

## Zomato API-2

**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.

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

Code:

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

data = pd.read_csv("zomato.csv", encoding='ISO-8859-1')

# Creating Frames
df1 = data.copy()

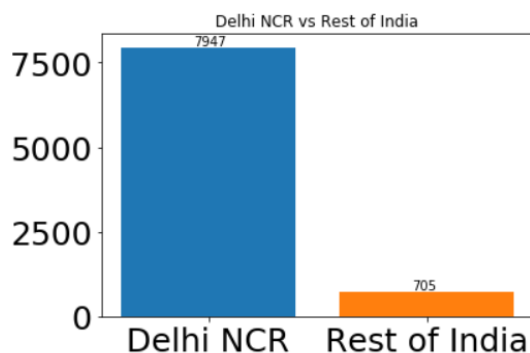
# Filtering the data
df1 = df1[df1["Country Code"]==1]

#delhi-ncr
ncr_count = df1.City[(df1.City == "Ghaziabad") | (df1.City == "Noida") | (df1.City == "Gurgaon") | (df1.City == "Faridabad") | (df1.City == "New Delhi")].count()

#rest of india
rest_count = df1.City.count() - ncr_count

# Plotting Graph
plt.title("Delhi NCR vs Rest of India")
plt.bar("Delhi NCR",ncr_count)
plt.bar("Rest of India",rest_count)
plt.text(-0.07,8000,ncr_count)
plt.text(0.93,800,rest_count)
plt.show()
```

Output:



### Explanation:

Here in this question firstly I created a copy of database and sorted data on the basis of city name and rest of country. And then applied count to count number of restaurants in delhi-ncr and rest of country. Then plotted graph for the above data and found out that:

**No. of restaurants in delhi—ncr: 7947**

**No. of restaurants in rest of india:705**

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

### With DATASET:

#### Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import requests
data = pd.read_csv("zomato.csv", encoding='ISO-8859-1')
# Creating Frames
df1 = data.copy()
df1 = df1[df1["Country Code"]==1]

# FOR NCR
ncr=df1[(df1.City == "Ghaziabad") | (df1.City == "Noida") | (df1.City ==
"Gurgaon") | (df1.City == "Faridabad") | (df1.City == "New Delhi")].Cuisines
```

#### **#sorting and splitting data**

```
ncr_cuisine=[]
for ele in ncr:
    ele1 = ele.split(',')
    for i in ele1:
        i=i.strip()
        ncr_cuisine.append(i)
ncr_cuisine_df = pd.DataFrame(ncr_cuisine)
unique_cuis_ncr = ncr_cuisine_df[0].unique()
```

#### **# FOR REST OF INDIA**

```
rest_india=df1[(df1.City != "Ghaziabad") & (df1.City != "Noida") & (df1.City !=
"Gurgaon") & (df1.City != "Faridabad") & (df1.City != "New Delhi")].Cuisines
rest_india_cuisine=[]
for ele in rest_india:
    ele1 = ele.split(',')
    for i in ele1:
        i=i.strip()
```

```

rest_india_cuisine.append(i)
rest_india_cuisine_df = pd.DataFrame(rest_india_cuisine)
unique_cuis_rest_india = rest_india_cuisine_df[0].unique()
#appending and printing unique cuisines
csv_cuisines = []
for i in unique_cuis_rest_india:
    if i not in unique_cuis_ncr:
        csv_cuisines.append(i)
print("Cuisines which are not present in restaurant of Delhi NCR but present
in rest of India according to the given CSV File is ")
for ele in csv_cuisines:
    print(ele)

```

#### Output:

```

Cuisines which are not present in restaurant of Delhi NCR but present in rest of India according to the given CSV File is
German
Malwani
BBQ
Cajun

```

---

#### Explanation:

Here in this question first I found out unique cuisines in delhi and rest of india then checked if the cuisine is present in rest of india and is not present in delhi-ncr then appended it into an array and printed it. And after printing the names we found out:

**German, Malwani, BBQ and Cajun are unique cuisines which are present in rest of india but not in delhi-ncr.**

#### With API:

```

user_key = "399720f6f904f106e162cd2bd0011a6f"
data=requests.get('https://developers.zomato.com/api/v2.1/cuisines',header
s={'user-key':user_key},params={'city_id':1})
ans=data.json()['cuisines']
api_cuisine=[]
for i in ans:
    api_cuisine.append(i['cuisine']['cuisine_name'])
for ele in csv_cuisines:
    if ele in api_cuisine:
        print(ele)

```

#### Output:

```

Cuisines which are present in restaurant of Delhi NCR according to API:
Malwani
BBQ

```

#### Explanation:

Here using api we got to know that data with csv was wrong and we have BBQ and Malwani were the cuisines that were present in delhi. And I did it by fetching data of cuisines using Zomato api.

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

Code:

**#used the above list created and changed it into dataframe and applied value counts to find top cuisine**

**# Delhi-NCR**

```
print("Top 10 cuisines served by maximum number of restaurants in Delhi NCR are: ")
```

```
ncr_cuisine_df = pd.DataFrame(ncr_cuisine)
```

```
ncr_top_10 = ncr_cuisine_df[0].value_counts()[:10]
```

```
print(ncr_top_10)
```

```
print("\n")
```

**# Rest Of India**

```
print("Top 10 cuisines served by maximum number of restaurants in Rest of India are: ")
```

```
rest_india_cuisine_df = pd.DataFrame(rest_india_cuisine)
```

```
rest_india_top_10 = rest_india_cuisine_df[0].value_counts()[:10]
```

```
print(rest_india_top_10)
```

Output:

```
Top 10 cuisines served by maximum 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
Name: 0, dtype: int64
```

```
Top 10 cuisines served by maximum number of restaurants in 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
Name: 0, dtype: int64
```

Explanation:

Here in this question used the list created in above question and then converted it into database and applied value counts to find top cuisines.. And after printing the names we found out the appropriate result.

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

Code:

***#used the above list created and changed it into dataframe***

***#delhi-ncr***

```
ncr_cuisine_df = pd.DataFrame(ncr_cuisine)
```

```
ncr_top_10 = ncr_cuisine_df[0].value_counts()[:10]
```

***#rest of india***

```
rest_india_cuisine_df = pd.DataFrame(rest_india_cuisine)
```

```
rest_india_top_10 = rest_india_cuisine_df[0].value_counts()[:10]
```

***#plotting the graph for delhi-ncr***

```
plt.pie(ncr_top_10.values,labels=ncr_top_10.index,textprops={'fontsize': 10})
```

```
plt.title("Cuisines In Delhi-NCR")
```

```
plt.show()
```

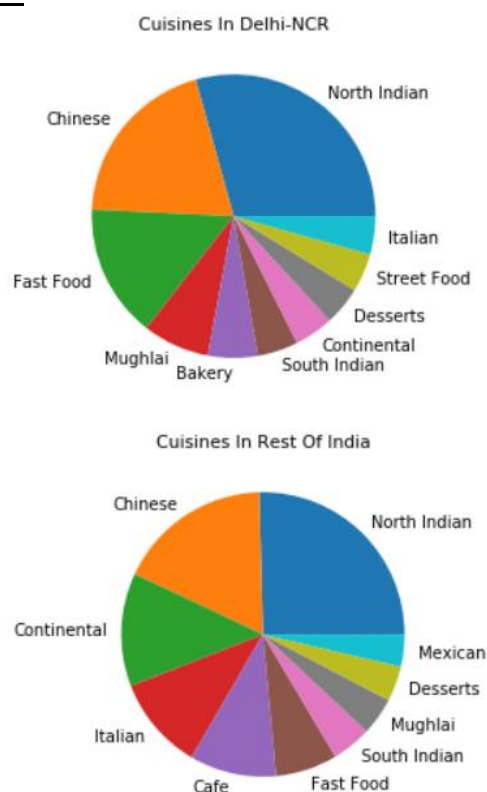
***#plotting the graph for rest of india***

```
plt.pie(rest_india_top_10.values,labels=rest_india_top_10.index,textprops={'f  
ontsize': 10})
```

```
plt.title("Cuisines In Rest Of India")
```

```
plt.show()
```

Output:



Explanation:

In this question I plotted graphs for cuisine I printed in the previous question i.e. the top 10 cuisines served in delhi-ncr and rest of india.

Analysis:

Let's observe the above given two pie graphs. They show the following points:

- ➔ North Indian and Chinese are common top two cuisines that are served in both delhi-ncr and rest of india.
- ➔ But in delhi-ncr fast food is more common and continental at third's place.
- ➔ Italian is more famous in rest of india than delhi-ncr.
- ➔ Desserts, South Indian has same popularity.

This is done with analysis.

## Ques 2:

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

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.

2.1.1 Number of Votes given Restaurant

Code:

***#filtered rating where rating is not zero***

```
df_rating = df1[df1["Aggregate rating"]!=0]
```

```
plt.figure(figsize=(20,10))
```

***#plotting graph***

```
plt.scatter(df_rating["Votes"],df_rating["Aggregate rating"],marker=".")
```

```
plt.title("Number of Votes vs Aggregate Rating")
```

```
plt.xlabel("Number of Votes")
```

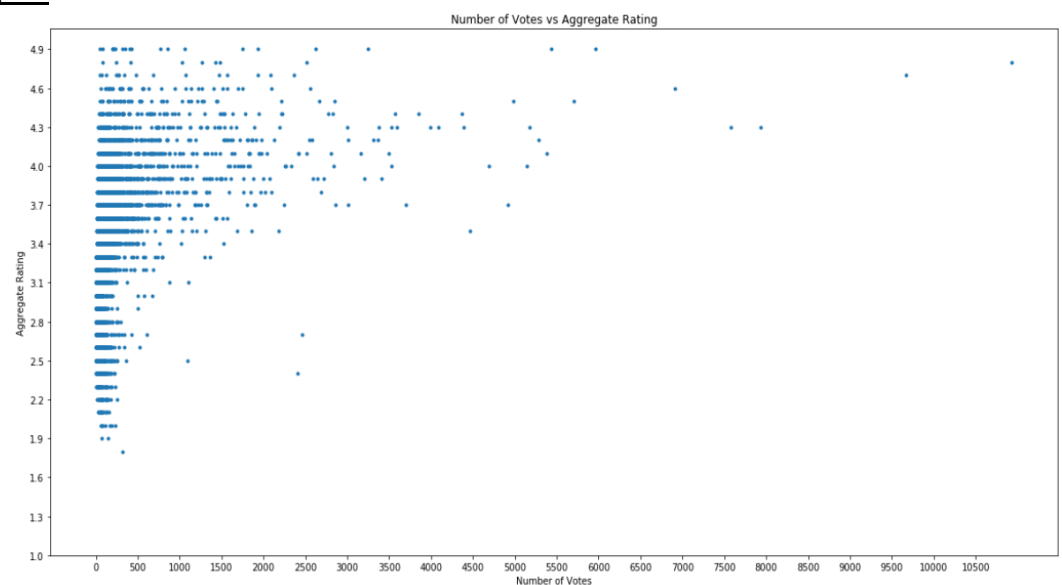
```
plt.ylabel("Aggregate Rating")
```

```
plt.xticks(np.arange(0,10800,step=500))
```

```
plt.yticks(np.arange(1,5,step=0.3))
```

```
plt.show()
```

Output:



Explanation:

In this question I filtered data on the basis of rating where rating is not zero and extracted number of votes and plotted a scatter graph between them.

### Analysis:

Let's observe the above given two pie graphs. They show the following points:

- > In the graph we can see as number of votes is increased the aggregate rating is also increased. Range: 3 to 4.2
- > As the rating is reached to 4.9 the number of votes is decreased.
- > For very lower rating in range 1.3 to 2.2 the number of votes are less.

This is done with analysis.

### 2.1.2 Restaurant serving more number of cuisines.

#### Code:

***# here I filtered the data on the basis of cuisines and found the number of cuisines served by every restaurant***

```
no_cuis=[]
```

```
for ele in df1.Cuisines:
```

```
    no_cuis.append(len(ele.strip().split(",")))
```

***# Plotting the graph***

```
plt.scatter(df1["Aggregate rating"],no_cuis, marker=".")
```

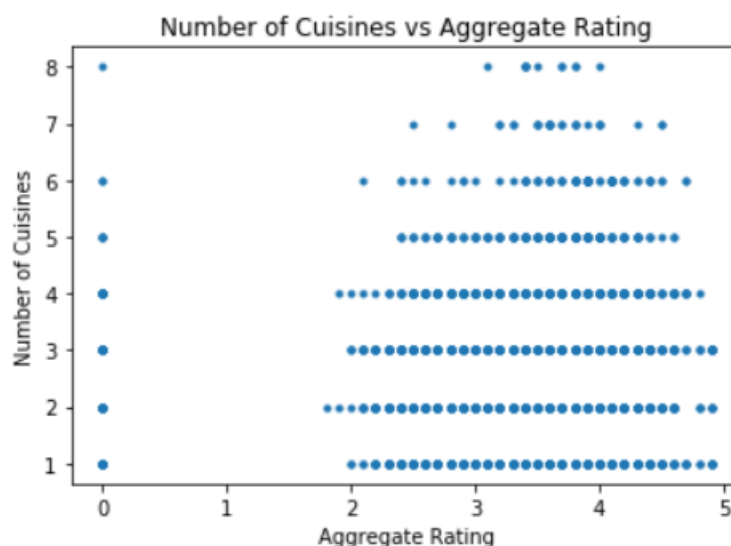
```
plt.title("Number of Cuisines vs Aggregate Rating")
```

```
plt.ylabel("Number of Cuisines")
```

```
plt.xlabel("Aggregate Rating")
```

```
plt.show()
```

#### Output:



#### Explanation:

In this question I filtered data on the basis of cuisines and found number of cuisines per restaurant and plotted a scatter graph between cuisines and aggregate rating.

#### Analysis:

Let's observe the above given two pie graphs. They show the following points:

- > The number of cuisines ranging from 1 to 5 the rating ranges from 2 to 5.

- > As the number of cuisines reaches 8 the rating is less for those restaurants.
- > The restaurants having number of cuisine is four the number of rating is maximum.

This is done with analysis.

### 2.1.3 Average Cost of Restaurant

Code:

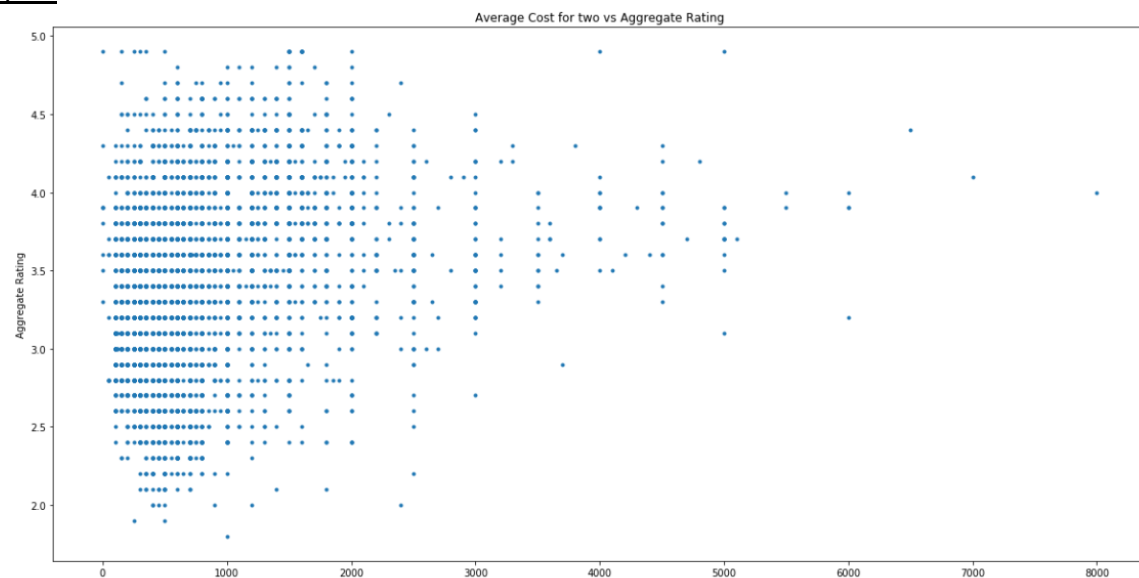
***#here I filtered data on the basis of rating and cost for two and plotted graph for it***

```
df_rating = df1[df1["Aggregate rating"]!=0]
plt.figure(figsize=(20,10))
plt.scatter(df_rating["Average Cost for two"],df_rating["Aggregate
rating"],marker=".")
```

***#plotting the graph***

```
plt.title("Average Cost for two vs Aggregate Rating")
plt.xlabel("Average Cost for two")
plt.ylabel("Aggregate Rating")
plt.show()
```

Output:



Explanation:

In this question I filtered data on the basis of cuisines and average cost for two and plotted a scatter graph between them.

Analysis:

Let's observe the above given two pie graphs. They show the following points:

- > If range of average cost for two lies between 500 to 2000 then rating of those restaurants is good enough.
- > As the average cost of two is increased then number of rating for the restaurants is less.



#### 2.1.4 Restaurant serving some specific cuisines.

Code:

***#here I filtered data on the basis of rating***

```
rating = list(df1['Aggregate rating'])
```

```
cui = list(df1['Cuisines'])
```

```
dict1={}
```

***#appended all the rating given to a particular with their respective cuisine name***

```
for i in range(len(cui)):
```

```
    cl = cui[i].split(",")
```

```
    for c in cl:
```

```
        c=c.strip()
```

```
        if c in dict1:
```

```
            if rating[i]!=0.0:
```

```
                temp_l=dict1[c]
```

```
                temp_l.append(rating[i])
```

```
                dict1[c] = temp_l
```

```
        else:
```

```
            dict1[c] = [rating[i]]
```

***# finding average of all the rating***

```
for k,v in dict1.items():
```

```
    dict1[k] = sum(v)/len(v)
```

***# Taking names and values***

```
cuisine = np.array(list(dict1.keys()))
```

```
rate = np.array(list(dict1.values()))
```

```
rate = np.round(rate, decimals=2)
```

***#TOP 10 Cuisines***

```
ind = rate.argsort()[::-1]
```

```
ind = ind[:10]
```

***# Taking names and values of top 10 for graph plot***

```
cuisine_t10 = cuisine[ind]
```

```
rate_t10 = rate[ind]
```

***# Giving Labels***

```
plt.ylabel("Aggregate Rating")
```

```
plt.xticks(rotation=40)
```

```
plt.title('Aggregate Rating of some Specific Cuisines')
```

```
# Plotting bars with for loop
```

```
for i in range(len(cuisine_t10)):
```

```
    plt.bar(cuisine_t10[i],rate_t10[i],edgecolor='black')
```

***# Giving graph text***

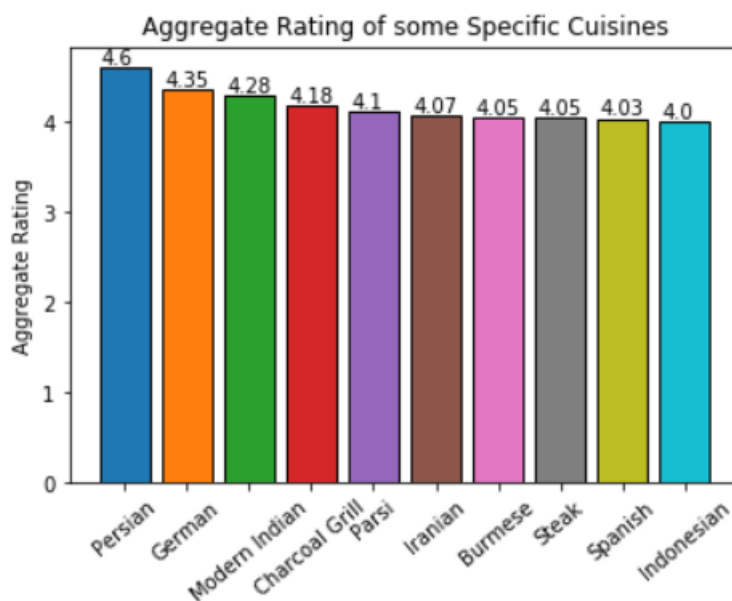
```
plt.text(-0.36,rate_t10[0]+0.04,rate_t10[0])
```

```

plt.text(0.64,rate_t10[1]+0.04,rate_t10[1])
plt.text(1.64,rate_t10[2]+0.04,rate_t10[2])
plt.text(2.64,rate_t10[3]+0.04,rate_t10[3])
plt.text(3.64,rate_t10[4]+0.04,rate_t10[4])
plt.text(4.64,rate_t10[5]+0.04,rate_t10[5])
plt.text(5.64,rate_t10[6]+0.04,rate_t10[6])
plt.text(6.64,rate_t10[7]+0.04,rate_t10[7])
plt.text(7.64,rate_t10[8]+0.04,rate_t10[8])
plt.text(8.64,rate_t10[9]+0.04,rate_t10[9])
plt.show()

```

Output:



Explanation:

In this question I filtered data on the basis of rating and cuisines and then made a dictionary to append all the ratings of their particular cuisine and then found out average of all ratings and plotted graph between them.

Analysis:

Let's observe the above given two pie graphs. They show the following points:

- > Persian, German are the cuisines that gets good rating when served but it is served in only one or two restaurants.
- > Modern Indian is the cuisine that also gets good rating when served in restaurants and it is served in many restaurants and liked by most of the people.
- > Charcoal Grill, Parsi, Iranian, Burmese, Steak, Spanish and Indonesian are also some cuisine that are being served and getting good ratings.

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

2.2.1 Weighted Restaurant Rating =  $\Sigma (\text{number of votes} * \text{rating}) / \Sigma (\text{number of votes})$  .

Code:

```
import numpy as np
locality=df1.Locality.unique()
# filtered data and created new column in database by multiplying number of votes and aggregate rating
df1['number_rating']=df1['Aggregate rating']*df1['Votes']
# then for every element grouped votes and number of rating with locality and found sum for votes and number of rating
for ele in locality:
    sigma2 = df1.groupby(["Locality"])["Votes"].sum()
    sigma1 = df1.groupby(["Locality"])["number_rating"].sum()
# created locality variable and made it equal to locality of sigma1
locality = sigma1.index
sigma1=np.array(sigma1.values)
sigma2=np.array(sigma2.values)
# Created dictionary for finding weighted rating
d={}
for i in range(len(locality)):
    if sigma2[i]==0:
        ans=sigma1[i]/1
    else:
        ans=sigma1[i]/sigma2[i]
    d[locality[i]] = ans
# Sorted array and printed top 10 with locality and weighted rating
arr = sorted(d.items(), key = lambda d:(d[1], d[0]) , reverse = True)[:10]
for ele in arr:
    print(ele[0], "%.5f"%ele[1])
```

Output:

```
Hotel Clarks Amer, Malviya Nagar 4.90000
Aminabad 4.90000
Friends Colony 4.88692
Powai 4.84187
Kirlampudi Layout 4.82016
Express Avenue Mall, Royapettah 4.80000
Deccan Gymkhana 4.80000
Banjara Hills 4.71876
Sector 5, Salt Lake 4.70702
Riverside Mall, Gomti Nagar 4.70000
```

Explanation:

In this question I filtered data and added column by finding aggregate rating \* votes and then grouped new column and votes with locality and created a dictionary to find weighted rating and then sorted the values and printed them .

### Ques 1: Visualization

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

Code:

***#filtered data on restaurant name and applied value counts and printed graph***

```
restaurant_outlets=df1['Restaurant Name'].value_counts()[:15]
```

```
names=restaurant_outlets.index
```

```
values=restaurant_outlets.values
```

***# Plotting The Graph***

```
plt.xlabel("No. Of Outlets")
```

```
plt.title("Maximum Number Of Outlets")
```

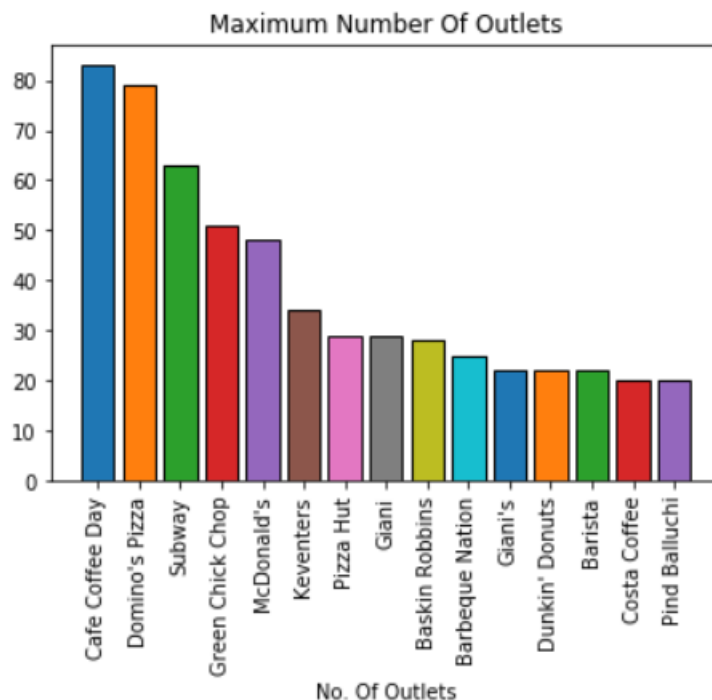
```
plt.xticks(rotation=90)
```

```
for i in range(len(values)):
```

```
plt.bar(names[i],values[i],edgecolor='black')
```

```
plt.show()
```

Output:



Explanation:

Here in this question firstly I created a copy of database and sorted data on the basis of restaurant name and applied value counts on it . Then plotted graph for the above data and found out that:

**Café Coffee Day: 85(approx.)**

**Domino's Pizza: 79 (approx.)**

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

Code:

***#filtered data on the basis of aggregate rating***

```
df_rating = df1[df1["Aggregate rating"]!=0]
```

```
data = df_rating["Aggregate rating"].values
```

***#plotting the graph***

```
plt.hist(data,bins = 30 , edgecolor="#000000")
```

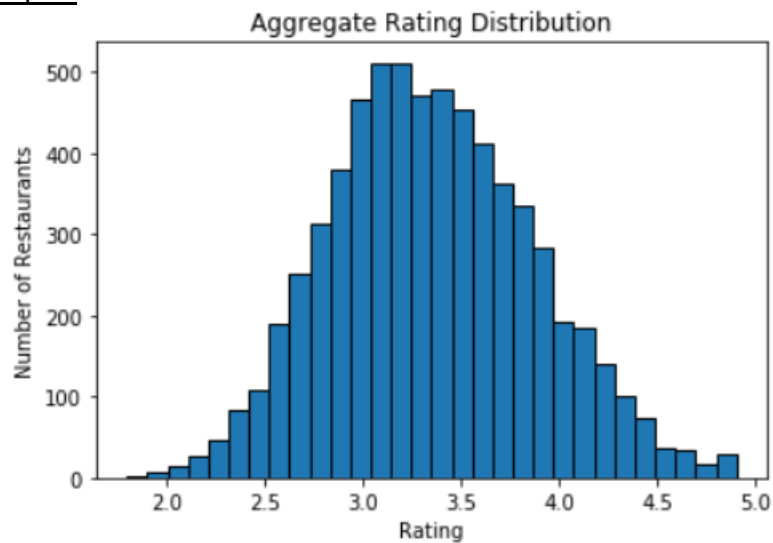
```
plt.ylabel("Number of Restaurants")
```

```
plt.xlabel("Rating")
```

```
plt.title("Aggregate Rating Distribution")
```

```
plt.show()
```

Output:



Explanation:

Here in this question firstly I created a copy of database and sorted data on the basis of aggregate rating. Then plotted graph for the above data.

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

Code:

```
data = pd.read_csv("zomato.csv", encoding ='ISO-8859-1')
```

```
df7 = data.copy()
```

***# Selecting top 10 with highest votes***

```
data_set = df7.groupby(['Restaurant
```

```
Name'])["Votes"].sum().sort_values(ascending=False)[:10]
```

***# Storing names and values***

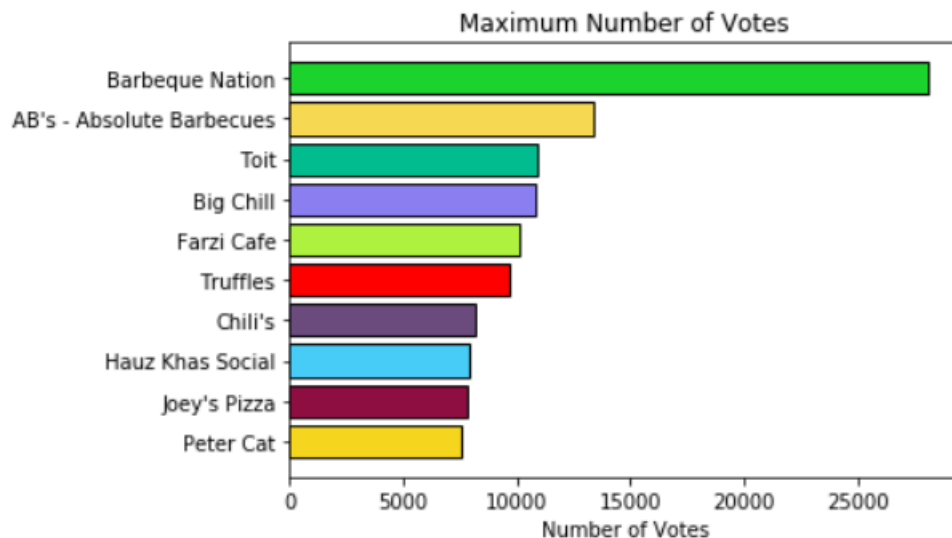
```
names = data_set.index
```

```
votes1 = data_set.values
```

### **# Plotting Graphs**

```
plt.xlabel("Number of Votes")
plt.title('Maximum Number of Votes')
colors=["#1cd32e","#f7d852","#00bc8e","#8a7ef1","#aff13f","r","#6b4b7e","#46ccf7","#8f0e42","#f6d51e"]
for i in range(9,-1,-1):
    plt.barh(names[i],votes1[i],color=colors[i],edgecolor='#000000')
```

Output:



Explanation:

Here in this question firstly I created a copy of database and sorted data on basis of restaurant name and votes and grouped votes by restaurants and found sum and sorted data. Then plotted graph for the above data.

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

Code:

```
data = pd.read_csv("zomato.csv", encoding = 'ISO-8859-1')
```

### **# Creating Frames**

```
df_USA=data.copy()
```

```
df_USA = df_USA[df_USA["Country Code"]==216]
```

### **#filtered data on the basis of cuisine**

```
cuisine_USA=df_USA.Cuisines
```

```
cuisine=[]
```

```
for ele in cuisine_USA:
```

```
    ele=str(ele)
```

```
    ele1 = ele.split(',')
```

```
    for i in ele1:
```

```
        i=i.strip()
```

```
        cuisine.append(i)
```

```
cuisine=pd.DataFrame(cuisine)
```

```
cuisine=cuisine[0].value_counts()[:10]
```

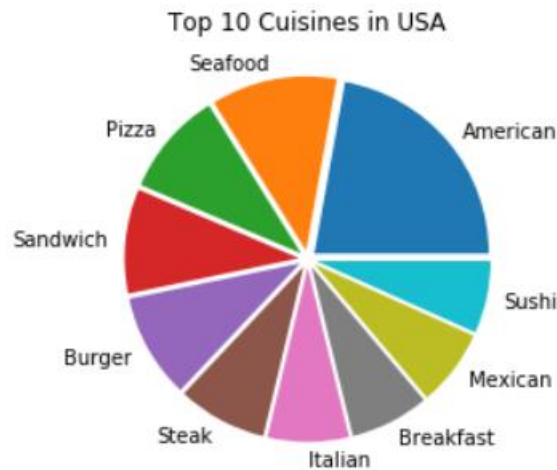
```
values = cuisine.values
```

```

labels = cuisine.index
explode=[0.05]*10
#plotting the graph
plt.pie(values,labels=labels,explode=explode)
plt.title("Top 10 Cuisines in USA")
plt.show()

```

Output:



Explanation:

Here in this question firstly I created a copy of database and sorted data on basis of country code and then found cuisines and splitted data on the basis of ',' and then applied value counts and plotted graph for the above.

**Conclusion:**

**American is the most served cuisine.**

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:

```
import numpy as np
```

**#used the above code for weighted graph to find out weighted graph.  
The difference is this time did it with city.**

```
locality=df1.Locality.unique()
```

```
df1['number_rating']=df1['Aggregate rating']*df1['Votes']
```

for ele in locality:

```
    sigma2 = df1.groupby(["City"])["Votes"].sum()
```

```
    sigma1 = df1.groupby(["City"])["number_rating"].sum()
```

```
locality = sigma1.index
```

```
sigma1=np.array(sigma1.values)
```

```
sigma2=np.array(sigma2.values)
```

```
d={}
```

```
for i in range(len(locality)):
```

```
    if sigma2[i]==0:
```

```
        ans=sigma1[i]/1
```

```
    else:
```

```
        ans=sigma1[i]/sigma2[i]
```

```
    d[locality[i]] = ans
```

**#then created a dataframe and appended city name, count and weighted rating**

```
number=df1['City'].value_counts()
```

```
a=number.index
```

```
df_number=pd.DataFrame(a)
```

```
df_number['number']=number.values
```

**#sorted data alphabetically**

```
df_number=df_number.sort_values(by=0)
```

```
new_dataframe=pd.DataFrame(d.keys())
```

```
new_dataframe['weighted']=d.values()
```

***#plotting the graph by assigning names and values***

```
names=np.array(new_dataframe[0])
```

```
values=np.array(df_number['number'])
```

```
weights=np.array(new_dataframe['weighted'])
```

```
colors=np.arange(len(names))+1
```



```

plt.rc('xtick', labels=25)

plt.rc('ytick', labels=25)

plt.figure(1, figsize=(30, 25))

plt.subplot(211)

plt.xticks(rotation=90)

for i in range(43):

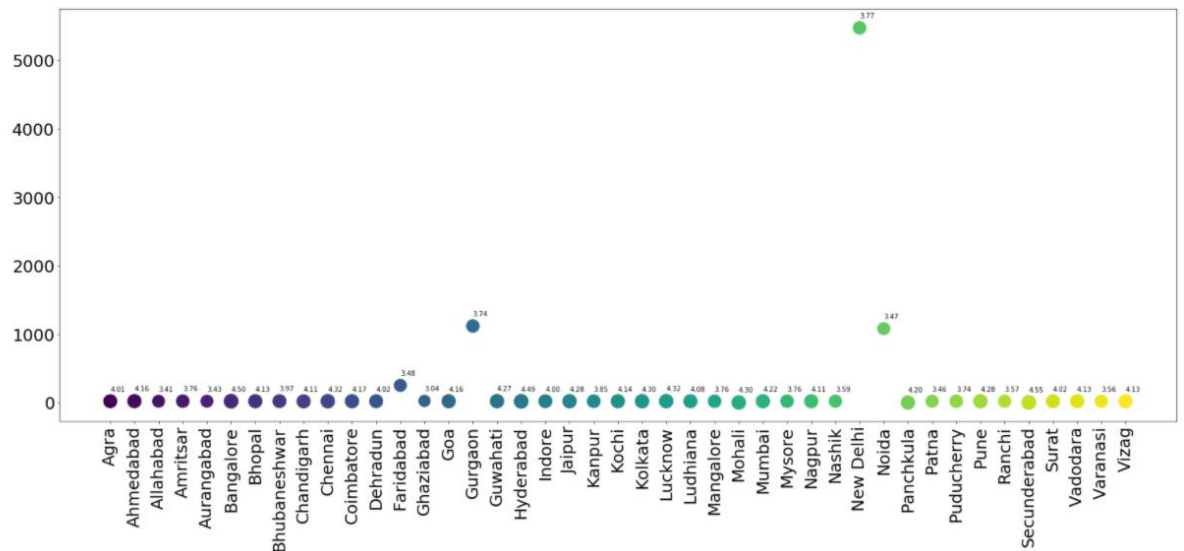
    plt.text(names[i], values[i]+150, "%.2f"%weights[i])

plt.scatter(names, values, s=weights*100, c=colors, alpha=1)

plt.show()

```

### Output:



### Explanation:

Here in this question firstly I created a copy of database and sorted data and used weighted graph and then used cities name and count values and then plotted graph between these three.

