

Project : Zomato API – II

Ques 1:- The dataset is highly skewed toward the cities included in Delhi-NCR. So, we will summarise all the other cities in Rest of India while those in New Delhi, Ghaziabad, Noida, Gurgaon, Faridabad to Delhi-NCR. Doing this would make our analysis turn toward Delhi-NCR v Rest of India

- In code first we find India restaurants only, data is taken for country code =1 only i.e. for India.

```
import pandas as pd
import numpy as np

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

df = df[df['Country Code'] == 1] # considering only INDIAN Restaurants
```

- Then Replacing all the names of the cities as mentioned in question to Delhi-NCR And having all the cities remaining to be called as city – “Rest of India”.

```
df.reset_index(drop = True, inplace = True) # resetting the index

# summarising given cities as Delhi-NCR and other as Rest of India
df['City'].replace('New Delhi', 'Delhi-NCR', inplace = True)
df['City'].replace('Ghaziabad', 'Delhi-NCR', inplace = True)
df['City'].replace('Noida', 'Delhi-NCR', inplace = True)
df['City'].replace('Gurgaon', 'Delhi-NCR', inplace = True)
df['City'].replace('Faridabad', 'Delhi-NCR', inplace = True)

# all other cities as Rest of India
indexes = df[df['City'] != 'Delhi-NCR'].index

df.loc[indexes, 'City'] = 'Rest of India'
df
```

Ques 1.1:- Plot the bar graph of number of restaurants present in Delhi NCR vs Rest of India.

- In code we first import matplotlib.pyplot for plotting the graphs And then counting number of restaurants according to City.

```
import matplotlib.pyplot as plt

count = df['City'].value_counts() # counting number of restaurants according to City

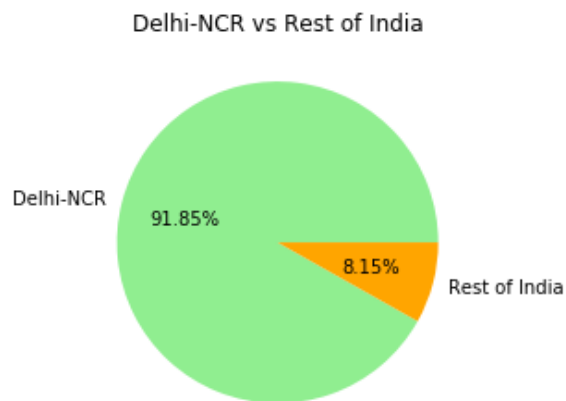
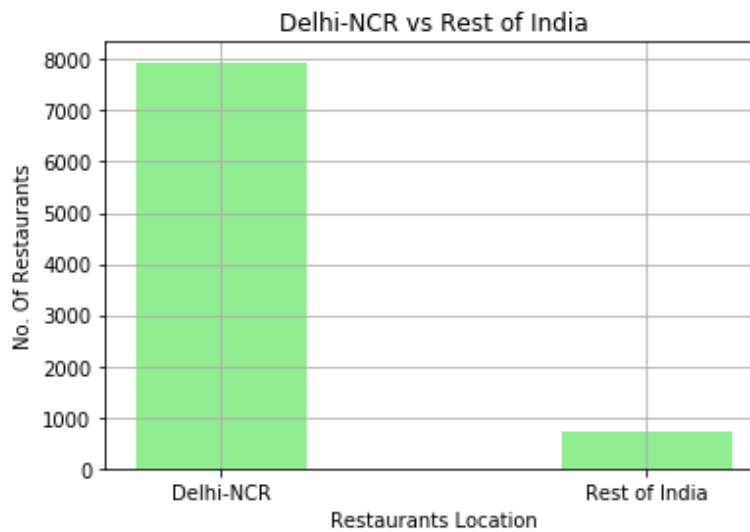
restaurants_loc = count.index
restaurants_count = count.values
```

- Now we plot Bar graph and pie chart using Delhi NCR vs Rest of India.

```
# plotting bar graph
plt.bar(restaurants_loc, restaurants_count, width = 0.4, color = 'orange')
plt.grid()
plt.title('Delhi-NCR vs Rest of India')
plt.xlabel("Restaurants Location")
plt.ylabel("No. Of Restaurants")
plt.show()

# plotting pie chart
plt.pie(restaurants_count, labels = restaurants_loc, colors = ['green', 'orange'], autopct = '%.2f%%')
plt.title('Delhi-NCR vs Rest of India')
plt.show()
```

Graphs:-



Result and Justification:-

As the result of the code shows and according to graphs also,

Delhi-NCR restaurants- 7947

Rest of India restaurants- 705

i.e. according to the pie chart we can see that the dataset we have for Indian restaurants, it contains 91.85% of data for Delhi-NCR restaurants and 8.15% for Rest of India.

Ques 1.2:- Find the cuisines which are not present in restaurant of Delhi NCR but present in rest of India. Check using Zomato API whether this cuisines are actually not served in restaurants of Delhi-NCR or just it due to incomplete dataset.

- In code first we created 2 sets that will contain the cuisines in Delhi-NCR and in Rest of India, then splitting different cuisines to make a list of cuisines using split() function by (,).

```
# creating 2 sets of cuisines according to the city.
NCR_cuisines = set()
Rest_cuisines = set()

df['Cuisines'] = df['Cuisines'].str.split(',') # splitting different cuisines to make a list of cuisines
```

- Then making 2 new dataframes containing data for each city and then running loops to add the cuisines to the sets of each city.

```
# 2 dataframes containing rows having specific city.
NCR = df[df['City'] == 'Delhi-NCR']
Rest = df[df['City'] == 'Rest of India']

# adding cuisines in according to city.
for i in NCR.index :
    for j in NCR.loc[i, 'Cuisines']:
        NCR_cuisines.add(j)

for i in Rest.index :
    for j in Rest.loc[i, 'Cuisines']:
        Rest_cuisines.add(j)
```

- Finally we get cuisines for Delhi-NCR and Rest of India, Now subtracting Delhi-NCR cuisines from Rest of India cuisines to get cuisines served in Rest of India But not in Delhi-NCR.

```
# cuisines served by restaurants of Rest of India but not in Delhi-NCR
unique_cuisines = Rest_cuisines - NCR_cuisines

for k in unique_cuisines:
    print(k)
```

Output:-

```
Cajun
Malwani
BBQ
German
```

Now checking Using API

- First we are creating header in which accept header and user key is passed for authentication, Now we are fetching details from the Zomato api to check whether the cuisines stored in diff are served in Delhi restaurant or not.

```
# fetching city-id for Delhi NCR
import requests

header = {"Accept": "application/json", "user-key": "0ab9b3ea3b2aa96bf4b2843fb711a48f", "User-agent": "curl/7.43.0"}
parameter = {'query': 'Delhi NCR'}
response = requests.get('https://developers.zomato.com/api/v2.1/locations', headers = header, params = parameter)
data = response.json()
for i in data['location_suggestions']:
    city_id = i['city_id']
```

- Now converting the stored data from json to python with the help of json() function.

```
# fetching all cuisines in Delhi NCR
res = requests.get('https://developers.zomato.com/api/v2.1/cuisines', headers = header, params = {'city_id' : city_id})
data1 = res.json()
```

- Then using for loop to find whether the cuisine_name is present in cuisines or not.

```
# checking if Restaurants in Delhi-NCR serves any of unique cuisines served in Rest of India but not in Delhi-NCR
for i in data1['cuisines']:
    if i['cuisine']['cuisine_name'] in unique_cuisines:
        print(i['cuisine']['cuisine_name'])
```

Output:-

```
BBQ
Malwani
```

Results and Justification:-

From the API code it is clear that restaurants presents in Delhi-NCR are serving two of these cuisines i.e. BBQ and Malwani and also in Rest of India. Therefore we can say that dataset is incomplete.

According to dataset we found that there are 4 cuisines that are not served in Delhi-NCR but actually only 2 cuisines are not served since BBQ and Malwani cuisines are served. So, it is due to incomplete dataset.

Ques 1.3:- Find the top 10 cuisines served by maximum number of restaurants in Delhi NCR and rest of India.

- First we create 2 dictionaries one for each Delhi-NCR cuisines and for Rest of India. And then running loops in both to fetch cuisines and count of each cuisine.

```
ncr = {}
rest = {}

for i in NCR.index :
    for j in NCR.loc[i, 'Cuisines']:
        ncr[j] = ncr.get(j, 0) + 1

for i in Rest.index :
    for j in Rest.loc[i, 'Cuisines']:
        rest[j] = rest.get(j, 0) + 1
```

- Now sorting the list of cuisines according to their number of counts and fetching top 10 cuisines and printing Top 10 cuisines and their counts for both Delhi-NCR and Rest of India.

```
# sorting the list of cuisines according to their number of counts and fetching top 10 cuisines.
sorted_ncr = sorted(ncr.items(), key = lambda kv:kv[1], reverse = True)[:10]
sorted_rest = sorted(rest.items(), key = lambda kv:kv[1], reverse = True)[:10]

print('Top 10 cuisines served by maximum number of restaurants in Delhi NCR - ')
print()
for i in sorted_ncr:
    print(i[0],i[1])
print()

print('Top 10 cuisines served by maximum number of restaurants in Rest of India -')
print()
for j in sorted_rest:
    print(j[0],j[1])
```

Output:-

Top 10 cuisines served by maximum number of restaurants in Delhi NCR -

North Indian 3597
Chinese 2448
Fast Food 1866
Mughlai 933
Bakery 697
South Indian 569
Continental 547
Desserts 542
Street Food 538
Italian 535

Top 10 cuisines served by maximum number of restaurants in Rest of India -

North Indian 349
Chinese 242
Continental 177
Italian 147
Cafe 136
Fast Food 97
South Indian 62
Mughlai 59
Desserts 55
Mexican 50

Result:-

We got top 10 cuisines for each Delhi-NCR cuisines and for Rest of India.

Ques 1.4:- Write a short detailed analysis of how cuisine served is different from Delhi NCR to Rest of India. Plot the suitable graph to explain your inference.

- In code first we import matplotlib.pyplot for plotting the graph then making list of Cuisines names and their counts for Delhi-NCR and Rest of India.

```
import matplotlib.pyplot as plt

# Cuisines names and their counts for Delhi-NCR and Rest of India
Delhi_cuisines_names = []
Delhi_cuisines_counts = []
rest_cuisine_name = []
rest_cuisine_count = []

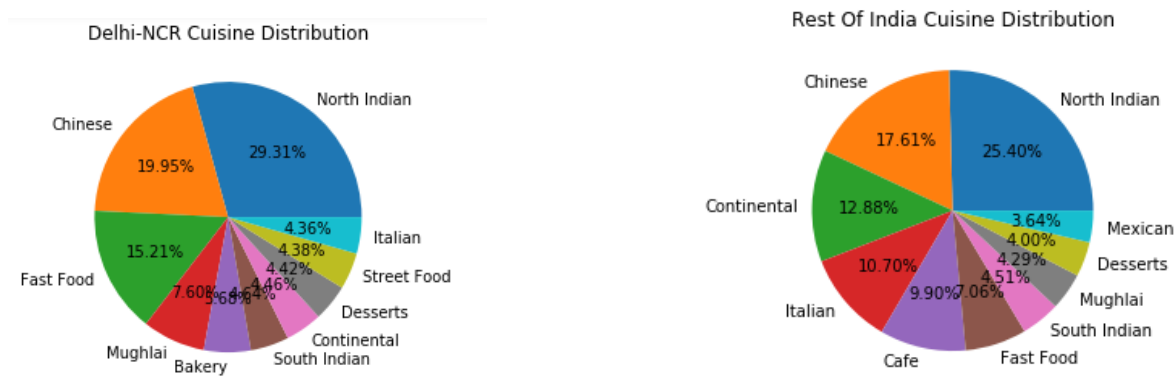
for i in range(10):
    Delhi_cuisines_names.append(sorted_ncr[i][0])
    Delhi_cuisines_counts.append(sorted_ncr[i][1])
    rest_cuisine_name.append(sorted_rest[i][0])
    rest_cuisine_count.append(sorted_rest[i][1])
```

- Now plotting pie charts for top ten cuisines in Delhi-NCR and in Rest of India.

```
# plotting pie charts for top ten cuisines in Delhi-NCR and in Rest of India.
plt.pie(Delhi_cuisines_counts, labels = Delhi_cuisines_names, autopct = '%.2f%%')
plt.title('Delhi-NCR Cuisine Distribution')
plt.show()

plt.pie(rest_cuisine_count, labels = rest_cuisine_name, autopct = '%.2f%%')
plt.title('Rest Of India Cuisine Distribution')
plt.show()
```

Graphs:-



Result and Justification:-

From the code and graph we can conclude that:-

1. North Indian and Chinese are very popular cuisines in Delhi-NCR as well as in Rest of India.
2. There are some cuisines which are more popular in rest of India like Mexican, cafe.
3. In Delhi-NCR Street food is loved where as its not the case with rest of India.
4. In Delhi-NCR fast food takes up a good chunk in cuisines whereas in Rest of India that chunk is took up by Italian cuisine .
5. South Indian cuisine is loved equally by both sides.

Ques 2:- User Rating of a restaurant plays a crucial role in selecting a restaurant or ordering the food from the restaurant.

Ques 2.1:- Write a short detail analysis of how the rating is affected by restaurant due following features:
Plot a suitable graph to explain your inference.

Ques 2.1.1:- Number of Votes given Restaurant

- First we plot a graph between Number of Votes v/s User Rating of restaurants.
- In code first we import matplotlib.pyplot for plotting the graph then find India restaurants only, data is taken for country code =1 only i.e. for India.

```
import pandas as pd
import numpy as np

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

df = df[df['Country Code'] == 1] # considering only INDIAN Restaurants

df.reset_index(drop = True, inplace = True)
```

- Now fetching votes and Rating color given to restaurants by sorting by number of votes given and then setting colors according to the Rating Color.

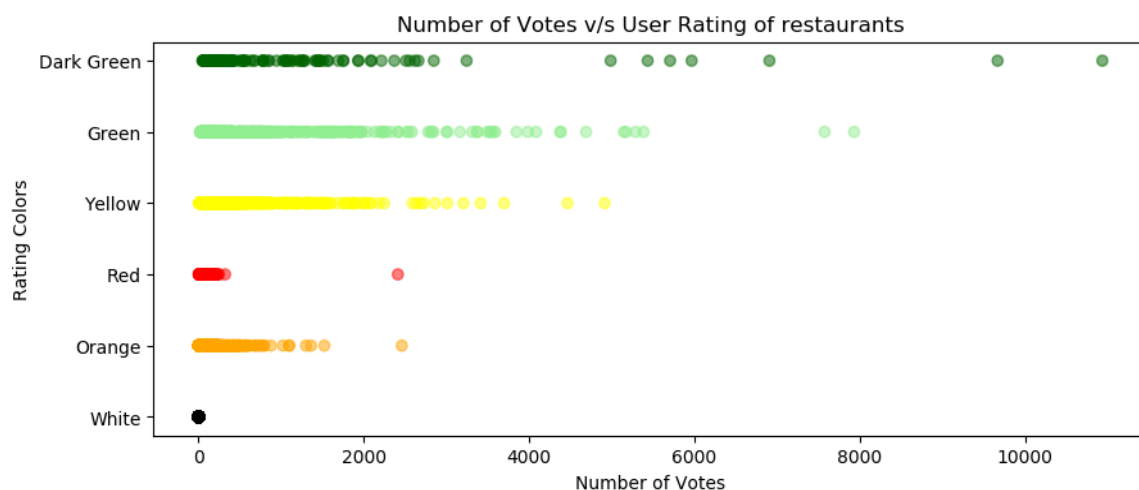
```
# fetching votes and Rating color given to restaurants by sorting by number of votes given.
x = df.sort_values(by = 'Votes')['Votes']
y = df.sort_values(by = 'Votes')['Rating color']

# setting colors according to the Rating Color.
color = []
for i in y:
    if i == 'White':
        color.append('Black')
    elif i == 'Dark Green':
        color.append('DarkGreen')
    elif i == 'Green':
        color.append('lightGreen')
    else:
        color.append(i)
```

- Finally, plotting scatter graph.

```
# plotting scatter graph
plt.figure(num=None, figsize=(10, 4), dpi=100, facecolor='w', edgecolor='green')
plt.scatter(x,y, c = color, alpha = 0.5)
plt.title('Number of Votes v/s User Rating of restaurants')
plt.xlabel('Number of Votes')
plt.ylabel('Rating Colors')
plt.show()
```

Graph:-



- Second we plot a graph between Number of Restaurants (Votes > 500) v/s Ratings.
- In code we first fetching data for restaurants having votes greater than 500 and taking number of votes for particular range of votings.

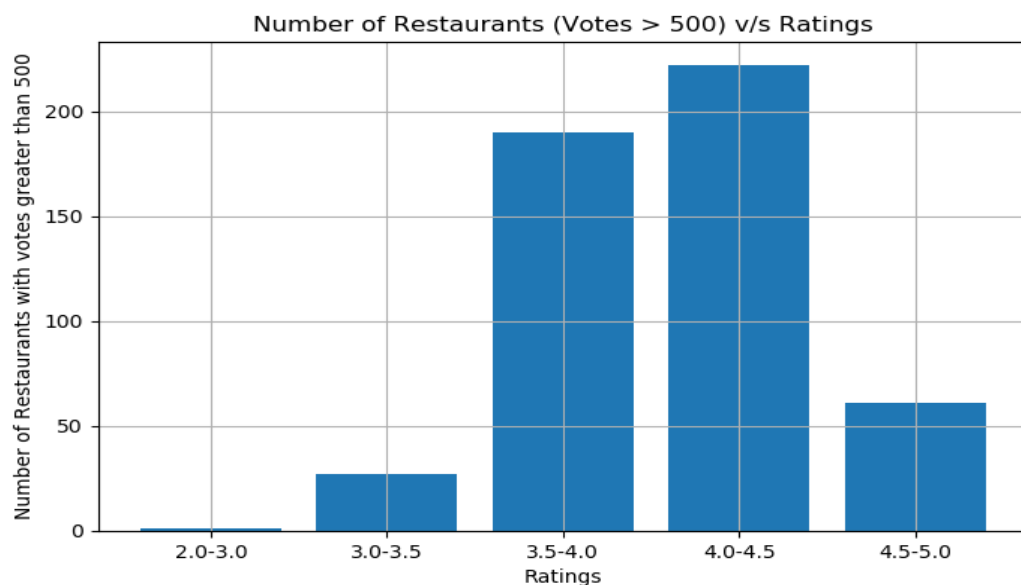
```
# fetching data for restaurants having votes greater than 500.
df1 = df[df['Votes'] > 500]
rating_count = df1['Rating color'].value_counts()

# taking number of votes for particular range of votings.
rating = ['2.0-3.0', '3.0-3.5', '3.5-4.0', '4.0-4.5', '4.5-5.0']
number_votes = [ rating_count['Red'], rating_count['Orange'], rating_count['Yellow'],
                 rating_count['Green'], rating_count['Dark Green']]
```

- Finally, plotting bar graph.

```
# plotting bar graph.
plt.figure(num=None, figsize=(8, 5), dpi=100, facecolor='w', edgecolor='green')
plt.bar(rating, number_votes)
plt.grid()
plt.title('Number of Restaurants (Votes > 500) v/s Ratings')
plt.xlabel('Ratings')
plt.ylabel('Number of Restaurants with votes greater than 500')
plt.show()
```

Graph:-



- Third we plot a graph between Rating vs Number of Votes.
- In code first we import matplotlib.pyplot for plotting the graph then find India restaurants only, data is taken for country code =1 only i.e. for India.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

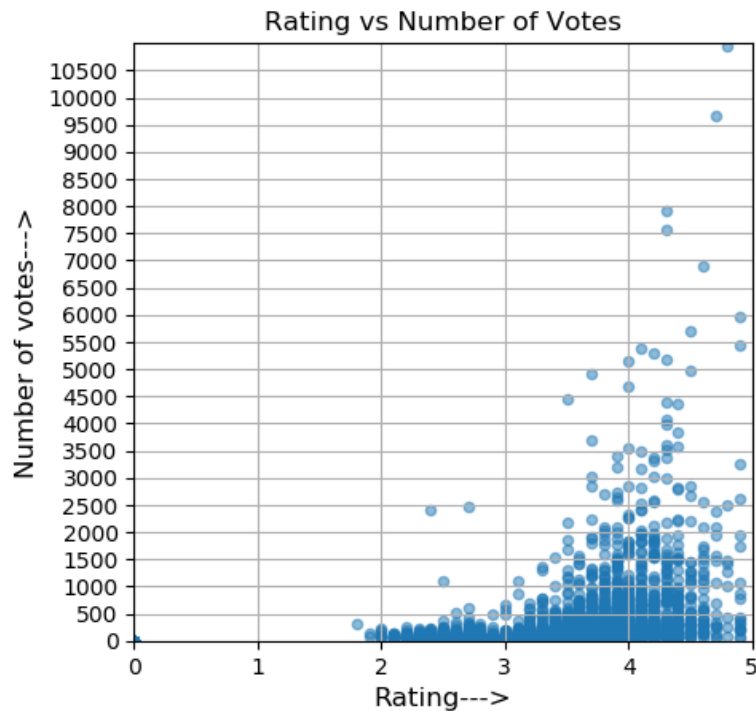
file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

# considering only INDIAN Restraunts -
df = df[df['Country Code'] == 1]
```


- Now plotting scatter graph between Number of Votes and Ratings.

```
# plotting scatter graph b/w Number of Votes and Ratings.
plt.figure(num=None, figsize=(5,5), dpi=100, facecolor='w', edgecolor='green')
plt.scatter(df['Aggregate rating'], df['Votes'], s=20, alpha=0.5)
plt.xticks(np.arange(0, 11000, 500))
plt.axis([0, 5, 0, 11000])
plt.xlabel('Rating--->', size=12)
plt.ylabel('Number of votes--->', size=12)
plt.title('Rating vs Number of Votes')
plt.grid()
plt.show()
```

Graph:-



Results and Justification:-

As the result of the code and the graphs we can deduce that:-

1. Looking at the number of votes ranging ≤ 500 , we see that the rating are in ranges from 0 to 5 from which we can say that votes less than 500 doesn't follow a trend and they have different types of ratings.
2. Now, for 500 to 2000, we see that the user ratings vary from 2.5 to 5.0 which favour's the hypothesis we created in our previous point. In this range, we see that user rating has improved a little bit as when the restaurants have gotten more votes, their variety of customers have also increased, hence earning them good ratings as well to neutralize the bad ones and getting an average to excellent user rating.
3. Also, most number of Restaurants receive votes between 500-2000.
4. Taking the Bar graph in consideration we see about most restaurant in the rating of 4.0-4.5 have votes greater than 500, and then about 190 restaurants in the rating of 3.5-4.0 have votes greater that 500.
5. So, major chunk of restaurants with votes > 500 have rating between 3.5 to 4.5.

6. Now, for restaurants with very high number of votes , we can see in the scatter plot the rating is generally excellent i.e. between 4.5 – 5.0.

So, Ratings for very good restaurants that have very high number of customers are generally excellent and they follow proportionality with number of votes .

But for Restaurants for votes < 8000 there, is no such deduction as they have good as well as bad reviews too so, ratings get neutralized b/w 3.5 – 4.5.

Ques 2.1.2:- Restaurant serving more number of cuisines.

- First we plot a graph between Number of Cuisines v/s Ratings.
- In code first we import matplotlib.pyplot for plotting the graph then find India restaurants only, data is taken for country code =1 only i.e. for India.

```
import pandas as pd
import numpy as np

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

# considering only INDIAN Restraunts -
df = df[df['Country Code'] == 1]

df.reset_index(drop = True, inplace = True)
```

- Then splitting Cuisines of each restaurant to have them in a list and next creating column having Number of cuisines for a particular restaurant.

```
# splitting Cuisines of each restaurant to have them in a list.
df['Cuisines'] = df['Cuisines'].str.split(pat = ', ')

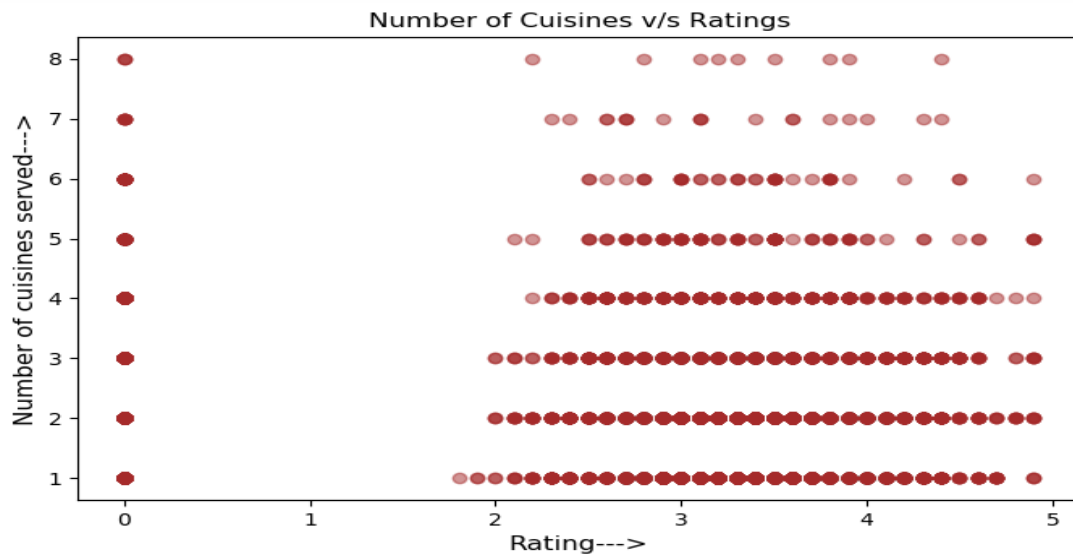
# creating column having Number of cuisines for a particular restaurant.
for i in df.index:
    df.loc[i, 'Number of cuisines'] = len(df.loc[i, 'Cuisines'])
```

- Now, sorting values according to the Aggregate Rating and finally plotting Scatter graph between Number of Cuisines v/s Ratings.

```
# sorting values according to the Aggregate Rating.
x = df.sort_values(by = 'Aggregate rating')['Aggregate rating']

# plotting Scatter graph.
plt.figure(num=None, figsize=(8, 5), dpi=100, facecolor='w', edgecolor='green')
plt.scatter(x, df['Number of cuisines'], c = 'brown', alpha = 0.5)
plt.title('Number of Cuisines v/s Ratings')
plt.xlabel('Rating-->', size=12)
plt.ylabel('Number of cuisines served-->', size=12)
plt.show()
```

Graph:-



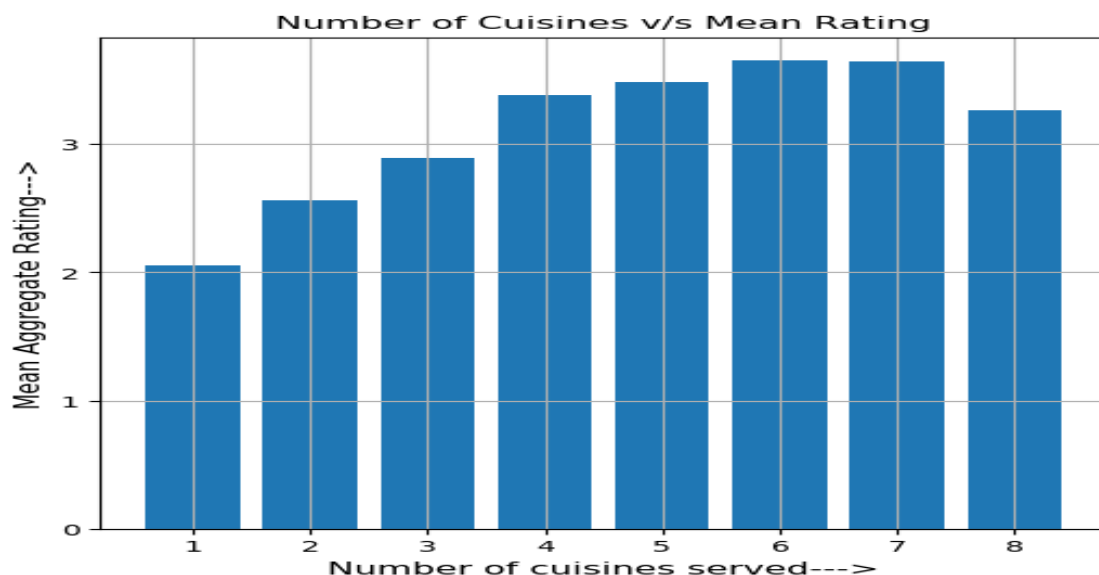
- Second we plot a graph between Number of Cuisines v/s Mean Rating.
- In code we first calculating mean rating for each number of cuisines.

```
# calculating mean rating for each number of cuisines
mean_rating = []
cuisine_count = []
for i in range(1,9):
    rating = df[df['Number of cuisines'] == i]['Aggregate rating'].mean()
    mean_rating.append(rating)
    cuisine_count.append(i)
```

- Now plotting scatter graph between Number of Cuisines v/s Mean Rating.

```
# plotting bar garph.
plt.figure(num=None, figsize=(6, 6), dpi=100, facecolor='w', edgecolor='green')
plt.grid()
plt.bar(cuisine_count, mean_rating)
plt.title('Number of Cuisines v/s Mean Rating')
plt.ylabel('Mean Aggregate Rating--->', size=12)
plt.xlabel('Number of cuisines served--->', size=12)
plt.yticks(np.arange(0, 4.0, 1))
plt.show()
```

Graph:-



Results and Justification:-

As the result of the code and the graphs we can deduce that:-

1. It is clear that when the number of cuisines provided increases from 3 to 8, generally the rating seems to converge between 3 and 4.
2. Restaurants providing more number of cuisines are not much likely to get higher ratings, especially when the number of cuisines provided exceeds 6.
3. It seems like when a restaurant provides too many cuisines, its focus on the quality of food offered diverges. while restaurants providing less cuisines focus on the quality of food to get good aggregate ratings.
4. Although there is no such connection between Cuisines on offer and their ratings.
5. But from the bar graph we can see that good ratings are given to restaurants having 3-7 cuisines on offer.

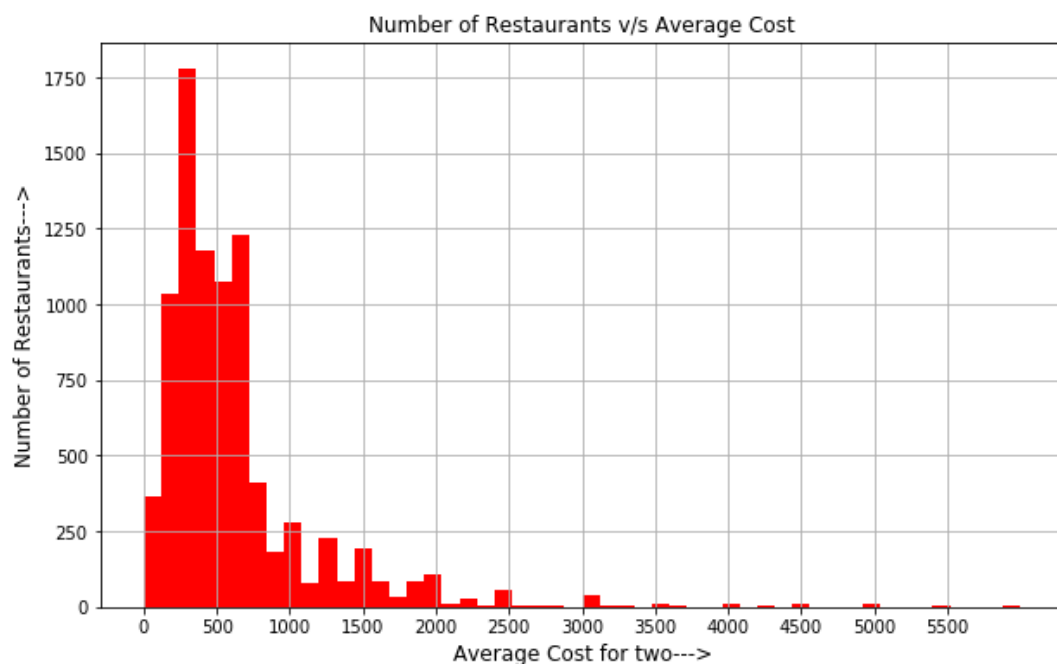
As long as the restaurants focuses on quality of food and have decent number of cuisines like between 3 – 6 and focuses on taste more that quantity etc. , Ratings will be excellent.

Ques 2.1.3:- Average Cost of Restaurant

- First we plot a graph between Number of Restaurants v/s Average Cost.
- Now plotting histogram for number of restaurants and their average cost.

```
# plotting histogram for number of restaurants and their average cost.
plt.figure(num=None, figsize=(10, 6), facecolor='w')
plt.hist(df['Average Cost for two'], range=[0, 6000], facecolor='red', align='mid', bins=50)
plt.xticks(np.arange(0, 6000, 500))
plt.grid()
plt.title('Number of Restaurants v/s Average Cost')
plt.xlabel('Average Cost for two-->', size=12)
plt.ylabel('Number of Restaurants-->', size=12)
plt.show()
```

Graph:-



- Second we plot a graph between Average Cost for Two v/s Rating.
- In code first we import matplotlib.pyplot for plotting the graph then find India restaurants only, data is taken for country code =1 only i.e. for India.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

# considering only INDIAN Restraunts -
df = df[df['Country Code'] == 1]

df['Aggregate rating'].dropna(inplace = True)
```

- Now plotting scatter graph between Average Cost for Two v/s Rating.

```
# plotting scatter graph.
plt.figure(num=None, figsize=(8,6), dpi=100, facecolor='w', edgecolor='green')
plt.scatter(df['Aggregate rating'], df['Average Cost for two'], s=20, alpha=0.5, c = 'purple')
plt.xlabel('Rating--->', size=12)
plt.ylabel('Average Cost for two--->', size=12)
plt.title('Average Cost for Two v/s Rating')
plt.grid()
plt.show()
```

Graph:-



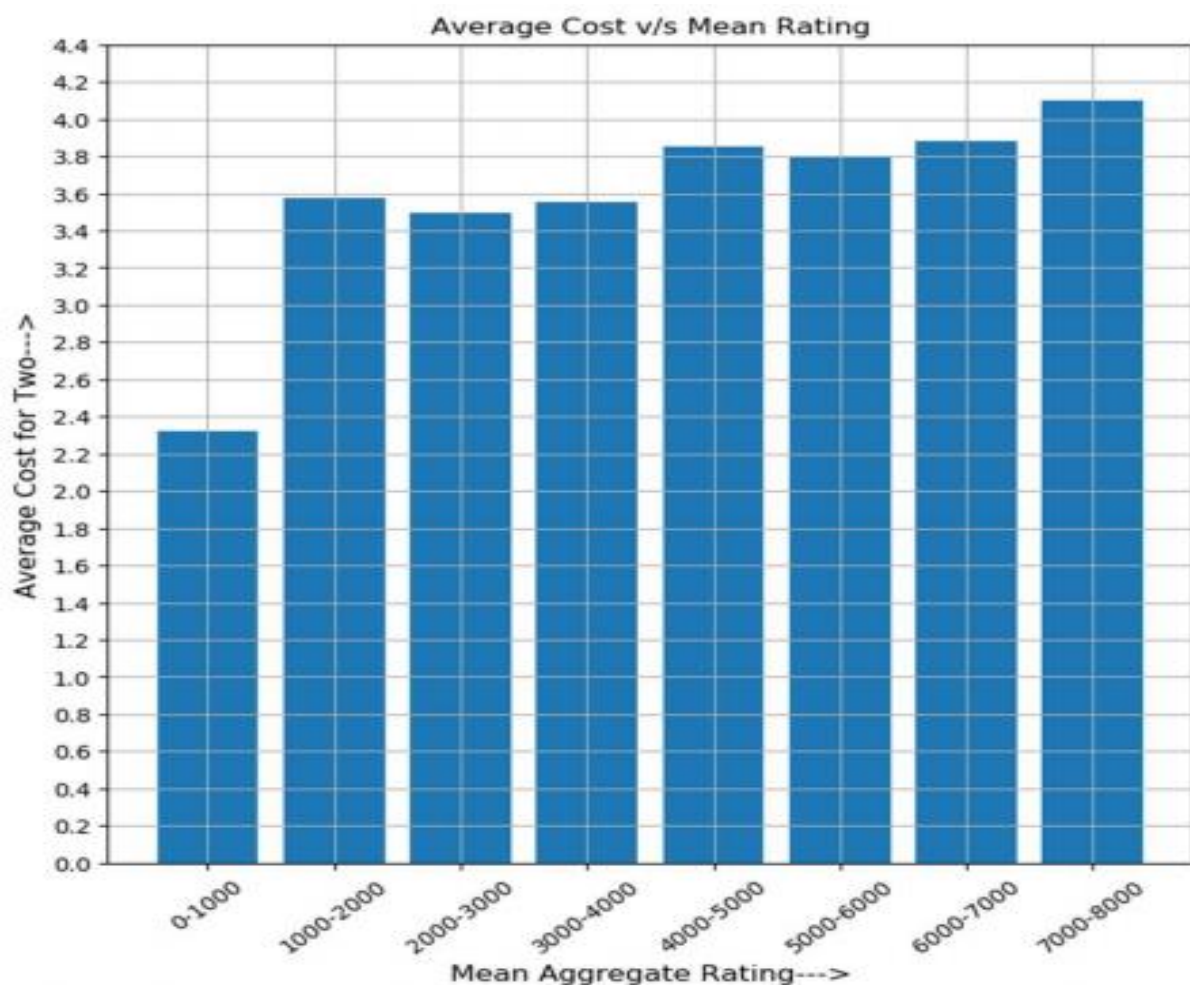
- Third we plot a graph between Average Cost v/s Mean Rating.
- In code we first calculating mean Rating for each range of average costs.

```
# calculating mean Rating for each range of average costs.
mean_rating = []
average_cost = []
for i in range(0,8000,1000):
    rating = df[(df['Average Cost for two'] >= i) & (df['Average Cost for two'] < (i + 1000))]['Aggregate rating'].mean()
    mean_rating.append(rating)
    cost = str(i) + '-' + str(i+1000)
    average_cost.append(cost)
```

- Now plotting bar graph between Average Cost v/s Mean Rating.

```
# plotting bar graph.
plt.figure(num=None, figsize=(8, 8), dpi=100, facecolor='w', edgecolor='green')
plt.grid()
plt.bar(average_cost, mean_rating)
plt.title('Average Cost v/s Mean Rating')
plt.yticks(np.arange(0, 4.5, 0.2))
plt.xlabel('Mean Aggregate Rating--->', size=12)
plt.ylabel('Average Cost for Two--->', size=12)
plt.xticks(rotation = 40)
plt.show()
```

Graph:-



Results and Justification:-

As the result of the code and the graphs we can deduce that:-

1. This histogram shows us the spread of price for two. We can see that it peaks before 500 ,therefore we can deduce that majority of the restaurants are in the price range of 200 – 1500.
2. Now, from the bar graph and the scatter plot it is clear that for range of cost between 1000 – 4000 , the rating hovers b/w 3.5 to 3.6 i.e. average .
3. Till 6000 cost the rating is under 4.0 for maximum restaurants.
4. It is seen that after that for expensive restaurants having cost for $2 > 6000$,the ratings are generally excellent i.e. b/w 4.0 – 5.0.

Therefore very expensive restaurants have excellent ratings.

Ques 2.1.4:- Restaurant serving some specific cuisines.

- First we plot a graph between Cuisines v/s User Rating.
- In code first we import matplotlib.pyplot for plotting the graph then find India restaurants only, data is taken for country code =1only i.e. for India.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

# considering only INDIAN Restraunts -
df = df[df['Country Code'] == 1]
```

- Now splitting cuisines and making dictionary of cuisines and their counts and then sorting dictionary according to the count of cuisines and fetching top 10 cuisines.

```
# splitting cuisines and making dictionary of cuisines and their counts
df['Cuisines'] = df['Cuisines'].str.split(',')
cuisines = {}

for i in df['Cuisines'] :
    for j in i:
        cuisines[j] = cuisines.get(j, 0) + 1

# sorting dictionary according to the count of cuisines and fetching top 10 cuisines.
sorted_cuisines = sorted(cuisines.items(), key = lambda kv:kv[1], reverse = True)[:10]
```

- Then fetching names of top 10 cuisines and next fetching name and ratings of the particular cuisine of particular restaurant.

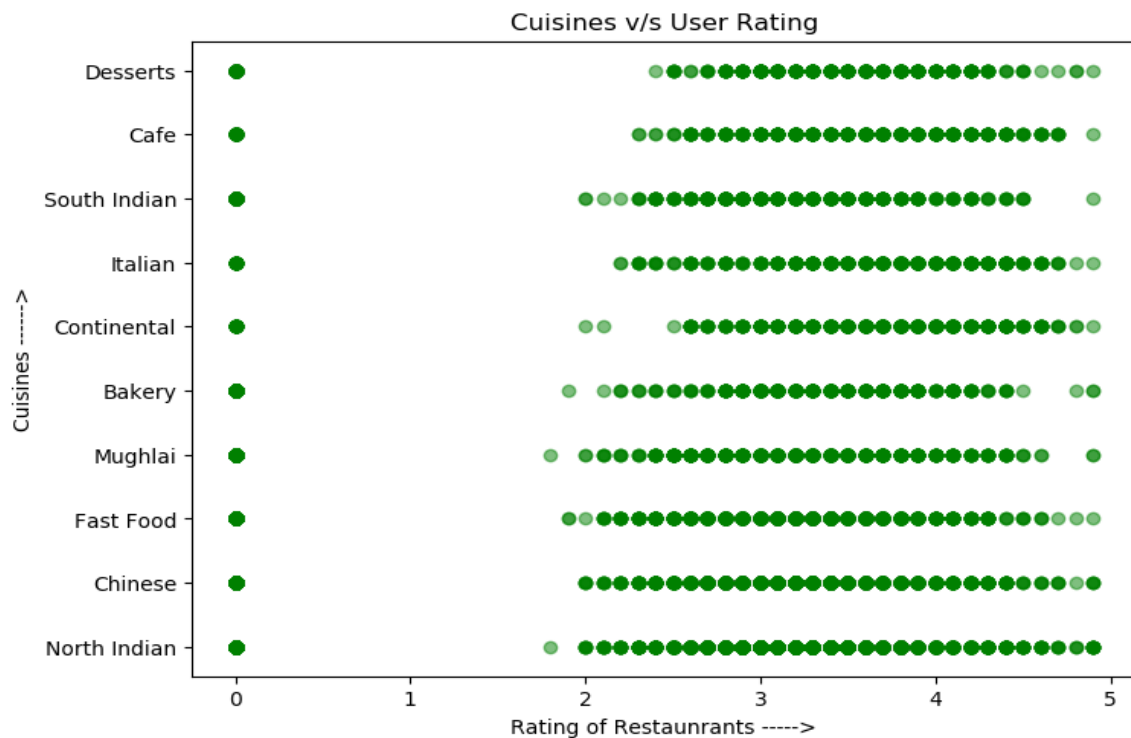
```
# fetching names of top 10 cuisines.
popular_cuisines=[]
for i in sorted_cuisines:
    popular_cuisines.append(i[0])

# fetching name and ratings of the particular cuisine of particular restaurant.
rating = []
cuisine = []
for i in popular_cuisines:
    for j, k in zip(df['Cuisines'], df['Aggregate rating']):
        if i in j:
            rating.append(k)
            cuisine.append(i)
```

- Finally, Plotting scatter graph between Cuisines v/s User Rating.

```
# plotting scatter graph for cuisine and its user rating.
plt.figure(num=None, figsize=(8, 6), dpi=100, facecolor='w', edgecolor='green')
plt.scatter(rating,cuisine, alpha = 0.5, c = 'green')
plt.title('Cuisines v/s User Rating')
plt.xlabel('Rating of Restaurants ----->')
plt.ylabel('Cuisines ----->')
plt.show()
```

Graph:-



- Second we plot a graph variation of aggregate rating with restaurants serving some specific cuisines(top 5).
- In code first we import matplotlib.pyplot for plotting the graph then find India restaurants only, data is taken for country code =1only i.e. for India.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

# considering only INDIAN Restraunts -
df = df[df['Country Code'] == 1]
```


- Now creating dictionary containing names of cuisines and their count and then sorting dictionary according to the count of cuisines and fetching top 10 cuisines.

```
# creating dict containing names of cuisines and their count.
cuisine=dict()
for i in df["Cuisines"]:
    a=i.split(',')
    for j in a:
        j=j.strip()
        cuisine[j]=cuisine.get(j, 0) + 1

# sorting dictionary according to the count of cuisines and fetching top 10 cuisines.
popular=sorted(cuisine.items(),key=lambda kv:kv[1], reverse=True)[:5]
```

- Then fetching names of top 10 cuisines and next fetching ratings of the particular cuisine of particular restaurant.

```
# fetching names of top 10 cuisines.
popular_cuisines=[]
for i in popular:
    popular_cuisines.append(i[0])

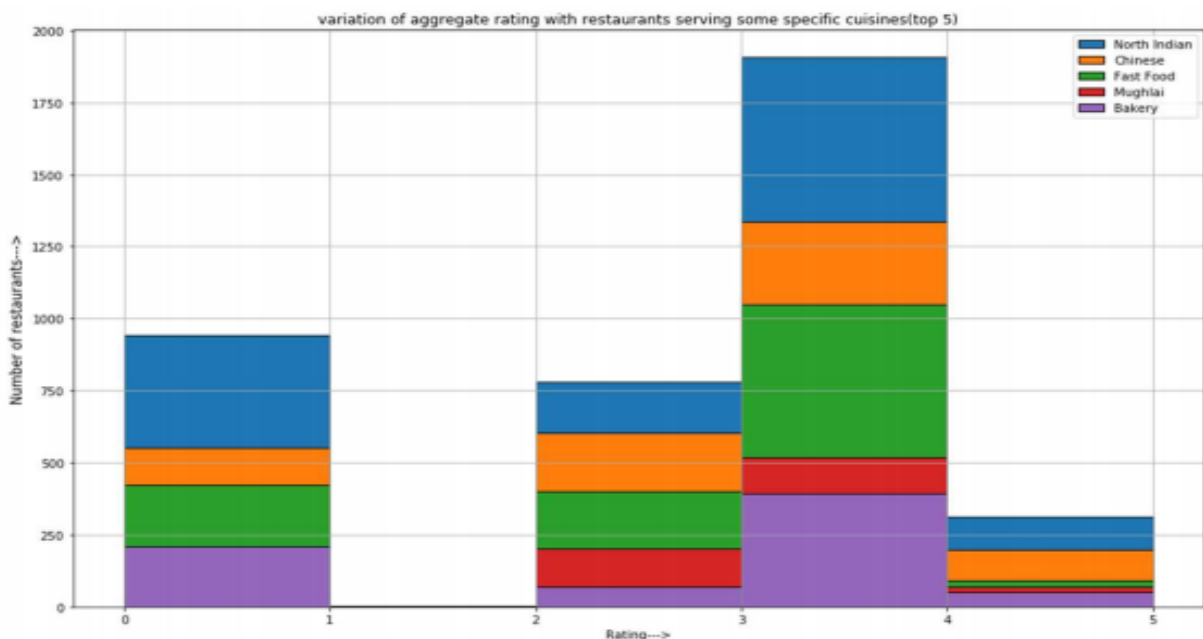
plt.figure(num=None, figsize=(16, 10))

# fetching ratings of the particular cuisine of particular restaurant.
for i in popular_cuisines:
    rating=[]
    for j, k in zip(df.Cuisines, df['Aggregate rating']):
        if i in j:
            rating.append(k)
```

- Finally, plotting histogram for each cuisine.

```
# plotting histogram for each cuisine .
plt.hist(rating, edgecolor='black', bins=[0, 1, 2, 3, 4, 5])
plt.xlabel('Rating-->')
plt.ylabel('Number of restaurants-->', size = 12)
plt.title('variation of aggregate rating with restaurants serving some specific cuisines(top 5)', size = 12)
plt.grid()
plt.legend(labels=popular_cuisines)
plt.show()
```

Graph:-



Results and Justification:-

As the result of the code and the graphs we can deduce that:-

1. There is no direct relation b/w restaurant serving a specific cuisine and its Rating.
2. For restaurants having cuisine North Indian , they have ratings b/w 2 – 5 spread evenly.
3. Cuisines like continental , Mughlai, Fast Food , South Indian Café etc have ratings mostly under 4.5 i.e. they mostly don't have excellent Ratings .
4. While 4 – 5 ratings are mostly given to North Indian, Chinese etc.
5. But at same time these cuisines also have very poor ratings also.

So, we can conclude that ratings vary for each cuisine , what matters is the taste and quality of food , there are cuisines that people prefer but if taste and quality of that cuisine is not up to standards then the ratings will be affected. So, the focus should be on quality on whichever cuisine Restaurant is serving.

Ques 2.2:- Find the weighted restaurant rating of each locality and find out the top 10 localities with more weighted restaurant rating?

Ques 2.2.1:- Weighted Restaurant Rating = $\frac{\sum (\text{number of votes} * \text{rating})}{\sum (\text{number of votes})}$.

- In code first we import matplotlib.pyplot for plotting the graph then find India restaurants only, data is taken for country code =1 only i.e. for India.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

# considering only INDIAN Restaurants -
df = df[df['Country Code'] == 1]
```

- Now grouping dataframe by Locality column and calculating Weighted Rating of Restaurant.

```
# grouping dataframe by Locality column and calculating Weighted Rating of Restaurant.
x = df.groupby(by = 'Locality').apply(lambda x: ((x['Votes'] * x['Aggregate rating']).sum()))
y = df.groupby(by = 'Locality').apply(lambda x: (x['Votes'].sum()))

weighted_rating = x/y
round(weighted_rating, 3)
```

- Then sorting Localities and their ratings by ratings and taking their top 10 localities and print top 10 localities with more weighted restaurant rating.

```
# sorting Localities and their ratings by ratings and taking their top 10 Localities.
sorted_weighted_rating = weighted_rating.sort_values(ascending = False)[0:10]

print('top 10 localities with more weighted restaurant rating ->')
print()
print(round(sorted_weighted_rating, 2))
```

Output:-

top 10 localities with more weighted restaurant rating ->

Locality	
Aminabad	4.90
Hotel Clarks Amer, Malviya Nagar	4.90
Friends Colony	4.89
Powai	4.84
Kirlampudi Layout	4.82
Express Avenue Mall, Royapettah	4.80
Deccan Gymkhana	4.80
Banjara Hills	4.72
Sector 5, Salt Lake	4.71
Riverside Mall, Gomti Nagar	4.70

dtype: float64

Result:-

We got the desired outcome i.e. top 10 localities sorted according to the ratings.

Ques 3:- Visualization

Ques 3.1:- Plot the bar graph top 15 restaurants have a maximum number of outlets.

- In code first we import matplotlib.pyplot for plotting the graph then find India restaurants only, data is taken for country code =1 only i.e. for India.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

# considering only INDIAN Restaurants -
df = df[df['Country Code'] == 1]
```

- Now counting No. of outlets for a particular restaurant using value_counts() function and then sorting those restaurants by the number of those restaurants and fetching top 15 of those.

```
# restaurants and the count of those restaurants.
restaurants = df['Restaurant Name'].value_counts()

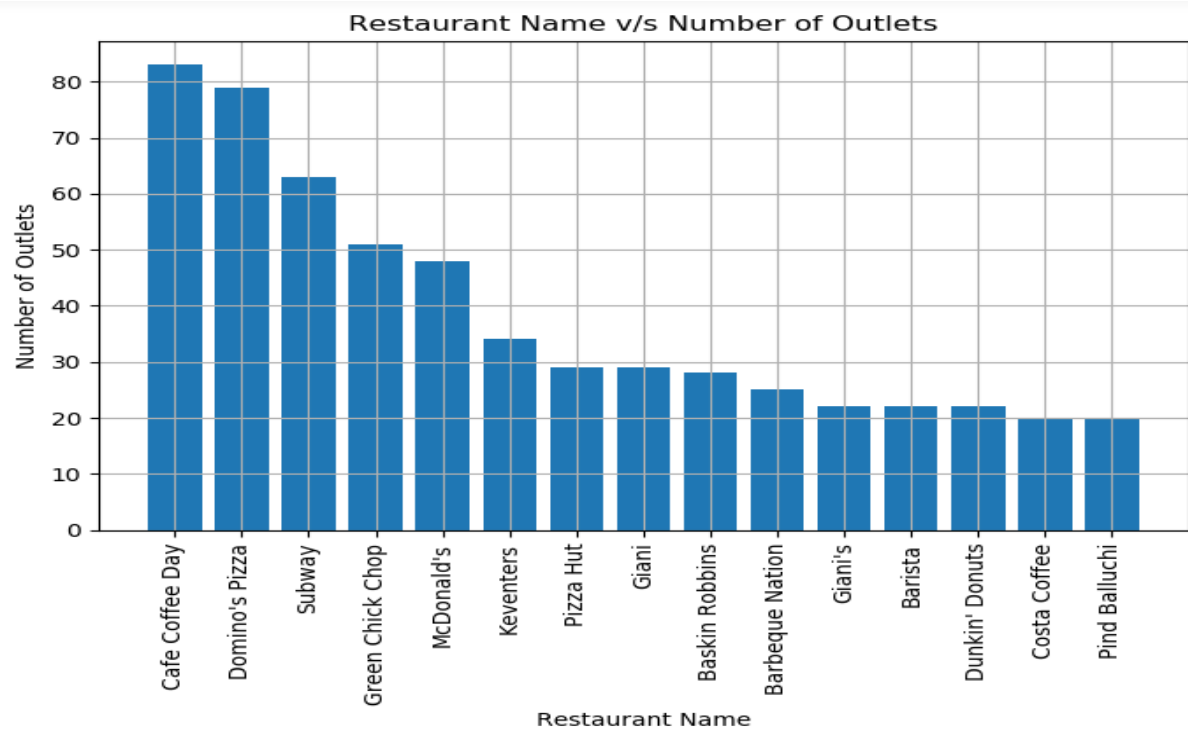
# sorting those restaurants by the number of those restaurants and fetching top 15 of those.
sorted_values = restaurants.sort_values(ascending = False)[0:15]

x = sorted_values.index
y = sorted_values.values
```

- Finally, Plotting bar graph between Restaurant Name v/s Number of Outlets.

```
# plotting bar graph
plt.figure(num=None, figsize=(8, 5), dpi=100, facecolor='w', edgecolor='green')
plt.bar(x,y)
plt.grid()
plt.title('Restaurant Name v/s Number of Outlets')
plt.xlabel('Restaurant Name')
plt.ylabel('Number of Outlets')
plt.xticks(rotation = 90)
plt.show()
print(sorted_values)
```

Graph:-



Output:-

```
Cafe Coffee Day      83
Domino's Pizza       79
Subway               63
Green Chick Chop     51
McDonald's           48
Keventers            34
Pizza Hut            29
Giani                29
Baskin Robbins        28
Barbeque Nation       25
Giani's              22
Barista              22
Dunkin' Donuts        22
Costa Coffee          20
Pind Balluchi         20
Name: Restaurant Name, dtype: int64
```

Result:-

We got the desired outcome i.e. top 15 Restaurants with most number of outlets.

Ques 3.2:- Plot the histogram of aggregate rating of restaurant(drop the unrated restaurant).

- In code first we import matplotlib.pyplot for plotting the graph then find India restaurants only, data is taken for country code =1 only i.e. for India.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

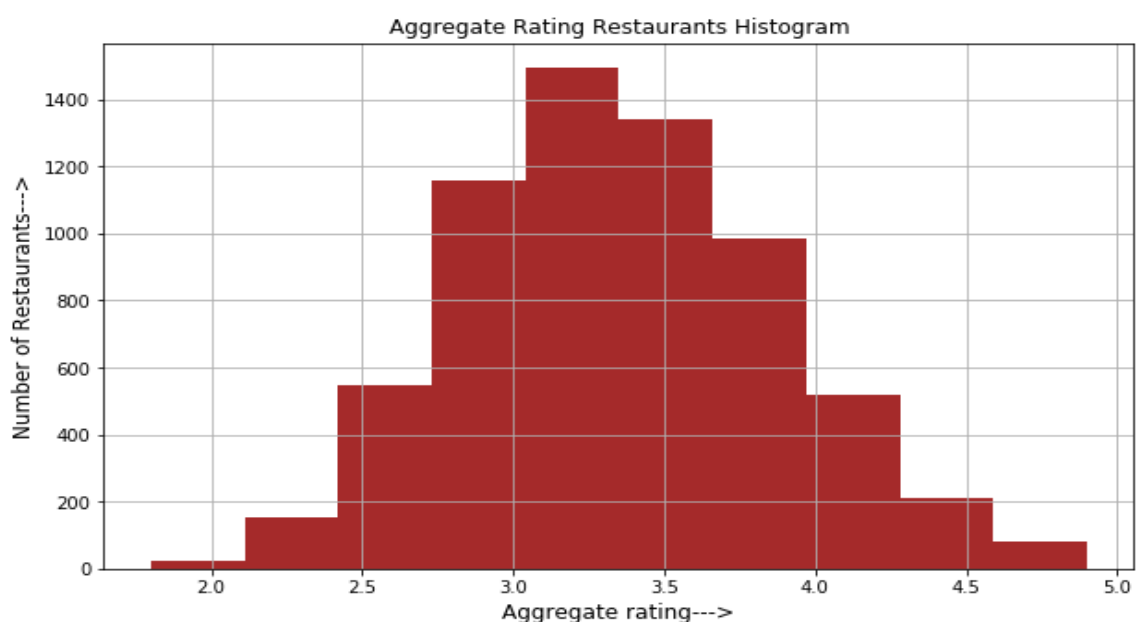
# considering only INDIAN Restaurants -
df = df[df['Country Code'] == 1]
```

- Now dropping unrated restaurants and then plotting Aggregate Rating Restaurants Histogram.

```
# dropping unrated restaurants.
df.drop(df[df['Rating text'] == 'Not rated'].index , inplace = True)

# plotting histogram.
plt.figure(num=None, figsize=(10, 6), facecolor='w')
plt.hist(df['Aggregate rating'] , facecolor='brown', align='mid')
plt.grid()
plt.title('Aggregate Rating Restaurants Histogram', size = 12)
plt.xlabel('Aggregate rating--->', size=12)
plt.ylabel('Number of Restaurants--->', size=12)
plt.show()
```

Graph:-



Result and Justification:-

We got the desired histogram.

Ques 3.3:- Plot the bar graph top 10 restaurants in the data with the highest number of votes.

- In code first we import matplotlib.pyplot for plotting the graph then find India restaurants only, data is taken for country code =1 only i.e. for India.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

# considering only INDIAN Restaurants -
df = df[df['Country Code'] == 1]
```

- Then grouping data by restaurants and taking sum of votes of those same restaurants and then now sorting the restaurants according to the count of the votes and fetching top 10 of those.

```
# grouping data by restaurants and taking sum of votes of those same restaurants
# now sorting the restaurants according to the count of the votes and fetching top 10 of those.
votes = df.groupby("Restaurant Name")["Votes"].sum().sort_values(ascending=False)[0:10]

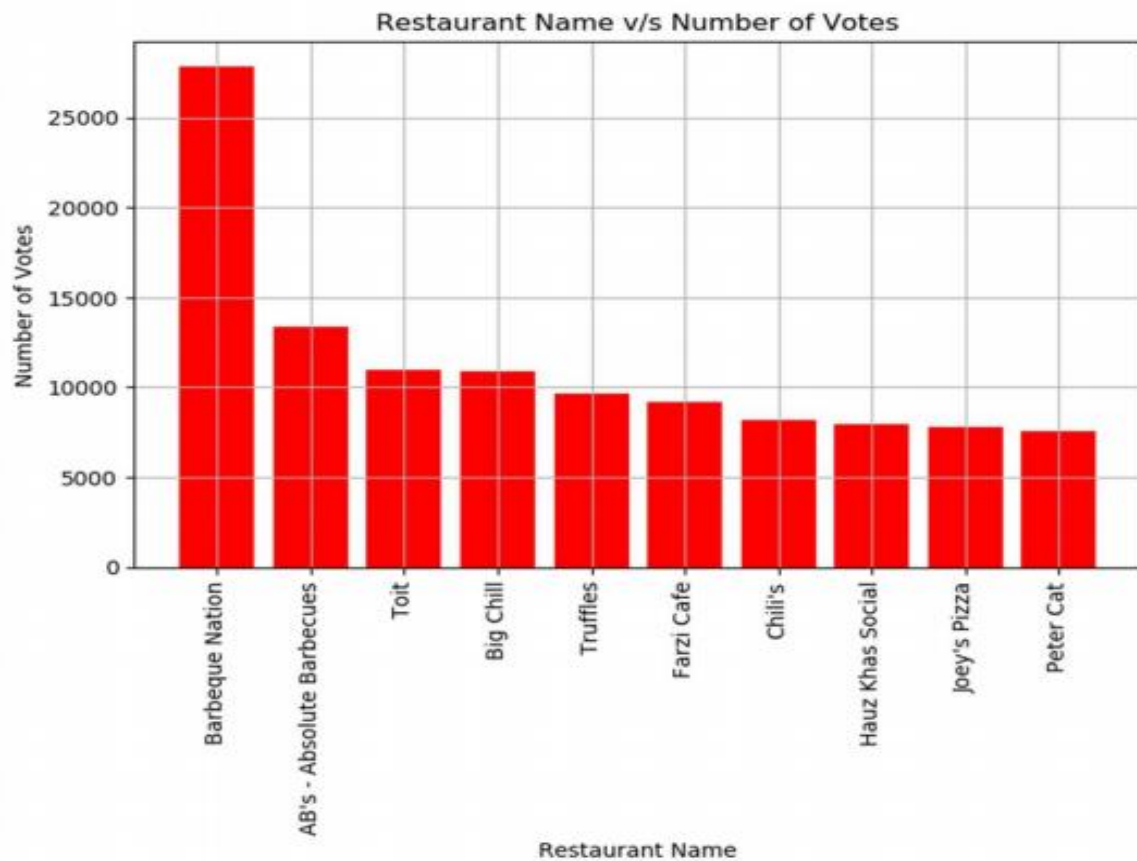
x = votes.index
y = votes.values
```

- Finally, Plotting bar graph between Restaurant Name v/s Number of Votes.

```
# plotting bar graph.
plt.figure(num=None, figsize=(8, 5), dpi=100, facecolor='w', edgecolor='green')
plt.bar(x,y, color = 'red')
plt.grid()
plt.title('Restaurant Name v/s Number of Votes')
plt.xlabel('Restaurant Name')
plt.ylabel('Number of Votes')
plt.xticks(rotation = 90)
plt.show()

print(votes)
```

Graph:-



Output:-

```
Restaurant Name
Barbeque Nation      27835
AB's - Absolute Barbecues  13400
Toit                  10934
Big Chill             10853
Truffles              9682
Farzi Cafe            9189
Chili's               8156
Hauz Khas Social      7931
Joey's Pizza          7807
Peter Cat             7574
Name: Votes, dtype: int64
```

Result and Justification:-

We got the desired graph and output i.e. for top 10 restaurants in the data with the highest number of votes.

Ques 3.4:- Plot the pie graph of top 10 cuisines present in restaurants in the USA.

- In code first we import matplotlib.pyplot for plotting the graph then fetching data USA only, data is taken for country code = 216 only i.e. for USA.

```
import pandas as pd
import numpy as np

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

# considering only UNITED STATES Restraunts -
df = df[df['Country Code'] == 216]
```

- Now resetting index and then filling na value with Not Known and then making dictionary for different cuisines in USA and their counts.

```
# resetting index and then filling na value with Not Known.
df.reset_index(drop = True, inplace = True)
df['Cuisines'].fillna('Not known',inplace = True)
df['Cuisines'] = df['Cuisines'].str.split(', ')

# making dictionary for different cuisines in USA and their counts
usa_cuisines = {}
for i in df['Cuisines'] :
    for j in i:
        usa_cuisines[j] = usa_cuisines.get(j, 0) + 1
```

- Then sorting dictionary by count and fetching top 10 cuisines and then Running loop cuisines and count of each cuisine. 8) Sorting the list of cuisines according to their number of counts and fetching top 10 cuisines.

```
# sorting dictionary by count and fetching top 10 cuisines.
top_cuisines = sorted(usa_cuisines.items(), key = lambda kv:kv[1], reverse = True)[:10]

x = []
label = []

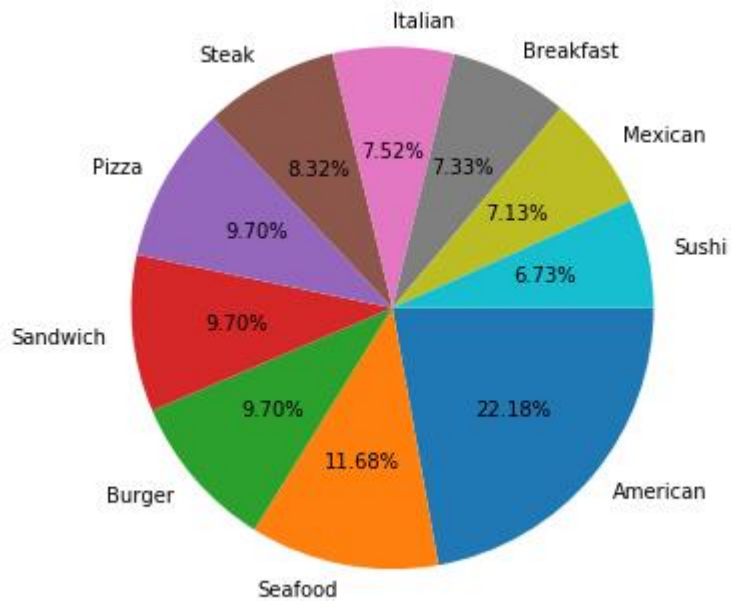
print('Top 10 cuisines served by maximum number of restaurants in USA - ')
print()

# making List of cuisines names and their counts.
for i in top_cuisines:
    print(i[0],i[1])
    x.append(i[1])
    label.append(i[0])
```

- Finally, Plotting pie chart.

```
# plotting pie chart.
plt.figure(num=None, figsize=(6, 6), facecolor='w', edgecolor='k')
plt.pie(x, labels = label, autopct='%.2f%%', counterclock=False)
plt.show()
```


Graph:-



Output:-

Top 10 cuisines served by maximum number of restaurants in USA -

American 112
Seafood 59
Burger 49
Sandwich 49
Pizza 49
Steak 42
Italian 38
Breakfast 37
Mexican 36
Sushi 34

Result and Justification:-

We got the desired graph and output i.e. for top 10 cuisines present in restaurants in the USA.

Ques 3.5:- Plot the bubble graph of a number of Restaurants present in the city of India and keeping the weighted restaurant rating of the city in a bubble.

- In code first we import matplotlib.pyplot for plotting the graph then find India restaurants only, data is taken for country code =1only i.e. for India.

```
import pandas as pd
import numpy as np

file = pd.read_csv('E:\zomato.csv', encoding = 'iso-8859-1')
df = file.copy()

# considering only UNITED STATES Restraunts -
df = df[df['Country Code'] == 216]
```

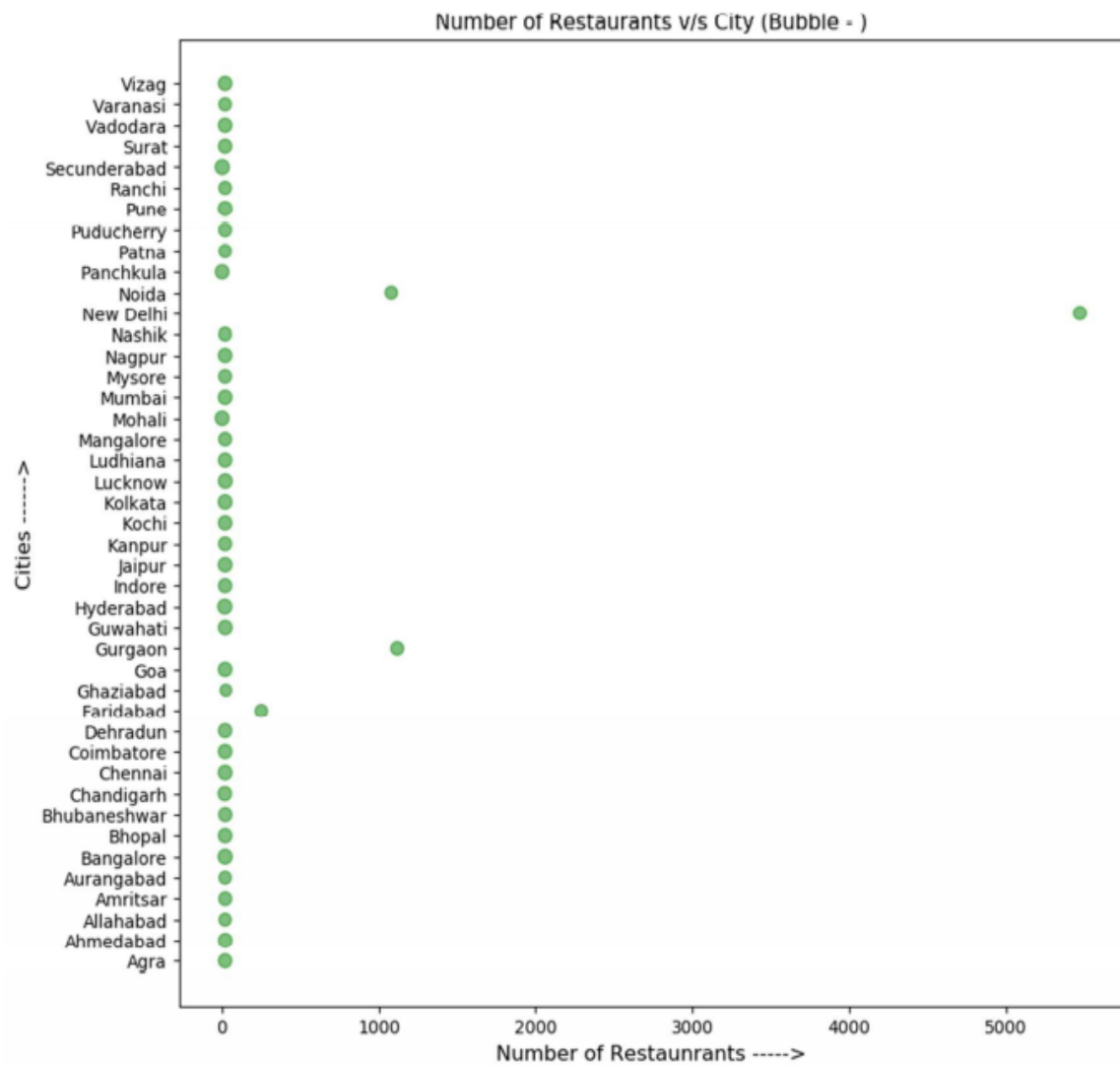
- Now Grouping the data according to the Cities and then counting the number of restaurants in that city by applying count() function on restaurant name.

```
# grouping dataframe by City and calculating Weighted rating.
x = df.groupby(by = 'City').apply(lambda x: ((x['Votes'] * x['Aggregate rating']).sum()))
y = df.groupby(by = 'City').apply(lambda x: (x['Votes'].sum()))
restaurant_count = df.groupby(by = 'City').count()['Restaurant Name']
weighted_rating = x/y
```

- Finally, Plotting bubble graph and keeping weighted rating as size of the bubble.

```
# plotting bubble graph and keeping weighted rating as size of the bubble.
plt.figure(num=None, figsize=(10, 10), dpi=100, facecolor='w', edgecolor='green')
plt.scatter(restaurant_count.values, restaurant_count.index, s = weighted_rating*15, alpha = 0.5, c = 'green')
plt.title('Number of Restaurants v/s City (Bubble -)')
plt.xlabel('Number of Restaunrants ----->', size = 12)
plt.ylabel('Cities ----->', size = 12)
plt.show()
```

Graph:-



Result and Justification:-

We got the desired graph of a number of Restaurants present in the city of India and keeping the weighted restaurant rating of the city in a bubble.