

Zomato Api Project Part 2-

Q1. Plot the bar graph for number of restaurants present in delhi ncr vs rest of india ?

Code-

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import requests
plt.style.use('seaborn')
```

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

df.head()

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking	On deliv
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenue...	Century City Mall, Poblacion, Makati City	Century City Mall, Poblacion, Makati City, Mak...	121.027535	14.565443	French, Japanese, Desserts	...	Botswana Pula(P)	Yes	
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City	Little Tokyo, Legaspi Village, Makati City, Ma...	121.014101	14.553708	Japanese	...	Botswana Pula(P)	Yes	
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...	121.056831	14.581404	Seafood, Asian, Filipino, Indian	...	Botswana Pula(P)	Yes	
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.056475	14.585318	Japanese, Sushi	...	Botswana Pula(P)	No	
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall,	SM Megamall, Ortigas, Mandaluyong	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.057508	14.584450	Japanese, Korean	...	Botswana Pula(P)	Yes	

```
df=df[df['Country Code']==1] # as country code of india is one we will only get indian countries by indexing in this way
```

```
cities=['New Delhi', 'Ghaziabad', 'Noida', 'Gurgaon', 'Faridabad'] # ncr cities
```

```
df['City'].replace(cities,'Delhi-Ncr',inplace=True) #replacing all ncr cities with cities in india
```

```
non_ncr=np.delete(df['City'].unique(),np.where(df['City'].unique()=='Delhi-Ncr')) #filtering out ncr cities to obtain non ncr
```

```
df.replace(non_ncr,'Rest of India',inplace=True) #replacing all of the non_ncr cities with rest of india
```

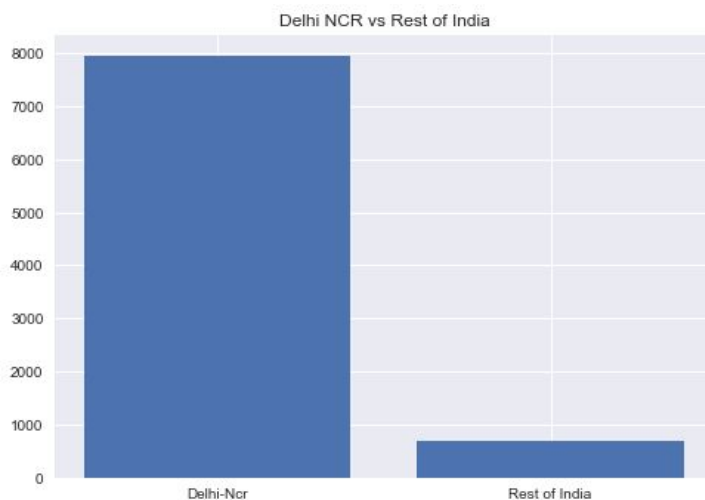
```
(df['City']=='Rest of India').value_counts()
False    7947
True      705
Name: City, dtype: int64
```

Non ncr=705 and ncr=7947

Creating the bar plot --

```
plt.bar(['Delhi-Ncr','Rest of India'],[7947,705])
```

```
plt.title('Delhi NCR vs Rest of India')
```



Justification - it is pretty evident from the plot that there are a fairly large amount of restaurants from delhi ncr in the dataset as compared to the rest of india. So the dataset is highly skewed towards delhi ncr restaurants .

Q2. Find the cuisines which are not present in restaurants of delhi ncr but present in rest of india . Cross check the result with zomato api to ensure if the cuisines are actually not served in delhi ncr or its just due to incomplete dataset ?

Code -

```
d_ncr={}
d_non_ncr={}

```

```
for i in df.values:
    cuisines=i[9].split(',')
    for cuisine in cuisines:
        cuisine=cuisine.strip(' ')
        if i[3]=='Delhi-Ncr':
            if cuisine not in d_ncr:
                d_ncr[cuisine]=1
            else:
                d_ncr[cuisine]+=1
        if i[3]=='Rest of India':
            if cuisine not in d_non_ncr:
                d_non_ncr[cuisine]=1
            else:
                d_non_ncr[cuisine]+=1

```

creating two dicts one for ncr and the other for non ncr to store cuisines with their counts ...

```
a=set(d_ncr.keys())
b=set(d_non_ncr.keys())

```

```
b_not_a=b-a

```

creating sets to get unique cuisines for ncr and rest of india both , and then subtracting the sets to obtain not of a(cuisines not in ncr).

```
b_not_a
# cuisines not in ncr
{'BBQ', 'Cajun', 'German', 'Malwani'}
```

these are the cuisines not present in delhi ncr (according to the dataset)

Cross checking the result from zomato api-

```
r=requests.get('https://developers.zomato.com/api/v2.1/cuisines?city_id=1',headers={'Accept': "application/json","user-key": "67aebde6b3540160370101f53e64fec3"})
data=r.json()
```

```
for i in data['cuisines']:
    if i['cuisine']['cuisine_name'] in b_not_a:
        print(i['cuisine']['cuisine_name'])
```

Output-

BBQ

Malwani

Justification -

According to the zomato api it looks like the german cuisine is present in delhi ncr , so we can make that out that the dataset is incomplete and we can not certainly say that the following cuisines are not there in delhi ncr ...

Q3 . Find the maximum no of cuisines served by restaurants in delhi ncr and rest of india ?

Code -

```
cuisines_count={}
```

```
for i in df.values:
    cuisines=i[9].split(',')
    for cuisine in cuisines:
        cuisine=cuisine.strip(' ')

        if cuisine not in cuisines_count:
            cuisines_count[cuisine]=1
```

```
else:  
    cuisines_count[cuisine]+=1
```

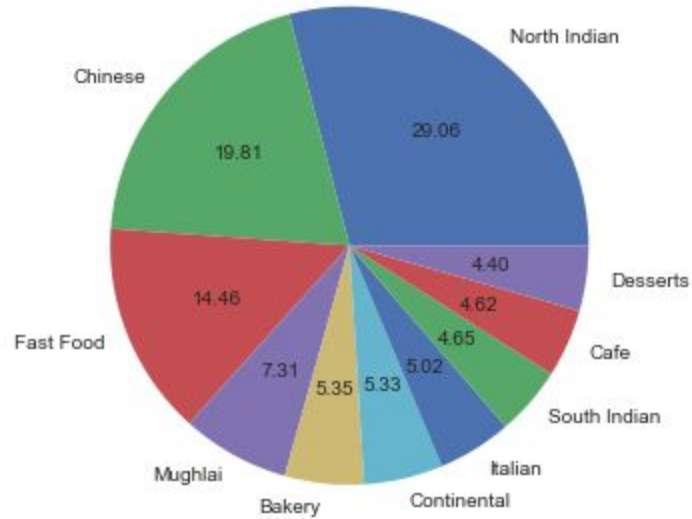
we created a dict for storing the cuisine count for both ncr and rest of india

```
ans=sorted(cuisines_count.items(),key=lambda x:x[1],reverse=True)  
#sorting to get cuisines with highest values first
```

Response-

```
[('North Indian', 3946), ('Chinese', 2690), ('Fast Food', 1963), ('Mughlai', 992), ('Bakery', 726),  
('Continental', 724), ('Italian', 682), ('South Indian', 631), ('Cafe', 627), ('Desserts', 597), ('Street  
Food', 554), ('Mithai', 380), ('Pizza', 313), ('American', 225), ('Ice Cream', 216), ('Beverages',  
216), ('Thai', 205), ('Asian', 186), ('Biryani', 175), ('Burger', 165), ('Healthy Food', 146),  
('Mexican', 130), ('European', 119), ('Raw Meats', 114), ('Finger Food', 109), ('Salad', 92),  
('Mediterranean', 90), ('Japanese', 82), ('Seafood', 81), ('Lebanese', 65), ('Tea', 44), ('Tibetan',  
44), ('Bengali', 29), ('Juices', 26), ('Hyderabadi', 25), ('Kerala', 22), ('Sushi', 21), ('Rajasthani',  
20), ('Arabian', 20), ('Kashmiri', 20), ('Goan', 19), ('Malaysian', 18), ('Modern Indian', 16),  
('Tex-Mex', 16), ('Korean', 16), ('Vietnamese', 14), ('Lucknowi', 13), ('Middle Eastern', 13),  
('French', 12), ('Gujarati', 11), ('Chettinad', 11), ('Awadhi', 11), ('Afghani', 11), ('Maharashtrian',  
10), ('Burmese', 10), ('Andhra', 10), ('Nepalese', 9), ('Parsi', 8), ('North Eastern', 8), ('Turkish', 8),  
('Naga', 8), ('Spanish', 7), ('Indian', 7), ('Portuguese', 6), ('Bihari', 6), ('Indonesian', 6), ('Greek',  
5), ('African', 5), ('Moroccan', 5), ('Charcoal Grill', 4), ('Assamese', 4), ('Mangalorean', 4),  
('Armenian', 3), ('Iranian', 3), ('British', 3), ('Sandwich', 2), ('Steak', 2), ('German', 2), ('South  
American', 2), ('Drinks Only', 2), ('Pakistani', 2), ('Oriya', 2), ('Belgian', 1), ('Persian', 1),  
('Malwani', 1), ('Cuisine Varies', 1), ('Deli', 1), ('Sri Lankan', 1), ('BBQ', 1), ('Cajun', 1)]
```

```
plt.pie([i[1] for i in ans[:10]],labels=[i[0] for i in ans],autopct='%%.2f',)  
plt.show() # crating pie chart for top 10 cuisines
```



I was not sure if the question expected a collective cuisine count for all of india or rest of india and delhi ncr separately so i have done both ...

Here's the code and pie chart for top 10 cuisines in delhi ncr and rest of india (separately)-

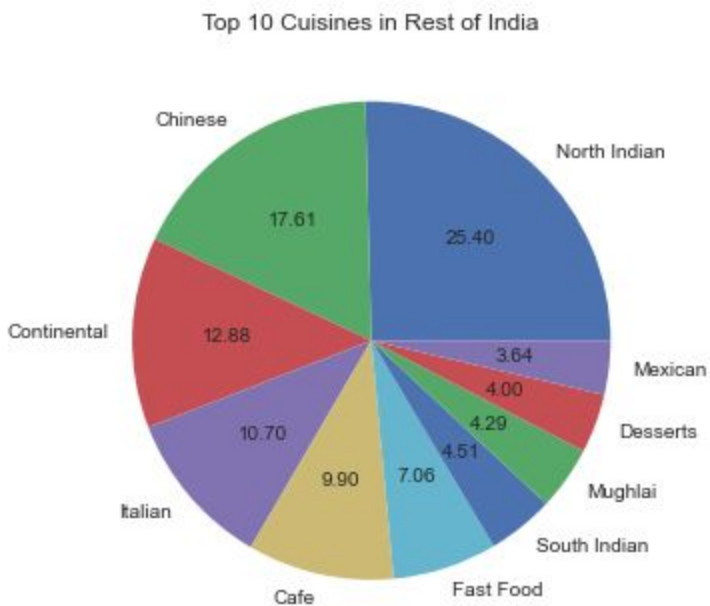
```
d_ncr={}
d_non_ncr={}
```

```
for i in df.values:
    cuisines=i[9].split(',')
    for cuisine in cuisines:
        cuisine=cuisine.strip(' ')
        if i[3]=='Delhi-Ncr':
            if cuisine not in d_ncr:
                d_ncr[cuisine]=1
            else:
                d_ncr[cuisine]+=1
        if i[3]=='Rest of India':
            if cuisine not in d_non_ncr:
                d_non_ncr[cuisine]=1
            else:
                d_non_ncr[cuisine]+=1
```

```
ans_ncr=sorted(d_ncr.items(),key=lambda x:x[1],reverse=True)
```

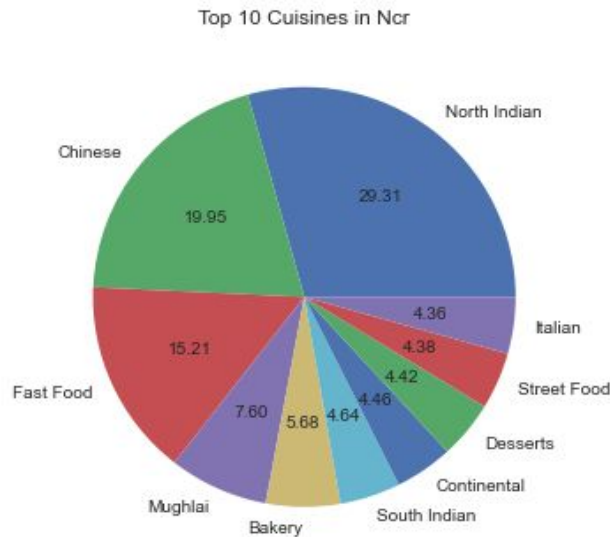
```
ans_non_ncr=sorted(d_non_ncr.items(),key=lambda x:x[1],reverse=True)
```

```
plt.pie(x=[i[1] for i in ans_non_ncr[:10]],labels=[i[0] for i in ans_non_ncr[:10]],autopct='%.2f')  
plt.title('Top 10 Cuisines in Rest of India')  
plt.show()
```



Plot for ncr -

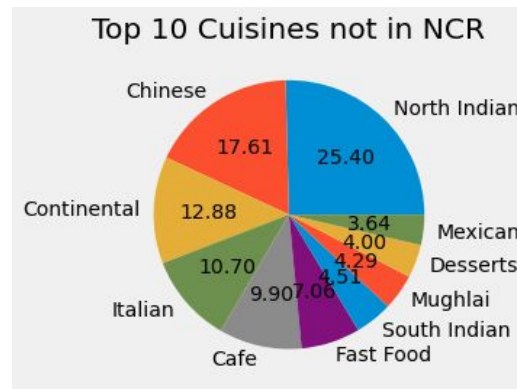
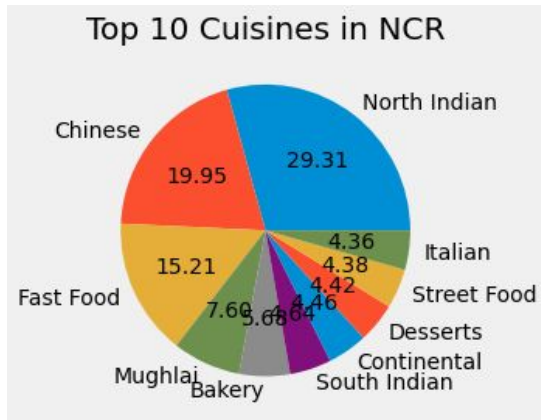
```
plt.pie(x=[i[1] for i in ans_ncr[:10]],labels=[i[0] for i in ans_ncr[:10]],autopct='%.2f')  
plt.title('Top 10 Cuisines in Ncr')  
plt.show()
```



Justification - from the plots its evident that north indian and chinese are the top 2 cuisines in both rest of india and delhi ncr whereas the top 3rd in delhi ncr is fast food and in rest of india its continental . Leaving the top 2 cuisines both delhi ncr and rest of india have varied results with different percentages and diff cuisines.

Q4. Write a Short detailed analysis of how cuisines served in delhi ncr is different from cuisine served in Rest of india . Plot a suitable graph to explain your inference ?

Lets look at the pie plots for the top 10 cuisines served in ncr and non ncr and try to draw some inferences from that -



Leaving the top 2 cuisines Chinese and North Indian behind, the two plots look varied. According to the plots, the 3rd top cuisine in Delhi NCR is Fast Food, whereas in ROI it is Continental. Similarly, the top 4th in Delhi NCR is Mughlai cuisine, whereas in ROI it is Italian. So, according to the plots, the cuisine preference in Delhi NCR and ROI is slightly different.

Now let's look at the cuisines which are served only in Delhi and ROI -

Code -

```
global cuisine_ncr
cuisine_ncr=[]
cuisine_non_ncr=[]
```

```
def cuisine_seperator(x,t):
    x=x.split(',')
    x=[i.strip() for i in x]
    for i in x:
        i=i.strip()
        if t==1 and i not in cuisine_ncr:
            cuisine_ncr.append(i)
        if t==2 and i not in cuisine_non_ncr:
            cuisine_non_ncr.append(i)
    return x
```

```
(df[df['City']=='Delhi-NCR']['Cuisines']).apply(cuisine_seperator,args=(1,))
(df[df['City']=='Rest Of India']['Cuisines']).apply(cuisine_seperator,args=(2,))
```

```
cuisine_ncr=set(cuisine_ncr)
cuisine_non_ncr=set(cuisine_non_ncr)
```

So cuisines not present in ncr are -
cuisines_not_inncr=cuisine_non_ncr-cuisine_ncr

```
{'BBQ', 'Cajun', 'German', 'Malwani'}
```

```
print('Cuisines served only in Delhi ')
for i in cuisine_ncr-cuisine_non_ncr:
    print(i)
```

Cuisines served only in Delhi

Cuisine Varies

Afghani

Belgian

Turkish

Drinks Only

Sri Lankan

Bihari

Deli

Persian

Iranian

Sushi

Moroccan

Pakistani

Raw Meats

Naga

Oriya

Assamese

Kashmiri

Nepalese

South American

Now lets look at the common cuisines and which out of roi and delhincr prefers which cuisine-

```
d_ncr={}
for i in (df[df['City']=='Delhi-NCR']['Cuisines']).apply(cuisine_seperator,args=(1,)):
    for j in i:
```

```

        if j not in d_ncr:
            d_ncr[j]=1
        else:
            d_ncr[j]+=1

d_ncr=sorted(d_ncr.items(),key=lambda x: x[1],reverse=True)

d_non_ncr={}
for i in (df[df['City']=='Rest Of India']['Cuisines']).apply(cuisine_seperator,args=(2,)):
    for j in i:
        if j not in d_non_ncr:
            d_non_ncr[j]=1
        else:
            d_non_ncr[j]+=1

d_non_ncr=sorted(d_non_ncr.items(),key=lambda x: x[1],reverse=True)

common_cuisines=cuisine_ncr.intersection(cuisine_non_ncr)

d={i:[] for i in common_cuisines}

for i,j in zip(d_ncr,d_non_ncr):
    if i[0] in d:
        d[i[0]].append((i[1],'Delhi-Ncr'))
    if j[0] in d:
        d[j[0]].append((j[1],'Rest Of India'))
for j in d:
    arr=d[j]
    if len(arr)==0:
        continue
    if len(arr)==1:
        print(j,arr[0])
        continue
    if arr[0][0]>arr[1][0]:
        print(f'cuisine: {j} is served more in {arr[0]}')
    if arr[1][0]>arr[0][0]:
        print(f'cuisine: {j} is served more in {arr[1]}')

```

output -

cuisine: Juices is served more in (23, 'Delhi-Ncr')
cuisine: Rajasthani is served more in (14, 'Delhi-Ncr')
cuisine: Mediterranean is served more in (60, 'Delhi-Ncr')
cuisine: Arabian is served more in (15, 'Delhi-Ncr')
cuisine: Mithai is served more in (379, 'Delhi-Ncr')
cuisine: Spanish is served more in (5, 'Delhi-Ncr')
cuisine: Andhra is served more in (7, 'Rest Of India')
cuisine: Gujarati is served more in (8, 'Delhi-Ncr')
cuisine: Bakery is served more in (697, 'Delhi-Ncr')
cuisine: Thai is served more in (176, 'Delhi-Ncr')
cuisine: Tea is served more in (38, 'Delhi-Ncr')
cuisine: Ice Cream is served more in (210, 'Delhi-Ncr')
cuisine: Lucknowi is served more in (12, 'Delhi-Ncr')
cuisine: North Eastern is served more in (7, 'Delhi-Ncr')
cuisine: African is served more in (4, 'Delhi-Ncr')
cuisine: Lebanese is served more in (57, 'Delhi-Ncr')
cuisine: Desserts is served more in (542, 'Delhi-Ncr')
cuisine: Hyderabadi is served more in (23, 'Delhi-Ncr')
cuisine: Goan is served more in (13, 'Rest Of India')
cuisine: Bengali is served more in (27, 'Delhi-Ncr')
cuisine: Seafood is served more in (53, 'Delhi-Ncr')
cuisine: Japanese is served more in (75, 'Delhi-Ncr')
cuisine: South Indian is served more in (569, 'Delhi-Ncr')
cuisine: Beverages is served more in (204, 'Delhi-Ncr')
Charcoal Grill (3, 'Rest Of India')
cuisine: Middle Eastern is served more in (11, 'Delhi-Ncr')
cuisine: American is served more in (193, 'Delhi-Ncr')
cuisine: Fast Food is served more in (1866, 'Delhi-Ncr')
cuisine: Vietnamese is served more in (12, 'Delhi-Ncr')
cuisine: Awadhi is served more in (10, 'Delhi-Ncr')
cuisine: Burmese is served more in (6, 'Delhi-Ncr')
cuisine: Finger Food is served more in (92, 'Delhi-Ncr')
cuisine: Modern Indian is served more in (9, 'Delhi-Ncr')
cuisine: Indonesian is served more in (4, 'Delhi-Ncr')
Steak (1, 'Rest Of India')
cuisine: Continental is served more in (547, 'Delhi-Ncr')
cuisine: Indian is served more in (5, 'Delhi-Ncr')
cuisine: European is served more in (87, 'Delhi-Ncr')
cuisine: Asian is served more in (151, 'Delhi-Ncr')
cuisine: Parsi is served more in (7, 'Delhi-Ncr')
cuisine: Mughlai is served more in (933, 'Delhi-Ncr')
cuisine: Burger is served more in (152, 'Delhi-Ncr')
cuisine: Healthy Food is served more in (143, 'Delhi-Ncr')
cuisine: Portuguese is served more in (5, 'Delhi-Ncr')
cuisine: Pizza is served more in (282, 'Delhi-Ncr')

cuisine: Salad is served more in (89, 'Delhi-Ncr')
cuisine: Korean is served more in (15, 'Delhi-Ncr')
British (1, 'Rest Of India')
cuisine: Tibetan is served more in (41, 'Delhi-Ncr')
Armenian (1, 'Rest Of India')
cuisine: Italian is served more in (535, 'Delhi-Ncr')
Mangalorean (3, 'Rest Of India')
cuisine: Biryani is served more in (160, 'Delhi-Ncr')
cuisine: Tex-Mex is served more in (9, 'Delhi-Ncr')
Greek (3, 'Rest Of India')
cuisine: Street Food is served more in (538, 'Delhi-Ncr')
cuisine: Cafe is served more in (491, 'Delhi-Ncr')
cuisine: Malaysian is served more in (15, 'Delhi-Ncr')
cuisine: North Indian is served more in (3597, 'Delhi-Ncr')
cuisine: Mexican is served more in (80, 'Delhi-Ncr')
cuisine: Chettinad is served more in (8, 'Delhi-Ncr')
cuisine: Kerala is served more in (15, 'Delhi-Ncr')
cuisine: Chinese is served more in (2448, 'Delhi-Ncr')
Sandwich (1, 'Rest Of India')
cuisine: Maharashtrian is served more in (6, 'Delhi-Ncr')

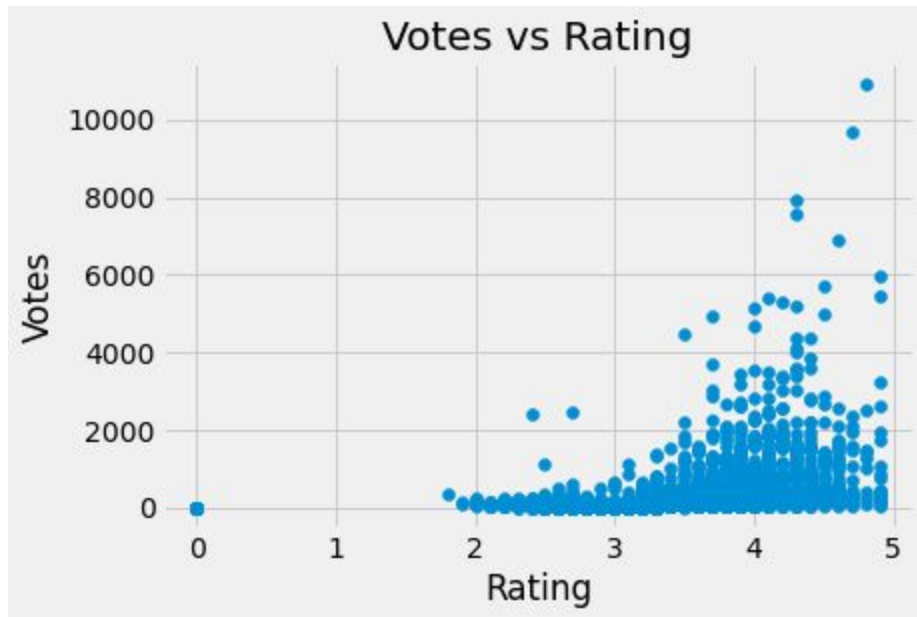
It is clear that the data is skewed more towards delhi ncr and comparing cuisines on that basis won't give us the actual view but we can say that according to our dataset I was able to draw the following inferences about cuisine difference between ncr and roi .

Ques2 : User rating of a restaurant plays as crucial role in selecting a restaurant or ordering the food from the restaurant .

2.1 Write a short detailed analysis of how rating is affected by the following features also plot suitable graphs to explain your inference

Lets create a scatter plot -

```
plt.scatter(y=df['Votes'],x=df['Aggregate rating'])  
plt.title('Votes vs Rating')  
plt.xlabel('Rating')  
plt.ylabel('Votes')
```



Inference - # from the scatter plot it looks like most of the differently rated restaurants lie between 0-2000 votes

there are some restaurants which have really high amount of votes and have a overall higher rating

apart from some outliers the graph appears to be climbing which makes sense as more the number of votes the more is the restaurant visited which means higher ratings

Ques 2.2 Restaurants serving more number of cuisines -?

Lets put the length of each cuisine acc to the restaurant -

```
def cuisine_length(x):
```

```
    x=x.split(',')

```

```
    return len(x)

```

```
cuisine=df['Cuisines'].apply(cuisine_length)
```

Now creating a scatter plot -

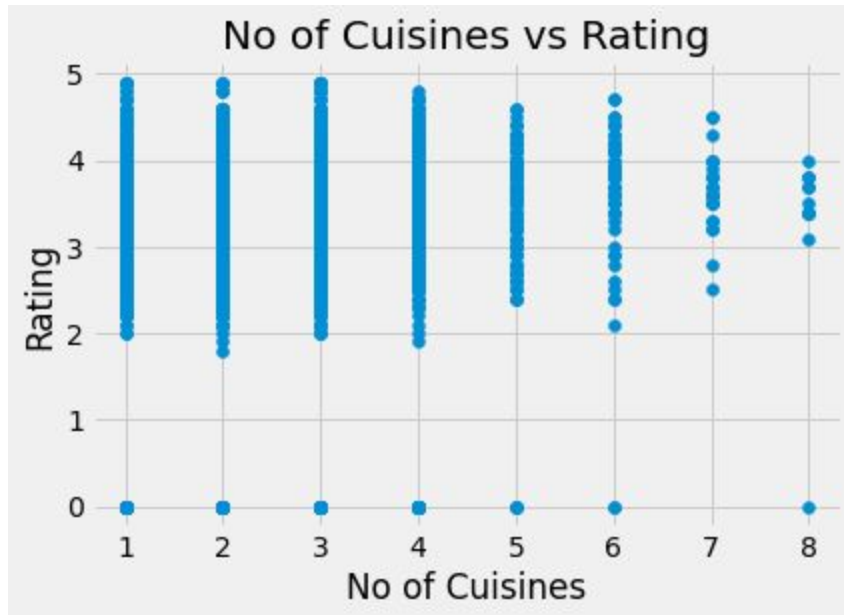
```
plt.scatter(x=cuisine,y=df['Aggregate rating'])
```

```
plt.title('No of Cuisines vs Rating')
```

```
plt.ylabel('Rating')
```

```
plt.xlabel('No of Cuisines')
```

```
plt.scatter(y=df['Aggregate rating'],x=df['Average Cost for two'])  
plt.title('Rating vs Average Cost')  
plt.xlabel('Average Cost')  
plt.ylabel('Rating')
```

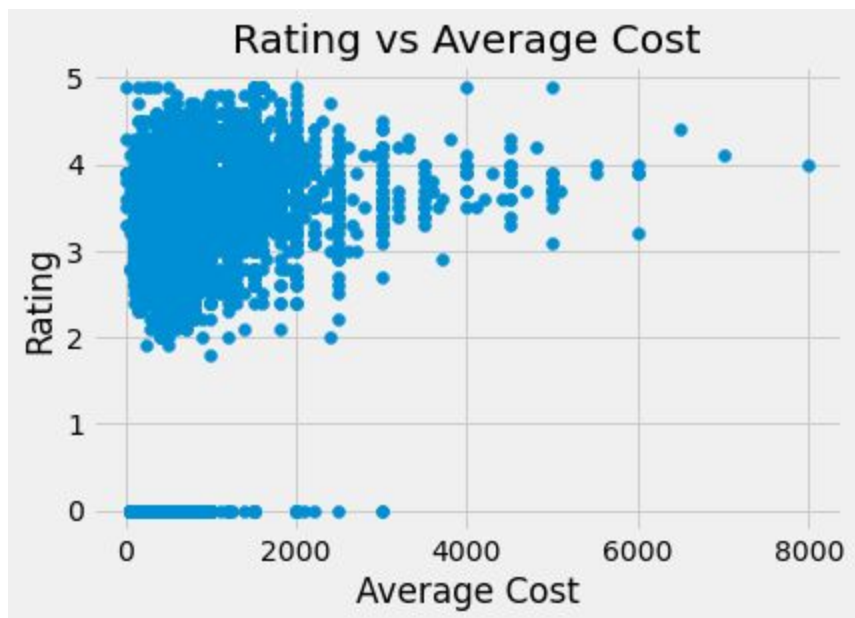


we can infer from the plot that restaurants with lower no of cuisines have varied rating or in other words they have relatively a higher spread of rating
we can also infer that restaurants with less cuisines have a relatively higher chance of getting a higher rating
so it makes sense as the more cuisines you serve the varied the food eaten would be and not all cuisines would be best so having more cuisines would generally point to a lower chance of grabbing a high rating

2.3 Average cost of the restaurant -

```
plt.scatter(y=df['Aggregate rating'],x=df['Average Cost for two'])  
plt.title('Rating vs Average Cost')  
plt.xlabel('Average Cost')
```

```
plt.ylabel('Rating')
```



```
# it looks like restaurants from average cost from 0-2000 have a more consistent and
more varied spread of rating
# theres one restaurant which has 8000 average cost and has only 4.0 rating
# it also looks like that from 2000 to 8000 cost the rating is dropping so it seems that
more costlier restaurants have a lower rating on average (excluding the outliers )
```

2.4 Restaurants serving some specific cuisines -

```
def cuisine_length(x):
    x=x.split(',')

    if len(x)==1:
        return x[0]
    else:
        return -1
```

```
cuisine=df['Cuisines'].apply(cuisine_length)
```



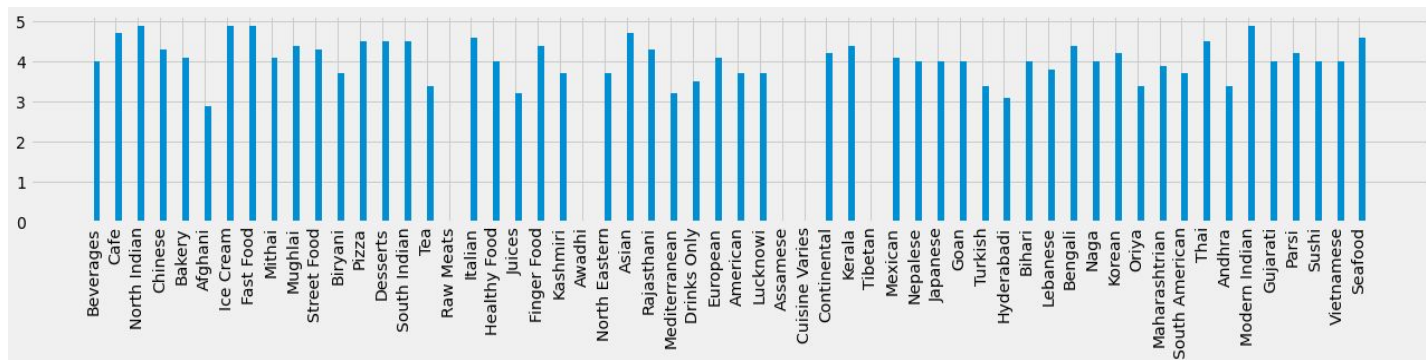
```
df['Cuisines']=cuisine
```

```
cuisine=cuisine[cuisine!=-1]
```

i defined a function to drop out the restaurants which have more number of cuisines and only saved the ones which were serving only one cuisine

Lets look at the plot now -

```
plt.figure(figsize=(20, 3))  
plt.bar(df['Cuisines'],df['Aggregate rating'],align='edge',width=0.3)  
plt.xticks(rotation=90)  
plt.show()
```



From the plot it looks like the ratings are relatively not that varied but there are some cuisines with on average a higher rating whereas some cuisines have lower average rating.

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

Code -

```
df['rating_votes']=df['Aggregate rating']*df['Votes']
```

```
d_locality={}
for i in list(df['rating_votes'].groupby(df['Locality'])):
    name,data=i
    d_locality[name]=data.sum()
```

#summing up the rating * votes as per locality

```
d_city_votes={}
for i in list(df['Votes'].groupby(df['Locality'])):
    city,data=i
    d_city_votes[city]=data.sum()
```

#summing up the votes as per locality

```
d_weighted_rating={}
for i in d_locality:
    d_weighted_rating[i]=d_locality[i]/d_city_votes[i]
```

#calculating weighted average of each locality

```
ans=d_weighted_rating.items()
```

Output-

```
dict_items([('ILD Trade Centre Mall, Sohna Road', 2.8166666666666664), ('12th Square Building, Banjara Hills', 4.3), ('A Hotel, Gurdev Nagar', 3.6), ('ARSS Mall, Paschim Vihar', 3.1), ('Aaya Nagar', 0.0), ('Acropolis Mall, Kasba ', 4.3845703125), ('Adajan Gam', 3.887726358148893), ('Adchini', 3.64275641025641), ('Aditya Mega Mall, Karkardooma', 3.4700460829493087), ('Adyar', 4.2), ('Aerocity', 3.095081967213115), ('Aggar Nagar', 3.5), ('Aggarwal City Mall, Pitampura', 2.976984126984127), ('Aggarwal City Plaza, Rohini', 3.118013468013468), ('Agra Cantt', 3.714426229508197), ('Akashwani', 3.6), ('Alaknanda', 3.50221793635487), ('Alkapuri', 4.218141870684244), ('Ambavadi', 4.2), ('Ambience Mall, Gurgaon', 4.210524358497582), ('Ambience Mall, Vasant Kunj', 3.4499593495934957), ('Aminabad', 4.9).....]
```

Now we will arrange and get the top 10 localities out of this -

```
ans=sorted(d_weighted_rating.items(),key=lambda x:x[1],reverse=True) # weighted rating per city
```

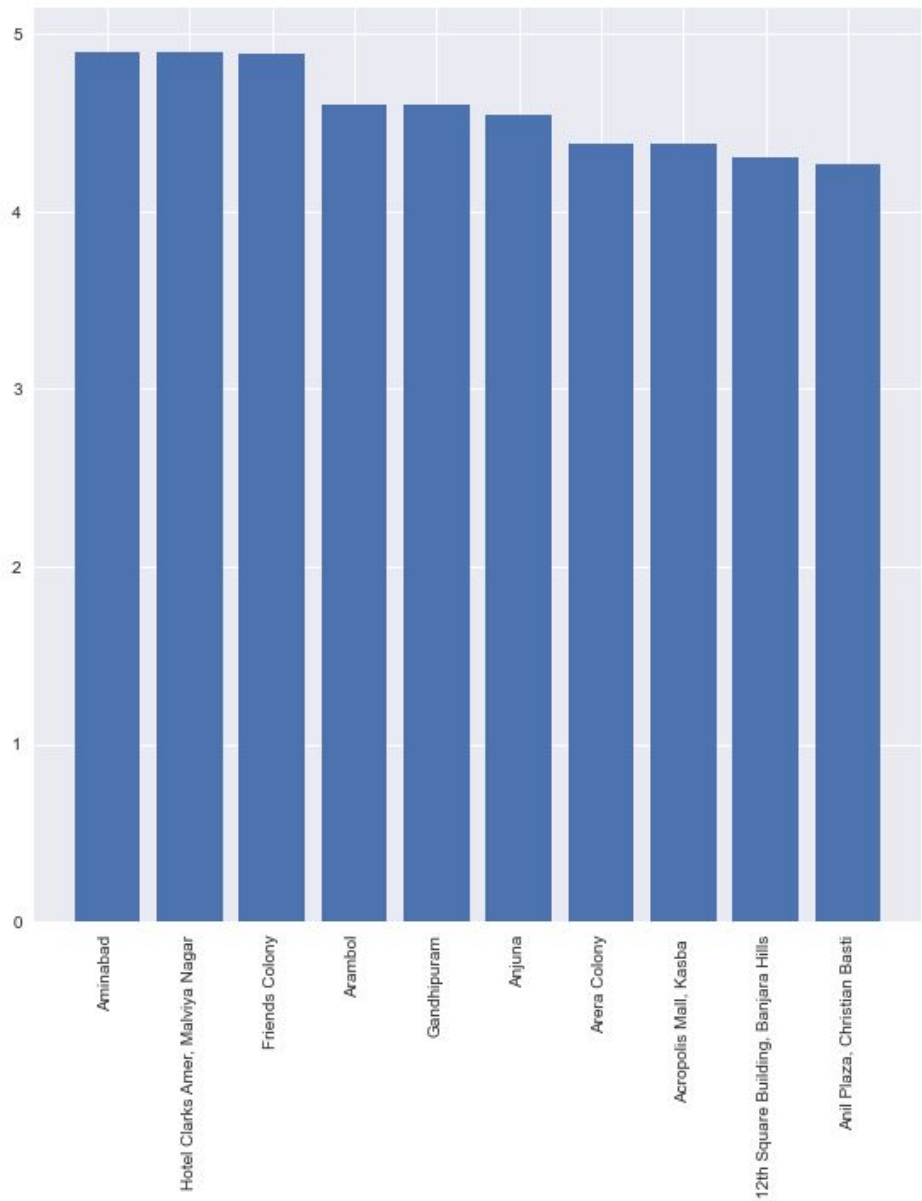
```
list(ans[:10])
```

Top 10 localities on the basis of weighted average rating -

```
[('Aminabad', 4.9),
 ('Hotel Clarks Amer, Malviya Nagar', 4.9),
```

('Friends Colony', 4.886916367367881),
('Arambol', 4.6),
('Gandhipuram', 4.6),
('Anjuna', 4.5446681580909765),
('Arera Colony', 4.385168539325843),
('Acropolis Mall, Kasba ', 4.3845703125),
('12th Square Building, Banjara Hills', 4.3),
('Anil Plaza, Christian Basti', 4.265782493368699)]

Plot -



Ans -

It looks like hotel clarke .. locality has a whopping rating of 4.9 followed by malvia nagar and you can check the later from the plot . So these were the top 10 localities on the basis of weighted average rating.v

Q3. Visualization -

3.1 Plot the bar graph of top 15 restaurants having maximum number of outlets in india -

Code-

```
df_copy=df.copy()
```

```
d_restaurant_count={}
```

```
for i in list(df_copy['Address'].groupby(df_copy['Restaurant Name'])):
```

```
    name,data=i
```

```
    no_of_entries=len(data)
```

```
    d_restaurant_count[name]=no_of_entries
```

```
ans=sorted(d_restaurant_count.items(),key=lambda x:x[1],reverse=True)[:15]
```

```
print(ans)
```

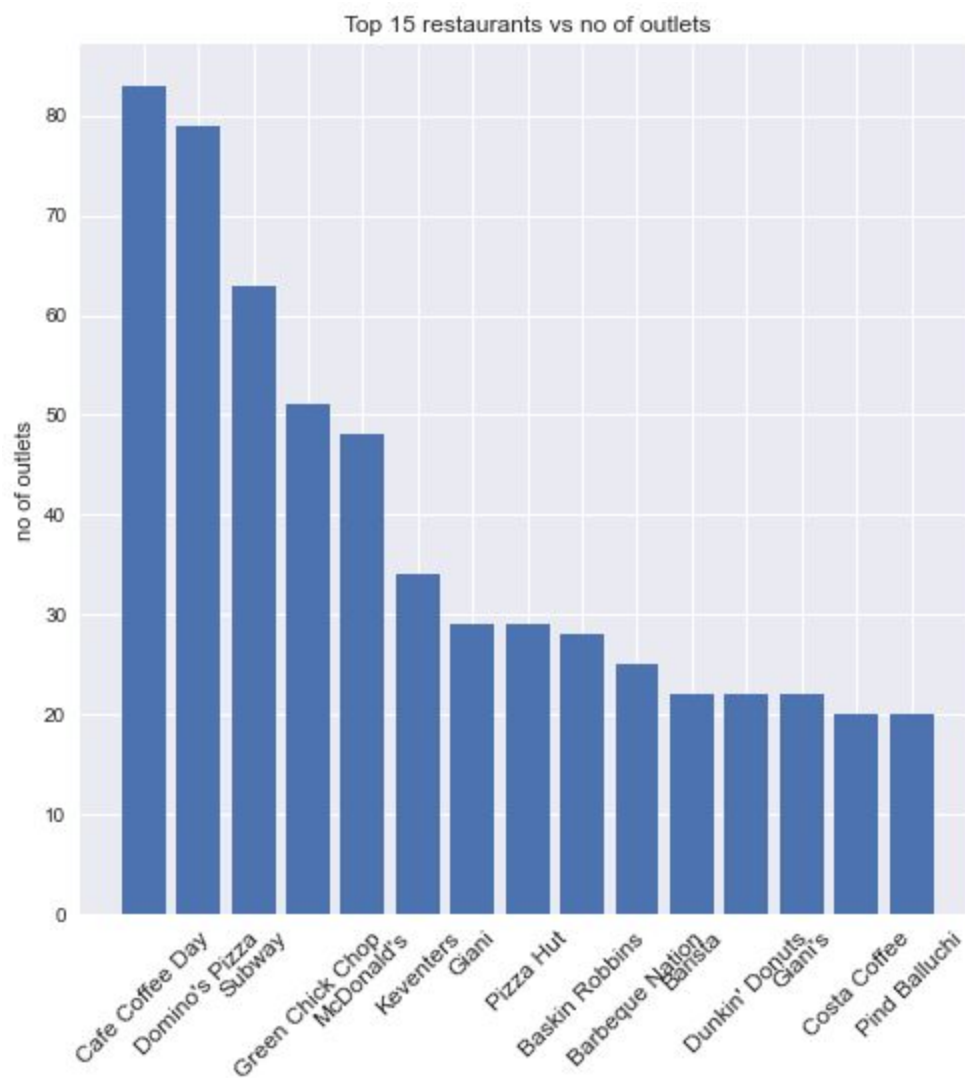
Output-

```
[('Cafe Coffee Day', 83),  
 ("Domino's Pizza", 79),  
 ('Subway', 63),  
 ('Green Chick Chop', 51),  
 ("McDonald's", 48),  
 ('Keventers', 34),  
 ('Giani', 29),  
 ('Pizza Hut', 29),  
 ('Baskin Robbins', 28),  
 ('Barbeque Nation', 25),  
 ('Barista', 22),  
 ("Dunkin' Donuts", 22),  
 ("Giani's", 22),  
 ('Costa Coffee', 20),
```

('Pind Balluchi', 20)]

Plot -

```
plt.figure(figsize=(8,8))
plt.bar([i[0] for i in ans],[i[1] for i in ans])
plt.xticks( size='large',rotation=45)
plt.ylabel('no of outlets')
plt.title('Top 15 restaurants vs no of outlets')
```



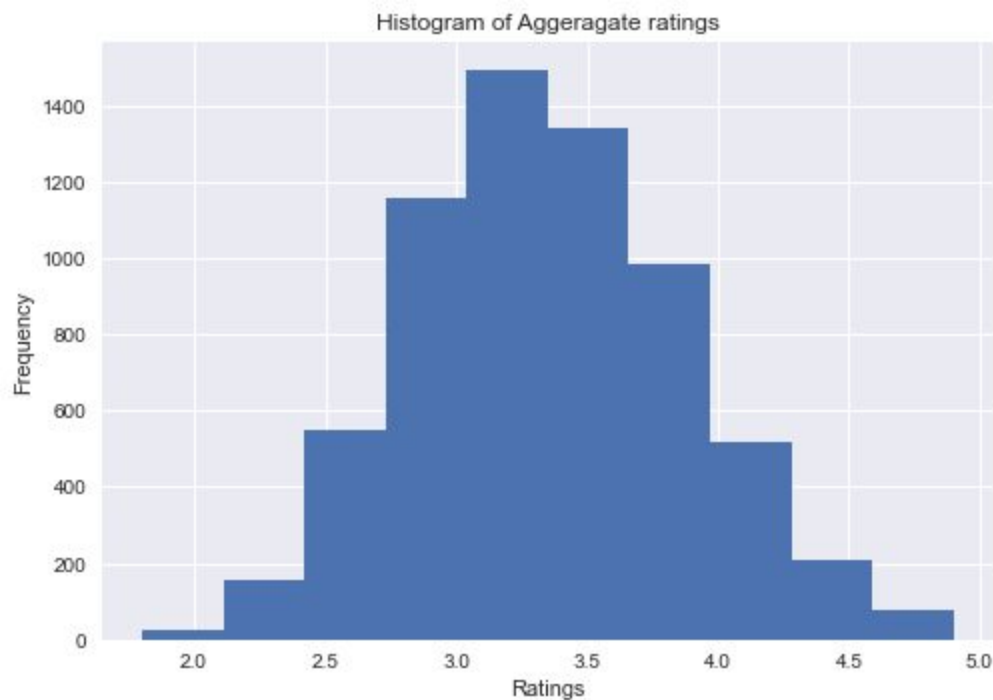
Justification - Again its pretty evident from the plot that according to our dataset - Ccd has 83 outlets followed by dominoes pizza and subway and for the later you can refer to the bar plot

Q3.2 Plot the histogram of the aggregate rating of the restaurant (drop the unrated restaurants)?

Code -

```
df_copy=df_copy[df_copy['Aggregate rating']!=0]
```

```
df_copy['Aggregate rating'].plot.hist()  
plt.xlabel('Ratings')  
plt.title('Histogram of Aggeragate ratings')
```



Justification - As expected we have more restaurants which have a rating in between 3.0 and 3.5 . There are lower no of restaurants on both the left side and the right side. So the best thing i can infer from the plot is that its an almost perfect normal distribution.

Ques. Top 10 restaurants in the data with the highest number of votes ?

```
d_restaurant_votes={}
for i in list(df["Votes"].groupby(df["Restaurant Name"])):
    name,data=i
    d_restaurant_votes[name]=data.sum()
```

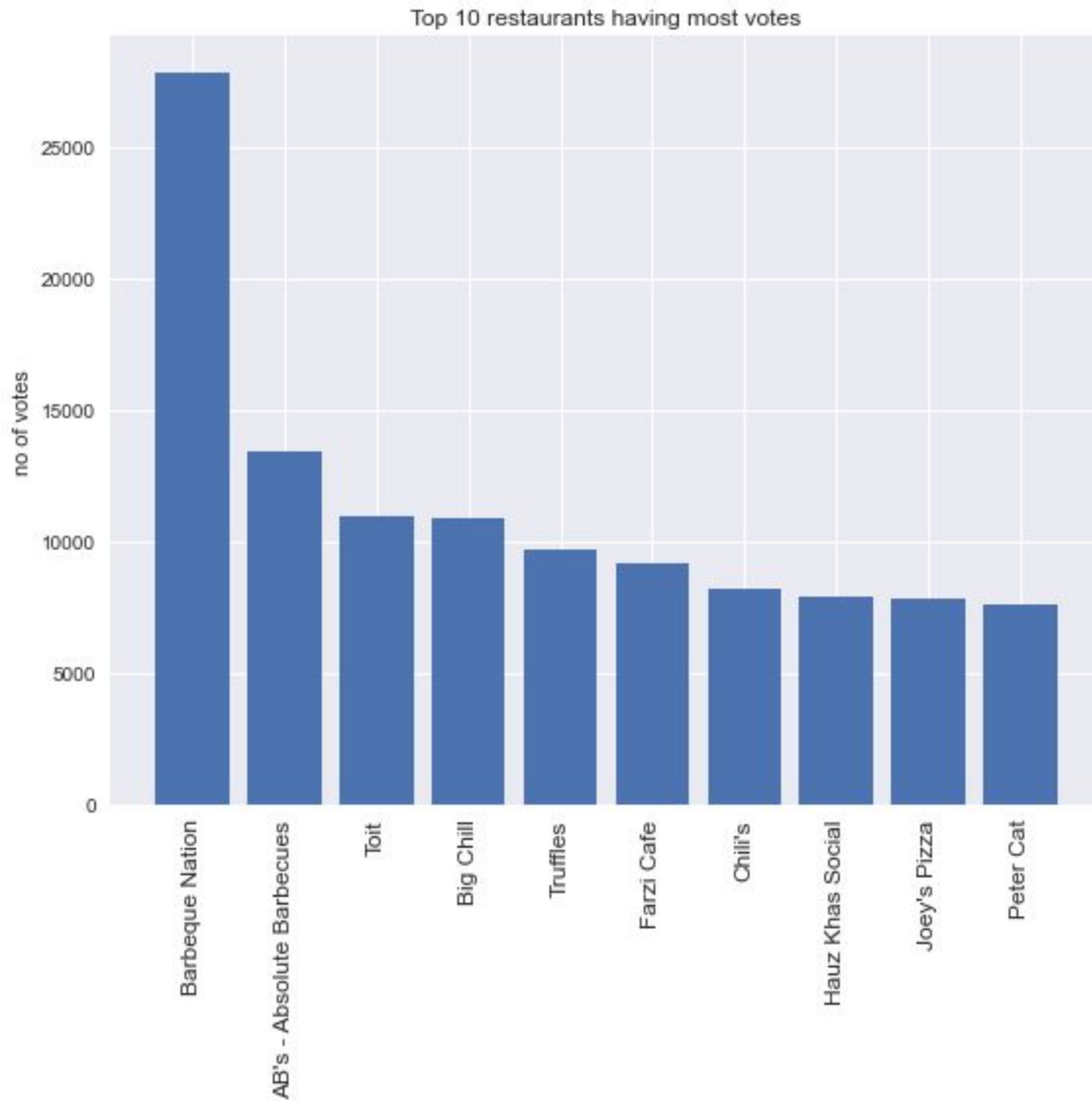
as restaurants have outlets i have grouped them by their name and then summed the result to get the total votes

```
ans=sorted(d_restaurant_votes.items(),key=lambda x:x[1],reverse=True)[:10] #sorting the answers
```

```
[('Barbeque Nation', 27835),
('AB's - Absolute Barbecues", 13400),
('Toit', 10934),
('Big Chill', 10853),
('Truffles', 9682),
('Farzi Cafe', 9189),
('Chili's", 8156),
('Hauz Khas Social', 7931),
('Joey's Pizza", 7807),
('Peter Cat', 7574)]
```

Plot-

```
plt.figure(figsize=(8,8))
plt.bar([i[0] for i in ans],[i[1] for i in ans])
plt.xticks( size='large',rotation=90)
plt.ylabel('no of votes')
plt.title('Top 10 restaurants having most votes')
plt.tight_layout()
```



Justification - This answer requires no justification as we can already infer from the plot that bbqn has most votes followed by abs and toit and big chill And so on ..

Ques 4 . Plot the pie chart of the top 10 cuisines present in the usa ?

```
df=pd.read_csv('zomato.csv',encoding='ISO-8859-1')
df_usa=df[df['Country Code']==216] # usa's country code is 216

df_usa['City'].unique()
```



```
array(['Albany', 'Athens', 'Augusta', 'Boise', 'Cedar Rapids/Iowa City',  
      'Clatskanie', 'Cochrane', 'Columbus', 'Dalton', 'Davenport',  
      'Des Moines', 'Dubuque', 'Fernley', 'Gainesville', 'Lakeview',  
      'Lincoln', 'Macon', 'Mc Millan', 'Miller', 'Monroe',  
      'Ojo Caliente', 'Orlando', 'Pensacola', 'Pocatello', 'Potrero',  
      'Princeton', 'Rest of Hawaii', 'Savannah', 'Sioux City',  
      'Tampa Bay', 'Valdosta', 'Vernonia', 'Waterloo', 'Weirton',  
      'Winchester Bay'], dtype=object)
```

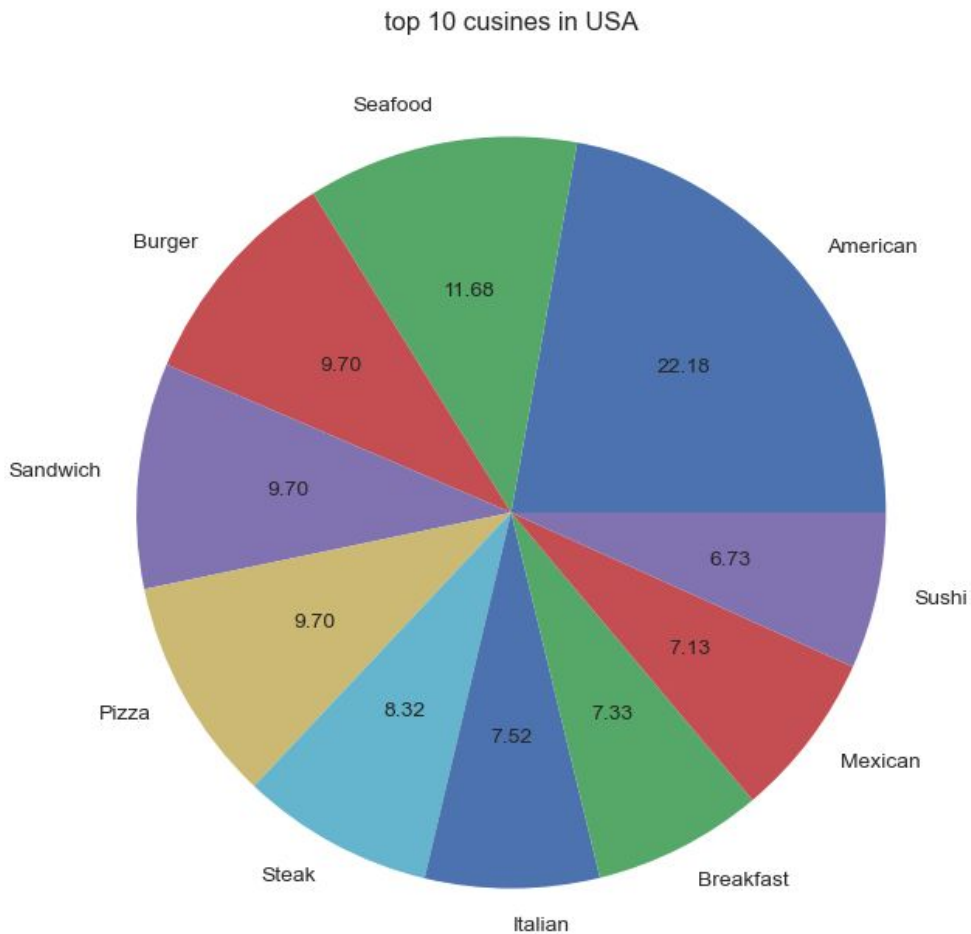
```
df_usa['Cuisines'].dropna(inplace=True)
```

```
d_cuisine={}  
def cuisine_split(x):  
    cuisines=x.split(',')  
    for i in cuisines:  
        i=i.strip()  
        if i not in d_cuisine:  
            d_cuisine[i]=1  
        else:  
            d_cuisine[i]+=1  
    return x
```

```
df_usa['Cuisines'].apply(cuisine_split)
```

```
ans=sorted(d_cuisine.items(),key=lambda x:x[1],reverse=True)[:10]
```

```
plt.figure(figsize=(8,8),dpi=100)  
plt.pie([i[1] for i in ans],labels=[i[0] for i in ans],autopct='%.2f')  
plt.title('top 10 cuisines in USA')  
plt.show()
```



Ques 3.4

Code-

```
df_india=df[df['Country Code']==1]
df_india['rating_votes']=df_india['Aggregate rating']*df_india['Votes']

d_city_rating_votes={}
for i in list(df_india['rating_votes'].groupby(df_india['City'])):
    name,data=i
    d_city_rating_votes[name]=data.sum()

d_city_rating_votes
```

```

d_city_votes={}
for i in list(df_india["Votes"].groupby(df_india["City"])):
    name,data=i
    d_city_votes[name]=data.sum()

d_city_weighted={}
for i in d_city_votes:
    d_city_weighted[i]=d_city_rating_votes[i]/d_city_votes[i]

```

```

ans_city_weighted=d_city_weighted.items()

```

```

city_no_of_restaurants={}
for i in df_india.groupby(df["City"]):
    name,data=i
    no=len(data)
    city_no_of_restaurants[name]=no

```

```

ans_city=city_no_of_restaurants.items()
ans_city

```

```

dict_items([('Agra', 20), ('Ahmedabad', 21), ('Allahabad', 20), ('Amritsar', 21), ('Aurangabad', 20), ('Bangalore', 20), ('Bhopal', 20), ('Bhubaneshwar', 21), ('Chandigarh', 18), ('Chennai', 20), ('Coimbatore', 20), ('Dehradun', 20), ('Faridabad', 251), ('Ghaziabad', 25), ('Goa', 20), ('Gurgaon', 1118), ('Guwahati', 21), ('Hyderabad', 18), ('Indore', 20), ('Jaipur', 20), ('Kanpur', 20), ('Kochi', 20), ('Kolkata', 20), ('Lucknow', 21), ('Ludhiana', 20), ('Mangalore', 20), ('Mohali', 1), ('Mumbai', 20), ('Mysore', 20), ('Nagpur', 20), ('Nashik', 20), ('New Delhi', 5473), ('Noida', 1080), ('Panchkula', 1), ('Patna', 20), ('Puducherry', 20), ('Pune', 20), ('Ranchi', 20), ('Secunderabad', 2), ('Surat', 20), ('Vadodara', 20), ('Varanasi', 20), ('Vizag', 20)])

```

Plot -