PROJECT: ZOMATO API - II

Zomato Dataset Details

Details of zomato.csv -

1. ***Restaurant Id** : Unique id of every restaurant across various cities of the world*
2. ***Restaurant Name** : Name of the restaurant*
3. ***Country Code** : Country in which restaurant is located*
4. ***City** : City in which restaurant is located*
5. ***Address** : Address of the restaurant*
6. ***Locality** : Location in the city*
7. ***Locality Verbose** : Detailed description of the locality*
8. ***Longitude** : Longitude coordinate of the restaurant's location*
9. ***Latitude** : Latitude coordinate of the restaurant's location*
10. ***Cuisines** : Cuisines offered by the restaurant*
11. ***Average Cost for two** : Cost for two people in different currencies*
12. ***Currency** : Currency of the country*
13. ***Has Table booking** : yes/no*
14. ***Has Online delivery** : yes/ no*
15. ***Is delivering** : yes/ no*
16. ***Switch to order menu** : yes/no*
17. ***Price range** : range of price of food*
18. ***Aggregate Rating** : Average rating out of 5*
19. ***Rating color** : depending upon the average rating color*
20. ***Rating text** : text on the basis of rating of rating*
21. ***Votes** : Number of ratings casted by people*

Country Codes :

Country Code	***Country***
1	*India*
14	*Australia*
30	*Brazil*
37	*Canada*
94	*Indonesia*
148	*New Zealand*
162	*Phillipines*
166	*Qatar*
184	*Singapore*
189	*South Africa*
191	*Sri Lanka*
208	*Turkey*
214	*UAE*
215	*United Kingdom*
216	*United States*

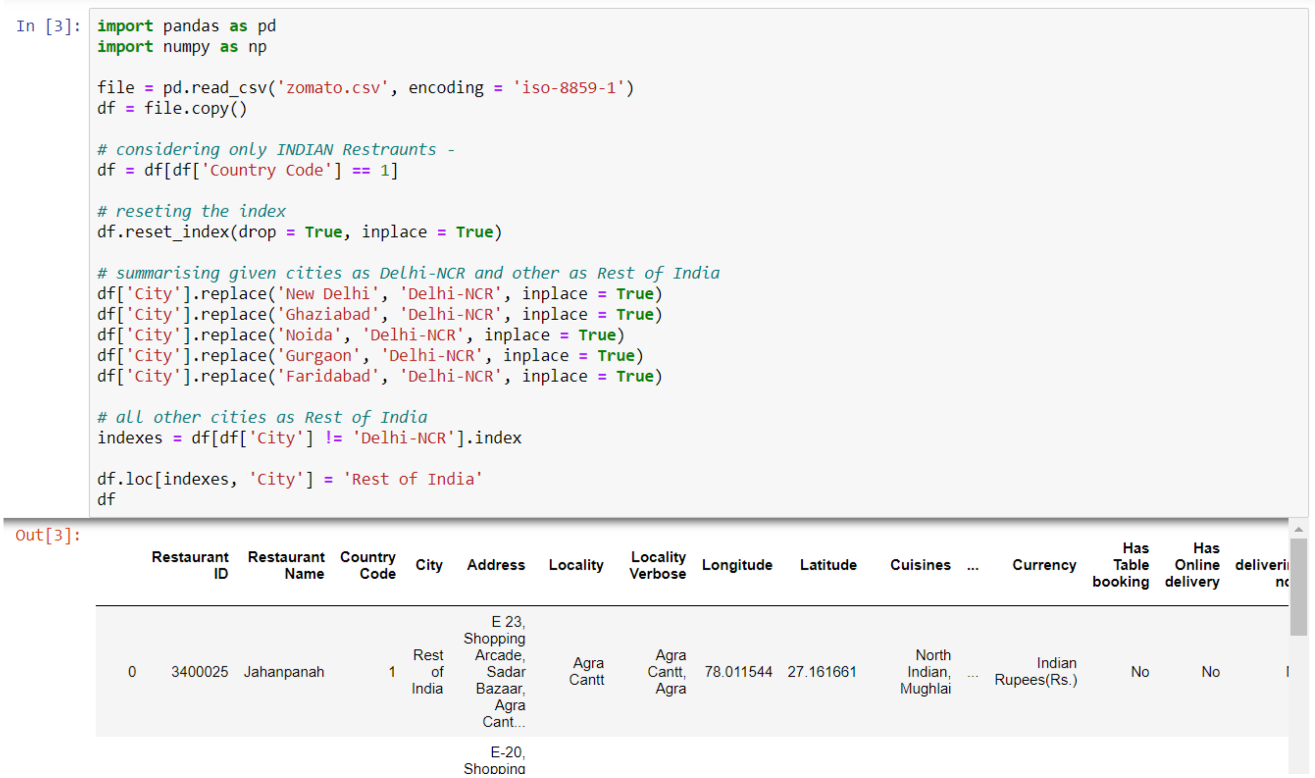
PROJECT: ZOMATO API - II

QUESTIONS

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.

CODE SNIPPETS –



```
In [3]: import pandas as pd
        import numpy as np

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

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

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

        # 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
```

Out[3]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking	Has Online delivery	deliveri nc
0	3400025	Jahanpanah	1	Rest of India	E 23, Shopping Arcade, Sadar Bazaar, Agra Cant...	Agra Cantt	Agra Cantt, Agra	78.011544	27.161661	North Indian, Mughlai	...	Indian Rupees(Rs.)	No	No	I
					E-20, Shopping										

CODE EXPLANATION –

1) *To find the divide our data according to cities , I have included the libraries required that are numpy , pandas , then –*
2) *For India restaurants only , data is taken for Country Code = 1 only i.e. for India.*
3) *Replacing all the names of the Cities as mentioned in question to Delhi-NCR.*
4) *And having all the cities remaining to be called as City – 'Rest of India'. 5) Resetting Index also*

RESULT –

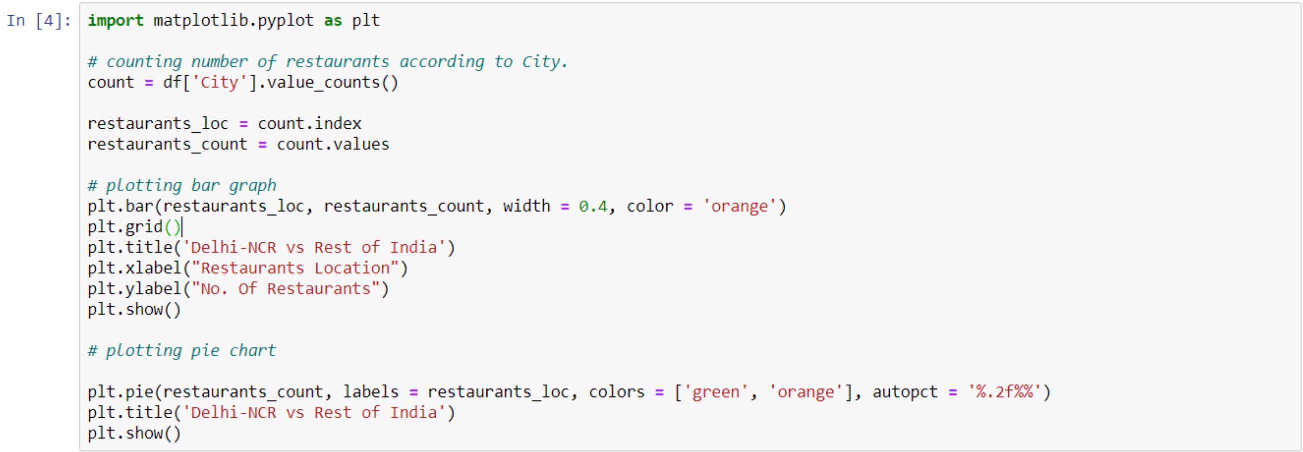
Our Data is now summarised accordingly and we'll use this data in further Questions of Part 1 since, our data was highly skewed towards the Delhi-NCR restaurants.

PROJECT: ZOMATO API - II

QUES. 1.1)

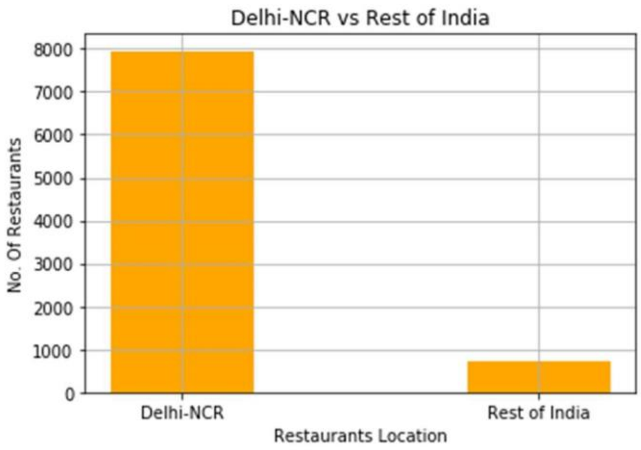
Plot the bar graph of number of restaurants present in Delhi NCR vs Rest of India.

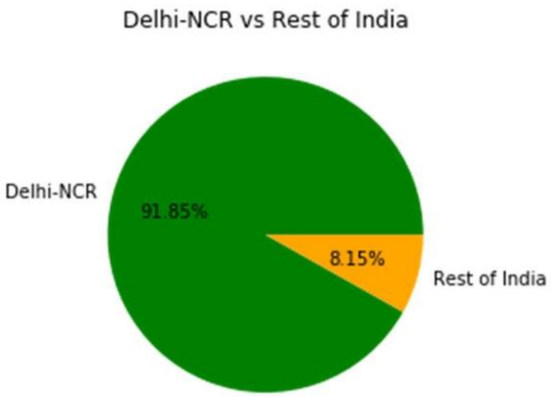
CODE SNIPPETS –



```python
import matplotlib.pyplot as plt

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

restaurants_loc = count.index
restaurants_count = count.values

# 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 –





CODE EXPLANATION –

1) *To Plot the graph imported library matplotlib.pyplot then –*
2) *Using value_counts(), to count number of Restaurants according to the location i.e. Delhi-NCR or Rest of India.*
3) *Plotted Bar graph and pie chart for the same.*

RESULT AND JUSTIFICATION –

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

Delhi-NCR restaurants-7947
Rest of India restaurants-705

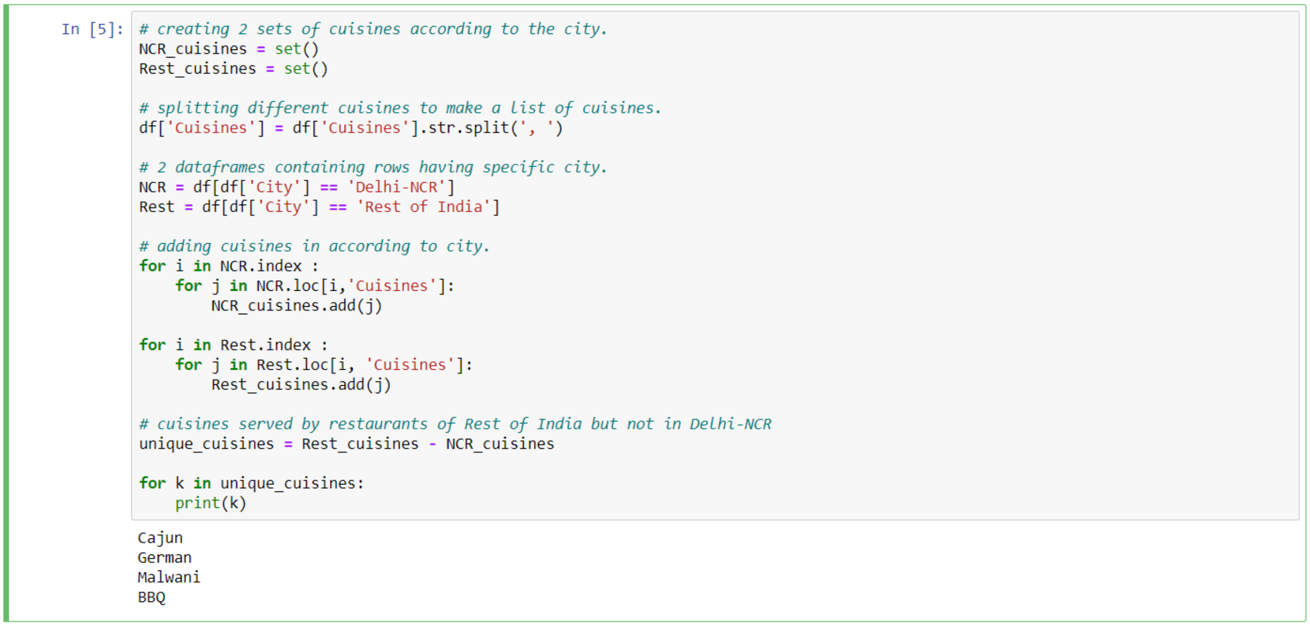
i.e. according to the pie chart we can see , in the dataset we have for Indian Restaurants , it contains 91.85% of data for Delhi-NCR restaurants and 8.15% for Rest of India.

PROJECT: ZOMATO API - II

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 these cuisines are actually not served in restaurants of Delhi-NCR or just it due to incomplete dataset.

CODE SNIPPETS –



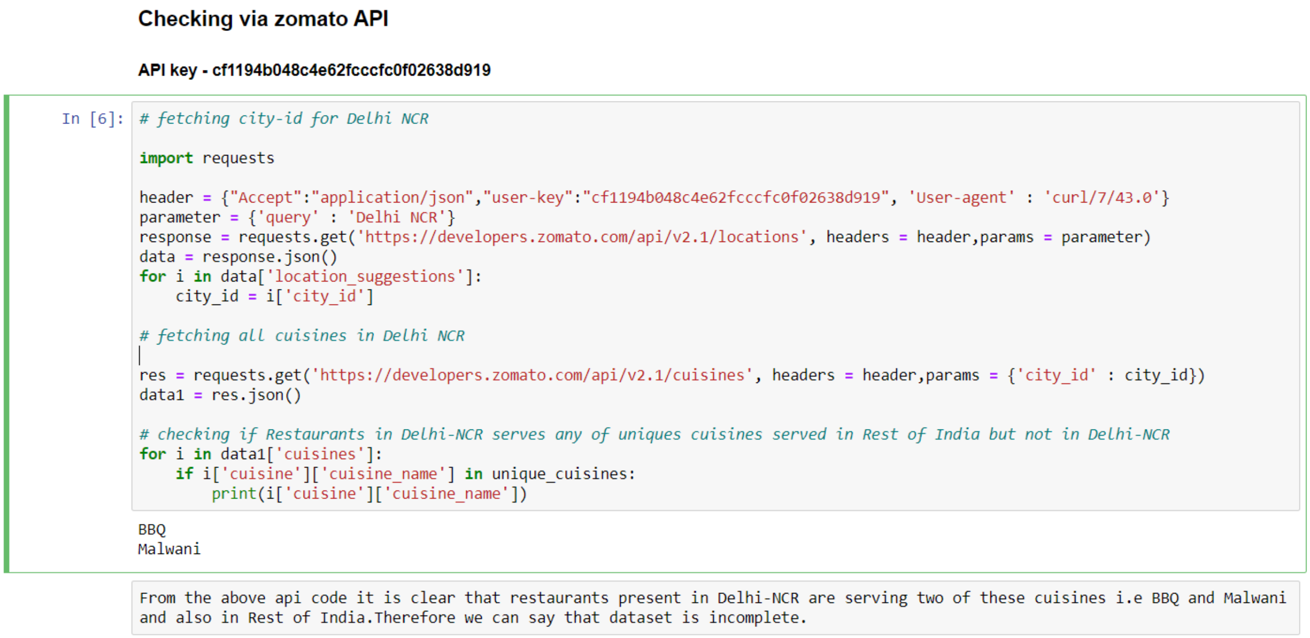
```
In [5]: # creating 2 sets of cuisines according to the city.
        NCR_cuisines = set()
        Rest_cuisines = set()

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

        # 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)

        # 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)

        Cajun
        German
        Malwani
        BBQ
```

CODE EXPLANATION –

1) *Created 2 sets that will contain the cuisines in Delhi-NCR and in Rest of India.*
2) *Since, a restaurant can serve many different cuisines which is given in a string to us.*
3) *So, to count for each cuisine , we split the cuisine column in df by using split() function by (,).*
4) *Made 2 new Dataframes containing data for each City.*
5) *Running loops to add the cuisines to the sets of each city.*
6) *We get cuisines for Delhi and for Rest of India.*
7) *Now subtracting Delhi-NCR cuisines from Rest of India cuisines to get cuisines served in Rest of India But Not in Delhi-NCR.*

PROJECT: ZOMATO API - II

CHECKING USING API –



Checking via zomato API

API key - cf1194b048c4e62fcccfc0f02638d919

In [6]:
```
# fetching city-id for Delhi NCR

import requests

header = {"Accept":"application/json","user-key":"cf1194b048c4e62fcccfc0f02638d919", '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']

# 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()

# checking if Restaurants in Delhi-NCR serves any of uniques 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'])
```

```
BBQ
Malwani
```

```
From the above api code it is clear that restaurants present 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.
```

CODE EXPLANATION –

1) *we are creating are header in which accept header and user key is passed for authentication*
2) *Now we are fetching details from the Zomato API to check whether the Cuisines stored in diff are served in Delhi restaurant or not.*
3) *data is storing the converted data (from json to python)*
4) *We are converting data with help of json() function*
5) *Now we are using for loop to find whether the cuisine_name is present in cuisines or not*

RESULT AND JUSTIFICATION –

From the above API code it is clear that restaurants present 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 the 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.

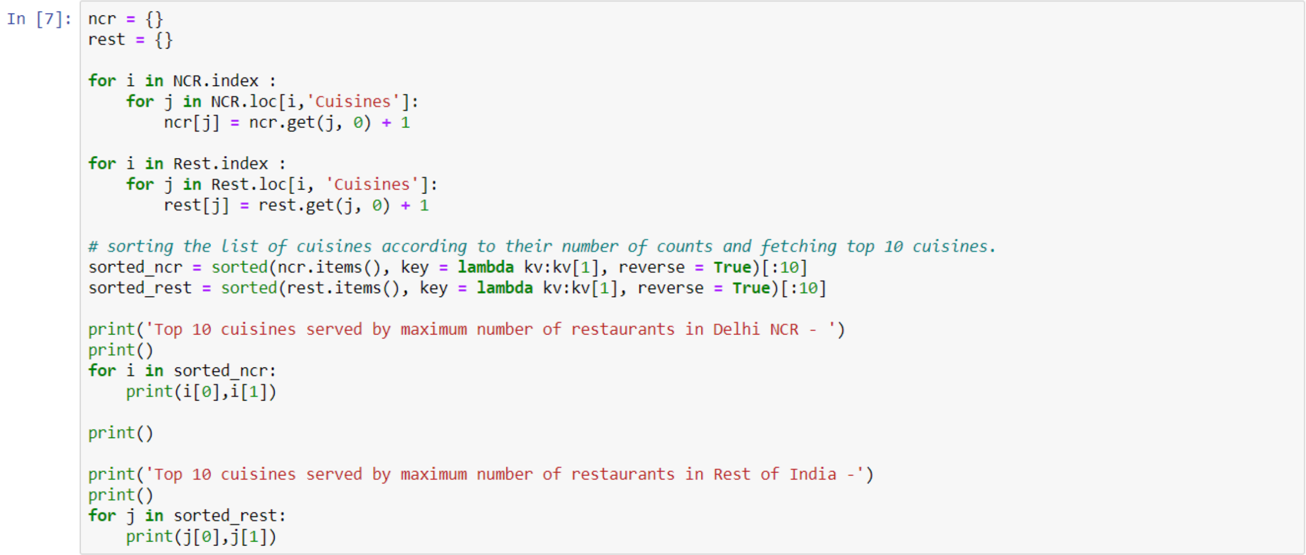
So, it is due to incomplete dataset.

PROJECT: ZOMATO API - II

QUES. 1.3)

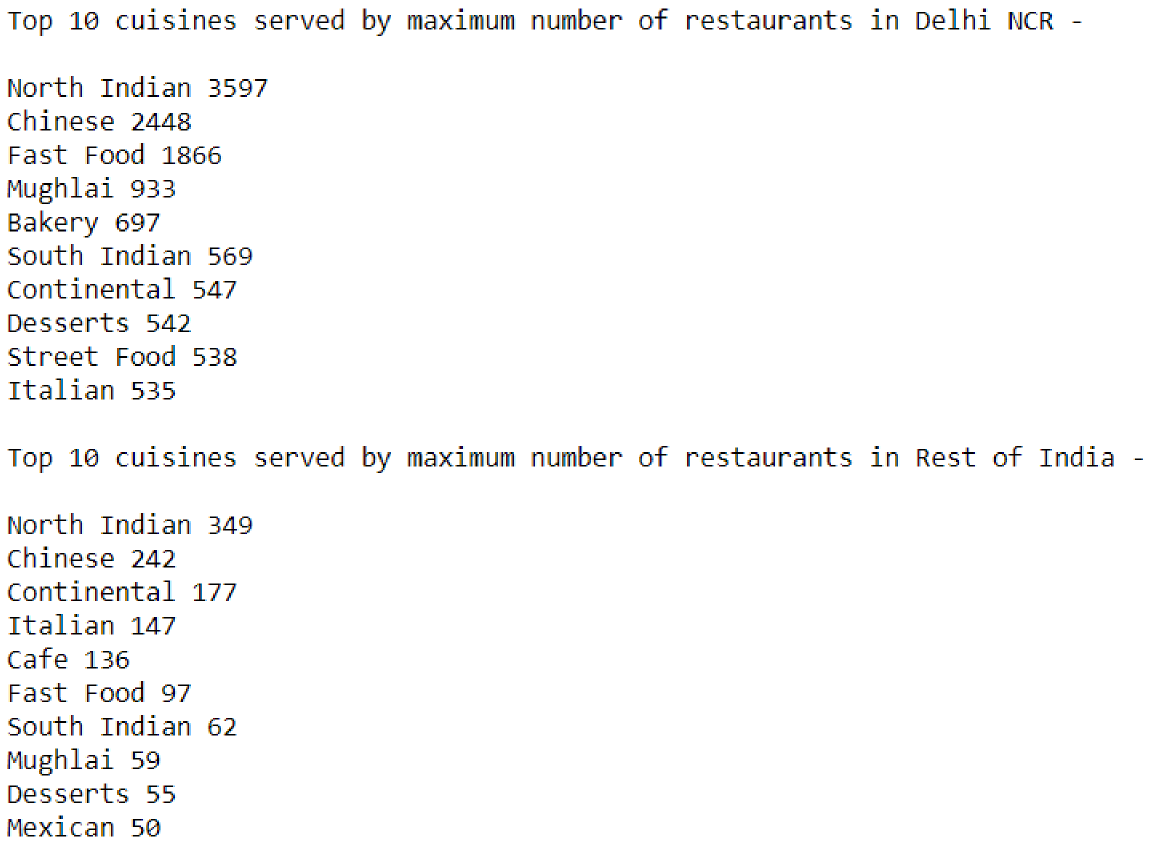
Find the top 10 cuisines served by maximum number of restaurants in Delhi NCR and rest of India.

CODE SNIPPETS –



```
In [7]: 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

        # 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
```

PROJECT: ZOMATO API - II

CODE EXPLANATION –

1) *Creating 2 dictionaries one for each Delhi-NCR cuisines and for Rest of India.*
2) *Now, Running loops in both to fetch cuisines and count of each cuisine.*
3) *Sorting the list of cuisines according to their number of counts and fetching top 10 cuisines.*
4) *Printing Top 10 cuisines and their counts for both Delhi – NCR and Rest of India.*

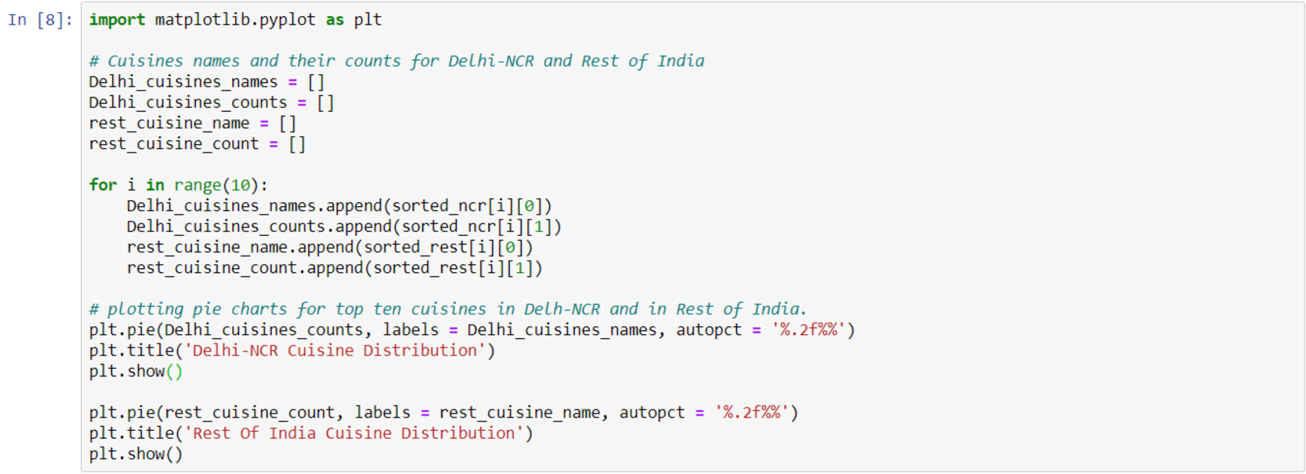
RESULT –

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

QUES. 1.3)

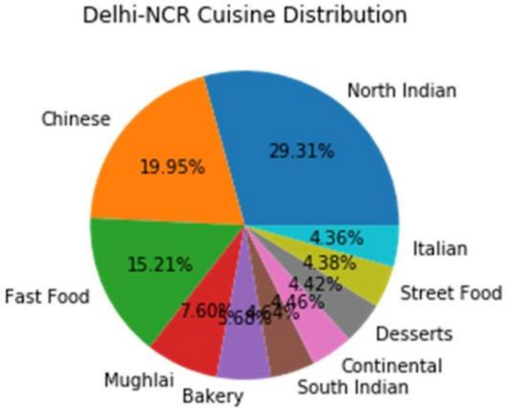
Write a short-detailed analysis of how cuisine served is different from Delhi NCR to Rest of India. Plot the suitable graph to explain your inference.

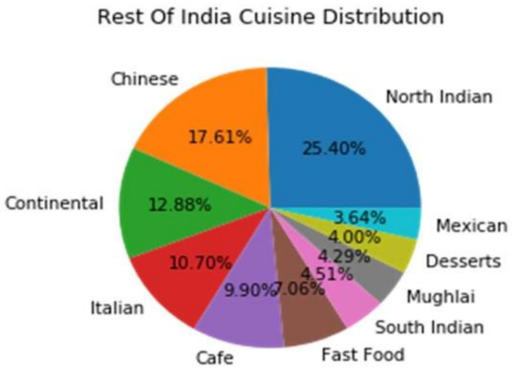
CODE SNIPPETS –



```
In [8]: 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])

        # plotting pie charts for top ten cuisines in Delh-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 –





ANALYSIS AND OUTCOME–

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 whereas it's 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.*

PROJECT: ZOMATO API - II

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)

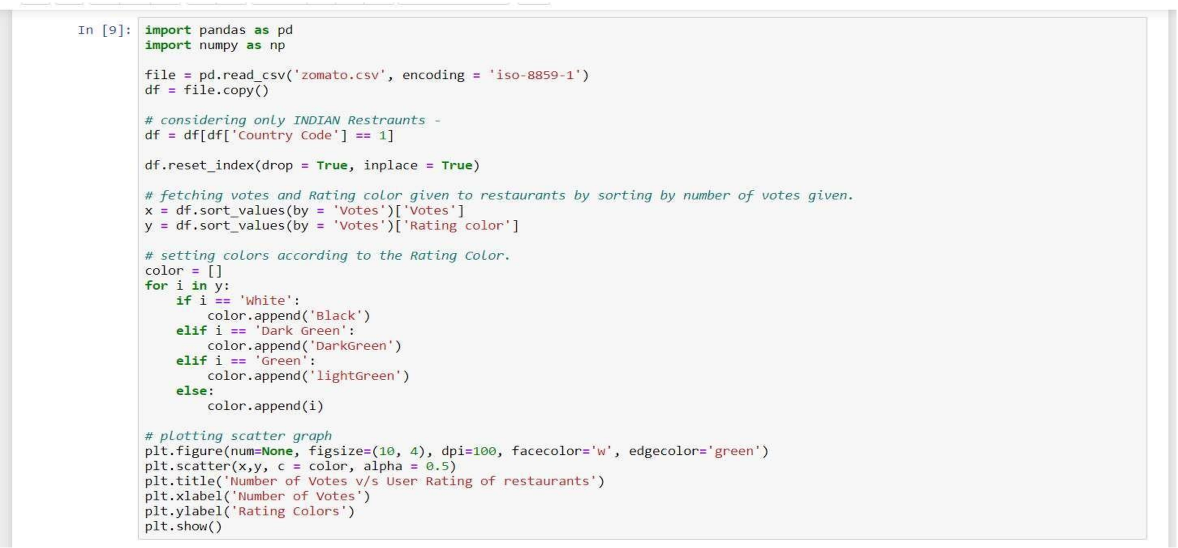
Write a short detail analysis of how the rating is affected by restaurant due following features: Plot a suitable graph to explain your inference.

QUES. 2.1.1)

Number of Votes given Restaurant.

CODE SNIPPETS –

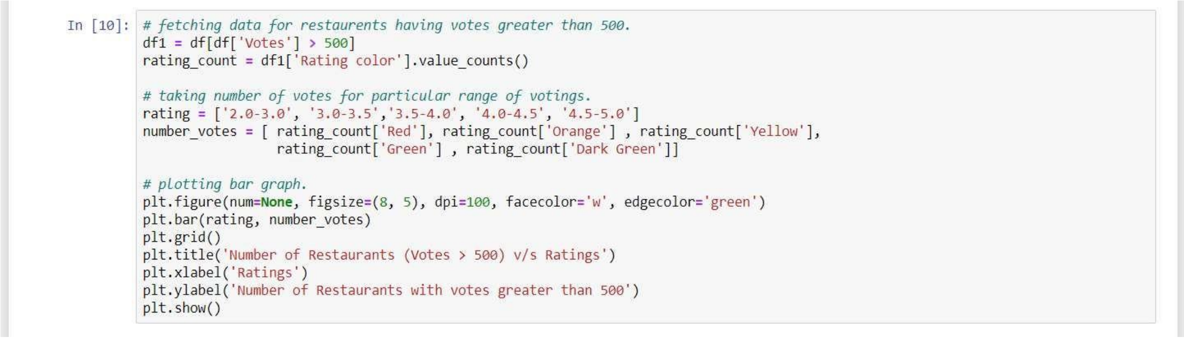
1)



In [9]:

```
import pandas as pd
import numpy as np

file = pd.read_csv('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)

# 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)

# 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()
```

2)

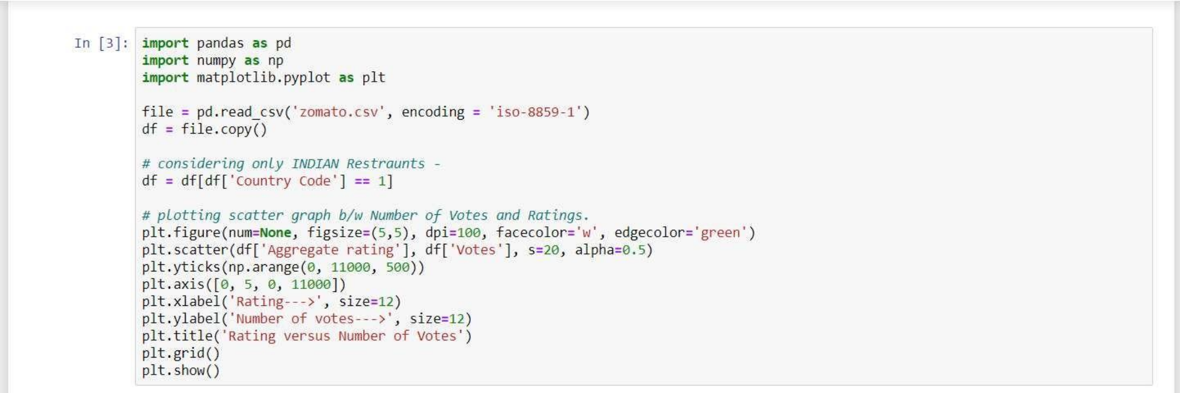


In [10]:

```
# fetching data for restaurents 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']]

# 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()
```

PROJECT: ZOMATO API - II

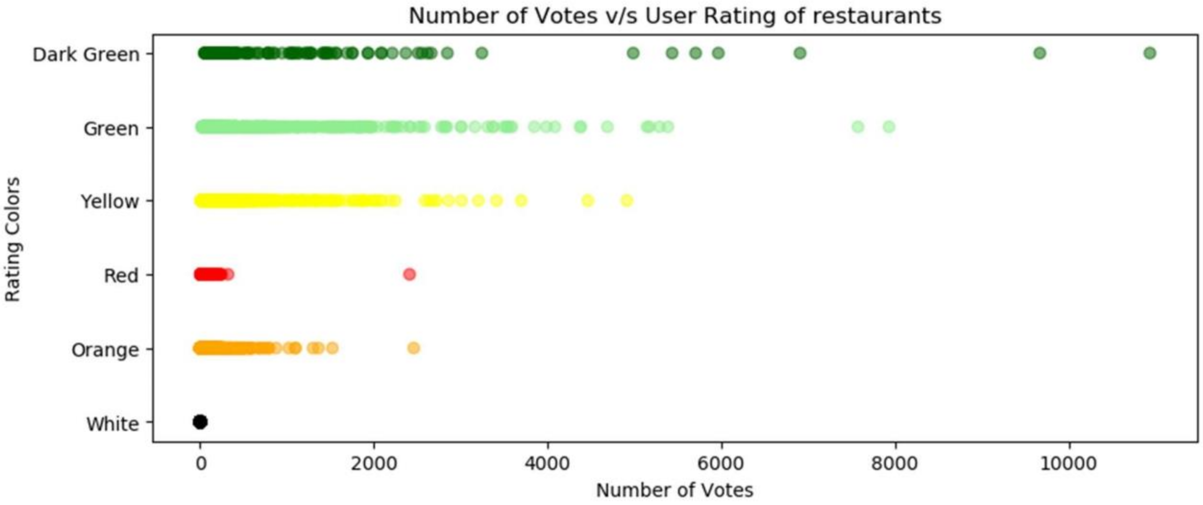
3)



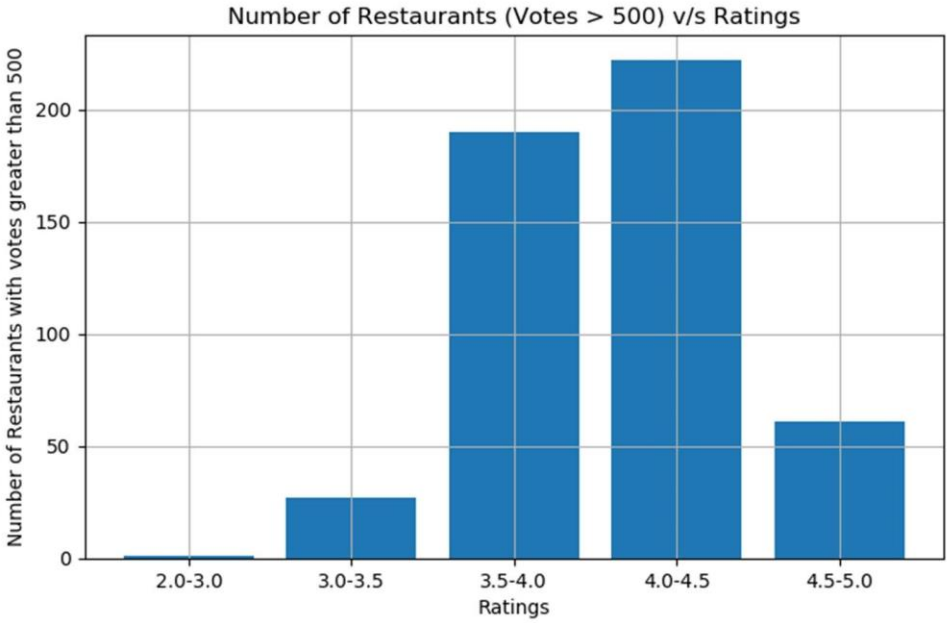
```
In [3]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt

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

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

        # 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.yticks(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 versus Number of Votes')
        plt.grid()
        plt.show()
```

GRAPHS –

1) *Scatter Plot according to the user rating color for number of votes given .*

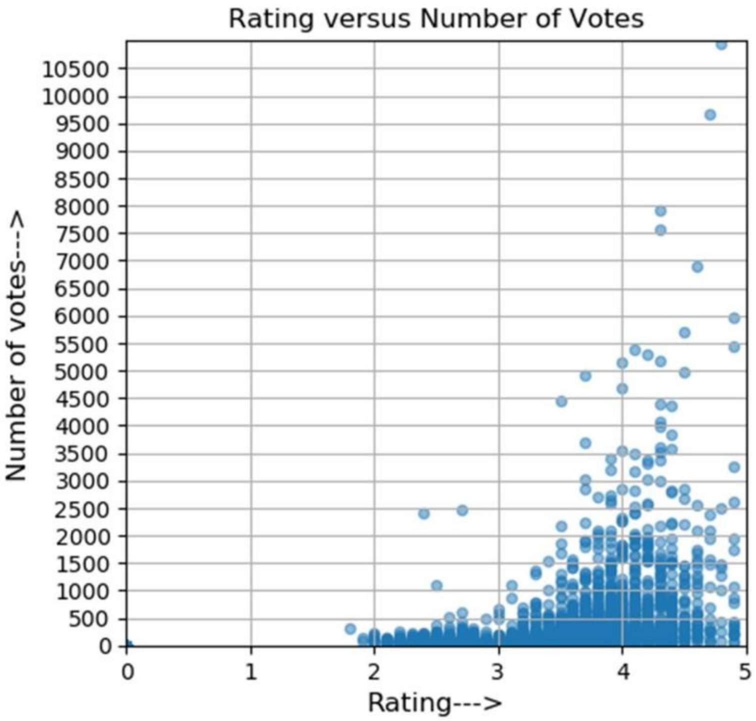


2) *Bar graph for votes > 500 which were showing some reflection of votes on Restaurant Ratings.*



PROJECT: ZOMATO API - II

3) *Simple Scatter plot for rating and Number of votes.*



ANALYSIS 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 .

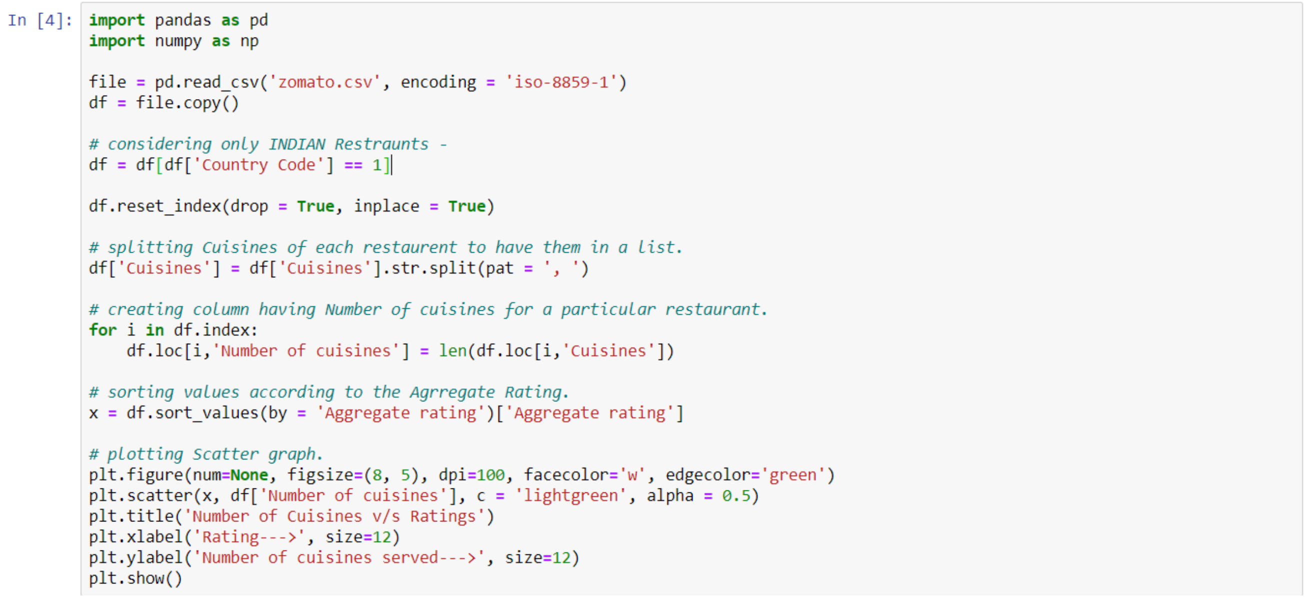
But for Restaurants for votes < 8000 there, is no such deduction as they have good as well as bad reviews too so, ratings get neutralized b/w 3.5 – 4.5.

PROJECT: ZOMATO API - II

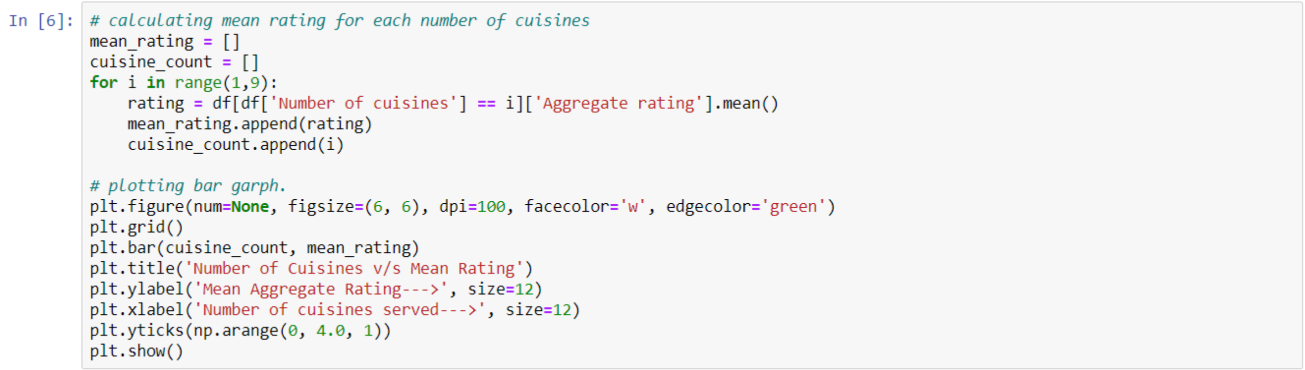
QUES. 2.1.2)

Restaurant serving a greater number of cuisines.

CODE SNIPPETS –



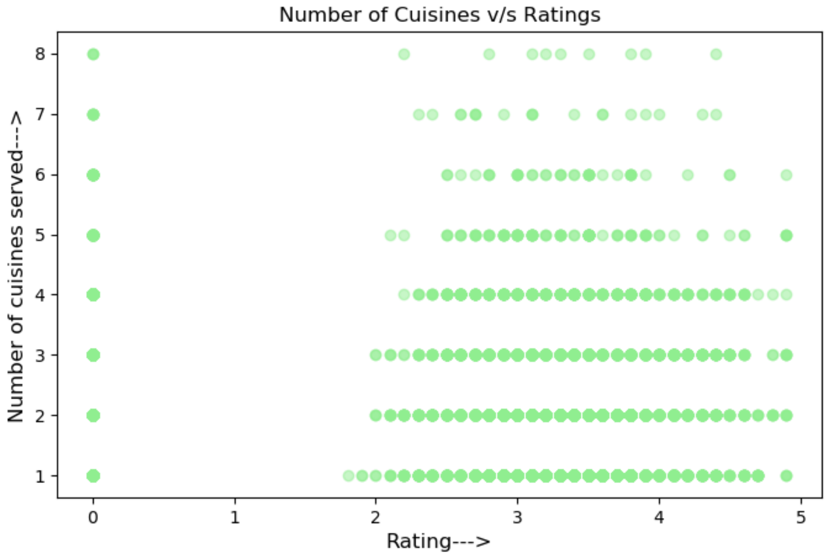
```
In [4]: import pandas as pd
        import numpy as np

        file = pd.read_csv('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)

        # splitting Cuisines of each restaurent 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'])

        # sorting values according to the Agrregate 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 = 'lightgreen', 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()
```



```
In [6]: # 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)

        # 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()
```

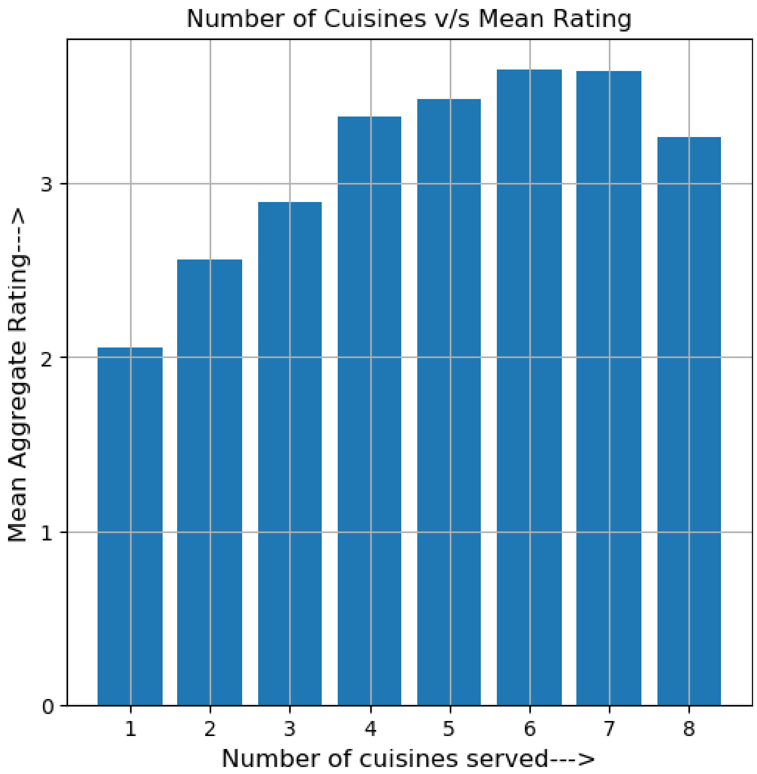
GRAPHS –

1. *Scatter Plot b/w Rating and Number of Cuisines.*



PROJECT: ZOMATO API - II

2. *Bar Graph b/w mean Rating of all the restaurants having specific number of cuisines.*



ANALYSIS 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.*

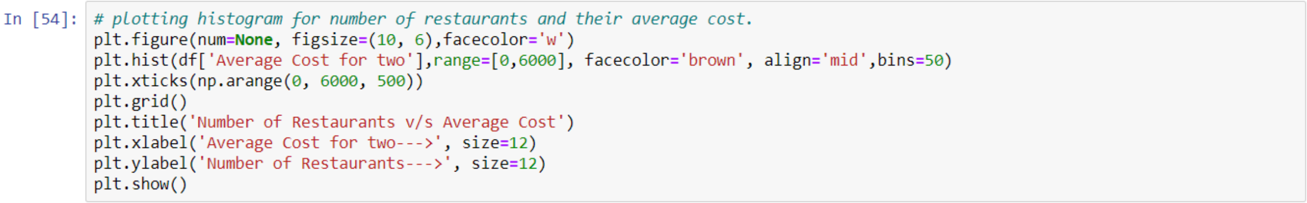
As long as the restaurants focuses on quality of food and have decent number of cuisines like between 3 – 6 and focuses on taste more that quantity etc. , Ratings will be excellent.

PROJECT: ZOMATO API - II

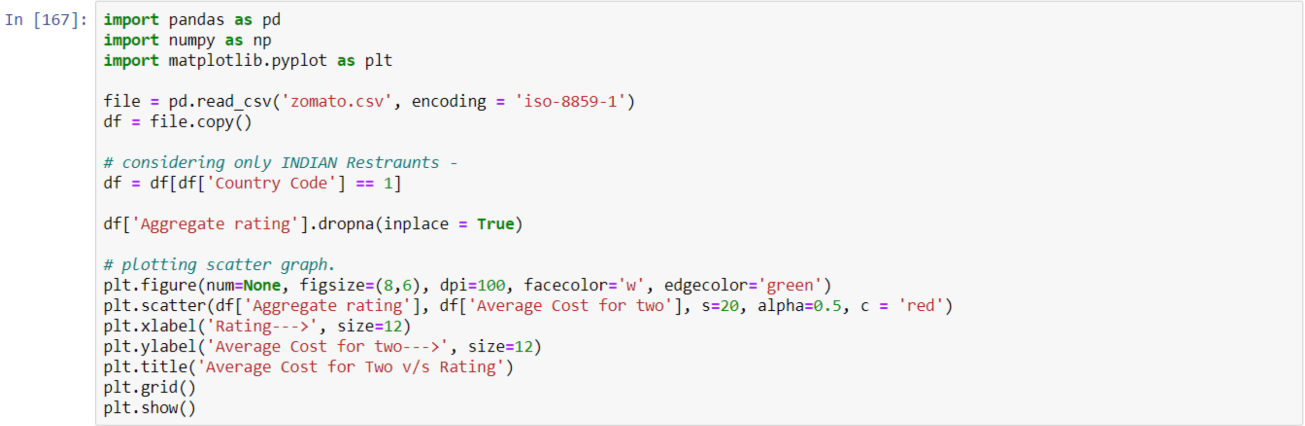
QUES. 2.1.3)

Average Cost of Restaurant

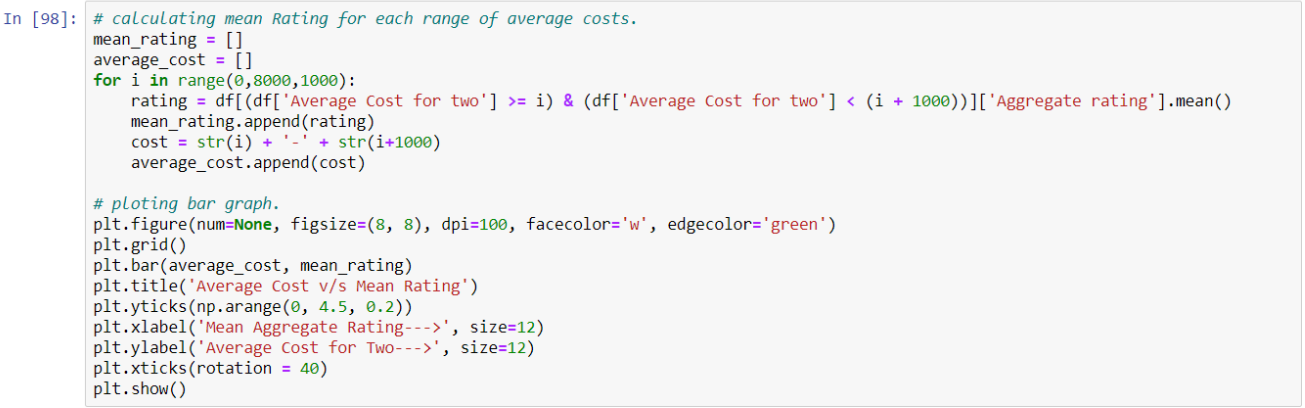
CODE SNIPPETS –



```
In [54]: # 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='brown', 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()
```



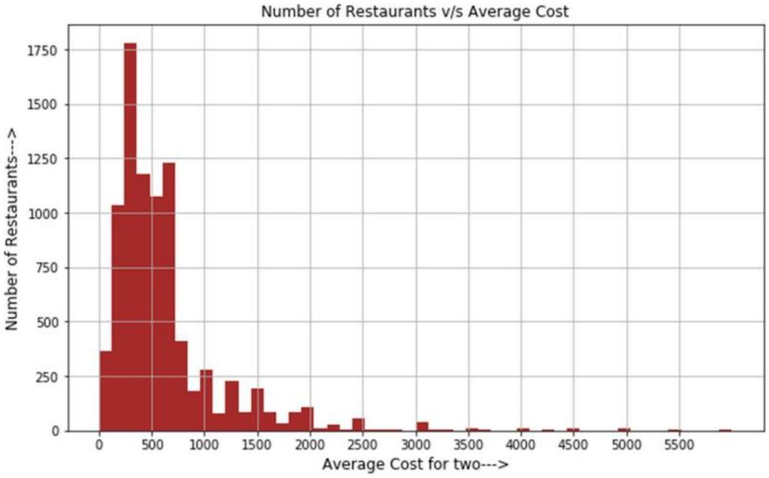
```
In [167]: import pandas as pd
          import numpy as np
          import matplotlib.pyplot as plt

          file = pd.read_csv('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)

          # 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 = 'red')
          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()
```



```
In [98]: # 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)

         # ploting 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()
```

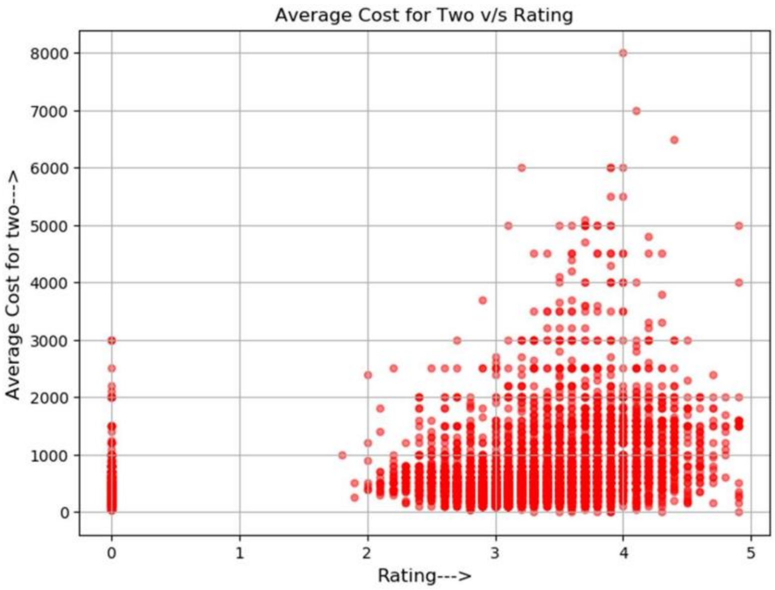
GRAPHS –

1) plotting histogram for number of restaurants to know average cost of most number of restaurants.

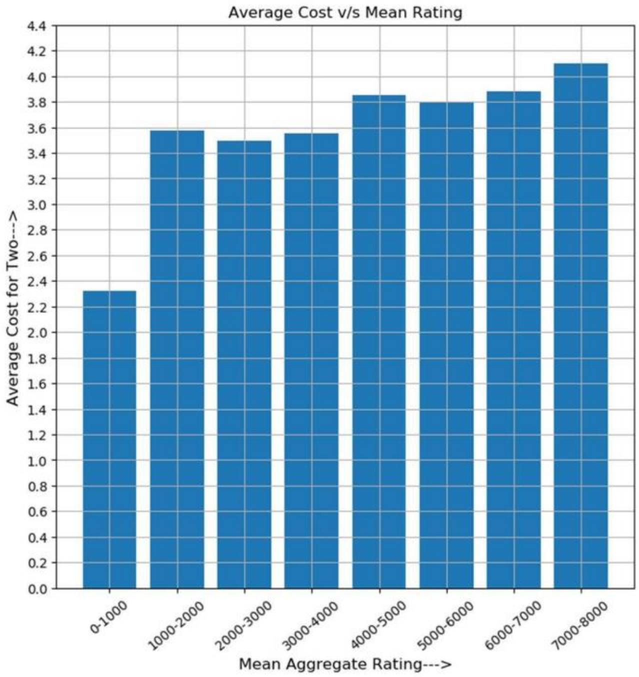


PROJECT: ZOMATO API - II

2) Scatter Plot b/w Rating and Average cost for Two.



3) Bar Graph for Rating of all restaurants having cost in the different ranges.



ANALYSIS AND JUSTIFICATION –

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

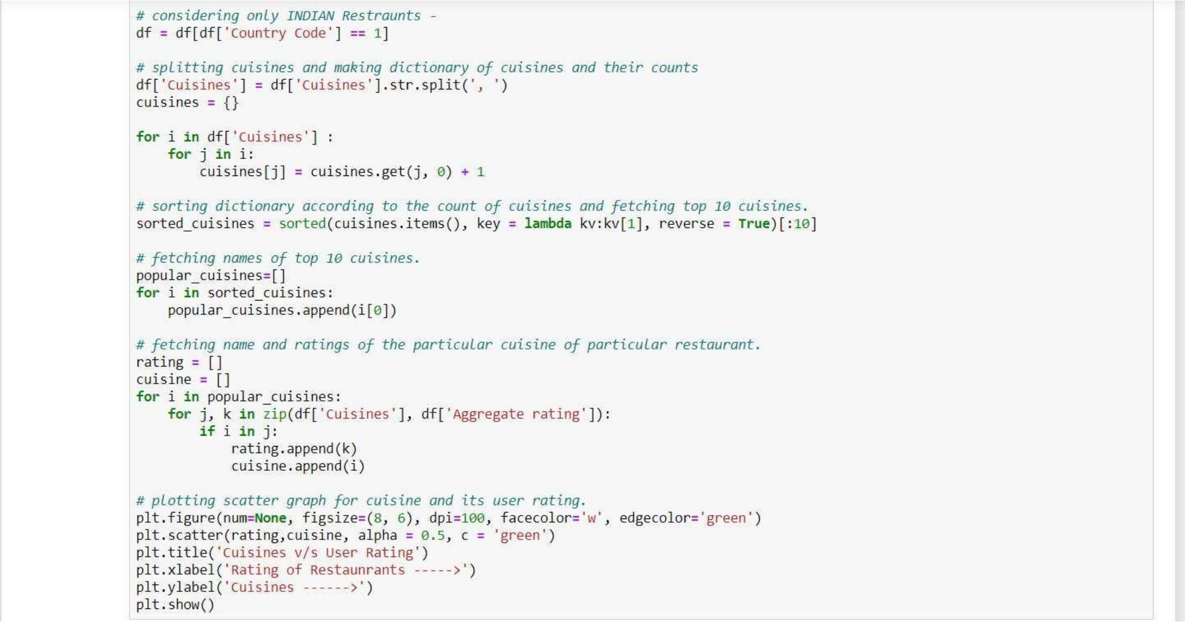
1) *This histogram shows us the spread of price for two. We can see that it peaks before 500 ,therefore we can deduce that majority of the restaurants are in the price range of 200 – 1500.*
2) *Now, from the bar graph and the scatter plot it is clear that for range of cost between 1000 – 4000 , the rating 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.

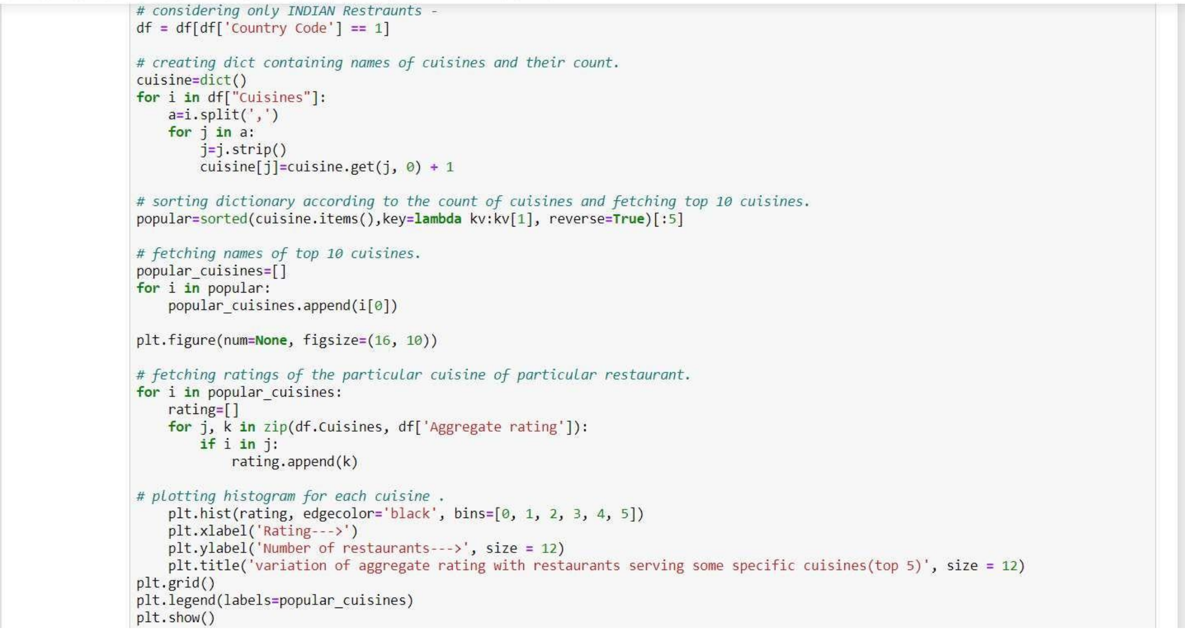
PROJECT: ZOMATO API - II

QUES. 2.1.4) *Restaurant serving some specific cuisines.*

CODE SNIPPETS –



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

# 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]

# 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)

# 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 Restaunrants ----->')
plt.ylabel('Cuisines ------>')
plt.show()
```

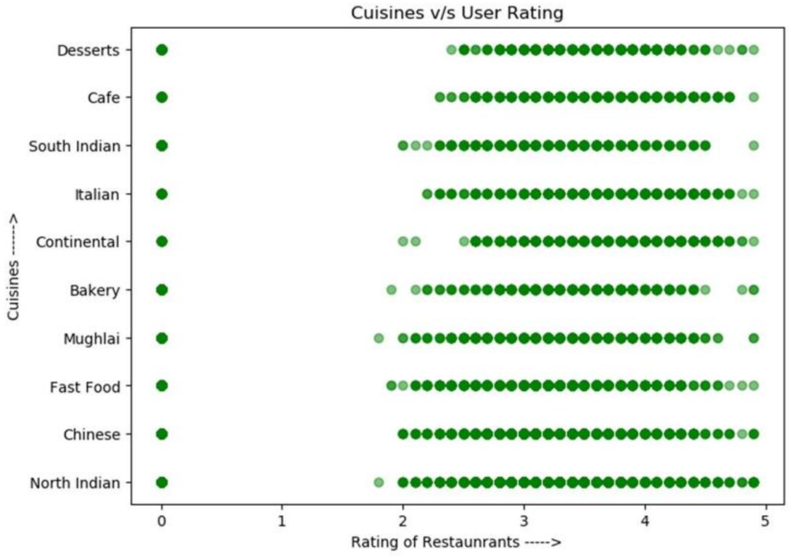


```
# considering only INDIAN Restraunts -
df = df[df['Country Code'] == 1]

# 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]

# 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)

# 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()
```

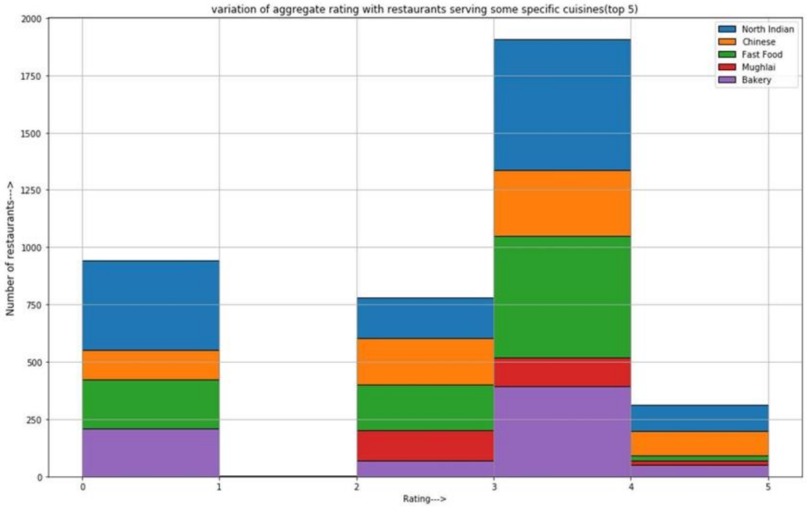
PROJECT: ZOMATO API - II

GRAPHS –

1) *Scatter Plot for top 10 cuisines and their ratings.*



2) *Histogram for number of cuisines having top cuisines and their ratings (top 5 cuisines).*



ANALYSIS 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.

PROJECT: ZOMATO API - II

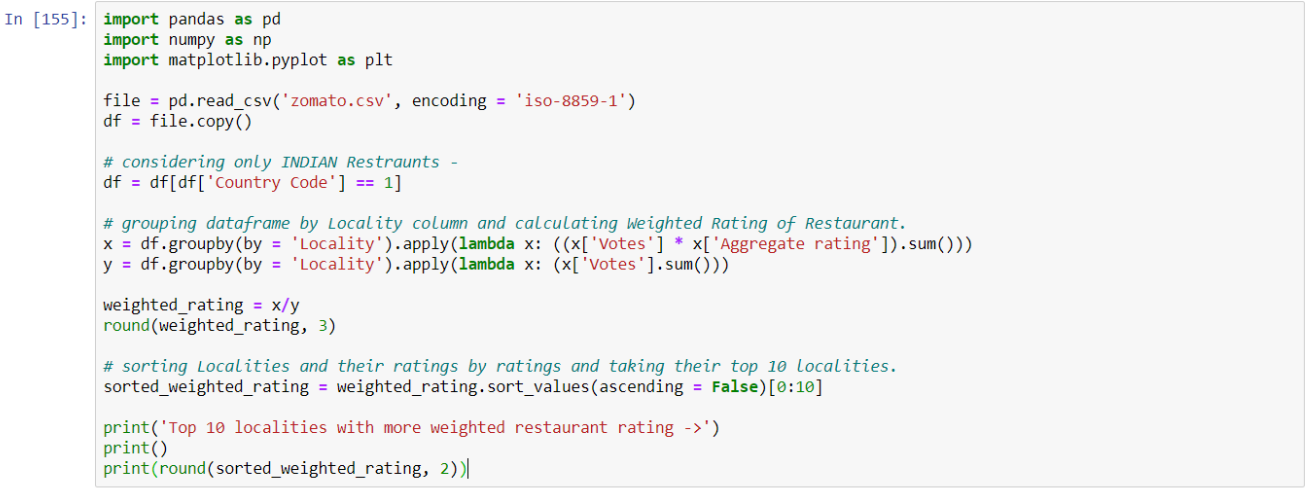
QUES. 2.2)

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

QUES. 2.2.1)

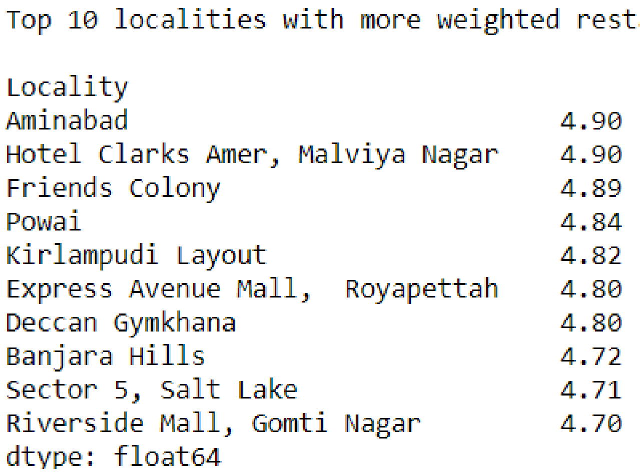
*Weighted Restaurant Rating=Σ (number of votes * rating) / Σ (number of votes) .*

CODE SNIPPETS –



```
In [155]: import pandas as pd
          import numpy as np
          import matplotlib.pyplot as plt

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

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

          # 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)

          # 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))
```

OUTPUTS –



```
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
```

CODE EXPLANATION –

1) *Importing all the necessary libraries and then reading the csv file using csv.read().*
2) *Fetching data only for India where country code = 1.*
3) *Grouping the data according to the Localities and then applying the function for calculating the Weighted Restaurant Rating*
4) *Calculating Weighted Restaurant Rating.*
5) *Sorting in descending order the localities according to the ratings.*
6) *Printing the top 10 localities along with their ratings.*

RESULT –

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

PROJECT: ZOMATO API - II

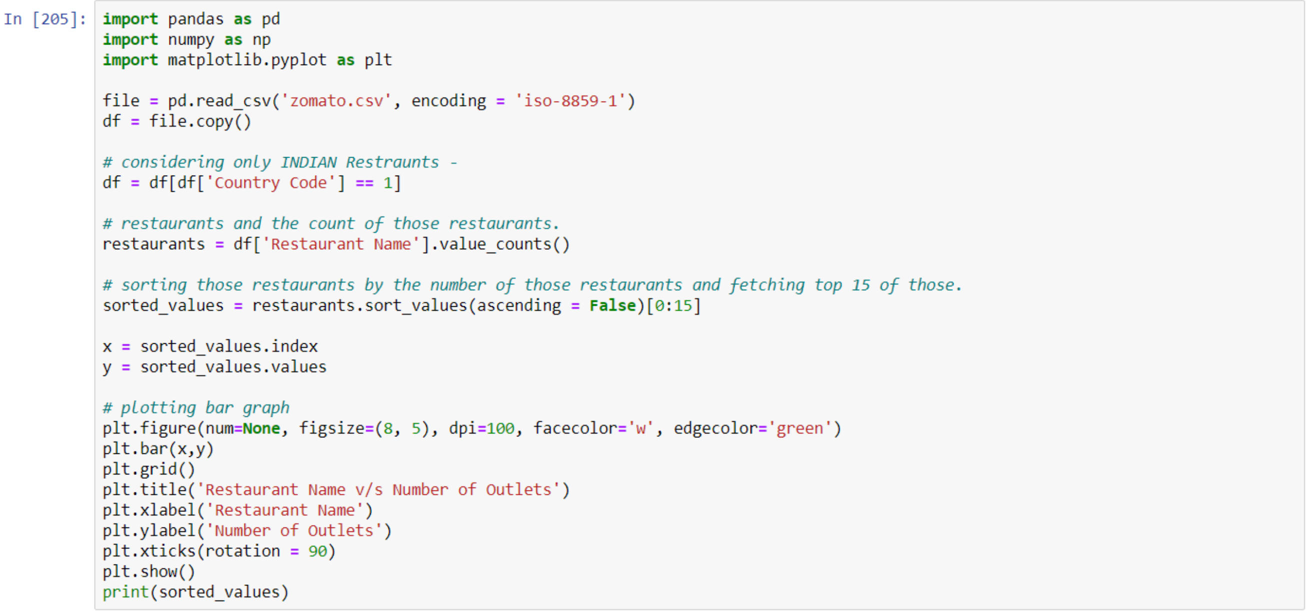
QUES. 3)

Visualization

QUES. 3.1)

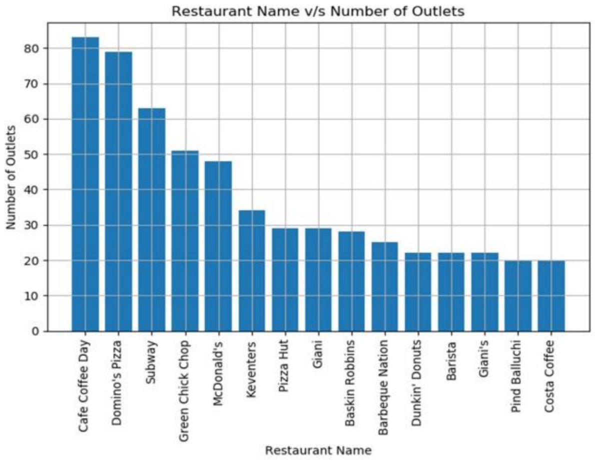
Plot the bar graph top 15 restaurants have a maximum number of outlets.

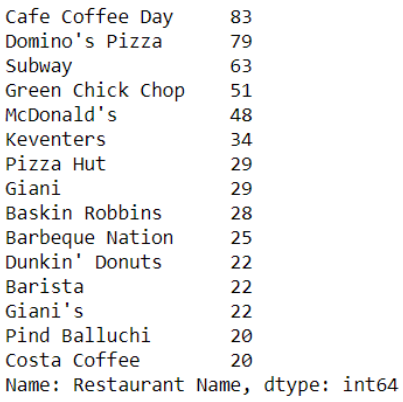
CODE SNIPPETS –



```
In [205]: import pandas as pd
          import numpy as np
          import matplotlib.pyplot as plt

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

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

          # 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

          # 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 AND 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
Dunkin' Donuts       22
Barista              22
Giani's              22
Pind Balluchi        20
Costa Coffee         20
Name: Restaurant Name, dtype: int64
```

CODE EXPLANATION –

1) *Read the file using pandas read_csv function*
2) *Fetching data only for India where country code = 1.*
3) *Counting No. of outlets for a particular restaurant using value_counts() function.*
4) *Sorting Restaurants name according to number of outlets in descending order.*
5) *Printed and plotted graph only for TOP 15 Restaurants with most number of outlets.*

RESULT–

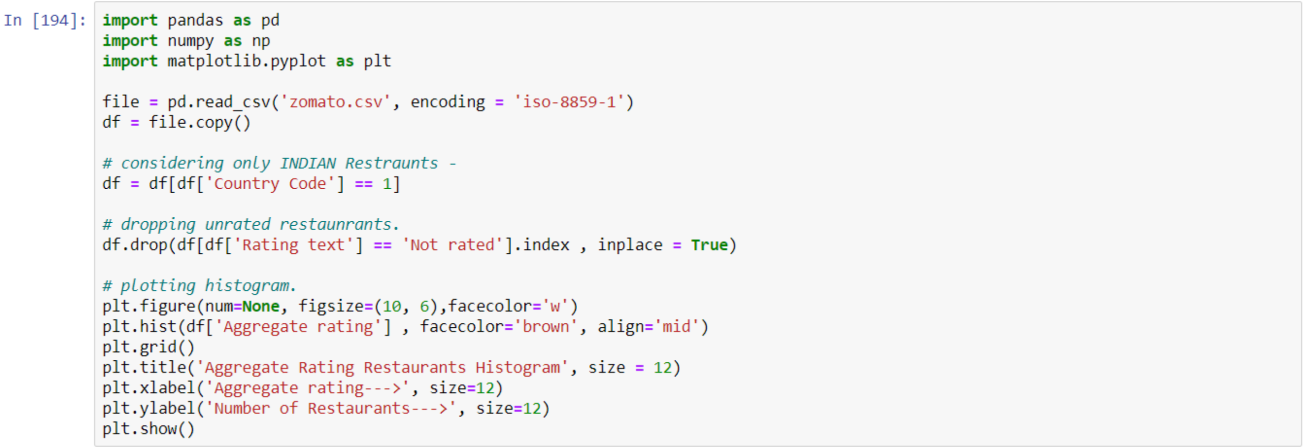
We got the desired outcome i.e. top 15 Restaurants with most number of outlets.

PROJECT: ZOMATO API - II

QUES. 3.2)

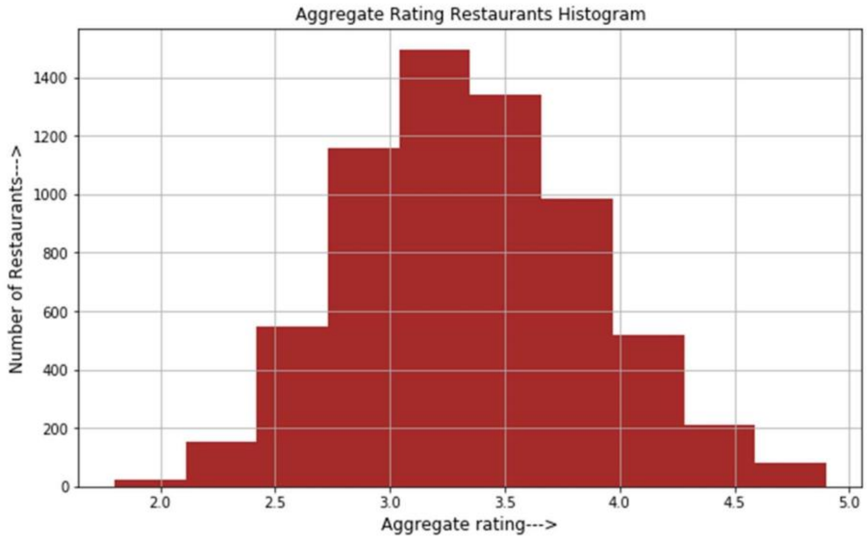
Plot the histogram of aggregate rating of restaurant (drop the unrated restaurant).

CODE SNIPPETS –



```
In [194]: import pandas as pd
          import numpy as np
          import matplotlib.pyplot as plt

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

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

          # dropping unrated restaunrants.
          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–



CODE EXPLANATION –

1) *Read the file using pandas read_csv function*
2) *Fetching data only for India where country code = 1.*
3) *Dropping the Data for unrated restaurants.*
4) *Plotting the histogram for aggregate rating of restaurant.*

RESULT AND JUSTIFICATION –

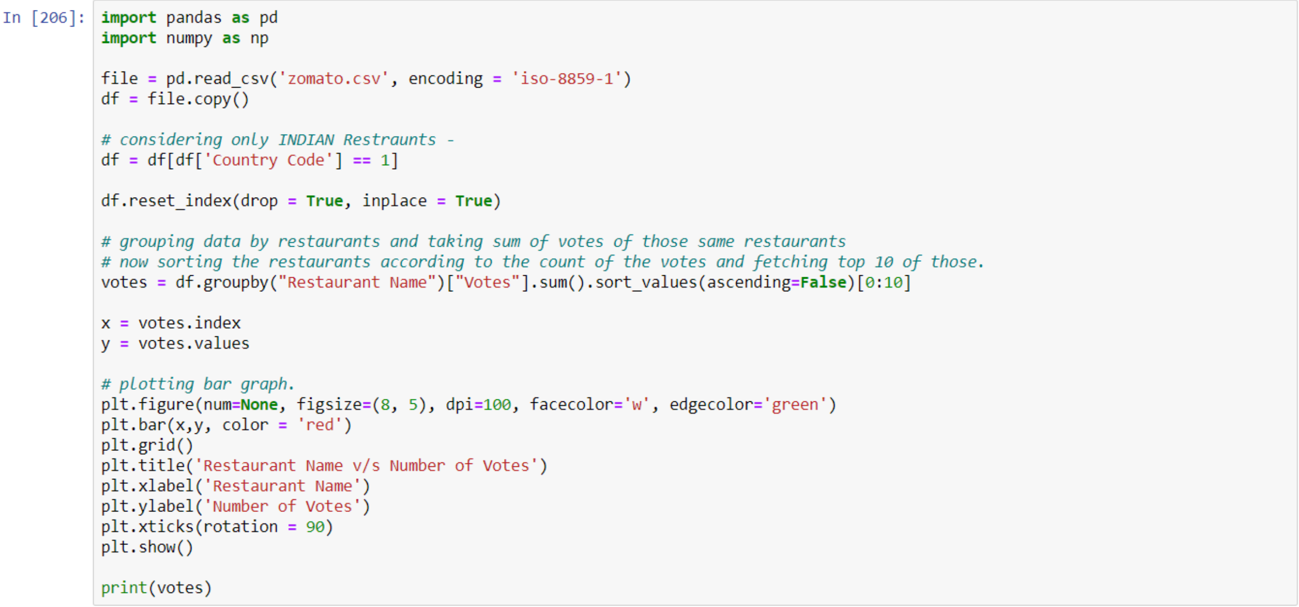
We got the desired histogram.

PROJECT: ZOMATO API - II

QUES. 3.3)

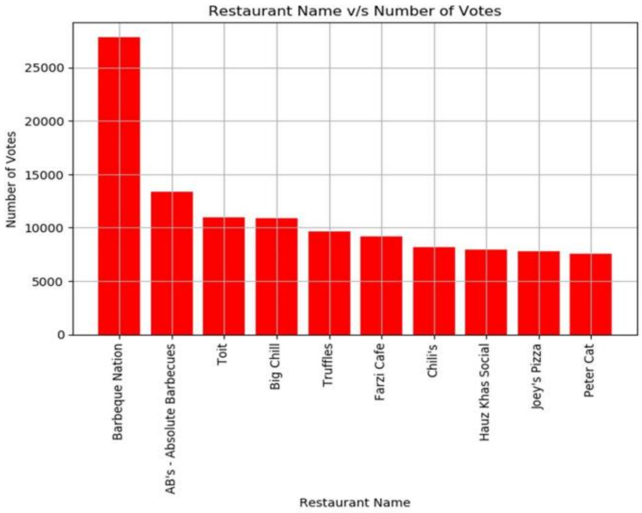
Plot the bar graph top 10 restaurants in the data with the highest number of votes.

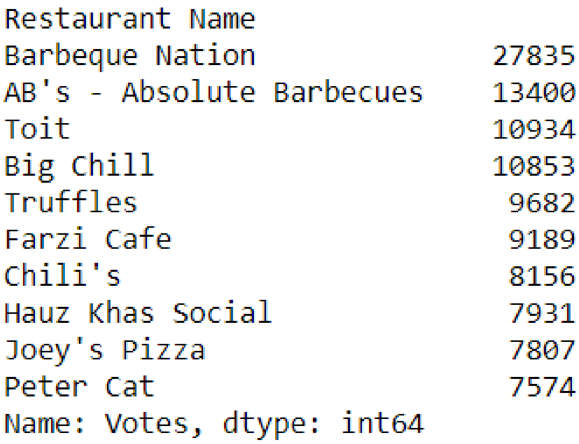
CODE SNIPPETS –



```
In [206]: import pandas as pd
          import numpy as np

          file = pd.read_csv('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)

          # 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

          # 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 AND 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
```

CODE EXPLANATION –

1) *Read the file using pandas read_csv function*
2) *Fetching data only for India where country code = 1.*
3) *Grouping data by restaurants and taking sum of votes of those same restaurants*
4) *Now, sorting the restaurants according to the count of the votes and fetching top 10 of those.*
5) *Plotting the bar graph for top 10 restaurants in the data with the highest number of votes.*

RESULT AND JUSTIFICATION –

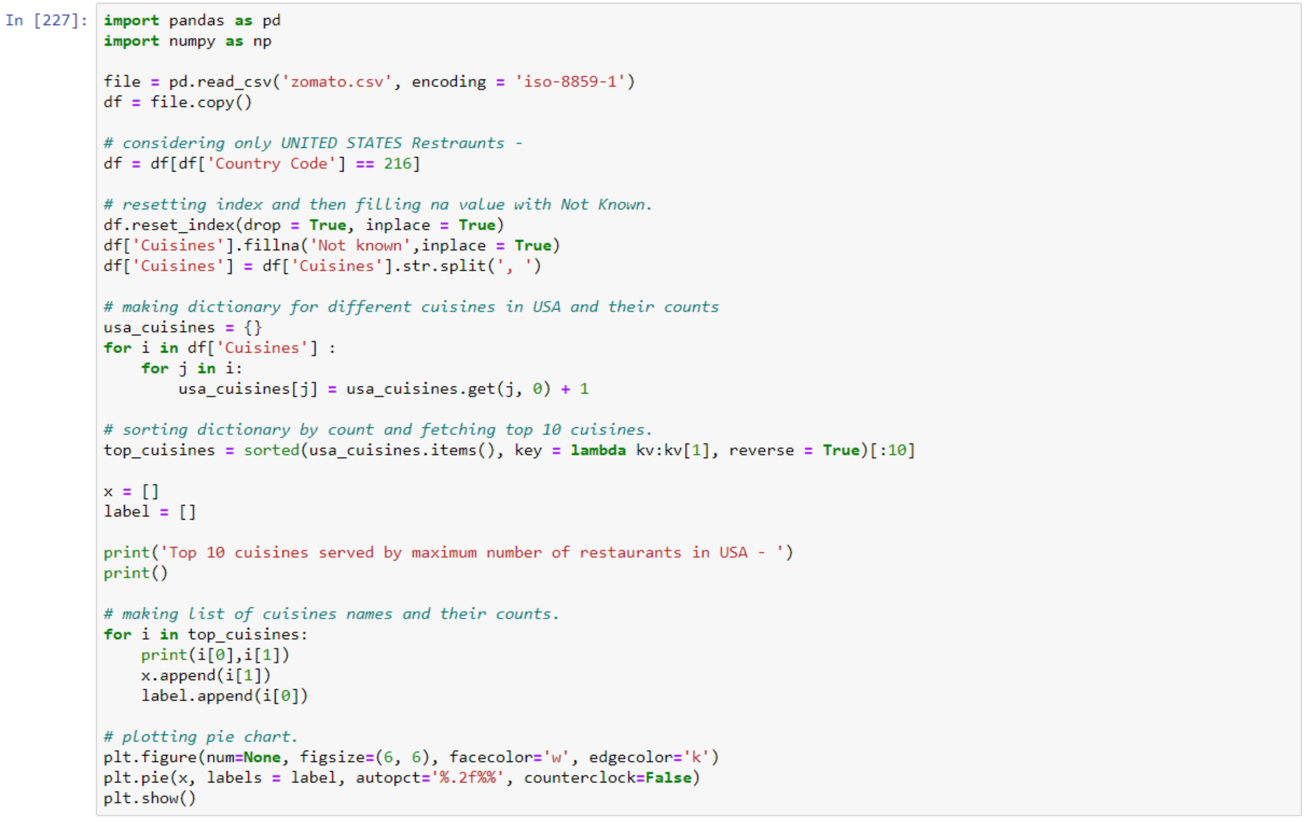
We got the desired graph and output i.e. for top 10 restaurants in the data with the highest number of votes.

PROJECT: ZOMATO API - II

QUES. 3.4)

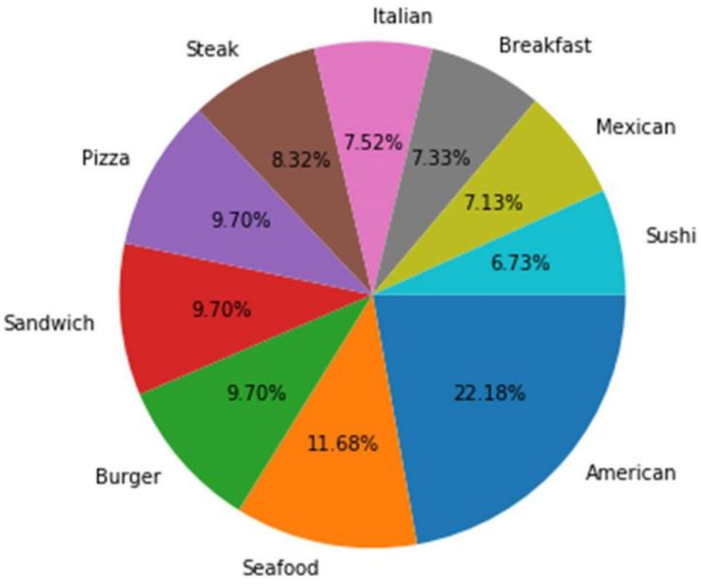
Plot the pie graph of top 10 cuisines present in restaurants in the USA.

CODE SNIPPETS –

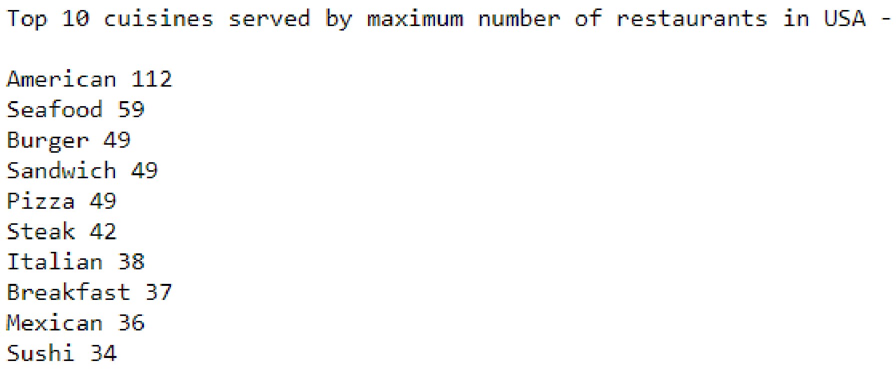


```
In [227]: import pandas as pd
          import numpy as np

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

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

          # 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

          # 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])

          # 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 AND OUTPUT–



PROJECT: ZOMATO API - II



```
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
```

CODE EXPLANATION –

1) Read the file using pandas read_csv function 2)
Fetching data only for USA where country code = 216.

3) *Resetting index and then filling NaN value with Not Known.*
4) *Since, a restaurant can serve many different cuisines which is given in a string to us.*
5) *So, to count for each cuisine , we split the cuisine column in df by using split() function by (,).*
6) *Creating a dictionary for cuisines in the USA.*
7) *Now, 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.*
9) *Plotting the pie chart for top 10 cuisines present in restaurants in the USA.*

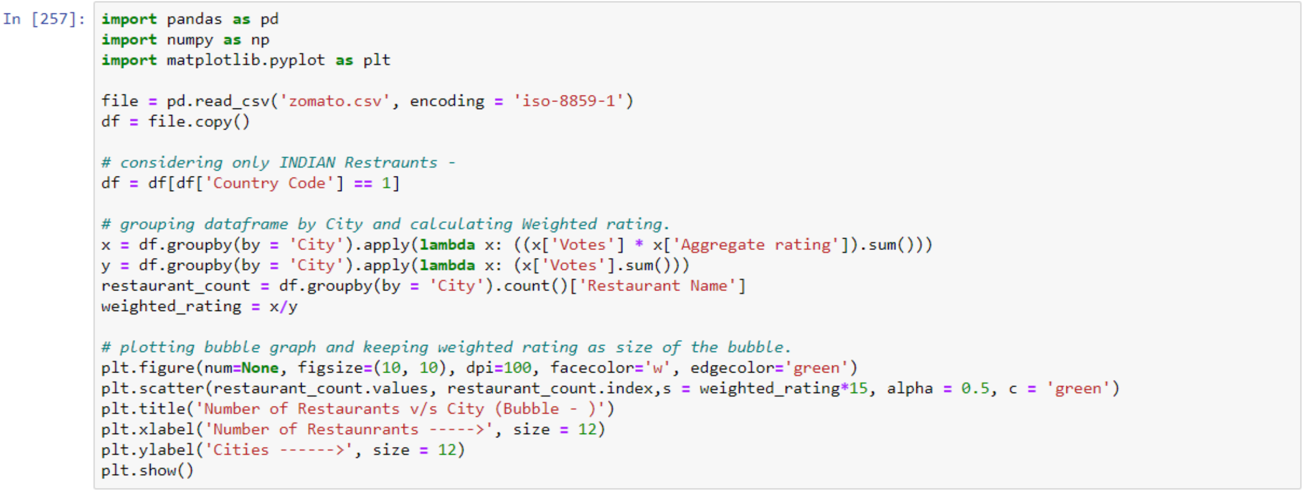
RESULT AND JUSTIFICATION –

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.

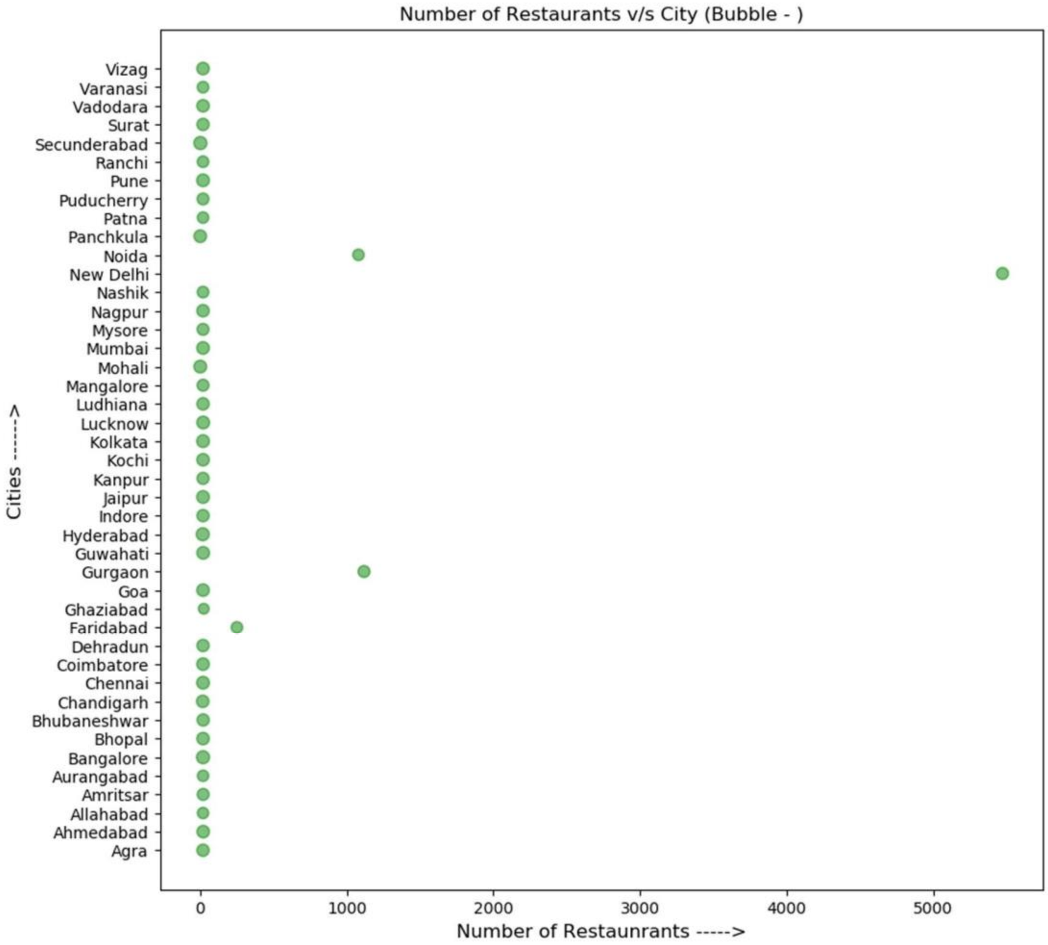
CODE SNIPPETS –



```
In [257]: import pandas as pd
          import numpy as np
          import matplotlib.pyplot as plt

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

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

          # 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

          # 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()
```

PROJECT: ZOMATO API - II

GRAPH –



CODE EXPLANATION –

1) *Read the file using pandas read_csv function*
2) *Fetching data only for India where country code = 1.*
3) *Grouping the data according to the Cities and then applying the function for calculating the Weighted Restaurant Rating*
4) *Calculating Weighted Restaurant Rating.*
5) *Grouping the data according to the Cities and then counting the number of restaurants in that city by applying count() function on restaurant name.*
6) *plotting bubble graph and keeping weighted rating as size of the bubble.*

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.