Swiggy Dataset Analysis

This project is an analysis of Swiggy (a food delivery app).
This dataset includes information of 50 metro cities and all the different restaurants they have.
This Dataset involves 31804 Rows and 10 Columns having shape of (31804, 10) , it contains information about different restaurants like their name, rating, veg/non-veg, cost for two, category and more.
The dataset was available on kaggle. In this project we are trying to get all the inferences and information from the data and giving results to users about which is best restaurants has less cost and high rating and many more.
In this Project we are using python and its libraries, for data analysis and computation we will use pandas and numpy libraries and for data visualization we will use matplotlib and seaborn libraries of python.
I have gained all the above mentioned skills from the course "Data Analysis with Python: Zero to Pandas" offered by Jovian which is a free certified course.

```
from urllib.request import urlretrieve
import matplotlib.pyplot as plt
```

```
urlretrieve('https://entrackr-bucket.s3.ap-south-1.amazonaws.com/wp-content/uploads/202
```

Here we are downloading the image in our jupyter notebook.

```
from PIL import Image
```

```
img = Image.open('Swiggy.jpg')
```

This is used to open image in our jupyter notebook.

```
plt.figure(figsize=(20, 10))
plt.grid(False)
plt.axis('off')
plt.imshow(img);
```

Downloading the Dataset

I have already downloaded the data from Kaggle (csv format) in my system and now I will uploaded it on jupyter notebook, hence we can directly access the file using pandas read_csv command.

```
import pandas as pd
import numpy as np
```

```
import seaborn as sns
%matplotlib inline
```

First we are going to import all the required Python libraries for data computations and visualization that are:-

- Pandas
- Numpy
- Matplotlib
- Seaborn
- Image
- urlretrieve

```
project_name = "swiggy-data-analysis"
```

```
!pip install jovian --upgrade -q
```

```
import jovian
```

```
jovian.commit(project='swiggy-data-analysis')
```

```
[jovian] Updating notebook "anujasthana2001/swiggy-data-analysis" on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/anujasthana2001/swiggy-data-analysis
```

'https://jovian.ai/anujasthana2001/swiggy-data-analysis'

Data Preparation and Cleaning

Here we will load the data in juypter notebook using read_csv command of pandas, get a brief view on dataset, the shape and size of dataset and perform any required functions to improve the overall dataset that may include adding or deleting columns or removing null values.

```
data_df=pd.read_csv("Swiggy_50.csv")
```

The dataset is now loaded as a data frame using pandas read_csv command.

```
data_df.head(10)
```

	Restaurant_Name	Category	Rating	Cost_for_two	Veg	city	Area	
0	KFC	American,Snacks,Biryani	3.9	400	False	Delhi	Paharganj	P
1	McDonald's	American	4.3	400	False	Delhi	Kashmere Gate	[D
2	Haldiram's	Sweets,Snacks,North Indian	4.0	350	True	Delhi	Chandni Chowk	
3	Chai Point	Bakery,Beverages,Maharashtrian,Snacks,Street F...	4.2	150	False	Delhi	Connaught Place	Cc
4	Bikanervala Chandni Chowk	Street Food,Sweets	4.1	400	False	Delhi	Old Delhi	
5	La Pino'z Pizza	Italian,Pizzas,Fast Food,Mexican,Desserts,Beve...	3.7	250	False	Delhi	Kamla Nagar	
6	Burger King	American	4.2	350	False	Delhi	Connaught Place	N
7	Chaayos Chai+Snacks=Relax	Bakery,Beverages,Chaat,Desserts,Home Food,Ital...	4.2	250	False	Delhi	Kashmere Gate	K
8	Capten Milkshakes	Beverages	NaN	300	True	Delhi	Chandni Chowk	(

	Restaurant_Name	Category	Rating	Cost_for_two	Veg	city	Area
9	The Brijwasi Restaurant	North Indian,South Indian,Continental,Chinese	NaN	300	False	Delhi	Chandni Chowk

```
data_df.shape
```

```
(31804, 10)
```

Shape shows the number of rows and columns a dataset contains.

```
data_df.columns
```

```
Index(['Restaurant_Name', 'Category', 'Rating', 'Cost_for_two', 'Veg', 'city',
       'Area', 'Locality', 'Address', 'Long_Distance_Delivery'],
      dtype='object')
```

It shows all the columns present in the dataset.

```
data_df.isnull().sum()
```

```
Restaurant_Name              0
Category                     0
Rating                   17666
Cost_for_two                 0
Veg                          0
city                         0
Area                         0
Locality                    24
Address                      2
Long_Distance_Delivery       0
dtype: int64
```

It shows how many fields in the dataset contains a null value.

```
data_df.describe()
```

	Rating	Cost_for_two	Long_Distance_Delivery
count	14138.000000	31804.000000	31804.000000
mean	3.907745	289.795592	0.358414
std	0.456514	174.896346	0.479542
min	1.100000	0.000000	0.000000
25%	3.700000	200.000000	0.000000
50%	4.000000	250.000000	0.000000
75%	4.200000	300.000000	1.000000
max	5.000000	3100.000000	1.000000

It gives the general description of the whole numeric data.

```
d1_df=data_df.copy()
```

This will make a copy of our original dataset so that the main data does not get lost.

```
d1_df
```

	Restaurant_Name	Category	Rating	Cost_for_two	Veg	city	A
0	KFC	American,Snacks,Biryani	3.9	400	False	Delhi	Paharg
1	McDonald's	American	4.3	400	False	Delhi	Kashm
2	Haldiram's	Sweets,Snacks,North Indian	4.0	350	True	Delhi	Char Ch
3	Chai Point	Bakery,Beverages,Maharashtrian,Snacks,Street F...	4.2	150	False	Delhi	Connau Pl
4	Bikanervala Chandni Chowk	Street Food,Sweets	4.1	400	False	Delhi	Old D
...	...	...	...	...	...	...	
31799	Khukri Desi Mutton & Chicken Point	Indian,Biryani,Beverages	NaN	300	False	Bareilly	Phec
31800	Monis Restaurant	South Indian,Chinese,Snacks,Desserts,North Indian	NaN	250	True	Bareilly	Sun Vist
31801	T3 Tasty Table Talks	Indian,Thalis	NaN	200	True	Bareilly	MahaNa Col
31802	The Green Lemon	Indian	NaN	400	False	Bareilly	Ch V
31803	Pizza Home City	Pizzas,Beverages,Pastas,Italian,Desserts,Snacks	NaN	200	True	Bareilly	Ashut

31804 rows × 10 columns

```
data_df=data_df.drop("Address",axis=1)
data_df=data_df.dropna()
```

```
data_df=data_df.drop("Area",axis=1)
data_df=data_df.dropna()
```

```
data_df=data_df.drop("Locality",axis=1)
data_df=data_df.dropna()
```

```
data_df=data_df.drop("Long_Distance_Delivery",axis=1)
data_df=data_df.dropna()
```

These will drop the columns that we now don't necessarily need in our analysis.

```
data_df
```

	Restaurant_Name	Category	Rating	Cost_for_two	Veg	city
0	KFC	American,Snacks,Biryani	3.9	400	False	Delhi
1	McDonald's	American	4.3	400	False	Delhi
2	Haldiram's	Sweets,Snacks,North Indian	4.0	350	True	Delhi
3	Chai Point	Bakery,Beverages,Maharashtrian,Snacks,Street F...	4.2	150	False	Delhi
4	Bikanervala Chandni Chowk	Street Food,Sweets	4.1	400	False	Delhi
...	...	...	...	...	...	...
31787	Dosa Planet	South Indian	3.8	200	True	Bareilly
31788	Baba Biryani	Mughlai	3.6	400	False	Bareilly
31792	Cafe Coffee Day	Beverages,Snacks,Desserts	3.8	300	False	Bareilly
31793	Chinese 'X' Press	Chinese	3.6	200	False	Bareilly
31798	Om Kaleshwar Sweets & Bhojnalaya	South Indian,Fast Food,Combo,Beverages	4.0	100	True	Bareilly

14120 rows × 6 columns

```
d1_df
```

	Restaurant_Name	Category	Rating	Cost_for_two	Veg	city	A
0	KFC	American,Snacks,Biryani	3.9	400	False	Delhi	Paharg
1	McDonald's	American	4.3	400	False	Delhi	Kashm C
2	Haldiram's	Sweets,Snacks,North Indian	4.0	350	True	Delhi	Char Ch
3	Chai Point	Bakery,Beverages,Maharashtrian,Snacks,Street F...	4.2	150	False	Delhi	Connau Pl

	Restaurant_Name	Category	Rating	Cost_for_two	Veg	city	A
4	Bikanervala Chandni Chowk	Street Food,Sweets	4.1	400	False	Delhi	Old D
...	...	...	...	...	...	...	
31799	Khukri Desi Mutton & Chicken Point	Indian,Biryani,Beverages	NaN	300	False	Bareilly	Phe(I
31800	Monis Restaurant	South Indian,Chinese,Snacks,Desserts,North Indian	NaN	250	True	Bareilly	Sun Vist
31801	T3 Tasty Table Talks	Indian,Thalis	NaN	200	True	Bareilly	MahaNa Col
31802	The Green Lemon	Indian	NaN	400	False	Bareilly	Ch V
31803	Pizza Home City	Pizzas,Beverages,Pastas,Italian,Desserts,Snacks	NaN	200	True	Bareilly	Ashut

31804 rows × 10 columns

The copy of the dataset remains unchanged irrespective of the changes that we are doing on the original dataset.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "anujasthana2001/swiggy-data-analysis" on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/anujasthana2001/swiggy-data-analysis
```

'https://jovian.ai/anujasthana2001/swiggy-data-analysis'

Exploratory Analysis and Visualization

In this section we will perform the necessary computations on dataset to get some brief and analysis about the data and get some inferences from the data it can be numeric or visual computation.
In visual computation we will use graphs, charts and plots to show the data visually which is lot better to the understand the data and trends in it.

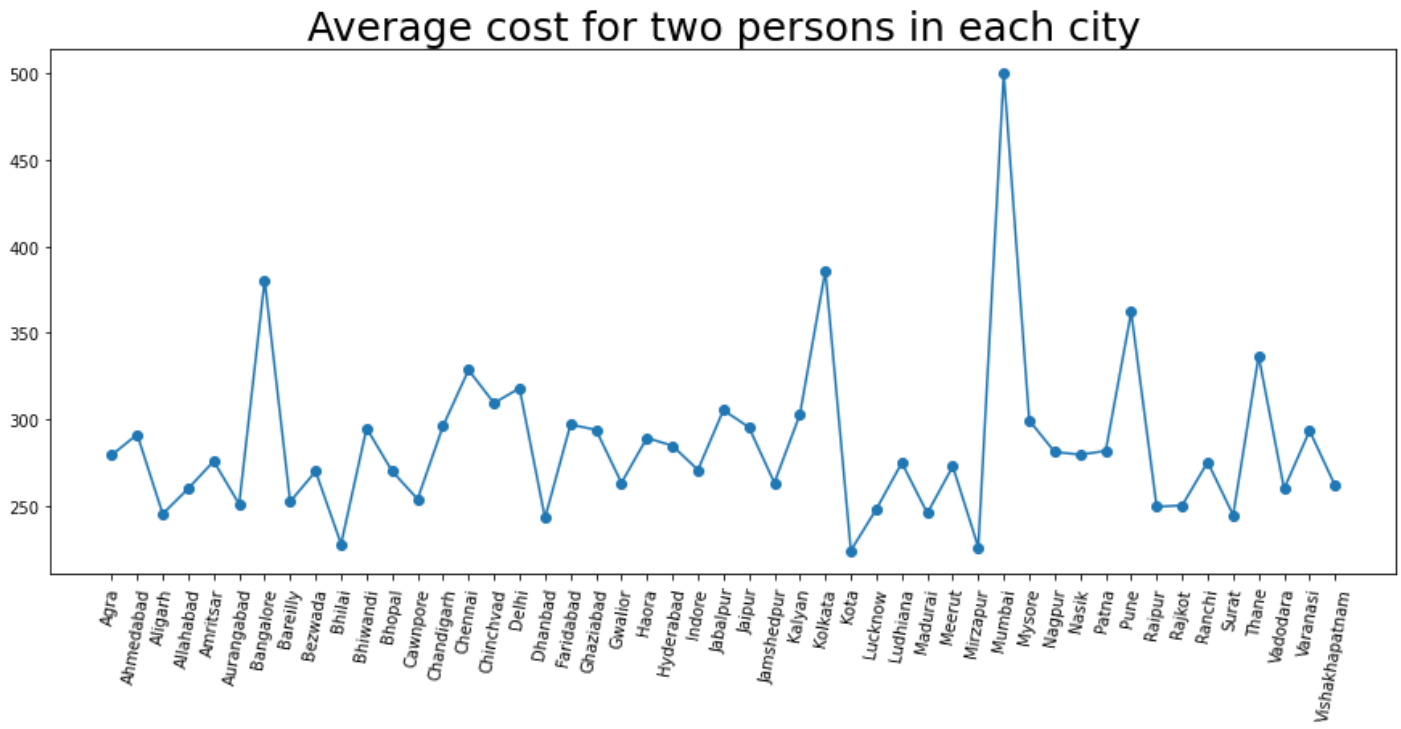
```
cost=data_df.groupby("city")["Cost_for_two"].mean()
cost
```

```
city
Agra              279.313253
Ahmedabad         291.201299
Aligarh           245.218750
```

```
Allahabad          259.855072
Amritsar           275.732484
Aurangabad         250.871134
Bangalore          380.553816
Bareilly           252.287879
Bezwada            269.849057
Bhilai             227.777778
Bhiwandi           294.828571
Bhopal             269.991189
Cawnpore           253.775561
Chandigarh         296.119512
Chennai            328.513514
Chinchvad          309.349462
Delhi              317.984536
Dhanbad            242.833333
Faridabad          296.949309
Ghaziabad          293.876333
Gwalior            263.175182
Haora              289.220690
Hyderabad          284.673344
Indore             270.698765
Jabalpur           305.288462
Jaipur             295.374150
Jamshedpur         263.225806
Kalyan             302.738562
Kolkata            386.015719
Kota               224.016393
Lucknow            247.978088
Ludhiana           274.787692
Madurai            245.652582
Meerut             272.713333
Mirzapur           226.000000
Mumbai             500.266026
Mysore             299.300000
Nagpur             281.077320
Nasik              279.525000
Patna              281.666667
Pune               362.046595
Raipur             249.318182
Rajkot             250.151515
Ranchi             275.194332
Surat              244.546632
Thane              336.552106
Vadodara           259.791809
Varanasi           293.343750
Vishakhapatnam     261.607692
Name: Cost_for_two, dtype: float64
```

Here we are finding the average cost for two persons to eat in each city using mean functions, we are also grouping the data w.r.t to City.

```
plt.figure(figsize=(15,6))
plt.tight_layout()
plt.xticks(rotation=80)
plt.title("Average cost for two persons in each city", size=25)
plt.plot(cost, marker='o');
```



The above Line graph shows the average cost for two persons to eat in a city.
From above graph the conclusion can be drawn that the most expensive place for two people is **Mumbai** and the least cost for two persons to eat is **Kota**.

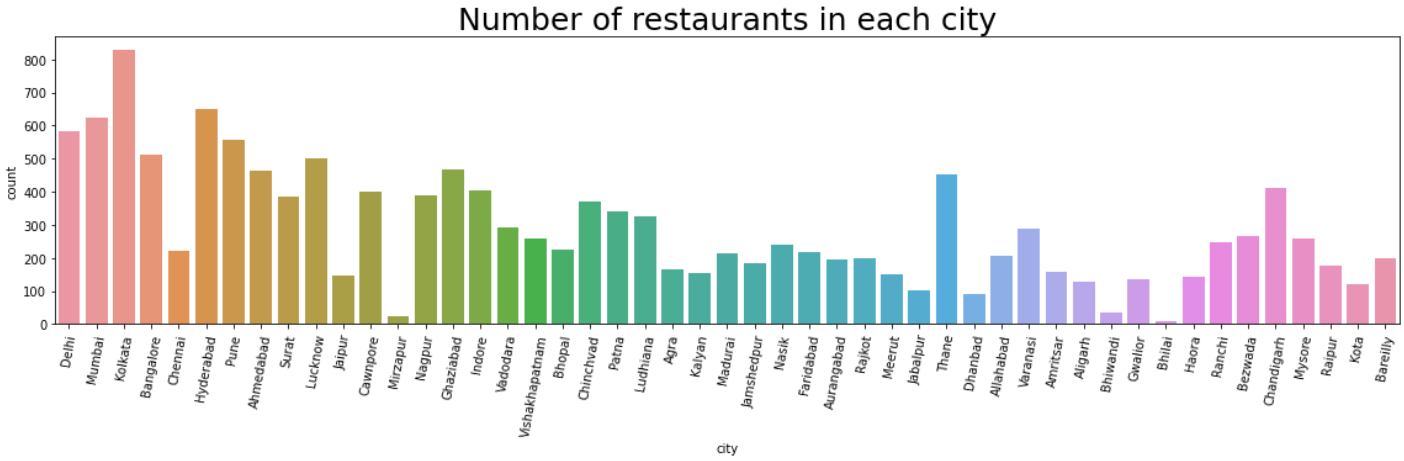
```
citywise_df=data_df.groupby('city')['Restaurant_Name'].count()
citywise_df
```

```
city
Agra              166
Ahmedabad         462
Aligarh           128
Allahabad         207
Amritsar          157
Aurangabad        194
Bangalore         511
Bareilly          198
Bezwada           265
Bhilai              9
Bhiwandi           35
Bhopal            227
Cawnpore          401
Chandigarh        410
Chennai           222
Chinchvad         372
Delhi             582
```

```
Dhanbad            90
Faridabad         217
Ghaziabad         469
Gwalior           137
Haora             145
Hyderabad         649
Indore            405
Jabalpur          104
Jaipur            147
Jamshedpur        186
Kalyan            153
Kolkata           827
Kota              122
Lucknow           502
Ludhiana          325
Madurai           213
Meerut            150
Mirzapur           25
Mumbai            624
Mysore            260
Nagpur            388
Nasik             240
Patna             339
Pune              558
Raipur            176
Rajkot            198
Ranchi            247
Surat             386
Thane             451
Vadodara          293
Varanasi          288
Vishakhapatnam    260
Name: Restaurant_Name, dtype: int64
```

Here we are counting the number of restaurants in each city using count functions and we are also grouping it w.r.t to City.

```
plt.figure(figsize=(16,5))
sns.countplot(x=data_df.city)
plt.xticks(rotation=80)
plt.tight_layout()
plt.title("Number of restaurants in each city",size=25);
```



The above Graph shows the number of restaurant situated in each city.
It can be seen that the maximum numbers of restaurants are located in **Kolkata** and the lowest number of restaurants are located in **Bhilai**.

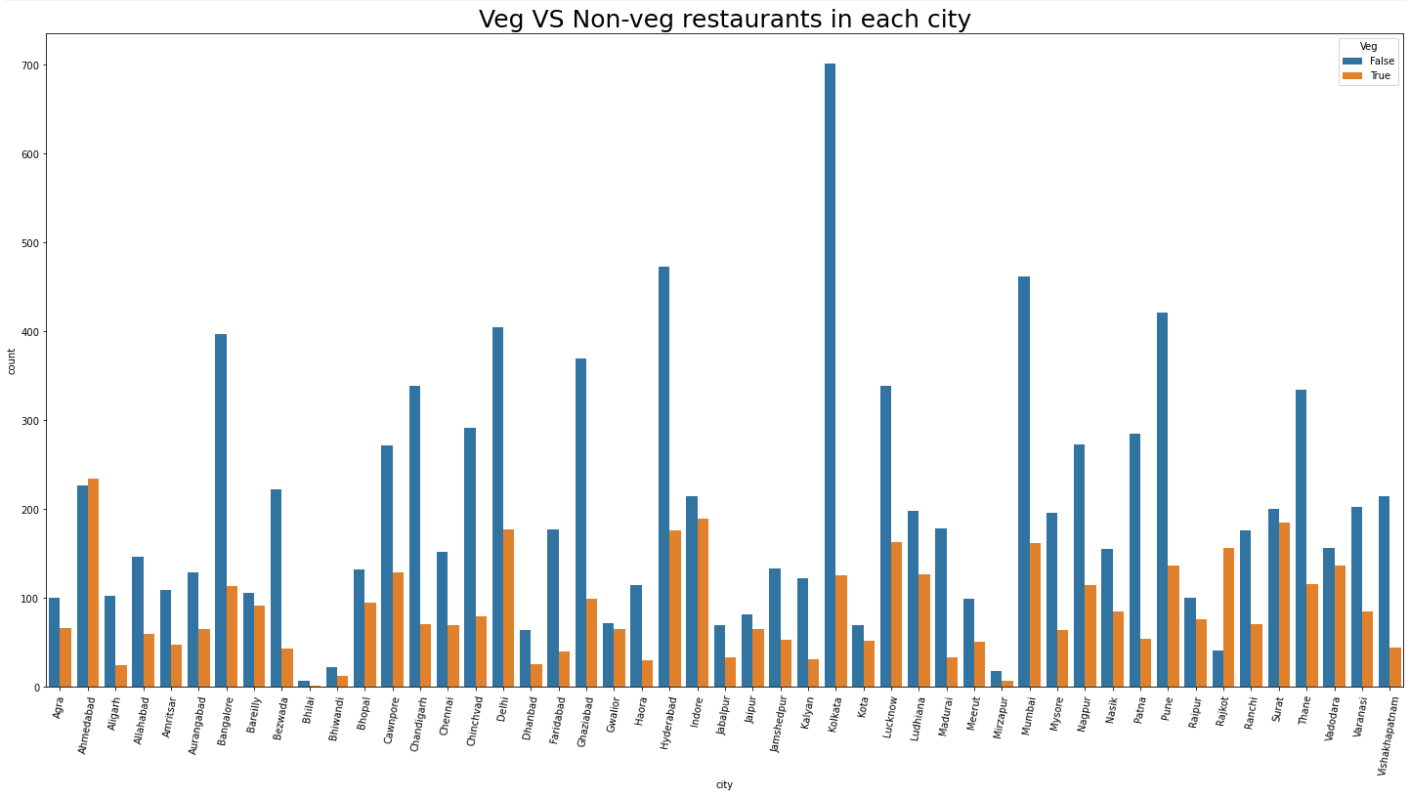
```
type_df=data_df.groupby(['Veg','city'])['Restaurant_Name'].count()
type_df.head(50)
```

Veg	city	
False	Agra	100
	Ahmedabad	227
	Aligarh	103
	Allahabad	147
	Amritsar	109
	Aurangabad	129
	Bangalore	397
	Bareilly	106
	Bezwada	222
	Bhilai	7
	Bhiwandi	22
	Bhopal	132
	Cawnpore	272
	Chandigarh	339
	Chennai	152
	Chinchvad	292
	Delhi	405
	Dhanbad	64
	Faridabad	177
	Ghaziabad	370
	Gwalior	72
	Haora	115
	Hyderabad	473
	Indore	215
	Jabalpur	70
	Jaipur	82
	Jamshedpur	133
	Kalyan	122
	Kolkata	701
	Kota	70

```
         Lucknow            339
         Ludhiana           198
         Madurai            179
         Meerut              99
         Mirzapur            18
         Mumbai             462
         Mysore             196
         Nagpur             273
         Nasik              155
         Patna              285
         Pune               421
         Raipur             100
         Rajkot              41
         Ranchi             176
         Surat              201
         Thane              335
         Vadodara           156
         Varanasi           203
         Vishakhapatnam     215
True     Agra                66
Name: Restaurant_Name, dtype: int64
```

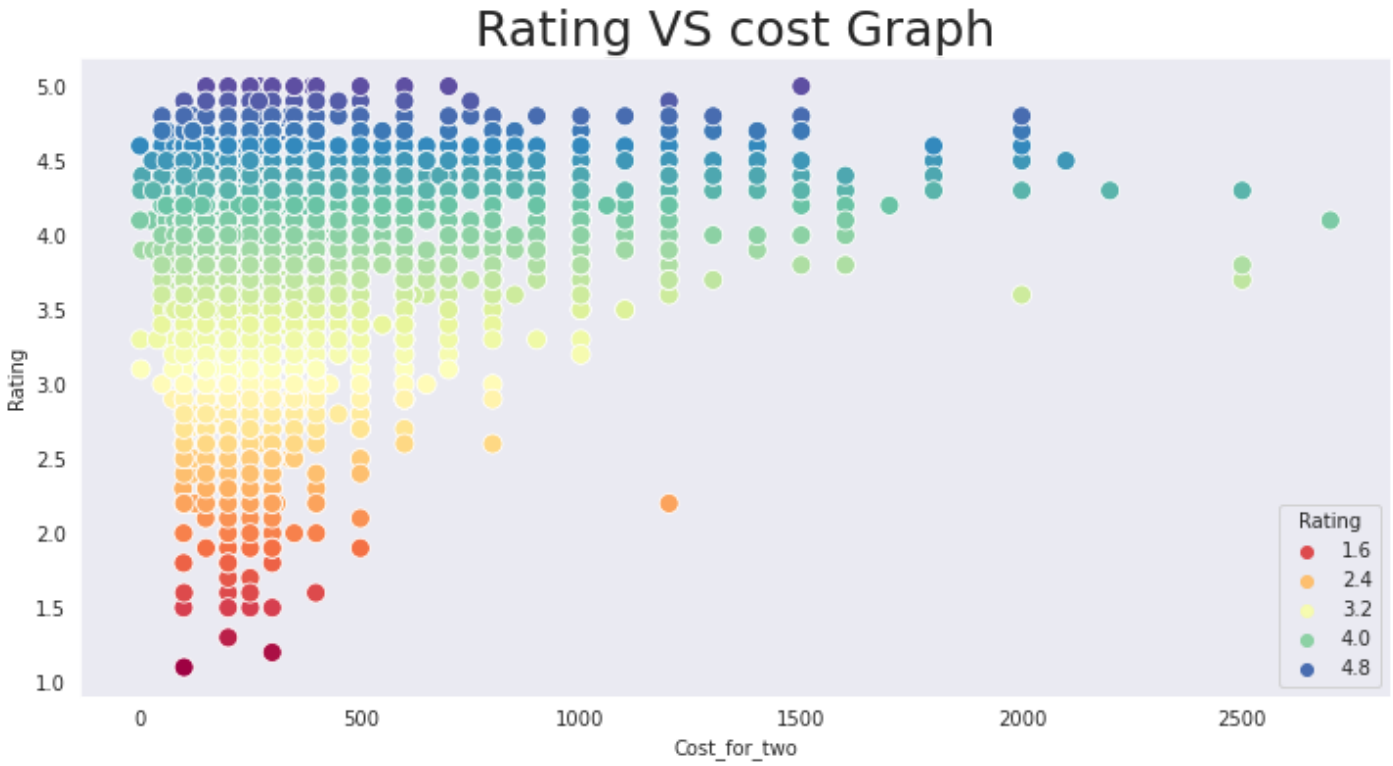
Here we are calculating the number of veg and non-veg restaurants of each city using count function and we are also grouping it w.r.t to City and veg/non-veg type.

```
plt.figure(figsize=(20,10))
plt.xticks(rotation=80)
plt.tight_layout()
sns.countplot(x=data_df["city"].sort_values(),hue=data_df["Veg"])
plt.title("Veg VS Non-veg restaurants in each city",size=25);
```



The above graph shows the number of veg and non-veg restaurants located in each city.
According to the Graph the maximum number of non-veg restaurants are located in **Kolkata** and the minimum number of non-veg restaurants are located in **Bhilai**.
Similarly, the maximum number of veg restaurants are located in **Ahmedabad** and the minimum number of veg restaurants are located in **Bhilai**.
One more information can be seen that **Bhilai** has very less amount of restaurants.

```
plt.figure(figsize=(12,6))
sns.set_style("dark")
plt.title("Rating VS cost Graph",size=25)
sns.scatterplot(x=data_df.Cost_for_two , y=data_df.Rating,hue=data_df.Rating,palette="S
```



The above scatter plot shows the rating and cost for two persons to eat in restaurants.
It shows the maximum rating restaurants are in range of **400-1200** which means a good rating restaurant on average will cost around **800 for two**.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "anujasthana2001/swiggy-data-analysis" on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/anujasthana2001/swiggy-data-analysis
```

```
'https://jovian.ai/anujasthana2001/swiggy-data-analysis'
```

Asking and Answering Questions

Till now we have gathered almost all the insights from the data and found out about every detail about the restaurants their expenses, rating, location etc.

Now its time that we can ask some questions regarding the dataset and give answers by using numeric computations and visual graphs.

Q1: Find the percent of Veg and Non-veg restaurants and also find Non-veg to Veg ratio?

To answer this question we will count the total numbers of veg and non-veg restaurants and then with the help of a graph we will visually show that information.

```
food_type = data_df.groupby('Veg')['city'].count()
food_type
```

```
Veg
False    9877
True     4243
Name: city, dtype: int64
```

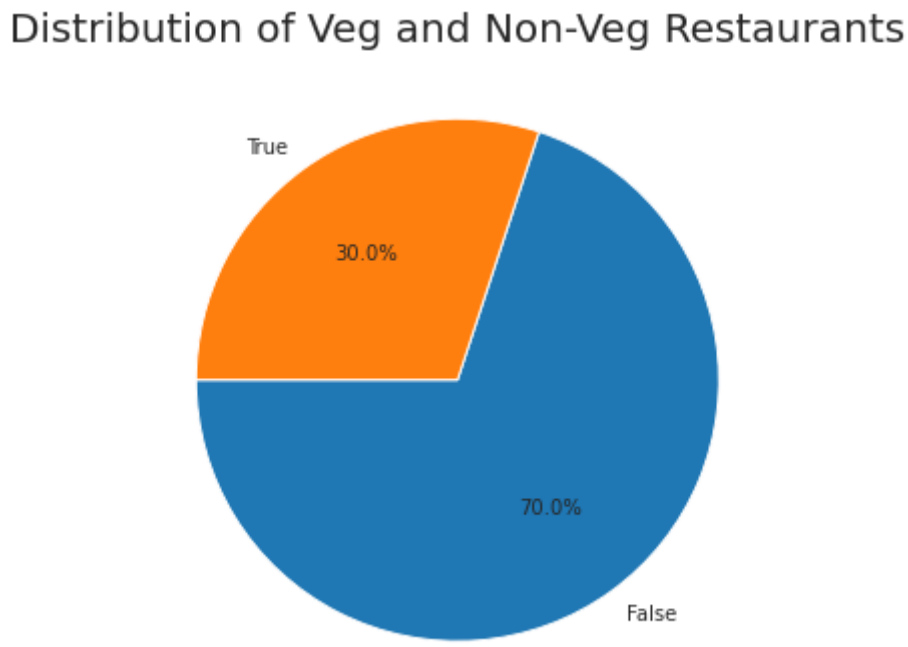
First we are counting the total number of veg and non-veg restaurants using count functions and grouping it w.r.t veg/non-veg type and saving the information in new dataframe called food_type.

```
food_type = food_type.reset_index()
food_type = food_type.rename(columns={'city':'Total_Restaurant'})
food_type
```

	Veg	Total_Restaurant
0	False	9877
1	True	4243

Now we are modifying and changing the dataframe into a meaningful tabular data form so that we can get values and insights from that table.
First we are initializing the index to the table and then changing the column name from city to Total_Restaurant which will tells us that it shows the total number of Veg and Non-veg restaurants.

```
plt.figure(figsize=(12,6))
plt.title("Distribution of Veg and Non-Veg Restaurants",size=20)
plt.pie(food_type.Total_Restaurant, labels=food_type.Veg,autopct='%1.1f%%', startangle=
```



The above graph is a pie graph or plot which shows the distribution of veg and non-veg restaurants across the metro cities.

From the numeric computations and the visual depiction we can infer some information about the total number of veg and non-veg restaurants located in metro cities.
We can see that the overall Non-Veg restaurants are more compared to Veg restaurants, almost **70% of the restaurants are non-veg restaurants and rest 30% are veg.**
This shows the ratio of non-veg to veg restaurants is approximately near to **2.33**.

Q2: Which is the best city to eat with high ratings and less cost for two?

To answer this question we have to find the ratings and cost for each restaurant of a city and then plot it on a graph to get our desired results from the data.

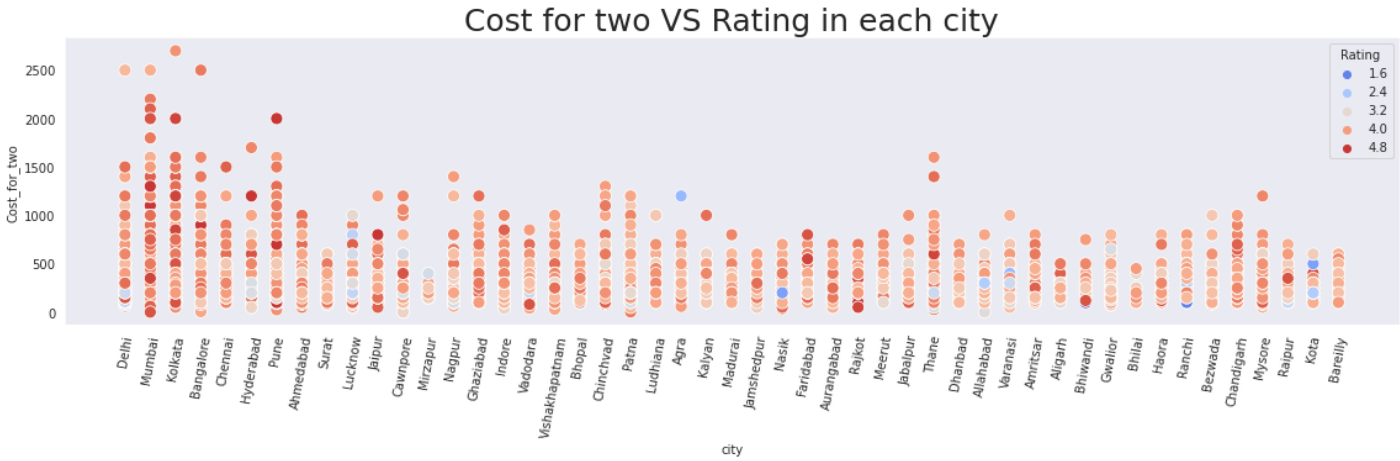
```
best_place=data_df[['city','Cost_for_two','Rating']]
best_place.head()
```

	city	Cost_for_two	Rating
0	Delhi	400	3.9
1	Delhi	400	4.3
2	Delhi	350	4.0
3	Delhi	150	4.2
4	Delhi	400	4.1

First we are creating a sub data frame or set from the main data frame in which we will include only the columns which are useful for answering the above question.
We are including the city name, cost for two persons to eat and the rating for them and naming this data frame as best_place

```
plt.figure(figsize=(16,5))
sns.scatterplot(data=best_place, x="city" ,y="Cost_for_two",hue="Rating",s=100,palette=
```

```
plt.xticks(rotation=80)
plt.tight_layout()
plt.title("Cost for two VS Rating in each city",size=25);
```



The above graph is a scatter plot graph which is drawn to show the cost for two and ratings of restaurants according to their respective cities.
The different color shows the rating of the restaurants with blue being minimum (1.6) and red being maximum (4.8).

Now moving on to the question to find a best place to eat food with high rating and less cost can be figured out by using the above scattered plot graph.

But before moving on to the answer we must acknowledge the fact that cost for two in Mumbai is reaching almost 0 this shows that there is some fault in the data of Mumbai.

Now answering the question we can find out that **Jaipur and Rajkot's maximum cost for two persons is less than 1000 and rating in both these cities is 4.0 or above.**
So it can be said **Jaipur** and **Rajkot** are best places to eat with less budget and high quality.

Q3: Which is the most expensive category of food and also give its rating?

To answer this question we will calculate the category which is mostly ordered and then plot a graph to find our desired answer.

```
category_count=data_df[['Category']]
```

```
category_count.value_counts().head(10)
```

```
Category
Indian                  467
North Indian            418
Chinese                 313
South Indian            255
Bakery                  209
Pizzas                  203
Desserts                194
North Indian,Chinese    184
Snacks                  174
```

```
Beverages                 160
dtype: int64
```

Here we have checked the highest amount of categories which are used or ordered the maximum number of times we created a category_count and counted the number of category of each snack and the highest category is selected to move further.

```
most_famous_category=data_df[data_df["Category"].isin(['Indian', 'Chinese', 'Snacks','B
```

```
most_famous_category
```

	Restaurant_Name	Category	Rating	Cost_for_two	Veg	city
11	Jung Bahadur Kachori Wala (Old Delhi Original)	Snacks	4.5	100	True	Delhi
39	Gupta Ji's Pav Bhaji	Snacks	3.8	150	True	Delhi
61	Ministry Of Cakes	Bakery	4.0	300	False	Delhi
74	SUGANDH	Indian	4.1	500	False	Delhi
105	Krispy Kreme	Desserts	4.0	400	False	Delhi
...	...	...	...	...	...	...
31742	Fast Food Junction	Chinese	3.8	150	False	Bareilly
31765	Craving Grave	Indian	4.0	350	False	Bareilly
31773	Hotel Hirapanna	Indian	4.2	350	False	Bareilly
31782	Cravings	Indian	3.6	200	False	Bareilly
31793	Chinese 'X' Press	Chinese	3.6	200	False	Bareilly

1517 rows × 6 columns

Here we have created a sub data frame where we are only including values or categories which are famous and mostly ordered by the people.
Some of the category are Snacks, Beverages etc.

```
plt.figure(figsize=(16,5))
sns.set_style("white")
plt.title("Rating and Cost of restaurants w.r.t Category",size=25)
sns.scatterplot(data=most_famous_category ,x="Cost_for_two" , y="Category", hue="Rating
```

In above diagram we have plotted a scatter plot which shows the cost of the most ordered category and also depicts its ratings with the help of colors from light to dark in increasing order of rating.
Now answering the question that which category is most expensive, it is clear by the diagram that the most expensive dish of all are **Desserts** and **Chinese** foods, these foods are the most expensive type of food.
Both these category of foods have a rating of **4.8**.

Q4: Find the total number of restaurants with respect to their ratings?

To find the solution to this question we will calculate the total number of restaurants according to their ratings.

```
no_of_rest=data_df[['Rating','Restaurant_Name']]
```

Here we have created a subset of data which contains only Ratings and Restaurant names we will use this to numerically calculate the number of restaurants according to their ratings.

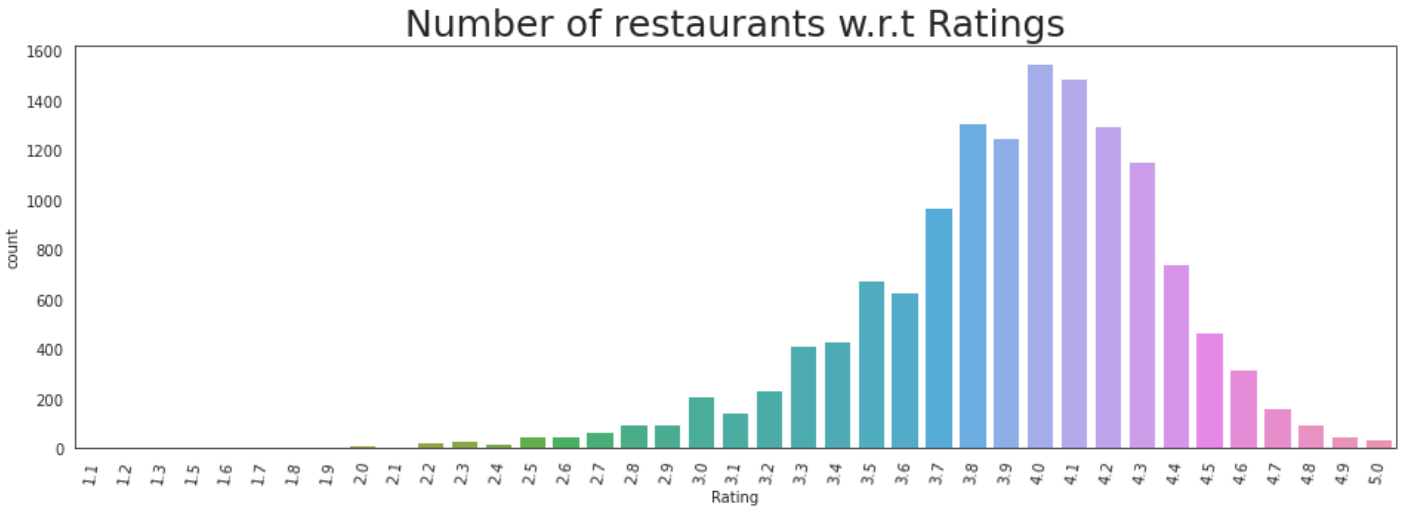
```
no_of_rest.groupby('Rating')['Restaurant_Name'].count()
```

```
Rating
1.1       2
1.2       1
1.3       3
1.5       4
1.6       4
1.7       2
1.8       6
1.9      10
2.0      12
2.1       8
2.2      28
2.3      32
2.4      21
2.5      48
2.6      49
2.7      65
2.8     100
2.9     100
3.0     210
3.1     144
3.2     237
3.3     411
3.4     429
3.5     677
3.6     629
3.7     971
3.8    1310
3.9    1249
4.0    1546
4.1    1490
4.2    1296
```

```
4.3    1154
4.4     745
4.5     465
4.6     321
4.7     161
4.8      96
4.9      47
5.0      37
Name: Restaurant_Name, dtype: int64
```

Here we have grouped data by using the groupby function on Ratings and used count to find the total number of restaurants with respect to ratings.

```
plt.figure(figsize=(16,5))
plt.xticks(rotation=80)
plt.title("Number of restaurants w.r.t Ratings",size=25)
sns.countplot(x=data_df.Rating);
```



In the above figure we have plotted a bar graph which shows the Number of restaurants with respect to their ratings.
We can clearly see the highest and lowest number of restaurants according to their ratings.
The graph above shows that the **maximum number of restaurants present have a rating of 4.0 followed by 4.1 and similarly the lowest number of restaurants are present of rating 1.2 followed by 1.1 and 1.7**.

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "anujasthana2001/swiggy-data-analysis" on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/anujasthana2001/swiggy-data-analysis
```

```
'https://jovian.ai/anujasthana2001/swiggy-data-analysis'
```

Inferences and Conclusion

There are various inferences and conclusions drawn from our analysis they are as follows:-

- The first conclusion that can be drawn is that the most expensive place for two people is **Mumbai** and the least expensive place for two persons to eat is **Kota**.
- It can be concluded that the maximum numbers of restaurants are located in **Kolkata** and the lowest number of restaurants are located in **Bhilai**.
- The maximum number of Non-Veg restaurants are located in **Kolkata** and the minimum number of non-veg restaurants are
- The maximum number of Veg restaurants are located in **Ahmedabad** and the minimum number of Veg restaurants are located in **Bhilai**.
- It can be concluded again from above points that **Bhilai** has the least amount of restaurants.
- The maximum rating of restaurants are in range of **400-1200** which means a good rating restaurant on average will cost around **800 for two**.
- It can be concluded that **70% of the restaurants are non-veg restaurants and rest 30% are veg.**
- The best place to eat which is budget friendly and has high rating is **Rajkot** and then **Jaipur**
- The most expensive type of food category is **Desserts** and **Chinese** food which also have high ratings.
- The maximum number of restaurants have a rating of 4.0 followed by 4.1, similarly the least number of restaurants have a rating of 1.2 followed by 1.1 and 1.7.
- One more information that we obtained from our analysis is that the cost for two persons in Mumbai is almost touching zero which is not possible this indicates that there is some mistake in the data for Mumbai in a restaurant

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "anujasthana2001/swiggy-data-analysis" on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/anujasthana2001/swiggy-data-analysis
```

```
'https://jovian.ai/anujasthana2001/swiggy-data-analysis'
```

References and Future Work

Resources that were useful for building this project are as follows:-

- Kaggle (The data set and some visual help):- https://www.kaggle.com/
- Seaborn user guide & tutorial (Seaborn attributes help):- https://seaborn.pydata.org/tutorial.html
- Stack Overflow Developer Survey (Helped in some errors):- https://stackoverflow.com/

Other analysis that can be done from this dataset can be the analysis of long distance delivery fee, the area and locality analysis etc.

```
import jovian
```

```
jovian.commit()
```