Question 1. *The dataset is highly skewed toward the cities included in Delhi-NCR. So, we will summarise all the other cities in the 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.*

Question 1.

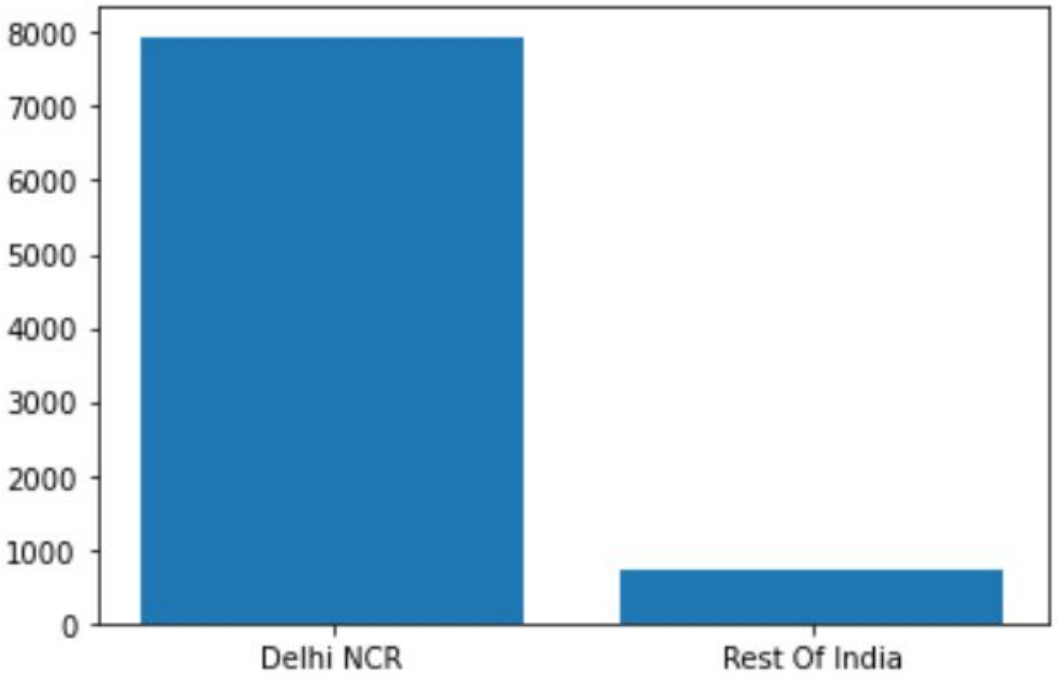
Part 1: *Plot the bar graph of the number of restaurants present in Delhi NCR vs the Rest of India.*

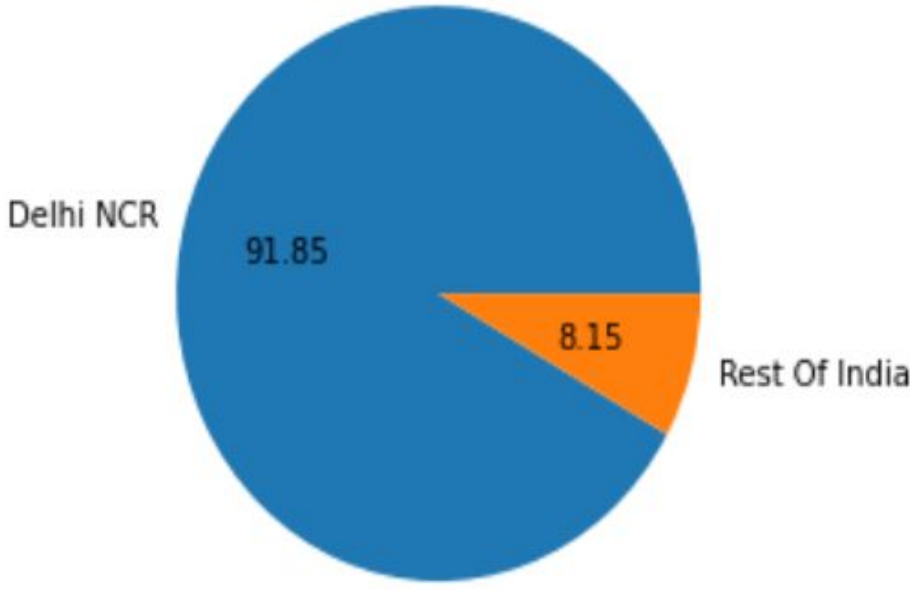
ANSWER:

The code to plot the bar graph of number of restaurants in Delhi NCR vs Rest Of India is written below:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==1]
data['ncr'] = ((data.City == "New Delhi")|(data.City == 'Ghaziabad')|(data.City == 'Noida')|(data.City == 'Gurgaon')|(data.City ==  'Faridabad'))
ncr =data.ncr.sum()
al= data.City.count()
other = al- ncr
names = ['Delhi NCR','Rest Of India']
plot = [ncr,other]
plt.bar(names,plot )
plt.show()
plt.pie(plot, labels = names, autopct =  '%.2f')
plt.show()
```

So, after importing all the required libraries (pandas and matplotlib.pyplot), I read the data given using pandas and stored it into the ‘data’ variable. Then, I extracted data only for India by providing its country code (1). After that, I created another column named ‘ncr’ in the data which stores ‘True’ for Delhi NCR cities (New Delhi, Ghaziabad, Noida, Gurgaon, Faridabad) and ‘False’ for all other cities (rest of India). Then in ‘ncr’ variable, I kept the count of NCR cities using summing all True values from the newly created ncr column. In another variable ‘al’, I stored the count of all cities and then subtracted the count of NCR cities from it to get the number of cities in the rest of India except Delhi NCR. After that, in ‘names’ variable, I stored ‘Delhi NCR’ and ‘Rest Of India’ as a list, and in plot variable, values of the count of NCR cities and other cities from variables ncr and other respectively as a list. Finally, I plotted a bar graph representing the count of Delhi NCR cities and Cities of the Rest of India.I also plotted a pie chart which will also compare the results. The graphs looks like:





So, it can be seen from the bar graph that the data is actually skewed towards Delhi NCR as Delhi NCR cities come about a little less than 8000 while the cities from the rest of India are not even 1000.

Question 1.

Part 2. *Find the cuisines which are not present in the restaurant of Delhi NCR but present in the rest of India. Check using Zomato API whether these cuisines are actually not served in restaurants of Delhi-NCR or just it due to an incomplete dataset.*

ANSWER:

The code to find the cuisines not present in the Delhi NCR restaurants but present in the rest of India is written below:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==1]
data.City.loc[(data.City == "New Delhi")|(data.City == 'Ghaziabad')|(data.City == 'Noida')|(data.City == 'Gurgaon')|(data.City ==  'Faridabad')] = 'NCR'
Cuisine = data.Cuisines.str.split(',')
al =sum(Cuisine, [])
s = set()
for i in al:
    s.add(i.strip())
loc = data.loc[(data.City == 'NCR')]
data['Cui'] = (data.loc[(data.City == 'NCR'), 'Cuisines'])
Cu = data.Cui.str.split(',')
new =[]
for i in data.index:
    if data['Cui'][i] != 'NaN'or data['Cui'][i] != nan :
        new.append((data['Cui'][i]))
n = set([str(s) for s in new])
m = set()
for j in n:
    for i in j.split(','):
        m.add(i.strip())
actual = s-m
for i in actual:
    print(i, end = ' ')
print()
```

After importing the required libraries, I read the data given using pandas read_csv function and stored it in the ‘data’ variable. Then I kept only the data of India using its country code, which is 1. The value of City column is made NCR at the locations where the Delhi NCR cities exist. In ‘Cuisine’ variable, I stored the data from Cuisines column by splitting it by commas as one restaurant could serve multiple cuisines. Next, the ‘al’ variable stores all cuisines in a single list, which is done using sum function, all the values from ‘Cuisine’ variable are taken and stored in ‘al’ as list. Then ‘s’ set is created and all cuisines are stored in it by removing extra spaces(strip) such that no cuisine is repeated. Now we have to get the Cuisines from Delhi NCR. To do this, I stored the locations of positions which has NCR cities as City value. Then, I created a new column named ‘Cui’ which stores Cuisines only for Delhi NCR cities. Then in ‘Cu’list, all the cuisines are stored by splitting from commas and finally if cuisine is not an empty value, it is appended into ‘new’ list and then added to set n. But lists of cuisines get stored in set n. So, I split them all and stored the Cuisines served in Delhi NCR cities finally in ‘m’ set. Finally, I subtracted set m from set s (Cuisines served in all over India - Cuisines served in Delhi NCR), which gave me Cuisines not served in Delhi NCR but served in the rest of the India.

The result is:

```
Cajun BBQ Malwani German
```

So, these four Cuisines- Cajun, BBQ, Malwani and German are not served in Delhi NCR but served somewhere in the rest of India.
Now, I have to verify this data with the help of Zomato API, the code for which is written below:

```
import requests as r
import json
response  = r.get('https://developers.zomato.com/api/v2.1/cuisines', headers
={'user-key':'3f1dee3542afbb39e71ce019a23dc89b'}, params={'city_id':1})
res = response.json()
cuisines =[]
for i in res['cuisines']:
    cuisines.append(i['cuisine']['cuisine_name'].split(','))
q = set()
for j in cuisines:
    for i in j:
        q.add(i.strip())
present = False
for i in q:
    if i in actual:
        present = True
print(present)
```

So, I imported the requests library as r and get the data from Zomato API by giving the URL, my user key, and stored data only for Delhi NCR(city id = 1) in 'response'. Then I converted the response data into a readable form (JSON) and stored it in 'res'. Then I created a list named 'cuisines' and stored all cuisines from all cities in it. To remove the repeated values, I created a set 'q' and stored unique cuisines from the cuisines list in it. Then finally, I printed only those cuisines which are not served in Delhi NCR according to csv file but served in Delhi NCR according to Zomato API. The result came out to be:

```
BBQ Malwani
```

From the API result, the conclusion can be drawn that restaurants present in Delhi-NCR are serving two of these cuisines(BBQ and Malwani) which were excluded by csv data. Hence, it can be said that the dataset is not complete. Otherwise, ideally, nothing should have been printed.

Question 1.

Part 3. *Find the top 10 cuisines served by the maximum number of restaurants in Delhi NCR and the rest of India.*

ANSWER:

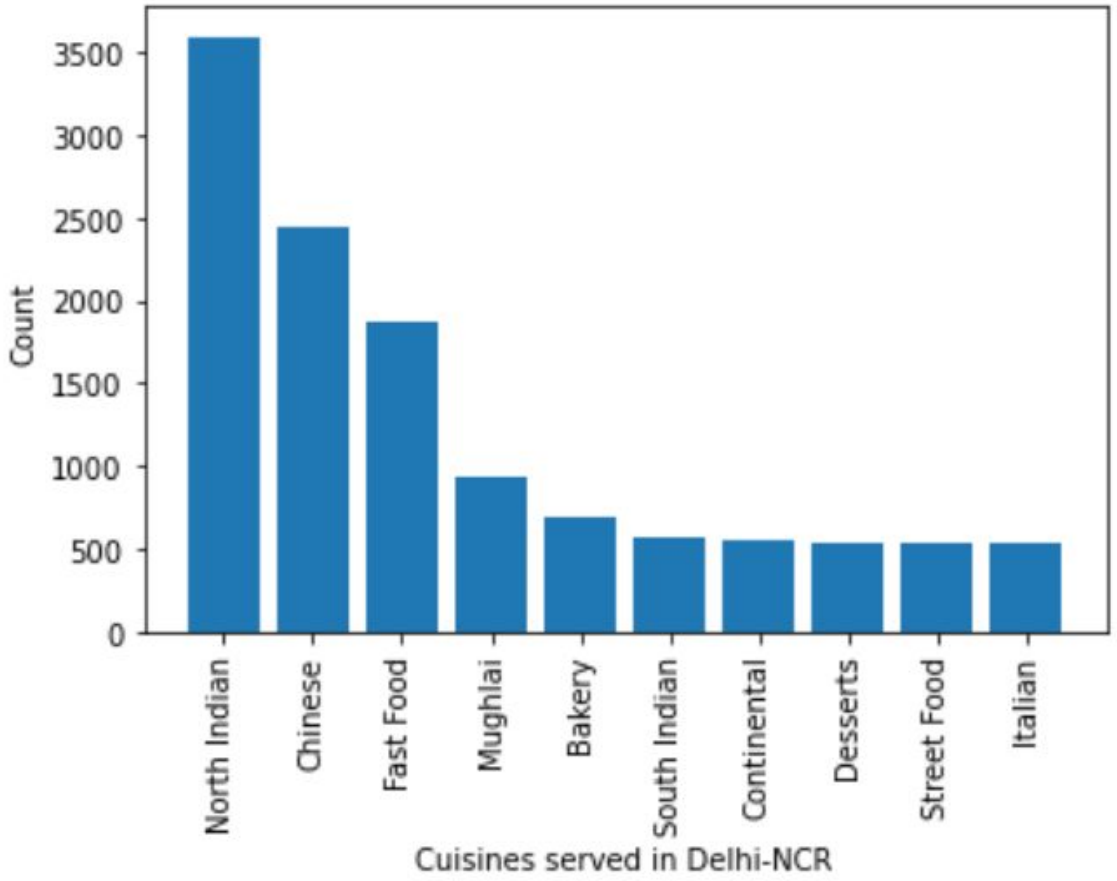
The code to find the top 10 cuisines served by maximum restaurants in Delhi NCR and the rest of India is written below:

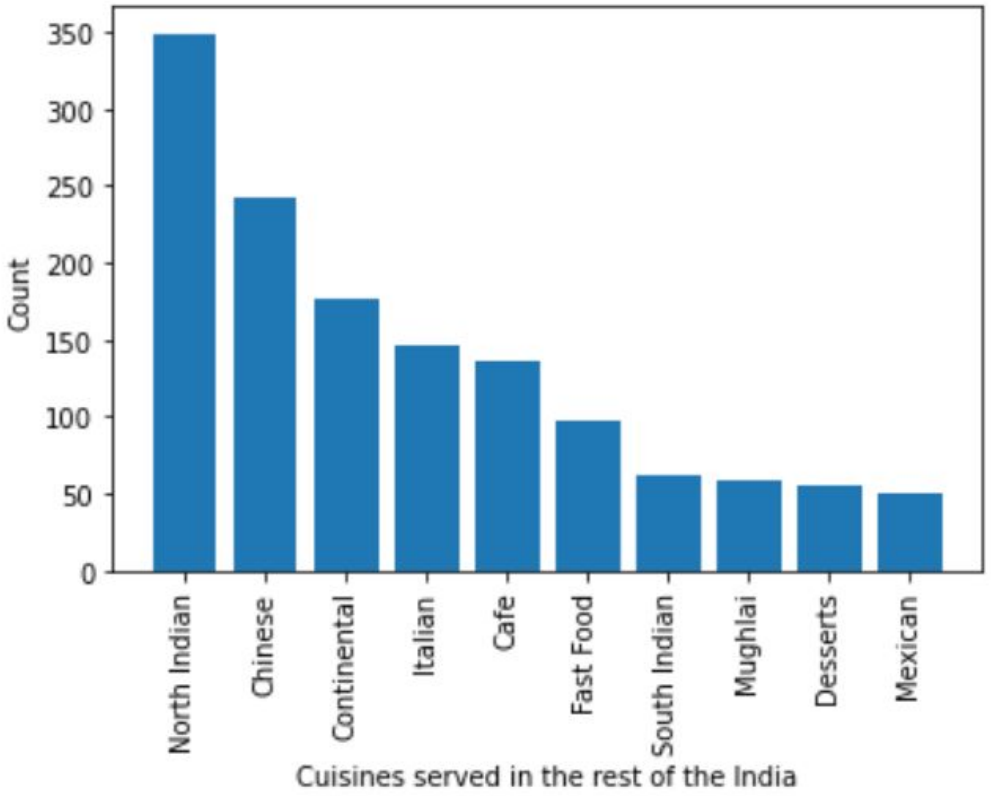
```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==1]
di = {}
```

```
    data.City.loc[(data.City == "New Delhi")|(data.City == 'Ghaziabad')|(data.City ==
'Noida')|(data.City == 'Gurgaon')|(data.City ==  'Faridabad')] = 'NCR'
    loc = data.loc[(data.City == 'NCR')]
    for i in data.index:
        if data['City'][i] == 'NCR':
            e = data['Cuisines'][i].split(',')
            for k in e:
                if k.strip() in di:
                    di[k.strip()] = di[k.strip()] + 1
                elif k.strip() not in di:
                    di[k.strip()] = 1
    sorted_di = dict(sorted(di.items(), key=lambda item:item[1], reverse=True))
    cuisine = [ i for i in sorted_di][:10]
    count = [ sorted_di[i] for i in sorted_di][:10]
    plt.bar(cuisine,count)
    plt.xticks(rotation=90)
    plt.xlabel("Cuisines served in Delhi-NCR")
    plt.ylabel("Count")
    plt.show()
    d = {}
    for i in data.index:
        if data['City'][i] != 'NCR':
            e = data['Cuisines'][i].split(',')
            for k in e:
                if k.strip() in d:
                    d[k.strip()] = d[k.strip()] + 1
                elif k.strip() not in d:
                    d[k.strip()] = 1
    sorted_d = dict(sorted(d.items(), key=lambda item:item[1], reverse=True))
    cuisines = [ i for i in sorted_d][:10]
    counts = [ sorted_d[i] for i in sorted_d][:10]
    plt.bar(cuisines,counts)
    plt.xticks(rotation=90)
    plt.xlabel("Cuisines served in the rest of the India")
    plt.ylabel("Count")
    plt.show()
```

After importing all the required libraries (pandas and matplotlib.pyplot), I read the data given using pandas and stored it into the ‘data’ variable. Then, I extracted data only for India by providing its country code (1). The value of the City column is made NCR at the locations where the Delhi NCR cities exist. After that, I created a dictionary di and stored all the Cuisine names(where City is NCR) as a key along with the number of times a single cuisine is repeated as values in the dictionary by splitting the cuisines from one row by commas and then stripping the extra spaces. After that, I sorted the dictionary in reverse order (descending order) by values and stored the sorted dictionary as ‘sorted_di’. Then, in a list named ‘cuisine’, I stored the top ten cuisines (keys of dictionary), and in the ‘count’ list, I stored the values of the top 10 cuisines (at how many places are those served). After that, I plotted the graph for them, gave the x and y labels, and also rotated the names of all bars by 90 degrees.

The same procedure is repeated for the rest of India (City is not NCR). The output looks like:





Hence,
The top 10 cuisines served by the maximum number of restaurants along with the number of restaurants in Delhi NCR are:
North Indian 3597
Chinese 2448
Fast Food 1866
Mughlai 933
Bakery 697
South Indian 569
Continental 547
Desserts 542
Street Food 538
Italian 535

The top 10 cuisines served by the maximum number of restaurants along with the number of restaurants in the Rest of India are:
North Indian 349
Chinese 242
Continental 177
Italian 147
Cafe 136
Fast Food 97
South Indian 62
Mughlai 59
Desserts 55
Mexican 50

Question 1:

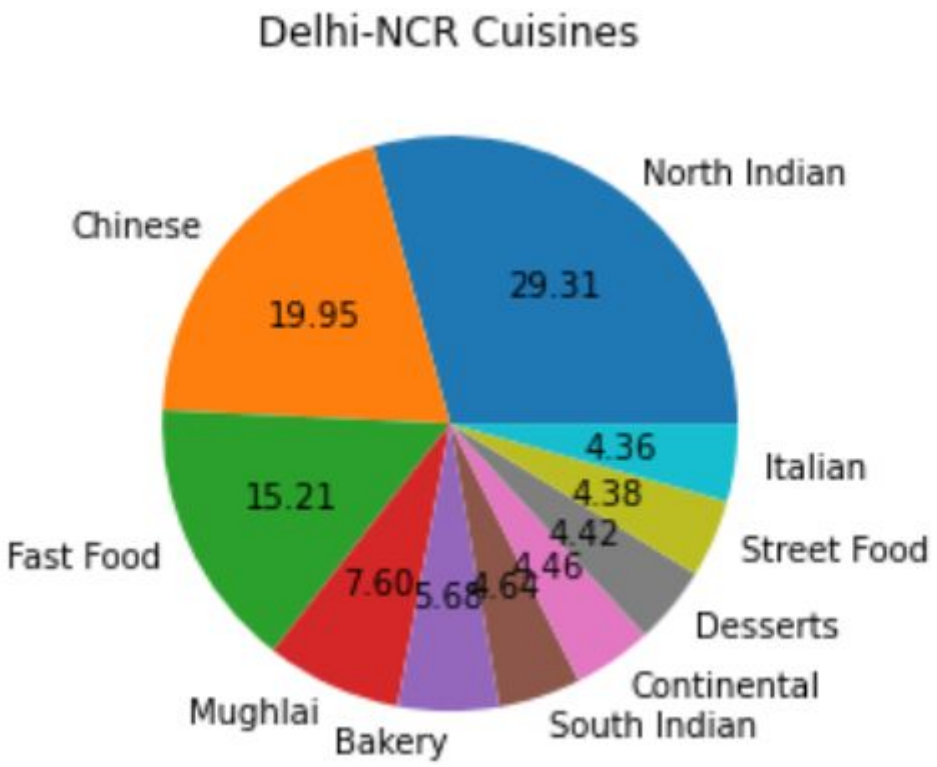
Part 4: *Write a short detailed analysis of how cuisine served is different from Delhi NCR to the Rest of India. Plot a suitable graph to explain your inference.*

ANSWER:
The code for finding different cuisines served in Delhi NCR and the Rest of India is:

```
import requests as r
import json
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==1]
data.City.loc[(data.City == "New Delhi")|(data.City == 'Ghaziabad')|(data.City == 'Noida')|(data.City == 'Gurgaon')|(data.City ==  'Faridabad')] = 'NCR'
d = {}
d1 =  {}
for i in data.index:
    if data['City'][i] == 'NCR':
        j = data['Cuisines'][i].split(',')
        for k in j:
            if k.strip() in d:
                d[k.strip()] = d[k.strip()] + 1
```

```
            elif k.strip() not in d:
                d[k.strip()] = 1
    else:
        j = data['Cuisines'][i].split(',')
        for k in j:
            if k.strip() in d1:
                d1[k.strip()] = d1[k.strip()] + 1
            elif k.strip() not in d1:
                d1[k.strip()] = 1
ncr = dict(sorted(d.items(),  key = lambda item:item[1], reverse = True))
restIndia = dict(sorted(d1.items(),  key = lambda item:item[1], reverse = True))
NCRcuisines = [ i for i in ncr][:10]
NCRcounts = [ ncr[i] for i in ncr][:10]
RestIndiacuisines = [ i for i in restIndia][:10]
RestIndiacounts = [ restIndia[i] for i in restIndia][:10]
plt.pie(NCRcounts, labels = NCRcuisines, autopct = '%.2f')
plt.title('Delhi-NCR Cuisines')
plt.show()
plt.pie(RestIndiacounts, labels = RestIndiacuisines,autopct = '%.2f')
plt.title('Rest Of India Cuisines')
plt.show()
```

This is basically the same as the previous question. Required libraries (pandas and matplotlib.pyplot) are imported, the data given is read using pandas and stored into the ‘data’ variable. Then, data only for India is extracted by providing the country code (1). The value of the City column is made NCR at the locations where the Delhi NCR cities exist. After that, two dictionaries are created and all the Cuisine names where City is NCR as a key along with the number of times a single cuisine is repeated as values are stored in the dictionary by splitting the cuisines from one row by commas and then stripping the extra spaces. In the second dictionary, Cuisines with respect to the number of restaurants where they are stored from the rest of India except NCR are stored. After that, both the dictionaries are sorted in reverse order (descending order) by values. Then the top ten cuisines (keys of dictionary), and the values of the top 10 cuisines (at how many places are those served) are stored in separate lists for Delhi NCR and the rest of India. After that, two pie charts are plotted- One for the top 10 cuisines of Delhi NCR and the second for the top 10 cuisines of the rest of India.



Question 2. *User Rating of a restaurant plays a crucial role in selecting a restaurant or ordering the food from the restaurant.*

Question 2.

Part 1. *Write a short detailed analysis of how the rating is affected by the restaurant due following features: Plot a suitable graph to explain your inference.*

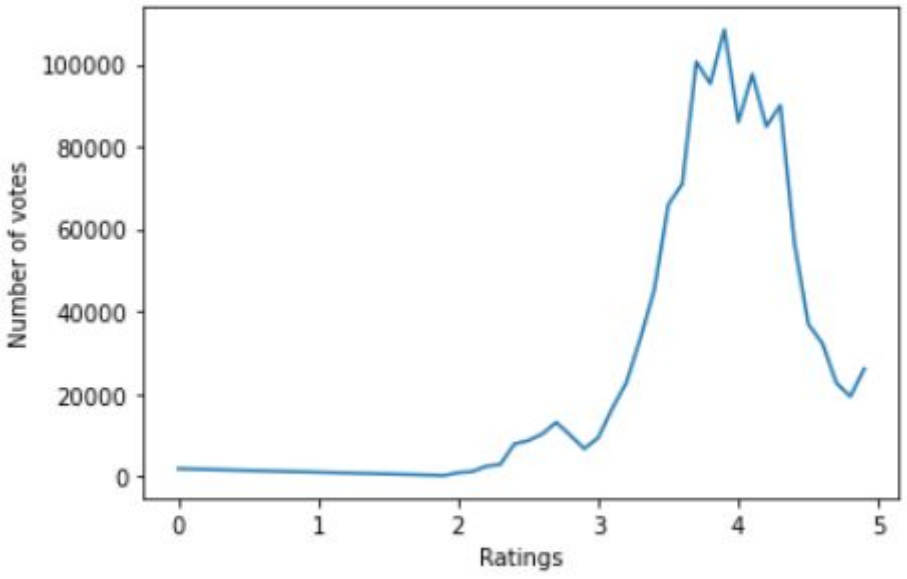
Subpart 1. *Number of Votes given Restaurant*

ANSWER:

The code to find how rating is affected by number of votes for restaurants is written below:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==1]
d = {}
for i in data.index:
    k = data['Aggregate rating'][i]
    d[k] = d.get(k,0) + data['Votes'][i]
d =  dict(sorted(d.items(), key= lambda item:item[0], reverse = True))
rating = [i for i in d]
votes = [d[i] for i in d]
plt.plot(rating,votes)
plt.xlabel('Ratings')
plt.ylabel('Number of votes')
plt.show()
```

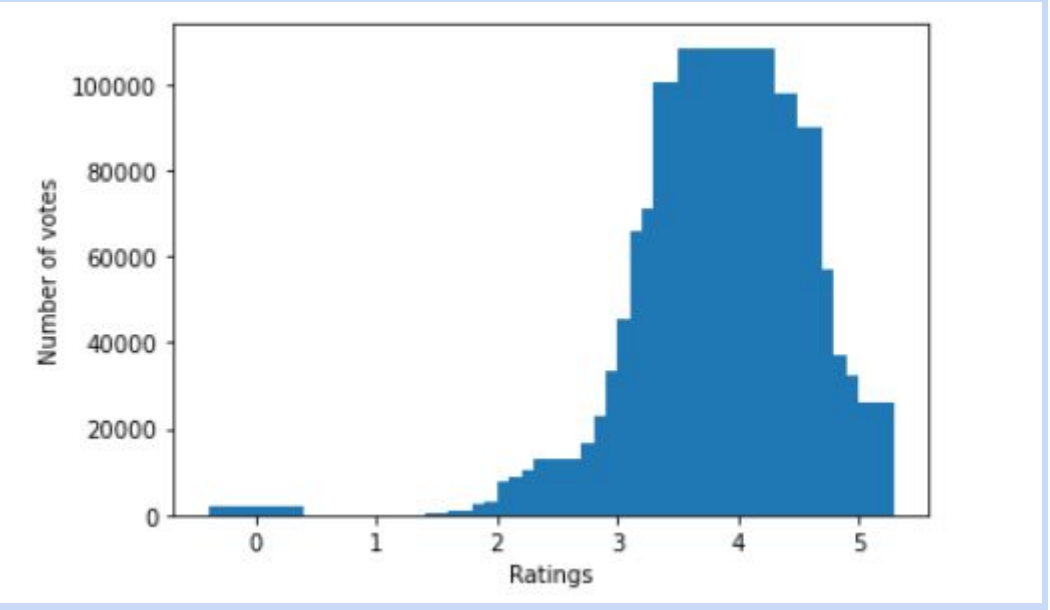
After important the required libraries and reading the data using pandas into 'data' variable, I discarded all the data that did not belong to India (by using the country code 1). Then I created a dictionary d and stored ratings as keys and number of votes as values. Then after sorting the dictionary according to the ratings in reverse (descending order), I saved the ratings(keys) in a list named 'rating' and number of votes(values) in a list named 'votes'. Finally, I plot a line graph between ratings and votes. After providing x and y labels, I showed the graph, which looks like this:



From this line graph, it can be noticed that number of votes near 500 did not effectively make the rating high. However when the number of votes exceeded 100000, the rating was near 4, which is really good. But there were again less votes for rating 5. It can be concluded that most number of people gave rating between 3 and 4.5.

I have created a bar graph for the same as well:

```
plt.bar(rating,votes)
plt.xlabel('Ratings')
plt.ylabel('Number of votes')
plt.show()
```



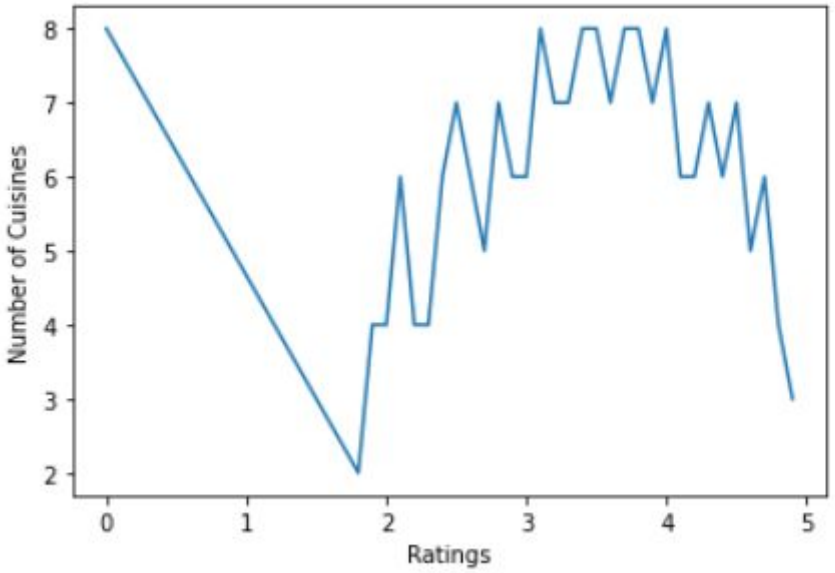
So, Ratings for very good restaurants having very large number of customers are generally excellent and they follow proportionality with number of votes But for Restaurants for votes <80000 there, is no such deduction as they have good as well as bad reviews too so, ratings get neutralized between 3.5 and 4.5.

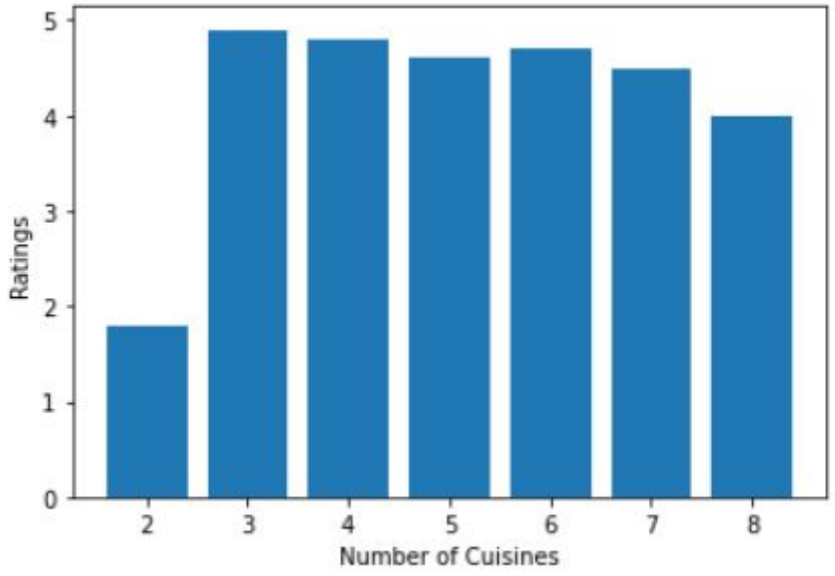
Question 2.
Part 1.
Subpart 2. *Restaurant serving more number of cuisines.*
ANSWER:
The code is :

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==1]
d = {}
for i in data.index:
    k = data['Aggregate rating'][i]
    m = data['Cuisines'][i].split(', ')
    q =len(m)
    d[k] = max(d.get(k,0), q)
d =  dict(sorted(d.items(), key= lambda item:item[0], reverse = True))
rating = [i for i in d]
cuisines = [d[i] for i in d]
plt.plot(rating,cuisines)
plt.xlabel('Ratings')
plt.ylabel('Number of Cuisines')
plt.show()
plt.bar(cuisines,rating)
```

```
plt.ylabel('Ratings')
plt.xlabel('Number of Cuisines')
plt.show()
```

So, I imported required libraries and read data using pandas' read_csv function into 'data' variable. Then I saved data only for India using country code = 1. Then I created a dictionary and saved rating as key and maximum number of cuisines (by counting by separating from commas) of that particular rating as values. After that I sorted the dictionary into descending order of ratings. After that, I stored numbers of cuisines in a list named 'Cuisines' and ratings in the list 'ratings'. Finally I plotted line graph as well as bar graph of number of cuisines versus ratings for visualization.





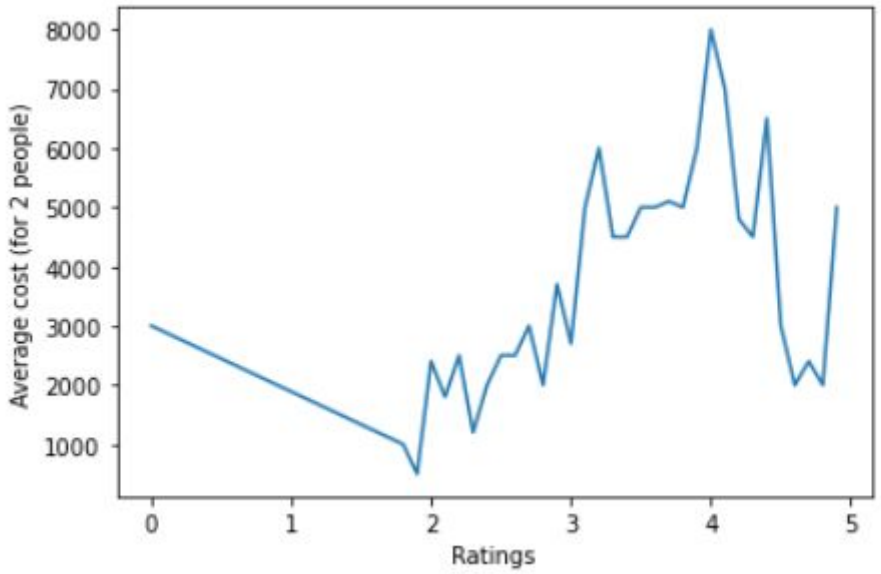
It can be concluded that when the number of cuisines provided increases from 3 to 8, the rating seems to converge between 3 and 4.5. So, the restaurants providing more number of cuisines are not much likely to get higher ratings, especially when the number of cuisines provided exceeds 6 as there is a drop seen in the graph after 6. Maybe it is because 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. Athough there is no such connection between Cuisines on offer and their ratings, from the bar graph we can see that good

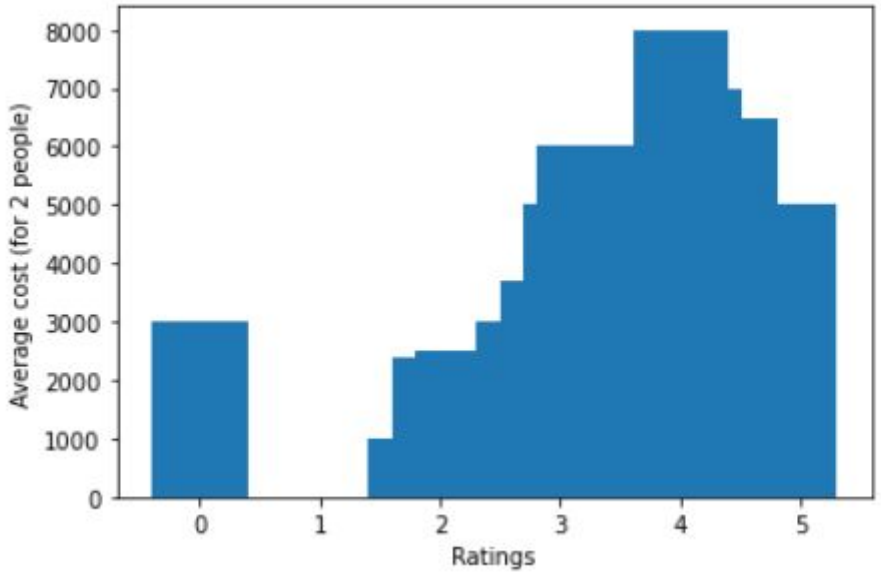
ratings are given to restaurants having 3-7 cuisines on offer. As long as the restaurants focuses on quality of food and have decent number of cuisines as between 3-6 and focuses on taste more that quantity etc., ratings will be excellent.

Question 2.
Part 1.
Subpart 3. *Average Cost of Restaurant*
ANSWER:
The code is:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==1]
d = {}
for i in data.index:
    k = data['Aggregate rating'][i]
    m = data['Average Cost for two'][i]
    d[k] = max(d.get(k,0), m)
d =  dict(sorted(d.items(), key= lambda item:item[0], reverse = True))
rating = [i for i in d]
price = [d[i] for i in d]
plt.plot(rating,price)
plt.xlabel('Ratings')
plt.ylabel('Average cost (for 2 people)')
plt.show()
plt.bar(rating,price)
plt.xlabel('Ratings')
plt.ylabel('Average cost (for 2 people)')
plt.show()
```

So, I imported required libraries and read data using pandas' read_csv function into 'data' variable. Then I saved data only for India using country code = 1. Then I created a dictionary and saved rating as key and maximum amount of Average cost for two people in that particular rating as values. After that I sorted the dictionary into descending order of ratings. After that, I stored amount in a list named 'price' and ratings in the list 'ratings'. Finally I plotted line graph as well as bar graph of number of cuisines versus ratings for visualization.





Now, from the bar graph and the line graph, it is clear that for range of cost between 1000 – 4000, the rating hovers b/w 3.5 to 3.6 which is simply the average. Till 6000 cost, the rating is under 4.0 for maximum restaurants. It is also seen that after that, for expensive restaurants having cost for 2 more than 6000,the ratings are generally excellent, between 4.0 and 5.0. So, it can be concluded that very expensive restaurants have excellent ratings. Basically expensive restaurants are worth it.

Question 2.

Part 1.

Subpart 4. *Restaurant serving some specific cuisines.*

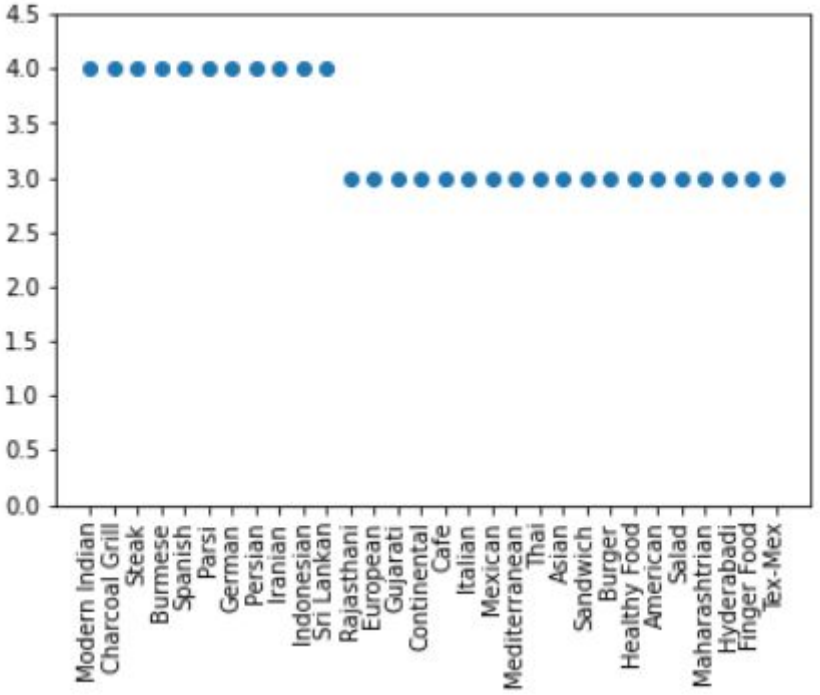
ANSWER:

The code is:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==1]
```

```
d = {}
for i in data.index:
    k = data['Cuisines'][i].split(', ')
    s = data['Aggregate rating'][i]
    for j in range(len(k)):
        m = k[j]
        if m not in d:
            d[m] = [s, 1]
        else:
            d[m][0] = d[m][0]+s
            d[m][1] = d[m][1]+1
for i in d:
    d[i] = [d[i][0]//d[i][1]]
d = dict(sorted(d.items(), key = lambda item:item[1], reverse = True))
rating =[d[i] for i in d][:30]
cuisines = [i for i in d][:30]
plt.scatter(cuisines,rating)
plt.xticks(rotation=90)
axes = plt.gca()
axes.set_ylim([0,4.5])
plt.show()
```

Here, again I imported the required libraries first and read the data into ‘data’ variable using read_csv function of the pandas. Then I kept only the data for India using Country Code 1 and discarded the rest of it. Then I created a dictionary ‘d’, stored names of cuisines as keys, ratings and number of restaurants where that cuisine is served as values. Then I found the average ratings and stored them in place of total ratings in dictionary. Finally I sorted the dictionary according to ratings and stored the average ratings for top 30 cuisines and names of top 30 cuisines in lists named ‘rating’ and ‘cuisines’ respectively. Finally, I plotted a scatter graph which looks like:



It can be observed that Cuisines like continental, Mughlai, Fast Food, South Indian Cafe etc have ratings mostly under 4.5 which means they mostly do not have excellent Ratings while 4-5 ratings are mostly given to Modern Indian', 'Charcoal Grill', 'Steak', 'Burmese', etc. 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.

Question 2.

Part 2. *Find the weighted restaurant rating of each locality and find out the top 10 localities with a more weighted restaurant rating?*

Subpart 1. *Weighted Restaurant Rating=Σ (number of votes * rating) / Σ (number of votes)*

ANSWER:

The code to find the top 10 localities with a more weighted restaurant rating is:

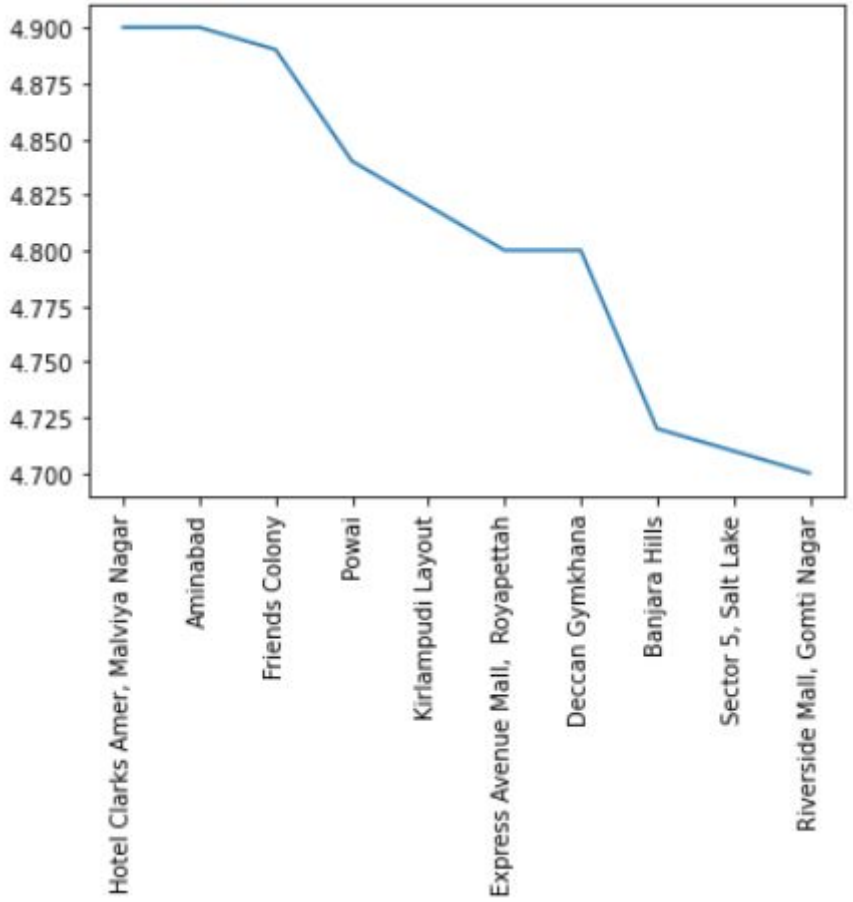
```
import pandas as pd
from math import isnan
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==1]
data['W1'] = data['Aggregate rating'] * data['Votes']
d = {}
for i in data.index:
    k = data['Locality'][i]
    if k not in d:
        d[k] = [data['W1'][i], data['Votes'][i]]
    else:
        d[k][0] = d[k][0] + data['W1'][i]
        d[k][1] =  d[k][1] + data['Votes'][i]
for i in d:
        d[i][0] = round(d[i][0]/d[i][1],2)
d = {k: d[k][0] for k in d if not isnan(d[k][0])}
d = dict(sorted(d.items(),  key = lambda item:item[1], reverse = True))
names = [i for i in d][:10]
rating = [d[i] for i in d][:10]
for i in range(10):
    print(names[i], rating[i])
```

Here, again I imported the required libraries first and read the data into ‘data’ variable using read_csv function of the pandas. Then I kept only the data for India using Country Code 1 and discarded the rest of it. Since Weighted Restaurant Rating=Σ (number of votes * rating) / Σ (number of votes) , I added the W1 column and stored the products of number of votes and rating in it. Then I created a dictionary ‘d’, added names of localities as Keys and W1 column(products of number of votes and rating), and Votes as Values of the dictionary. Then to find out the weighted rating, I divided the sum of all W1 values with the total of Votes and stored it as value in the same dictionary. After that I dropped null values of weighted rating, if any using isnan function of math library. After that I sorted the dictionary according to reverse (descending) order of the Weighted rating values. Then I extracted top 10 keys (names of localities) into ‘names’ list and top 10 weighted rating values into ‘rating’ list. Finally I printed the top 10 localities having the highest values of weighted restaurant ratings. The top 10 localities are:

```
Hotel Clarks Amer, Malviya Nagar 4.9
Aminabad 4.9
Friends Colony 4.89
Powai 4.84
Kirlampudi Layout 4.82
Express Avenue Mall,  Royapettah 4.8
Deccan Gymkhana 4.8
Banjara Hills 4.72
Sector 5, Salt Lake 4.71
Riverside Mall, Gomti Nagar 4.7
```

I also plotted a line graph to visualize this data:

```
plt.plot(names,rating)
plt.xticks(rotation=90)
plt.show()
```



Question 3. *Visualization*

Question 3.

Part 1.

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

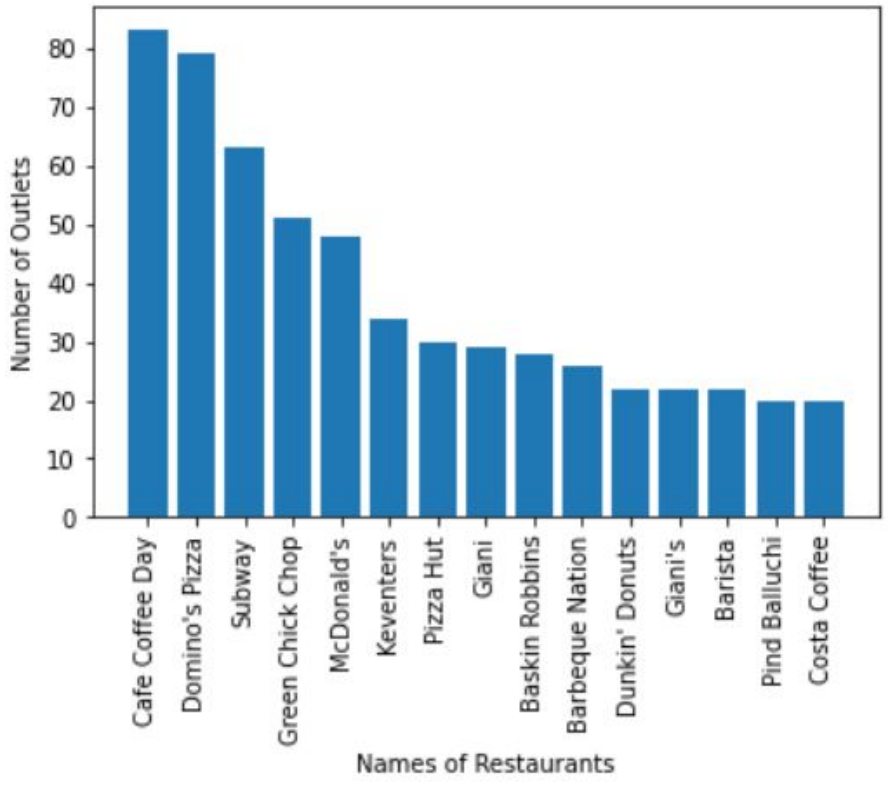
ANSWER:

The code to find the top 15 restaurants having maximum outlets is:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
```

```
d = {}
for i in data.index:
    k = data['Restaurant Name'][i]
    if k not in d:
        d[k]= 1
    else:
        d[k] += 1
di = dict(sorted( d.items(),  key = lambda item:item[1], reverse = True))
names = [i for i in di][:15]
outlets = [di[i] for i in di][:15]
plt.bar(names, outlets)
plt.xticks(rotation=90)
plt.xlabel("Names of Restaurants")
plt.ylabel("Number of Outlets")
plt.show()
```

In this, first I have imported all the required libraries and then read the given csv file using read_csv of pandas and the data is stored in 'data' variable. Then simpy a dictionary 'd' is created and Names of restaurants are stored as keys, and number of outlets for all the restaurant as values in dictionary. Then dictionary is sorted in descending order according to the number of outlets and stored in di. After that top 15 are selected and names of restaurants are stored in variable 'names' and number of outlets are stored in 'outlets' list. Then the bar graph is plotted and labels are given. The output is:



Question 3.

Part 2. *Plot the histogram of the aggregate rating of the restaurant(drop the unrated restaurant).*

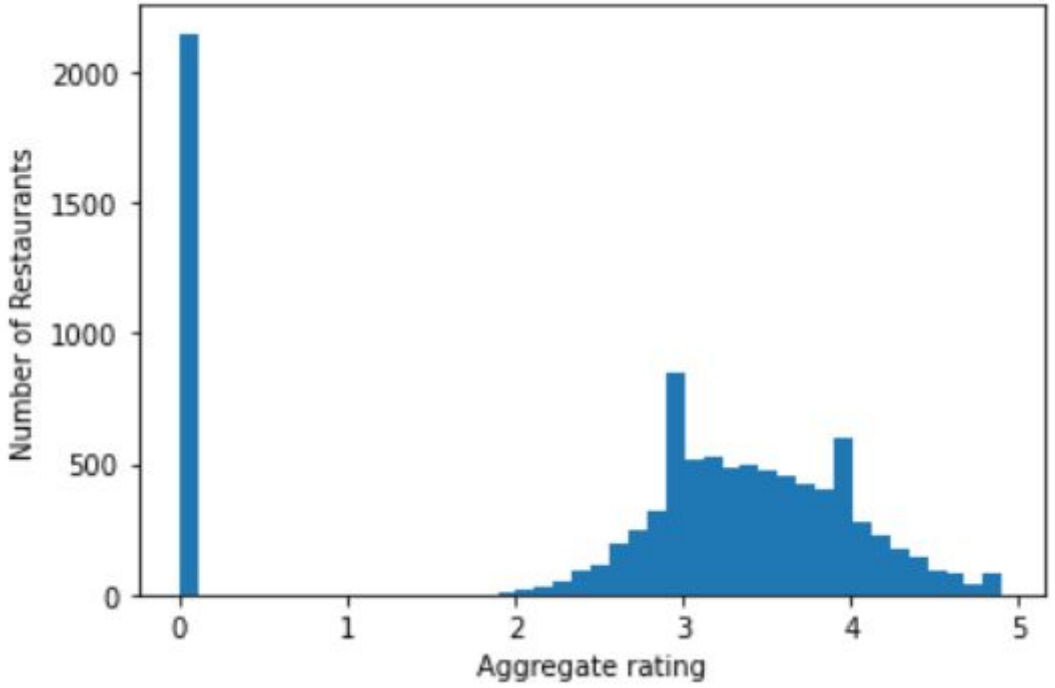
ANSWER:

The code to plot histogram of the aggregate ratings of the restaurants is:

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[(data['Rating text']!='Not Rated')|(data['Rating text']!='Not rated')|(data['Rating text']!='not rated')|(data['Rating text']!='')]
data.dropna(subset = ['Aggregate rating'],inplace = True)
ratings = data['Aggregate rating']
plt.hist(ratings,bins = "auto",  align='mid')
plt.xlabel('Aggregate rating')
plt.ylabel('Number of Restaurants')
plt.show()
```

Here, important libraries are imported and given csv file is read into variable ‘data’ using pandas. Then the columns where Rating text is ‘Not Rated’ or empty are dropped. Then if ‘Aggregate rating’ column is NULL, the row is dropped using dropna function of pandas. After that the aggregate ratings are stored in ‘ratings variable’, which is then plotted as histogram. The output is:



Question 3.

Part 3. *Plot the bar graph top 10 restaurants in the data with the highest number of votes.*

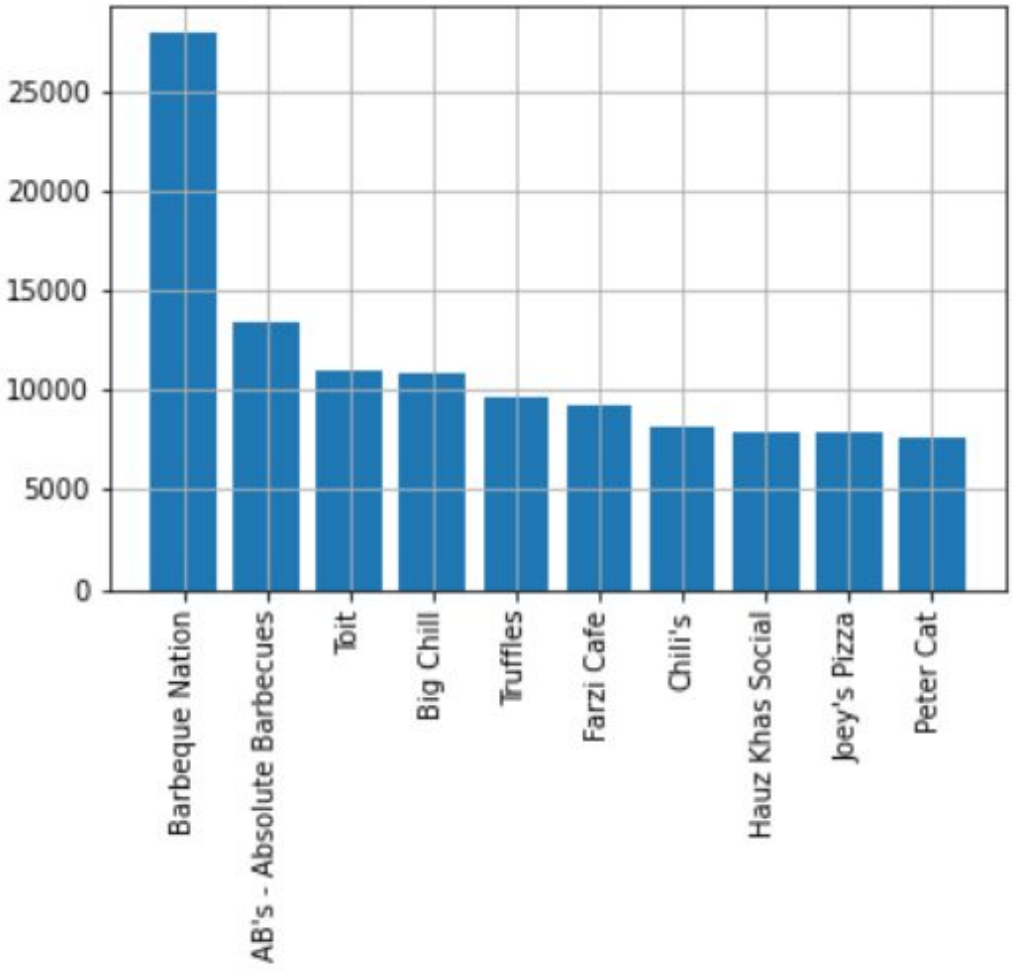
ANSWER:

The code for plotting the bar graph showing top 10 restaurants with the highest number of votes is:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==1]
d = {}
for i in data.index:
    k = data['Restaurant Name'][i]
    d[k] =  d.get(k,0) + data['Votes'][i]
dicti = dict(sorted(d.items(), key = lambda item:item[1], reverse = True))
names = [i for i in dicti][:10]
votes =  [dicti[i] for i in dicti][:10]
```

```
plt.bar(names,votes)
plt.grid()
plt.xticks(rotation=90)
plt.show()
```

After importing the required libraries and loading data into ‘data’ variable, I created a dictionary ‘d’, in which names of Restaurants are stored as keys and total number of votes they got as values. After that the dictionary is sorted in descending order to find the restaurants with maximum number of votes. After that keys (names of restaurants) of top 10 restaurants are stored into ‘names’ variable and values (number of votes) of top 10 restaurants are stored into ‘votes’ list. Then, simply the bar graph is plotted, which looks like:



Question 3.

Part 4. *Plot the pie graph of top 10 cuisines present in restaurants in the USA.*

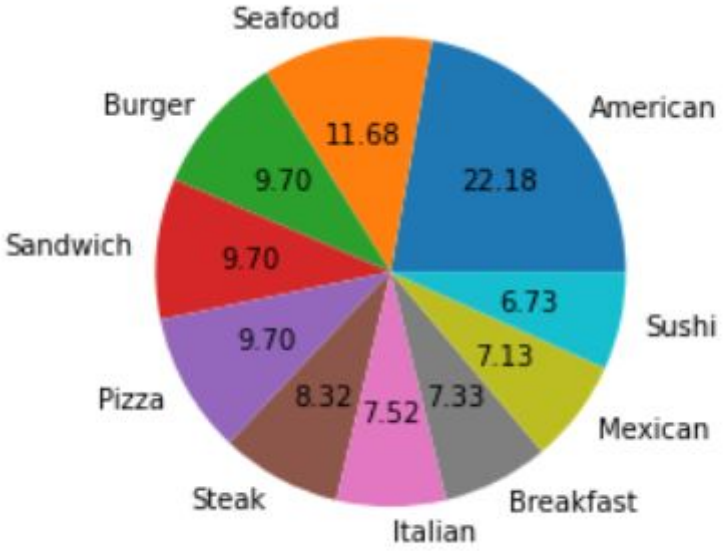
ANSWER:

The code for pie chart of top 10 cuisines of USA is:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==216]
data.dropna(subset = ['Cuisines'], inplace = True)
d = {}
for i in data.index:
```

```
    k = data['Cuisines'][i].split(', ')
    for j in k[:]:
        d[j] = d.get(j,0) + 1
di = dict(sorted(d.items(), key = lambda item:item[1], reverse = True))
names = [i for i in di][:10]
number = [di[i] for i in di][:10]
plt.pie(number,  labels = names, autopct = '%.2f')
plt.show()
```

Important libraries are important and data is read into 'data' variable using pandas. Since Country Code of USA is 216, data is kept only for rows where country code is 216 or country is USA. Then, wherever Cuisines value is NULL, the rows are dropped using dropna function of the pandas. After that, Cuisines are split on commas (since one restaurant can serve multiple cuisines) and stored as keys of dictionary 'd' while the numbers of restaurants serving that cuisine are served as values. Then the dictionary is sorted based on number of restaurants serving cuisines in descending order and stored in dictionary 'di'. Names of top 10 Cuisines are stored as a list named 'names' and number of restaurants serving top 10 cuisines are stored in list 'number'. Finally the pie chart is plotted and shown.



Question 3.

Part 5. *Plot the bubble graph of a number of Restaurants present in the city of India by keeping the weighted restaurant rating of the city in a bubble.*

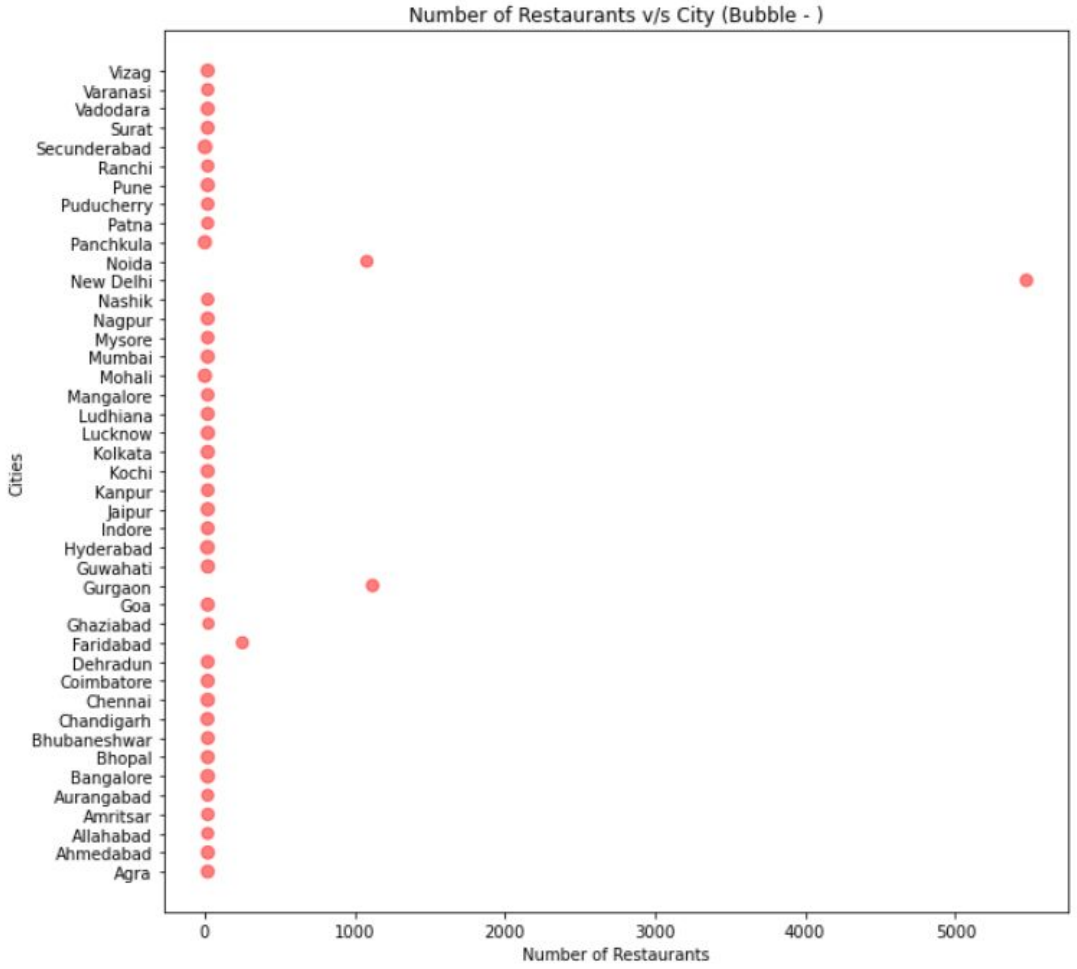
ANSWER:

The code of plotting bubble graph representing number of restaurants in different cities of India with the bubble showing the Weighted restaurant rating of the city is:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('zomato.csv',encoding = 'ISO-8859-1')
data = data[data['Country Code']==1]
data['W1'] = data['Aggregate rating'] * data['Votes']
d = {}
for i in data.index:
    k = data['City'][i]
```

```
    if k not in d:
        d[k] = [data['W1'][i], data['Votes'][i], 1]
    else:
        d[k][0] = d[k][0] + data['W1'][i]
        d[k][1] =  d[k][1] + data['Votes'][i]
        d[k][2] =  d[k][2] + 1
Weighted = [(d[i][0]/d[i][1])*15 for i in d]
rest = [d[i][2] for  i in d]
names = [i for i in d]
plt.figure(figsize=(10, 10))
plt.scatter(rest, names,Weighted,alpha = 0.5, c = 'red')
plt.title('Number of Restaurants v/s City (Bubble - )')
plt.xlabel('Number of Restaurants')
plt.ylabel('Cities')
plt.show()
```

Importing of important libraries and reading of csv file using pandas is done first. Then the data is kept for India only (using country code 1). Since Weighted Restaurant Rating=Σ (number of votes * rating) / Σ (number of votes) , I added the W1 column and stored the products of number of votes and rating in it. Then I created a dictionary ‘d’, added names of cities as Keys and W1 column(products of number of votes and rating), Votes, and number of restaurants as Values of the dictionary. Then I founded the Weighted Restaurants Rating by diving the W1 values sum and votes sum, and stored it in list named ‘Weighted’ (every value is made its 15 times as the true values plotted very small bubbles). I also stored the total restaurants per city in list ‘rest’ and names of cities in list ‘names’. Then I plotted the scatter graph simply by taking number of restaurants as x-axis, names of cities as y-axis and weighted restaurant rating as bubble size. I also extended the size of figure from regular to 10 by 10 using the figure function of matplotlib as the regular size mixed up the bubbles. Finally, labels are provided.



It can be seen that New Delhi has highest number of restaurants followed by Noida, Gurgaon and Goa while the rest of the cities have almost same number of restaurants. The weighted restaurant rating does not vary a lot.