Coffee Shops Analysis in Manhattan; Segmenting and Clustering Neighborhoods

Capstone Project - Battle of the Neighborhoods: Applied Data Science Capstone by IBM(Coursera)

Author: Lopamudra Nayak

Table of Contents

1. Introduction

New York is one of the most famous cities in the world also referred as the "Big Apple", this vibrant city is known for its exclusive shops, flashy Broadway performances, and high-flying business tycoons, and it's a city that has long captivated people from all over the world. But, aside from the flashing lights, it's also home to the beloved Statue of Liberty, a symbol of freedom and hope, and Central Park, one of the tidiest and best-kept parks in the world. The proverbial apple of New York State's eye, Manhattan is the most popular tourist destination and one of the best places to visit in New York State. It's also one of the best cities in New York and, arguably, the world. One can find literally everything your heart desires starting from the rarest antique cufflinks to the most delicious entree to grace your lips. An activity that is touristy but also for its residents is drinking coffee. The activity can be done alone, accompanied by friends, and anytime in this city.

Americans drink about 146 billion cups of coffee per year. In 2020, Americans drank an average of 1.87 cups of coffee per day. Between 2011 and 2020, US. coffee consumption per capita per day remained relatively close to two cups, with the exception of 2016, when US survey respondents drank roughly 1.64 cups of coffee per day. In US, people aged seventy and older drank the most coffee per day in 2020. This particular age group drank an average of almost 2.2 cups per capita per day that year. Americans aged eighteen and nineteen drank the least per capita, namely about 0.8 cups a day. In terms of gender, men drank more than women per capita in the United States in the same year. In 2020, Americans drank about 1.3 cups of coffee during breakfast each day, making it the most common time of day for people to drink coffee in the US that year. The least common time for Americans to drink coffee was during other meals. (total-us-coffee-per-capita-consumption (https://www.statista.com/statistics/456360/total-us-coffee-per-capita-consumption/))

In this project, we will try to find an optimal location for a coffee shop. This report will be targeted at stakeholders who want to start from scratch, buy an existing business, or anybody interested in a good coffee in **Manhattan**, New York City.

We will try to detect **locations that are not already crowded with coffee shops**. We are also particularly interested in **areas with no coffee shops in the vicinity**. We would also prefer locations **as close to the city center as possible**, assuming that the first two conditions are met.

Import Libraries:

Before we get the data and start exploring it, let's download all the dependencies that we will need.

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

#!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas.io.json 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

#!conda install -c conda-forge folium=0.5.0 --yes
import folium # map rendering library

print('Libraries imported.')
```

```
Libraries imported.
```

2. Download and Explore Dataset

In order to segement the neighborhoods and explore them, we will essentially need a dataset that contains the 5 boroughs and the neighborhoods that exist in each borough as well as the the latitude and longitude coordinates of each neighborhood. I have used the prepared files from IBM cloud server and simply run a `wget` command to access the data.

In [2]:
```
!wget -q -O 'newyork_data.json' https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DS0701EN-SkillsNetwork/labs/newyork_data.json
print('Data downloaded!')
```

```
Data downloaded!
```

Load and explore the data

Next, let's load the data.

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

Let's take a quick look at the data.

In [4]:
```
newyork_data
```

```
    'id': 'nyu_2451_34572.3',

   'geometry': {'type': 'Point',
    'coordinates': [-73.82780644716412, 40.887555677350775]},
   'geometry_name': 'geom',
   'properties': {'name': 'Eastchester',
    'stacked': 1,
    'annoline1': 'Eastchester',
    'annoline2': None,
    'annoline3': None,
    'annoangle': 0.0,
    'borough': 'Bronx',
    'bbox': [-73.82780644716412,
     40.887555677350775,
     -73.82780644716412,
     40.887555677350775]}},
  {'type': 'Feature',
   'id': 'nyu_2451_34572.4',
   'geometry': {'type': 'Point',
    'coordinates': [-73.90564259591682, 40.89543742690383]},
```

Notice how all the relevant data is in the *features* key, which is basically a list of the neighborhoods. So, let's define a new variable that includes this data.

In [5]:
```
neighborhoods_data = newyork_data['features']
```

Let's take a look at the first item in this list.

In [6]:
```
neighborhoods_data[0]
```

Out[6]:
```
{'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]}}
```

Tranform the data into a *pandas* dataframe

Now we need to transform this data of nested Python dictionaries into a *pandas* dataframe. So, let's first create an empty dataframe.

In [7]:
```
# define the dataframe columns
column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

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

Let's look at the empty dataframe to confirm that the columns are as intended.

In [8]:
```
neighborhoods
```

Out[8]:

Borough	Neighborhood	Latitude	Longitude

Then let's loop through the data and fill the dataframe one row at a time.

In [9]:
```
for data in neighborhoods_data:
    borough  = 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)
```

Now just quickly examine the resulting dataframe.

In [10]:
```
neighborhoods.tail()
```

Out[10]:

	Borough	Neighborhood	Latitude	Longitude
301	Manhattan	Hudson Yards	40.756658	-74.000111
302	Queens	Hammels	40.587338	-73.805530
303	Queens	Bayswater	40.611322	-73.765968
304	Queens	Queensbridge	40.756091	-73.945631
305	Staten Island	Fox Hills	40.617311	-74.081740

In [11]:
```
neighborhoods.shape
```

Out[11]:
```
(306, 4)
```

And make sure that the dataset has all 5 boroughs and 306 neighborhoods.

In [12]:
```
print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(neighborhoods['Borough'].unique()),
        neighborhoods.shape[0]
    )
)
```

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

Use geopy library to get the latitude and longitude values of New York City.

In order to define an instance of the geocoder, we need to define a user_agent. We will name our agent *ny_explorer*, as shown below.

In [13]:
```
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.
```

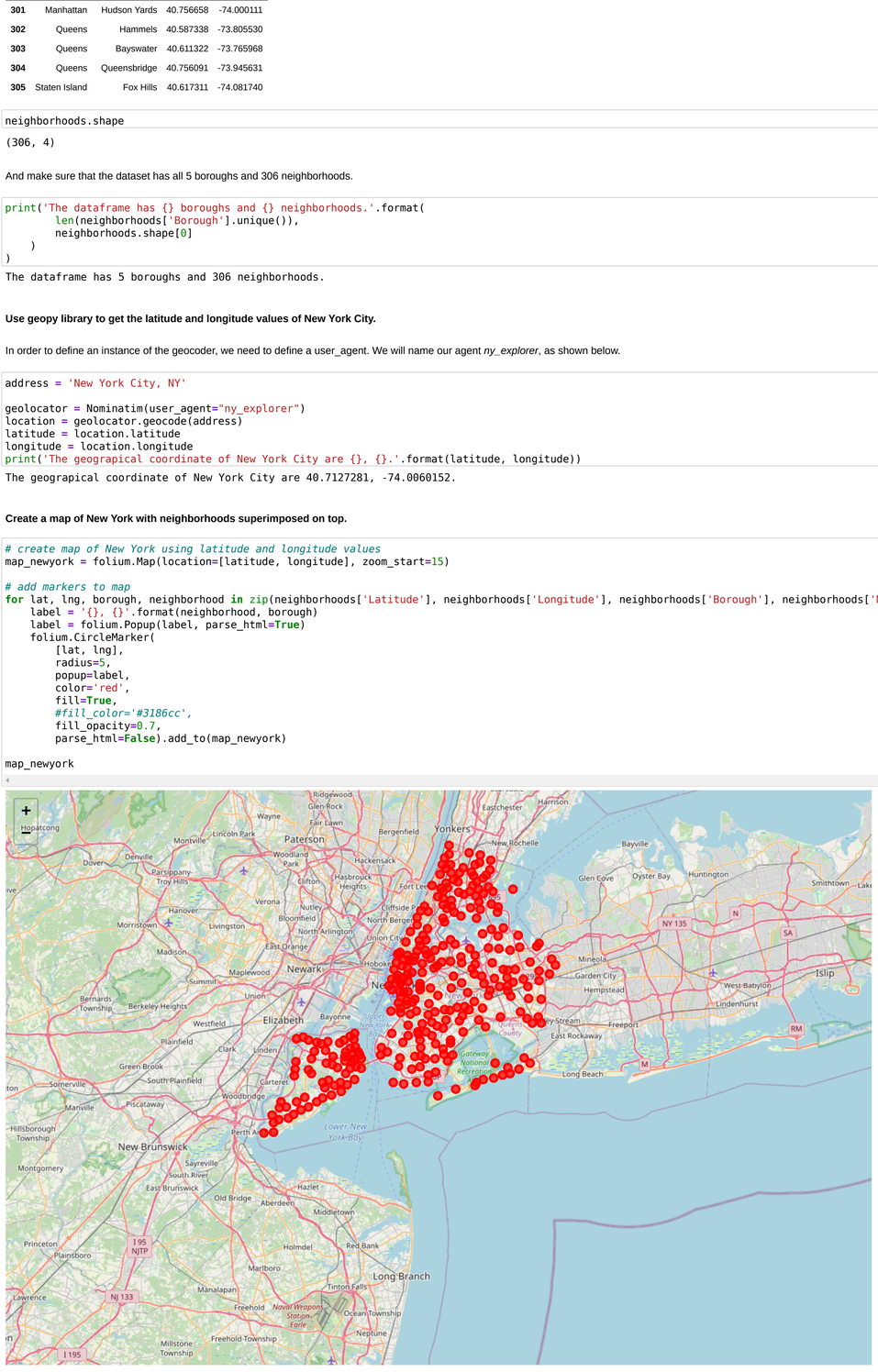
Create a map of New York with neighborhoods superimposed on top.

In [14]:
```
# create map of New York using latitude and longitude values
map_newyork = folium.Map(location=[latitude, longitude], zoom_start=15)

# 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='red',
        fill=True,
        #fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_newyork)

map_newyork
```

Out[14]:



Leaflet (http://leafletjs.com)

Folium is a great visualization library. Feel free to zoom into the above map, and click on each circle mark to reveal the name of the neighborhood and its respective borough.

However, for illustration purposes, let's simplify the above map and segment and cluster only the neighborhoods in Manhattan. So let's slice the original dataframe and create a new dataframe of the Manhattan data.

In [15]:
```
manhattan_data = neighborhoods[neighborhoods['Borough'] == 'Manhattan'].reset_index(drop=True)
manhattan_data.head()
```

Out[15]:

	Borough	Neighborhood	Latitude	Longitude
0	Manhattan	Marble Hill	40.876551	-73.910660
1	Manhattan	Chinatown	40.715618	-73.994279
2	Manhattan	Washington Heights	40.851903	-73.936900
3	Manhattan	Inwood	40.867684	-73.921210
4	Manhattan	Hamilton Heights	40.823604	-73.949688

In [16]:
```
manhattan_data.shape
```

Out[16]:
```
(40, 4)
```

Let's get the geographical coordinates of Manhattan.

In [17]:
```
address = 'Manhattan'

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

```
The geograpical coordinate of Manhattan are 40.7896239, -73.9598939.
```

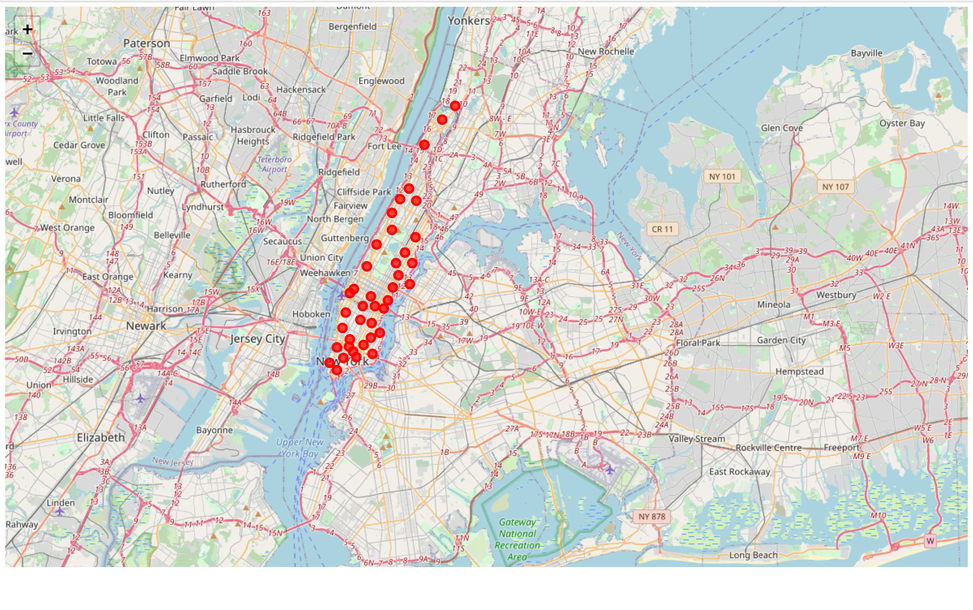
As we did with all of New York City, let's visualizat Manhattan the neighborhoods in it.

In [18]:
```
# create map of Manhattan using latitude and longitude values
map_manhattan = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(manhattan_data['Latitude'], manhattan_data['Longitude'], manhattan_data['Neighborhood']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='red',
        fill=True,
        #fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_manhattan)

map_manhattan
```

Out[18]:



Next, we are going to start utilizing the Foursquare API to explore the neighborhoods and segment them.

3. Methodology

The objective of this project is to find areas of Manhattan with a low density of coffee shops.

- First, we will check the candidate neighborhoods. It will be created the latitude & longitude coordinates for centroids of our candidate neighborhoods. We will create a grid of cells covering our area of interest which is centered around the Neighborhood Upper West Side.
- Secondly, we will get the top 100 venues that are in Central Park within a radius of 3.5 kilometers and explore their neighborhods.

In [19]:
```
manhattan_data.loc[manhattan_data['Neighborhood']== 'Upper West Side']
```

Out[19]:

	Borough	Neighborhood	Latitude	Longitude
12	Manhattan	Upper West Side	40.787658	-73.977059

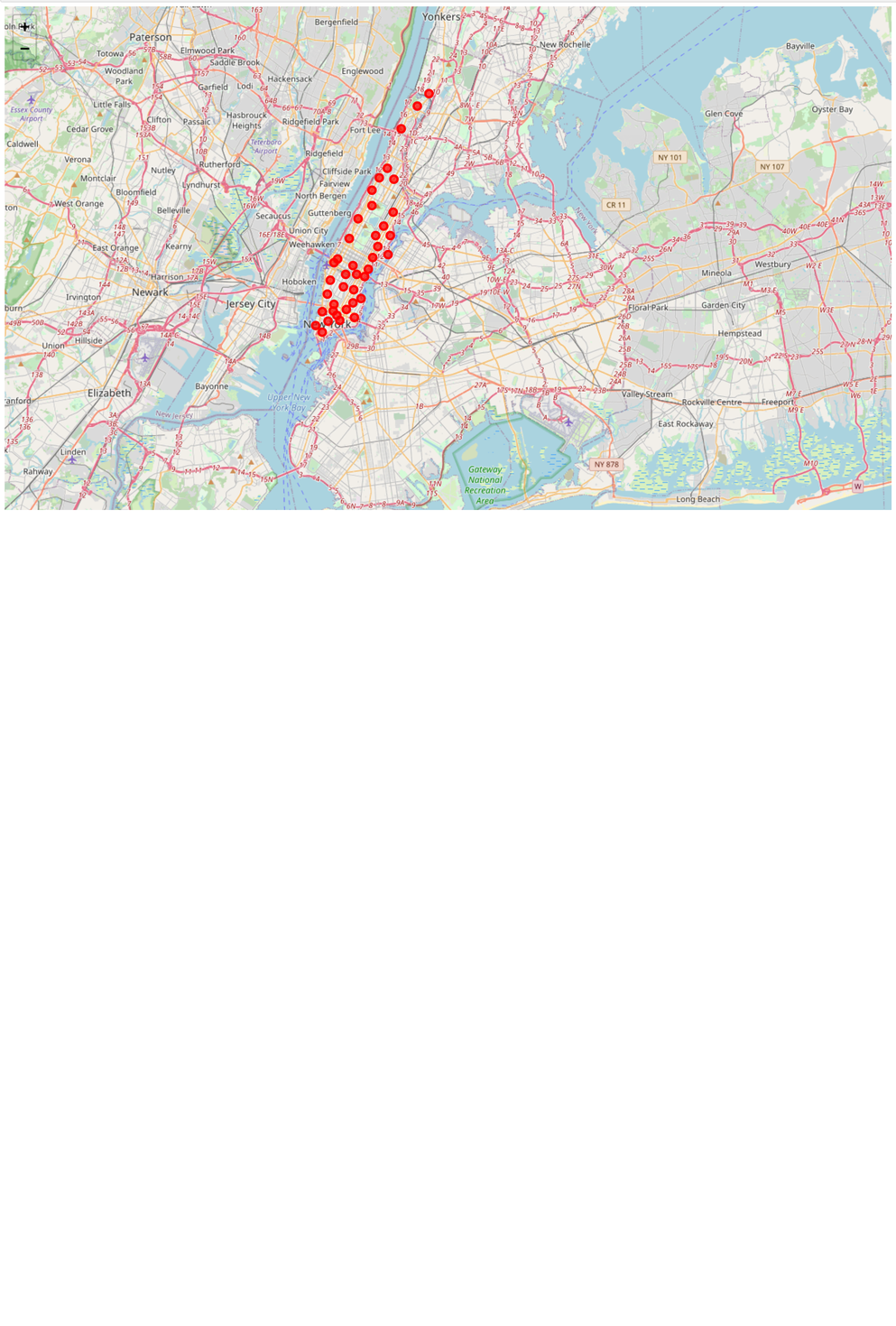
In [20]:
```
# create map of Montreal using latitude and longitude values
latitude = 40.787658
longitude = -73.977059

map_manhattan = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(manhattan_data['Latitude'], manhattan_data['Longitude'], manhattan_data['Neighborhood']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='red',
        fill=True,
        #fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_manhattan)

map_manhattan
```

Out[20]:



Now, we are going to start utilizing the Foursquare API to explore the neighborhoods and segment them.

Define Foursquare Credentials and Version

In [21]:
```
CLIENT_ID = 'WSXE233GWXGSUAOVKYCT4LBDMMQ2G4GY1BGK5JUD0GHNPZPJ' # your Foursquare ID
CLIENT_SECRET = 'CVC44UHNFX2SCREGZB0JBXI3045J3PFIYD1U4PG4T1A3ZVJZ' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
LIMIT = 100 # A default Foursquare API limit value

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

```
Your credentails:
CLIENT_ID: WSXE233GWXGSUAOVKYCT4LBDMMQ2G4GY1BGK5JUD0GHNPZPJ
CLIENT_SECRET:CVC44UHNFX2SCREGZB0JBXI3045J3PFIYD1U4PG4T1A3ZVJZ
```

Let's explore the first neighborhood in our dataframe.

Get the neighborhood's name.

In [22]:
```
manhattan_data.loc[0, 'Neighborhood']
```

Out[22]:
```
'Marble Hill'
```

Get the neighborhood's latitude and longitude values.

In [23]:
```
neighborhood_latitude = manhattan_data.loc[0, 'Latitude'] # neighborhood latitude value
neighborhood_longitude = manhattan_data.loc[0, 'Longitude'] # neighborhood longitude value

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

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

```
Latitude and longitude values of Marble Hill are 40.87655077879964, -73.91065965862981.
```

4. Explore Neighborhoods in Manhattan

Let's create a function to get 100 venues of the neighborhoods within a radius of 3.5kms

In [24]:
```
def getNearbyVenues(names, latitudes, longitudes, radius=3500):

    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)
```

In [ ]:
```
```

Now write the code to run the above function on each neighborhood and create a new dataframe called *manhattan_venues*.

In [25]:
```
# type your answer here
manhattan_venues = getNearbyVenues(names=manhattan_data['Neighborhood'],
                                   latitudes=manhattan_data['Latitude'],
                                   longitudes=manhattan_data['Longitude']
                                  )
```

```
Marble Hill
Chinatown
Washington Heights
Inwood
Hamilton Heights
Manhattanville
Central Harlem
East Harlem
Upper East Side
Yorkville
Lenox Hill
Roosevelt Island
Upper West Side
Lincoln Square
Clinton
Midtown
Murray Hill
Chelsea
Greenwich Village
```

Let's check the size of the resulting dataframe

In [26]:
```
print(manhattan_venues.shape)
manhattan_venues.head()
```

```
(4000, 7)
```

Out[26]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Marble Hill	40.876551	-73.91066	Bikram Yoga	40.876844	-73.906204	Yoga Studio
1	Marble Hill	40.876551	-73.91066	Tibbett Diner	40.880404	-73.908937	Diner
2	Marble Hill	40.876551	-73.91066	Sam's Pizza	40.879435	-73.905859	Pizza Place
3	Marble Hill	40.876551	-73.91066	Garden Gourmet Market	40.881350	-73.903389	Gourmet Shop
4	Marble Hill	40.876551	-73.91066	El Malecon	40.879338	-73.904457	Caribbean Restaurant

Let's check how many venues were returned for each neighborhood

In [27]:
```
manhattan_venues.groupby('Neighborhood').count()
```

Out[27]:

Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Battery Park City	100	100	100	100	100	100
Carnegie Hill	100	100	100	100	100	100
Central Harlem	100	100	100	100	100	100
Chelsea	100	100	100	100	100	100
Chinatown	100	100	100	100	100	100
Civic Center	100	100	100	100	100	100
Clinton	100	100	100	100	100	100
East Harlem	100	100	100	100	100	100
East Village	100	100	100	100	100	100
Financial District	100	100	100	100	100	100
Flatiron	100	100	100	100	100	100
Gramercy	100	100	100	100	100	100
Greenwich Village	100	100	100	100	100	100
Hamilton Heights	100	100	100	100	100	100
Hudson Yards	100	100	100	100	100	100
Inwood	100	100	100	100	100	100
Lenox Hill	100	100	100	100	100	100
Lincoln Square	100	100	100	100	100	100
Little Italy	100	100	100	100	100	100
Lower East Side	100	100	100	100	100	100
Manhattan Valley	100	100	100	100	100	100
Manhattanville	100	100	100	100	100	100
Marble Hill	100	100	100	100	100	100
Midtown	100	100	100	100	100	100
Midtown South	100	100	100	100	100	100
Morningside Heights	100	100	100	100	100	100
Murray Hill	100	100	100	100	100	100
Noho	100	100	100	100	100	100
Roosevelt Island	100	100	100	100	100	100
Soho	100	100	100	100	100	100
Stuyvesant Town	100	100	100	100	100	100
Sutton Place	100	100	100	100	100	100
Tribeca	100	100	100	100	100	100
Tudor City	100	100	100	100	100	100
Turtle Bay	100	100	100	100	100	100
Upper East Side	100	100	100	100	100	100
Upper West Side	100	100	100	100	100	100
Washington Heights	100	100	100	100	100	100
West Village	100	100	100	100	100	100
Yorkville	100	100	100	100	100	100

Let's find out how many unique categories can be curated from all the returned venues

In [28]:
```
print('There are {} uniques categories.'.format(len(manhattan_venues['Venue Category'].unique())))
```

```
There are 202 uniques categories.
```

5. Analyze Each Neighborhood

In [29]:
```
# one hot encoding
manhattan_onehot = pd.get_dummies(manhattan_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
manhattan_onehot['Neighborhood'] = manhattan_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [manhattan_onehot.columns[-1]] + list(manhattan_onehot.columns[:-1])
manhattan_onehot = manhattan_onehot[fixed_columns]

manhattan_onehot.head()
```

Out[29]:

	Neighborhood	African Restaurant	American Restaurant	Animal Shelter	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	Asian Restaurant	Athletics & Sports	Australian Restaurant	BBQ Joint	Bagel Shop	Bakery	Bar	Baseball Field	Baseball Stadium	Basketball Court	Beach	Beer Bar	Beer Store	Bike Trail	Bistro	Bookstore	Botanical Garden
0	Marble Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	Marble Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Marble Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Marble Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Marble Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	Boutique	Boxing Gym	Brewery	Bridge	Building	Burger Joint	Burrito Place	Butcher	Café	Caribbean Restaurant	Castle	Cheese Shop	Chinese Restaurant	Chocolate Shop	Church	Climbing Gym	Clothing Store	Club House	Cocktail Bar	Coffee Shop	College Arts Building	Comedy Club	Comfort Food Restaurant	Community Center
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

And let's examine the new dataframe size.

In [30]:
```
manhattan_onehot.shape
```

Out[30]:
```
(4000, 203)
```

Next, let's group rows by neighborhood and by taking the mean of the frequency of occurrence of each category

In [31]:
```
manhattan_grouped = manhattan_onehot.groupby('Neighborhood').mean().reset_index()
manhattan_grouped
```

Out[31]:

	Neighborhood	African Restaurant	American Restaurant	Animal Shelter	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	Asian Restaurant	Athletics & Sports	Australian Restaurant	BBQ Joint	Bagel Shop	Bakery	Bar	Baseball Field	Baseball Stadium	Basketball Court	Beach	Beer Bar	Beer Store	Bike Trail	Bistro	Bookstore	Botanical Garden
0	Battery Park City	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.00
1	Carnegie Hill	0.00	0.02	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.01	0.04	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00
2	Central Harlem	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.04	0.03	0.01	0.01	0.01	0.00	0.02	0.00	0.01	0.00	0.00	0.00
3	Chelsea	0.00	0.03	0.00	0.00	0.00	0.03	0.01	0.00	0.00	0.00	0.00	0.01	0.04	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.00
4	Chinatown	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.00	0.03	0.00
5	Civic Center	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.02	0.00
6	Clinton	0.00	0.00	0.00	0.00	0.00	0.04	0.01	0.00	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00
7	East Harlem	0.01	0.02	0.01	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.02	0.05	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00
8	East Village	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.05	0.00
9	Financial District	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.00
10	Flatiron	0.00	0.02	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.04	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.03	0.00
11	Gramercy	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.06	0.00
12	Greenwich Village	0.00	0.02	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.04	0.00
13	Hamilton Heights	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.05	0.02	0.02	0.02	0.01	0.00	0.02	0.00	0.01	0.00	0.01	0.00
14	Hudson Yards	0.00	0.01	0.00	0.00	0.00	0.04	0.02	0.00	0.00	0.00	0.00	0.01	0.03	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00
15	Inwood	0.00	0.02	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.04	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.00
16	Lenox Hill	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00
17	Lincoln Square	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	Little Italy	0.00	0.01	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.04	0.00
19	Lower East Side	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.05	0.00
20	Manhattan Valley	0.02	0.03	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.04	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.03	0.00
21	Manhattanville	0.02	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.04	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.02	0.00
22	Marble Hill	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.02	0.02	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.01
23	Midtown	0.00	0.01	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.00
24	Midtown South	0.00	0.03	0.00	0.00	0.00	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.04	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.03	0.00
25	Morningside Heights	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.02	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.02	0.00
26	Murray Hill	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.00
27	Noho	0.00	0.02	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.05	0.00
28	Roosevelt Island	0.00	0.00	0.00	0.01	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.02	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00
29	Soho	0.00	0.02	0.00	0.00	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.03	0.00
30	Stuyvesant Town	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.05	0.00
31	Sutton Place	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00
32	Tribeca	0.00	0.02	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.03	0.00
33	Tudor City	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00
34	Turtle Bay	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00
35	Upper East Side	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.06	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00
36	Upper West Side	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00
37	Washington Heights	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.01	0.03	0.03	0.00	0.02	0.02	0.01	0.00	0.00	0.00	0.01	0.01	0.00
38	West Village	0.00	0.03	0.00	0.00	0.00	0.02	0.02	0.00	0.00	0.01	0.00	0.01	0.04	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.04	0.00
39	Yorkville	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.01	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00

	Neighborhood	Boutique	Boxing Gym	Brewery	Bridge	Building	Burger Joint	Burrito Place	Butcher	Café	Caribbean Restaurant	Castle	Cheese Shop	Chinese Restaurant	Chocolate Shop	Church	Climbing Gym	Clothing Store	Club House	Cocktail Bar	Coffee Shop	College Arts Building	Comedy Club	Comfort Food Restaurant	Community Center
0	Battery Park City	0.02	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.03	0.00	0.00	0.00	0.00
1	Carnegie Hill	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.00	0.01	0.00
2	Central Harlem	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.05	0.07	0.00	0.00	0.00	0.00
3	Chelsea	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.02	0.00	0.00
4	Chinatown	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.01	0.02	0.00	0.02	0.00	0.00
5	Civic Center	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.02	0.04	0.00	0.02	0.00	0.00
6	Clinton	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.01
7	East Harlem	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.03	0.00	0.00	0.00	0.01
8	East Village	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.01	0.01	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.02	0.00	0.00
9	Financial District	0.01	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.03	0.03	0.00	0.00	0.00	0.00
10	Flatiron	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.02	0.00	0.00
11	Gramercy	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.02	0.00	0.00
12	Greenwich Village	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.02	0.00	0.00
13	Hamilton Heights	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.04	0.07	0.00	0.00	0.00	0.00
14	Hudson Yards	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.01
15	Inwood	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
16	Lenox Hill	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.01	0.00
17	Lincoln Square	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.00	0.01
18	Little Italy	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.02	0.00	0.02	0.00	0.00
19	Lower East Side	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.03	0.01	0.00	0.02	0.00	0.00
20	Manhattan Valley	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.01
21	Manhattanville	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.03	0.09	0.00	0.00	0.00	0.00
22	Marble Hill	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
23	Midtown	0.01	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01
24	Midtown South	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00
25	Morningside Heights	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.08	0.00	0.00	0.00	0.00
26	Murray Hill	0.00	0.01	0.01	0.00	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.01
27	Noho	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.01	0.02	0.01	0.00	0.02	0.00	0.00
28	Roosevelt Island	0.02	0.01	0.02	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.01	0.00	0.01	0.01	0.01	0.00	0.00	0.02	0.01	0.00	0.00	0.01	0.00
29	Soho	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.03	0.00	0.02	0.00	0.00
30	Stuyvesant Town	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.04	0.02	0.00	0.00	0.00	0.00
31	Sutton Place	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.01	0.01	0.00	0.00	0.01	0.01
32	Tribeca	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.02	0.00	0.00
33	Tudor City	0.00	0.02	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.01	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00
34	Turtle Bay	0.01	0.02	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.01	0.01	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.01
35	Upper East Side	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.03	0.00	0.00	0.01	0.00
36	Upper West Side	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00
37	Washington Heights	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
38	West Village	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.04	0.00	0.02	0.00	0.00
39	Yorkville	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.01	0.00

In [32]:
```
manhattan_grouped.describe()
```

Out[32]:

	African Restaurant	American Restaurant	Animal Shelter	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	Asian Restaurant	Athletics & Sports	Australian Restaurant	BBQ Joint	Bagel Shop	Bakery	Bar	Baseball Field	Baseball Stadium	Basketball Court	Beach	Beer Bar	Beer Store	Bike Trail
count	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000
mean	0.002750	0.013750	0.000250	0.000250	0.001000	0.009250	0.010250	0.000750	0.001250	0.001500	0.001000	0.009000	0.026500	0.009000	0.000750	0.001250	0.001500	0.001500	0.008500	0.004000	0.001000
std	0.007157	0.009789	0.001581	0.001581	0.003038	0.010715	0.006975	0.002667	0.003349	0.003616	0.003038	0.005905	0.014242	0.010328	0.003499	0.004634	0.004267	0.003616	0.006622	0.004961	0.003038
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.010000	0.000000	0.000000	0.000000	0.000000	0.010000	0.000000	0.000000	0.000000	0.000000	0.010000	0.020000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.010000	0.000000	0.000000	0.000000	0.010000	0.010000	0.000000	0.000000	0.000000	0.000000	0.010000	0.020000	0.010000	0.000000	0.000000	0.000000	0.000000	0.010000	0.000000	0.000000
75%	0.000000	0.020000	0.000000	0.000000	0.000000	0.010000	0.010000	0.000000	0.000000	0.000000	0.000000	0.010000	0.040000	0.012500	0.000000	0.000000	0.000000	0.000000	0.010000	0.010000	0.000000
max	0.030000	0.030000	0.010000	0.010000	0.010000	0.040000	0.030000	0.010000	0.010000	0.010000	0.010000	0.020000	0.060000	0.040000	0.020000	0.020000	0.020000	0.010000	0.020000	0.010000	0.010000

	Bistro	Bookstore	Botanical Garden	Boutique	Boxing Gym	Brewery	Bridge	Building	Burger Joint	Burrito Place	Butcher	Café	Caribbean Restaurant	Castle	Cheese Shop	Chinese Restaurant	Chocolate Shop	Church	Climbing Gym	Clothing Store	H
count	40.000000	40.000000	40.000000	40.000000	40.000000	40.00000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.00
mean	0.000750	0.022750	0.000250	0.004250	0.004500	0.00250	0.001750	0.000500	0.005500	0.000250	0.001500	0.014750	0.000750	0.000750	0.006500	0.004500	0.003750	0.002500	0.002500	0.000500	0.00
std	0.002667	0.015357	0.001581	0.007472	0.006385	0.00543	0.003848	0.002207	0.007494	0.001581	0.003616	0.009055	0.002667	0.002667	0.006998	0.006385	0.004903	0.004385	0.004385	0.002207	0.00
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
25%	0.000000	0.010000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.010000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
50%	0.000000	0.020000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.010000	0.000000	0.000000	0.010000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
75%	0.000000	0.030000	0.000000	0.010000	0.010000	0.00000	0.000000	0.000000	0.010000	0.000000	0.000000	0.020000	0.000000	0.000000	0.010000	0.010000	0.010000	0.002500	0.002500	0.000000	0.00
max	0.010000	0.060000	0.010000	0.020000	0.020000	0.02000	0.010000	0.010000	0.030000	0.010000	0.010000	0.040000	0.010000	0.010000	0.020000	0.020000	0.010000	0.010000	0.010000	0.010000	0.01

Let's confirm the new size

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

Out[33]:
```
(40, 203)
```

In [34]:
```
manhattan_part = manhattan_grouped[['Neighborhood', 'Coffee Shop']]
manhattan_part
```

Out[34]:

	Neighborhood	Coffee Shop
0	Battery Park City	0.03
1	Carnegie Hill	0.02
2	Central Harlem	0.07
3	Chelsea	0.03
4	Chinatown	0.02
5	Civic Center	0.04
6	Clinton	0.02
7	East Harlem	0.03
8	East Village	0.01
9	Financial District	0.03
10	Flatiron	0.02
11	Gramercy	0.01
12	Greenwich Village	0.01
13	Hamilton Heights	0.07
14	Hudson Yards	0.02
15	Inwood	0.01
16	Lenox Hill	0.01
17	Lincoln Square	0.02
18	Little Italy	0.02
19	Lower East Side	0.01
20	Manhattan Valley	0.04
21	Manhattanville	0.09
22	Marble Hill	0.02
23	Midtown	0.00
24	Midtown South	0.02
25	Morningside Heights	0.08
26	Murray Hill	0.02
27	Noho	0.01
28	Roosevelt Island	0.01
29	Soho	0.03
30	Stuyvesant Town	0.02
31	Sutton Place	0.01
32	Tribeca	0.03
33	Tudor City	0.00
34	Turtle Bay	0.00
35	Upper East Side	0.03
36	Upper West Side	0.01
37	Washington Heights	0.02
38	West Village	0.04
39	Yorkville	0.02

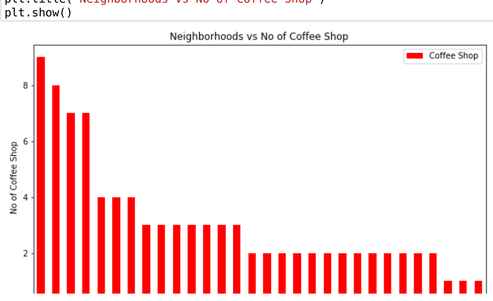
In [35]:
```
manhattan_merged = pd.merge(manhattan_data, manhattan_part, on='Neighborhood')
manhattan_merged.head()
```

Out[35]:

	Borough	Neighborhood	Latitude	Longitude	Coffee Shop
0	Manhattan	Marble Hill	40.876551	-73.910660	0.02
1	Manhattan	Chinatown	40.715618	-73.994279	0.02
2	Manhattan	Washington Heights	40.851903	-73.936900	0.02
3	Manhattan	Inwood	40.867684	-73.921210	0.01
4	Manhattan	Hamilton Heights	40.823604	-73.949688	0.07

Bar chart of coffee shops

In [36]:
```
import matplotlib.pyplot as plt
graph = pd.DataFrame(manhattan_onehot.groupby('Neighborhood')['Coffee Shop'].sum())
graph = graph.sort_values(by ='Coffee Shop', ascending=False)
graph.iloc[:30].plot(kind='bar', figsize=(10,6),color= 'red')
plt.xlabel("Neighborhoods")
plt.ylabel("No of Coffee Shop")
plt.title("Neighborhoods vs No of Coffee Shop")
plt.show()
```



In [37]:
```
graph
```

Out[37]:

Neighborhood	Coffee Shop
Manhattanville	9
Morningside Heights	8
Hamilton Heights	7
Central Harlem	7
Manhattan Valley	4
West Village	4
Civic Center	4
Upper East Side	3
Tribeca	3
Soho	3
Battery Park City	3
East Harlem	3
Chelsea	3
Financial District	3
Yorkville	2
Murray Hill	2
Clinton	2
Midtown South	2
Chinatown	2
Marble Hill	2
Carnegie Hill	2
Little Italy	2
Lincoln Square	2
Washington Heights	2
Hudson Yards	2
Flatiron	2
Stuyvesant Town	2
Upper West Side	1
Sutton Place	1
Gramercy	1
Roosevelt Island	1
Noho	1
Greenwich Village	1
Lower East Side	1
Lenox Hill	1
Inwood	1
East Village	1
Tudor City	0
Turtle Bay	0
Midtown	0

In [38]:
```
coffee_df = manhattan_grouped[['Neighborhood','Coffee Shop']]

coffee_df.head()
```

Out[38]:

	Neighborhood	Coffee Shop
0	Battery Park City	0.03
1	Carnegie Hill	0.02
2	Central Harlem	0.07
3	Chelsea	0.03
4	Chinatown	0.02

5. Cluster Neighborhoods

In [39]:
```
df_clustering = coffee_df.drop('Neighborhood', 1)
```

In [40]:
```
from sklearn.cluster import KMeans
from yellowbrick.cluster.elbow import kelbow_visualizer
from yellowbrick.datasets.loaders import load_nfl

df_clustering, y = load_nfl()

# Use the quick method and immediately show the figure
kelbow_visualizer(KMeans(random_state=4), df_clustering, k=(2,10))
```


```
/home/lopa/.local/lib/python3.6/site-packages/sklearn/base.py:213: FutureWarning: From version 0.24, get_params will raise an AttributeError if a parameter cannot be retrieved as an instance attribute. Previously it would return None.
  FutureWarning)
```

Out[40]:
```
KElbowVisualizer(ax=<matplotlib.axes._subplots.AxesSubplot object at 0x7f0929e2cf60>,
                 estimator=KMeans(n_clusters=9, random_state=4), k=None)
```

Run *k*-means to cluster the neighborhood into 4 clusters.

In [41]:
```
# set number of clusters
kclusters = 4

coffee_grouped_clustering = coffee_df.drop('Neighborhood', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(coffee_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

Out[41]:
```
array([0, 3, 1, 0, 3, 0, 3, 0, 2, 0], dtype=int32)
```

In [42]:
```
#make a copy of manhattan_data

coffee_merged = coffee_df

# add clustering labels

coffee_merged['Cluster Labels'] = kmeans.labels_

# add latitude/longitude for each neighborhood
coffee_merged = coffee_merged.join(manhattan_data.set_index('Neighborhood'), on='Neighborhood')

coffee_merged.head() # check the last columns!
```

```
/home/lopa/.local/lib/python3.6/site-packages/ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  import sys
```

Out[42]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude
0	Battery Park City	0.03	0	Manhattan	40.711932	-74.016869
1	Carnegie Hill	0.02	3	Manhattan	40.782683	-73.953256
2	Central Harlem	0.07	1	Manhattan	40.815976	-73.943211
3	Chelsea	0.03	0	Manhattan	40.744035	-74.003116
4	Chinatown	0.02	3	Manhattan	40.715618	-73.994279

In [43]:
```
# Sort by cluster

coffee_merged.sort_values(['Cluster Labels'], inplace=True)

coffee_merged.head()
```

Out[43]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude
0	Battery Park City	0.03	0	Manhattan	40.711932	-74.016869
29	Soho	0.03	0	Manhattan	40.722184	-74.000657
38	West Village	0.04	0	Manhattan	40.734434	-74.006180
9	Financial District	0.03	0	Manhattan	40.707107	-74.010665
7	East Harlem	0.03	0	Manhattan	40.792249	-73.944182

In [44]:
```
coffee_merged = coffee_merged.reset_index(drop=True)
coffee_merged.head()
```

Out[44]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude
0	Battery Park City	0.03	0	Manhattan	40.711932	-74.016869
1	Soho	0.03	0	Manhattan	40.722184	-74.000657
2	West Village	0.04	0	Manhattan	40.734434	-74.006180
3	Financial District	0.03	0	Manhattan	40.707107	-74.010665
4	East Harlem	0.03	0	Manhattan	40.792249	-73.944182

In [45]:
```
coffee_merged.shape
```

Out[45]:
```
(40, 6)
```

Finally, let's visualize the resulting clusters

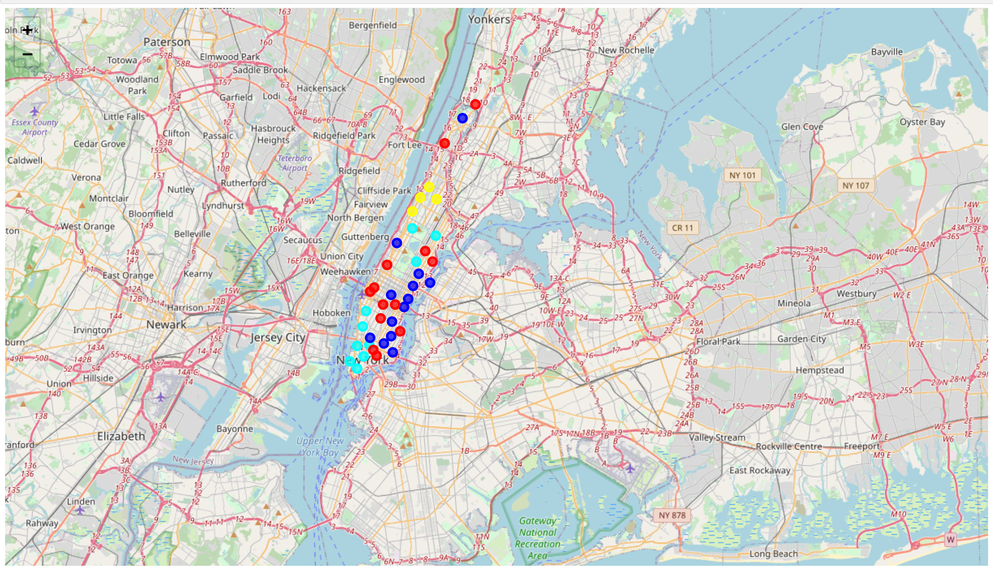
In [46]:
```
# create map
latitude = 40.787658
longitude = -73.977059
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
#colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
#rainbow = [colors.rgb2hex(i) for i in colors_array]
rainbow = ['yellow','blue','red','cyan']

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(coffee_merged['Latitude'], coffee_merged['Longitude'], coffee_merged['Neighborhood'], coffee_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

Out[46]:



5. Examine Clusters

In [47]:
```
# minimum frequency of coffee shop
coffee_merged['Coffee Shop'].min()
```

Out[47]:
```
0.0
```

In [48]:
```
# maximum frequency of coffee shop
coffee_merged['Coffee Shop'].max()
```

Out[48]:
```
0.09
```

Cluster 1

In [49]:
```
coffee_merged.loc[coffee_merged['Cluster Labels'] == 0]
```

Out[49]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude
0	Battery Park City	0.03	0	Manhattan	40.711932	-74.016869
1	Soho	0.03	0	Manhattan	40.722184	-74.000657
2	West Village	0.04	0	Manhattan	40.734434	-74.006180
3	Financial District	0.03	0	Manhattan	40.707107	-74.010665
4	East Harlem	0.03	0	Manhattan	40.792249	-73.944182
5	Tribeca	0.03	0	Manhattan	40.721522	-74.010683
6	Upper East Side	0.03	0	Manhattan	40.775639	-73.960508
7	Chelsea	0.03	0	Manhattan	40.744035	-74.003116
8	Civic Center	0.04	0	Manhattan	40.715229	-74.005415
9	Manhattan Valley	0.04	0	Manhattan	40.797307	-73.964286

Cluster 2

In [50]:
```
coffee_merged.loc[coffee_merged['Cluster Labels'] ==1]
```

Out[50]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude
10	Central Harlem	0.07	1	Manhattan	40.815976	-73.943211
11	Hamilton Heights	0.07	1	Manhattan	40.823604	-73.949688
12	Morningside Heights	0.08	1	Manhattan	40.808000	-73.963896
13	Manhattanville	0.09	1	Manhattan	40.816934	-73.957385

Cluster 3

In [51]:
```
coffee_merged.loc[coffee_merged['Cluster Labels'] ==2]
```

Out[51]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude
14	Turtle Bay	0.00	2	Manhattan	40.752042	-73.967708
15	Tudor City	0.00	2	Manhattan	40.746917	-73.971219
16	Sutton Place	0.01	2	Manhattan	40.760280	-73.963556
17	Upper West Side	0.01	2	Manhattan	40.787658	-73.977059
18	Roosevelt Island	0.01	2	Manhattan	40.762160	-73.949168
19	Noho	0.01	2	Manhattan	40.723259	-73.988434
20	Midtown	0.00	2	Manhattan	40.754691	-73.981669
21	Lower East Side	0.01	2	Manhattan	40.717807	-73.980890
22	Lenox Hill	0.01	2	Manhattan	40.768113	-73.958860
23	Inwood	0.01	2	Manhattan	40.867684	-73.921210
24	Greenwich Village	0.01	2	Manhattan	40.726933	-73.999914
25	East Village	0.01	2	Manhattan	40.727847	-73.982226
26	Gramercy	0.01	2	Manhattan	40.737210	-73.981376

Cluster 4

In [52]:
```
coffee_merged.loc[coffee_merged['Cluster Labels'] ==3]
```

Out[52]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude
27	Marble Hill	0.02	3	Manhattan	40.876551	-73.910660
28	Washington Heights	0.02	3	Manhattan	40.851903	-73.936900
29	Carnegie Hill	0.02	3	Manhattan	40.782683	-73.953256
30	Chinatown	0.02	3	Manhattan	40.715618	-73.994279
31	Clinton	0.02	3	Manhattan	40.759101	-73.996119
32	Stuyvesant Town	0.02	3	Manhattan	40.731000	-73.974052
33	Little Italy	0.02	3	Manhattan	40.719324	-73.997305
34	Hudson Yards	0.02	3	Manhattan	40.756658	-74.000111
35	Murray Hill	0.02	3	Manhattan	40.748303	-73.978332
36	Midtown South	0.02	3	Manhattan	40.748510	-73.988713
37	Lincoln Square	0.02	3	Manhattan	40.773529	-73.985338
38	Flatiron	0.02	3	Manhattan	40.739673	-73.990947
39	Yorkville	0.02	3	Manhattan	40.775930	-73.947118

In [ ]:
```

```

In [53]:
```
# locations that are not already crowded with coffee shops
low_coffee = (coffee_merged.loc[coffee_merged['Cluster Labels'] == 2]).reset_index(drop=True)
low_coffee
```

Out[53]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude
0	Turtle Bay	0.00	2	Manhattan	40.752042	-73.967708
1	Tudor City	0.00	2	Manhattan	40.746917	-73.971219
2	Sutton Place	0.01	2	Manhattan	40.760280	-73.963556
3	Upper West Side	0.01	2	Manhattan	40.787658	-73.977059
4	Roosevelt Island	0.01	2	Manhattan	40.762160	-73.949168
5	Noho	0.01	2	Manhattan	40.723259	-73.988434
6	Midtown	0.00	2	Manhattan	40.754691	-73.981669
7	Lower East Side	0.01	2	Manhattan	40.717807	-73.980890
8	Lenox Hill	0.01	2	Manhattan	40.768113	-73.958860
9	Inwood	0.01	2	Manhattan	40.867684	-73.921210
10	Greenwich Village	0.01	2	Manhattan	40.726933	-73.999914

In [54]:
```
# locations that are  crowded with coffee shops
high_coffee = (coffee_merged.loc[coffee_merged['Cluster Labels'] == 1]).reset_index(drop=True)
high_coffee
```

Out[54]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude
0	Central Harlem	0.07	1	Manhattan	40.815976	-73.943211
1	Hamilton Heights	0.07	1	Manhattan	40.823604	-73.949688
2	Morningside Heights	0.08	1	Manhattan	40.808000	-73.963896
3	Manhattanville	0.09	1	Manhattan	40.816934	-73.957385

In [ ]:
```

```

In [55]:
```
from numpy import cos, sin, arcsin, sqrt
from math import radians

def haversine(row):
    lat1 = 40.787658
    lon1 = -73.977059
    lat2 = row['Latitude']
    lon2 = row['Longitude']
    lat1, lon1, lat2, lon2 = map(radians, [lat1, lon1, lat2, lon2])
    dlat = lat2 - lat1
    dlon = lon2 - lon1

    a = sin(dlon/2)**2 + cos(lon1) * cos(lon2) * sin(dlat/2)**2
    c = 2 * arcsin(sqrt(a))
    km = 6367 * c
    return km

low_coffee['Upper West Side (km)'] = low_coffee.apply(lambda row: haversine(row), axis=1)
low_coffee
```

Out[55]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude	Upper West Side (km)
0	Turtle Bay	0.00	2	Manhattan	40.752042	-73.967708	1.507924
1	Tudor City	0.00	2	Manhattan	40.746917	-73.971219	1.408285
2	Sutton Place	0.01	2	Manhattan	40.760280	-73.963556	1.719677
3	Upper West Side	0.01	2	Manhattan	40.787658	-73.977059	0.000026
4	Roosevelt Island	0.01	2	Manhattan	40.762160	-73.949168	3.196744
5	Noho	0.01	2	Manhattan	40.723259	-73.988434	2.344543
6	Midtown	0.00	2	Manhattan	40.754691	-73.981669	1.133421
7	Lower East Side	0.01	2	Manhattan	40.717807	-73.980890	2.184195
8	Lenox Hill	0.01	2	Manhattan	40.768113	-73.958860	2.109483
9	Inwood	0.01	2	Manhattan	40.867684	-73.921210	6.675504
10	Greenwich Village	0.01	2	Manhattan	40.726933	-73.999914	3.148803
11	East Village	0.01	2	Manhattan	40.727847	-73.982226	1.922074
12	Gramercy	0.01	2	Manhattan	40.737210	-73.981376	1.619858

In [56]:
```
low_coffee.sort_values(['Upper West Side (km)'])[:5]
```

Out[56]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude	Upper West Side (km)
3	Upper West Side	0.01	2	Manhattan	40.787658	-73.977059	0.000026
6	Midtown	0.00	2	Manhattan	40.754691	-73.981669	1.133421
1	Tudor City	0.00	2	Manhattan	40.746917	-73.971219	1.408285
0	Turtle Bay	0.00	2	Manhattan	40.752042	-73.967708	1.507924
12	Gramercy	0.01	2	Manhattan	40.737210	-73.981376	1.619858

In [57]:
```
#Minimum distance
distance_min = low_coffee.loc[low_coffee['Coffee Shop']== 0.00].reset_index(drop=True)
distance_min['Upper West Side (km)'].min()
```

Out[57]: 1.1334214504689875

In [58]:
```
distance_min['Upper West Side (km)'].idxmin()
```

Out[58]: 2

In [59]:
```
high_coffee['Upper West Side (km)'] = high_coffee.apply(lambda row: haversine(row), axis=1)
high_coffee
```

Out[59]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude	Upper West Side (km)
0	Central Harlem	0.07	1	Manhattan	40.815976	-73.943211	3.860539
1	Hamilton Heights	0.07	1	Manhattan	40.823604	-73.949688	3.235603
2	Morningside Heights	0.08	1	Manhattan	40.808000	-73.963896	1.590325
3	Manhattanville	0.09	1	Manhattan	40.816934	-73.957385	2.363677

In [60]:
```
high_coffee.loc[high_coffee['Coffee Shop']== 0.09].reset_index(drop=True)
```

Out[60]:

	Neighborhood	Coffee Shop	Cluster Labels	Borough	Latitude	Longitude	Upper West Side (km)
0	Manhattanville	0.09	1	Manhattan	40.816934	-73.957385	2.363677

6. Results

This work analyzed 40 neighborhoods in Manhattan with the objective of detecting places that are no longer full of coffee shops. The area with the lowest density of cafeterias in Montreal is concentrated in the peripheral region of the analyzed area. This region is related to cluster 3 (red dots on the map).In this cluster, 13 neighborhoods have a low or no cafeteria density. The neighborhoods where there is a high incidence of coffee shops are located in cluster 2, and they are appx 2-4 Km away to our central coordinate.

Midtown is the closest neighborhood to Upper West Side and has no coffee shop. It is 1.13 km away from the central coordinate. Tudor City and Turtle Bay are also the nearby places closer appx 1.4-1.5 Km with no coffee shops.

7. Conclusion

The neighborhoods where there is a high incidence of coffee shops are located in cluster 2, many of them are appx 2-4 Km away to the central coordinate. The neighborhoods with the highest coffee frequencies are Central Harlem, Hamilton Heights, Morningside Heights, Manhattanville. Project requirements include detecting places that are not already crowded with cafes, or areas with no cafes nearby, and as close as possible to the city center.The following three options are suggested that meet the project requirements : **Midtown**, the closest neighborhood and has no coffee shop with 1.13 km away from the central coordinate, Tudor City and Turtle Bay (1.4-1.5 Km).

In [ ]:
```

```