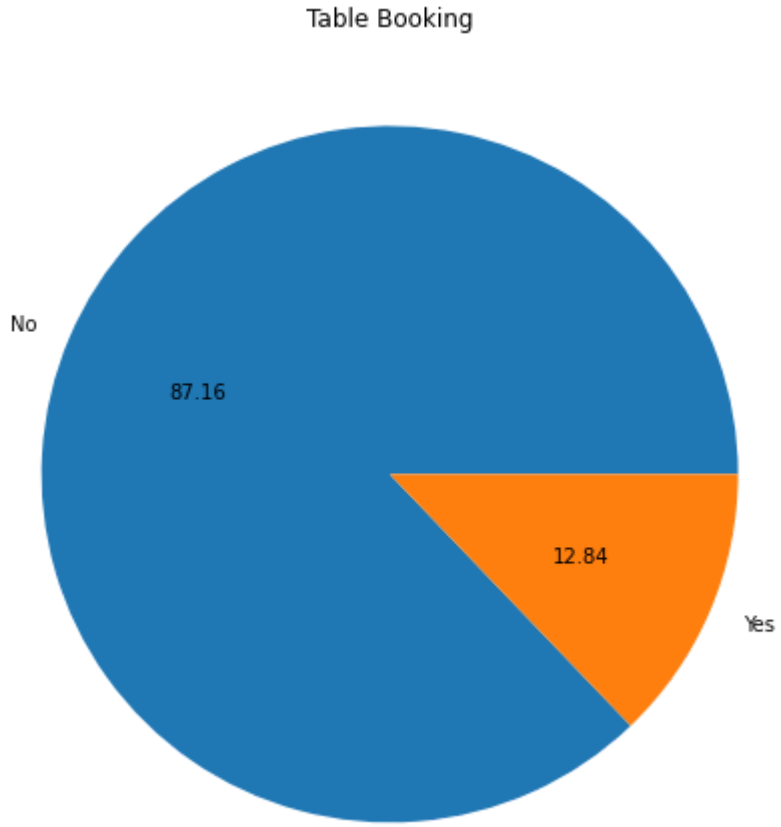
```
agg_rate_for_top_fran = df_India[df_India['Restaurant_Name']=='Cafe Coffee Day']['Aggre
agg_rate_for_top_fran

# There is no agg_rating for 16 restaurants for Cafe Coffee Day franchise
# Highest rating is 3.6 and 2 of them have this
```

Out[22]:
```
0.0    16
3.1    10
2.7     7
3.2     7
3.0     7
2.8     6
3.3     6
2.9     6
3.4     5
2.6     4
2.3     2
3.6     2
2.4     2
3.5     2
2.5     1
Name: Aggregate_rating, dtype: int64
```

In [23]:
```
# EDA 3: Ratio between restaurants that allow table booking vs. those that do not allow

booking_yes = df_India['Has_Table_booking'].value_counts()['Yes']
booking_no = df_India['Has_Table_booking'].value_counts()['No']
print('Ratio between restaurants that allow table booking vs. those that do not allow t
      round((booking_yes/booking_no),2))

# Ratio is 0.15, means lot of the resturants doesn't allow table booking
```

```
Ratio between restaurants that allow table booking vs. those that do not allow table boo
king:  0.15
```

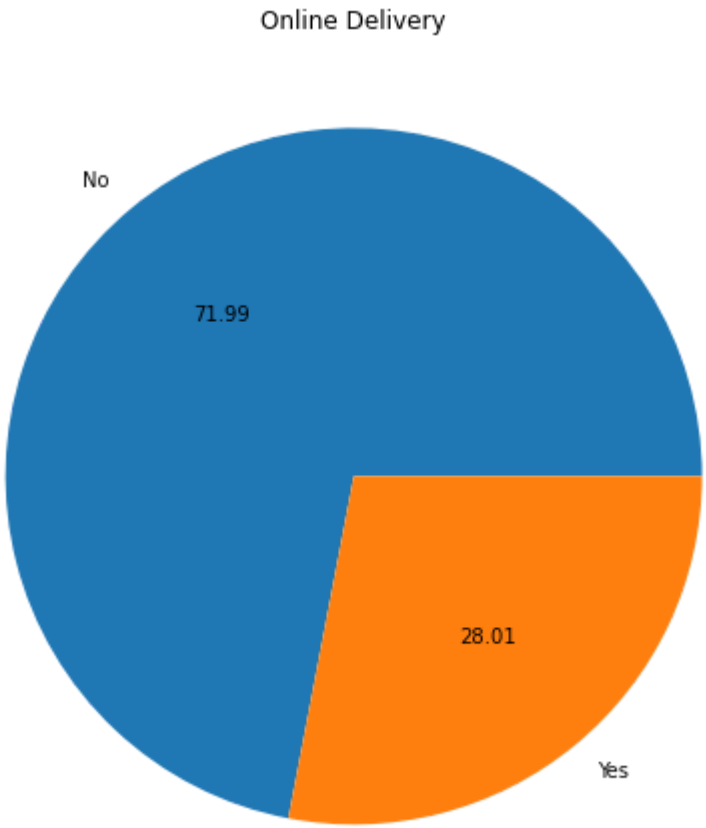
In [24]:
```
#Pie chart showing percentage difference for table booking Yes/No

plt.figure(figsize=(10,8))
x = df_India.Has_Table_booking.value_counts()
y = df_India['Has_Table_booking'].value_counts().index
plt.pie(x, labels=y, autopct='%.2f')
plt.title('Table Booking')
plt.show()
```



In [25]:
```
# EDA 4: Percentage of restaurants providing online delivery
online_delivery_yes =df_India['Has_Online_delivery'].value_counts()['Yes']
print('Percentage of restaurants providing online delivery : {} %'.format((round(online

# 28% percent of restaurants allow online delivery
```

```
Percentage of restaurants providing online delivery : 28.000000000000004 %
```

In [114…
```
#Pie chart showing online delivery yes/no

plt.figure(figsize=(10,8))
x = df_India.Has_Online_delivery.value_counts()
y = df_India['Has_Online_delivery'].value_counts().index
plt.pie(x, labels=y, autopct='%.2f')
plt.title('Online Delivery')
plt.show()
```



In [26]:
```
# EDA 5: Difference in number of votes for the restaurants that deliver and the restaur

res_deliver = df_India[df_India['Has_Online_delivery'] == "Yes"]['Votes'].sum()
print(res_deliver)
res_no_deliver =  df_India[df_India['Has_Online_delivery'] == "No"]['Votes'].sum()
print(res_no_deliver)
print('Difference in number of votes for restaurants that deliver and dont deliver: ',a

#Difference in number of votes is coming to 173166
```

```
506614
679780
Difference in number of votes for restaurants that deliver and dont deliver:  173166
```

In [27]:
```
# EDA 6: Top 10 cuisines served across cities
# What is the maximum and minimum number of cuisines that a restaurant serves?
# Which is the most served cuisine across the restaurant for each city?

#Cuisine has multiple entries combined hence splitting them and adding the entries to l
total_cuisines=[]
k = df_India[df_India['Cuisines'].notnull()]
k['Cuisines'] = k['Cuisines'].apply(lambda x:x.strip())
for i in k['Cuisines']:
    for j in i.split(','):
        j = j.strip()
        total_cuisines.append(j)
```

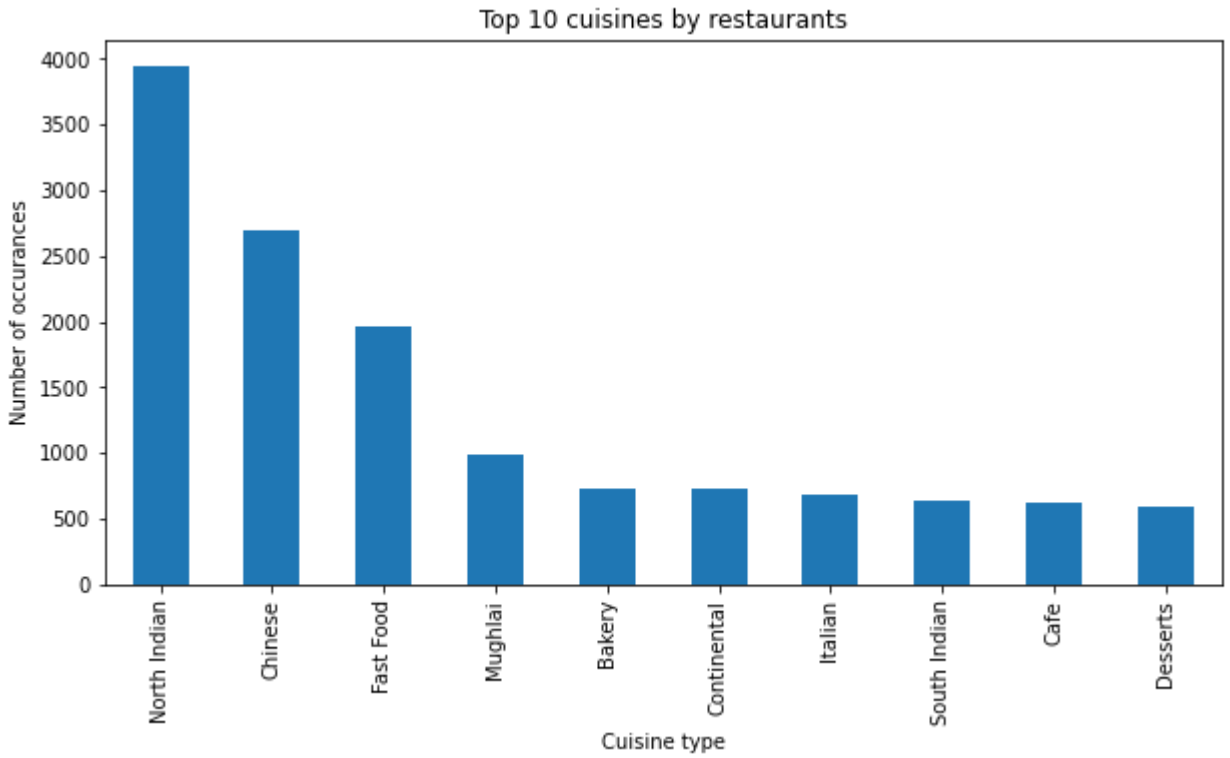
In [28]:
```
# Getting top 10 cuisines from the list
top_10_cuisines=pd.Series(total_cuisines).value_counts()[:10]
top_10_cuisines
```

```
# North Indian, Chinese, Fast Food, Mughlai, Bakery, Continental, Italian, South Indian
```

Out[28]:
```
North Indian    3945
Chinese         2690
Fast Food       1963
Mughlai          992
Bakery           726
Continental      723
Italian          681
South Indian     631
Cafe             627
Desserts         597
dtype: int64
```

In [29]:
```
plt.figure(figsize=(10,5))
a=pd.Series(total_cuisines).value_counts()[:10]
a.plot(kind='bar')
plt.title('Top 10 cuisines by restaurants')
plt.xlabel('Cuisine type')
plt.ylabel('Number of occurances')
```

Out[29]:
```
Text(0, 0.5, 'Number of occurances')
```



In [30]:
```
# Which is the most served cuisine across the restaurant for each city?

#Creating data frame for restaurants with the cuisine list

df_2 = df_India[df_India['Cuisines'].isin(total_cuisines)]
df_2
```

Out[30]:

	Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_V

	Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_V
21	2701	Orient Express - Taj Palace Hotel	1	New Delhi	Taj Palace Hotel, Diplomatic Enclave, Chanakya...	The Taj Palace Hotel, Chanakyapuri	The Taj Chanak Ne
23	2742	Bukhara - ITC Maurya	1	New Delhi	ITC Maurya, Chanakyapuri, New Delhi	ITC Maurya, Chanakyapuri	ITC M Chanak Ne
30	2689	House of Ming - The Taj Mahal Hotel	1	New Delhi	The Taj Mahal Hotel, 1, Mansingh Road, New Delhi	The Taj Mahal Hotel, Mansingh Road	The Taj Hotel, Ma Road, Ne
32	2443	Wildfire - Crowne Plaza	1	Gurgaon	Crowne Plaza, National Highway 8, Sector 29, G...	Crowne Plaza, Sector 29	Crown Se G
34	18345728	Masala Library	1	New Delhi	21 A, Janpath, New Delhi	Janpath	Janpat
...	...	...	...	...	...	...	
9003	302835	Aggarwal Jalebi Wale	1	New Delhi	Opposite Avtaar Dhaba, Milap Nagar, Uttam Naga...	Uttam Nagar	Uttam Ne
9005	18424202	Famous Parantha and Poori Sabzi	1	New Delhi	44, Gali Number 1, Block A, New Ashok Nagar, V...	Vasundhara Enclave	Vasu Enclav
9006	18372694	Sweets n Treats	1	New Delhi	Main Market, New Ashok Nagar, Vasundhara Encla...	Vasundhara Enclave	Vasu Enclav
9007	18435795	Indian Special Hot Momos	1	Noida	Ashirwad Complex, Sector 53, Noida	Sector 53	Sector 53
9548	3900245	Deena Chat Bhandar	1	Varanasi	D-47/184, Luxa Road, Dashaswmedh Road, Varanasi	Dashaswmedh Road	Dashas Road, V

3097 rows × 20 columns

In [31]:
```
# Which is the most served cuisine across the restaurant for each city?

most_serv =pd.DataFrame(df_2.groupby(by=['Restaurant_Name','City','Cuisines']).size().r
```

```
most_serv.columns=['Restaurant_Name','City','Cuisines','Count']
most_serv.sort_values('Count',ascending=False).head()

# Cafe Coffee Day in New Delhi has most cuisines served which is 57
# Least number Cuisine served is 1
```

Out[31]:

	Restaurant_Name	City	Cuisines	Count
396	Cafe Coffee Day	New Delhi	Cafe	57
1259	Keventers	New Delhi	Beverages	24
270	Baskin Robbins	New Delhi	Ice Cream	13
397	Cafe Coffee Day	Noida	Cafe	13
395	Cafe Coffee Day	Gurgaon	Cafe	11

In [32]:

```
# What is the maximum and minimum number of cuisines that a restaurant serves?
data = df_India  # Making copy of the dataframe

cuis_cnt = data.Cuisines.apply(lambda x: pd.Series(str(x).split(",")))
cuis_cnt.head(20)

# Splitting the 'Cuisines' column is divided into 8 different cols depending on number
# Max number of cuisines restaurant serves is 8 and min number if 1 based on the split
```

Out[32]:

	0	1	2	3	4	5	6	7
21	European	NaN	NaN	NaN	NaN	NaN	NaN	NaN
22	Asian	Japanese	Korean	Thai	Chinese	NaN	NaN	NaN
23	North Indian	NaN	NaN	NaN	NaN	NaN	NaN	NaN
25	European	Continental	NaN	NaN	NaN	NaN	NaN	NaN
26	North Indian	Chinese	South Indian	Italian	NaN	NaN	NaN	NaN
27	Malaysian	Thai	Kerala	Vietnamese	Sri Lankan	NaN	NaN	NaN
28	Japanese	Sushi	NaN	NaN	NaN	NaN	NaN	NaN
29	Japanese	Sushi	NaN	NaN	NaN	NaN	NaN	NaN
30	Chinese	NaN	NaN	NaN	NaN	NaN	NaN	NaN
31	Continental	North Indian	Italian	Asian	NaN	NaN	NaN	NaN
32	South American	NaN	NaN	NaN	NaN	NaN	NaN	NaN
33	Chinese	Italian	Continental	North Indian	NaN	NaN	NaN	NaN
34	Modern Indian	NaN	NaN	NaN	NaN	NaN	NaN	NaN
35	Chinese	NaN	NaN	NaN	NaN	NaN	NaN	NaN
36	Chinese	NaN	NaN	NaN	NaN	NaN	NaN	NaN
37	Seafood	Italian	NaN	NaN	NaN	NaN	NaN	NaN
38	Chinese	NaN	NaN	NaN	NaN	NaN	NaN	NaN
39	Italian	NaN	NaN	NaN	NaN	NaN	NaN	NaN

	0	1	2	3	4	5	6	7
40	Finger Food	NaN	NaN	NaN	NaN	NaN	NaN	NaN
41	French	Italian	NaN	NaN	NaN	NaN	NaN	NaN

In [34]:
```
# Checking rating text count for Indian restaurants
df_India['Rating_text'].value_counts()

# Majority of restaurants are either Average of Not rated in India
# There are only 116 restaurants that are excellent and 691 very good
#Ordering online food is still gaining momentum hence most of the restaurants are still
#unrated as people might be preferring to visiting the restaurant for a meal.
```

Out[34]:
```
Average      3678
Not rated    2139
Good         1847
Very Good     691
Poor          180
Excellent     116
Name: Rating_text, dtype: int64
```

In [35]:
```
# Verifying cities with high rated restaurants

df_r = df_India.loc[(df_India['Rating_text']== 'Excellent') | (df_India['Rating_text']=
df_r['City'].value_counts().head()

# New Delhi has high number of restaurants that are excellent rating
```

Out[35]:
```
New Delhi    328
Gurgaon       95
Noida         29
Pune          18
Bangalore     18
Name: City, dtype: int64
```

In [49]:
```
# New Delhi has most of these so checking cuisines for New Delhi
delhi_excel = df_r[df_r['City']=='New Delhi']
delhi_excel.head()

# Cuisines for the restaurants vary
# Ratings for these vary from 4.0 to 4.9
# Not all of them have table booking or Online delivery so we can conclude that these a
```

Out[49]:

	Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_Verbose
21	2701	Orient Express - Taj Palace Hotel	1	New Delhi	Taj Palace Hotel, Diplomatic Enclave, Chanakya...	The Taj Palace Hotel, Chanakyapuri	The Taj Palace Hotel, Chanakyapuri, New Delhi
22	309548	Tian - Asian Cuisine Studio - ITC Maurya	1	New Delhi	ITC Maurya, Diplomatic Enclave, Chanakyapuri, ...	ITC Maurya, Chanakyapuri	ITC Maurya, Chanakyapuri, New Delhi
23	2742	Bukhara - ITC Maurya	1	New Delhi	ITC Maurya, Chanakyapuri, New Delhi	ITC Maurya, Chanakyapuri	ITC Maurya, Chanakyapuri, New Delhi

	Restaurant_ID	Restaurant_Name	Country_Code	City	Address	Locality	Locality_Verbose
27	2725	The Spice Route - The Imperial	1	New Delhi	The Imperial, Janpath, New Delhi	The Imperial, Janpath	The Imperial, Janpath, New Delhi
30	2689	House of Ming - The Taj Mahal Hotel	1	New Delhi	The Taj Mahal Hotel, 1, Mansingh Road, New Delhi	The Taj Mahal Hotel, Mansingh Road	The Taj Mahal Hotel, Mansingh Road, New Delhi

In [50]:
```
# Cuisines served at high rated restaurants
delhi_excel = df_r[df_r['City']=='New Delhi']
delhi_excel['Cuisines'].head(20)
```

Out[50]:
```
21                                              European
22              Asian, Japanese, Korean, Thai, Chinese
23                                           North Indian
27     Malaysian, Thai, Kerala, Vietnamese, Sri Lankan
30                                             Chinese
34                                        Modern Indian
44                                             Chinese
48                                             Chinese
51         Continental, European, North Indian, French
52                                     Spanish, Italian
57                                 Seafood, North Indian
67                                        Modern Indian
70                                         South Indian
76    North Indian, Continental, European, Chinese, ...
85                      European, Italian, North Indian
91                                           North Indian
97                      European, Italian, North Indian
118                               Mughlai, Street Food
120                                            Italian
129                                               Thai
Name: Cuisines, dtype: object
```

In [51]:
```
delhi_excel['Price_range'].value_counts()

#Most of the restaurants with excellent and very good ratings are in price range
# 3 and 2.
```

Out[51]:
```
3    123
2     91
1     62
4     52
Name: Price_range, dtype: int64
```

In [52]:
```
delhi_excel['Locality'].value_counts()

# Restaurants with high ratings are in Connaught place, could be these are
# areas with educated population and provide feedback more often
```

Out[52]:
```
Connaught Place          32
Rajouri Garden           24
Greater Kailash (GK) 1   17
Satyaniketan             15
Khan Market              14
```

```
                              ..
Feroze Shah Road               1
Laxmi Nagar                    1
Paharganj                      1
Pragati Maidan                 1
Lajpat Nagar 4                 1
Name: Locality, Length: 94, dtype: int64
```

In [53]:
```
delhi_excel['Has_Online_delivery'].value_counts()
```

Out[53]:
```
No     181
Yes    147
Name: Has_Online_delivery, dtype: int64
```

In [54]:
```
delhi_excel['Has_Table_booking'].value_counts()
# Online delivery and table booking are not options for majority of these restaurants
```

Out[54]:
```
No     193
Yes    135
Name: Has_Table_booking, dtype: int64
```

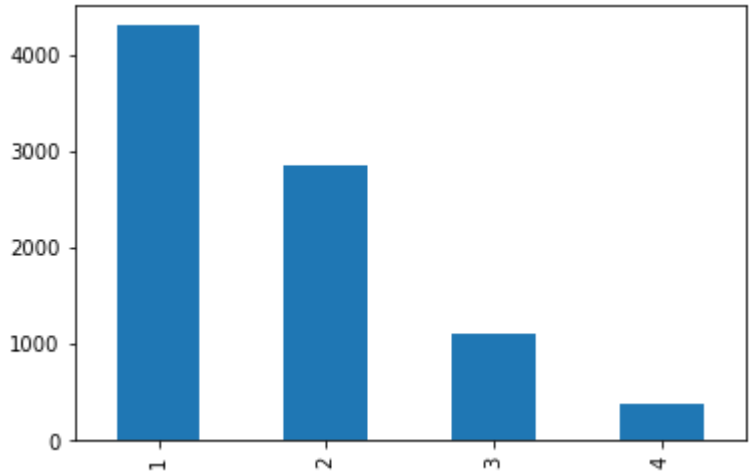
In [38]:
```
# EDA 7: What is the distribution cost across the restaurants?

df_India.describe()

# Following inferences from the stats:
# Max price range is 4 for the data set and min is 1 and mean is 1.72
# Max Aggregate_rating is 4.9 and min rating is 0
# Max Votes that restaurant got are 10934 and min Votes is 0. Avg votes are 137
# Max for Average cost for two is 8000 (INR) and average is 623
```

Out[38]:

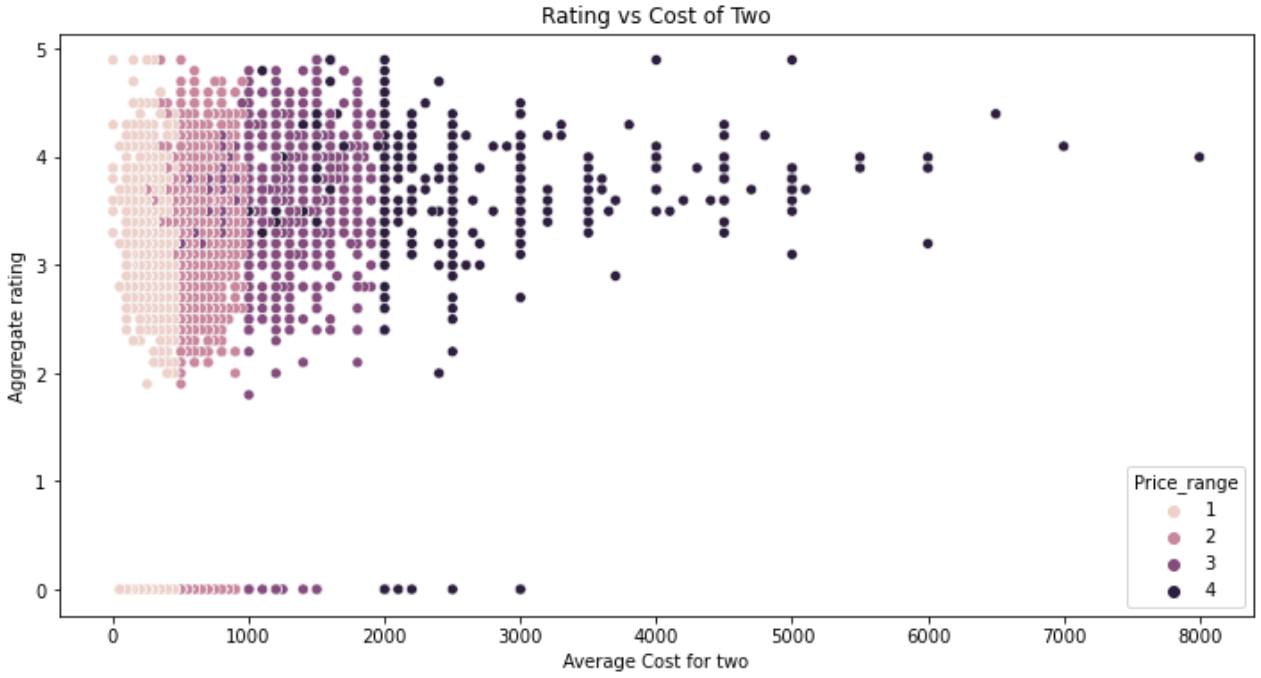
	Restaurant_ID	Country_Code	Longitude	Latitude	Average_Cost_for_two	Price_range	Agg
count	8.651000e+03	8651.0	8651.000000	8651.000000	8651.000000	8651.000000	
mean	8.658073e+06	1.0	72.856827	26.421043	623.349902	1.721535	
std	8.959484e+06	0.0	18.018577	6.976885	595.707528	0.852482	
min	5.300000e+01	1.0	0.000000	0.000000	0.000000	1.000000	
25%	3.007465e+05	1.0	77.098916	28.491474	300.000000	1.000000	
50%	2.200011e+06	1.0	77.203469	28.569889	450.000000	2.000000	
75%	1.836120e+07	1.0	77.285357	28.637870	700.000000	2.000000	
max	1.850065e+07	1.0	91.806493	35.000000	8000.000000	4.000000	

In [39]:
```
df_India['Price_range'].value_counts().plot(kind='bar')

# Bar graph for Price_range shows that >4000 restaurants are in price range 1
# About <3000 restaurants fall in price range 2
# About 1200 fall in price range 3
# Between 0 - 200 fall in price range 4
```

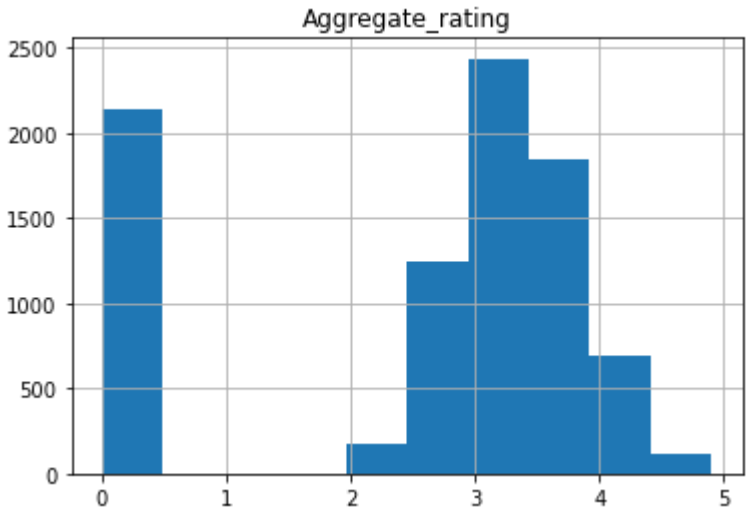
Out[39]:
```
<AxesSubplot:>
```



In [40]:
```
plt.figure(figsize=(12,6))
sns.scatterplot(x="Average_Cost_for_two", y="Aggregate_rating", hue='Price_range', data

plt.xlabel("Average Cost for two")
plt.ylabel("Aggregate rating")
plt.title('Rating vs Cost of Two');

#I observe that there is no linear relation between price and rating. For instance,
#most of the Restaurants with good rating (like 4-5) seem to be in 1000 or less average
# that rating will be higher if cost is more
#Restaurants with good rating (like 4-5) have restaurants with all price ranges and spr
```



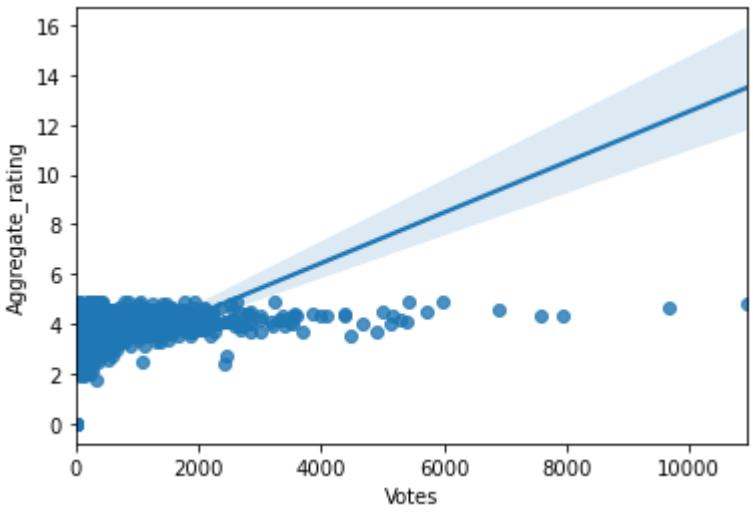
In [41]:
```
# EDA 8: How ratings are distributed among various factors?

hist = df_India.hist(column=['Aggregate_rating'])

# From the histogram we can analyze that there are >2000 restaurants that have 0 rating
# Close to 2500 restaurants are between 3 - 3.5 rating
# Around 700 restaurants have > 4 rating and very few are in 5 rating range
```



In [42]:
```
sns.regplot(x='Votes',y='Aggregate_rating',data=df_India)
df_India[['Votes', 'Aggregate_rating']].corr()

#We can see that restaurants with 4.9-5 rating have votes between 0-2000.
# Corr 0.28 indicates that there is moderate positive linear relationship
```

Out[42]:

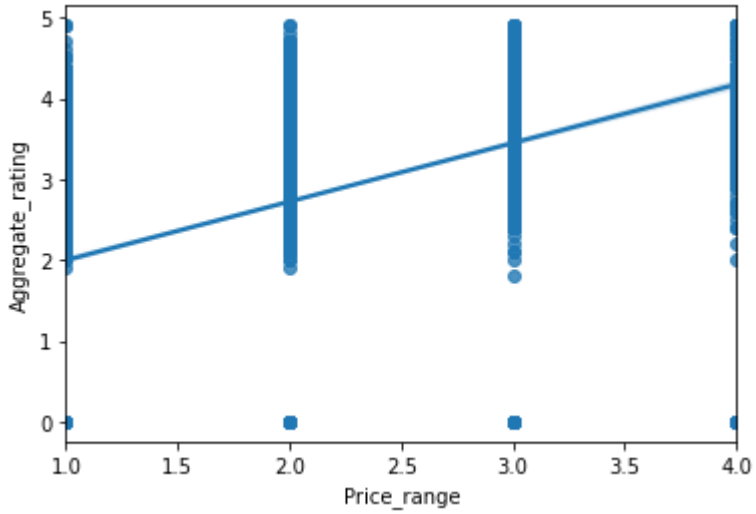
	Votes	Aggregate_rating
Votes	1.000000	0.287569
Aggregate_rating	0.287569	1.000000



In [43]:
```
sns.regplot(x='Price_range',y='Aggregate_rating',data=df_India)
df_India[['Price_range', 'Aggregate_rating']].corr()
```

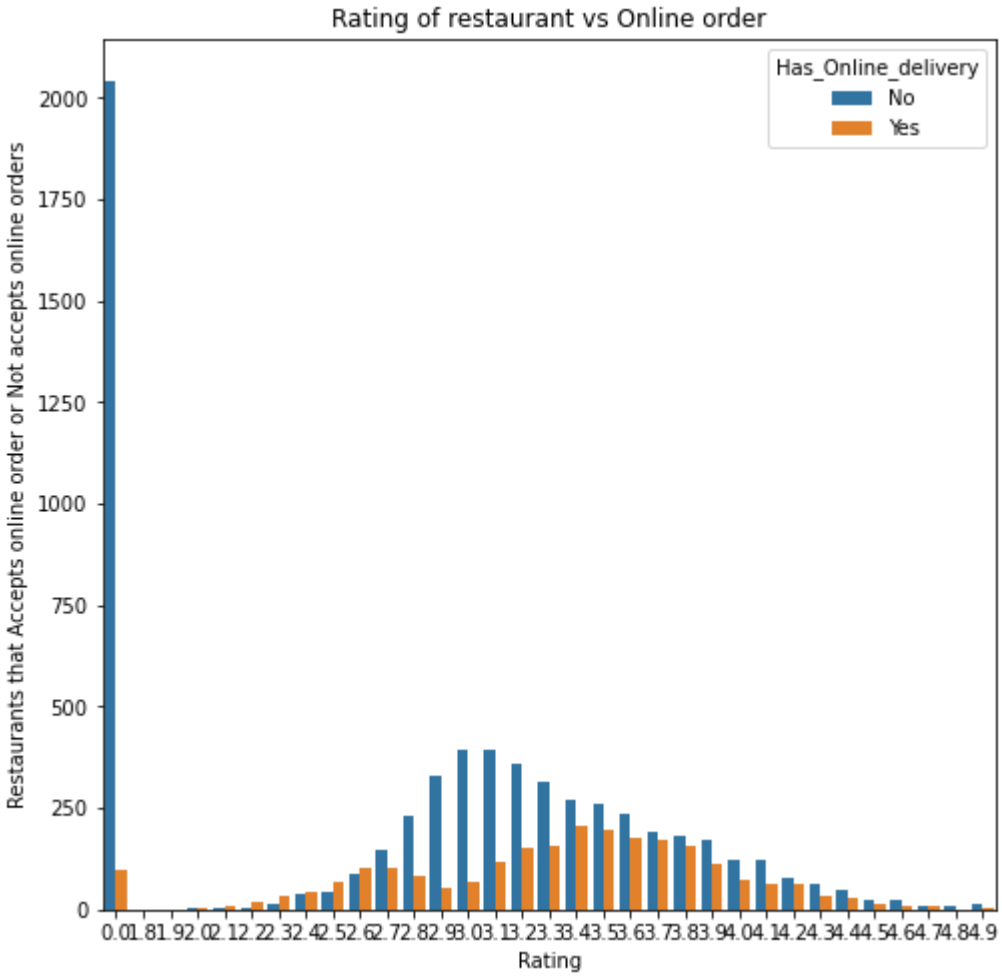
Out[43]:

	Price_range	Aggregate_rating
Price_range	1.000000	0.409019
Aggregate_rating	0.409019	1.000000



In [44]:
```
plt.figure(figsize = (8,8))
sns.countplot(x=df_India['Aggregate_rating'], hue = df_India['Has_Online_delivery'])
plt.ylabel("Restaurants that Accepts online order or Not accepts online orders")
plt.xlabel('Rating')
plt.title("Rating of restaurant vs Online order")
# Highest rating restaurants doesn't seem to have online orders so we can't tell if onl
# a restaurant
```

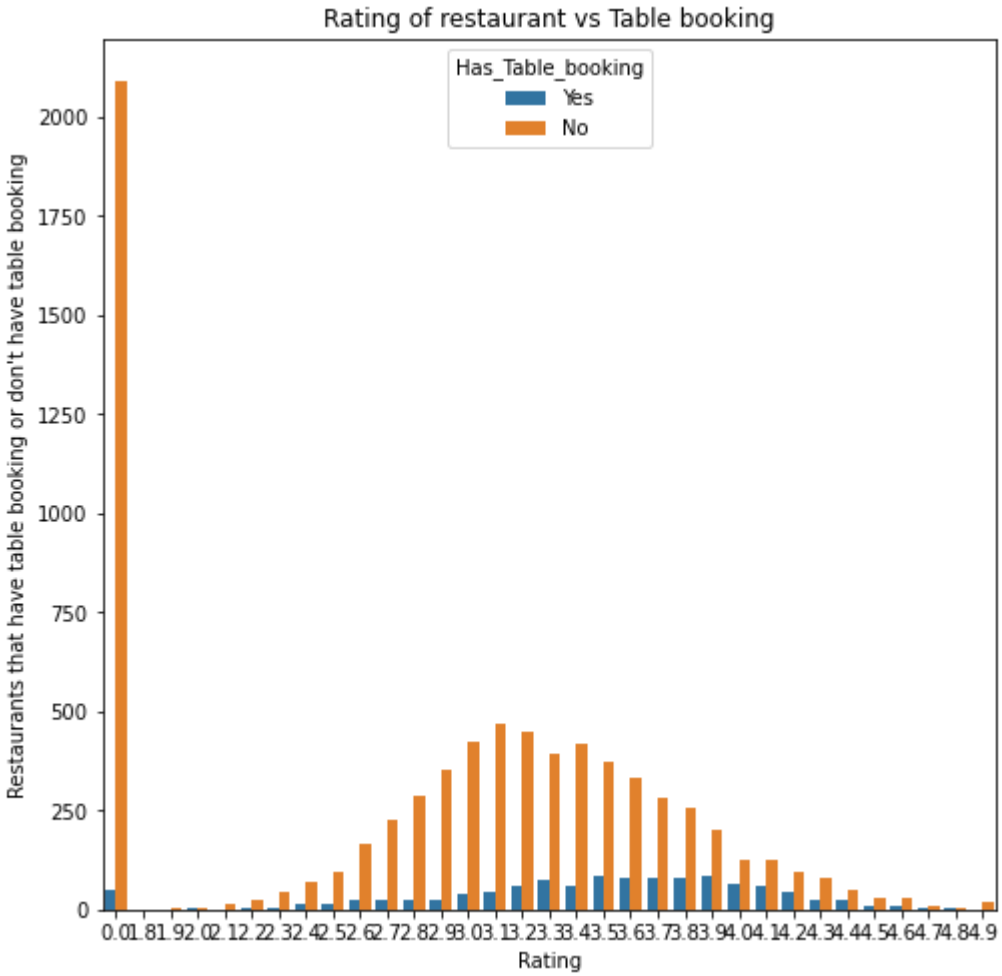
Out[44]: Text(0.5, 1.0, 'Rating of restaurant vs Online order')



In [45]:
```
plt.figure(figsize = (8,8))
```

```
sns.countplot(x=df_India['Aggregate_rating'], hue = df_India['Has_Table_booking'])
plt.ylabel("Restaurants that have table booking or don't have table booking")
plt.xlabel('Rating')
plt.title("Rating of restaurant vs Table booking")
# Restaurants that have table booking have ratings between 2.3 - 4.4
# Restaurants with 4.9 rating have less table booking
# We can say that table booking has less influence with rating of the restaurant
```

Out[45]: Text(0.5, 1.0, 'Rating of restaurant vs Table booking')



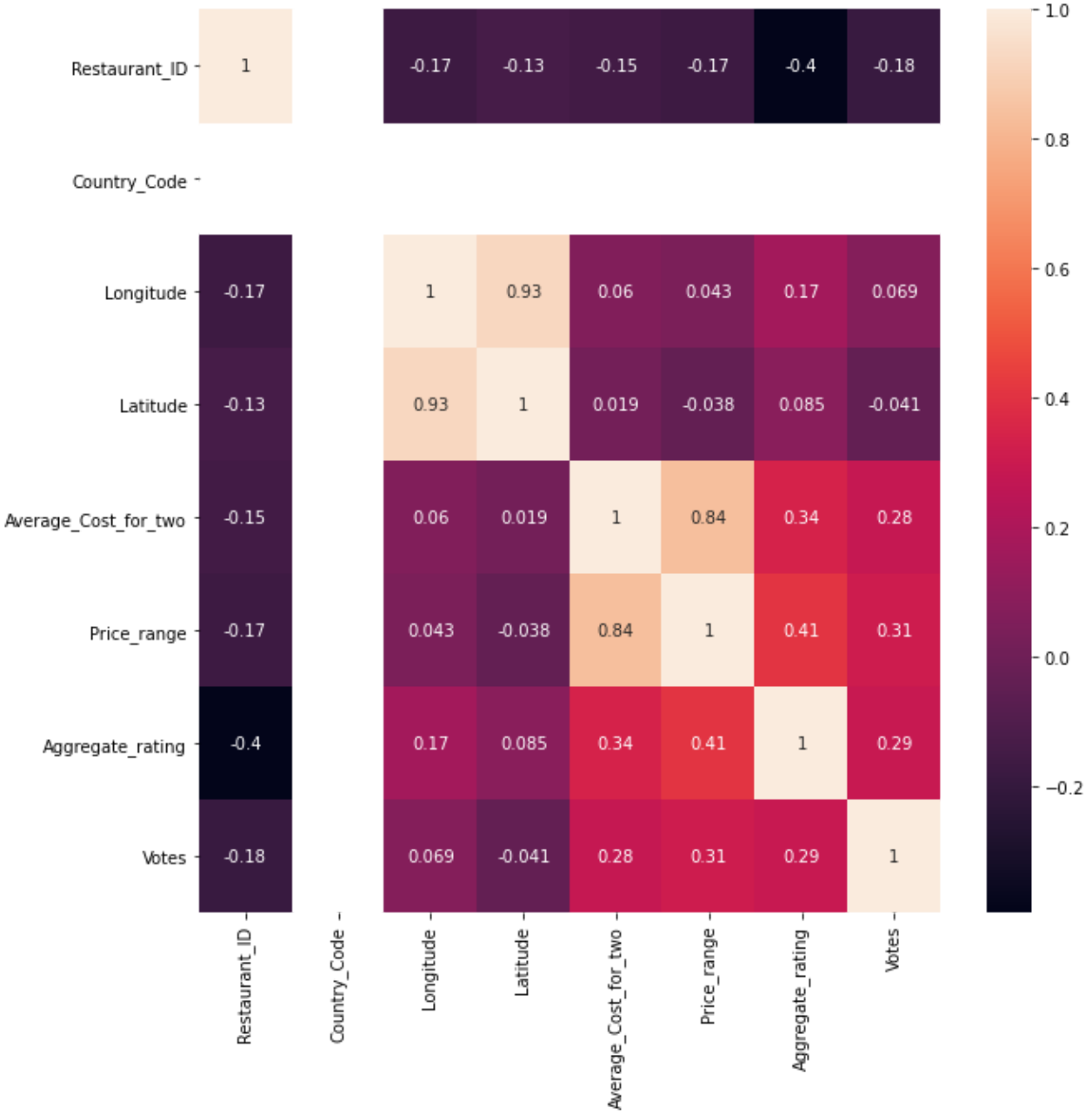
In [46]:
```
#correlation, relationship between variable attributes
df_India.corr()
```

Out[46]:

	Restaurant_ID	Country_Code	Longitude	Latitude	Average_Cost_for_two	Price_r
Restaurant_ID	1.000000	NaN	-0.168828	-0.134928	-0.151713	-0.16
Country_Code	NaN	NaN	NaN	NaN	NaN	
Longitude	-0.168828	NaN	1.000000	0.926780	0.060101	0.04
Latitude	-0.134928	NaN	0.926780	1.000000	0.018621	-0.03
Average_Cost_for_two	-0.151713	NaN	0.060101	0.018621	1.000000	0.83
Price_range	-0.167194	NaN	0.042803	-0.037750	0.838026	1.00
Aggregate_rating	-0.395272	NaN	0.171342	0.085225	0.344159	0.40
Votes	-0.183479	NaN	0.069104	-0.040849	0.281751	0.3

In [47]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(df_India.corr(), annot=True)
```

Out[47]: <AxesSubplot:>



In [48]:
```
# EDA 9: Explain the factors in the data that may have an effect on ratings. For exampl
# number of cuisines, cost, delivery option, etc.

#We can conclude that there is no single variable that affects the rating strongly,
# however table booking,online delivery,avg price for two and price range, number of vo
# partially play role affecting the rating of a restaurant.
# Correlation is 0.40 for Price range and Aggregate rating which is little higher than
# poeple who visit these restaurants may be educated folks and can afford the cost and
# People who visit low price restaurants don't care about ratings. Social media presenc
# ratings to go up for these.
```

In [ ]: