

Q1) 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.

In [ ]:

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

```
In [15]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import display, Markdown
iris=pd.read_csv("zomato.csv",encoding ="ISO-8859-1")
df=iris.copy()

#Fixing Column Names with spaces
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('(', '').str.replace(')', '')

#Applying country code India
df=df[(df.country_code==1)]

#Making NCR List
NCR=['New Delhi', 'Ghaziabad', 'Noida', 'Gurgaon', 'Faridabad']

#Initializing varibales for count
ncr_count=0
rest=0

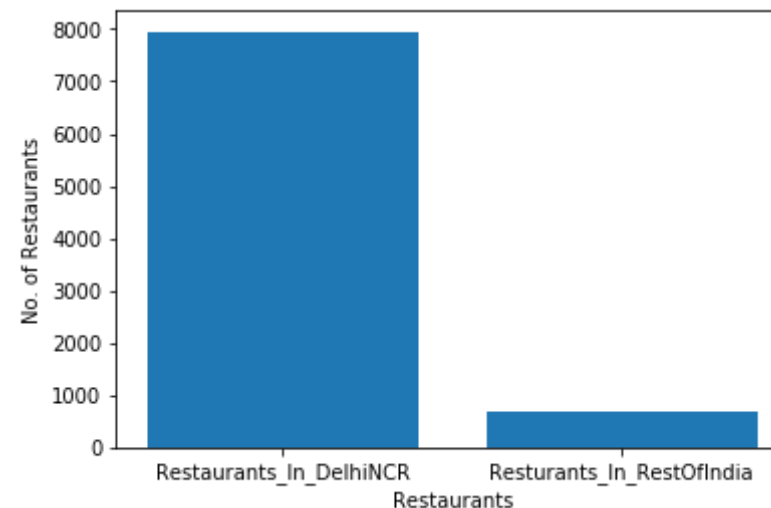
#Calculating Count
for i in range(df.shape[0]):
    if any(x in df.iloc[i,3] for x in NCR):
```

```

        ncr_count=ncr_count+1
    else:
        rest=rest+1

#Plotting with Bar Graph
x=["Restaurants_In_DelhiNCR","Restaurants_In_RestOfIndia"]
y=[ncr_count,rest]
pt.xlabel("Restaurants")
pt.ylabel("No. of Restaurants")
pt.bar(x,y)
pt.show()
display(Markdown('*Conclusion - Restaurants in Delhi-NCR are more than Rest of the India*'))

```



*Conclusion - Restaurants in Delhi-NCR are more than Rest of the India*

In [ ]:

Que 1 - Part 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.

```

In [16]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from IPython.display import display, Markdown
import requests
import json
from requests.auth import HTTPBasicAuth

iris=pd.read_csv("zomato.csv",encoding ="ISO-8859-1")
df=iris.copy()

#Fixing column name with spaces
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('(', '').str.replace(')', '')

#Applying country as India
df=df[(df.country_code==1)]

#Making NCR List
ncr=["New Delhi","Ghaziabad","Noida","Gurgaon","Faridabad"]

#Creating dictionary city as key and value as cuisine
d={}
for i in range(df.shape[0]):
    key=df.iloc[i,3]
    value=df.iloc[i,9]
    if key in d:
        d[key]=d[key]+","+value
    else:
        d[key]=value

#Creating ncr and rest of india list
ncr_l=[]
rest_l=[]
for i in d:
    if i in ncr:
        ll=d[i].split(",")
        for j in ll:
            ncr_l.append(j)

```

```

else:
    l2=d[i].split(",")
    for j in l2:
        rest_l.append(j)

#Removing Duplicates
s1=set()
s2=set()
for i in ncr_l:
    i=i.strip()
    s1.add(i)
for i in rest_l:
    i=i.strip()
    s2.add(i)

#Recreating list from set
ncr_l=list(s1)
rest_l=list(s2)

#Filtering the data
cuisine_list_csv=[]
display(Markdown('*Cuisines which are not present in NCR but are present in rest of the India*'))
for i in rest_l:
    if i not in ncr_l:
        print(i)
        cuisine_list_csv.append(i.strip())
#Part2 Verifying whether these cuisines actually don't exist
Complete_Cuisine_List=[]
para={"city_id":1}
header={"user-key":"db5cc0c3fa8e6679c4f19aeb201c1c0c"}
a=requests.get('https://developers.zomato.com/api/v2.1/cuisines',params=para,headers=header)
pdata=json.loads(a.text)
for i in pdata['cuisines']:
    Complete_Cuisine_List.append(i['cuisine']['cuisine_name'].strip())

display(Markdown('*Verifying whether above dishes are actually not present via API call*'))

```

```
for i in cuisine_list_csv:
    if i in Complete_Cuisine_List:
        print(i+" is present")
```

```
display(Markdown('*Conclusion - Based on the above inferences we can conclude that this result from csv is incorrect due to incomplete dataset.*'))
```

*Cuisines which are not present in NCR but are present in rest of the India*

BBQ  
Cajun  
Malwani  
German

*Verifying whether above dishes are actually not present via API call*

BBQ is present  
German is present

*Conclusion - Based on the above inferences we can conclude that this result from csv is incorrect due to incomplete dataset.*

In [ ]:

Que 1- Part 3) Find the top 10 cuisines served by maximum number of restaurants in Delhi NCR and rest of India.

In [17]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from IPython.display import display, Markdown
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()

#Fixing columns with space in name
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('(', '').str.replace(')', '')
```

```

#Applying country code as India
df=df[(df.country_code==1)]

#Making NCR List
ncr=["New Delhi","Ghaziabad","Noida","Gurgaon","Faridabad"]

#ncr calculation for all cuisines
cuisine_list_ncr=[]
for i in range(df.shape[0]):
    if df.iloc[i,3] in ncr:
        l=[]
        l.append(df.iloc[i,9])
        cuisine_list_ncr.append(l)
cuisines_ncr=[]
for i in cuisine_list_ncr:
    for j in i:
        l=j.split(",")
        for k in l:
            cuisines_ncr.append(k.strip())

#Creating a DataFrame for ncr
df1=pd.DataFrame(cuisines_ncr)

vc=df1[0].value_counts()
keys=vc.keys().tolist()
values=vc.tolist()
display(Markdown('*Delhi NCR Top 10 Cuisines Vs No. of Restaurants*'))
ncr_top_10_x=[]
ncr_top_10_y=[]
for i in range(10):
    print(keys[i],end=' ')
    ncr_top_10_x.append(keys[i])
    print(values[i])
    ncr_top_10_y.append(values[i])
pt.xlabel("Cuisines")
pt.ylabel("No. of Restaurants")
pt.xticks(rotation=90)
pt.bar(ncr_top_10_x,ncr_top_10_y)

```

```

pt.show()

#rest of india calculation
cuisine_list_non_ncr=[]
for i in range(df.shape[0]):
    if df.iloc[i,3] not in ncr:
        l=[]
        l.append(df.iloc[i,9])
        cuisine_list_non_ncr.append(l)

cuisines_non_ncr=[]
for i in cuisine_list_non_ncr:
    for j in i:
        l=j.split(",")
        for k in l:
            cuisines_non_ncr.append(k.strip())

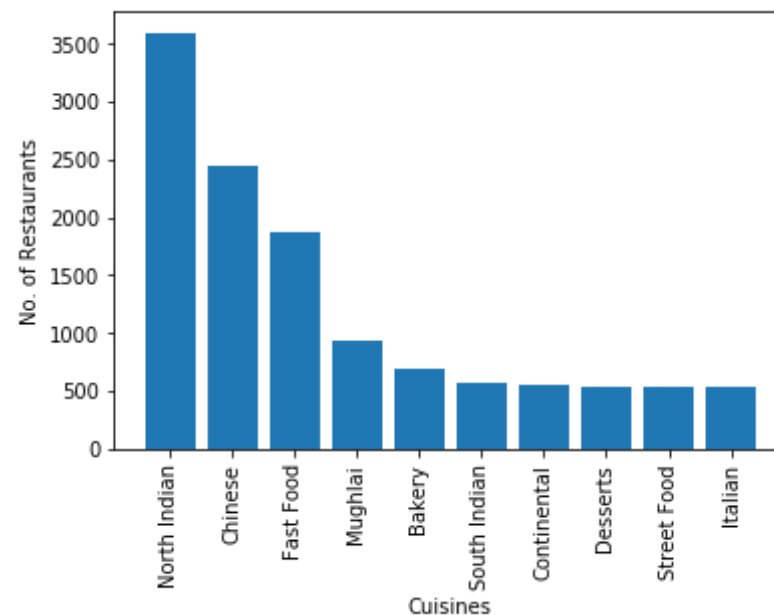
#Making data frame for Non NCR
df2=pd.DataFrame(cuisines_non_ncr)
vc=df2[0].value_counts()
keys=vc.keys().tolist()
values=vc.tolist()
non_ncr_top_10_x=[]
non_ncr_top_10_y=[]

display(Markdown('*Non Delhi NCR Top 10 Cuisines Vs No. of Restaurants*'))
for i in range(10):
    print(keys[i],end=' ')
    non_ncr_top_10_x.append(keys[i])
    print(values[i])
    non_ncr_top_10_y.append(values[i])
pt.xlabel("Cuisines")
pt.ylabel("No. of Restaurants")
pt.xticks(rotation=90)
pt.bar(non_ncr_top_10_x,non_ncr_top_10_y)
pt.show()

```

*Delhi NCR Top 10 Cuisines Vs No. of Restaurants*

North Indian 3597  
Chinese 2448  
Fast Food 1866  
Mughlai 933  
Bakery 697  
South Indian 569  
Continental 547  
Desserts 542  
Street Food 538  
Italian 535

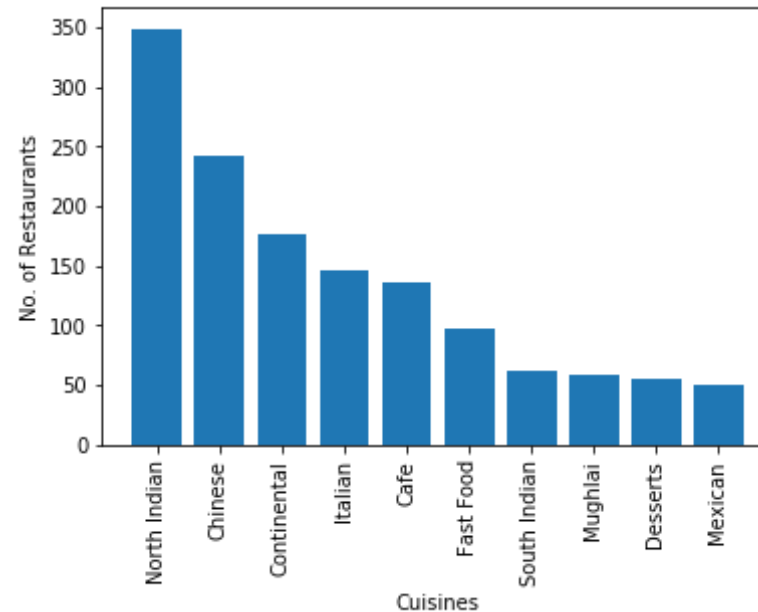


*Non Delhi NCR Top 10 Cuisines Vs No. of Restaurants*

North Indian 349  
Chinese 242  
Continental 177  
Italian 147  
Cafe 136  
Fast Food 97  
South Indian 62  
Mughlai 59



Mughlai 55  
Desserts 55  
Mexican 50



In [ ]:

Que1-Part4) 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.

In [4]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from IPython.display import display, Markdown
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()

#Fixing columns with space in name
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').s
tr.replace('(', '').str.replace(')', '')
```

```

#Applying country code as India
df=df[(df.country_code==1)]

#Making NCR List
ncr=["New Delhi","Ghaziabad","Noida","Gurgaon","Faridabad"]

#Creating Dictionary for Cuisines Served in NCR
d1={}
for i in range(df.shape[0]):
    key=df.iloc[i,3]
    value=df.iloc[i,9]
    if key in ncr:
        if key in d1:
            d1[key]=d1[key]+","+value
        else:
            d1[key]=value
#Creating a ncr list out of d1 and removing duplicates
l=[]
for i in d1:
    x=d1[i].split(",")
    for j in x:
        l.append(j.strip())
s=set(l)
l=list(s)
ncr_cuisine_list=l

#Creating Dictionary for Cuisines Served in non NCR area
d2={}
for i in range(df.shape[0]):
    key=df.iloc[i,3]
    value=df.iloc[i,9]
    if key not in ncr:
        if key in d2:
            d2[key]=d2[key]+","+value
        else:
            d2[key]=value
#Creating a non ncr list out of d2 and removing duplicates
l=[]

```

```

for i in d2:
    x=d2[i].split(",")
    for j in x:
        l.append(j.strip())
s=set(l)
l=list(s)
non_ncr_cuisine_list=l

display(Markdown('*Cuisines present in NCR but not present in rest of the india*'))

for i in ncr_cuisine_list:
    if i not in non_ncr_cuisine_list:
        print(i,end=',')

print()
print()
display(Markdown('*Cuisines present in Rest of the India but not present in NCR*'))
for i in non_ncr_cuisine_list:
    if i not in ncr_cuisine_list:
        print(i,end=',')
print()
print()
display(Markdown('*Also from Que1-Part3 above, common cuisines among both region from the set of top 10 cuisines of both region along with the re share are-*'))
x1=[]
x2=[]
y1=[]
y2=[]
for i in range(len(ncr_top_10_x)):
    if ncr_top_10_x[i] in non_ncr_top_10_x:

        x1.append(ncr_top_10_x[i])
        y1.append(ncr_top_10_y[i])

display(Markdown('*NCR Region*'))
pt.xticks(rotation=40)

```

```

pt.bar(x1,y1)
pt.xlabel("Cuisines")
pt.ylabel("No. of Restaurants")
pt.show()
print()
for i in range(len(non_ncr_top_10_x)):
    if non_ncr_top_10_x[i] in ncr_top_10_x:
        x2.append(non_ncr_top_10_x[i])
        y2.append(non_ncr_top_10_y[i])
display(Markdown('*Non NCR Region*'))
pt.xticks(rotation=40)
pt.xlabel("Cuisines")
pt.ylabel("No. of Restaurants")
pt.bar(x2,y2)
pt.show()

```

*Cuisines present in NCR but not present in rest of the india*

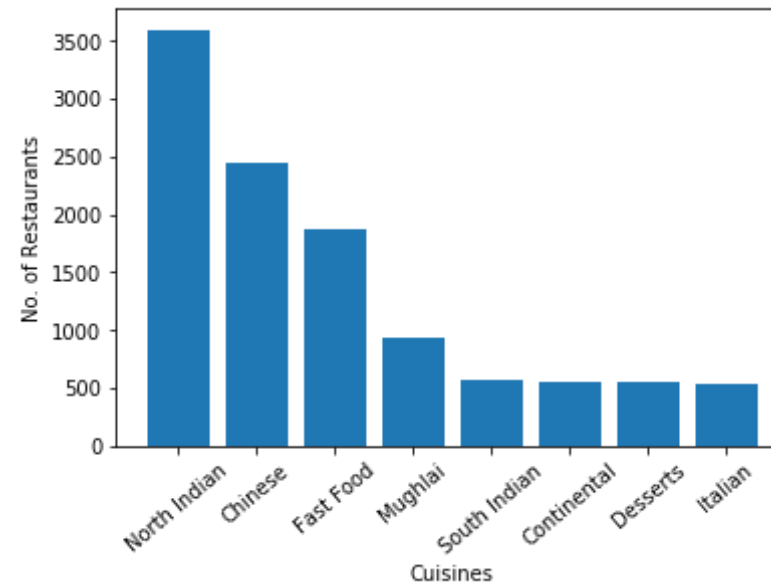
Assamese,Bihari,Pakistani,Persian,Raw Meats,Drinks Only,Deli,South American,Belgian,Nepalese,Iranian,Afghani,Naga,Cuisine Varies,Kashmiri,Sushi,Moroccan,Oriya,Turkish,Sri Lankan,

*Cuisines present in Rest of the India but not present in NCR*

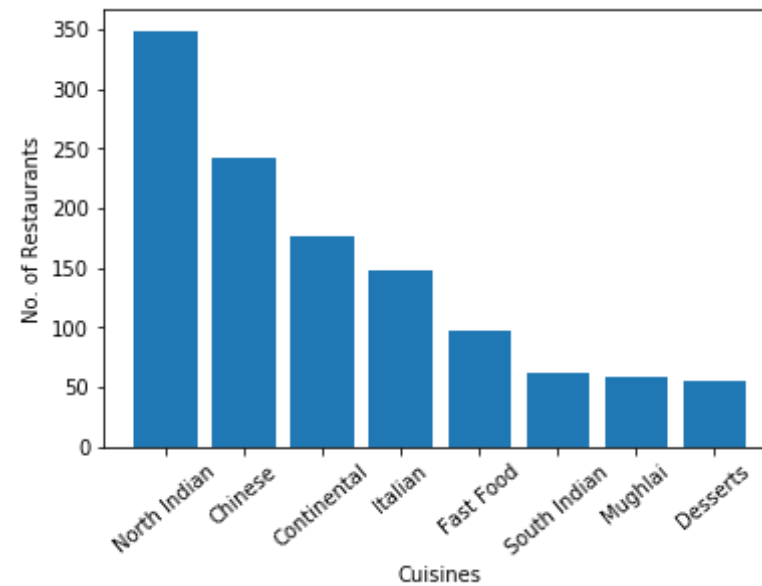
BBQ,Cajun,Malwani,German,

*Also from Que1-Part3 above, common cuisines among both region from the set of top 10 cuisines of both region along with there share are-*

*NCR Region*



#### *Non NCR Region*



In [ ]:

Que2) User Rating of a restaurant plays a crucial role in selecting a restaurant or ordering the food from the restaurant.

Que 2-Part1) Write a short detail analysis of how the rating is affected by restaurant due following features: Plot a suitable graph to explain your inference.

a) Number of Votes given Restaurant

```
In [18]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from IPython.display import display, Markdown
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()

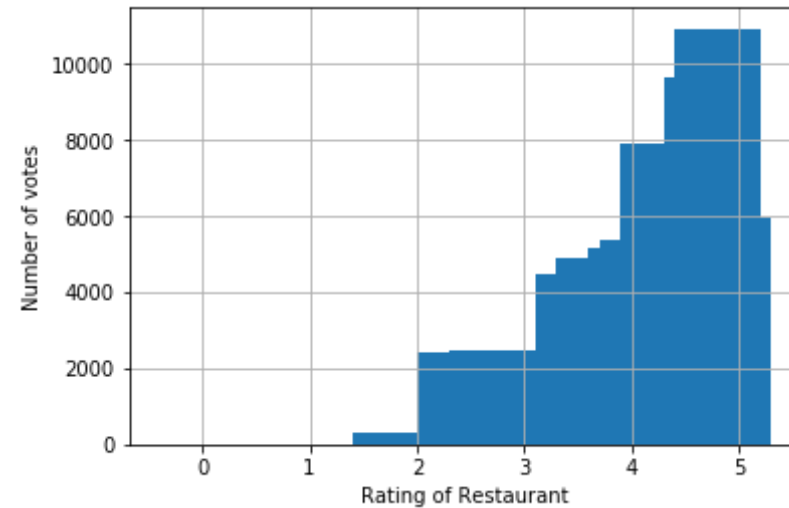
#Fixing columns with spaces
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').s
tr.replace('(', '').str.replace(')', '')

#Applying country code as India
df=df[(df.country_code==1)]

#Creating x and y list for plotting
x=[]
y=[]
for i in range(df.shape[0]):
    x.append(df.iloc[i,17])
    y.append(df.iloc[i,20])

#Plotting bar graph
plt.bar(x,y)
plt.grid()
plt.xlabel("Rating of Restaurant")
plt.ylabel("Number of votes")
plt.show()
```

```
display(Markdown('*Conclusion- As we can see the rating getting increased with votes, we can conclude that both entities are directly proportional*'))
```



*Conclusion- As we can see the rating getting increased with votes, we can conclude that both entities are directly proportional*

In [ ]:

Que2-Part1-b) Restaurant serving more number of cuisines.

```
In [19]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from IPython.display import display, Markdown
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()

#Fixing columns with spaces in name
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('(', '').str.replace(')', '')
```

```

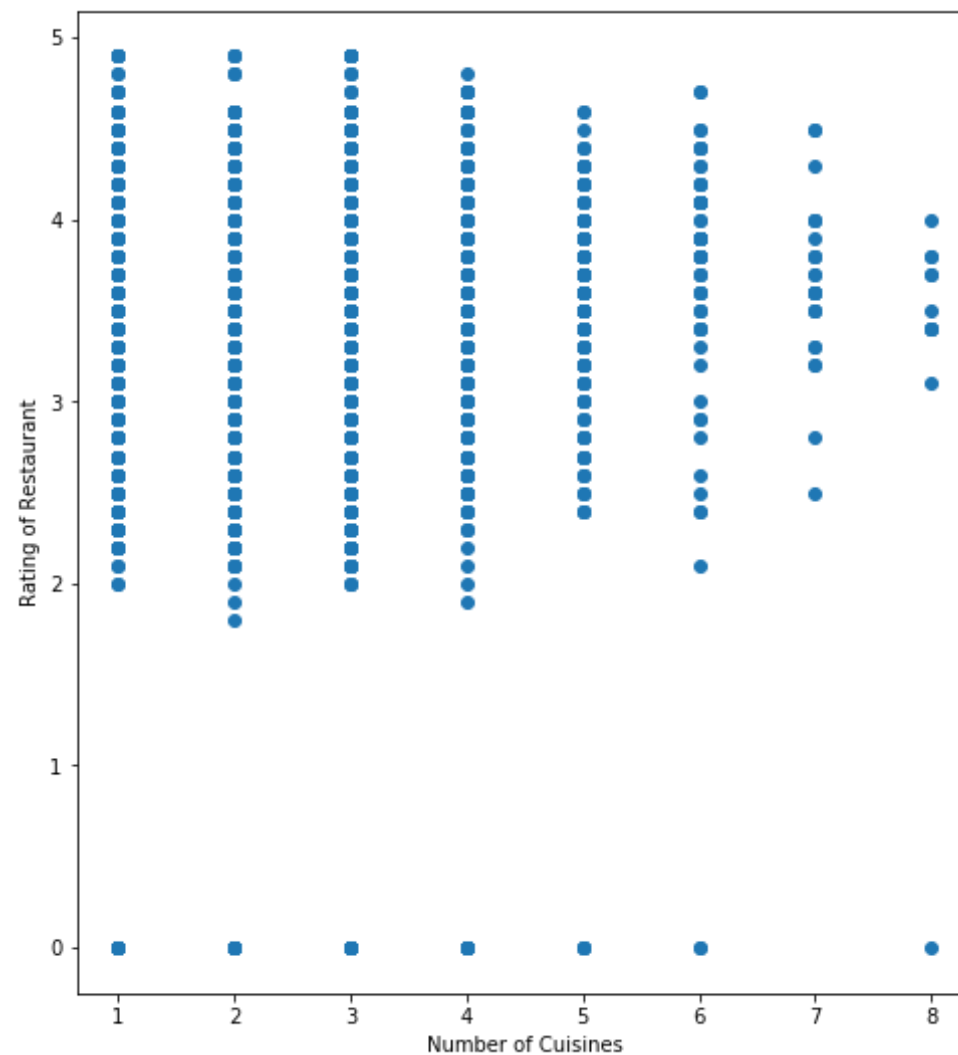
#Applying country code as India
df=df[(df.country_code==1)]

#Creating list for plotting
x=[]
y=[]
for i in range(df.shape[0]):
    s=str(df.iloc[i,9])

    x.append(len(s.split(",")))
    y.append(df.iloc[i,17])
pt.figure(figsize=(8,9))
pt.scatter(x,y)
pt.xlabel("Number of Cuisines")
pt.ylabel("Rating of Restaurant")
pt.show()
display(Markdown('*Conclusion- As we can see that highest rated restaurants have less number of cuisines in comparison to restaurants having more cuisines, we can conclude that rating is somewhat inversely proportional here with number of cuisines*'))

```





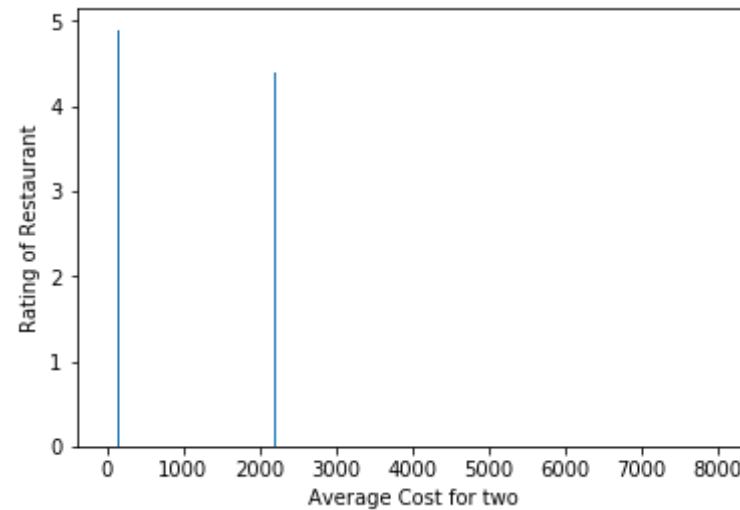
*Conclusion- As we can see that highest rated restaurants have less number of cuisines in comparison to restaurants having more cuisines, we can conclude that rating is somewhat inversely proportional here with number of cuisines*

In [ ]:

## Que2 Part 1-c) Average Cost of Restaurant

```
In [7]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from IPython.display import display, Markdown
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()
#Fixing columns
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').s
tr.replace('(', '').str.replace(')', '')
#Taking country code of India

df=df[(df.country_code==1)]
#Plotting with x,y list
x=[]
y=[]
for i in range(df.shape[0]):
    x.append(df.iloc[i,10])
    y.append(df.iloc[i,17])
plt.bar(x,y)
plt.xlabel("Average Cost for two")
plt.ylabel("Rating of Restaurant")
plt.show()
display(Markdown('*Inference- Highest rated restaurants have less avera
ge cost in comparison to resturants having more average cost*'))
```



*Inference- Highest rated restaurants have less average cost in comparison to restaurants having more average cost*

In [ ]:

Que 2- Part 1-d) Restaurant serving some specific cuisines.

```
In [8]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from IPython.display import display, Markdown
import numpy as np
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()

#Fixing columns with spaces in name
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').s
tr.replace('(', '').str.replace(')', '')
#Applying country code as India
df=df[(df.country_code==1)]
```

```

#Creating dictionary for cuisines vs Rating of restuarant serving the same
d={}
for i in range(df.shape[0]):
    key=df.iloc[i,9]
    value=df.iloc[i,17]
    if key in d:
        d[key]=str(d[key])+","+str(value)
    else:
        d[key]=str(value)

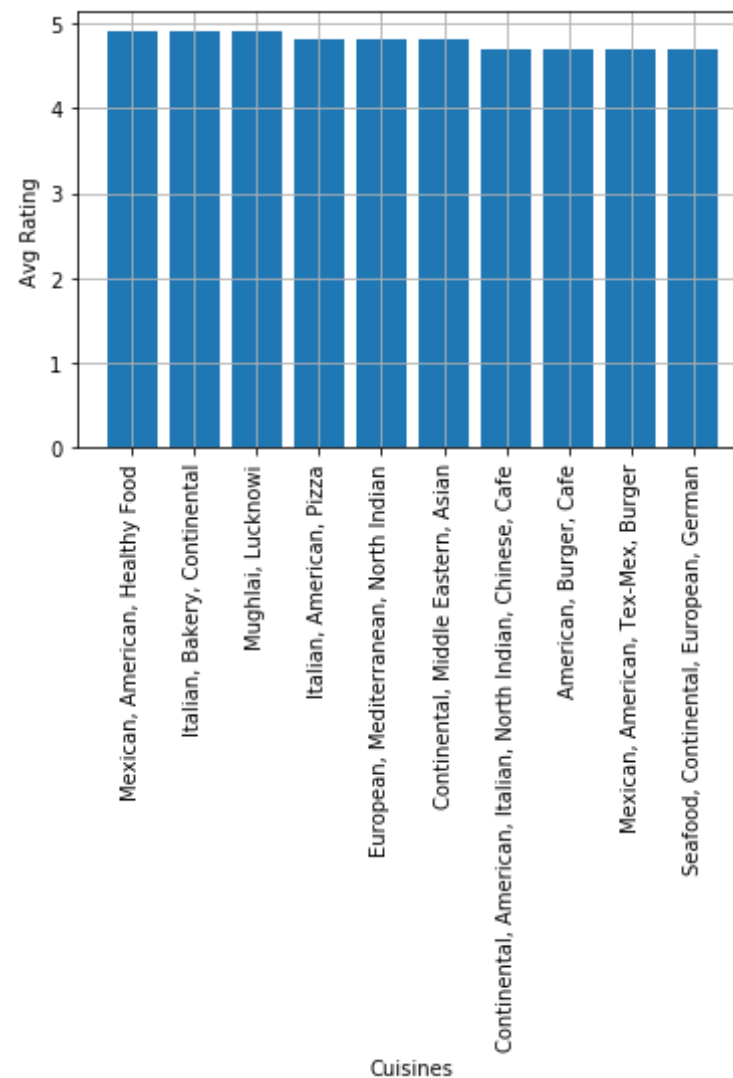
def Average(lst):
    return round(sum(lst) / len(lst),2)
#Finding average rating of each restaurant for cuisines in dictionary
d
for i in d:
    l=d[i].split(",")
    for j in range(len(l)):
        l[j]=l[j].strip()
        l[j]=float(l[j])
    d[i]=Average(l)

#Finding Top 10 Cuisines with maximum average rating of restaurants serving them
#Sorting the final dictionary in reverse order and fetching 10 records
counter=0
x1=[]
y1=[]
for key, value in sorted(d.items(), key=lambda item: item[1],reverse=True):
    counter=counter+1
    x1.append(key)
    y1.append(value)
    if counter==10:
        break
pt.bar(x1,y1)
pt.xlabel("Cuisines")
pt.ylabel("Avg Rating")

```

```
pt.grid()
pt.xticks(rotation=90)
pt.show()
print()
print()

display(Markdown('*Conclusion- We can infer that serving above dishes l
eads to highest rating for resturants*'))
```



*Conclusion- We can infer that serving above dishes leads to highest rating for restaurants*

In [ ]:

Que 2- Part2-a)Find the weighted restaurant rating of each locality and find out the top 10 localities with more weighted restaurant rating?

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

```
In [9]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from IPython.display import display, Markdown
import numpy as np
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()

#Fixing columns with spaces in name
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('(', '').str.replace(')', '')
#Applying country code as India
df=df[(df.country_code==1)]
#Dictionary1 - Locality vs Ratings of restaurant
d1={}
for i in range(df.shape[0]):
    l=[df.iloc[i,17]]
    key=df.iloc[i,5]
    value=l
    if key in d1:
        d1[key]=d1[key]+value
    else:
        d1[key]=value
#Dictionary2- Locality vs Votes of restaurant
d2={}
for i in range(df.shape[0]):
    l=[df.iloc[i,20]]
    key=df.iloc[i,5]
    value=(l)
    if key in d2:
        d2[key]=d2[key]+value
    else:
        d2[key]=value
```

```

#Weighted Rating calculation and making dictionary of Locality to Weighted Rating
for i in d1:
    key=i
    vr=0
    tv=0
    for j in range(len(d1[key])):
        vr=vr+d1[key][j]*d2[key][j]
        tv=tv+d2[key][j]

    if(tv!=0):
        wr=vr/tv
    else:
        wr=0
    d2[key]=wr

#Sorting the final dictionary in reverse order and fetching 10 records
counter=0
x=[]
y=[]
display(Markdown('*Top 10 localities with weighted rating in India*'))
for key, value in sorted(d2.items(), key=lambda item: item[1], reverse=True):
    counter=counter+1
    print("%s: %s" %(key, "%0.3f"%(value)))
    x.append(key)
    y.append(value)
    if counter==10:
        break
pt.bar(x,y)
pt.xlabel("Locality")
pt.ylabel("Weighted Rating")
pt.xticks(rotation=90)
pt.show()

```

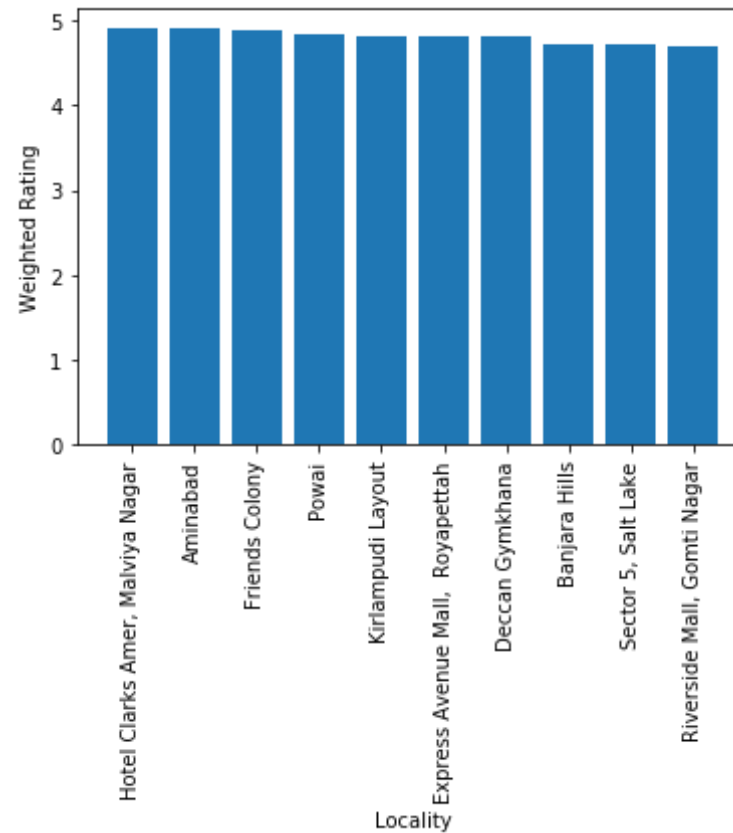
*Top 10 localities with weighted rating in India*

Hotel Clarks Amer, Malviya Nagar: 4.900  
Aminabad: 4.900

Friends Colony: 4.887  
Devi: 4.842



Powai: 4.842  
Kirlampudi Layout: 4.820  
Express Avenue Mall, Royapettah: 4.800  
Deccan Gymkhana: 4.800  
Banjara Hills: 4.719  
Sector 5, Salt Lake: 4.707  
Riverside Mall, Gomti Nagar: 4.700



In [ ]:

Que3-Part1)Plot the bar graph top 15 restaurants have a maximum number of outlets.

```

In [10]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from IPython.display import display, Markdown
import numpy as np
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()

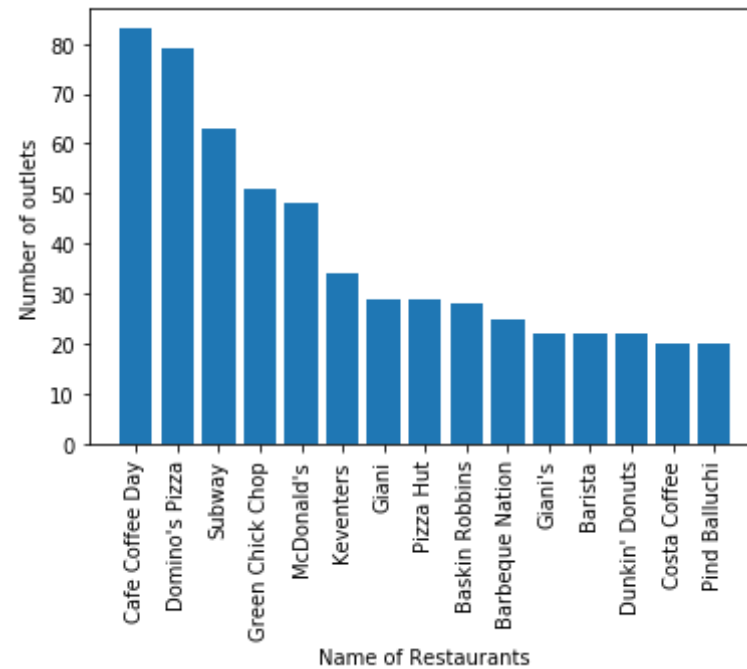
#Fixing columns with spaces in name
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').s
tr.replace('(', '').str.replace(')', '')
#Applying country code as India
df=df[(df.country_code==1)]

#Getting value Counts
vc=df.restaurant_name.value_counts()

keys=vc.keys().tolist()
values=vc.tolist()
display(Markdown('*Top 15 retaurants in India with respect to number of
outlets*'))
#Making blank lists for plotting
x=[]
y=[]
for i in range(15):
    x.append(keys[i])
    y.append(values[i])
#Plotting the bar graph
plt.xlabel("Name of Restaurants")
plt.ylabel("Number of outlets")
plt.bar(x,y)
plt.xticks(rotation=90)
plt.show()

```

*Top 15 retaurants in India with respect to number of outlets*



In [ ]:

Que3-Part2)Plot the histogram of aggregate rating of restaurant( drop the unrated restaurant).

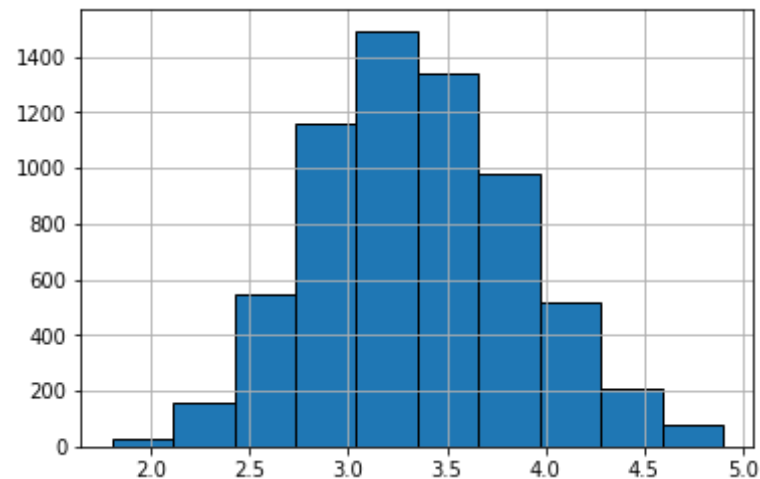
```
In [11]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from IPython.display import display, Markdown
import numpy as np
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()

#Fixing columns with spaces in name
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').s
tr.replace('(', '').str.replace(')', '')
#Applying country code as India
```

```
df=df[(df.country_code==1)]
display(Markdown('*Aggregate rating of restuarants in India without con
sidering unrated restaurants*'))
x=[]
for i in range(df.shape[0]):
    if df.iloc[i,17]!=0:
        x.append(df.iloc[i,17])
pt.grid()
pt.hist(x,edgecolor='black')

pt.show()
```

*Aggregate rating of restuarants in India without considering unrated restaurants*



In [ ]:

Que3-Part3)Plot the bar graph top 10 restaurants in the data with the highest number of votes.

```
In [12]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```

from IPython.display import display, Markdown
import numpy as np
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()

#Fixing columns with spaces in name
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').s
tr.replace('(', '').str.replace(')', '')
#Applying country code as India
df=df[(df.country_code==1)]

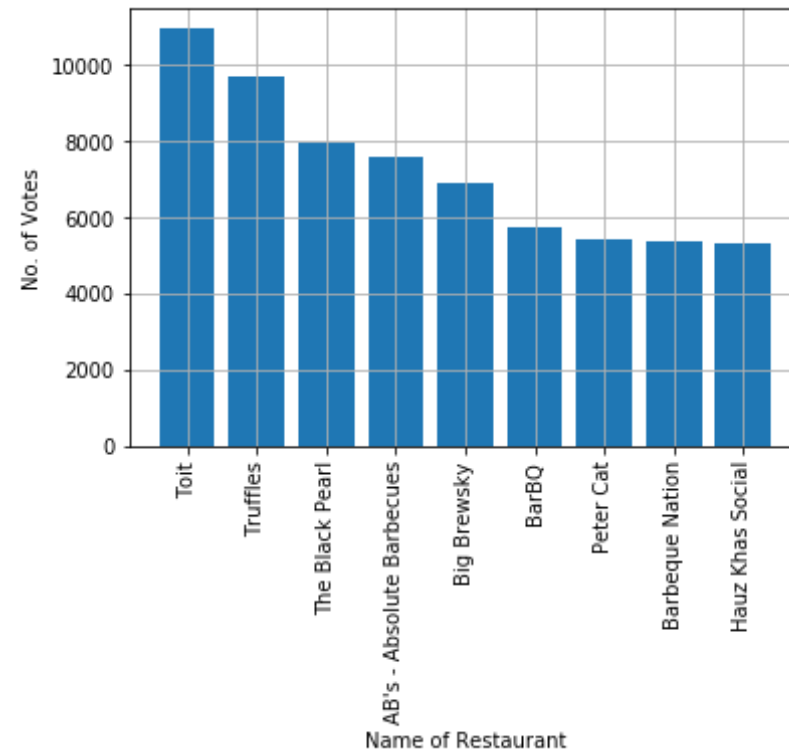
#Making dictionary of keys as Restaurant Name and values as Number of v
otes
d={}
for i in range(df.shape[0]):
    key=df.iloc[i,0]
    value=df.iloc[i,20]
    if key in d:
        d[key]=d[key]+value
    else:
        d[key]=value

#Sorting the dictionary in reverse order and fetching 10 records
counter=0
x=[]
y=[]
for key, value in sorted(d.items(), key=lambda item: item[1],reverse=Tr
ue):
    counter=counter+1
    x.append(key)
    y.append(value)
    if counter==10:
        break
display(Markdown('*Top 10 Restaurants in India with maximum number of v
otes*'))
z=[]
for i in range(df.shape[0]):
    if df.iloc[i,0] in x:
        z.append(df.iloc[i,1])

```

```
pt.xticks(rotation=90)
pt.bar(z,y)
pt.grid()
pt.xlabel("Name of Restaurant")
pt.ylabel("No. of Votes")
pt.show()
```

*Top 10 Restaurants in India with maximum number of votes*



In [ ]:

Que3-Part4)Plot the pie graph of top 10 cuisines present in restaurants in the USA.

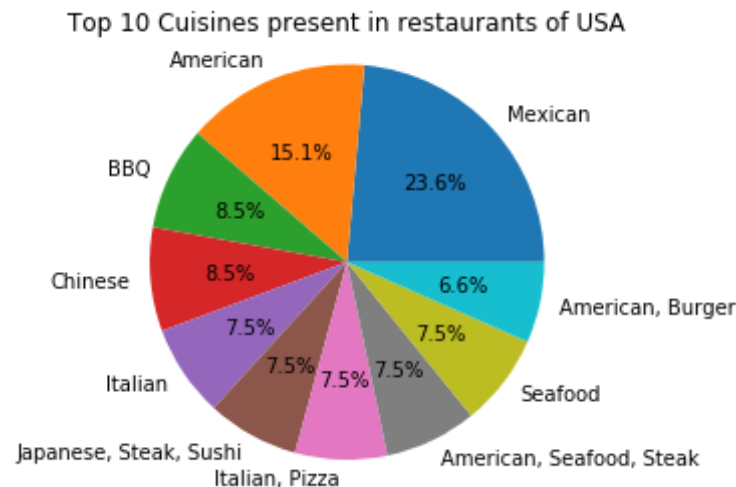
```
In [13]: import pandas as pd
import matplotlib.pyplot as plt
```

```

import numpy as np
from IPython.display import display, Markdown
import numpy as np
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()

#Fixing columns with spaces in name
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('(', '').str.replace(')', '')
#Applying country code as USA
df=df[(df.country_code==216)]
vc=df.cuisines.value_counts()
keys=vc.keys().tolist()
values=vc.tolist()
labels=[]
sizes=[]
for i in range(10):
    labels.append(keys[i])
    sizes.append(values[i])
pt.title("Top 10 Cuisines present in restaurants of USA")
pt.pie(sizes,labels=labels,autopct="%.1f%%")
pt.axis("equal")
pt.show()

```



In [ ]:

Que3-Part5)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.

```
In [14]: import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
from IPython.display import display, Markdown
import numpy as np
iris=pd.read_csv("zomato.csv",encoding="iso-8859-1")
df=iris.copy()

#Fixing columns with spaces in name
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_').str.replace('(', '').str.replace(')', '')
#Applying country code as India
df=df[(df.country_code==1)]
#Dictionary1 - City vs Ratings of restaurant
d1={}
for i in range(df.shape[0]):
    l=df.iloc[i,17]
    key=df.iloc[i,3]
    value=l
    if key in d1:
        d1[key]=d1[key]+value
    else:
        d1[key]=value
#Dictionary2- City vs Votes of restaurant
d2={}
for i in range(df.shape[0]):
    l=df.iloc[i,20]
    key=df.iloc[i,3]
    value=l
    if key in d2:
        d2[key]=d2[key]+value
    else:
```



```

        d2[key]=value

#Weighted Rating calculation and making dictionary(d2) of City to Weighted Rating
for i in d1:
    key=i
    vr=0
    tv=0
    for j in range(len(d1[key])):
        vr=vr+d1[key][j]*d2[key][j]
        tv=tv+d2[key][j]

    if(tv!=0):
        wr=vr/tv
    else:
        wr=0
    d2[key]=wr

#Now getting number of resturants in a City(d3)
d3={}
for i in range(df.shape[0]):
    key=df.iloc[i,3]
    if key in d3:
        d3[key]=d3[key]+1
    else:
        d3[key]=1

x=[] #CityName
y=[] #Number Of resturants
z=[] #Weighted Rating
#Now d2 has data of City VS Weighted Rating and d3 has data of City Vs Number of Restaurants
counter=0
for i in d2:
    for j in d3:
        if(i==j):
            counter=counter+1
            x.append(i)
            y.append(d3[j])
            z.append(d2[i])

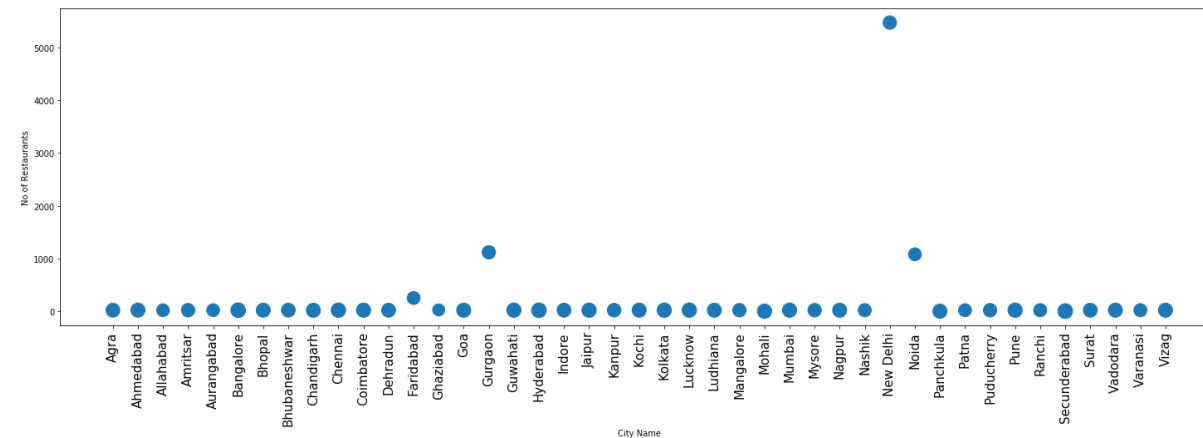
```

```
display(Markdown('*Bubble graph of number of Restaurants present in the  
cities of India and keeping the weighted restaurant rating of the city  
in a bubble.*'))
```

```
y=np.array(y)  
z=np.array(z)  
pt.figure(figsize=(25,7))  
pt.xticks(fontsize=15,rotation=90)
```

```
pt.xlabel("City Name")  
pt.ylabel("No of Restaurants")  
pt.scatter(x,y,s=(z*70))  
pt.plot()  
pt.show()
```

*Bubble graph of number of Restaurants present in the cities of India and keeping the weighted restaurant rating of the city in a bubble.*



In [ ]: