In [2]:

```
#Importation of necessary libraries
import time
import pandas as pd # library for data analsysis
import numpy as np # library to handle data in a vectorized manner
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files
import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dat
aframe

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

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you ha
ven't completed the Foursquare API lab
import folium # map rendering library
import folium # map rendering library
from folium import plugins

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

import seaborn as sns

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


print('Done! Libraries imported.')
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

Done! Libraries imported.
```

# WEEK ONE

## 1. Description of the problem and a discussion of the background.

### Background

I am moving to New York City in a few months. I am looking for a single flat and I am hoping to use data science to help me find the most place to live in based on several factors. I want to be in Manhattan, near a tube station, walking distance to work-out studios, food, and entertainment. As much as possible, I want to be cost efficient.

### Problem to be solved

The task is to find the most cost efficient rental flat in Manhattan, NYC within walking distance to groceries, shops, a work-out studio, and a subway station. Additionally, flat must have 2 bedrooms, a living room, and bathroom. The rent must be around $6000/ month.

### Interested Audience

This excercise will be beneficial to and interesting for other young professionals moving to Manhattan.

# WEEK TWO

## 2. Description of the data and how it will be used to solve the problem.

I require the the Geodata (longitude, latitude) of the following enterprises in Manhattan, NYC.

- Cycling studios
- Coffee shops
- Tube stations
- Places for rent and cost

## Source of Data

- Wikipedia (https://en.wikipedia.org/wiki/List_of_New_York_City_Subway_stations_in_Manhattan (https://en.wikipedia.org/wiki/List_of_New_York_City_Subway_stations_in_Manhattan))
- NY Transit authority and Google maps (https://www.google.com/maps/search/manhattan+subway+metro+stations/@40.7837297,-74.1033043,1 (https://www.google.com/maps/search/manhattan+subway+metro+stations/@40.7837297,-74.1033043,1
- Real estate websites (http://www.rentmanhattan.com/index.cfm?page=search&state=results (http://www.rentmanhattan.com/index.cfm?page=search&state=results) https://www.nestpick.com/search?city=new-york&page=1&order=relevance&district=manhattan&gclid=CjwKCAiAjNjgBRAgEiwAGLlf2hkP3A-cPxjZYkURqQEswQK2jKQEpv_MvKcrIhRWRzNkc_r-fGi0lxoCA7cQAvD_BwE&type=apartment&display=list (https://www.nestpick.com/search?city=new-york&page=1&order=relevance&district=manhattan&gclid=CjwKCAiAjNjgBRAgEiwAGLlf2hkP3A-cPxjZYkURqQEswQK2jKQEpv_MvKcrIhRWRzNkc_r-fGi0lxoCA7cQAvD_BwE&type=apartment&display=list) https://www.realtor.com/apartments/Manhattan_NY (https://www.realtor.com/apartments/Manhattan_NY))

# 3. Methodology Section

Course lab generated cluster neighborhood data. Csv file is used for convenience of the project.

In [6]:

```
# Read csv file with clustered neighborhoods with geodata
manhattan_data  = pd.read_csv('mh_neigh_data.csv')
manhattan_data.head()
```

Out[6]:

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

In [7]:

```
#Finding the 10 clustered neighborhoods I can move into
manhattan_merged = pd.read_csv('manhattan_merged.csv')
manhattan_merged.head()
```

Out[7]:

| | Borough | Neighborhood | Latitude | Longitude | Cluster Labels | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue |
|---|---|---|---|---|---|---|---|---|
| 0 | Manhattan | Marble Hill | 40.876551 | -73.910660 | 2 | Coffee Shop | Discount Store | Yoga Studio |
| 1 | Manhattan | Chinatown | 40.715618 | -73.994279 | 2 | Chinese Restaurant | Cocktail Bar | Dim Sum Restaurant |
| 2 | Manhattan | Washington Heights | 40.851903 | -73.936900 | 4 | Café | Bakery | Mobile Phone Shop |
| 3 | Manhattan | Inwood | 40.867684 | -73.921210 | 3 | Mexican Restaurant | Lounge | Pizza Place |
| 4 | Manhattan | Hamilton Heights | 40.823604 | -73.949688 | 0 | Mexican Restaurant | Coffee Shop | Café |

In [8]:

```
#Now I will map Manhattan neighborhoods, showing top 10 clusters

# create map of Manhattan using latitude and longitude values from Nominatim
latitude= 40.7308619
longitude= -73.9871558

kclusters=5
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=13)

# 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]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(manhattan_merged['Latitude'], manhattan_merged
['Longitude'], manhattan_merged['Neighborhood'], manhattan_merged['Cluster Label
s']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=20,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)
  # add markers for rental places to map
for lat, lng, label in zip(manhattan_data['Latitude'], manhattan_data['Longitud
e'], manhattan_data['Neighborhood']):
    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_clusters)


map_clusters
```

Out[8]:

In [9]:

```
#Assign value to kk - cluster number to explore. I'm just going to pick any clus
ter.
kk = 2
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == kk, manhattan_merged.
columns[[1] + list(range(5, manhattan_merged.shape[1]))]]
```

Out[9]:

| | Neighborhood | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7t Co |
|---|---|---|---|---|---|---|---|---|
| 0 | Marble Hill | Coffee Shop | Discount Store | Yoga Studio | Steakhouse | Supplement Shop | Tennis Stadium | |
| 1 | Chinatown | Chinese Restaurant | Cocktail Bar | Dim Sum Restaurant | American Restaurant | Vietnamese Restaurant | Salon / Barbershop | |
| 6 | Central Harlem | African Restaurant | Seafood Restaurant | French Restaurant | American Restaurant | Cosmetics Shop | Chinese Restaurant | |
| 9 | Yorkville | Coffee Shop | Gym | Bar | Italian Restaurant | Sushi Restaurant | Pizza Place | M Res |
| 14 | Clinton | Theater | Italian Restaurant | Coffee Shop | American Restaurant | Gym / Fitness Center | Hotel | Wir |
| 23 | Soho | Clothing Store | Boutique | Women's Store | Shoe Store | Men's Store | Furniture / Home Store | Res |
| 26 | Morningside Heights | Coffee Shop | American Restaurant | Park | Bookstore | Pizza Place | Sandwich Place | |
| 34 | Sutton Place | Gym / Fitness Center | Italian Restaurant | Furniture / Home Store | Indian Restaurant | Dessert Shop | American Restaurant | |
| 39 | Hudson Yards | Coffee Shop | Italian Restaurant | Hotel | Theater | American Restaurant | Café | |

Now we will use the csv of webscrapped Manhattan real estate data. to explore rental options. Use Nominatim.

In [11]:

```
# csv files with rental places with basic data but still wihtout geodata ( latit
ude and longitude)
# pd.read_csv(' le.csv', header=None, nrows=5)
mh_rent=pd.read_csv('MH_flats_price.csv')
mh_rent.head()
```

Out[11]:

| | Address | Area | Price_per_ft2 | Rooms | Area-ft2 | Rent_Price | Lat | Long |
|---|---|---|---|---|---|---|---|---|
| 0 | West 105th Street | Upper West Side | 2.94 | 5.0 | 3400 | 10000 | NaN | NaN |
| 1 | East 97th Street | Upper East Side | 3.57 | 3.0 | 2100 | 7500 | NaN | NaN |
| 2 | West 105th Street | Upper West Side | 1.89 | 4.0 | 2800 | 5300 | NaN | NaN |
| 3 | CARMINE ST. | West Village | 3.03 | 2.0 | 1650 | 5000 | NaN | NaN |
| 4 | 171 W 23RD ST. | Chelsea | 3.45 | 2.0 | 1450 | 5000 | NaN | NaN |

Now we need the latitude and longitude (geodata) of the rental places we webscrapped. We will use Nominatum.

In [ ]:

```

```

In [14]:

```
mh_rent=pd.read_csv('MH_rent_latlong.csv')
mh_rent.head()
```

Out[14]:

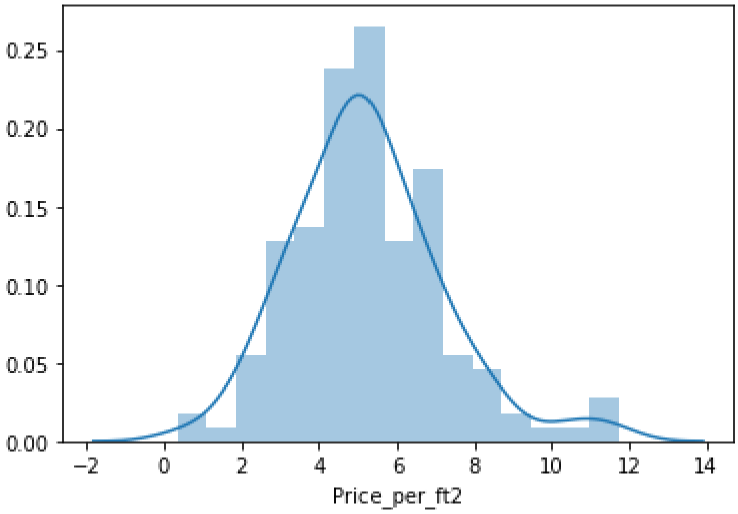
| | Address | Area | Price_per_ft2 | Rooms | Area-ft2 | Rent_Price | Lat | Long |
|---|---|---|---|---|---|---|---|---|
| 0 | West 105th Street | Upper West Side | 2.94 | 5.0 | 3400 | 10000 | 40.799771 | -73.966213 |
| 1 | East 97th Street | Upper East Side | 3.57 | 3.0 | 2100 | 7500 | 40.788585 | -73.955277 |
| 2 | West 105th Street | Upper West Side | 1.89 | 4.0 | 2800 | 5300 | 40.799771 | -73.966213 |
| 3 | CARMINE ST. | West Village | 3.03 | 2.0 | 1650 | 5000 | 40.730523 | -74.001873 |
| 4 | 171 W 23RD ST. | Chelsea | 3.45 | 2.0 | 1450 | 5000 | 40.744118 | -73.995299 |

In [16]:

```
import seaborn as sns
sns.distplot(mh_rent['Price_per_ft2'],bins=15)
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1d1e1f50>
```



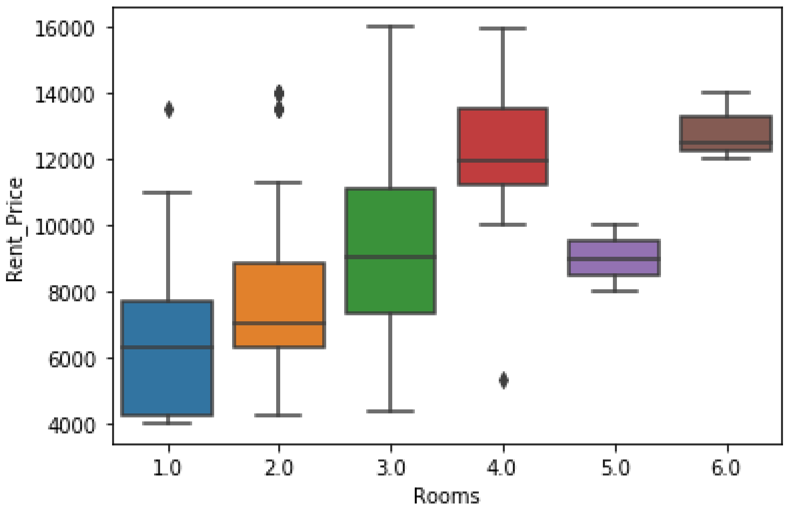

This graph shows that the mean rental price is within our $6000 range.

In [17]:

```
sns.boxplot(x='Rooms', y= 'Rent_Price', data=mh_rent)
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1d9526d0>
```




Now we will check the monthly rental prices and addresses so that we can select which apartment matches our price range.

In [19]:

```
# create map of Manhattan using latitude and longitude values from Nominatim
latitude= 40.7308619
longitude= -73.9871558

map_manhattan_rent = folium.Map(location=[latitude, longitude], zoom_start=12.5)

# add markers to map
for lat, lng, label in zip(mh_rent['Lat'], mh_rent['Long'],'$ ' + mh_rent['Rent_Price'].astype(str)+ ',  '+ mh_rent['Address']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=6,
        popup=label,
        color='green',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_manhattan_rent)


map_manhattan_rent
```

Out[19]:

We need to determine rental places that match us based on pricess and location of amenities around the apartment.

In [20]:

```
# create map of Manhattan using latitude and longitude values from Nominatim
latitude= 40.7308619
longitude= -73.9871558

map_manhattan_rent = folium.Map(location=[latitude, longitude], zoom_start=12.5)

# add markers to map
for lat, lng, label in zip(mh_rent['Lat'], mh_rent['Long'],'$ ' + mh_rent['Rent_
Price'].astype(str)+ ',  '+ mh_rent['Address']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=6,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_manhattan_rent)


map_manhattan_rent
```

Out[20]:

Time to drill down on a particular apartment and the venues near it - i.e. Near subway stations?

In [21]:

```
kk = 3
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == kk, manhattan_merged.
columns[[1] + list(range(5, manhattan_merged.shape[1]))]]
```

Out[21]:

| | Neighborhood | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7t Co |
|---|---|---|---|---|---|---|---|---|
| 3 | Inwood | Mexican Restaurant | Lounge | Pizza Place | Café | Wine Bar | Bakery | Ar Res |
| 5 | Manhattanville | Deli / Bodega | Italian Restaurant | Seafood Restaurant | Mexican Restaurant | Sushi Restaurant | Beer Garden | |
| 10 | Lenox Hill | Sushi Restaurant | Italian Restaurant | Coffee Shop | Gym / Fitness Center | Pizza Place | Burger Joint | I |
| 12 | Upper West Side | Italian Restaurant | Bar | Bakery | Vegetarian / Vegan Restaurant | Indian Restaurant | Coffee Shop | Cos |
| 16 | Murray Hill | Sandwich Place | Hotel | Japanese Restaurant | Gym / Fitness Center | Coffee Shop | Salon / Barbershop | |
| 17 | Chelsea | Coffee Shop | Italian Restaurant | Ice Cream Shop | Bakery | Nightclub | Theater | Art |
| 18 | Greenwich Village | Italian Restaurant | Sushi Restaurant | French Restaurant | Clothing Store | Chinese Restaurant | Café | Res |
| 27 | Gramercy | Italian Restaurant | Restaurant | Thrift / Vintage Store | Cocktail Bar | Bagel Shop | Coffee Shop | |
| 29 | Financial District | Coffee Shop | Hotel | Gym | Wine Shop | Steakhouse | Bar | Res |
| 31 | Noho | Italian Restaurant | French Restaurant | Cocktail Bar | Gift Shop | Bookstore | Grocery Store | M Res |
| 32 | Civic Center | Gym / Fitness Center | Bakery | Italian Restaurant | Cocktail Bar | French Restaurant | Sandwich Place | |
| 35 | Turtle Bay | Italian Restaurant | Coffee Shop | Steakhouse | Wine Bar | Sushi Restaurant | Hotel | |
| 36 | Tudor City | Café | Park | Pizza Place | Mexican Restaurant | Greek Restaurant | Sushi Restaurant | |
| 38 | Flatiron | Italian Restaurant | American Restaurant | Gym | Gym / Fitness Center | Yoga Studio | Vegetarian / Vegan Restaurant | |

We need to map out subway stations against the rental apartments. We do this by using the csv file from our webscrapping.

In [23]:

```
mh=pd.read_csv('NYC_subway_list.csv')
mh.head()
```

Out[23]:

| | sub_station | sub_address |
|---|---|---|
| 0 | Dyckman Street Subway Station | 170 Nagle Ave, New York, NY 10034, USA |
| 1 | 57 Street Subway Station | New York, NY 10106, USA |
| 2 | Broad St | New York, NY 10005, USA |
| 3 | 175 Street Station | 807 W 177th St, New York, NY 10033, USA |
| 4 | 5 Av and 53 St | New York, NY 10022, USA |

In [25]:

```
#  Add columns 'lat'  and 'long' to mh dataframe - with random temporary numbers
to get started
sLength = len(mh['sub_station'])
lat = pd.Series(np.random.randn(sLength))
long =pd.Series(np.random.randn(sLength))
mh = mh.assign(lat=lat.values)
mh = mh.assign(long=long.values)
#After this we need to get the geodata for each station and add to our dataframe
```

In [26]:

```
mh=pd.read_csv('MH_subway.csv')
print(mh.shape)
mh.head()
```

```
(76, 4)
```

Out[26]:

| | sub_station | sub_address | lat | long |
|---|---|---|---|---|
| 0 | Dyckman Street Subway Station | 170 Nagle Ave, New York, NY 10034, USA | 40.861857 | -73.924509 |
| 1 | 57 Street Subway Station | New York, NY 10106, USA | 40.764250 | -73.954525 |
| 2 | Broad St | New York, NY 10005, USA | 40.730862 | -73.987156 |
| 3 | 175 Street Station | 807 W 177th St, New York, NY 10033, USA | 40.847991 | -73.939785 |
| 4 | 5 Av and 53 St | New York, NY 10022, USA | 40.764250 | -73.954525 |

In [27]:

```
# removing duplicate rows and creating new set mhsub1
mhsub1=mh.drop_duplicates(subset=['lat','long'], keep="last").reset_index(drop=T
rue)
mhsub1.shape
```

Out[27]:

```
(22, 4)
```

Generate map of subway stations using Noninatim geolocator to show desirable rental apartemnts and their nearest stations.

In [28]:

```
latitude=40.7308619
longitude=-73.9871558

map_mhsub1 = folium.Map(location=[latitude, longitude], zoom_start=12)

# add markers of subway locations to map
for lat, lng, label in zip(mhsub1['lat'], mhsub1['long'],  mhsub1['sub_station']
.astype(str) ):
    label = folium.Popup(label, parse_html=True)
    folium.RegularPolygonMarker(
        [lat, lng],
        number_of_sides=6,
        radius=6,
        popup=label,
        color='red',
        fill_color='red',
        fill_opacity=2.5,
    ).add_to(map_mhsub1)
map_mhsub1
```

Out[28]:

In [29]:

```
mh_rent.head()
```

Out[29]:

| | Address | Area | Price_per_ft2 | Rooms | Area-ft2 | Rent_Price | Lat | Long |
|---|---|---|---|---|---|---|---|---|
| 0 | West 105th Street | Upper West Side | 2.94 | 5.0 | 3400 | 10000 | 40.799771 | -73.966213 |
| 1 | East 97th Street | Upper East Side | 3.57 | 3.0 | 2100 | 7500 | 40.788585 | -73.955277 |
| 2 | West 105th Street | Upper West Side | 1.89 | 4.0 | 2800 | 5300 | 40.799771 | -73.966213 |
| 3 | CARMINE ST. | West Village | 3.03 | 2.0 | 1650 | 5000 | 40.730523 | -74.001873 |
| 4 | 171 W 23RD ST. | Chelsea | 3.45 | 2.0 | 1450 | 5000 | 40.744118 | -73.995299 |

In [30]:

```
# create map of Manhattan using latitude and longitude values from Nominatim
latitude= 40.7308619
longitude= -73.9871558

map_manhattan_rent = folium.Map(location=[latitude, longitude], zoom_start=13.3)

# add markers to map
for lat, lng, label in zip(mh_rent['Lat'], mh_rent['Long'],'$ ' + mh_rent['Rent_
Price'].astype(str)+ mh_rent['Address']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=6,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_manhattan_rent)

    # add markers of subway locations to map
for lat, lng, label in zip(mhsub1['lat'], mhsub1['long'],  mhsub1['sub_station']
.astype(str) ):
    label = folium.Popup(label, parse_html=True)
    folium.RegularPolygonMarker(
        [lat, lng],
        number_of_sides=6,
        radius=6,
        popup=label,
        color='red',
        fill_color='red',
        fill_opacity=2.5,
    ).add_to(map_manhattan_rent)

    # Adds tool to the top right
from folium.plugins import MeasureControl
map_manhattan_rent.add_child(MeasureControl())

# Measurement ruler icon tool to measure distances in map
from folium.plugins import FloatImage
url = ('https://media.licdn.com/mpr/mpr/shrinknp_100_100/AAEAAQAAAAAAAAlgAAAAJGE
3OTA4YTdlLTkzZjUtNDFjYy1iZThlLWQ5OTNkYzlhNzM4OQ.jpg')
FloatImage(url, bottom=5, left=85).add_to(map_manhattan_rent)

map_manhattan_rent
```

Out[30]:

# 4. Results Section

Finally! Now we can use our data to pick out our perfect apartment! We begin by mapping the apartments, subway stations, and amenity venues.

In [31]:

```
# create map of Manhattan using latitude and longitude values from Nominatim
latitude= 40.7308619
longitude= -73.9871558

map_mh_one = folium.Map(location=[latitude, longitude], zoom_start=13.3)

# add markers to map
for lat, lng, label in zip(mh_rent['Lat'], mh_rent['Long'],'$ ' + mh_rent['Rent_
Price'].astype(str)+ ', '+mh_rent['Address']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=6,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_mh_one)

    # add markers of subway locations to map
for lat, lng, label in zip(mhsub1['lat'], mhsub1['long'],  mhsub1['sub_station']
.astype(str) ):
    label = folium.Popup(label, parse_html=True)
    folium.RegularPolygonMarker(
        [lat, lng],
        number_of_sides=6,
        radius=6,
        popup=label,
        color='red',
        fill_color='red',
        fill_opacity=2.5,
    ).add_to(map_mh_one)


# set color scheme for the clusters
kclusters=5
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]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(manhattan_merged['Latitude'], manhattan_merged
['Longitude'], manhattan_merged['Neighborhood'], manhattan_merged['Cluster Label
s']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=15,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_mh_one)

    # Adds tool to the top right
from folium.plugins import MeasureControl
```

```
map_mh_one.add_child(MeasureControl())

# Measurement ruler icon tool to measure distances in map
from folium.plugins import FloatImage
url = ('https://media.licdn.com/mpr/mpr/shrinknp_100_100/AAEAAQAAAAAAAAlgAAAAJGE
3OTA4YTdlLTkzZjUtNDFjYy1iZThlLWQ5OTNkYzlhNzM4OQ.jpg')
FloatImage(url, bottom=5, left=85).add_to(map_mh_one)

map_mh_one
```

Out[31]:

### 19 Dutch Street in the Financial District Neighborhood and near 'Fulton Street Subway' station, Cluster # 3 Monthly rent : $6935, looks like a good apartmnet.

Time to find the cafes, restaurants, gyms, and other amenities hear this apartment.

In [32]:

```
kk = 3 #cluster 3
manhattan_merged.loc[manhattan_merged['Cluster Labels'] == kk, manhattan_merged.
columns[[1] + list(range(5, manhattan_merged.shape[1]))]]
```

Out[32]:

| | Neighborhood | 1st Most Common Venue | 2nd Most Common Venue | 3rd Most Common Venue | 4th Most Common Venue | 5th Most Common Venue | 6th Most Common Venue | 7t Cc |
|---|---|---|---|---|---|---|---|---|
| 3 | Inwood | Mexican Restaurant | Lounge | Pizza Place | Café | Wine Bar | Bakery | Ar Res |
| 5 | Manhattanville | Deli / Bodega | Italian Restaurant | Seafood Restaurant | Mexican Restaurant | Sushi Restaurant | Beer Garden | |
| 10 | Lenox Hill | Sushi Restaurant | Italian Restaurant | Coffee Shop | Gym / Fitness Center | Pizza Place | Burger Joint | I |
| 12 | Upper West Side | Italian Restaurant | Bar | Bakery | Vegetarian / Vegan Restaurant | Indian Restaurant | Coffee Shop | Cos |
| 16 | Murray Hill | Sandwich Place | Hotel | Japanese Restaurant | Gym / Fitness Center | Coffee Shop | Salon / Barbershop | |
| 17 | Chelsea | Coffee Shop | Italian Restaurant | Ice Cream Shop | Bakery | Nightclub | Theater | Art |
| 18 | Greenwich Village | Italian Restaurant | Sushi Restaurant | French Restaurant | Clothing Store | Chinese Restaurant | Café | Res |
| 27 | Gramercy | Italian Restaurant | Restaurant | Thrift / Vintage Store | Cocktail Bar | Bagel Shop | Coffee Shop | |
| 29 | Financial District | Coffee Shop | Hotel | Gym | Wine Shop | Steakhouse | Bar | Res |
| 31 | Noho | Italian Restaurant | French Restaurant | Cocktail Bar | Gift Shop | Bookstore | Grocery Store | N Res |
| 32 | Civic Center | Gym / Fitness Center | Bakery | Italian Restaurant | Cocktail Bar | French Restaurant | Sandwich Place | |
| 35 | Turtle Bay | Italian Restaurant | Coffee Shop | Steakhouse | Wine Bar | Sushi Restaurant | Hotel | |
| 36 | Tudor City | Café | Park | Pizza Place | Mexican Restaurant | Greek Restaurant | Sushi Restaurant | |
| 38 | Flatiron | Italian Restaurant | American Restaurant | Gym | Gym / Fitness Center | Yoga Studio | Vegetarian / Vegan Restaurant | |

COOL. Basically this apartment is within budget range of $6000, located near a subway station, and have multiple amenities nearby.

# 5. Discussion Section

This course take time to digest and apply but it is very rewarding once you see how much you have learned and progressed.

I am quite proud of myself and I feel like I have a good background as I continue learning about the world of data science.

# 6. Conclusion Section

This program is such a great way to get into the world of data science. This course covered topics really well and I look forward to applying this in my work.

Stay home and we will get through this COVID-19 - this too shall pass!