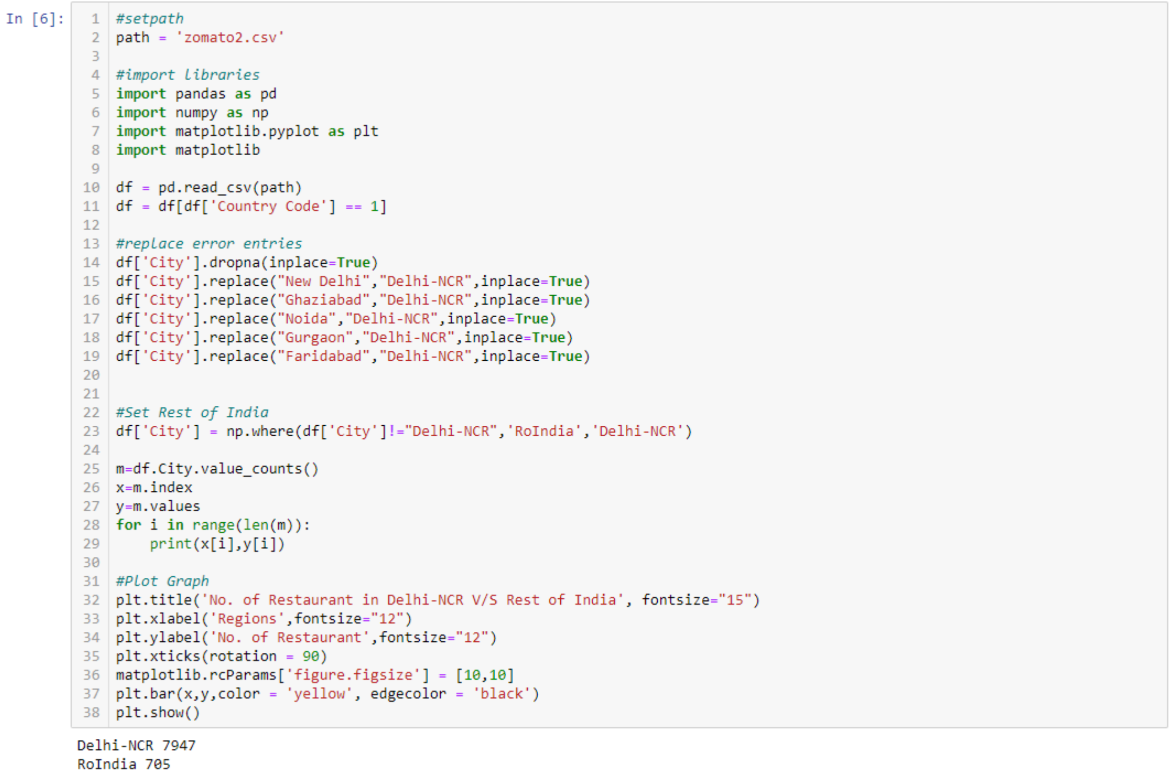
QUESTION 1

I. 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.

   a. Plot the bar graph of number of restaurants present in Delhi NCR vs Rest of India.
   b. 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.
   c. Find the top 10 cuisines served by maximum number of restaurants in Delhi NCR and rest of India.
   d. 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.

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

Code:

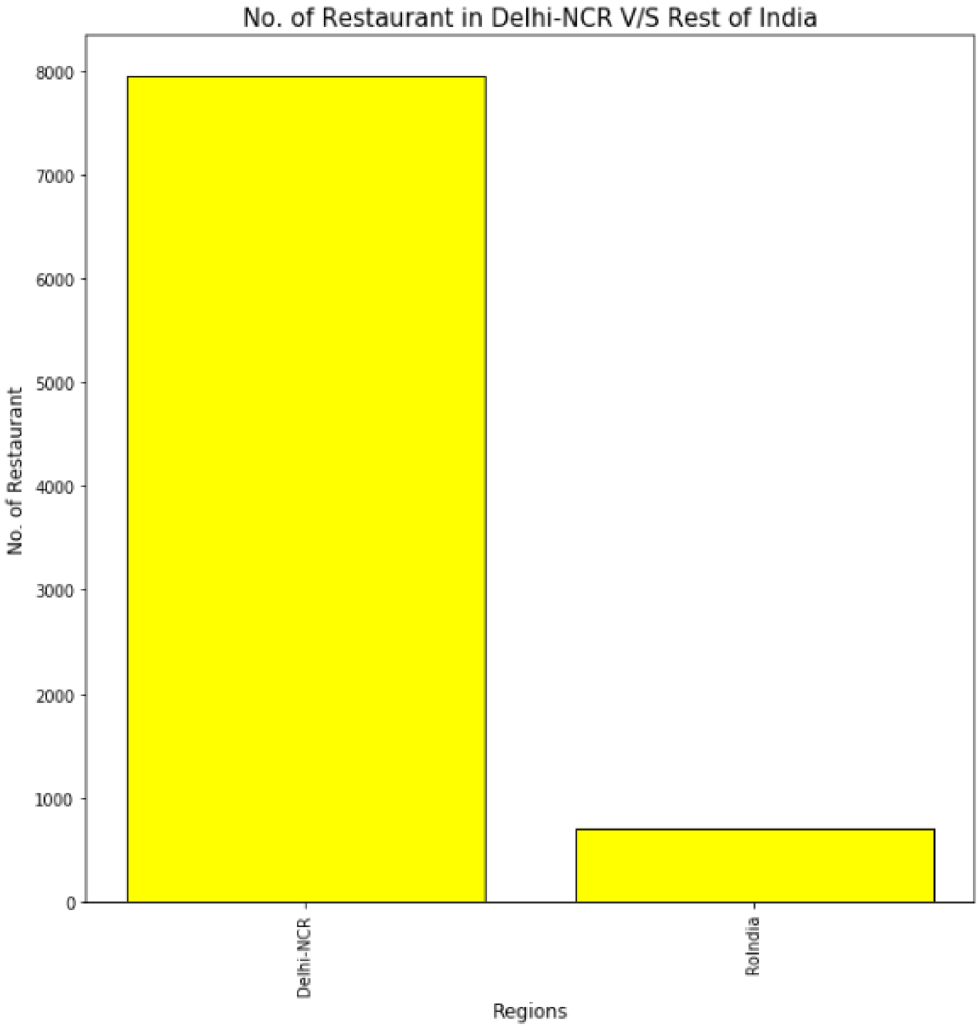


In [6]:

```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path)
df = df[df['Country Code'] == 1]

#replace error entries
df['City'].dropna(inplace=True)
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)


#Set Rest of India
df['City'] = np.where(df['City']!="Delhi-NCR",'RoIndia','Delhi-NCR')

m=df.City.value_counts()
x=m.index
y=m.values
for i in range(len(m)):
    print(x[i],y[i])

#Plot Graph
plt.title('No. of Restaurant in Delhi-NCR V/S Rest of India', fontsize="15")
plt.xlabel('Regions',fontsize="12")
plt.ylabel('No. of Restaurant',fontsize="12")
plt.xticks(rotation = 90)
matplotlib.rcParams['figure.figsize'] = [10,10]
plt.bar(x,y,color = 'yellow', edgecolor = 'black')
plt.show()
```

```
Delhi-NCR 7947
RoIndia 705
```

Graph:



Justification:

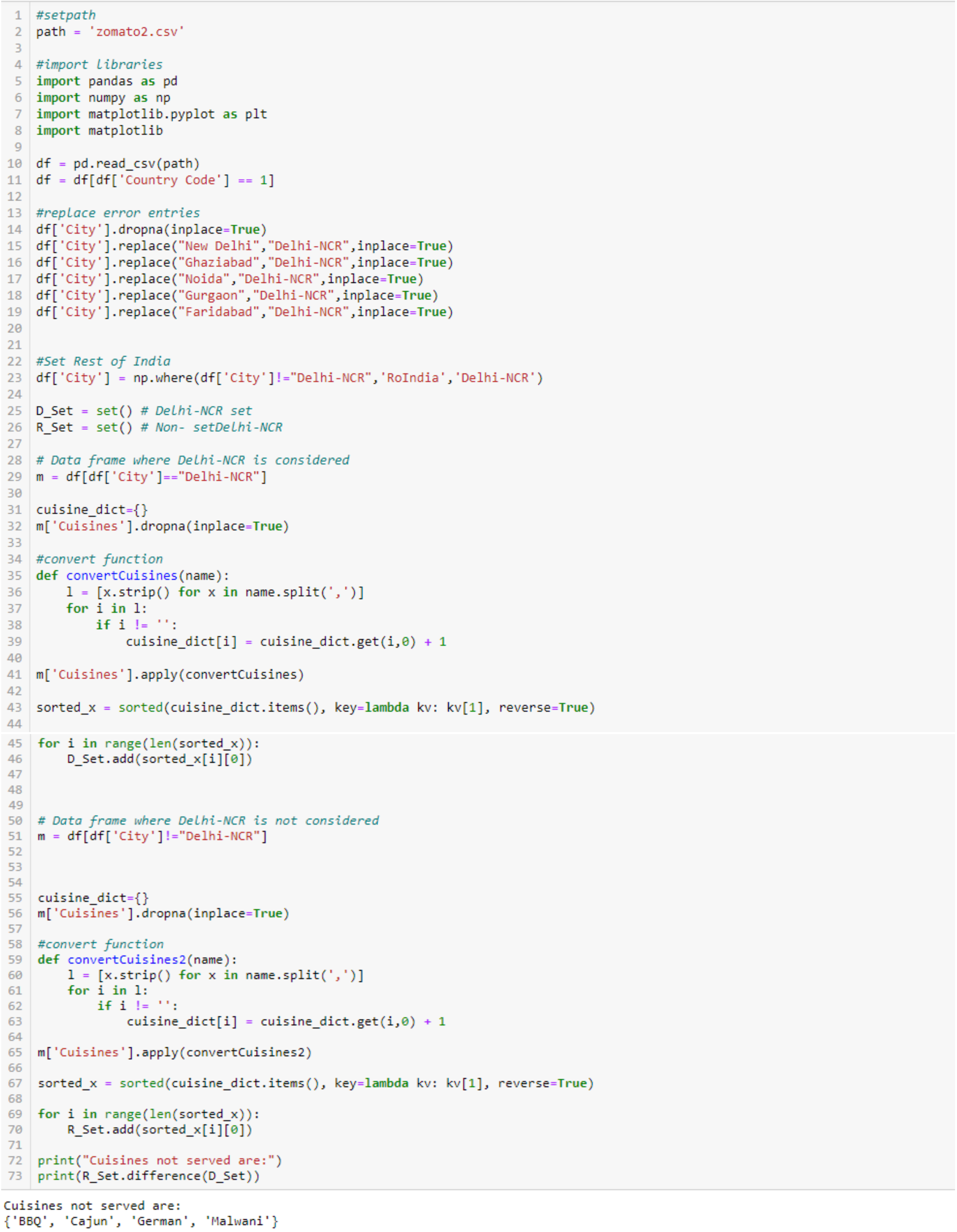
All the cities i.e. New Delhi, Ghaziabad, Noida, Gurgaon, Faridabad have been renamed to Delhi-NCR. Also all other cities name are converted to RestofIndia. A simple function of pandas value_counts() give us expected results. Plot is easily implemented on the output.

Libraries used:

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

I b) Find the cuisines which are not present in restaurant of Delhi NCR but present in rest of India.

Code:



```python
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path)
df = df[df['Country Code'] == 1]

#replace error entries
df['City'].dropna(inplace=True)
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)


#Set Rest of India
df['City'] = np.where(df['City']!="Delhi-NCR",'RoIndia','Delhi-NCR')

D_Set = set() # Delhi-NCR set
R_Set = set() # Non- setDelhi-NCR

# Data frame where Delhi-NCR is considered
m = df[df['City']=="Delhi-NCR"]

cuisine_dict={}
m['Cuisines'].dropna(inplace=True)

#convert function
def convertCuisines(name):
    l = [x.strip() for x in name.split(',')]
    for i in l:
        if i != '':
            cuisine_dict[i] = cuisine_dict.get(i,0) + 1

m['Cuisines'].apply(convertCuisines)

sorted_x = sorted(cuisine_dict.items(), key=lambda kv: kv[1], reverse=True)

for i in range(len(sorted_x)):
    D_Set.add(sorted_x[i][0])



# Data frame where Delhi-NCR is not considered
m = df[df['City']!="Delhi-NCR"]



cuisine_dict={}
m['Cuisines'].dropna(inplace=True)

#convert function
def convertCuisines2(name):
    l = [x.strip() for x in name.split(',')]
    for i in l:
        if i != '':
            cuisine_dict[i] = cuisine_dict.get(i,0) + 1

m['Cuisines'].apply(convertCuisines2)

sorted_x = sorted(cuisine_dict.items(), key=lambda kv: kv[1], reverse=True)

for i in range(len(sorted_x)):
    R_Set.add(sorted_x[i][0])

print("Cuisines not served are:")
print(R_Set.difference(D_Set))
```

```
Cuisines not served are:
{'BBQ', 'Cajun', 'German', 'Malwani'}
```

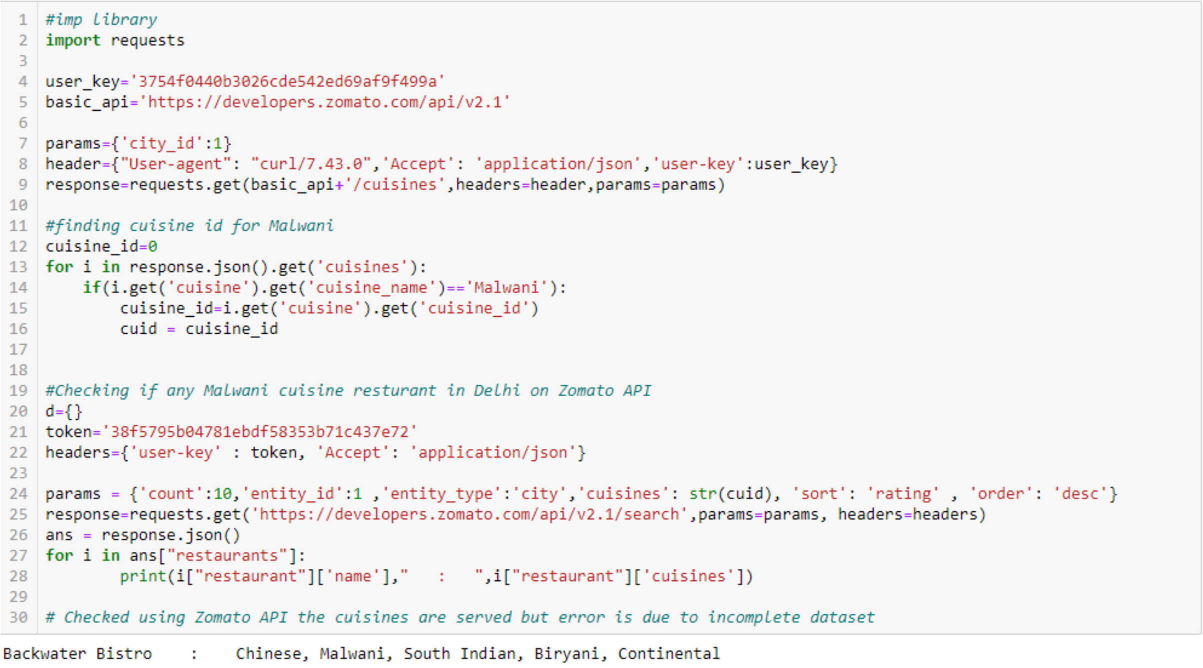
Justification:

In the output it can be seen cuisines not served are:

BBQ, Cajun, German, Malwani

Check using Zomato API whether this cuisines are actually not served in restaurants of Delhi-NCR or just it due to incomplete dataset.

Code:



```
#imp library
import requests

user_key='3754f0440b3026cde542ed69af9f499a'
basic_api='https://developers.zomato.com/api/v2.1'

params={'city_id':1}
header={"User-agent": "curl/7.43.0",'Accept': 'application/json','user-key':user_key}
response=requests.get(basic_api+'/cuisines',headers=header,params=params)

#finding cuisine id for Malwani
cuisine_id=0
for i in response.json().get('cuisines'):
    if(i.get('cuisine').get('cuisine_name')=='Malwani'):
        cuisine_id=i.get('cuisine').get('cuisine_id')
        cuid = cuisine_id


#Checking if any Malwani cuisine resturant in Delhi on Zomato API
d={}
token='38f5795b04781ebdf58353b71c437e72'
headers={'user-key' : token, 'Accept': 'application/json'}

params = {'count':10,'entity_id':1 ,'entity_type':'city','cuisines': str(cuid), 'sort': 'rating' , 'order': 'desc'}
response=requests.get('https://developers.zomato.com/api/v2.1/search',params=params, headers=headers)
ans = response.json()
for i in ans["restaurants"]:
        print(i["restaurant"]['name'],"   :   ",i["restaurant"]['cuisines'])

# Checked using Zomato API the cuisines are served but error is due to incomplete dataset
```

```
Backwater Bistro    :    Chinese, Malwani, South Indian, Biryani, Continental
```

Justification:

In the output:

Code for Delhi is used to find the cuisine_id for Malwani.

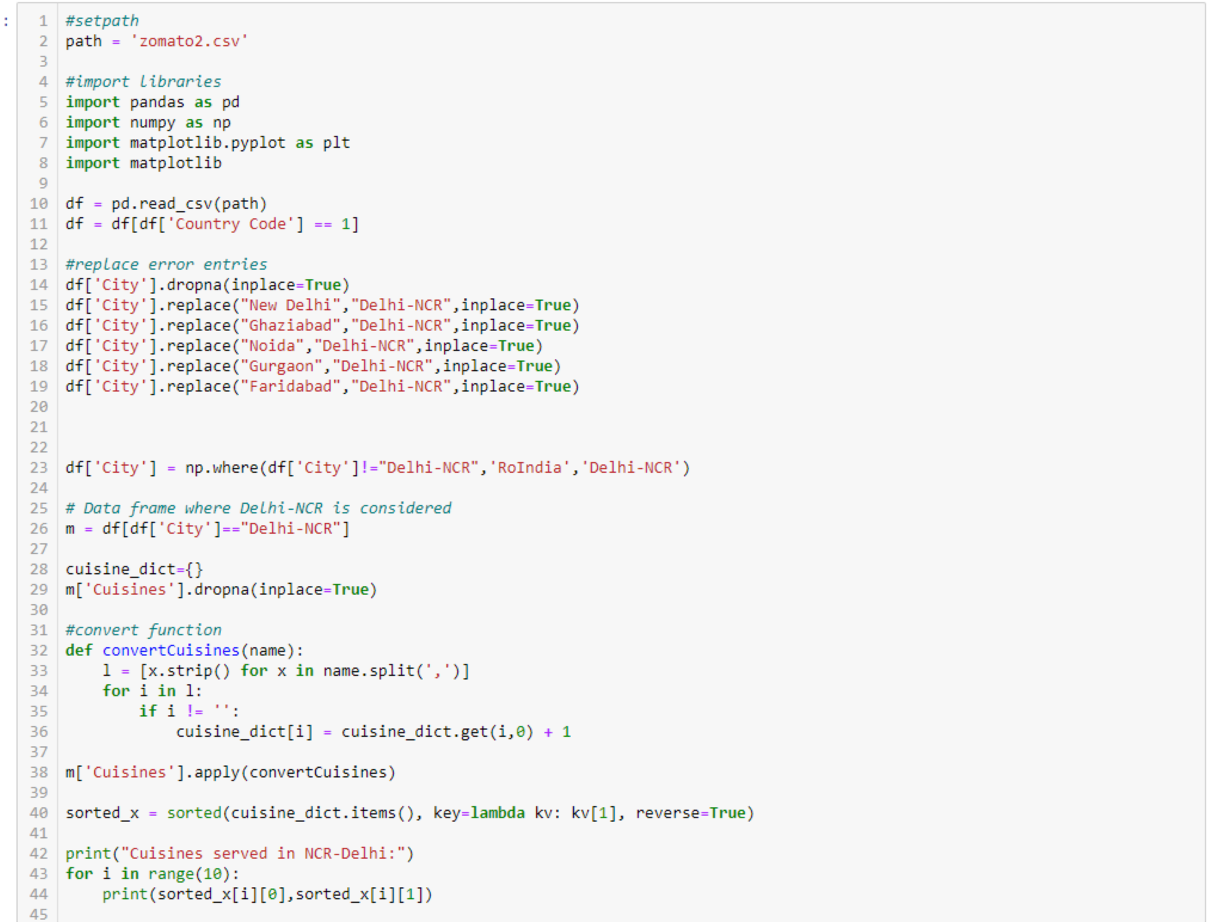
Checking if any Malwani cuisine restaurant is present in Delhi on Zomato API.

Yes a restaurant is present in Delhi serving Malwani food.

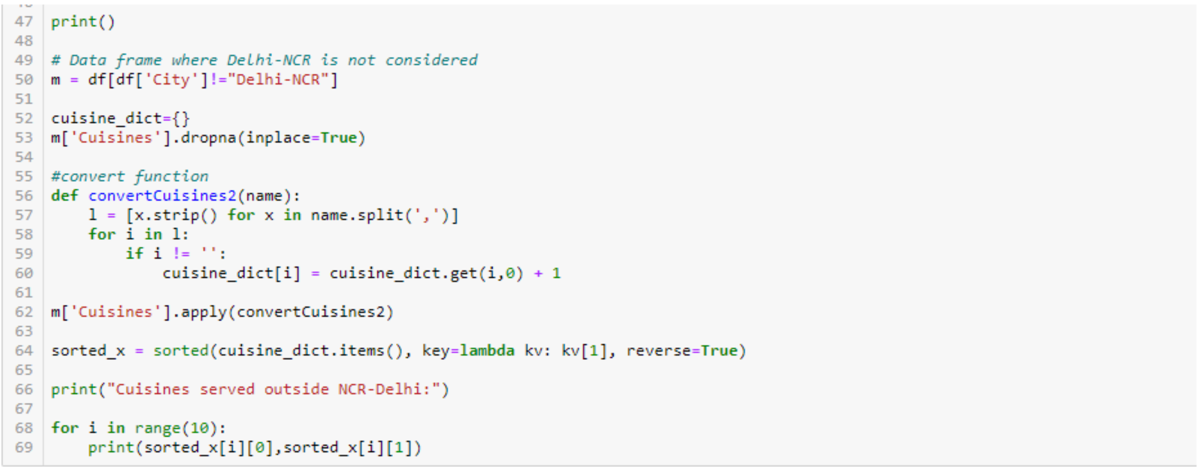
Checked using Zomato API the cuisines are served but error is due to incomplete dataset

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

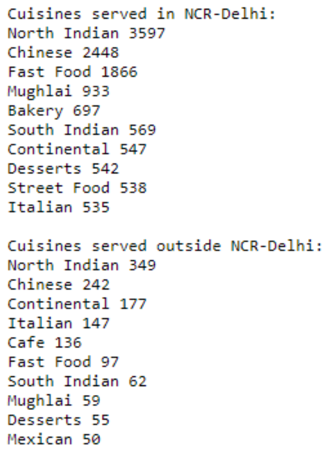
Code:



```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path)
df = df[df['Country Code'] == 1]

#replace error entries
df['City'].dropna(inplace=True)
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)



df['City'] = np.where(df['City']!="Delhi-NCR",'RoIndia','Delhi-NCR')

# Data frame where Delhi-NCR is considered
m = df[df['City']=="Delhi-NCR"]

cuisine_dict={}
m['Cuisines'].dropna(inplace=True)

#convert function
def convertCuisines(name):
    l = [x.strip() for x in name.split(',')]
    for i in l:
        if i != '':
            cuisine_dict[i] = cuisine_dict.get(i,0) + 1

m['Cuisines'].apply(convertCuisines)

sorted_x = sorted(cuisine_dict.items(), key=lambda kv: kv[1], reverse=True)

print("Cuisines served in NCR-Delhi:")
for i in range(10):
    print(sorted_x[i][0],sorted_x[i][1])

```



```
print()

# Data frame where Delhi-NCR is not considered
m = df[df['City']!="Delhi-NCR"]

cuisine_dict={}
m['Cuisines'].dropna(inplace=True)

#convert function
def convertCuisines2(name):
    l = [x.strip() for x in name.split(',')]
    for i in l:
        if i != '':
            cuisine_dict[i] = cuisine_dict.get(i,0) + 1

m['Cuisines'].apply(convertCuisines2)

sorted_x = sorted(cuisine_dict.items(), key=lambda kv: kv[1], reverse=True)

print("Cuisines served outside NCR-Delhi:")

for i in range(10):
    print(sorted_x[i][0],sorted_x[i][1])
```



```
Cuisines served in NCR-Delhi:
North Indian 3597
Chinese 2448
Fast Food 1866
Mughlai 933
Bakery 697
South Indian 569
Continental 547
Desserts 542
Street Food 538
Italian 535

Cuisines served outside NCR-Delhi:
North Indian 349
Chinese 242
Continental 177
Italian 147
Cafe 136
Fast Food 97
South Indian 62
Mughlai 59
Desserts 55
Mexican 50
```

Justification:

In the output:

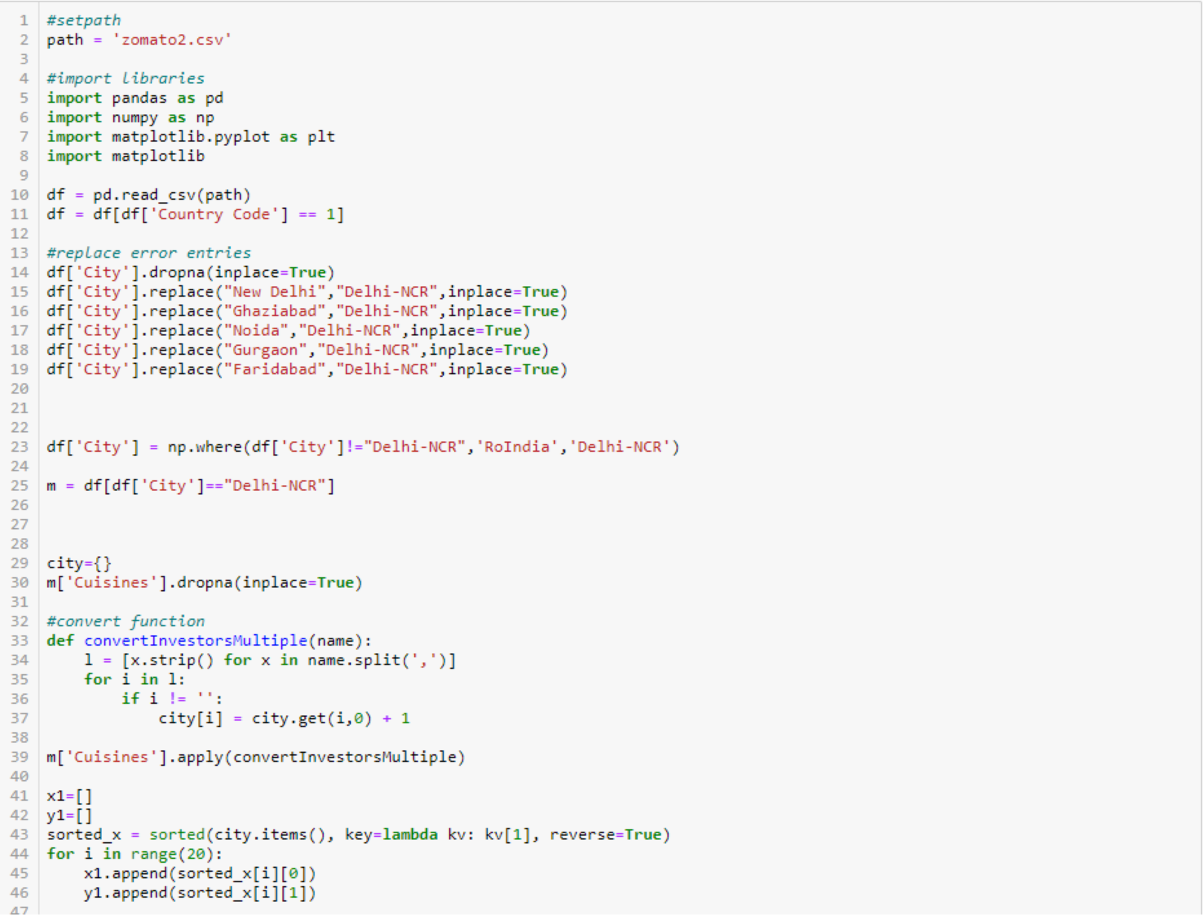
All the cities i.e. New Delhi, Ghaziabad, Noida, Gurgaon, Faridabad have been renamed to Delhi-NCR. Also all other cities name are converted to RestofIndia. A simple function of pandas value_counts() give us expected results.

A dictionary cuisine_dict counts the cuisines present in Delhi NCR.

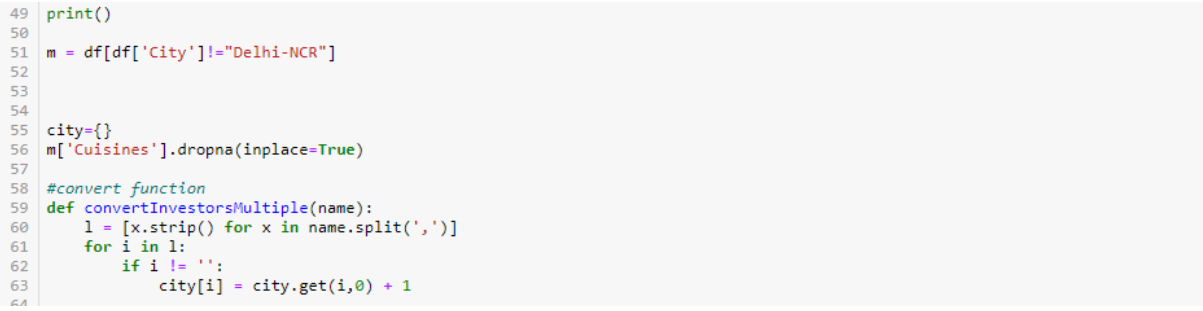
A simple convert function is applied on column ‘Cuisines’ to get desired output for Delhi NCR and Rest of India.

I d) 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.

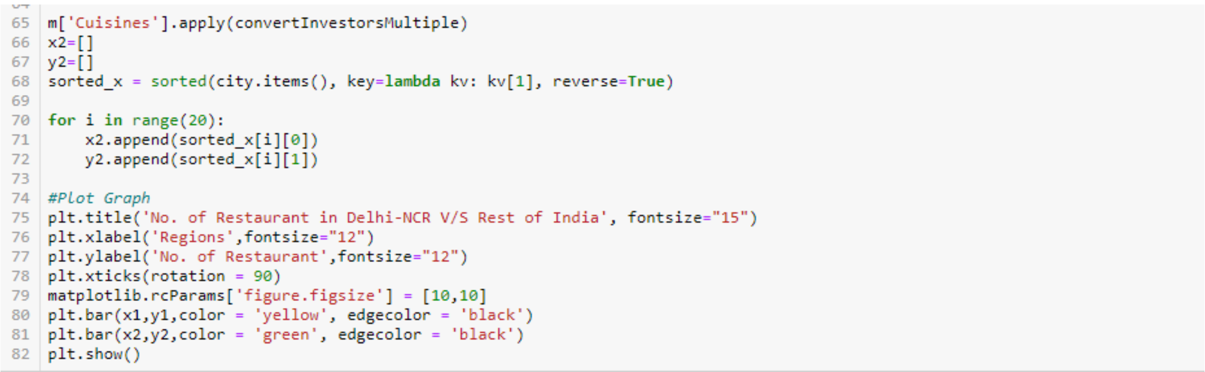
Code:



```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path)
df = df[df['Country Code'] == 1]

#replace error entries
df['City'].dropna(inplace=True)
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)



df['City'] = np.where(df['City']!="Delhi-NCR",'RoIndia','Delhi-NCR')

m = df[df['City']=="Delhi-NCR"]



city={}
m['Cuisines'].dropna(inplace=True)

#convert function
def convertInvestorsMultiple(name):
    l = [x.strip() for x in name.split(',')]
    for i in l:
        if i != '':
            city[i] = city.get(i,0) + 1

m['Cuisines'].apply(convertInvestorsMultiple)

x1=[]
y1=[]
sorted_x = sorted(city.items(), key=lambda kv: kv[1], reverse=True)
for i in range(20):
    x1.append(sorted_x[i][0])
    y1.append(sorted_x[i][1])
```

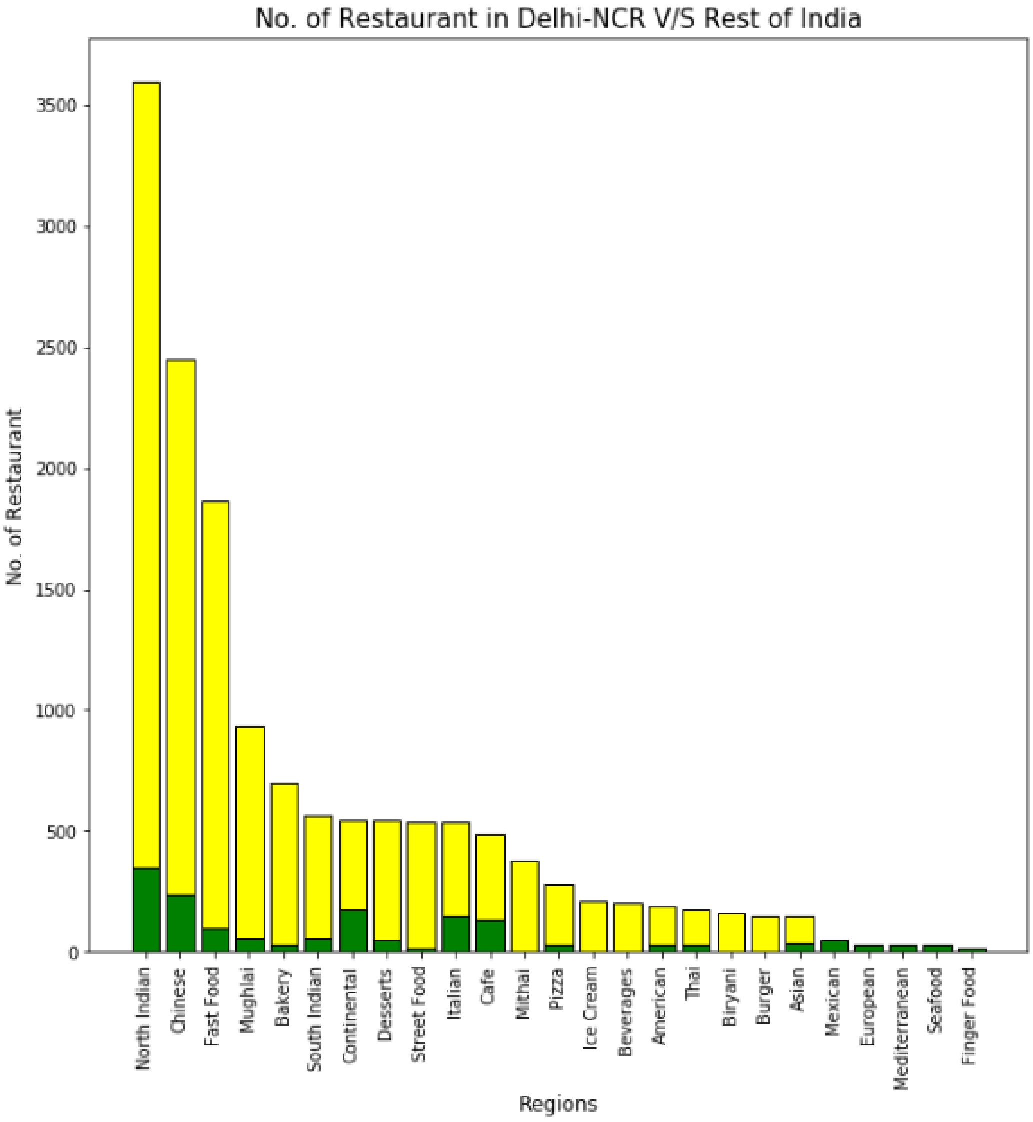


```
print()

m = df[df['City']!="Delhi-NCR"]



city={}
m['Cuisines'].dropna(inplace=True)

#convert function
def convertInvestorsMultiple(name):
    l = [x.strip() for x in name.split(',')]
    for i in l:
        if i != '':
            city[i] = city.get(i,0) + 1
```



```
m['Cuisines'].apply(convertInvestorsMultiple)
x2=[]
y2=[]
sorted_x = sorted(city.items(), key=lambda kv: kv[1], reverse=True)

for i in range(20):
    x2.append(sorted_x[i][0])
    y2.append(sorted_x[i][1])

#Plot Graph
plt.title('No. of Restaurant in Delhi-NCR V/S Rest of India', fontsize="15")
plt.xlabel('Regions',fontsize="12")
plt.ylabel('No. of Restaurant',fontsize="12")
plt.xticks(rotation = 90)
matplotlib.rcParams['figure.figsize'] = [10,10]
plt.bar(x1,y1,color = 'yellow', edgecolor = 'black')
plt.bar(x2,y2,color = 'green', edgecolor = 'black')
plt.show()
```

Plot:



Justification:

Prominent Cuisnes served are [Top 20]:

Cuisines served in NCR-Delhi:

North Indian, Chinese, Fast Food, Mughlai, Bakery, South Indian, Continental, Desserts, Street Food, Italian, Cafe, Mithai, Pizza, Ice Cream, Beverages, American, Thai, Biryani, Burger, Asian

Cuisines served outside NCR-Delhi:

North Indian, Chinese, Continental, Italian, Cafe, Fast Food, South Indian, Mughlai, Desserts, Mexican, Asian, European, American, Pizza, Mediterranean, Thai, Bakery, Seafood, Finger Food, Street Food

More rare and exotic food don’t make it to top 20 in Delhi-NCR.:

Mexican, European, Mediterranean, Seafood, Finger Food aren’t served.

While people of Delhi are mostly fond of:

Mughali, Biryani and Burger Joints

QUESTION 2

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

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

      i. Number of Votes given Restaurant

      ii. Restaurant serving more number of cuisines.

      iii. Average Cost of Restaurant

      iv. Restaurant serving some specific cuisines.

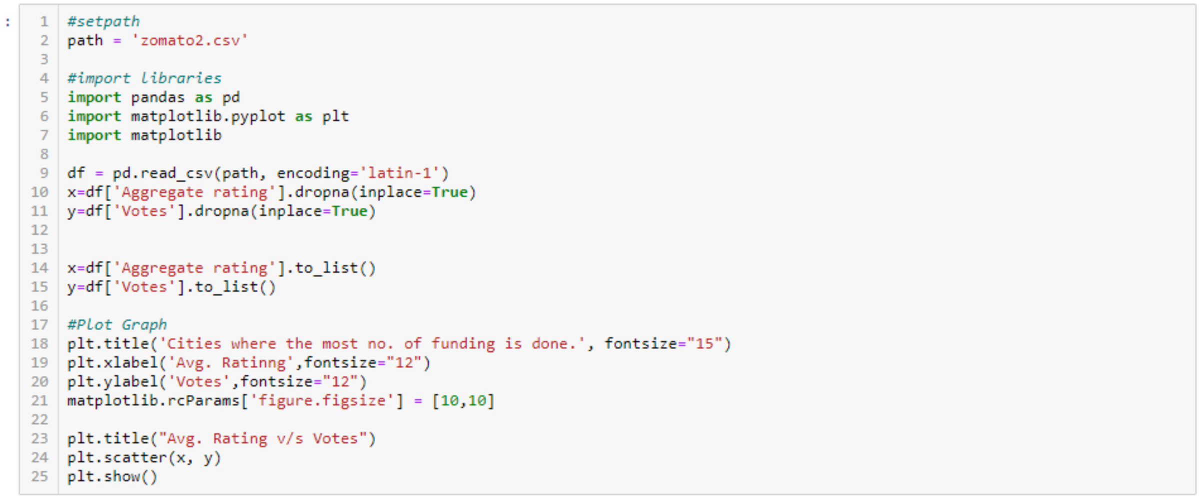
   b. Find the weighted restaurant rating of each locality and find out the top 10 localities with more weighted restaurant rating?

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

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

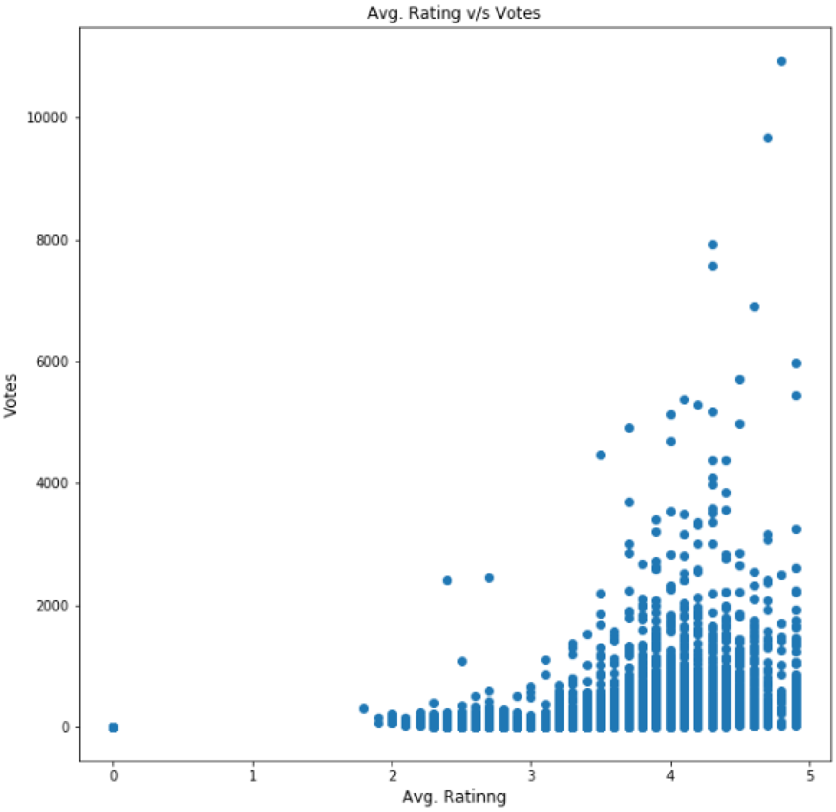
II a) i) Number of Votes given Restaurant

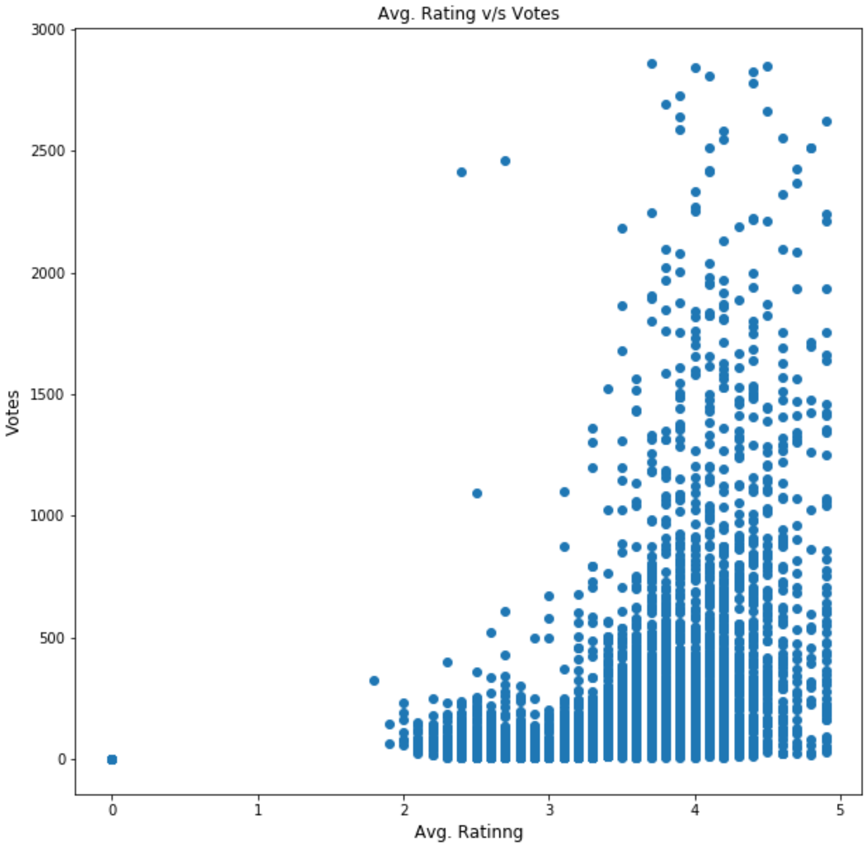
Code:



```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path, encoding='latin-1')
x=df['Aggregate rating'].dropna(inplace=True)
y=df['Votes'].dropna(inplace=True)


x=df['Aggregate rating'].to_list()
y=df['Votes'].to_list()

#Plot Graph
plt.title('Cities where the most no. of funding is done.', fontsize="15")
plt.xlabel('Avg. Ratinng',fontsize="12")
plt.ylabel('Votes',fontsize="12")
matplotlib.rcParams['figure.figsize'] = [10,10]

plt.title("Avg. Rating v/s Votes")
plt.scatter(x, y)
plt.show()
```

Plot:


Avg. Rating v/s Votes
Votes
Avg. Ratinng
0
2000
4000
6000
8000
10000
0
1
2
3
4
5


Avg. Rating v/s Votes
Votes
Avg. Ratinng
0
500
1000
1500
2000
2500
3000
0
1
2
3
4
5

Justification:

As can be seen in the output:

Most votes are concentrated around 4.0 Rating. In general more the number of votes higher the rating.

It can be easily seen that as votes increase there is a concertation around high numbered Rating.

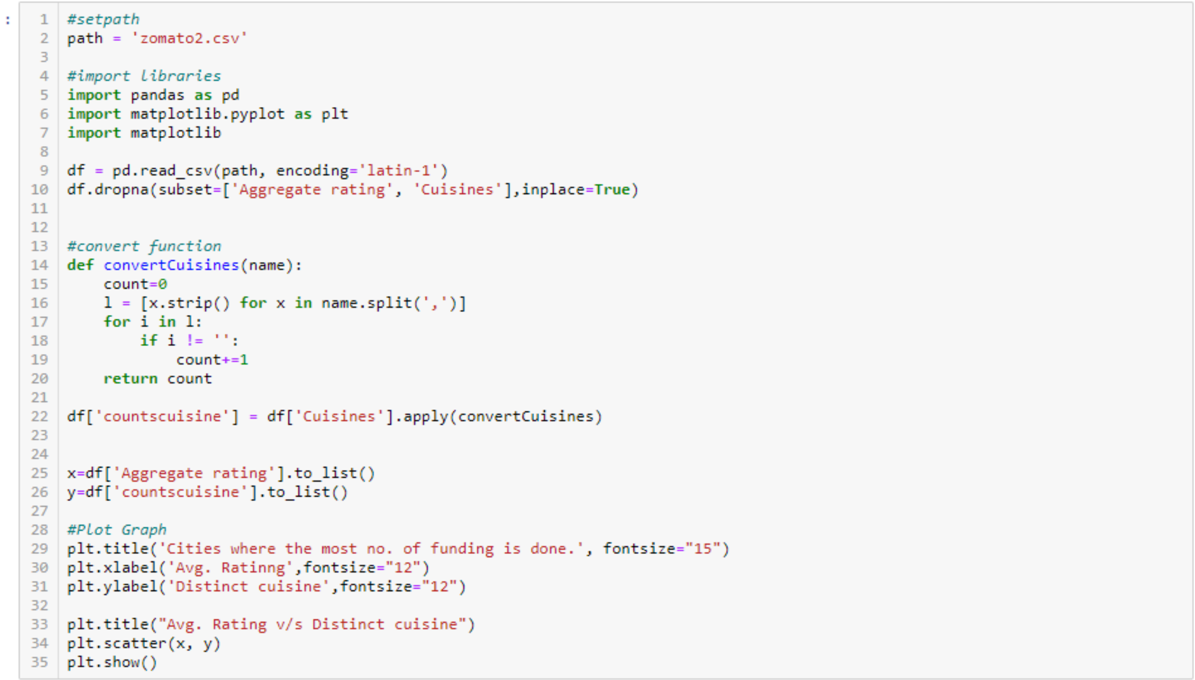
There are dots around (0,0) i.e. no votes no rating.

Libraries used:

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

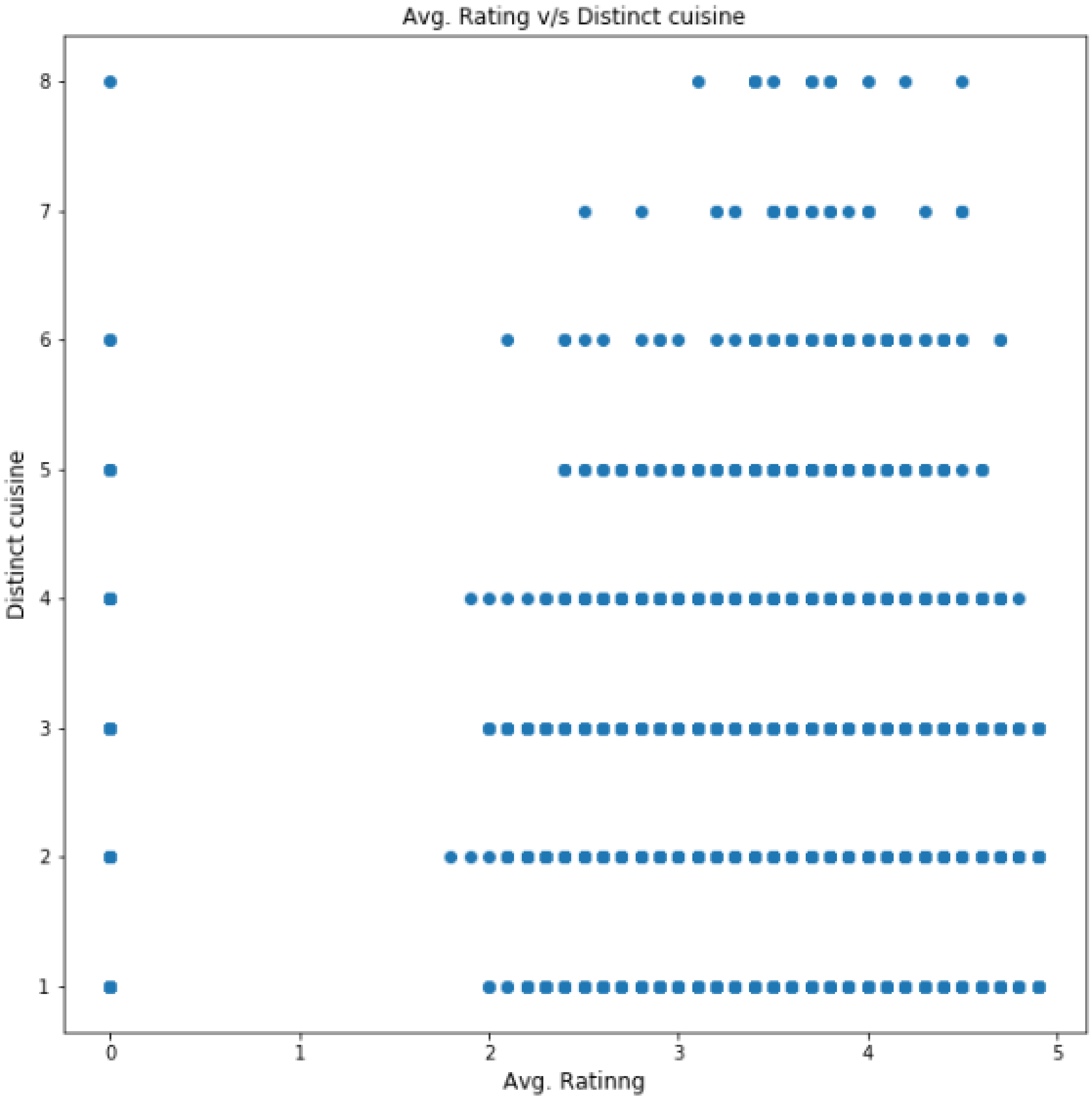
II a) ii) Restaurant serving more number of cuisines.

Code:



```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path, encoding='latin-1')
df.dropna(subset=['Aggregate rating', 'Cuisines'],inplace=True)


#convert function
def convertCuisines(name):
    count=0
    l = [x.strip() for x in name.split(',')]
    for i in l:
        if i != '':
            count+=1
    return count

df['countscuisine'] = df['Cuisines'].apply(convertCuisines)


x=df['Aggregate rating'].to_list()
y=df['countscuisine'].to_list()

#Plot Graph
plt.title('Cities where the most no. of funding is done.', fontsize="15")
plt.xlabel('Avg. Ratinng',fontsize="12")
plt.ylabel('Distinct cuisine',fontsize="12")

plt.title("Avg. Rating v/s Distinct cuisine")
plt.scatter(x, y)
plt.show()
```

Plot:



Justification:

As can be seen in the output:

Most Ratings are concentrated around 2.0 - 5.0 Rating. In general more the number of cuisines ratings lie in higher range 3.0 – 5.0.

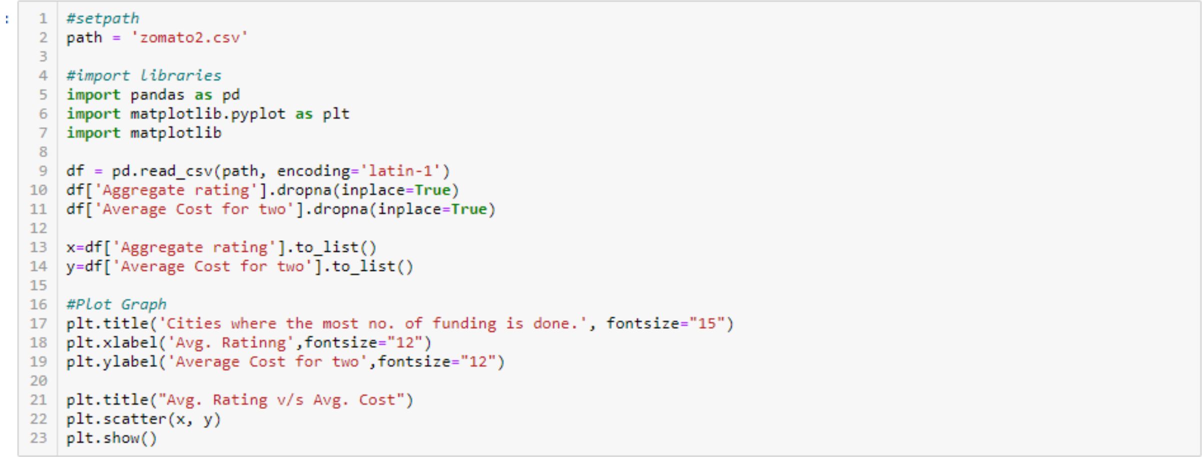
It can be easily seen that as cuisines increase there is a concertation around high numbered Rating.

Libraries used:

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

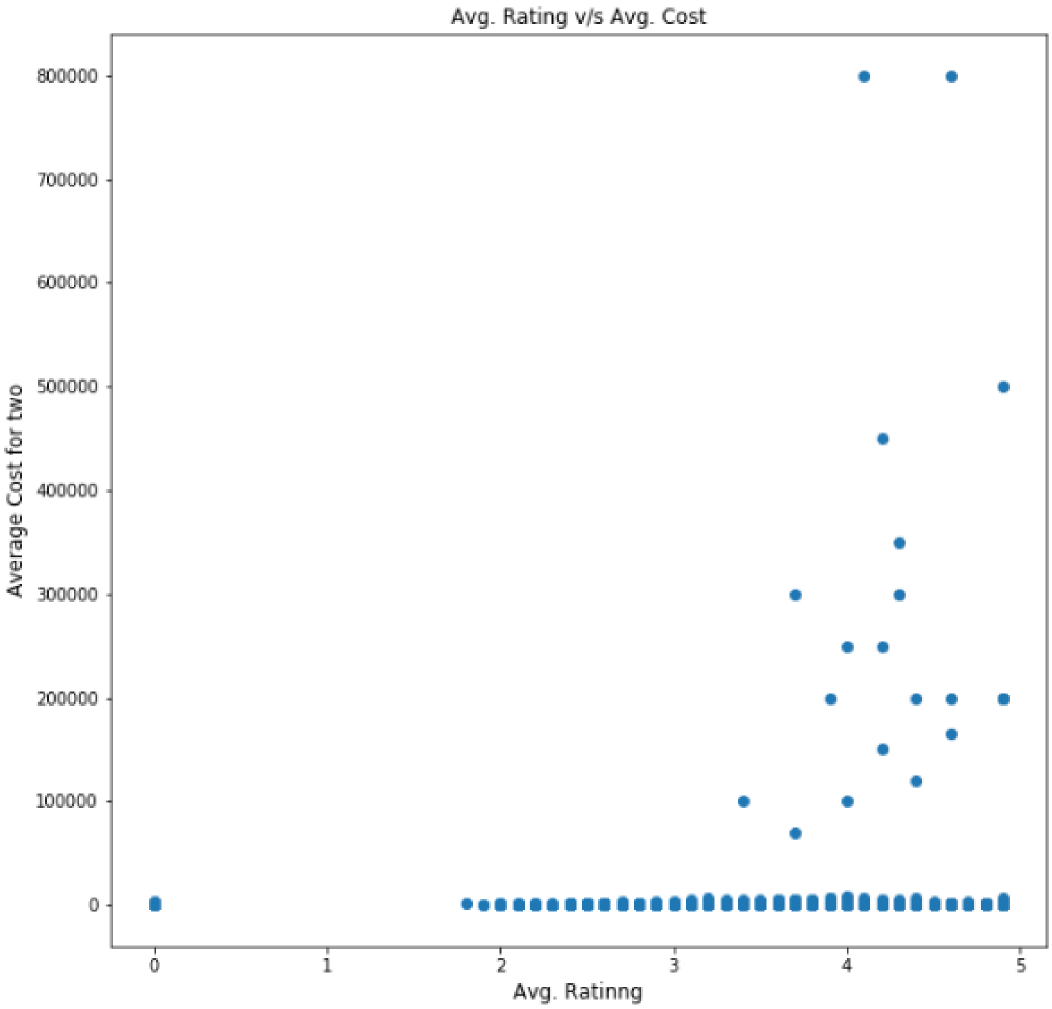
II a) iii) Average Cost of Restaurant

Code:

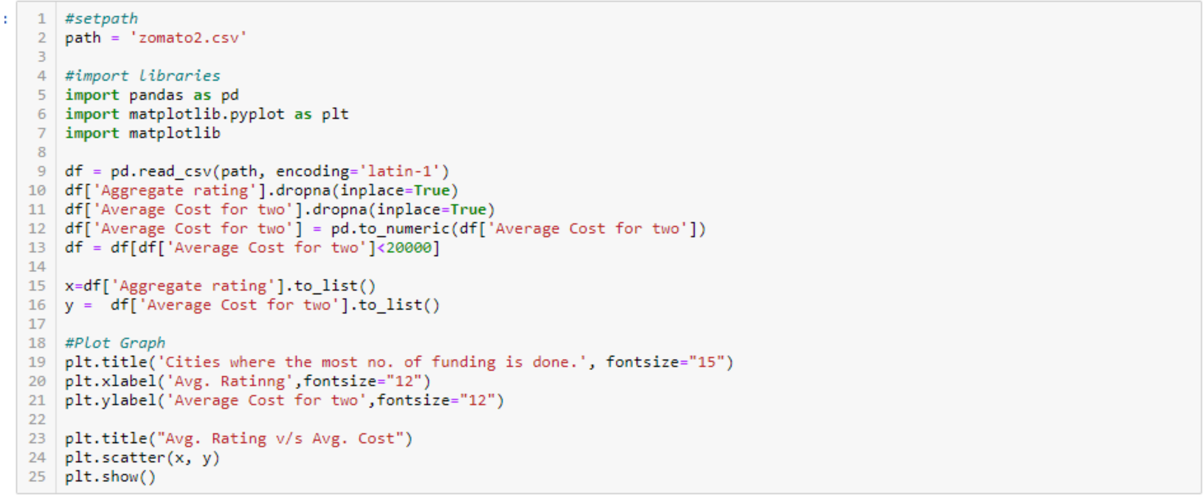


```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path, encoding='latin-1')
df['Aggregate rating'].dropna(inplace=True)
df['Average Cost for two'].dropna(inplace=True)

x=df['Aggregate rating'].to_list()
y=df['Average Cost for two'].to_list()

#Plot Graph
plt.title('Cities where the most no. of funding is done.', fontsize="15")
plt.xlabel('Avg. Ratinng',fontsize="12")
plt.ylabel('Average Cost for two',fontsize="12")

plt.title("Avg. Rating v/s Avg. Cost")
plt.scatter(x, y)
plt.show()
```

Plot:

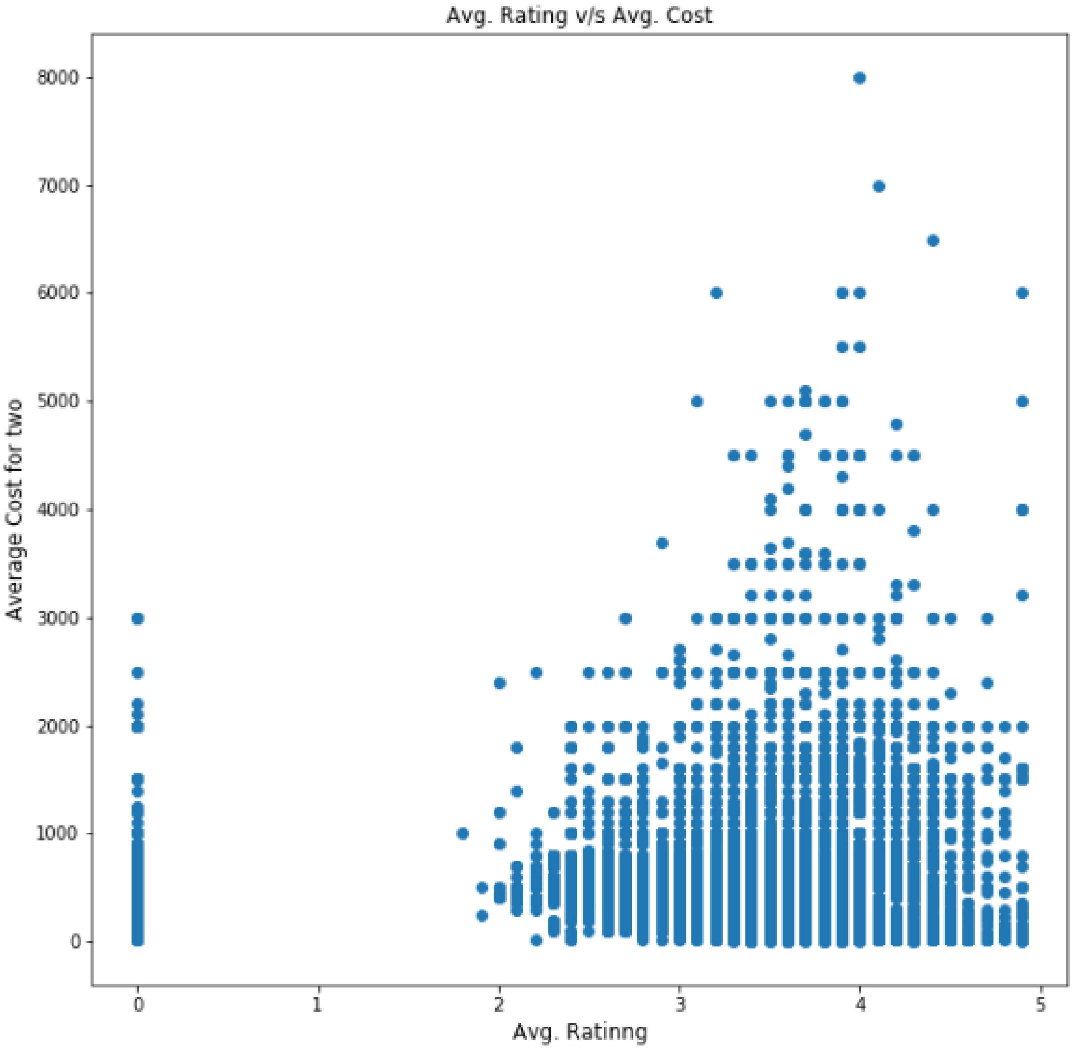


Code:



```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path, encoding='latin-1')
df['Aggregate rating'].dropna(inplace=True)
df['Average Cost for two'].dropna(inplace=True)
df['Average Cost for two'] = pd.to_numeric(df['Average Cost for two'])
df = df[df['Average Cost for two']<20000]

x=df['Aggregate rating'].to_list()
y =  df['Average Cost for two'].to_list()

#Plot Graph
plt.title('Cities where the most no. of funding is done.', fontsize="15")
plt.xlabel('Avg. Ratinng',fontsize="12")
plt.ylabel('Average Cost for two',fontsize="12")

plt.title("Avg. Rating v/s Avg. Cost")
plt.scatter(x, y)
plt.show()
```

Plot:



Justification:

As can be seen in the output:

Most Ratings are concentrated around 2.0 - 5.0 Rating. In general more Avg. cost ratings lie in higher range 3.0 – 5.0 . There is higher concentration around 4.0.

It can be easily seen that as Avg. cost increase there is a concertation around high numbered Rating.

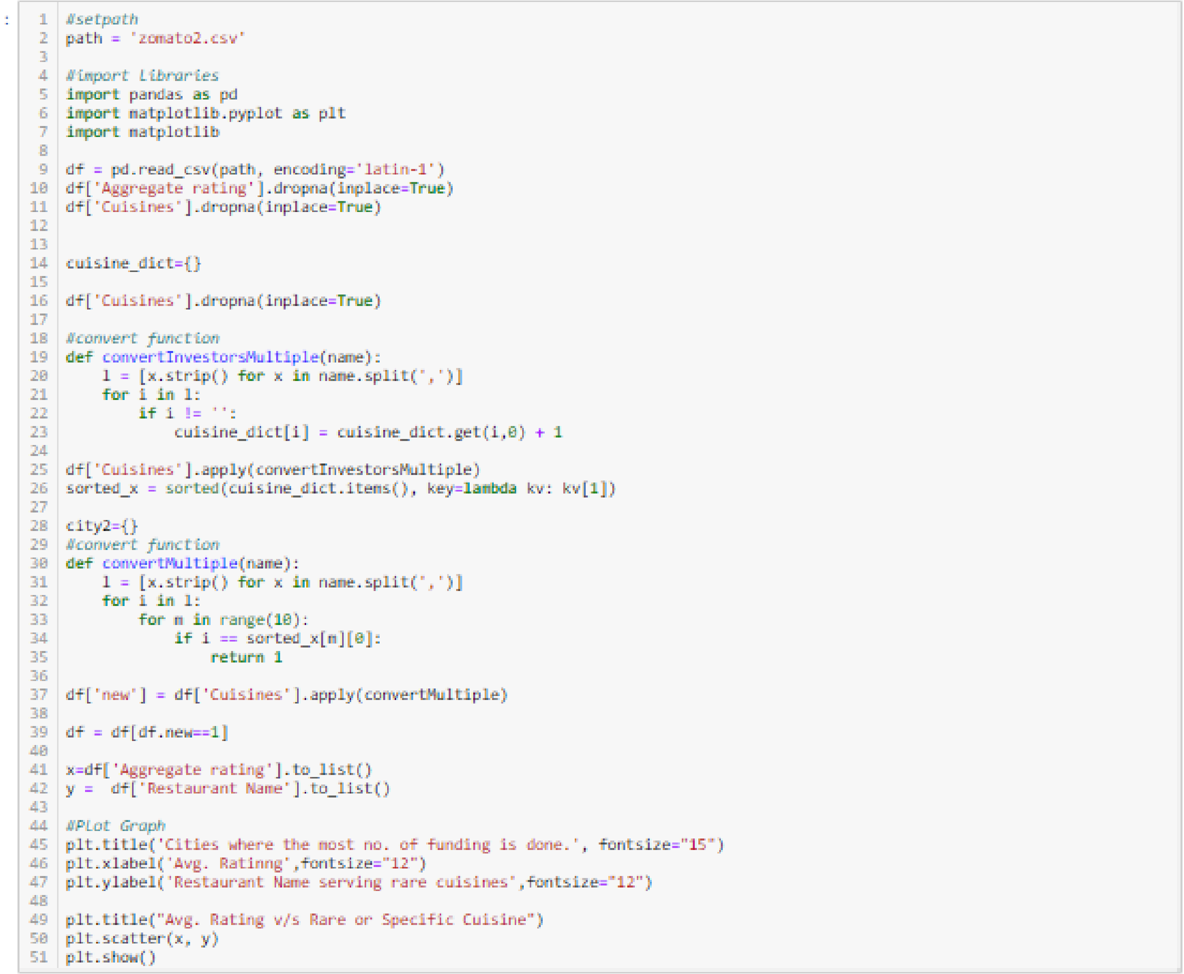
There are dots around (:,0) i.e. no votes no rating.

Libraries used:

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

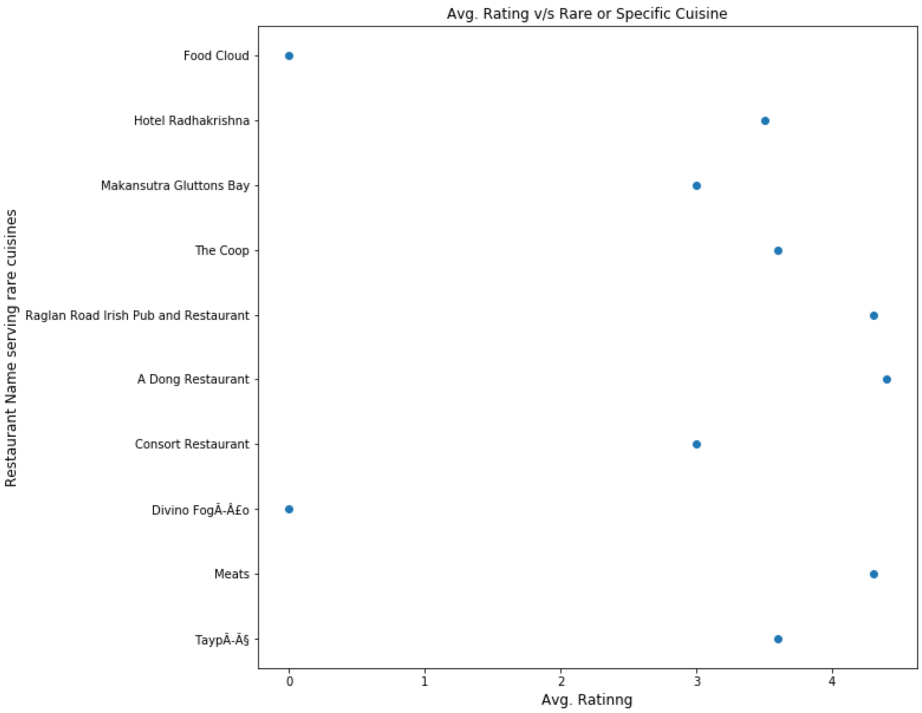
II a) iv) Restaurant serving some specific cuisines.

Code:

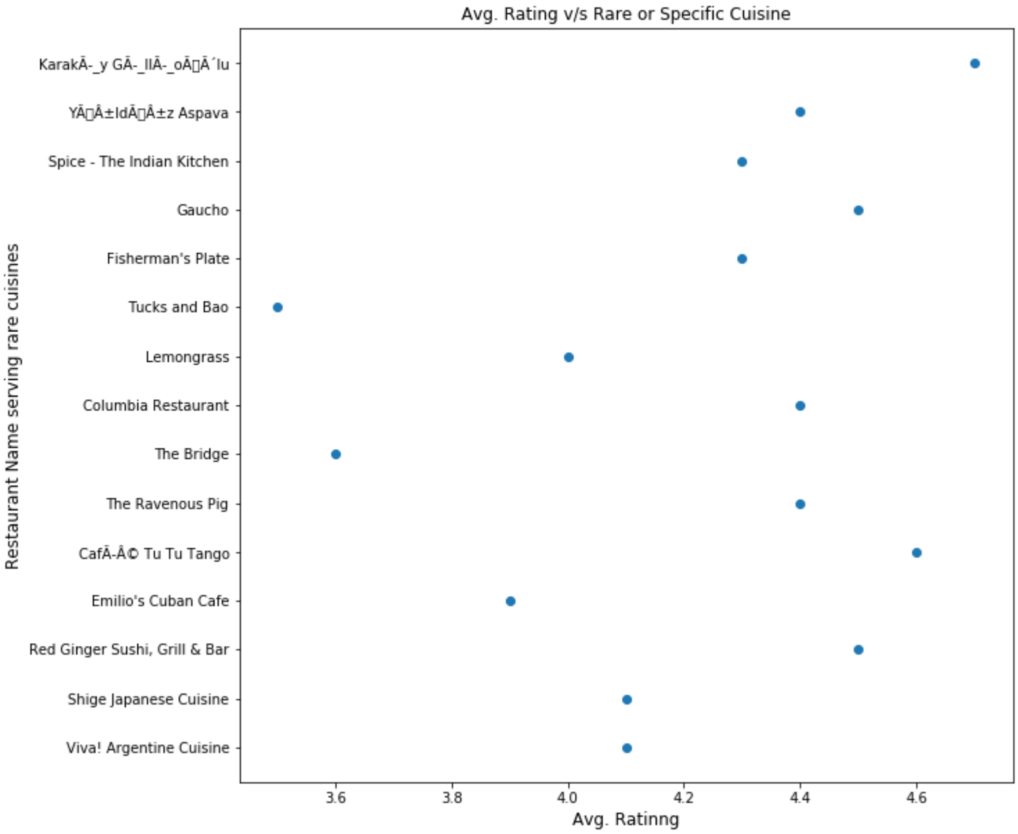


```
#setpath
path = 'zomato2.csv'

#import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path, encoding='latin-1')
df['Aggregate rating'].dropna(inplace=True)
df['Cuisines'].dropna(inplace=True)


cuisine_dict={}

df['Cuisines'].dropna(inplace=True)

#convert function
def convertInvestorsMultiple(name):
    l = [x.strip() for x in name.split(',')]
    for i in l:
        if i != '':
            cuisine_dict[i] = cuisine_dict.get(i,0) + 1

df['Cuisines'].apply(convertInvestorsMultiple)
sorted_x = sorted(cuisine_dict.items(), key=lambda kv: kv[1])

city2={}
#convert function
def convertMultiple(name):
    l = [x.strip() for x in name.split(',')]
    for i in l:
        for m in range(10):
            if i == sorted_x[m][0]:
                return 1

df['new'] = df['Cuisines'].apply(convertMultiple)

df = df[df.new==1]

x=df['Aggregate rating'].to_list()
y =  df['Restaurant Name'].to_list()

#PLot Graph
plt.title('Cities where the most no. of funding is done.', fontsize="15")
plt.xlabel('Avg. Ratinng',fontsize="12")
plt.ylabel('Restaurant Name serving rare cuisines',fontsize="12")

plt.title("Avg. Rating v/s Rare or Specific Cuisine")
plt.scatter(x, y)
plt.show()
```

Plot: (Top 10 Rare Cuisines)



Plot: (Top 10-20 Rare Cuisines)



Justification:

As can be seen in the output:

Most Ratings are concentrated around 3.0 - 5.0 Rating. In general more Rare Cuisines ratings lie in higher range 3.6 – 4.9 rating. There is higher concentration around 4.0. for top 10-20 rare cuisines.

It can be easily seen that for Rare Cuisines there is a concertation around high numbered Rating.

There are dots around (:,0) i.e. no votes no rating.

Libraries used:

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

Summary:

Number of Votes given Restaurant

Most votes are concentrated around 4.0 Rating. In general more the number of votes higher the rating.

It can be easily seen that as votes increase there is a concertation around high numbered Rating.

There are dots around (0,0) i.e. no votes no rating.

Restaurant serving more number of cuisines.

Most Ratings are concentrated around 2.0 - 5.0 Rating. In general more the number of cuisines ratings lie in higher range 3.0 – 5.0.

It can be easily seen that as cuisines increase there is a concertation around high numbered Rating.

Average Cost of Restaurant

Most Ratings are concentrated around 2.0 - 5.0 Rating. In general more Avg. cost ratings lie in higher range 3.0 – 5.0 . There is higher concentration around 4.0.

It can be easily seen that as Avg. cost increase there is a concertation around high numbered Rating.

There are dots around (:,0) i.e. no votes no rating.

Restaurant serving some specific cuisines.

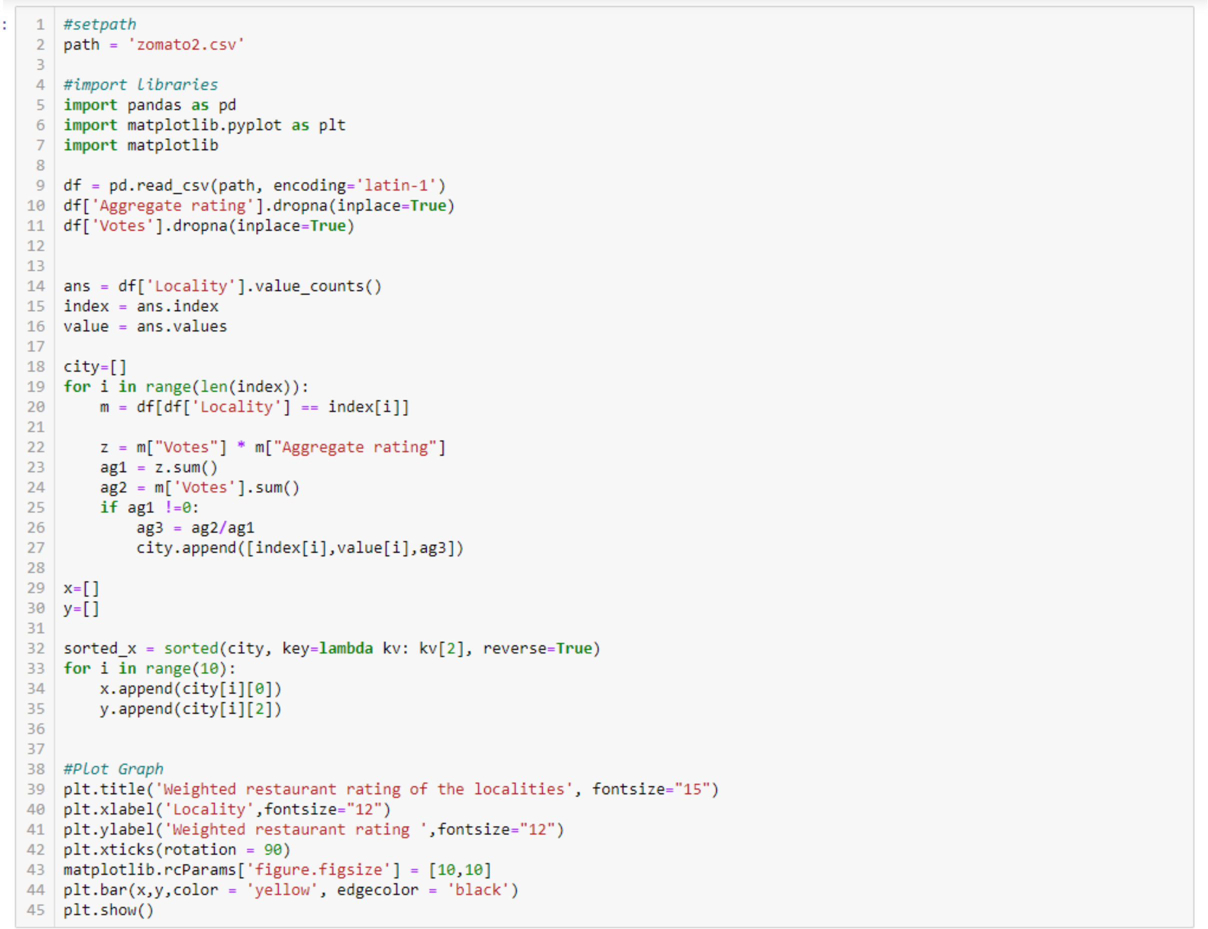
Most Ratings are concentrated around 3.0 - 5.0 Rating. In general more Rare Cuisines ratings lie in higher range 3.6 – 4.9 rating. There is higher concentration around 4.0. for top 10-20 rare cuisines.

It can be easily seen that for Rare Cuisines there is a concertation around high numbered Rating.

There are dots around (:,0) i.e. no votes no rating.

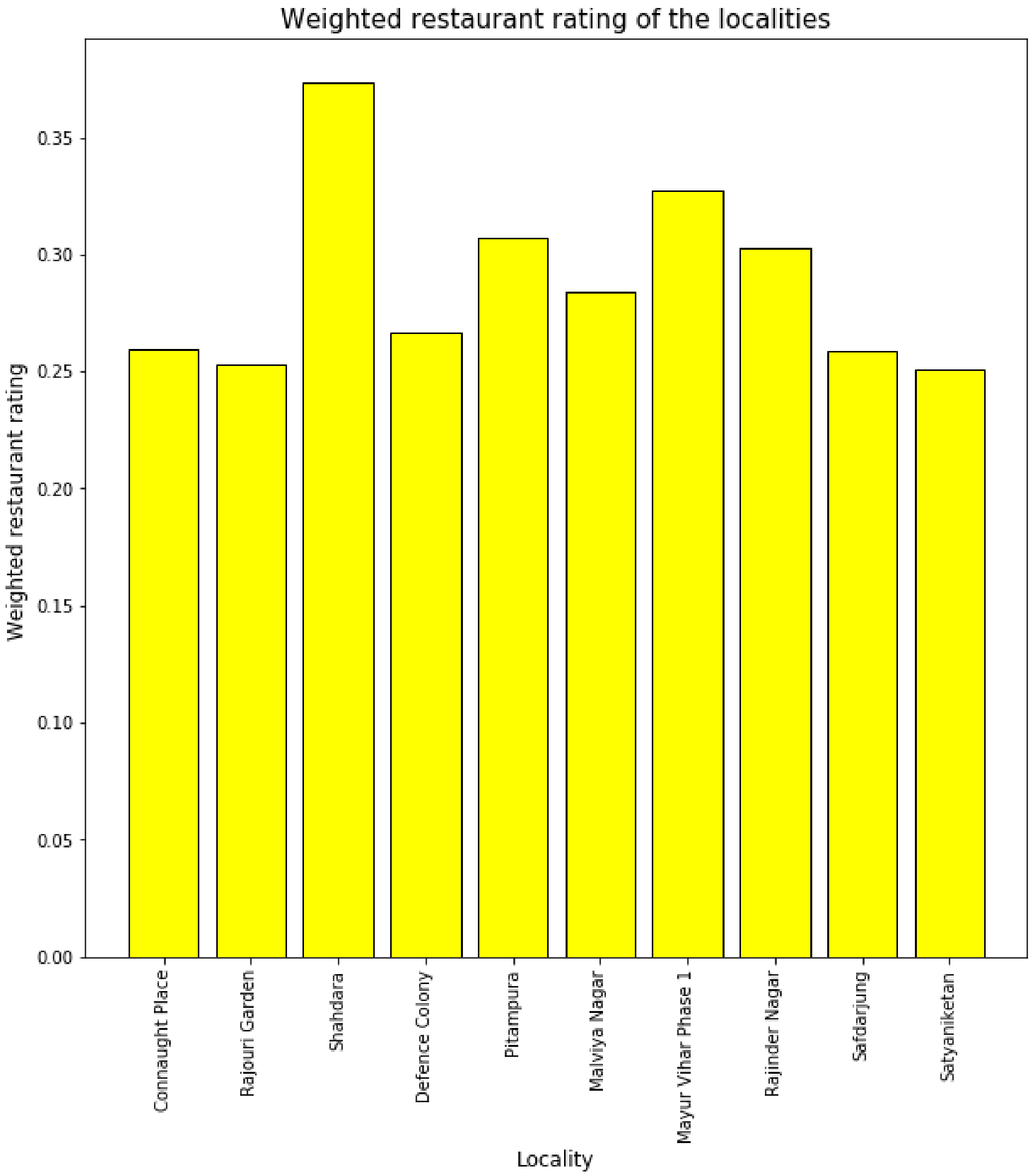
II b) i) a. Find the weighted restaurant rating of each locality and find out the top 10 localities with more weighted restaurant rating?

Code:



```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path, encoding='latin-1')
df['Aggregate rating'].dropna(inplace=True)
df['Votes'].dropna(inplace=True)


ans = df['Locality'].value_counts()
index = ans.index
value = ans.values

city=[]
for i in range(len(index)):
    m = df[df['Locality'] == index[i]]

    z = m["Votes"] * m["Aggregate rating"]
    ag1 = z.sum()
    ag2 = m['Votes'].sum()
    if ag1 !=0:
        ag3 = ag2/ag1
        city.append([index[i],value[i],ag3])

x=[]
y=[]

sorted_x = sorted(city, key=lambda kv: kv[2], reverse=True)
for i in range(10):
    x.append(city[i][0])
    y.append(city[i][2])


#Plot Graph
plt.title('Weighted restaurant rating of the localities', fontsize="15")
plt.xlabel('Locality',fontsize="12")
plt.ylabel('Weighted restaurant rating ',fontsize="12")
plt.xticks(rotation = 90)
matplotlib.rcParams['figure.figsize'] = [10,10]
plt.bar(x,y,color = 'yellow', edgecolor = 'black')
plt.show()
```

Plot:



Weighted restaurant rating of the localities
Weighted restaurant rating
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Connaught Place
Rajouri Garden
Shahdara
Defence Colony
Pitampura
Malviya Nagar
Mayur Vihar Phase 1
Rajinder Nagar
Safdarjung
Satyaniketan
Locality

Justification:

Weighted Restaurant Rating can be found using:

With Pandas multiplication is simple due to Vectorization.

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

Weighted restaurant ratings lie between 0.25 to 0.35 for localities.

Libraries used:

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

QUESTION 3

III. Visualization

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

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

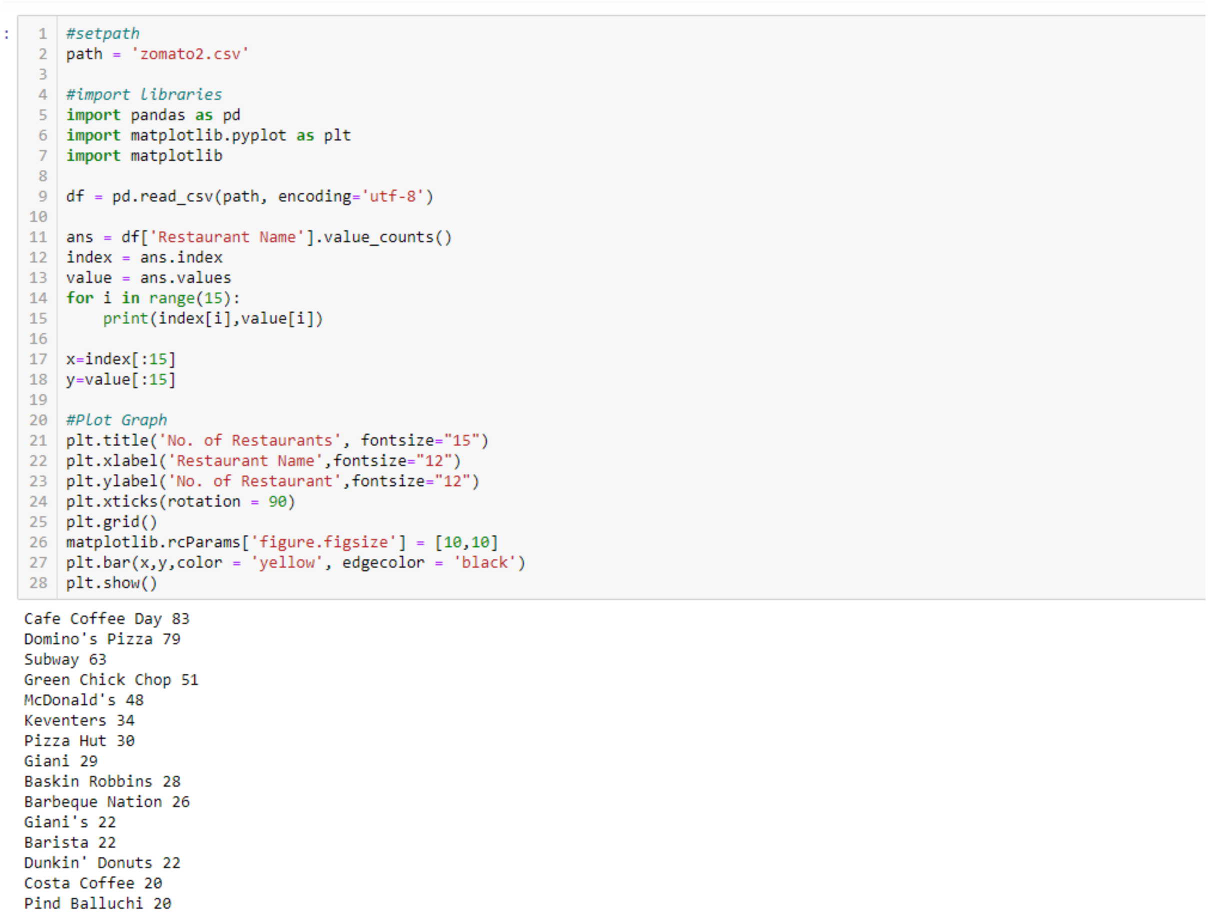
c. Plot the bar graph top 10 restaurants in the data with the highest number of votes.

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

e. 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.

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

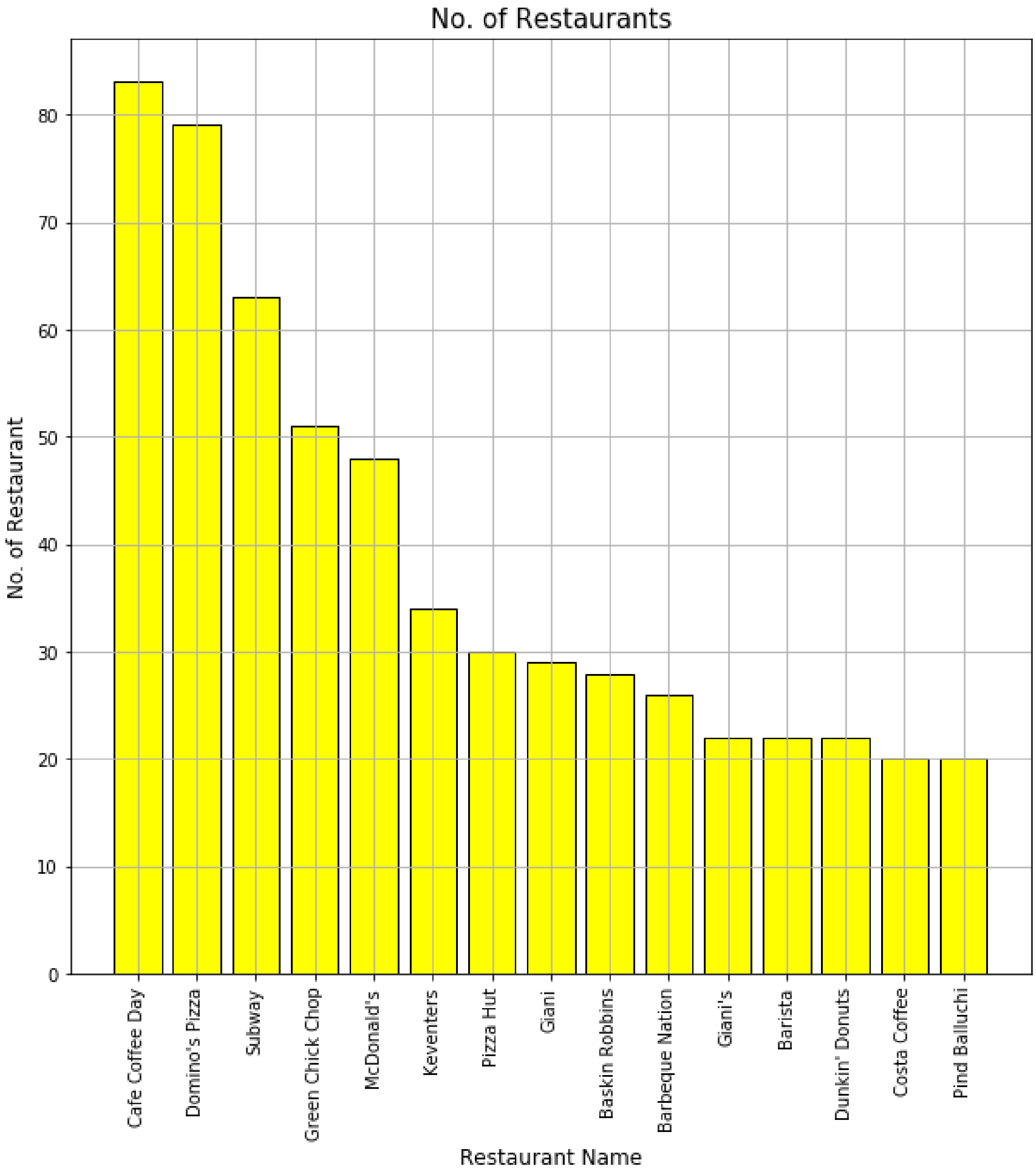
Code:



```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path, encoding='utf-8')

ans = df['Restaurant Name'].value_counts()
index = ans.index
value = ans.values
for i in range(15):
    print(index[i],value[i])

x=index[:15]
y=value[:15]

#Plot Graph
plt.title('No. of Restaurants', fontsize="15")
plt.xlabel('Restaurant Name',fontsize="12")
plt.ylabel('No. of Restaurant',fontsize="12")
plt.xticks(rotation = 90)
plt.grid()
matplotlib.rcParams['figure.figsize'] = [10,10]
plt.bar(x,y,color = 'yellow', edgecolor = 'black')
plt.show()
```

```
Cafe Coffee Day 83
Domino's Pizza 79
Subway 63
Green Chick Chop 51
McDonald's 48
Keventers 34
Pizza Hut 30
Giani 29
Baskin Robbins 28
Barbeque Nation 26
Giani's 22
Barista 22
Dunkin' Donuts 22
Costa Coffee 20
Pind Balluchi 20
```

Plot:

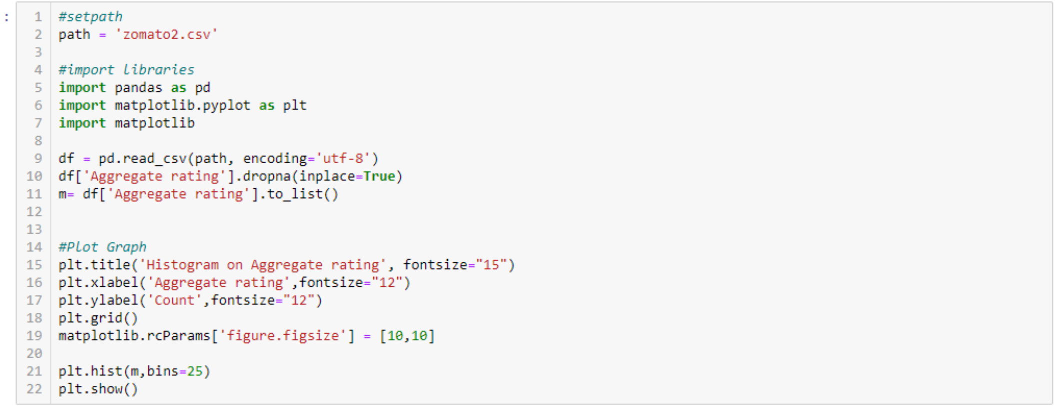


Justification:

A simple function of pandas value_counts() give us expected results. Plot is easily implemented on the Restaurant Name to find maximum outlets. Plot is easily implemented on the output using to_list() funcotionaliy of Pandas.

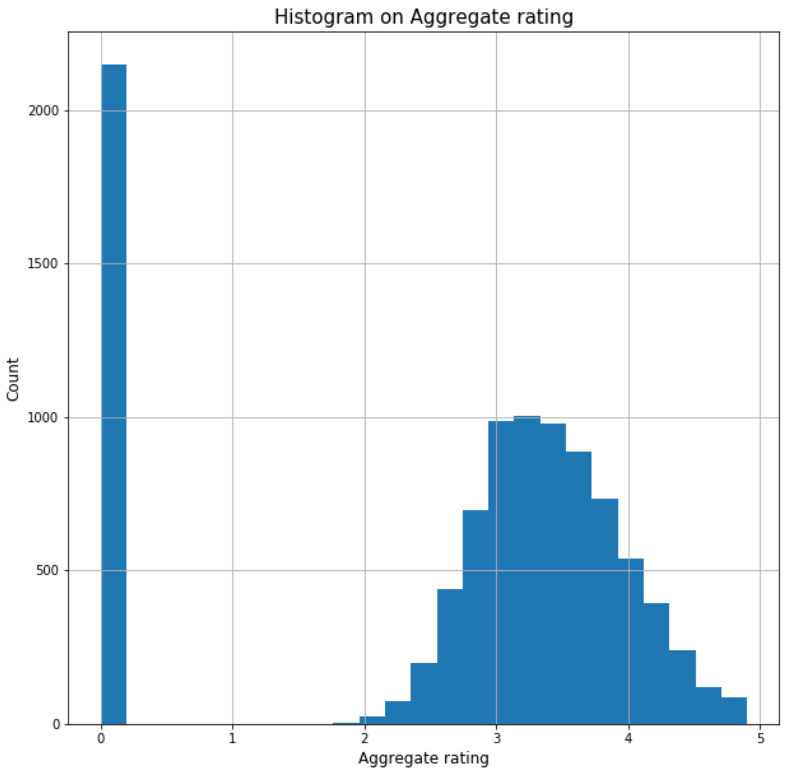
III b) Plot the histogram of aggregate rating of restaurant(drop the unrated restaurant).

Code:



```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path, encoding='utf-8')
df['Aggregate rating'].dropna(inplace=True)
m= df['Aggregate rating'].to_list()


#Plot Graph
plt.title('Histogram on Aggregate rating', fontsize="15")
plt.xlabel('Aggregate rating',fontsize="12")
plt.ylabel('Count',fontsize="12")
plt.grid()
matplotlib.rcParams['figure.figsize'] = [10,10]

plt.hist(m,bins=25)
plt.show()
```

Plot:

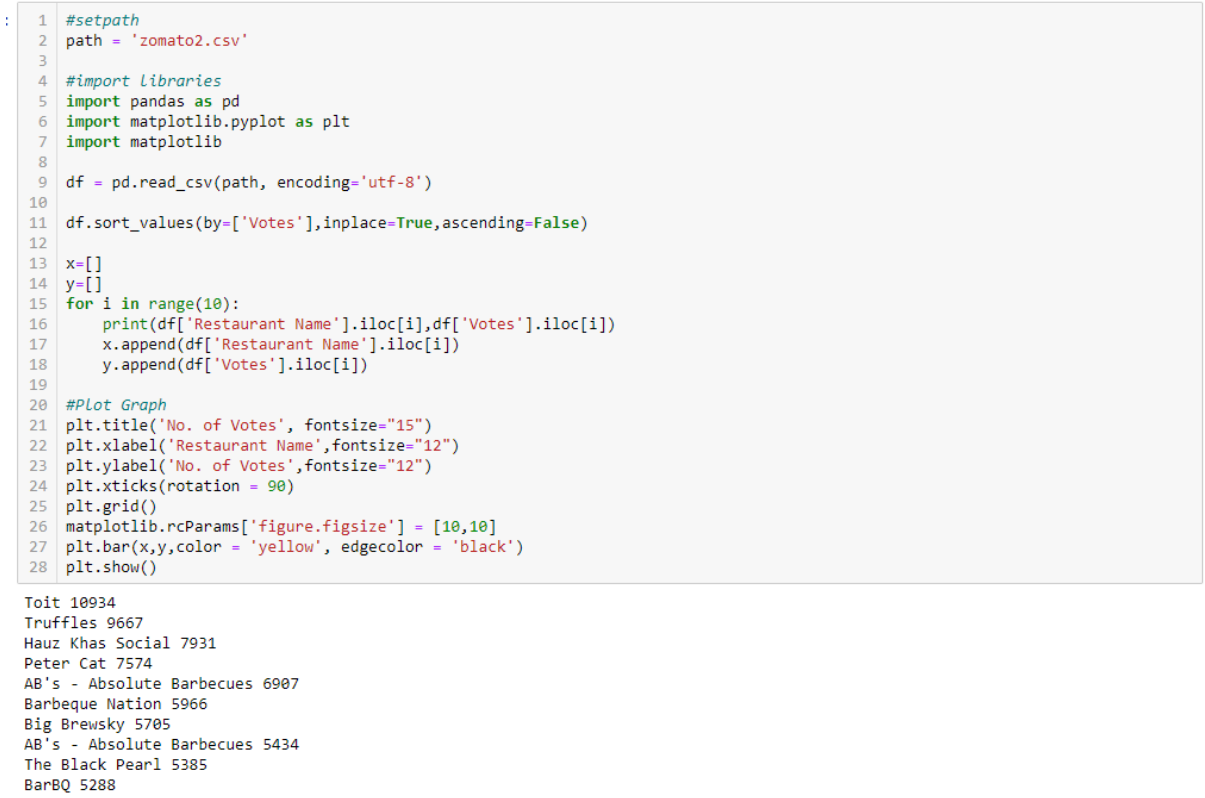


Justification:

Plot is easily implemented on the output using to_list() funcotionaliy of Pandas on Aggregate Ratings column. Histogram is aggregated around 3.5.

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

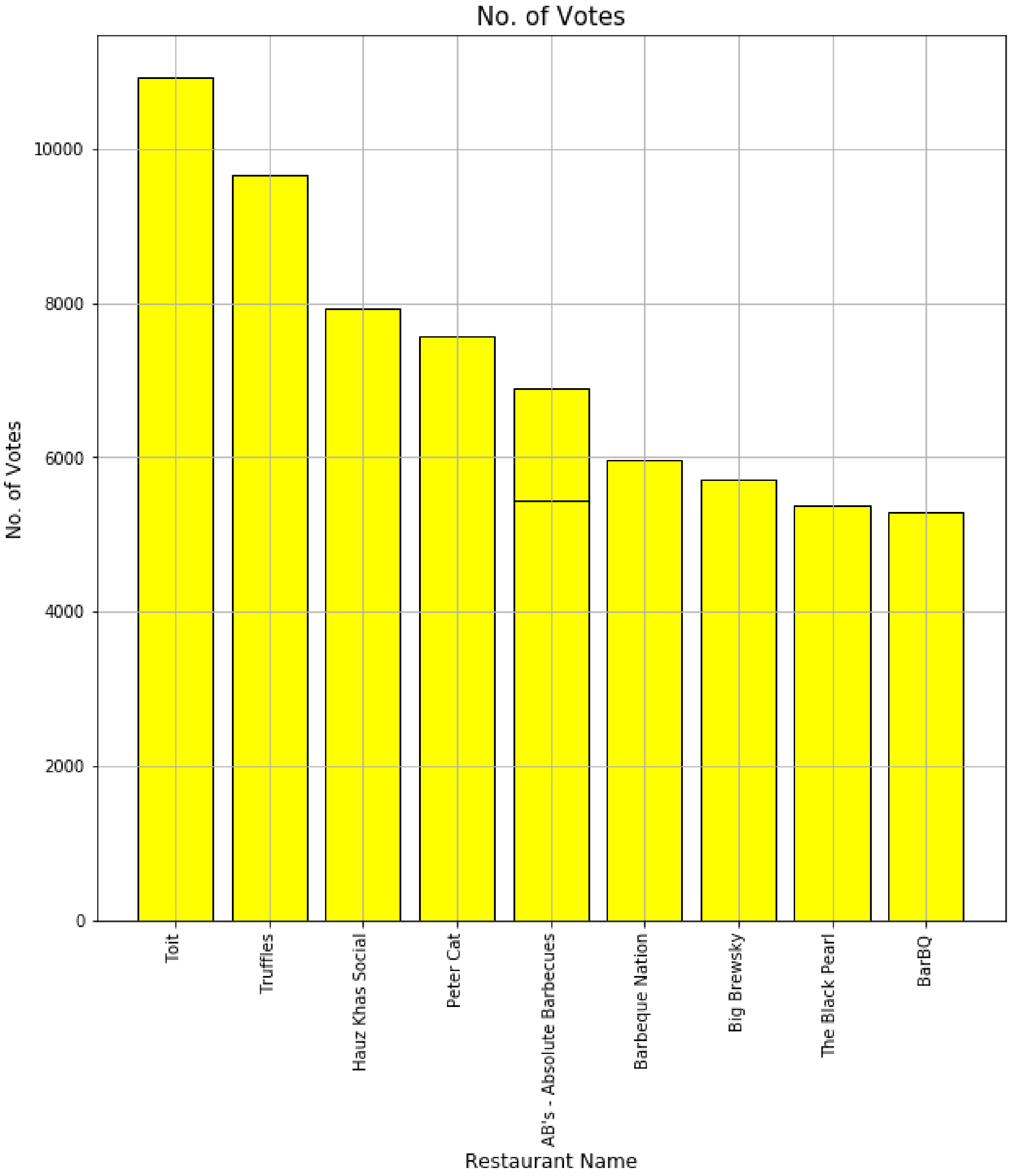
Code:



```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path, encoding='utf-8')

df.sort_values(by=['Votes'],inplace=True,ascending=False)

x=[]
y=[]
for i in range(10):
    print(df['Restaurant Name'].iloc[i],df['Votes'].iloc[i])
    x.append(df['Restaurant Name'].iloc[i])
    y.append(df['Votes'].iloc[i])

#Plot Graph
plt.title('No. of Votes', fontsize="15")
plt.xlabel('Restaurant Name',fontsize="12")
plt.ylabel('No. of Votes',fontsize="12")
plt.xticks(rotation = 90)
plt.grid()
matplotlib.rcParams['figure.figsize'] = [10,10]
plt.bar(x,y,color = 'yellow', edgecolor = 'black')
plt.show()
```

```
Toit 10934
Truffles 9667
Hauz Khas Social 7931
Peter Cat 7574
AB's - Absolute Barbecues 6907
Barbeque Nation 5966
Big Brewsky 5705
AB's - Absolute Barbecues 5434
The Black Pearl 5385
BarBQ 5288
```

Plot:

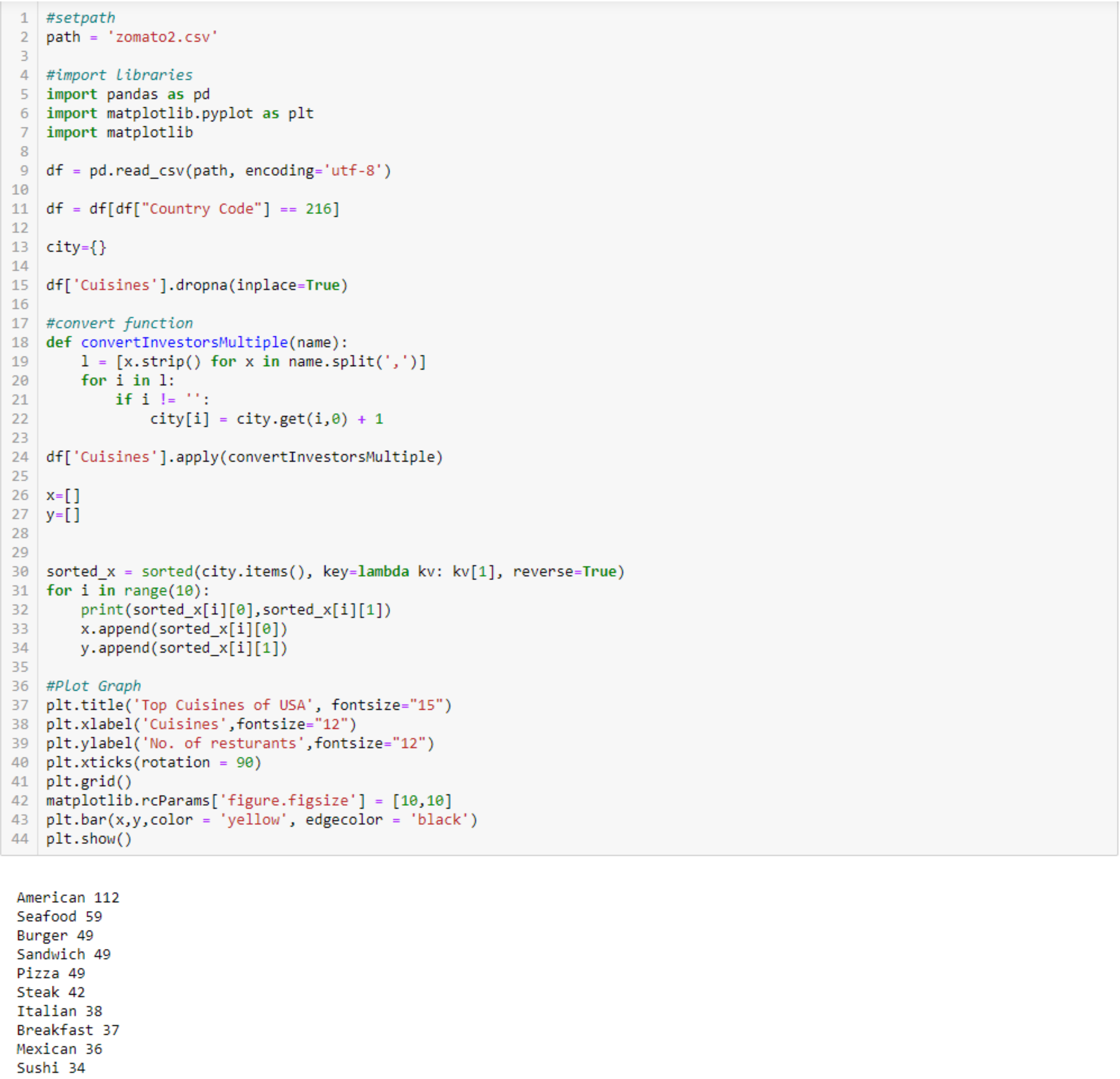


Justification:

Plot is easily implemented on the output on column ‘Votes’ on basis of maximum no. of votes to find top 10 restaraunts.

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

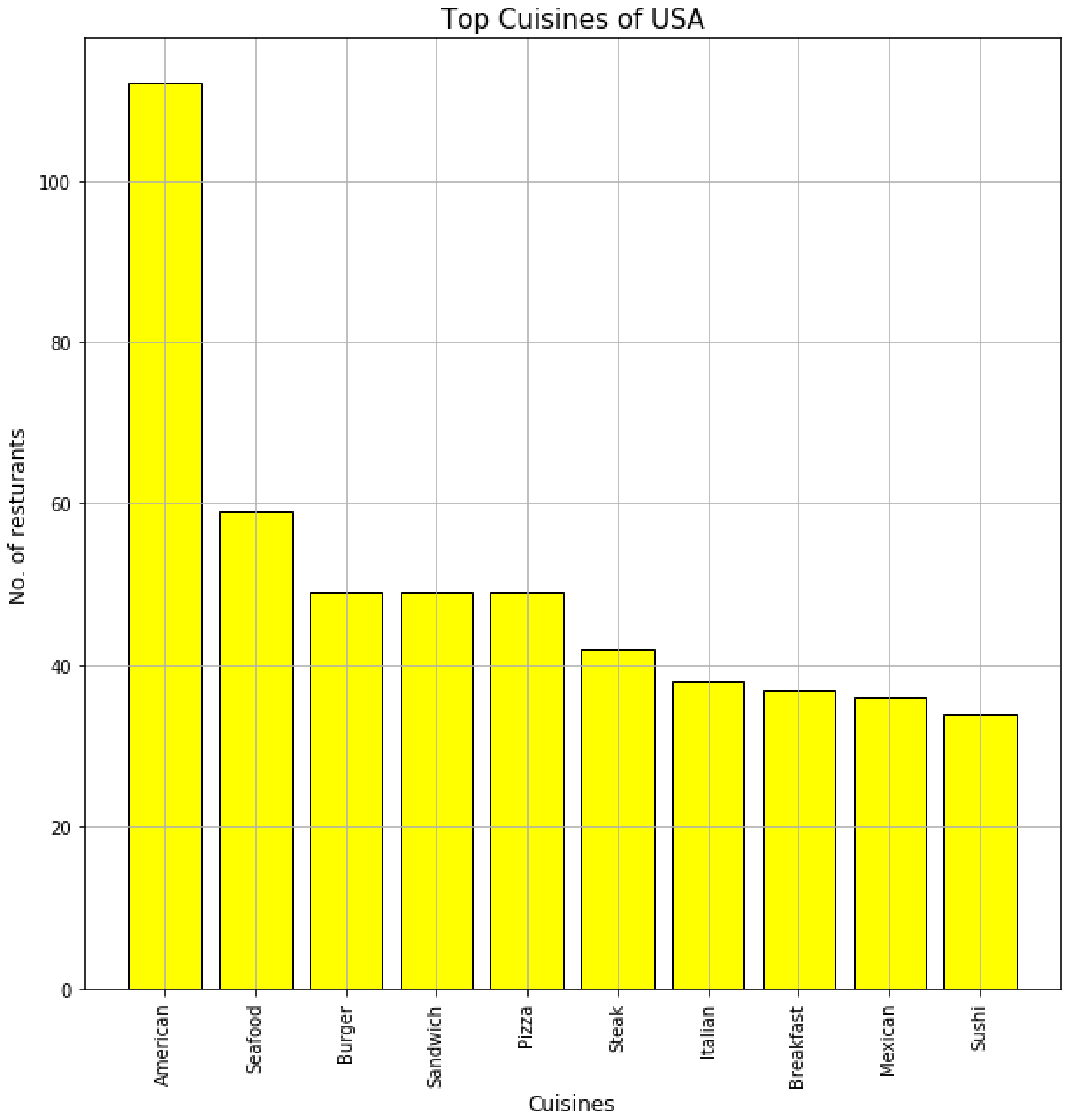
Code:



```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path, encoding='utf-8')

df = df[df["Country Code"] == 216]

city={}

df['Cuisines'].dropna(inplace=True)

#convert function
def convertInvestorsMultiple(name):
    l = [x.strip() for x in name.split(',')]
    for i in l:
        if i != '':
            city[i] = city.get(i,0) + 1

df['Cuisines'].apply(convertInvestorsMultiple)

x=[]
y=[]


sorted_x = sorted(city.items(), key=lambda kv: kv[1], reverse=True)
for i in range(10):
    print(sorted_x[i][0],sorted_x[i][1])
    x.append(sorted_x[i][0])
    y.append(sorted_x[i][1])

#Plot Graph
plt.title('Top Cuisines of USA', fontsize="15")
plt.xlabel('Cuisines',fontsize="12")
plt.ylabel('No. of resturants',fontsize="12")
plt.xticks(rotation = 90)
plt.grid()
matplotlib.rcParams['figure.figsize'] = [10,10]
plt.bar(x,y,color = 'yellow', edgecolor = 'black')
plt.show()
```

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

Plot:

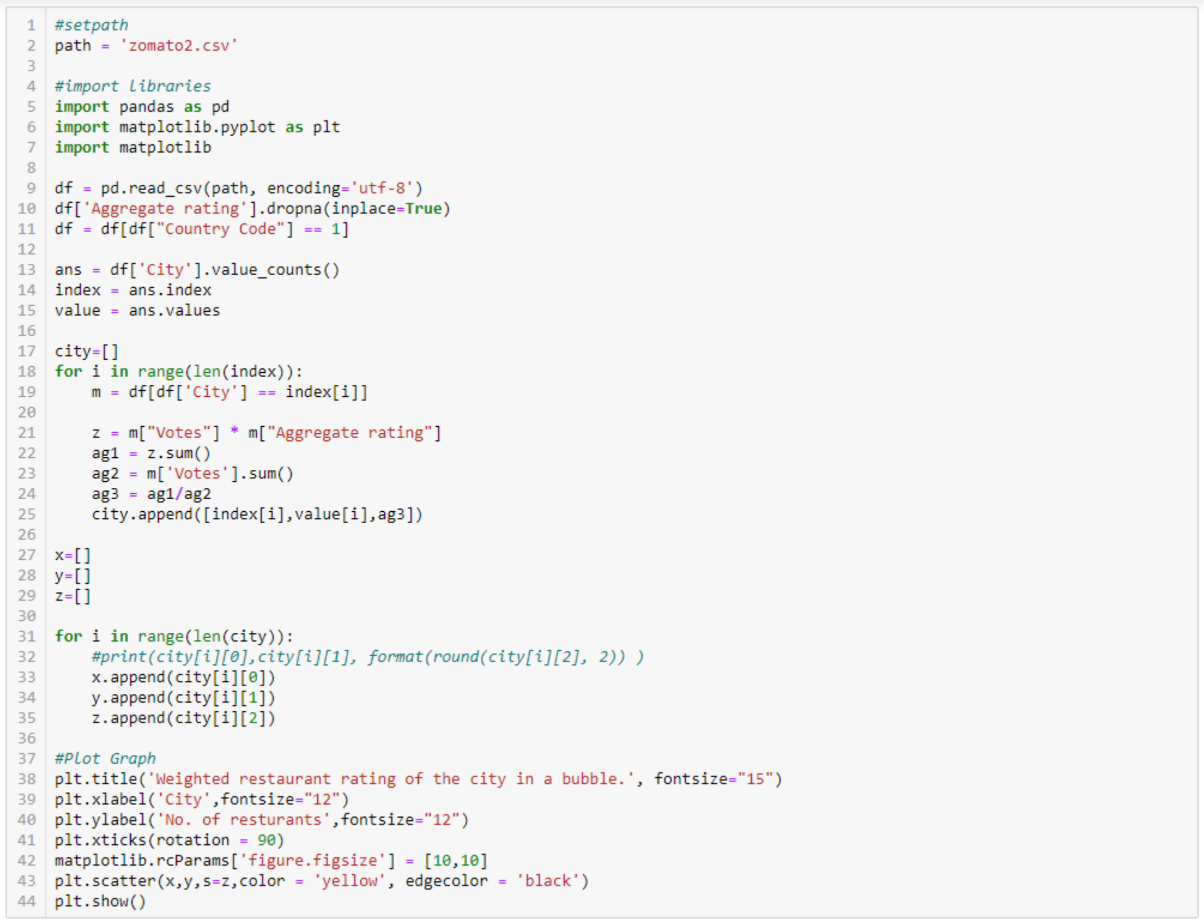


Justification:

Plot is easily implemented on the output on column ‘Cuisines’ to find No. of Restaurants per cuisines.

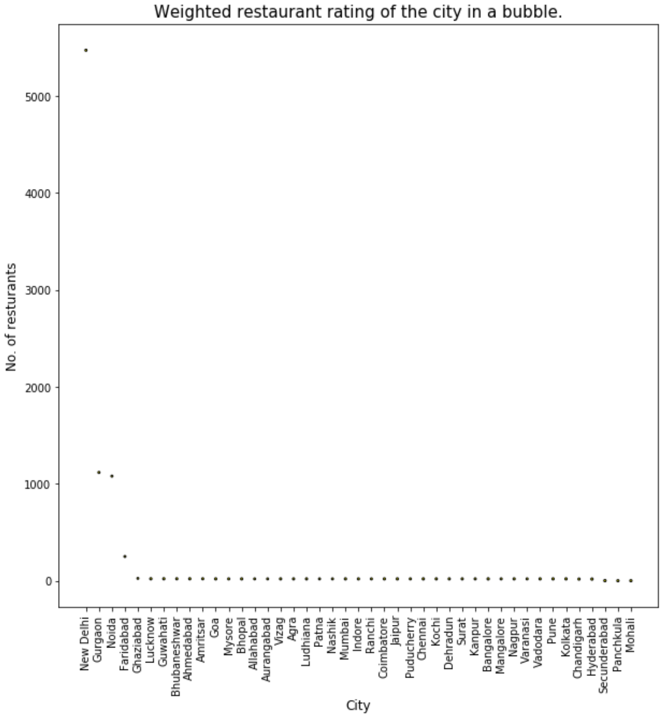
II e) Plot the bubble graph of a number of Restaurants present in the city of India and keeping the weighted restaurant rating of the city in a bubble.

Code:

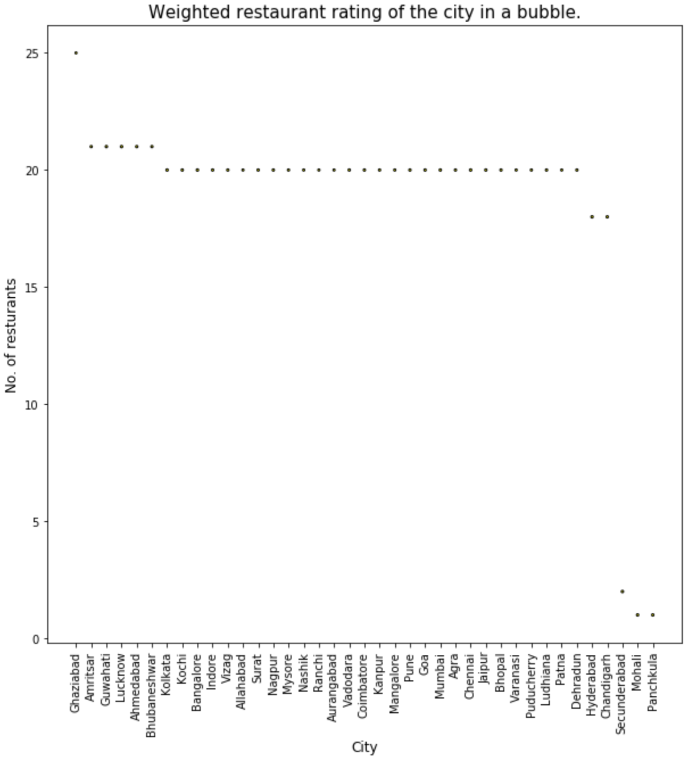


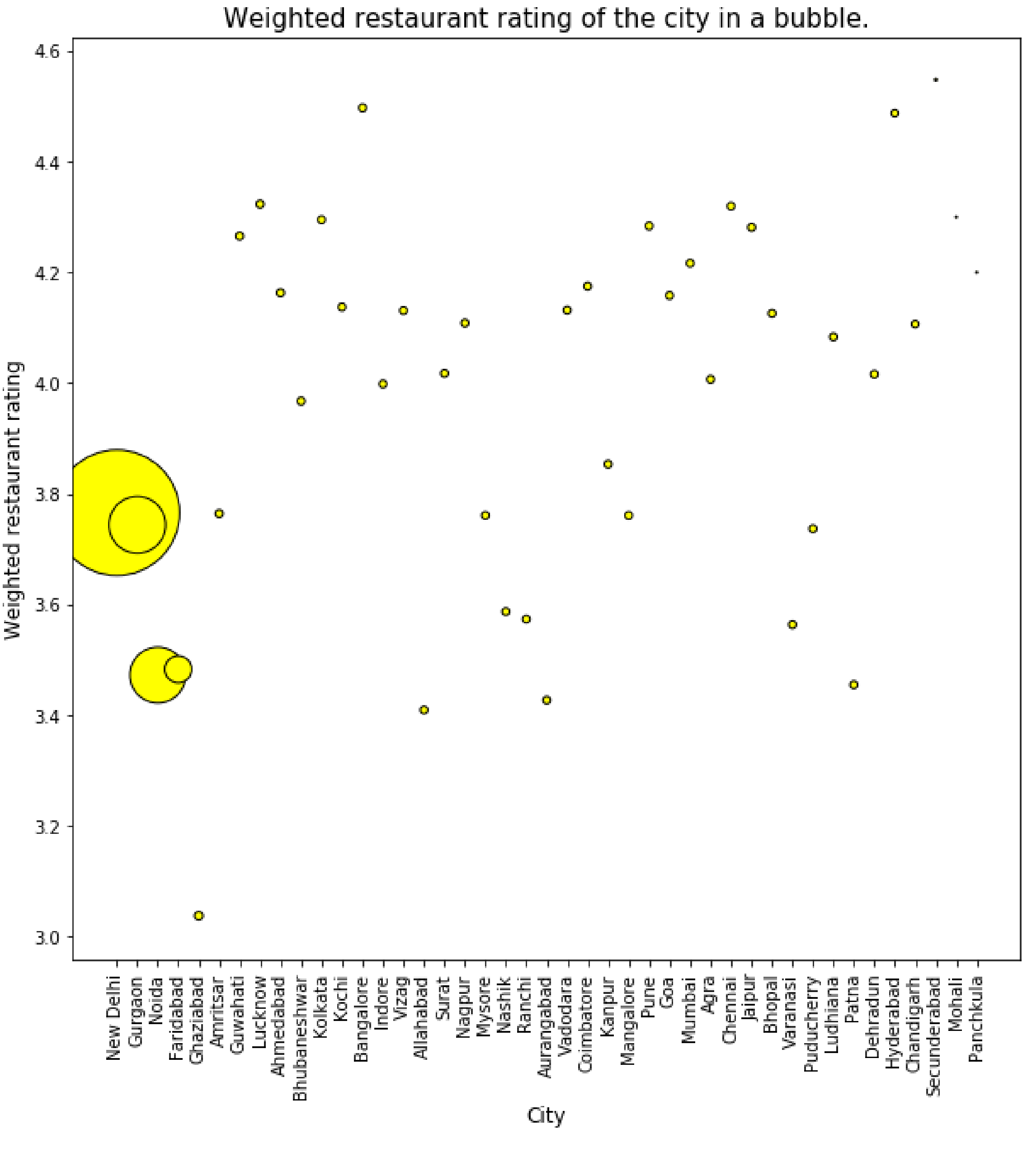
```
#setpath
path = 'zomato2.csv'

#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib

df = pd.read_csv(path, encoding='utf-8')
df['Aggregate rating'].dropna(inplace=True)
df = df[df["Country Code"] == 1]

ans = df['City'].value_counts()
index = ans.index
value = ans.values

city=[]
for i in range(len(index)):
    m = df[df['City'] == index[i]]

    z = m["Votes"] * m["Aggregate rating"]
    ag1 = z.sum()
    ag2 = m['Votes'].sum()
    ag3 = ag1/ag2
    city.append([index[i],value[i],ag3])

x=[]
y=[]
z=[]

for i in range(len(city)):
    #print(city[i][0],city[i][1], format(round(city[i][2], 2)) )
    x.append(city[i][0])
    y.append(city[i][1])
    z.append(city[i][2])

#Plot Graph
plt.title('Weighted restaurant rating of the city in a bubble.', fontsize="15")
plt.xlabel('City',fontsize="12")
plt.ylabel('No. of resturants',fontsize="12")
plt.xticks(rotation = 90)
matplotlib.rcParams['figure.figsize'] = [10,10]
plt.scatter(x,y,s=z,color = 'yellow', edgecolor = 'black')
plt.show()
```

Plot:



No of outlets < 50:




Weighted restaurant rating of the city in a bubble.
Weighted restaurant rating
City
4.6
4.4
4.2
4.0
3.8
3.6
3.4
3.2
3.0
New Delhi
Gurgaon
Noida
Faridabad
Ghaziabad
Amritsar
Guwahati
Lucknow
Ahmedabad
Bhubaneshwar
Kolkata
Kochi
Bangalore
Indore
Vizag
Allahabad
Surat
Nagpur
Mysore
Nashik
Ranchi
Aurangabad
Vadodara
Coimbatore
Kanpur
Mangalore
Pune
Goa
Mumbai
Agra
Chennai
Jaipur
Bhopal
Varanasi
Puducherry
Ludhiana
Patna
Dehradun
Hyderabad
Chandigarh
Secunderabad
Mohali
Panchkula

Justification:

A simple function of pandas value_counts() give us expected results on column City. Plot is easily implemented on the Weighted Restaurant Rating to find maximum outlets.

Weighted Restaurant Rating can be found using:

With Pandas multiplication is simple due to Vectorization.

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

In bubble Graph if weighted rating is in bubble. Graph can’t give any inshight. But with Weighted Rating on y axis give highly insightful graph.