Introduction

New York City's demographics show that it is a large and ethnically diverse metropolis. It is the largest city in the United States with a long history of international immigration. New York City was home to nearly 8.5 million people in 2014, accounting for over 40% of the population of New York State and a slightly lower percentage of the New York metropolitan area, home to approximately 23.6 million. Over the last decade the city has been growing faster than the region. The New York region continues to be by far the leading metropolitan gateway for legal immigrants admitted into the United States. Throughout its history, New York City has been a major point of entry for immigrants; the term "melting pot" was coined to describe densely populated immigrant neighborhoods on the Lower East Side. As many as 800 languages are spoken in New York, making it the most linguistically diverse city in the world. English remains the most widely spoken language, although there are areas in the outer boroughs in which up to 25% of people speak English as an alternate language, and/or have limited or no English language fluency. English is least spoken in neighborhoods such as Flushing, Sunset Park, and Corona. With its diverse culture , comes diverse food items. There are many restaurants in NewYork City, each belonging to different categories like Chinese , Indian , French etc. So as part of this project , we will list and visualize all major parts of New York City that has great indian restaurants.

Data

For this project we need the following data : New York City data that contains list Boroughs, Neighborhoods along with their latitude and longitude.
Data source : https://cocl.us/new_york_dataset
Description : This data set contains the required information. And we will use this data set to explore various neighborhoods of New York city.

Indian restaurants in each neighborhood of New York city.
Data source : Fousquare API
Description : By using this api we will get all the venues in each neighborhood. We can filter these venues to get only indian restaurants.

GeoSpace data Data source :
https://data.cityofnewyork.us/City-Government/Borough-Boundaries/tqmj-j8zm

Description : By using this geo space data we will get the New york Borough boundaries that will help us visualize choropleth map.

Questions that can be asked using the above mentioned datasets

What is the best location in New York City for Indian Cuisine ?

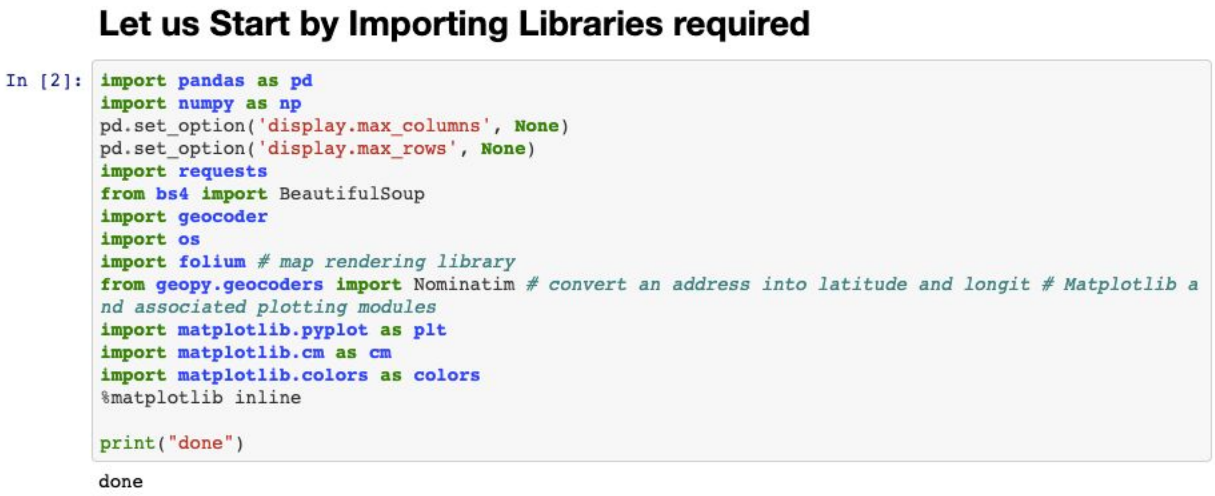
Which areas have potential Indian Restaurant Market ?

Which all areas lack Indian Restaurants ?

Which is the best place to stay if I prefer Indian Cuisine ?

Target Audience: People looking to open a Indian restaurant in New York or looking for a Indian place to eat at.

Let's discuss the Analysis

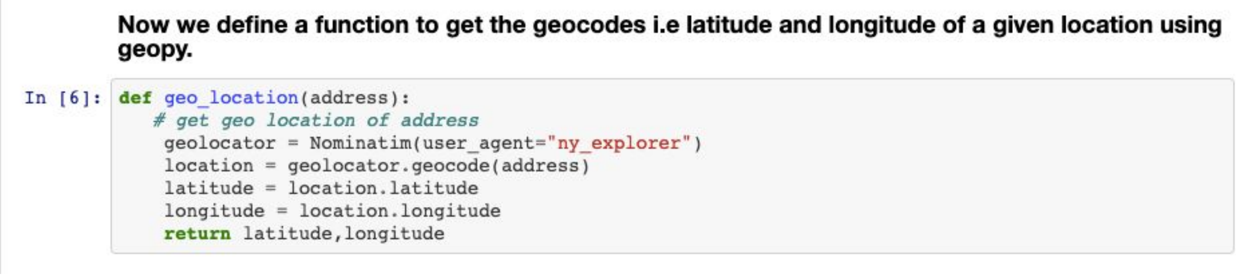


Let us Start by Importing Libraries required

In [2]:
```
import pandas as pd
import numpy as np
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
import requests
from bs4 import BeautifulSoup
import geocoder
import os
import folium # map rendering library
from geopy.geocoders import Nominatim # convert an address into latitude and longit # Matplotlib a
nd associated plotting modules
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors
%matplotlib inline

print("done")
```

```
done
```

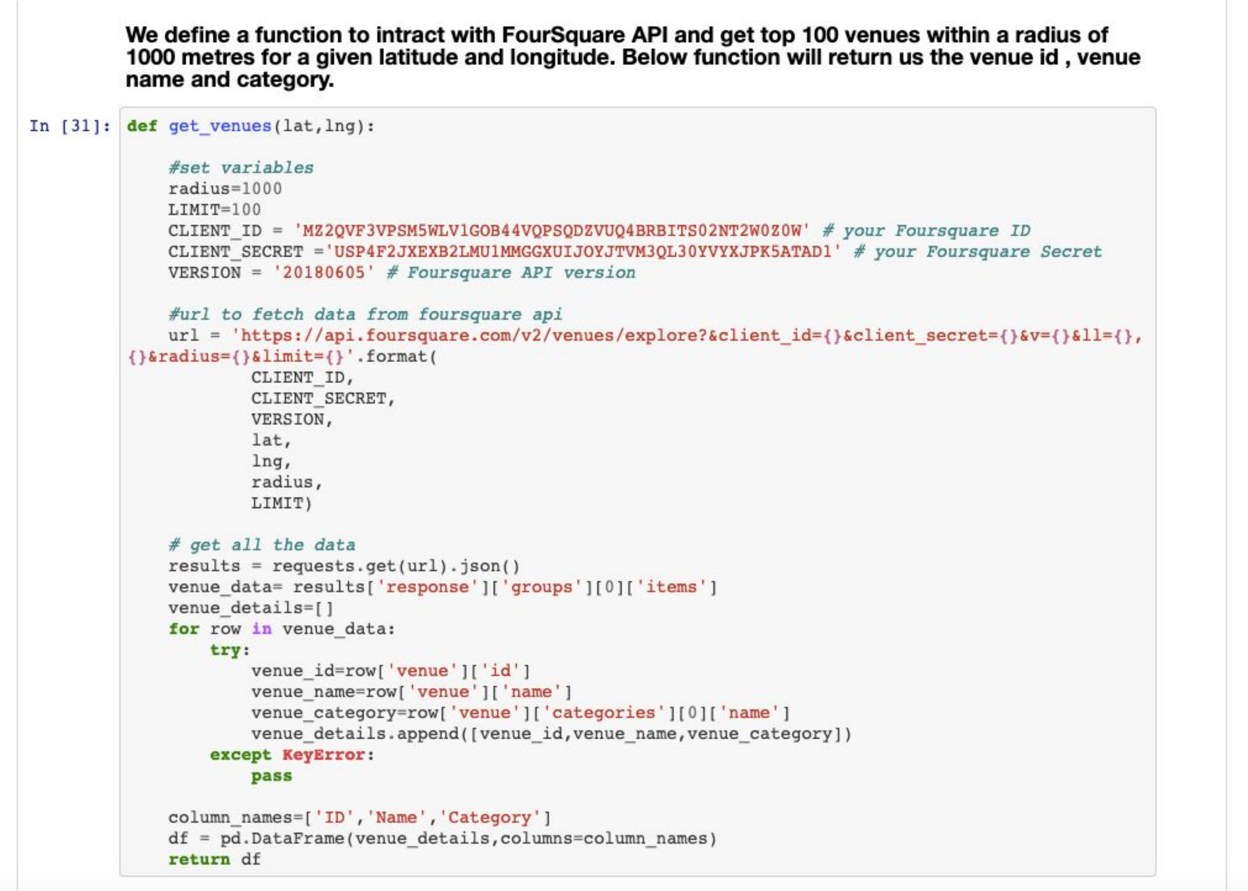
For any Analysis we would need certain Libraries. For the Analysis of Indian Restaurants in NY these libraries were required.



Now we define a function to get the geocodes i.e latitude and longitude of a given location using geopy.

```
In [6]: def geo_location(address):
           # get geo location of address
            geolocator = Nominatim(user_agent="ny_explorer")
            location = geolocator.geocode(address)
            latitude = location.latitude
            longitude = location.longitude
            return latitude,longitude
```

Here I created a function to get the Longitude and Latitude values of a certain address by using Geopy Library.

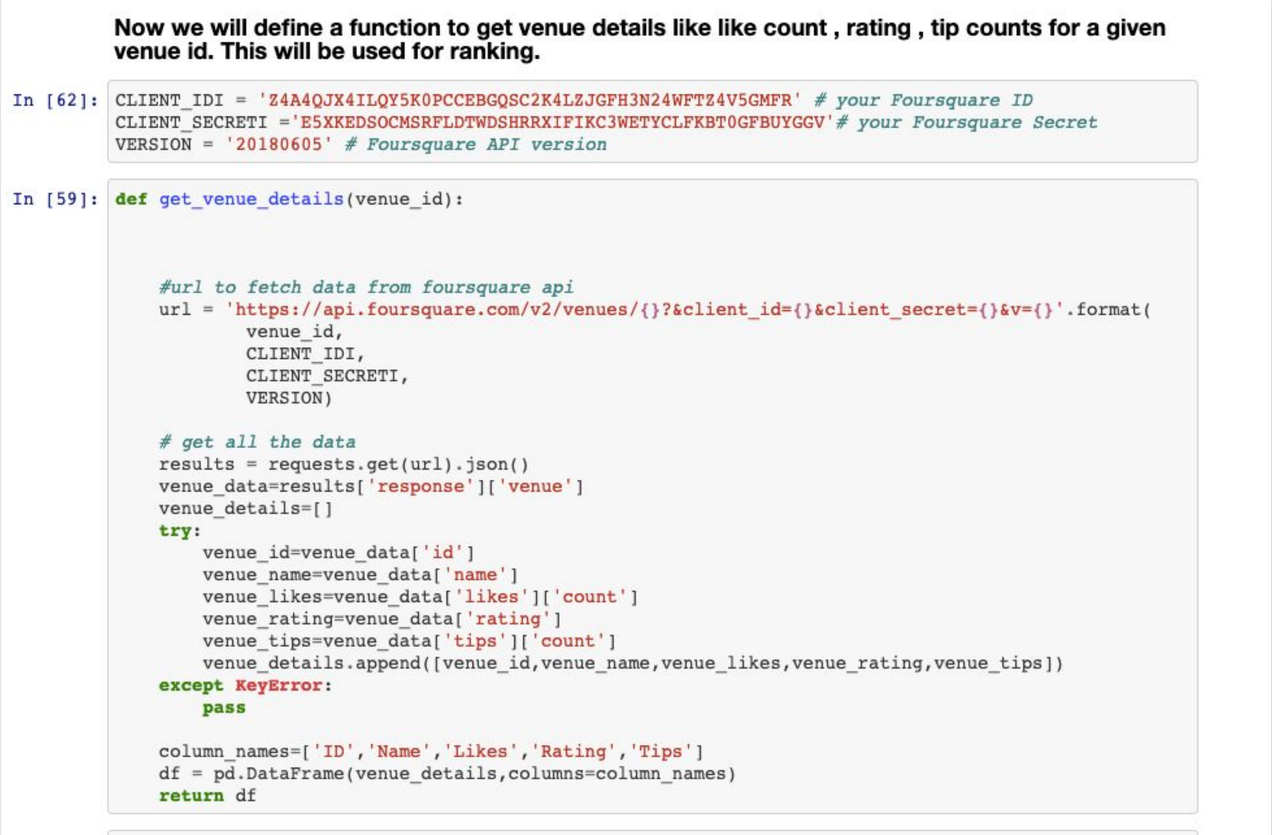


We define a function to intract with FourSquare API and get top 100 venues within a radius of 1000 metres for a given latitude and longitude. Below function will return us the venue id , venue name and category.

```
In [31]: def get_venues(lat,lng):

             #set variables
             radius=1000
             LIMIT=100
             CLIENT_ID = 'MZ2QVF3VPSM5WLV1GOB44VQPSQDZVUQ4BRBITS02NT2W0Z0W' # your Foursquare ID
             CLIENT_SECRET ='USP4F2JXEXB2LMU1MMGGXUIJOYJTVM3QL30YVYXJPK5ATAD1' # your Foursquare Secret
             VERSION = '20180605' # Foursquare API version

             #url to fetch data from foursquare api
             url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},
         {}&radius={}&limit={}'.format(
                     CLIENT_ID,
                     CLIENT_SECRET,
                     VERSION,
                     lat,
                     lng,
                     radius,
                     LIMIT)

             # get all the data
             results = requests.get(url).json()
             venue_data= results['response']['groups'][0]['items']
             venue_details=[]
             for row in venue_data:
                 try:
                     venue_id=row['venue']['id']
                     venue_name=row['venue']['name']
                     venue_category=row['venue']['categories'][0]['name']
                     venue_details.append([venue_id,venue_name,venue_category])
                 except KeyError:
                     pass

             column_names=['ID','Name','Category']
             df = pd.DataFrame(venue_details,columns=column_names)
             return df
```

Here I created a function which will return top 100 visited venues which will fall under the radius of 1000 metres from a given Latitude and Longitude values.

The function will return The Venue_id, Venue_name abd Category which will be later stored in a dataframe for proper processing, cleaning and analysis.

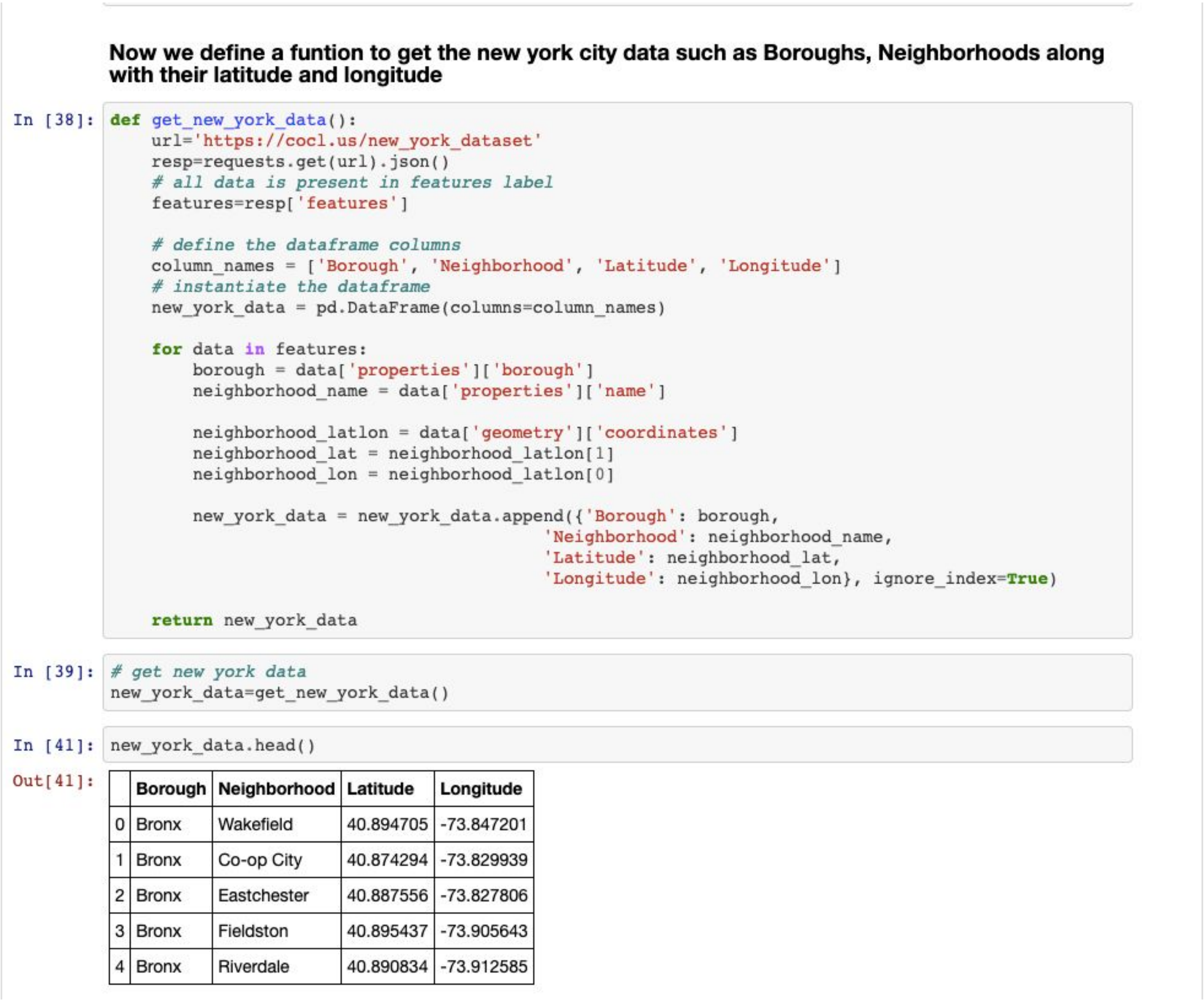


Now we will define a function to get venue details like like count , rating , tip counts for a given venue id. This will be used for ranking.

```
In [62]: CLIENT_IDI = 'Z4A4QJX4ILQY5K0PCCEBGQSC2K4LZJGFH3N24WFTZ4V5GMFR' # your Foursquare ID
         CLIENT_SECRETI ='E5XKEDSOCMSRFLDTWDSHRRXIFIKC3WETYCLFKBT0GFBUYGGV'# your Foursquare Secret
         VERSION = '20180605' # Foursquare API version
```

```
In [59]: def get_venue_details(venue_id):



             #url to fetch data from foursquare api
             url = 'https://api.foursquare.com/v2/venues/{}?&client_id={}&client_secret={}&v={}'.format(
                     venue_id,
                     CLIENT_IDI,
                     CLIENT_SECRETI,
                     VERSION)

             # get all the data
             results = requests.get(url).json()
             venue_data=results['response']['venue']
             venue_details=[]
             try:
                 venue_id=venue_data['id']
                 venue_name=venue_data['name']
                 venue_likes=venue_data['likes']['count']
                 venue_rating=venue_data['rating']
                 venue_tips=venue_data['tips']['count']
                 venue_details.append([venue_id,venue_name,venue_likes,venue_rating,venue_tips])
             except KeyError:
                 pass

             column_names=['ID','Name','Likes','Rating','Tips']
             df = pd.DataFrame(venue_details,columns=column_names)
             return df
```

Here I have created another function which will fetch me details like Count, Rating, Tips for a given value_id from the Database of Foursquare and these values too will be stored in a DataFrame for further Analysis.



Now we define a funtion to get the new york city data such as Boroughs, Neighborhoods along with their latitude and longitude

In [38]:

```
def get_new_york_data():
    url='https://cocl.us/new_york_dataset'
    resp=requests.get(url).json()
    # all data is present in features label
    features=resp['features']

    # define the dataframe columns
    column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']
    # instantiate the dataframe
    new_york_data = pd.DataFrame(columns=column_names)

    for data in features:
        borough = data['properties']['borough']
        neighborhood_name = data['properties']['name']

        neighborhood_latlon = data['geometry']['coordinates']
        neighborhood_lat = neighborhood_latlon[1]
        neighborhood_lon = neighborhood_latlon[0]

        new_york_data = new_york_data.append({'Borough': borough,
                                          'Neighborhood': neighborhood_name,
                                          'Latitude': neighborhood_lat,
                                          'Longitude': neighborhood_lon}, ignore_index=True)

    return new_york_data
```

In [39]:

```
# get new york data
new_york_data=get_new_york_data()
```

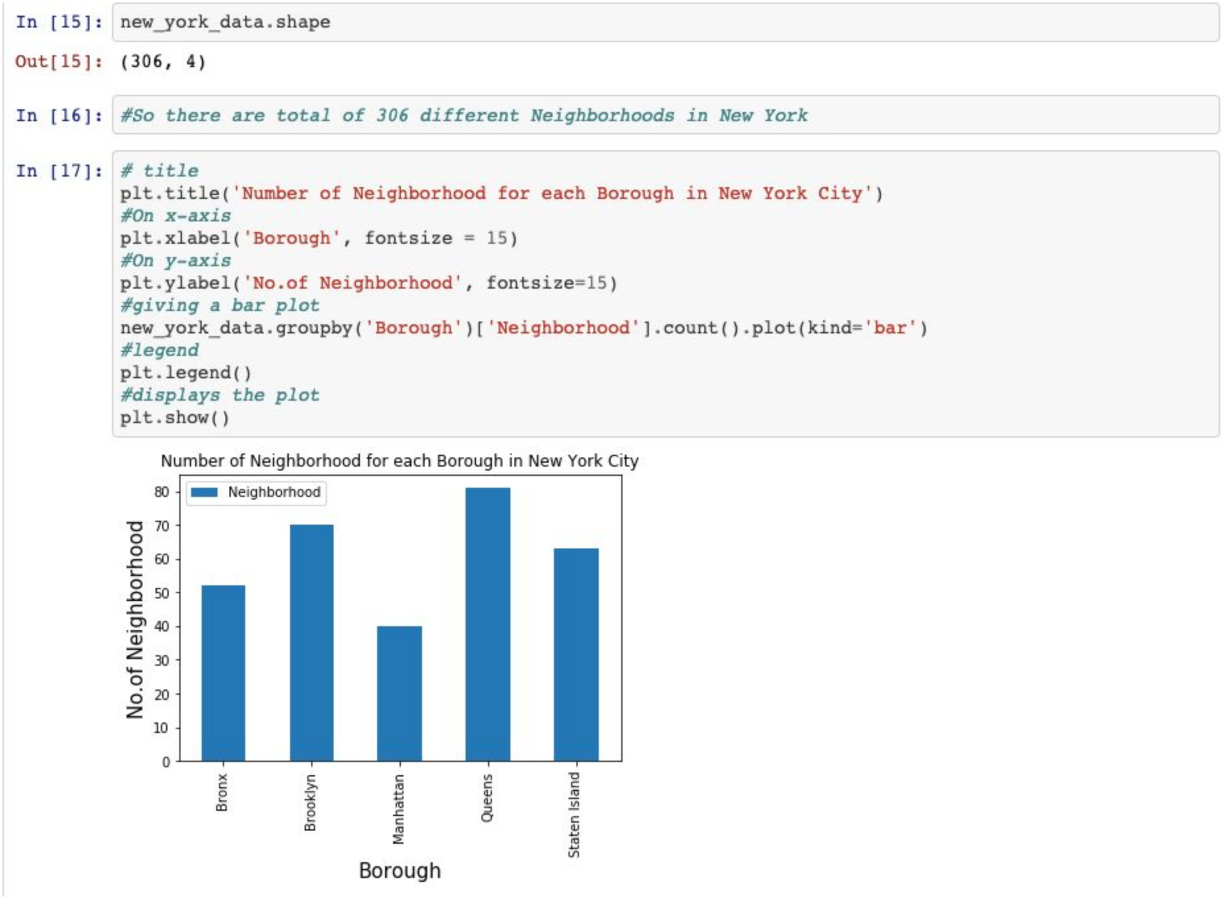
In [41]:

```
new_york_data.head()
```

Out[41]:

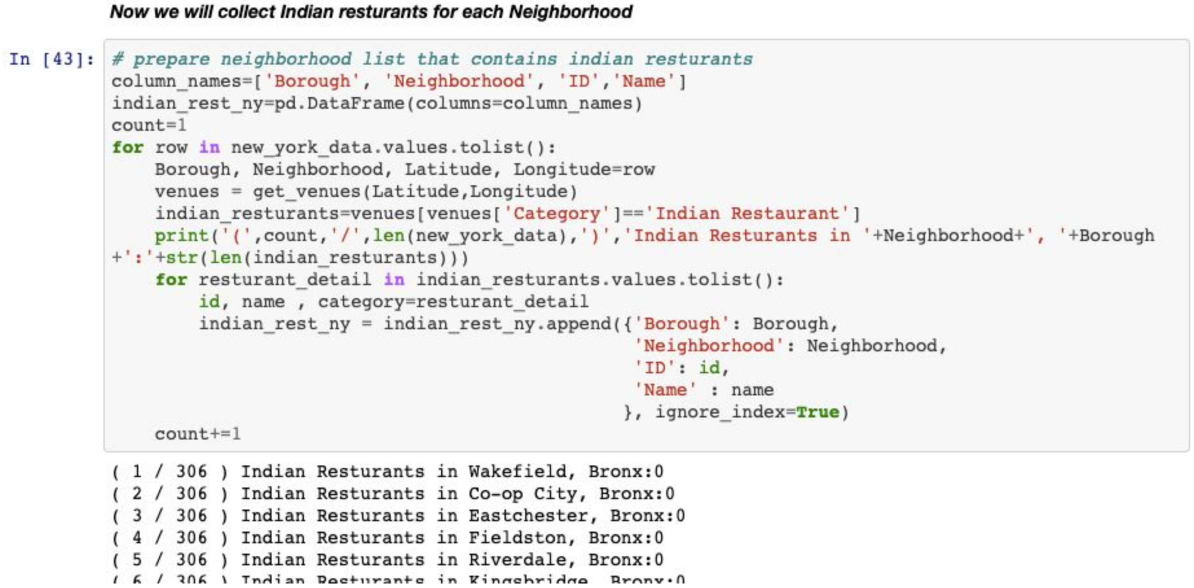
	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585

As we know that to get the details from FourSquare DataBase we would require the longitude and latitude values so we create a function to fetch the values for Neighborhood and Boroughs present in the dataset of Url and then these values are stored in a DataFrame and we have here seen the head of the dataframe as well.



```
In [15]: new_york_data.shape
Out[15]: (306, 4)

In [16]: #So there are total of 306 different Neighborhoods in New York

In [17]: # title
         plt.title('Number of Neighborhood for each Borough in New York City')
         #On x-axis
         plt.xlabel('Borough', fontsize = 15)
         #On y-axis
         plt.ylabel('No.of Neighborhood', fontsize=15)
         #giving a bar plot
         new_york_data.groupby('Borough')['Neighborhood'].count().plot(kind='bar')
         #legend
         plt.legend()
         #displays the plot
         plt.show()
```

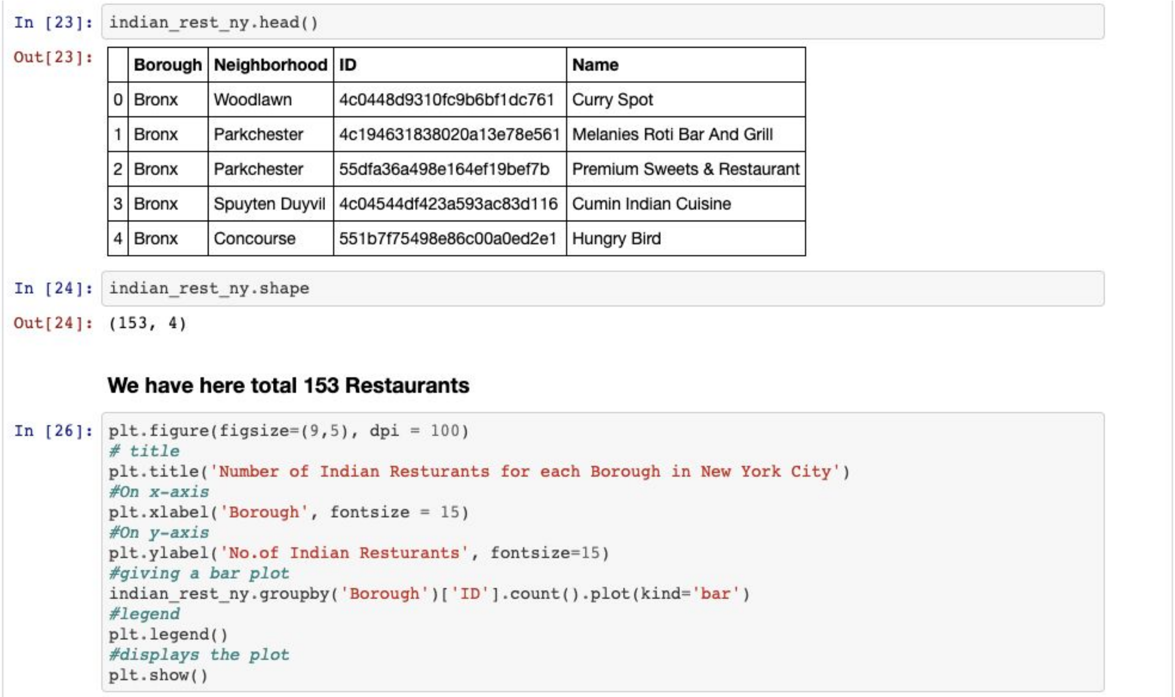
After Analysis of the DataFrame we found that there are overall 306 neighborhoods.



Now we will collect Indian resturants for each Neighborhood

```
In [43]: # prepare neighborhood list that contains indian resturants
         column_names=['Borough', 'Neighborhood', 'ID','Name']
         indian_rest_ny=pd.DataFrame(columns=column_names)
         count=1
         for row in new_york_data.values.tolist():
             Borough, Neighborhood, Latitude, Longitude=row
             venues = get_venues(Latitude,Longitude)
             indian_resturants=venues[venues['Category']=='Indian Restaurant']
             print('(',count,'/',len(new_york_data),')','Indian Resturants in '+Neighborhood+', '+Borough
         +':'+str(len(indian_resturants)))
             for resturant_detail in indian_resturants.values.tolist():
                 id, name , category=resturant_detail
                 indian_rest_ny = indian_rest_ny.append({'Borough': Borough,
                                                         'Neighborhood': Neighborhood,
                                                         'ID': id,
                                                         'Name' : name
                                                        }, ignore_index=True)
             count+=1

( 1 / 306 ) Indian Resturants in Wakefield, Bronx:0
( 2 / 306 ) Indian Resturants in Co-op City, Bronx:0
( 3 / 306 ) Indian Resturants in Eastchester, Bronx:0
( 4 / 306 ) Indian Resturants in Fieldston, Bronx:0
( 5 / 306 ) Indian Resturants in Riverdale, Bronx:0
( 6 / 306 ) Indian Resturants in Kingsbridge, Bronx:0
```

Now we will get the restaurant list for each neighborhood using the function (get_venues) that we created before and we will store them in a newly created DataFrame(indian_rest_ny)



```
In [23]: indian_rest_ny.head()
Out[23]:
```

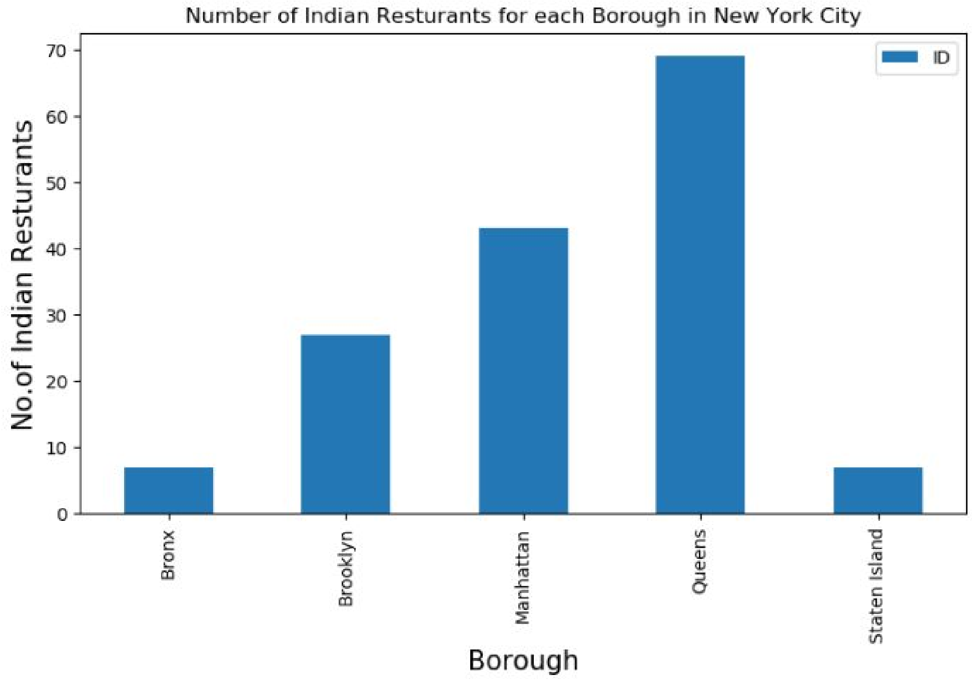
	Borough	Neighborhood	ID	Name
0	Bronx	Woodlawn	4c0448d9310fc9b6bf1dc761	Curry Spot
1	Bronx	Parkchester	4c194631838020a13e78e561	Melanies Roti Bar And Grill
2	Bronx	Parkchester	55dfa36a498e164ef19bef7b	Premium Sweets & Restaurant
3	Bronx	Spuyten Duyvil	4c04544df423a593ac83d116	Cumin Indian Cuisine
4	Bronx	Concourse	551b7f75498e86c00a0ed2e1	Hungry Bird

```
In [24]: indian_rest_ny.shape
Out[24]: (153, 4)
```

We have here total 153 Restaurants

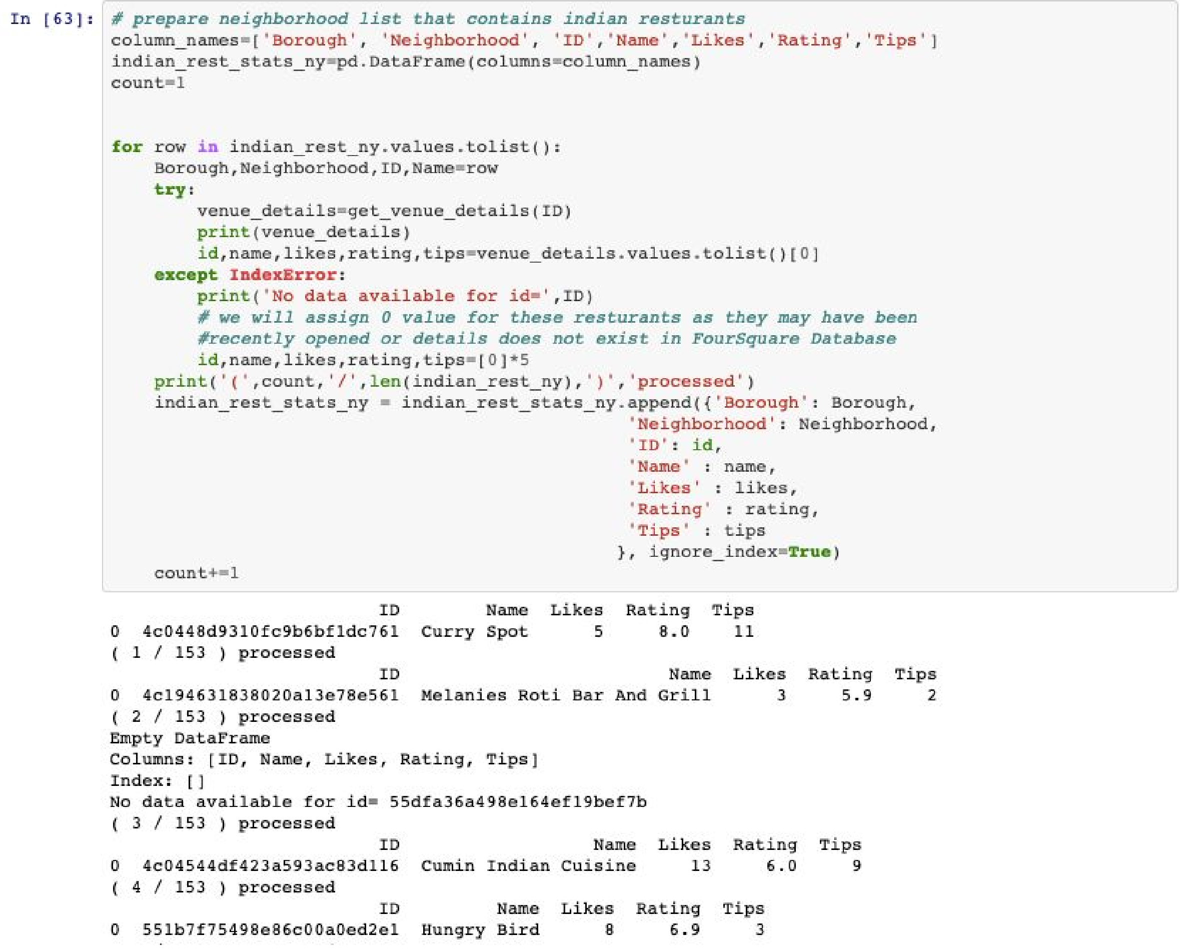
```
In [26]: plt.figure(figsize=(9,5), dpi = 100)
         # title
         plt.title('Number of Indian Resturants for each Borough in New York City')
         #On x-axis
         plt.xlabel('Borough', fontsize = 15)
         #On y-axis
         plt.ylabel('No.of Indian Resturants', fontsize=15)
         #giving a bar plot
         indian_rest_ny.groupby('Borough')['ID'].count().plot(kind='bar')
         #legend
         plt.legend()
         #displays the plot
         plt.show()
```

By analysis we found that there are overall 153 Indian restaurants which are to be divided on the basis of Boroughs.



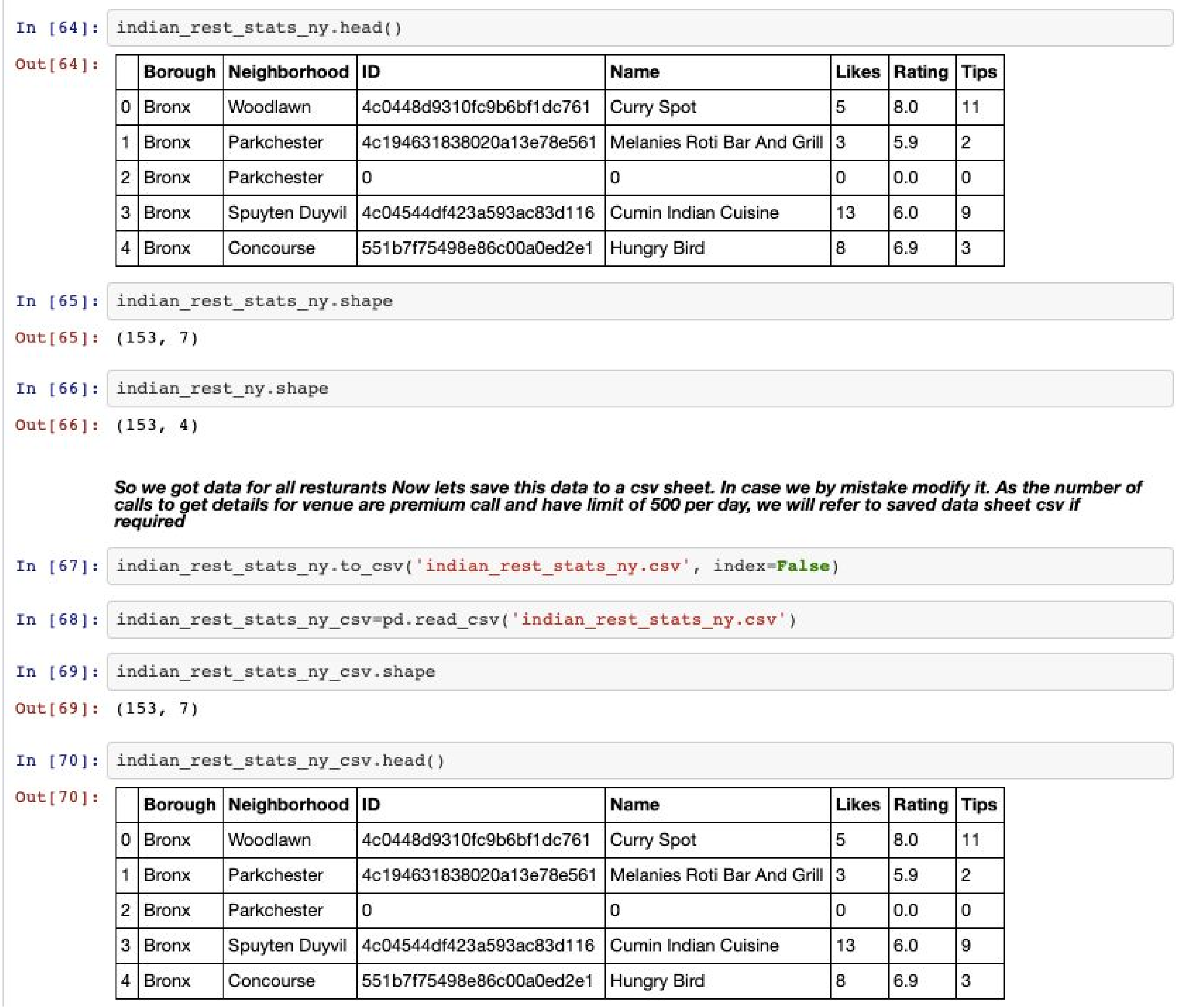
As we can see that the maximum number of restaurants are in the Queens.

Now We will create a Restaurant list according to the neighborhood and store that in a new dataframe (indian_rest_stats_ny)



```
In [63]: # prepare neighborhood list that contains indian resturants
column_names=['Borough', 'Neighborhood', 'ID','Name','Likes','Rating','Tips']
indian_rest_stats_ny=pd.DataFrame(columns=column_names)
count=1


for row in indian_rest_ny.values.tolist():
    Borough,Neighborhood,ID,Name=row
    try:
        venue_details=get_venue_details(ID)
        print(venue_details)
        id,name,likes,rating,tips=venue_details.values.tolist()[0]
    except IndexError:
        print('No data available for id=',ID)
        # we will assign 0 value for these resturants as they may have been
        #recently opened or details does not exist in FourSquare Database
        id,name,likes,rating,tips=[0]*5
    print('(',count,'/',len(indian_rest_ny),')','processed')
    indian_rest_stats_ny = indian_rest_stats_ny.append({'Borough': Borough,
                                                        'Neighborhood': Neighborhood,
                                                        'ID': id,
                                                        'Name' : name,
                                                        'Likes' : likes,
                                                        'Rating' : rating,
                                                        'Tips' : tips
                                                       }, ignore_index=True)
    count+=1
```

```
                         ID        Name  Likes  Rating  Tips
0  4c0448d9310fc9b6bf1dc761  Curry Spot      5     8.0    11
( 1 / 153 ) processed
                         ID                           Name  Likes  Rating  Tips
0  4c194631838020a13e78e561  Melanies Roti Bar And Grill      3     5.9     2
( 2 / 153 ) processed
Empty DataFrame
Columns: [ID, Name, Likes, Rating, Tips]
Index: []
No data available for id= 55dfa36a498e164ef19bef7b
( 3 / 153 ) processed
                         ID                   Name  Likes  Rating  Tips
0  4c04544df423a593ac83d116  Cumin Indian Cuisine     13     6.0     9
( 4 / 153 ) processed
                         ID         Name  Likes  Rating  Tips
0  551b7f75498e86c00a0ed2e1  Hungry Bird      8     6.9     3
```



```
In [64]: indian_rest_stats_ny.head()
```

Out[64]:

	Borough	Neighborhood	ID	Name	Likes	Rating	Tips
0	Bronx	Woodlawn	4c0448d9310fc9b6bf1dc761	Curry Spot	5	8.0	11
1	Bronx	Parkchester	4c194631838020a13e78e561	Melanies Roti Bar And Grill	3	5.9	2
2	Bronx	Parkchester	0	0	0	0.0	0
3	Bronx	Spuyten Duyvil	4c04544df423a593ac83d116	Cumin Indian Cuisine	13	6.0	9
4	Bronx	Concourse	551b7f75498e86c00a0ed2e1	Hungry Bird	8	6.9	3

```
In [65]: indian_rest_stats_ny.shape

Out[65]: (153, 7)
```

```
In [66]: indian_rest_ny.shape

Out[66]: (153, 4)
```

So we got data for all resturants Now lets save this data to a csv sheet. In case we by mistake modify it. As the number of calls to get details for venue are premium call and have limit of 500 per day, we will refer to saved data sheet csv if required

```
In [67]: indian_rest_stats_ny.to_csv('indian_rest_stats_ny.csv', index=False)
```

```
In [68]: indian_rest_stats_ny_csv=pd.read_csv('indian_rest_stats_ny.csv')
```

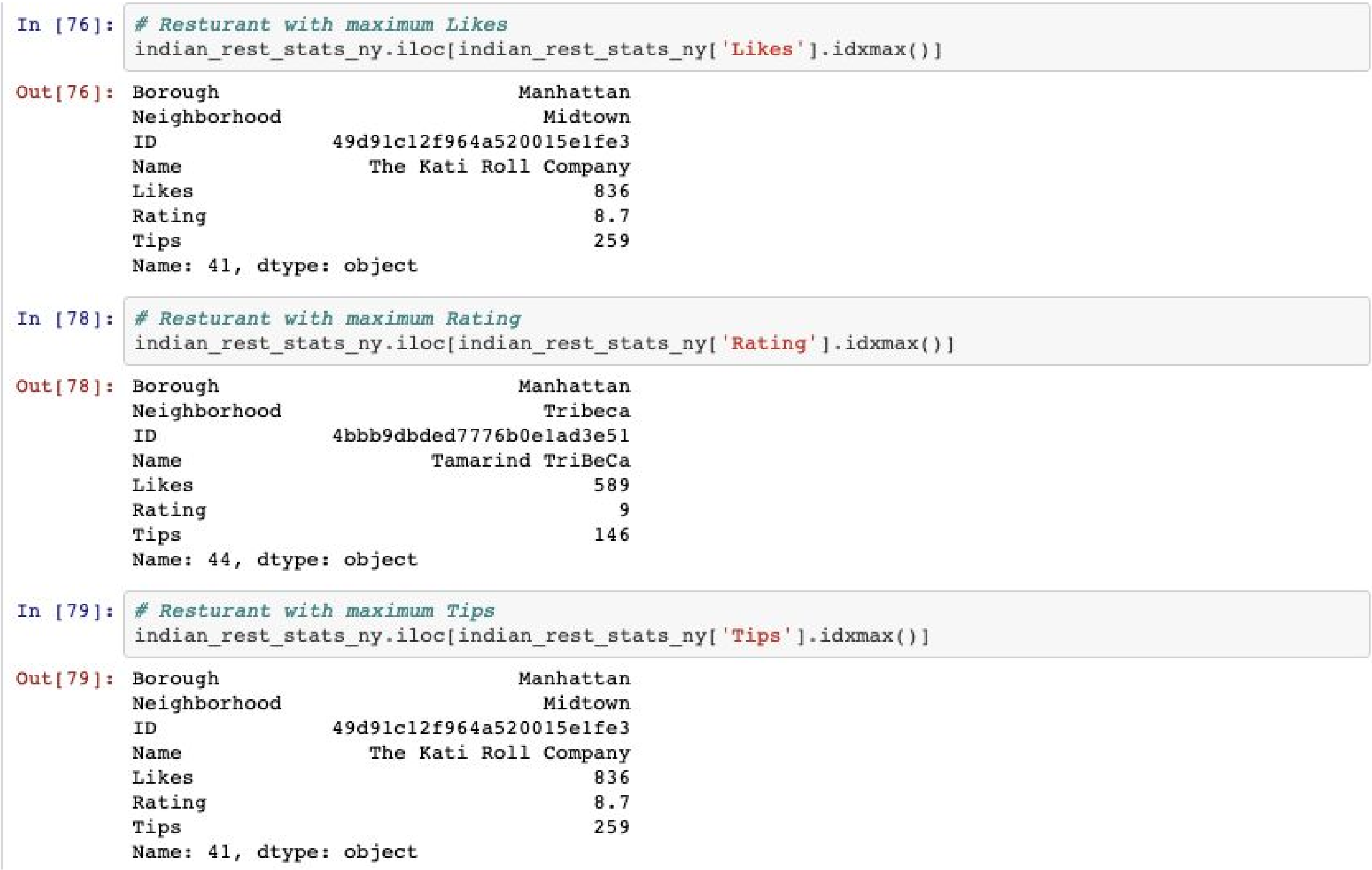
```
In [69]: indian_rest_stats_ny_csv.shape

Out[69]: (153, 7)
```

```
In [70]: indian_rest_stats_ny_csv.head()
```

Out[70]:

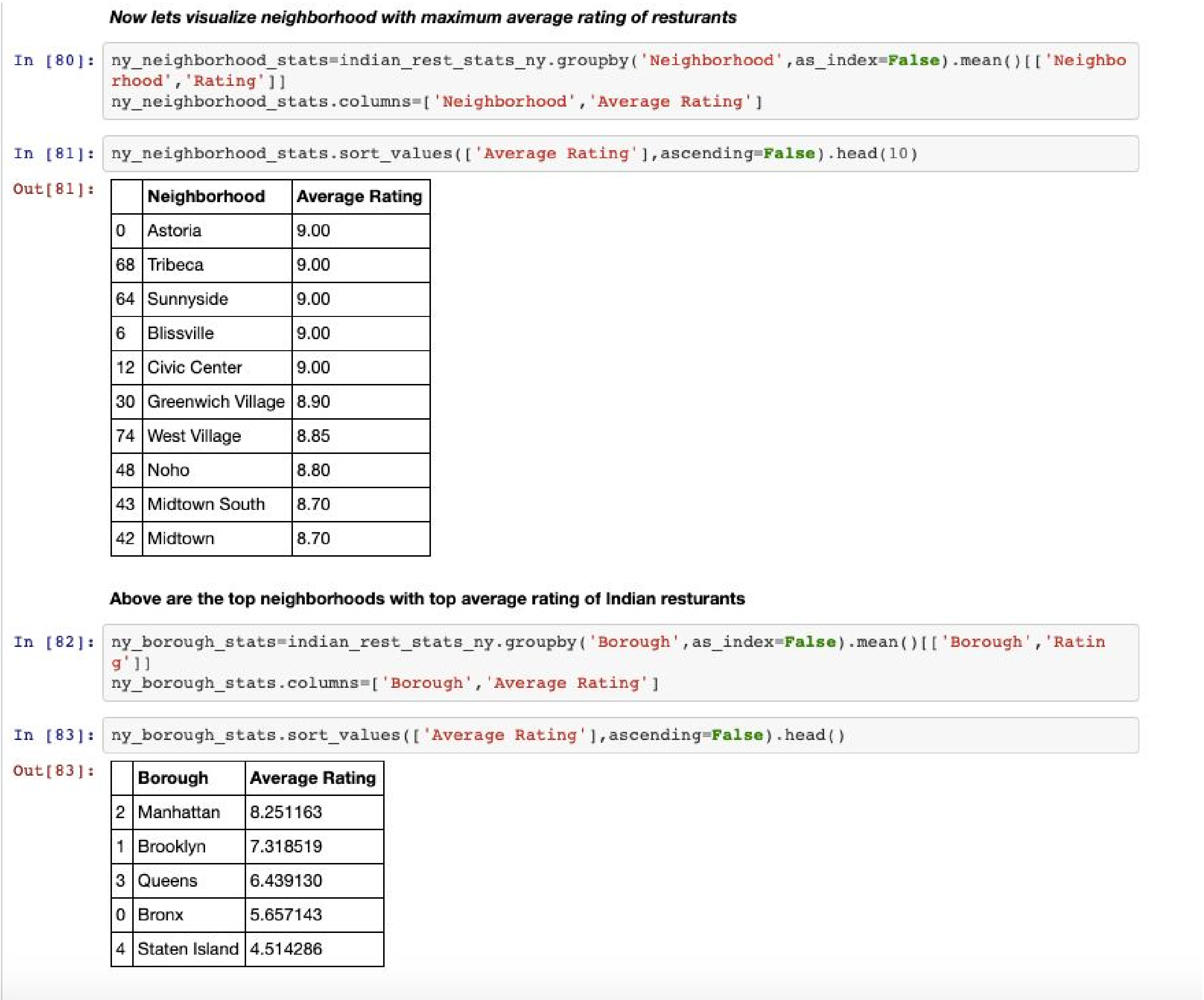
	Borough	Neighborhood	ID	Name	Likes	Rating	Tips
0	Bronx	Woodlawn	4c0448d9310fc9b6bf1dc761	Curry Spot	5	8.0	11
1	Bronx	Parkchester	4c194631838020a13e78e561	Melanies Roti Bar And Grill	3	5.9	2
2	Bronx	Parkchester	0	0	0	0.0	0
3	Bronx	Spuyten Duyvil	4c04544df423a593ac83d116	Cumin Indian Cuisine	13	6.0	9
4	Bronx	Concourse	551b7f75498e86c00a0ed2e1	Hungry Bird	8	6.9	3

As we know that there are various call limitations in the Foursquare we create a csv file for the last dataframe and now we are in a safe spot even if we unintentionally temper with the dataframe.



```
In [76]: # Resturant with maximum Likes
         indian_rest_stats_ny.iloc[indian_rest_stats_ny['Likes'].idxmax()]

Out[76]: Borough                        Manhattan
         Neighborhood                     Midtown
         ID               49d91c12f964a520015e1fe3
         Name                 The Kati Roll Company
         Likes                                  836
         Rating                                 8.7
         Tips                                   259
         Name: 41, dtype: object

In [78]: # Resturant with maximum Rating
         indian_rest_stats_ny.iloc[indian_rest_stats_ny['Rating'].idxmax()]

Out[78]: Borough                        Manhattan
         Neighborhood                     Tribeca
         ID               4bbb9dbded7776b0e1ad3e51
         Name                      Tamarind TriBeCa
         Likes                                  589
         Rating                                   9
         Tips                                   146
         Name: 44, dtype: object

In [79]: # Resturant with maximum Tips
         indian_rest_stats_ny.iloc[indian_rest_stats_ny['Tips'].idxmax()]

Out[79]: Borough                        Manhattan
         Neighborhood                     Midtown
         ID               49d91c12f964a520015e1fe3
         Name                 The Kati Roll Company
         Likes                                  836
         Rating                                 8.7
         Tips                                   259
         Name: 41, dtype: object
```

We can perform various Analysis with the dataframe.



Now lets visualize neighborhood with maximum average rating of resturants

In [80]:
```
ny_neighborhood_stats=indian_rest_stats_ny.groupby('Neighborhood',as_index=False).mean()[['Neighborhood','Rating']]
ny_neighborhood_stats.columns=['Neighborhood','Average Rating']
```

In [81]:
```
ny_neighborhood_stats.sort_values(['Average Rating'],ascending=False).head(10)
```

Out[81]:

	Neighborhood	Average Rating
0	Astoria	9.00
68	Tribeca	9.00
64	Sunnyside	9.00
6	Blissville	9.00
12	Civic Center	9.00
30	Greenwich Village	8.90
74	West Village	8.85
48	Noho	8.80
43	Midtown South	8.70
42	Midtown	8.70

Above are the top neighborhoods with top average rating of Indian resturants

In [82]:
```
ny_borough_stats=indian_rest_stats_ny.groupby('Borough',as_index=False).mean()[['Borough','Rating']]
ny_borough_stats.columns=['Borough','Average Rating']
```

In [83]:
```
ny_borough_stats.sort_values(['Average Rating'],ascending=False).head()
```

Out[83]:

	Borough	Average Rating
2	Manhattan	8.251163
1	Brooklyn	7.318519
3	Queens	6.439130
0	Bronx	5.657143
4	Staten Island	4.514286

In [93]:
```
# create map and display it
ny_map = folium.Map(location=geo_location('New York'), zoom_start=12)
```

In [94]:
```
# instantiate a feature group for the incidents in the dataframe
incidents = folium.map.FeatureGroup()

# loop through the 100 crimes and add each to the incidents feature group
for lat, lng, in ny_neighborhood_stats[['Latitude','Longitude']].values:
    incidents.add_child(
        folium.CircleMarker(
            [lat, lng],
            radius=10, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='blue',
            fill_opacity=0.6
        )
    )
```

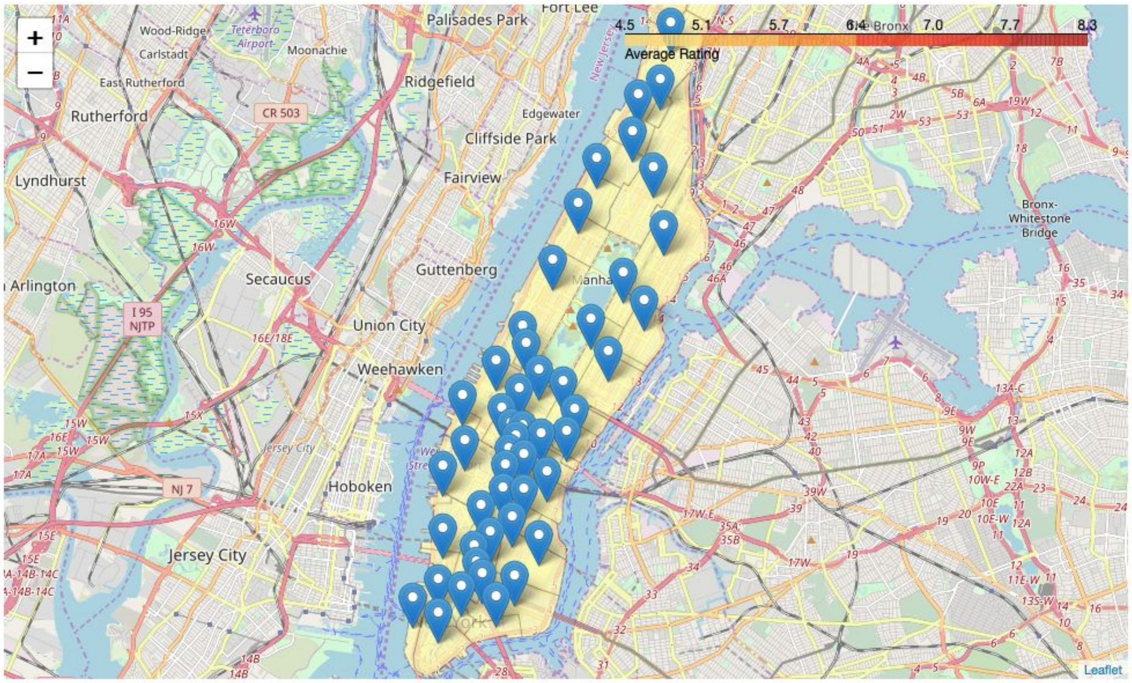
In [95]:
```
#let's add new field to dataframe for labelling
```

In [97]:
```
ny_neighborhood_stats['Label']=ny_neighborhood_stats['Neighborhood']+', '+ny_neighborhood_stats['B
orough']+'('+ny_neighborhood_stats['Average Rating'].map(str)+')'
```

In [98]:
```
ny_neighborhood_stats['Label']=ny_neighborhood_stats['Neighborhood']+', '+ny_neighborhood_stats['B
orough']+'('+ny_neighborhood_stats['Average Rating'].map(str)+')'
```

In [119]:
```
ny_map = folium.Map(location=geo_location('New York'), zoom_start=12)

json_url=r'USA New York City neighborhood 20190128.geojson'
ny_map.choropleth(
    geo_data=json_url,
    data=ny_borough_stats,
    columns=['Borough', 'Average Rating'],
    key_on='feature.properties.boro_name',
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Average Rating'
)
ny_map
```

Out[119]:



Now we can plot the restaurants according to ratings as shown above to see where are the restaurants having Average or above Average ratings.

Conclusion: By the Analysis of Indian Restaurants we can help or target the audience looking for indians restaurant to eat with satisfactory ratings i.e which provide good indian food.

Analysis can also be helpful for people who are looking to open a new Indian restaurant in New York as by analysis we can easily conclude that Queens has the most number of Indian restaurants. Whereas, Manhattan has some of the finest ones and Manhattan can prove to be a better market for any upcoming Indian Restaurants.

Limitations: We have limited data (from FourSquare DataBase)

Our Analysis is mainly focussed upon ratings.