Introduction

This project will analyze the New York City data. First, we will find the most visited commercial shop according to the number of check-ins, then we will try to find the neighborhoods that are lacking the selected type of shop which could be potential business opportunity.

Target Audience

The target audience of this report is any one that is interested in opening a shop but have no idea what kind of and in which neighborhood.

Data Section

The data comes from **Dingqi Yang** from the following link https://sites.google.com/site/yangdingqi/home/foursquare-dataset (https://sites.google.com/site/yangdingqi/home/foursquare-dataset). It contains 227,428 check-ins in New York city. The data contains a file in tsv format. Each file contains 8 columns, which are:

1. User ID (anonymized)
2. Venue ID (Foursquare)
3. Venue category ID (Foursquare)
4. Venue category name (Foursquare)
5. Latitude
6. Longitude
7. Time zone offset in minutes (The offset in minutes between when this check-in occurred and the same time in UTC)
8. UTC time

Application

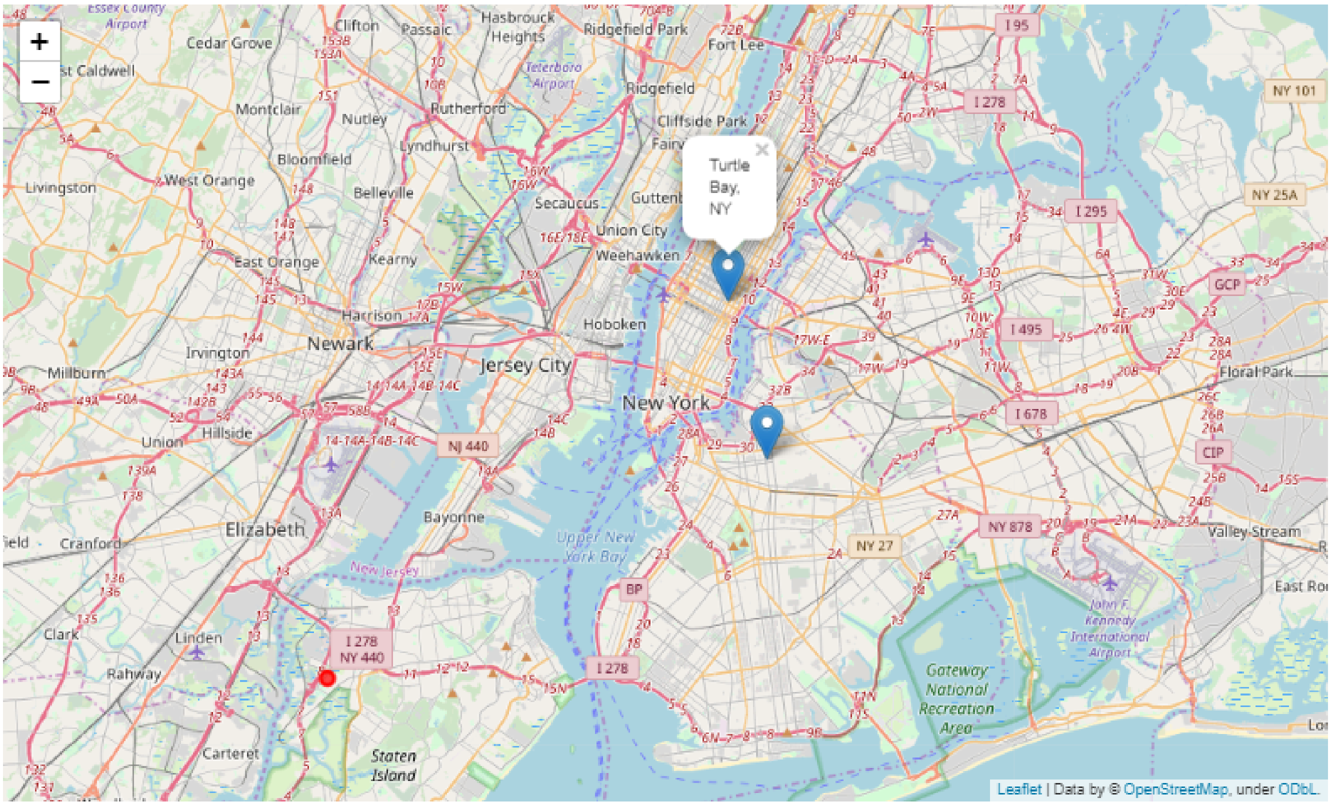
We will find the most visited type of shop (commercial) according to the number of check-ins given in the data, then we will try to find neighborhoods that has none of this type of shop.

Examples are for 2000 venues, and the red dot is the center neighborhood which has the most number of Bars between selected coordinates. We did find two neighborhoods that are closest to it having none Bars within 4 kilometers.

	VenueID	CategoryName	Visitor Count	Latitude	Longitude
0	49bbd6c0f964a520f4531fe3	Arts & Crafts Store	7	40.719810375488535	-74.00258103213994
1	4a43c0aef964a520c6a61fe3	Bridge	37	40.60679958140643	-74.04416981025437
2	4c5cc7b485a1e21e00d35711	Home (private)	1	40.716161684843215	-73.88307005845945
3	4bc7086715a7ef3bef9878da	Medical Center	1	40.7451638	-73.982518775
4	4cf2c5321d18a143951b5cec	Food Truck	4	40.74010382743943	-73.98965835571289

```
[('Train Station', 943), ('Park', 778), ('Airport', 769), ('Bar', 756), ('Subway', 587), ('Coffee Shop', 447), ('Gym / Fitness
Center', 447), ('Food & Drink Shop', 426), ('Neighborhood', 362), ('Plaza', 342), ('Stadium', 339), ('Bridge', 272), ('Office',
264), ('Department Store', 240), ('Mall', 238), ('Burger Joint', 206), ('American Restaurant', 202), ('Road', 201), ('Bus Stati
```

```
'Bar' is the most visited commercial category according to given data.
```



Starting The Project

In [6]:

```
# Import libaries
import sys
import numpy as np
import pandas as pd
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
import json
from geopy.geocoders import Nominatim
import requests
from pandas.io.json import json_normalize
import matplotlib.cm as cm
import matplotlib.colors as colors
from sklearn.cluster import KMeans
import folium
from collections import defaultdict
```

In [13]:

```
venues = defaultdict(list)
categories = {}
coordinates = []

with open('dataset_tsmc2014/dataset_TSMC2014_NYC.txt',encoding="ISO-8859-1") as
nyc_data:
    for venue in nyc_data.readlines():
        data = venue.split('\t')

        # get the coordinates for the shop
        coordinates.insert(len(coordinates), [data[4], data[5]])

        # store the shop id and the user ids
        if data[1] not in venues:
            venues[data[1]].append(data[0])
        elif data[0] not in venues[data[1]]:
            venues[data[1]].append(data[0])

        # store the type of the shop and its id
        if data[1] not in categories:
            categories[data[1]] = data[3]


# example for 5 places and their visitors according to check-ins
for i in range(5):
    print(list(venues)[i], ": ", len(venues.get(list(venues)[i])), "  --> ", ven
ues.get(list(venues)[i]))
    print()
```

```
49bbd6c0f964a520f4531fe3 :  7   -->  ['470', '1034', '445', '882',
'806', '878', '949']

4a43c0aef964a520c6a61fe3 :  37   -->  ['979', '319', '1047', '582',
'783', '43', '724', '2', '335', '508', '120', '784', '285', '699',
'716', '50', '721', '521', '160', '583', '590', '494', '1000', '94
9', '111', '29', '864', '580', '472', '322', '673', '343', '554', '8
44', '59', '398', '555']

4c5cc7b485a1e21e00d35711 :  1   -->  ['69']

4bc7086715a7ef3bef9878da :  1   -->  ['395']

4cf2c5321d18a143951b5cec :  4   -->  ['87', '977', '427', '372']
```

In [14]:

```
# define the dataframe columns
column_names = ['VenueID', 'CategoryName', 'Visitor Count', 'Latitude', 'Longitu
de']

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

venue_data
```

Out[14]:

VenueID	CategoryName	Visitor Count	Latitude	Longitude

In [15]:

```
# loop through the data and fill the dataframe one row at a time.
for idx in range(0, 2000):
    venue_id = list(venues)[idx]
    coords = list(coordinates)[idx]
    visitorCount = len(venues.get(venue_id))
    venue_type = categories.get(venue_id)


    venue_data = venue_data.append({ 'CategoryName': venue_type,
                                     'VenueID': venue_id,
                                     'Visitor Count': visitorCount,
                                     'Latitude': coords[0],
                                     'Longitude': coords[1]}, ignore_index=True)
```

In [16]:

```
# Quickly examine the resulting dataframe.
venue_data.head()
```

Out[16]:

	VenueID	CategoryName	Visitor Count	Latitude	Longitud
0	49bbd6c0f964a520f4531fe3	Arts & Crafts Store	7	40.719810375488535	-74.0025810321399
1	4a43c0aef964a520c6a61fe3	Bridge	37	40.60679958140643	-74.0441698102543
2	4c5cc7b485a1e21e00d35711	Home (private)	1	40.716161684843215	-73.8830700584594
3	4bc7086715a7ef3bef9878da	Medical Center	1	40.7451638	-73.98251877
4	4cf2c5321d18a143951b5cec	Food Truck	4	40.74010382743943	-73.9896583557128

In [27]:

```
venue_data['Visitor Count'].value_counts()
```

Out[27]:

```
Home (private)                              150
Office                                      126
Bar                                         101
Gym / Fitness Center                         91
Coffee Shop                                  79
Subway                                       75
College Academic Building                    62
Food & Drink Shop                            53
Other Great Outdoors                         48
Bus Station                                  43
Road                                         37
Deli / Bodega                                37
Building                                     36
Train Station                                33
Park                                         31
Medical Center                               31
Drugstore / Pharmacy                         30
Residential Building (Apartment / Condo)     29
Neighborhood                                 29
American Restaurant                          28
Sandwich Place                               26
Bank                                         24
Hotel                                        23
Sushi Restaurant                             23
Clothing Store                               21
Pizza Place                                  20
Mexican Restaurant                           19
BBQ Joint                                    19
Fast Food Restaurant                         18
Miscellaneous Shop                           17
University                                   17
Airport                                      17
Government Building                          16
Burger Joint                                 15
Café                                         14
Department Store                             14
Chinese Restaurant                           13
Theater                                      12
Italian Restaurant                           12
Salon / Barbershop                           12
Food Truck                                   12
School                                       11
General Entertainment                        11
Mall                                         11
Electronics Store                            10
Music Venue                                  10
Bakery                                       10
General Travel                               10
Laundry Service                              10
Church                                        9
Bridge                                        9
Vegetarian / Vegan Restaurant                 9
Donut Shop                                    9
Beach                                         8
Ferry                                         8
Ice Cream Shop                                8
Sporting Goods Shop                           7
Playground                                    7
High School                                   7
```

```
Seafood Restaurant                             7
Diner                                          7
Parking                                        7
French Restaurant                              7
Light Rail                                     6
Bagel Shop                                     6
Post Office                                    6
Latin American Restaurant                      6
Fried Chicken Joint                            6
Bookstore                                      6
Plaza                                          6
Scenic Lookout                                 6
Convention Center                              6
General College & University                   6
Cosmetics Shop                                 5
Stadium                                        5
Automotive Shop                                5
Athletic & Sport                               5
Japanese Restaurant                            5
Pet Store                                      5
Cupcake Shop                                   5
Gas Station / Garage                           5
Library                                        5
Gastropub                                      4
Community College                              4
Convenience Store                              4
Asian Restaurant                               4
Breakfast Spot                                 4
Student Center                                 4
Restaurant                                     4
Thai Restaurant                                4
Art Gallery                                    4
Indian Restaurant                              4
Record Shop                                    4
Event Space                                    4
Toy / Game Store                               4
Arts & Crafts Store                            4
Paper / Office Supplies Store                  4
Tattoo Parlor                                  3
Furniture / Home Store                         3
Spanish Restaurant                             3
Smoke Shop                                     3
Middle Eastern Restaurant                      3
Arcade                                         3
Comedy Club                                    3
Candy Store                                    3
Moving Target                                  3
Other Nightlife                                3
Movie Theater                                  3
Snack Place                                    3
Wings Joint                                    2
Caribbean Restaurant                           2
Dessert Shop                                   2
Concert Hall                                   2
Burrito Place                                  2
Salad Place                                    2
Food                                           2
Performing Arts Venue                          2
African Restaurant                             2
Steakhouse                                     2
Museum                                         2
```

```
Soup Place                                    2
College Theater                               2
Bike Shop                                     2
Harbor / Marina                               2
Taxi                                          2
Beer Garden                                   2
Ramen /  Noodle House                         2
Bowling Alley                                 2
Tanning Salon                                 2
Spa / Massage                                 2
Housing Development                           2
Music Store                                   2
Southern / Soul Food Restaurant               2
College & University                          2
Recycling Facility                            1
South American Restaurant                     1
Cemetery                                      1
Law School                                    1
Rest Area                                     1
Animal Shelter                                1
Thrift / Vintage Store                        1
Synagogue                                     1
Casino                                        1
Greek Restaurant                              1
Nail Salon                                    1
Outdoors & Recreation                         1
Historic Site                                 1
Nursery School                                1
Temple                                        1
Bridal Shop                                   1
Medical School                                1
Garden                                        1
Antique Shop                                  1
Taco Place                                    1
Pool                                          1
Malaysian Restaurant                          1
Pool Hall                                     1
Winery                                        1
Campground                                    1
Arts & Entertainment                          1
Brazilian Restaurant                          1
History Museum                                1
Dumpling Restaurant                           1
Sorority House                                1
Professional & Other Places                   1
River                                         1
Camera Store                                  1
German Restaurant                             1
Jewelry Store                                 1
Cuban Restaurant                              1
Hot Dog Joint                                 1
Tea Room                                      1
Mobile Phone Shop                             1
Factory                                       1
Video Game Store                              1
Hardware Store                                1
Sculpture Garden                              1
Mediterranean Restaurant                      1
Name: CategoryName, dtype: int64
```

In [28]:

```
import matplotlib.pyplot as plt

x = venue_data['CategoryName']
plt.hist(x, bins = 10)
plt.show()
```

```
<Figure size 640x480 with 1 Axes>
```

In [17]:

```
print('The dataframe has {} venues with total of {} visitors.'.format(
        len(venue_data['VenueID'].unique()),
        sum(venue_data['Visitor Count'])
    )
)
```

```
The dataframe has 2000 venues with total of 12248 visitors.
```

In [18]:

```
import math as Math
def pointInCircle(lat0, lon0, r, lat, lon):
    C = 40075.04                                         # Earth circumfer
ence
    A = 360*r/C                                          # semi-minor in n
orth-south direction
    B = A/Math.cos(Math.radians(float(lat0)));           # semi-major in e
ast-west direction
    return Math.pow((float(lat)-float(lat0))/A, 2) + Math.pow((float(lon)-float(
lon0))/B, 2) < 1
```

In [20]:

```
someCategories = ['Food & Drink Shop', 'Electronics Store', 'Coffee Shop', 'Restaurant', 'Arts & Crafts Store', 'Gastropub',
                  'Mobile Phone Shop', 'Café', 'Automative Shop', 'American Restaurant', 'Food & Drink Shop', 'Burger Joint',
                  'Mexican Restaurant', 'Sandwich Place', 'Clothing Store', 'Ice Cream Shop', 'Pizza Place', 'Jewelry Store',
                  'Soup Place', 'Tattoo Parlor', 'Deli / Bodega', 'Diner', 'Salon / Barbershop', 'Laundry Service', 'Bar',
                  'Gym / Fitness Center', 'Hotel', 'Music Venue', 'BBQ Joint', 'Bookstore', 'Drugstore / Pharmacy',
                  'Sporting Goods Shop', 'Bakery', 'Fast Food Restaurant', 'Chinese Restaurant', 'Theater', 'Movie Theater',
                  'Sushi Restaurant', 'Miscellaneous Shop', 'French Restaurant', 'Seafood Restaurant', 'Fried Chicken Joint',
                  'Italian Restaurant', 'Toy / Game Store', 'Vegetarian / Vegan Restaurant', 'Donut Shop', 'German Restaurant',
                  'Bowling Alley', 'Beer Garden', 'Candy Store', 'Bagel Shop', 'Cuban Restaurant', 'Cupcake Shop',
                  'Breakfast Spot', 'Hardware Store', 'Japanese Restaurant',  'Latin American Restaurant', 'Spanish Restaurant',
                  'Spa / Massage', 'Middle Eastern Restaurant', 'Malaysian Restaurant', 'Record Shop', 'Wings Joint',
                  'Gas Station / Garage', 'Asian Restaurant', 'Burrito Place', 'Thai Restaurant', 'Salad Place',
                  'Ramen /  Noodle House', 'Automotive Shop', 'Convenience Store', 'Tea Room',  'Indian Restaurant',
                  'Thrift / Vintage Store', 'Paper / Office Supplies Store', 'Cosmetics Shop', 'Southern / Soul Food Restaurant',
                  'Smoke Shop', 'Snack Place', 'Furniture / Home Store', 'Caribbean Restaurant', 'Video Game Store', 'Steakhouse',
                  'Greek Restaurant', 'Dumpling Restaurant', 'Mediterranean Restaurant', 'African Restaurant', 'Taco Place',
                  'Jewelry Store', 'Hot Dog Joint', 'South American Restaurant', 'Winery']
```

In [21]:

```
visitForCategories = {}
maxVisited = ""

# count the number of visitors for categories
for idx in range(0, 2000):
    venue_id = list(venues)[idx]
    visitorCount = len(venues.get(venue_id))
    venue_type = categories.get(venue_id)

    if venue_type not in visitForCategories:
        visitForCategories[venue_type] = visitorCount
    else:
        visitForCategories[venue_type] = visitForCategories[venue_type]+visitorCount

# visitForCategories
sorted_dict = sorted(visitForCategories.items(), key=lambda x: x[1], reverse=True)
for v in sorted_dict:
    if v[0] in someCategories:
        maxVisited = v[0]
        break

# category names with their visit number
sorted_dict = sorted(visitForCategories.items(), key=lambda x: x[1], reverse=True)
print(sorted_dict)

print()

# Max visited category
print("'" + maxVisited + "'", "is the most visited commercial category according to given data.")
```

```
[('Train Station', 943), ('Park', 778), ('Airport', 769), ('Bar', 75
6), ('Subway', 587), ('Coffee Shop', 447), ('Gym / Fitness Center',
447), ('Food & Drink Shop', 426), ('Neighborhood', 362), ('Plaza', 3
42), ('Stadium', 339), ('Bridge', 272), ('Office', 264), ('Departmen
t Store', 240), ('Mall', 238), ('Burger Joint', 206), ('American Res
taurant', 202), ('Road', 201), ('Bus Station', 196), ('Hotel', 184),
('Other Great Outdoors', 178), ('Music Venue', 166), ('Home (privat
e)', 158), ('Mexican Restaurant', 154), ('Electronics Store', 137),
('Ferry', 126), ('College Academic Building', 116), ('Sandwich Plac
e', 115), ('BBQ Joint', 109), ('Bookstore', 105), ('Building', 100),
('Medical Center', 94), ('University', 94), ('Clothing Store', 89),
('Drugstore / Pharmacy', 83), ('Beach', 72), ('Government Building',
70), ('Convention Center', 70), ('Sporting Goods Shop', 68), ('Baker
y', 68), ('Fast Food Restaurant', 59), ('Chinese Restaurant', 59),
('Theater', 57), ('Deli / Bodega', 55), ('Movie Theater', 53), ('Foo
d Truck', 51), ('Sushi Restaurant', 50), ('Pizza Place', 47), ('Gene
ral Entertainment', 47), ('Ice Cream Shop', 46), ('Bank', 45), ('Mis
cellaneous Shop', 41), ('Light Rail', 40), ('Church', 38), ('Concert
Hall', 38), ('French Restaurant', 36), ('Seafood Restaurant', 35),
('Fried Chicken Joint', 34), ('Residential Building (Apartment / Con
do)', 33), ('Italian Restaurant', 33), ('Comedy Club', 33), ('Dine
r', 30), ('Toy / Game Store', 29), ('Vegetarian / Vegan Restaurant',
27), ('Café', 26), ('Community College', 26), ('Scenic Lookout', 2
4), ('Donut Shop', 23), ('German Restaurant', 23), ('Bowling Alley',
22), ('Beer Garden', 22), ('Gastropub', 19), ('Candy Store', 19),
('Bagel Shop', 19), ('Cuban Restaurant', 18), ('School', 18), ('Cupc
ake Shop', 18), ('Breakfast Spot', 18), ('General Travel', 17), ('Sa
lon / Barbershop', 16), ('General College & University', 16), ('Hard
ware Store', 16), ('Japanese Restaurant', 16), ('Latin American Rest
aurant', 15), ('Athletic & Sport', 15), ('Restaurant', 13), ('Spanis
h Restaurant', 12), ('Spa / Massage', 12), ('Middle Eastern Restaura
nt', 12), ('Malaysian Restaurant', 12), ('Record Shop', 12), ('Libra
ry', 12), ('Student Center', 11), ('Wings Joint', 11), ('High Schoo
l', 11), ('Arts & Crafts Store', 10), ('Laundry Service', 10), ('Gas
Station / Garage', 10), ('Asian Restaurant', 9), ('Burrito Place',
9), ('Parking', 9), ('Harbor / Marina', 9), ('Thai Restaurant', 9),
('Playground', 8), ('Campground', 8), ('Salad Place', 8), ('Event Sp
ace', 8), ('Pool Hall', 8), ('Ramen /  Noodle House', 8), ('Automoti
ve Shop', 7), ('Convenience Store', 7), ('Tea Room', 7), ('Post Offi
ce', 7), ('Indian Restaurant', 7), ('River', 7), ('Thrift / Vintage
Store', 7), ('Paper / Office Supplies Store', 6), ('Cosmetics Shop',
6), ('Dessert Shop', 6), ('Museum', 6), ('Pet Store', 6), ('College
& University', 6), ('Brazilian Restaurant', 6), ('Tanning Salon',
5), ('Bike Shop', 5), ('Art Gallery', 5), ('Arts & Entertainment',
5), ('Food', 5), ('Southern / Soul Food Restaurant', 4), ('Smoke Sho
p', 4), ('College Theater', 4), ('Snack Place', 4), ('Furniture / Ho
me Store', 4), ('Sculpture Garden', 4), ('Mobile Phone Shop', 3),
('Tattoo Parlor', 3), ('Other Nightlife', 3), ('Moving Target', 3),
('Caribbean Restaurant', 3), ('Video Game Store', 3), ('Arcade', 3),
('Steakhouse', 3), ('Greek Restaurant', 3), ('Soup Place', 2), ('Mus
ic Store', 2), ('Dumpling Restaurant', 2), ('Performing Arts Venue',
2), ('Camera Store', 2), ('Housing Development', 2), ('Synagogue',
2), ('Mediterranean Restaurant', 2), ('African Restaurant', 2), ('Ta
xi', 2), ('Professional & Other Places', 2), ('Taco Place', 2), ('Je
welry Store', 1), ('Animal Shelter', 1), ('Factory', 1), ('Cemeter
y', 1), ('Medical School', 1), ('Pool', 1), ('Garden', 1), ('Hot Dog
Joint', 1), ('Outdoors & Recreation', 1), ('Sorority House', 1), ('C
asino', 1), ('Temple', 1), ('Historic Site', 1), ('Rest Area', 1),
('History Museum', 1), ('Recycling Facility', 1), ('Bridal Shop',
1), ('Nail Salon', 1), ('Nursery School', 1), ('Antique Shop', 1),
('South American Restaurant', 1), ('Law School', 1), ('Winery', 1)]
```

```
'Bar' is the most visited commercial category according to given dat
a.
```

In [22]:

```
mostVisitedCommercialPlace = {}
visited = []

r = 4               # kilometers
n = 2000            # venues

# maxVisited ='Food & Drink Shop'

for idx in range(0, n):
    coords = list(coordinates)[idx]
    if tuple(coords) in visited:
        continue
    visited.append(tuple(coords))

    storeCount = {}
    for tempVal in range(0, n):
        venue_id = list(venues)[idx]
        temp_coord = list(coordinates)[tempVal]
        venue_type = categories.get(venue_id)

        if pointInCircle(coords[0], coords[1], r, temp_coord[0], temp_coord[1]) and tuple(temp_coord) not in visited:

            visited.append(tuple(temp_coord))
            if venue_type not in storeCount:
                storeCount[venue_type] = 1
            else:
                storeCount[venue_type] = storeCount.get(venue_type)+1

        if maxVisited not in storeCount:
            mostVisitedCommercialPlace[tuple(coords)] = 0
        else:
            mostVisitedCommercialPlace[tuple(coords)] = storeCount.get(maxVisited)

noneShops = []
sorted_dict = sorted(mostVisitedCommercialPlace.items(), key=lambda x: x[1], reverse=True)
print("Coordinates with number of " + maxVisited + " shops within", r, "kilometers according to", n, "venues.")
print()
for c in sorted_dict:
    print(c[0], ":", c[1])
    if c[1] < 2:
        noneShops.append(tuple(c[0]))


mostShopCoord = list(sorted_dict)[0][0]
del sorted_dict[0]
print("Coordinate that has the given specific shop the most: ", mostShopCoord)
```

```
Coordinates with number of Bar shops within 4 kilometers according t
o 2000 venues.

('40.60613336268842', '-74.17904376983643') : 2
('40.719810375488535', '-74.00258103213994') : 0
('40.60679958140643', '-74.04416981025437') : 0
('40.716161684843215', '-73.88307005845945') : 0
('40.69042711809854', '-73.95468677509598') : 0
('40.751591431346306', '-73.9741214009634') : 0
('40.61900594093755', '-73.99037472596906') : 0
('40.71976226666666', '-74.250014') : 0
('40.86198150306815', '-74.04790453737951') : 0
('40.82678953781387', '-73.94950923509141') : 0
('40.906627', '-73.777774') : 0
('40.73067679262482', '-74.06567180055883') : 0
('40.64531729239498', '-73.77383708953857') : 0
('40.79059946897114', '-73.98023377661316') : 0
('40.655535144394925', '-74.00862937888984') : 0
('40.86283581665962', '-74.19723987579346') : 0
('40.901057866884024', '-74.1507625579834') : 0
('40.963240757086346', '-74.09463109843134') : 0
('40.77615805031661', '-73.82361593360073') : 0
('40.92431190858341', '-73.99688829591487') : 0
('40.786713', '-74.175476') : 0
('40.74710920467287', '-74.15280867121942') : 0
('40.771046', '-74.065758') : 0
('40.725891', '-73.79143081000001') : 0
('40.83162208784265', '-74.13679361343384') : 0
('40.828602195433966', '-73.87925863265991') : 0
('40.60014371940161', '-73.94659322025349') : 0
('40.8706300946324', '-74.09792627389562') : 0
('40.677558813432675', '-73.74452479795166') : 0
('40.82358935335903', '-74.22380201337529') : 0
('40.8134844303353', '-74.07433032989502') : 0
('40.75568181015381', '-73.88307005845945') : 0
('40.68968493541091', '-74.17938709259033') : 0
('40.89621269901943', '-73.87670483270945') : 0
('40.66421141550086', '-73.91474954903893') : 0
('40.925826', '-73.835898') : 0
('40.77226863280218', '-73.93010370763156') : 0
('40.87097828520945', '-73.82876808947316') : 0
('40.57562689089684', '-73.98329850027537') : 0
('40.83309951197968', '-74.01021480560303') : 0
('40.6444055420497', '-74.0729570388794') : 0
('40.917335441802095', '-74.0756607055664') : 0
('40.634045330266964', '-74.14161270853154') : 0
('40.896905775860006', '-74.03017044067383') : 0
('40.78898753071118', '-74.25901478846667') : 0
('40.66342760828679', '-74.23399686813354') : 0
('40.75605706725121', '-74.23554035653764') : 0
('40.7541179388902', '-73.73833613617441') : 0
('40.5899058561196', '-73.8995361328125') : 0
('40.78280355158054', '-73.77417256445159') : 0
('40.579124179746714', '-73.82426211435131') : 0
('40.72465693431455', '-73.70641708374023') : 0
('40.86107855194721', '-73.93010370763155') : 0
('40.69526162612937', '-73.8441376028445') : 0
('40.93865397046296', '-74.12986841096105') : 0
('40.94975203974169', '-73.93010370763155') : 0
('40.65079702473058', '-73.95877360445469') : 0
```

```
Coordinate that has the given specific shop the most:  ('40.60613336268842', '-74.17904376983643')
```

In [23]:

```
# get the closest coordinates with less than 2 of the given type of shop to the coordinate that attracts the most visitors

from math import cos, asin, sqrt
nearNeighborhoods = []

def distance(lat1, lon1, lat2, lon2):
    p = 0.017453292519943295
    a = 0.5 - cos((lat2-lat1)*p)/2 + cos(lat1*p)*cos(lat2*p) * (1-cos((lon2-lon1)*p)) / 2
    return 12742 * asin(sqrt(a))

def closest(data, v):
    return min(data, key=lambda p: distance(float(v[0]),float(v[1]),float(p[0][0]),float(p[0][1])) if p[0] not in nearNeighborhoods else 9999)

# for i in range(10):
#     nearNeighborhoods.append(closest(list(sorted_dict), list(sorted_dict)[0][0])[0])

nearNeighborhoods
```

Out[23]:

```
[]
```

In [24]:

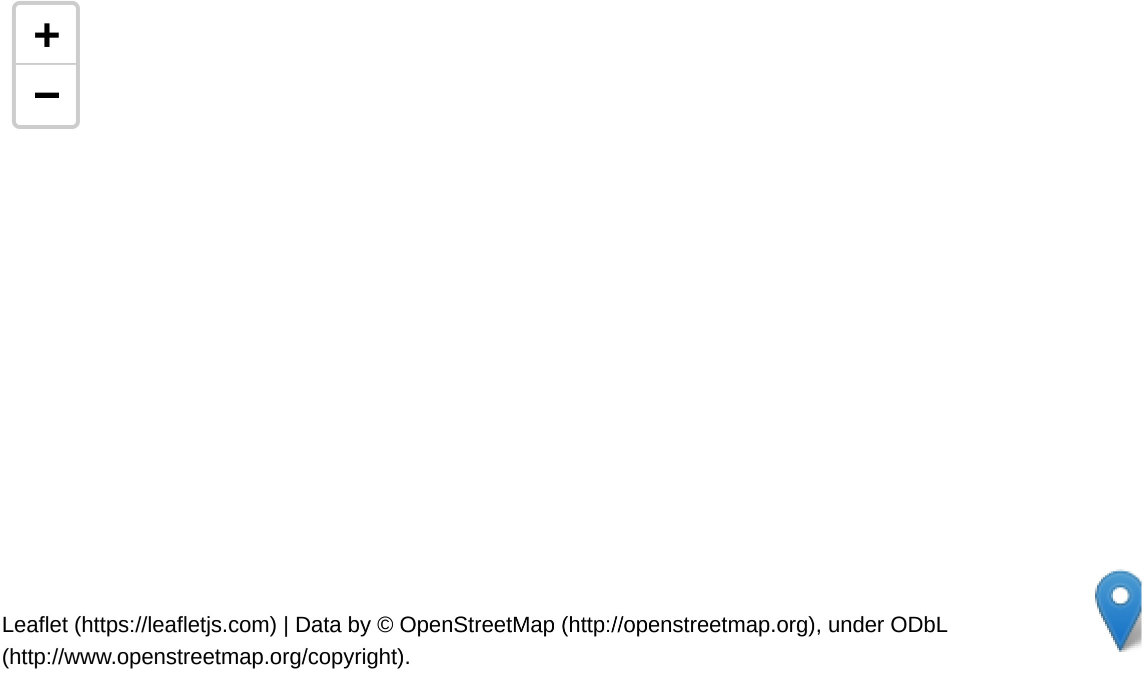
```
# get the 2 coordinates with less than 2 shops within range and find their neigh
borhoods
neighborhoods = {}
findNumOfPlaces = 2

centerNeighborhoodData = requests.get('https://api.foursquare.com/v2/venues/sear
ch?&client_id=JGGBRN5XODTLZGJOMCSWIQMRH1JLGJKPSFR10XNB2R5U25GR&client_secret=KWR
AMLK2HOJBQ2XLICLKXRU3M4HOCC1U2VG4Y4OPP5JF03QX&v=20180605&ll={},{}&limit=1'.forma
t(
    float(mostShopCoord[0]), float(mostShopCoord[1]))).json()

centerNeighborhood = centerNeighborhoodData['response']['venues'][0]['location']
['formattedAddress']

while len(neighborhoods) != findNumOfPlaces:
    nearNeighborhoods.append(closest(list(sorted_dict), mostShopCoord)[0])
    lat = nearNeighborhoods[-1][0]
    lng = nearNeighborhoods[-1][1]
    url = 'https://api.foursquare.com/v2/venues/search?&client_id=JGGBRN5XODTLZG
JOMCSWIQMRH1JLGJKPSFR10XNB2R5U25GR&client_secret=KWRAMLK2HOJBQ2XLICLKXRU3M4HOCC1
U2VG4Y4OPP5JF03QX&v=20180605&ll={},{}&limit=1'.format(
    lat, lng)

    results = requests.get(url).json()

    try:
        neighborhoods[results['response']['venues'][0]['location']['neighborhoo
d']] = tuple([lat,lng])
    except:
        continue

# print out the selected neighborhoods which are okay to get in. ( having no mor
e than 1 shop within given range )
for ne in neighborhoods:
    print(ne)
```

```
Bensonhurst
Bedford-Stuyvesant
```

In [25]:

```
# the red dot in the map is the center which has the most shops within given ran
ge

import folium

mapit = folium.Map( location=[40.7128, -74.0060], zoom_start=11 )
latlon = []
neighboorhoodNames = []

for name, coords in neighborhoods.items():    # for name, age in dictionary.iter
items():  (for Python 2.x)
    latlon.append(tuple([float(coords[0]), float(coords[1])]))
    neighboorhoodNames.append(name)


# shop's coordinates with the most number of the given shop
folium.CircleMarker(
        [mostShopCoord[0], mostShopCoord[1]],
        radius=5,
        color='#ff0000',
        fill=True,
        fill_color='#ff0000',
        popup=folium.Popup('{}, NY'.format(centerNeighborhood), parse_html=True
),
        parse_html=False,
        fill_opacity=0.7).add_to(mapit)


# label the potential neighborhoods
for c, n in zip(latlon, neighboorhoodNames):
    label = '{}, NY'.format(n)
    label = folium.Popup(label, parse_html=True)

    folium.Marker( location=[ c[0], c[1] ], fill_color='#43d9de', radius=8, popu
p=label, parse_html=False ).add_to( mapit )

mapit
```

Out[25]:


+
−
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [26]:

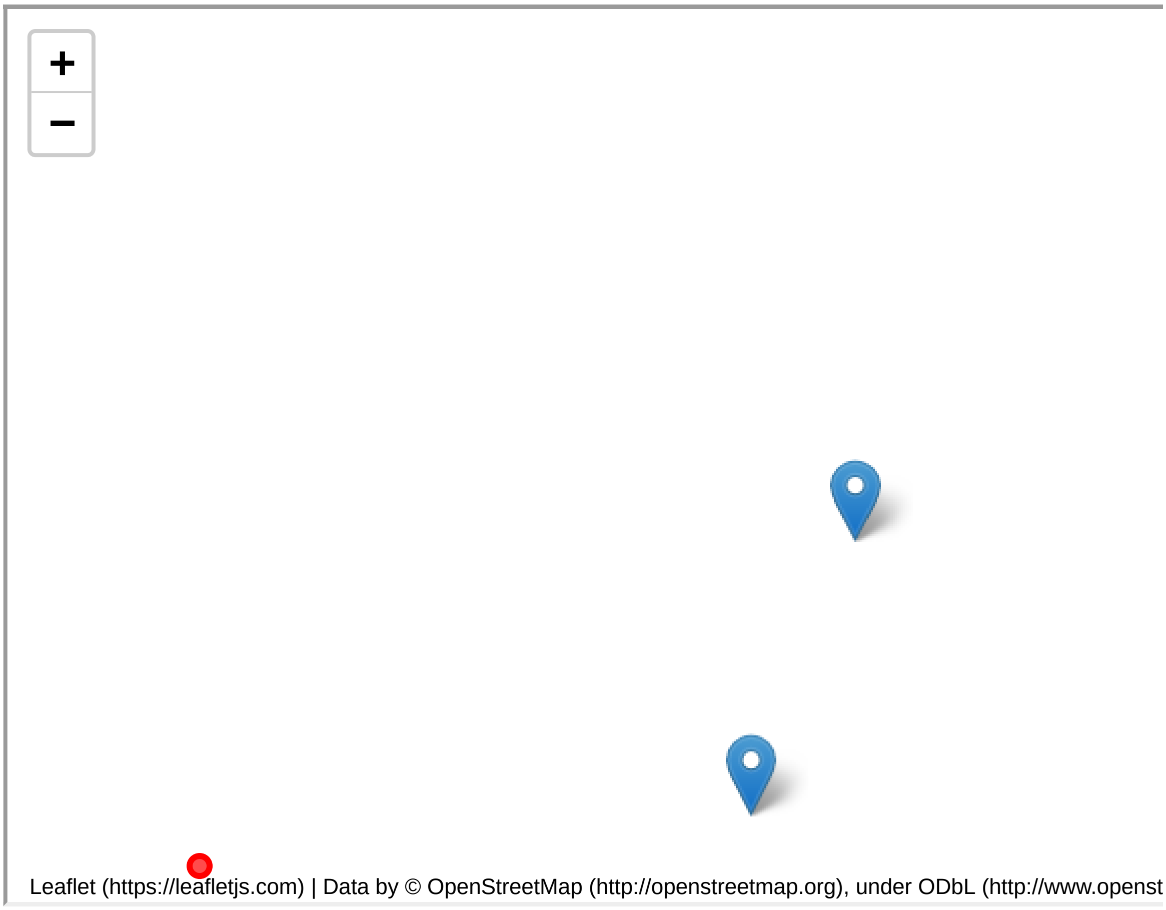
```
mapit.save( 'map.html')
```

In [29]:

```
from IPython.display import HTML

HTML('<iframe src=map.html width=700 height=450></iframe>')
```

```
/home/hunglv/.virtualenvs/age_gender/lib/python3.6/site-packages/IPython/core/display.py:694: UserWarning: Consider using IPython.display.IFrame instead
  warnings.warn("Consider using IPython.display.IFrame instead")
```

Out[29]:


+
−
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openst

In [ ]: