The Battle of Neighbourhoods Code

In [1]:
```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

#!pip install geopy

#!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas import json_normalize # tranform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

#!pip install folium

#!conda install -c conda-forge folium=0.4.0 --yes # uncomment this line if you haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Libraries imported.
```

In [2]:
```
#!pip install wget

import wget

wget.download('https://cocl.us/new_york_dataset', 'newyork_data.json')
print('Data downloaded!')
```

```
Data downloaded!
```

In [3]:
```
with open('newyork_data.json') as json_data:
    newyork_data = json.load(json_data)
```

In [4]:
```
neighborhoods_data = newyork_data['features']
neighborhoods_data[0]
```

Out[4]:
```
{'type': 'Feature',
 'id': 'nyu_2451_34572.1',
 'geometry': {'type': 'Point',
  'coordinates': [-73.84720052054902, 40.89470517661]},
 'geometry_name': 'geom',
 'properties': {'name': 'Wakefield',
  'stacked': 1,
  'annoline1': 'Wakefield',
  'annoline2': None,
  'annoline3': None,
  'annoangle': 0.0,
  'borough': 'Bronx',
  'bbox': [-73.84720052054902,
   40.89470517661,
   -73.84720052054902,
   40.89470517661]}}
```

In [5]:
```
# define the dataframe columns
column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

# instantiate the dataframe
neighborhoods = pd.DataFrame(columns=column_names)
neighborhoods
```

Out[5]:

	Borough	Neighborhood	Latitude	Longitude

In [6]:
```
for data in neighborhoods_data:
    borough = neighborhood_name = data['properties']['borough']
    neighborhood_name = data['properties']['name']

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

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

Out[6]:

	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

In [7]:
```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(neighborhoods['Borough'].unique()),
        neighborhoods.shape[0]
    )
)
```

```
The dataframe has 5 boroughs and 306 neighborhoods.
```

In [8]:
```
address = 'New York City, NY'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of New York City are {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinate of New York City are 40.7127281, -74.0060152.
```

In [9]:
```
# create map of New York using latitude and longitude values
map_newyork = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(neighborhoods['Latitude'], neighborhoods['Longitude'], neighborhoods['Borough'], neighborhoods
['Neighborhood']):
    label = '{}, {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_newyork)

map_newyork
```

Out[9]:
```
Make this Notebook Trusted to load map: File -> Trust Notebook
```

In [10]:
```
CLIENT_ID = 'UJDGZH02O10QRTD3YAFNK0QJJZNLVYATZY2TN1LDQGNVYXYU' # your Foursquare ID
CLIENT_SECRET = 'HVWENYFLNTLB5L4LZ3OJQXYVQMGA0TEW3OSNTWUKW4SEFJBR' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: UJDGZH02O10QRTD3YAFNK0QJJZNLVYATZY2TN1LDQGNVYXYU
CLIENT_SECRET:HVWENYFLNTLB5L4LZ3OJQXYVQMGA0TEW3OSNTWUKW4SEFJBR
```

In [11]:
```
neighborhood_latitude = neighborhoods.loc[0, 'Latitude'] # neighborhood latitude value
neighborhood_longitude = neighborhoods.loc[0, 'Longitude'] # neighborhood longitude value

neighborhood_name = neighborhoods.loc[0, 'Neighborhood'] # neighborhood name

print('Latitude and longitude values of {} are {}, {}.'.format(neighborhood_name,
                                                               neighborhood_latitude,
                                                               neighborhood_longitude))
```

```
Latitude and longitude values of Wakefield are 40.89470517661, -73.84720052054902.
```

In [12]:
```
LIMIT = 100
radius = 500
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    neighborhood_latitude,
    neighborhood_longitude,
    radius,
    LIMIT)
url
```

Out[12]:
```
'https://api.foursquare.com/v2/venues/explore?&client_id=UJDGZH02O10QRTD3YAFNK0QJJZNLVYATZY2TN1LDQGNVYXYU&client_secret=HVWENYFLNTLB5L4LZ
3OJQXYVQMGA0TEW3OSNTWUKW4SEFJBR&v=20180605&ll=40.89470517661,-73.84720052054902&radius=500&limit=100'
```

In [13]:
```
results = requests.get(url).json()
```

In [14]:
```
# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

In [15]:
```
venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
```

Out[15]:

	name	categories	lat	lng
0	Lollipops Gelato	Dessert Shop	40.894123	-73.845892
1	Walgreens	Pharmacy	40.896528	-73.844700
2	Carvel Ice Cream	Ice Cream Shop	40.890487	-73.848568
3	Rite Aid	Pharmacy	40.896649	-73.844846
4	Dunkin'	Donut Shop	40.890459	-73.849089

In [16]:
```
print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))
```

```
11 venues were returned by Foursquare.
```

In [17]:
```
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        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)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

```
Wakefield
Co-op City
Eastchester
Fieldston
Riverdale
Kingsbridge
Marble Hill
Woodlawn
Norwood
Williamsbridge
Baychester
Pelham Parkway
City Island
Bedford Park
University Heights
Morris Heights
Fordham
East Tremont
West Farms
High  Bridge
Melrose
Mott Haven
Port Morris
Longwood
Hunts Point
Morrisania
Soundview
Clason Point
Throgs Neck
Country Club
Parkchester
```

In [19]:
```
print(newyork_venues.shape)
newyork_venues.head()
```

```
(10120, 7)
```

Out[19]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Wakefield	40.894705	-73.847201	Lollipops Gelato	40.894123	-73.845892	Dessert Shop
1	Wakefield	40.894705	-73.847201	Walgreens	40.896528	-73.844700	Pharmacy
2	Wakefield	40.894705	-73.847201	Carvel Ice Cream	40.890487	-73.848568	Ice Cream Shop
3	Wakefield	40.894705	-73.847201	Rite Aid	40.896649	-73.844846	Pharmacy
4	Wakefield	40.894705	-73.847201	Dunkin'	40.890459	-73.849089	Donut Shop

In [20]:
```
#choose only rows which refer to coffee shops, and therefore, only neighborhoods which have at least one coffee shop
filtered_category = newyork_venues[newyork_venues['Venue Category'] == 'Coffee Shop']
filtered_category
```

Out[20]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
71	Kingsbridge	40.881687	-73.902818	Mon Amour Coffee & Wine	40.885009	-73.900332	Coffee Shop
132	Marble Hill	40.876551	-73.910660	Starbucks	40.877531	-73.905582	Coffee Shop
137	Marble Hill	40.876551	-73.910660	Starbucks	40.873755	-73.908613	Coffee Shop
179	Norwood	40.877224	-73.879391	Nicky's Coffee Shop	40.874933	-73.879404	Coffee Shop
242	Pelham Parkway	40.857413	-73.854756	Liberty Donut & Coffee Shop	40.855339	-73.855333	Coffee Shop
317	Bedford Park	40.870185	-73.885512	National Coffee Shop	40.872841	-73.889053	Coffee Shop
473	West Farms	40.839475	-73.877745	Prospect Coffee Shop	40.837577	-73.880839	Coffee Shop
550	Mott Haven	40.806239	-73.916100	Brook Lunch	40.807472	-73.919510	Coffee Shop
649	Throgs Neck	40.815109	-73.816350	The Miles Coffee Bar	40.819462	-73.817352	Coffee Shop
726	Van Nest	40.843608	-73.866299	Conti's Pastry Shoppe	40.845906	-73.862836	Coffee Shop
758	Morris Park	40.847549	-73.850402	La Casa Del Caffe	40.848675	-73.854973	Coffee Shop
828	Belmont	40.857277	-73.888452	Starbucks	40.860636	-73.890270	Coffee Shop
836	Belmont	40.857277	-73.888452	Starbucks	40.861106	-73.886148	Coffee Shop
882	North Riverdale	40.908543	-73.904531	Noni's Coffee Shop	40.907355	-73.904161	Coffee Shop
957	Edgewater Park	40.821986	-73.813885	The Miles Coffee Bar	40.819462	-73.817352	Coffee Shop
976	Edgewater Park	40.821986	-73.813885	Bridges	40.818697	-73.817371	Coffee Shop
1076	Bay Ridge	40.625801	-74.030621	Caffe Café	40.624946	-74.030404	Coffee Shop
1235	Greenpoint	40.730201	-73.954241	Homecoming	40.729696	-73.957525	Coffee Shop

In [21]:
```
#make sure no two rows are duplicates
filtered_category = filtered_category.drop_duplicates()
filtered_category
```

Out[21]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
71	Kingsbridge	40.881687	-73.902818	Mon Amour Coffee & Wine	40.885009	-73.900332	Coffee Shop
132	Marble Hill	40.876551	-73.910660	Starbucks	40.877531	-73.905582	Coffee Shop
137	Marble Hill	40.876551	-73.910660	Starbucks	40.873755	-73.908613	Coffee Shop
179	Norwood	40.877224	-73.879391	Nicky's Coffee Shop	40.874933	-73.879404	Coffee Shop
242	Pelham Parkway	40.857413	-73.854756	Liberty Donut & Coffee Shop	40.855339	-73.855333	Coffee Shop
317	Bedford Park	40.870185	-73.885512	National Coffee Shop	40.872841	-73.889053	Coffee Shop
473	West Farms	40.839475	-73.877745	Prospect Coffee Shop	40.837577	-73.880839	Coffee Shop
550	Mott Haven	40.806239	-73.916100	Brook Lunch	40.807472	-73.919510	Coffee Shop
649	Throgs Neck	40.815109	-73.816350	The Miles Coffee Bar	40.819462	-73.817352	Coffee Shop
726	Van Nest	40.843608	-73.866299	Conti's Pastry Shoppe	40.845906	-73.862836	Coffee Shop
758	Morris Park	40.847549	-73.850402	La Casa Del Caffe	40.848675	-73.854973	Coffee Shop
828	Belmont	40.857277	-73.888452	Starbucks	40.860636	-73.890270	Coffee Shop
836	Belmont	40.857277	-73.888452	Starbucks	40.861106	-73.886148	Coffee Shop
882	North Riverdale	40.908543	-73.904531	Noni's Coffee Shop	40.907355	-73.904161	Coffee Shop
957	Edgewater Park	40.821986	-73.813885	The Miles Coffee Bar	40.819462	-73.817352	Coffee Shop
976	Edgewater Park	40.821986	-73.813885	Bridges	40.818697	-73.817371	Coffee Shop
1076	Bay Ridge	40.625801	-74.030621	Caffe Café	40.624946	-74.030404	Coffee Shop
1235	Greenpoint	40.730201	-73.954241	Homecoming	40.729696	-73.957525	Coffee Shop

In [22]:
```
#venue name, latitude and longitude and category are not relevant
coffee_shop = filtered_category[['Neighborhood', 'Neighborhood Latitude', 'Neighborhood Longitude']].copy()
coffee_shop
```

Out[22]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude
71	Kingsbridge	40.881687	-73.902818
132	Marble Hill	40.876551	-73.910660
137	Marble Hill	40.876551	-73.910660
179	Norwood	40.877224	-73.879391
242	Pelham Parkway	40.857413	-73.854756
317	Bedford Park	40.870185	-73.885512
473	West Farms	40.839475	-73.877745
550	Mott Haven	40.806239	-73.916100
649	Throgs Neck	40.815109	-73.816350
726	Van Nest	40.843608	-73.866299
758	Morris Park	40.847549	-73.850402
828	Belmont	40.857277	-73.888452
836	Belmont	40.857277	-73.888452
882	North Riverdale	40.908543	-73.904531
957	Edgewater Park	40.821986	-73.813885
976	Edgewater Park	40.821986	-73.813885
1076	Bay Ridge	40.625801	-74.030621

In [23]:
```
#add a column which counts the number of times a neighbourhood appears in the dataframe, i.e. the number of coffee shops there are in the neighbourhood
coffee_shop['count'] = coffee_shop.groupby('Neighborhood')['Neighborhood'].transform('count')
coffee_shop = coffee_shop.rename(columns={'count': 'Number of Coffee Shops'})
coffee_shop = coffee_shop.rename(columns={'Neighborhood Latitude': 'Latitude'})
coffee_shop = coffee_shop.rename(columns={'Neighborhood Longitude': 'Longitude'})
coffee_shop
```

Out[23]:

	Neighborhood	Latitude	Longitude	Number of Coffee Shops
71	Kingsbridge	40.881687	-73.902818	1
132	Marble Hill	40.876551	-73.910660	2
137	Marble Hill	40.876551	-73.910660	2
179	Norwood	40.877224	-73.879391	1
242	Pelham Parkway	40.857413	-73.854756	1
317	Bedford Park	40.870185	-73.885512	1
473	West Farms	40.839475	-73.877745	1
550	Mott Haven	40.806239	-73.916100	1
649	Throgs Neck	40.815109	-73.816350	1
726	Van Nest	40.843608	-73.866299	1
758	Morris Park	40.847549	-73.850402	1
828	Belmont	40.857277	-73.888452	2
836	Belmont	40.857277	-73.888452	2
882	North Riverdale	40.908543	-73.904531	1
957	Edgewater Park	40.821986	-73.813885	2

In [24]:
```
#final dataframe showing number of coffee shops in every neighbourhood in New York
coffee_shop = coffee_shop.drop_duplicates()
coffee_shop
```

Out[24]:

	Neighborhood	Latitude	Longitude	Number of Coffee Shops
71	Kingsbridge	40.881687	-73.902818	1
132	Marble Hill	40.876551	-73.910660	2
179	Norwood	40.877224	-73.879391	1
242	Pelham Parkway	40.857413	-73.854756	1
317	Bedford Park	40.870185	-73.885512	1
473	West Farms	40.839475	-73.877745	1
550	Mott Haven	40.806239	-73.916100	1
649	Throgs Neck	40.815109	-73.816350	1
726	Van Nest	40.843608	-73.866299	1
758	Morris Park	40.847549	-73.850402	1
828	Belmont	40.857277	-73.888452	2
882	North Riverdale	40.908543	-73.904531	1
957	Edgewater Park	40.821986	-73.813885	2
1076	Bay Ridge	40.625801	-74.030621	1
1235	Greenpoint	40.730201	-73.954241	6

In [25]:
```
#number of neighborhoods (out of 306) in New York which contain at least one coffee shop - this is the number of markers which will be sh
own on the map
coffee_shop.shape
```

Out[25]:
```
(122, 4)
```

In [26]:
```
lat = coffee_shop['Latitude']
lng = coffee_shop['Longitude']
neighborhood = coffee_shop['Neighborhood']
number = coffee_shop['Number of Coffee Shops']
```

In [27]:
```
#create map which shows number of coffee shops in neighborhoods
coffee_shop_map = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, neighborhood, number in zip(coffee_shop['Latitude'], coffee_shop['Longitude'], coffee_shop['Neighborhood'], coffee_shop['Nu
mber of Coffee Shops']):
    label = '{}, {}'.format(neighborhood, number)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(coffee_shop_map)

coffee_shop_map
```

Out[27]:
```
Make this Notebook Trusted to load map: File -> Trust Notebook
```

In [28]:
```
#first group will be neighborhoods containing 3 or less coffee shops
group_a = coffee_shop[coffee_shop['Number of Coffee Shops'] <= 3]

lat_a = group_a['Latitude']
lng_a = group_a['Longitude']
neighborhood_a = group_a['Neighborhood']
number_a = group_a['Number of Coffee Shops']

group_a
```

Out[28]:

	Neighborhood	Latitude	Longitude	Number of Coffee Shops
71	Kingsbridge	40.881687	-73.902818	1
132	Marble Hill	40.876551	-73.910660	2
179	Norwood	40.877224	-73.879391	1
242	Pelham Parkway	40.857413	-73.854756	1
317	Bedford Park	40.870185	-73.885512	1
473	West Farms	40.839475	-73.877745	1
550	Mott Haven	40.806239	-73.916100	1
649	Throgs Neck	40.815109	-73.816350	1
726	Van Nest	40.843608	-73.866299	1
758	Morris Park	40.847549	-73.850402	1
828	Belmont	40.857277	-73.888452	2
882	North Riverdale	40.908543	-73.904531	1
957	Edgewater Park	40.821986	-73.813885	2

In [29]:
```
#second group will be neighborhoods containing between 4 and 6 coffee shops
group_b = coffee_shop[(coffee_shop['Number of Coffee Shops'] >= 4) & (coffee_shop['Number of Coffee Shops'] <= 6)]

lat_b = group_b['Latitude']
lng_b = group_b['Longitude']
neighborhood_b = group_b['Neighborhood']
number_b = group_b['Number of Coffee Shops']

group_b
```

Out[29]:

	Neighborhood	Latitude	Longitude	Number of Coffee Shops
1235	Greenpoint	40.730201	-73.954241	6
1712	Bushwick	40.698116	-73.925258	5
1931	Cobble Hill	40.687920	-73.998561	4
2283	Park Slope	40.672321	-73.977050	5
2654	Downtown	40.690844	-73.983463	5
2748	Boerum Hill	40.685683	-73.983748	5
3054	East Williamsburg	40.708492	-73.938858	4
3659	Hamilton Heights	40.823604	-73.949688	4
3719	Manhattanville	40.816934	-73.957385	4
3839	Upper East Side	40.775639	-73.960508	6
3934	Yorkville	40.775930	-73.947118	5
4361	Clinton	40.759101	-73.996119	4
5082	Little Italy	40.719324	-73.997305	4
5454	Gramercy	40.737210	-73.981376	4
5568	Battery Park City	40.711932	-74.016869	4

In [30]:
```
#third group will be neighborhoods containing 7 or more coffee shops
group_c = coffee_shop[coffee_shop['Number of Coffee Shops'] >= 7]

lat_c = group_c['Latitude']
lng_c = group_c['Longitude']
neighborhood_c = group_c['Neighborhood']
number_c = group_c['Number of Coffee Shops']

group_c
```

Out[30]:

	Neighborhood	Latitude	Longitude	Number of Coffee Shops
1997	Carroll Gardens	40.680540	-73.994654	7
3127	North Side	40.714823	-73.958809	9
3225	South Side	40.710861	-73.958001	7
4072	Lenox Hill	40.768113	-73.958860	7
4460	Midtown	40.754691	-73.981669	8
4542	Murray Hill	40.748303	-73.978332	7
4647	Chelsea	40.744035	-74.003116	9
5207	Soho	40.722184	-74.000657	7
5614	Financial District	40.707107	-74.010665	8
6234	Long Island City	40.750217	-73.939202	8
7236	Murray Hill	40.764126	-73.812763	7
8314	Carnegie Hill	40.782683	-73.953256	7

In [31]:
```
#check all 122 neighborhoods from complete dataframe have been included
group_a.shape
```

Out[31]: (90, 4)

In [32]:
```
group_b.shape
```

Out[32]:
```
(20, 4)
```

In [33]:
```
group_c.shape
```

Out[33]:
```
(12, 4)
```

In [34]:
```
#create map showing categories: less than 3, 4-6, more than 7 coffee shops in neighborhoods
coffee_shop_with_categories = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map GROUP A
for lat_a, lng_a, neighborhood_a, number_a in zip(group_a['Latitude'], group_a['Longitude'], group_a['Neighborhood'], group_a['Number of Coffee Shops']):
    label = '{}, {}'.format(neighborhood_a, number_a)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat_a, lng_a],
        radius=5,
        popup=label,
        color='sunflower yellow',
        fill=True,
        fill_color='#fffb00',
        fill_opacity=0.7,
        parse_html=False).add_to(coffee_shop_with_categories)

# add markers to map GROUP B
for lat_b, lng_b, neighborhood_b, number_b in zip(group_b['Latitude'], group_b['Longitude'], group_b['Neighborhood'], group_b['Number of Coffee Shops']):
    label = '{}, {}'.format(neighborhood_b, number_b)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat_b, lng_b],
        radius=5,
        popup=label,
        color='orange red',
        fill=True,
        fill_color='#ff3700',
        fill_opacity=0.7,
        parse_html=False).add_to(coffee_shop_with_categories)

# add markers to map GROUP C
for lat_c, lng_c, neighborhood_c, number_c in zip(group_c['Latitude'], group_c['Longitude'], group_c['Neighborhood'], group_c['Number of Coffee Shops']):
    label = '{}, {}'.format(neighborhood_c, number_c)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat_c, lng_c],
        radius=5,
        popup=label,
        color='dark violet',
        fill=True,
        fill_color='#7c00c4',
        fill_opacity=0.7,
        parse_html=False).add_to(coffee_shop_with_categories)

coffee_shop_with_categories
```

Out[34]:
Make this Notebook Trusted to load map: File -> Trust Notebook