Capstone Project: The Battle of Neighborhoods

Table of Contents

This notebook contains the following parts:

1. Introduction

1.1 Background

Nowadays, due to various reasons, people tend to move from one city to another, or even from one country to another. However, sometimes people are not quite familiar with the new country or city that they are going to move to. So in order to help people decide where to live, information associated with people's requirements (eg: distance from workplace, monthly rent, crime rate, restaurants nearby) need to be gathered. Futhermore, based on personal preference, optimal options should be generated for people to choose.

1.2 Specific problem to solve

I would like to find a suitable place to live in New York city. There are several requirements that I hope the potential rental place can meet.

1. I hope the monthly rent won't be higher than US $2000.
2. The distance between New York university (70 Washington Square South New York, NY) and the rental place is within 3 miles.
3. Safety is a critical issue, so I would also choose those have lower crime rates.
4. Since I prefer Asian food, especially Chinese food and Japanese food, it will be great if there are great Chinese and Japanese food restaurants nearby.

1.3 Interested audience

Although the outcome is a case-by-case suggestion, depending on the requirements of different people in different city or country, the general methodologies are applicable to all cases. Anyone who would like to get customized suggestions in where to live will find it to be beneficial.

2. Data

We will need the following data for this project:

1. Using www.rent.com (http://www.rent.com) website, I searched rental place near New York university, with US$2000 as the maximum monthly rent. House renting data containing name of rental place, address, type and price are scraped from https://www.rent.com/new-york/new-york-university-houses/max-price-2000 (https://www.rent.com/new-york/new-york-university-houses/max-price-2000), and saved as **"rent_data_mining.csv"**.
2. Get the coordinations of places for rent using **foursquare agent**.
3. Gather both **violent crime rate** and **property crime rate** for those potential rental places. I will use **beautifulsoup 4** to scrape the cirme rates from the following website https://www.bestplaces.net/crime/zip-code/new_york/new_york/11210 (https://www.bestplaces.net/crime/zip-code/new_york/new_york/11210), and change the zip code according to different potential rental places.
4. Restaurants information (venue, venue id, venue category and rating) are obtained from **foursquare API**.

3. Methodology

This section represents the main component of the report where the data is gathered, prepared for analysis. The tools described are used here and the notebook cells indicate the execution of steps.

The analysis and the stragegy: The strategy is based on filtering data in section 2, in order to facilitate the choice of at least two candidate places for rent. The choice is made based on the demands imposed : monthly rental price, distance from New York university, crime rate and restaurants nearby. The maps with popup labels allow quick identification of location, thus making the selection more straightforward.

Import necessary libraries.

In [1]:

```
import requests # library to handle requests
import pandas as pd # library for data analsysis
import numpy as np # library to handle data in a vectorized manner
import re

#!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # module to convert an address in

print('Libraries imported.')
```

```
Libraries imported.
```

Define Foursquare Credentials and Version

In [2]:

```
# @hidden_cell
CLIENT_ID = 'UOQC5VBGMXKL0MWLS**********************' # your Foursquar
CLIENT_SECRET = 'HH4WDD4VPMN0RG3ZIANAZY****************' # your Foursq
VERSION = '20190607'
Limit = 30
#print('Your credentails:')
#print('CLIENT_ID: ' + CLIENT_ID)
#print('CLIENT_SECRET:' + CLIENT_SECRET)
```

In order to explore near New York university, let's start by converting the its address to its latitude and longitude coordinates.¶

In [3]:

```
address = '70 Washington Square South New York, NY'

geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
NYU_latitude = location.latitude
NYU_longitude = location.longitude
print('New York University''s latitude is:',NYU_latitude)
print('New York University''s longitude is:',NYU_longitude)
```

```
New York Universitys latitude is: 40.72942865
New York Universitys longitude is: -73.9972178045625
```

3.1 Import rental information and clean the data

I scraped the rental information from www.rent.com (http://www.rent.com) and saved into **rent_data_mining.csv**. Note: I have already applied the first requirement, which is the monthly rent is not higher than US $2000. So now, I need to import this file.

In [4]:
```
import types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage.
# You might want to remove those credentials before you share your note
client_********* = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='***********',
    ibm_auth_endpoint="https://iam.bluemix.net/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3-api.us-geo.objectstorage.service.networkla

body = client_**********.get_object(Bucket='******',Key='rent_2k.csv')[
# add missing __iter__ method, so pandas accepts body as file-like obje
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __i
```

In [5]:
```
df = pd.read_csv(body)
df.head()
```

Out[5]:

	Name	Address	Type	Price
0	1 Pennsylvania Plaza	New York, NY 10119	3 Beds • 1 Bath	$1,750.00
1	139A Jefferson Ave	Brooklyn, NY 11216	1 Bed • 1 Bath	$975.00
2	6605 Boulevard E	West New York, NJ 07093	1 Bed • 1 Bath	$1,450.00
3	41-07 42nd St	Queens, NY 11104	Studio • 1 Bath	$1,685.00
4	212 Lewis Ave	Brooklyn, NY 11221	1 Bed • 1 Bath	$1,600.00

In [6]:
```
df.shape
```

Out[6]:
```
(237, 4)
```

Let's simply focused on rental places in New York city.

In [7]:
```
df_1 = df[df['Address'].str.contains("NY")]
df_1
df_1.shape
```

Out[7]:
```
(82, 4)
```

Since I would like to live alone, I prefer to choose either **"1 Bed, 1Bath"** or **"Studio, 1 Bath"** types.

In [8]:

```
df_2 = df_1[df_1['Type'].str.contains("Studio")| df_1['Type'].str.conta
df_2
#df_2.shape
```

Out[8]:

	Name	Address	Type	Price
1	139A Jefferson Ave	Brooklyn, NY 11216	1 Bed • 1 Bath	$975.00
3	41-07 42nd St	Queens, NY 11104	Studio • 1 Bath	$1,685.00
4	212 Lewis Ave	Brooklyn, NY 11221	1 Bed • 1 Bath	$1,600.00
6	56-72 64th St	Queens, NY 11378	1 Bed • 1 Bath	$1,900.00
7	174 Mulberry St 2	174 Mulberry St New York, NY 10013	Studio • 1 Bath	$1,650.00
8	510 E 5th St B	510 E 5th St New York, NY 10009	Studio • 1 Bath	$1,975.00
9	226 E 26th St B-E	226 E 26th St New York, NY 10010	Studio • 1 Bath	$1,825.00
10	152 E 35th St 6-H	152 E 35th St New York, NY 10016	Studio • 1 Bath	$1,975.00

I would like to get all the addresses of these rental places, so that I can get the coordination information later.

In [9]:

```
ad_list = df_2['Address'].tolist()
ad_list
```

Out[9]:

```
['Brooklyn, NY 11216',
 'Queens, NY 11104',
 'Brooklyn, NY 11221',
 'Queens, NY 11378',
 '174 Mulberry St New York, NY 10013',
 '510 E 5th St New York, NY 10009',
 '226 E 26th St New York, NY 10010',
 '152 E 35th St New York, NY 10016',
 '5 Tudor City Pl New York, NY 10017',
 '25 Tudor City Pl New York, NY 10017',
 '349 W 45th St New York, NY 10036',
 '297 Broadway Brooklyn, NY 11211',
 '81 Waverly Ave Brooklyn, NY 11205',
 '177 Congress St Brooklyn, NY 11201',
 '430 E 66th St New York, NY 10065',
 '28 Carroll St Brooklyn, NY 11231',
 '698 Fulton St Brooklyn, NY 11217',
 '403 E 70th St New York, NY 10021',
 '23-10 41st Ave Queens, NY 11101',
 '482 E 74th St New York, NY 10021',
 '1431 1st Avenue New York, NY 10021',
 '94 3rd Pl Brooklyn, NY 11231',
 '1313 3rd Ave New York, NY 10021',
 '328 E 77th St New York, NY 10075',
 '418 E 77th St New York, NY 10075',
 '38-05 Crescent St Queens, NY 11101',
 '223 E 78th St New York, NY 10075',
 '448 E 78th St New York, NY 10075',
 '158 6th Ave Brooklyn, NY 11217',
 '321 Flatbush Ave Brooklyn, NY 11217',
 '528 E 79th St New York, NY 10075',
 '304 E 81st St New York, NY 10028',
 '410 West End Ave New York, NY 10024',
 '126 E 83rd St New York, NY 10028',
 '47-07 41st St Queens, NY 11104',
 '128 E 84th St New York, NY 10028',
 '128 E 84th St New York, NY 10028',
 '128 E 84th St New York, NY 10028',
 '128 E 84th St New York, NY 10028',
 '239 Madison St Brooklyn, NY 11216',
 '249 Madison St Brooklyn, NY 11216',
 '88 East End Ave New York, NY 10028',
 '635 Grand Ave Brooklyn, NY 11238',
 '960 Willoughby Ave Brooklyn, NY 11221',
 '230 Troutman St Brooklyn, NY 11237',
 'Queens, NY 11104',
 '430 E 87th St New York, NY 10128',
 '46-09 47th Ave Queens, NY 11377',
 '57 Herkimer St Brooklyn, NY 11216',
 '414 E 88th St New York, NY 10128',
 '1727 2nd Ave New York, NY 10128',
 '950 Bergen St Brooklyn, NY 11216',
 '204 13th St Brooklyn, NY 11215',
 '293 Central Park West New York, NY 10024',
 '23-39 31st Dr Queens, NY 11106',
 '476 6th Ave Brooklyn, NY 11215',
 '21-29 31st Ave Queens, NY 11106',
 '27 Himrod St Brooklyn, NY 11221',
 '82 Rogers Ave Brooklyn, NY 11216',
 '29-08 31st Ave Queens, NY 11106',
 '41-15 51st St Queens, NY 11377',
 '30-60 Crescent St Queens, NY 11102',
 '39-89 51st St Queens, NY 11377',
 '39-89 51st St Queens, NY 11377',
 '30-49 Crescent St Queens, NY 11102',
 '548 Hancock St Brooklyn, NY 11233',
 '4-33 27th Ave Queens, NY 11102',
 '825 Putnam Ave Brooklyn, NY 11221',
 '31-12 42nd St Queens, NY 11103',
 '31-12 42nd St Queens, NY 11103',
 '18-16 26th Rd Queens, NY 11102',
 '362 16th St Brooklyn, NY 11215',
 '1024 Gates Ave Brooklyn, NY 11221',
 '562 Fairview Ave Queens, NY 11385',
 '12 W 103rd St New York, NY 10025',
 '62-49 Forest Ave Queens, NY 11385',
 '214 W 102nd St New York, NY 10025']
```

To get the coordination information for those rental places, I need to use the **foursquare agent**.

```
In [10]: la_list = list()
         lo_list = list()
         for ad in ad_list:
             geolocator = Nominatim(user_agent="foursquare_agent")
             location = geolocator.geocode(ad)
             #if location is not None and location.latitude is not None:
             latitude = location.latitude
             longitude = location.longitude
             la_list.append(latitude)
             lo_list.append(longitude)
         #print(la_list)
         #print(lo_list)
         print('The number of latitude obtained is:',len(la_list))
         print('The number of longitude obtained is:',len(lo_list))
```

```
The number of latitude obtained is: 77
The number of longitude obtained is: 77
```

```
In [11]: # Write the latitude and longitude information into dataframe.
         df_2['Latitude'] = la_list
         df_2['Longitude'] = lo_list
         df_2
```

[illegible]	BW	NY 10036	Bath	[illegible]	[illegible]	[illegible]
34	297 Broadway 3F	297 Broadway Brooklyn, NY 11211	1 Bed • 1 Bath	$2,000.00	40.708554	-73.958006
48	81 Waverly Ave 3	81 Waverly Ave Brooklyn, NY 11205	Studio • 1 Bath	$1,850.00	40.695231	-73.968025
50	177 Congress St 4	177 Congress St Brooklyn, NY 11201	Studio • 1 Bath	$2,000.00	40.688475	-73.995646
75	430 E 66th St 14	430 E 66th St New York, NY 10065	1 Bed • 1 Bath	$1,925.00	40.763254	-73.957494
85	28 Carroll St TOPFLOOR	28 Carroll St Brooklyn, NY 11231	1 Bed • 1 Bath	$2,000.00	40.683215	-74.004820
87	698 Fulton St 2R	698 Fulton St Brooklyn, NY 11217	Studio • 1 Bath	$1,950.00	40.685934	-73.974001
93	403 E 70th St 12	403 E 70th St New York, NY 10021	1 Bed • 1 Bath	$2,000.00	40.766431	-73.956275
98	23-10 41st Ave 9-H	23-10 41st Ave Queens, NY 11101	Studio • 1 Bath	$1,759.00	40.752586	-73.940195

```
In [12]: df_2.to_csv('rent_lat_lng.csv')
```

```
In [13]: !conda install -c conda-forge folium=0.5.0 --yes
In [15]: !conda install -c conda-forge folium=0.5.0 --yes
         import folium # plotting library
         print('Folium imported!')
```

```
Solving environment: done

## Package Plan ##

  environment location: /opt/conda/envs/DSX-Python35

  added / updated specs: 
    - folium=0.5.0


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    altair-2.2.2               |          py35_1         462 KB  c
onda-forge
    folium-0.5.0               |            py_0          45 KB  c
onda-forge
    openssl-1.0.2r             |       h14c3975_0         3.1 MB  c
onda-forge
    ca-certificates-2019.6.16  |       hecc5488_0         145 KB  c
onda-forge
    certifi-2018.8.24          |        py35_1001         139 KB  c
onda-forge
    branca-0.3.1               |            py_0          25 KB  c
onda-forge
    vincent-0.4.4              |            py_1          28 KB  c
onda-forge
    ------------------------------------------------------------
                                           Total:         4.0 MB

The following NEW packages will be INSTALLED:

    altair:          2.2.2-py35_1      conda-forge
    branca:          0.3.1-py_0        conda-forge
    folium:          0.5.0-py_0        conda-forge
    vincent:         0.4.4-py_1        conda-forge

The following packages will be UPDATED:

    ca-certificates: 2019.1.23-0                     --> 2019.6.16-he
cc5488_0 conda-forge
    certifi:         2018.8.24-py35_1                --> 2018.8.24-py
35_1001  conda-forge

The following packages will be DOWNGRADED:

    openssl:         1.0.2s-h7b6447c_0               --> 1.0.2r-h14c3
975_0    conda-forge
```

```
Downloading and Extracting Packages
altair-2.2.2         | 462 KB    | ################################
##### | 100%
folium-0.5.0         | 45 KB     | ################################
##### | 100%
openssl-1.0.2r       | 3.1 MB    | ################################
##### | 100%
ca-certificates-2019 | 145 KB    | ################################
##### | 100%
certifi-2018.8.24    | 139 KB    | ################################
##### | 100%
branca-0.3.1         | 25 KB     | ################################
##### | 100%
vincent-0.4.4        | 28 KB     | ################################
##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Folium imported!
```

Now, let's have a look at the locations of these rental places and have a rough idea about the distance between the rental place and New York University respectively.

In [14]:

```
rental_map = folium.Map(location=[NYU_latitude, NYU_longitude], zoom_st

# add a red circle marker to represent the New York University
folium.features.CircleMarker(
    [NYU_latitude, NYU_longitude],
    radius=10,
    color='red',
    popup='New York University',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(rental_map)

# add the Italian restaurants as blue circle markers
for address, lat, lng in zip(df_2.Address, df_2.Latitude, df_2.Longitud
    folium.features.CircleMarker(
        [lat, lng],
        radius=5,
        color='blue',
        popup=address,
        fill = True,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(rental_map)

# display map
rental_map
```

Out[14]:

Since now Google starts to charge for their Google Maps Geocoding API, I need to use geopy.distance to get the distance between the rental place and New York University.

In [15]:

```
from geopy.distance import distance
print('Imported!')
```

```
Imported!
```

In [16]:

```
coordination_list = list(zip(df_2['Latitude'], df_2['Longitude']))
coordination_list
```

```
 (40.6884753, -73.9956456905611),
 (40.76325425, -73.9574944331219),
 (40.6832151, -74.0048201706379),
 (40.685934, -73.9740013),
 (40.76643095, -73.9562745618977),
 (40.7525859, -73.9401946),
 (40.7684321, -73.9538842184601),
 (40.7693997, -73.9551854834534),
 (40.67858985, -73.9982194410135),
 (40.77175465, -73.9587923343222),
 (40.6297445714286, -74.0275684285714),
 (40.6290073061225, -74.0252030204082),
 (40.7555337, -73.9359009),
 (40.77327415, -73.956618550178),
 (40.77076615, -73.9515355515006),
 (40.6771488, -73.977079643265),
 (40.6781126, -73.9729088),
 (40.7704863, -73.9486994),
 (40.6269646938775, -74.029351755102),
 (40.78462655, -73.9805715437754)
```

In [17]:
```
NYU = (40.72942865,-73.9972178045625)
DISTANCE_LIST = list()
for coordination in coordination_list:
    d = distance(NYU,coordination)
    DISTANCE_LIST.append(d.miles)
print(DISTANCE_LIST)
len(DISTANCE_LIST)
```

```
[6.018297972964093, 12.042266813340438, 6.018297972964093, 12.04226
6813340438, 0.6509346547204209, 0.7807362197929842, 1.1388675913973
203, 1.535379353555283, 1.9299484570031737, 1.9293262822948352, 2.1
53141462115093, 2.5125206016757207, 2.8138496771103356, 2.827076954
811914, 3.1294677661577985, 3.2137261171128104, 3.239371784720339,
3.337013913287653, 3.392633029802341, 3.5234198943497184, 3.5316221
948177553, 3.508380646717996, 3.5490624097797974, 7.060750238049578
, 7.083491872723959, 3.6878897882462076, 3.7003005911023106, 3.7259
740764864797, 3.7592318658660995, 3.7639897486653897, 3.80904596226
7381, 7.268931092528627, 3.9076747978742894, 3.9222396426689685, 3.
9447628496328684, 3.968467881950467, 3.968467881950467, 3.968467881
950467, 3.968467881950467, 3.9598942111670783, 3.9707313342148796,
4.048689647400895, 4.105068101171428, 4.128694496855742, 4.17447448
76186735, 12.042266813340438, 7.618722916053908, 4.20576368137502,
4.181223745907076, 4.2233006365001655, 4.2697526751059796, 4.252371
37327989, 4.248513915196025, 4.298949831381771, 4.366741549876512,
4.369498426554544, 4.432863136820121, 4.4731597610135925, 4.4763659
23616866, 4.507516866802952, 4.563230095715183, 4.554818261485559,
4.5603321062088265, 4.5603321062088265, 4.594037744287549, 4.577848
175006112, 4.626055749579631, 4.663447139728746, 4.764267378489083,
4.764267378489083, 4.768554241514655, 4.736731145825862, 4.79747301
0052826, 4.924179152383224, 4.960790987563459, 5.216869882263389, 4
.970816187625557]
```

Out[17]: 77

In [18]:

```
df_2['Distance from NYU (miles)'] = DISTANCE_LIST
df_2
```

```
See the caveats in the documentation: http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy
(http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing
-view-versus-copy)
  if __name__ == '__main__':
```

Out[18]:

	Name	Address	Type	Price	Latitude	Longitude	Distance from NYU (miles)
1	139A Jefferson Ave	Brooklyn, NY 11216	1 Bed • 1 Bath	$975.00	40.650104	-73.949582	6.018298
3	41-07 42nd St	Queens, NY 11104	Studio • 1 Bath	$1,685.00	40.652493	-73.791421	12.042267
4	212 Lewis Ave	Brooklyn, NY 11221	1 Bed • 1 Bath	$1,600.00	40.650104	-73.949582	6.018298
6	56-72 64th St	Queens, NY 11378	1 Bed • 1 Bath	$1,900.00	40.652493	-73.791421	12.042267

Since now I know the distance from NYU, then I can drop those places whose distance is greater than 3 miles.

In [19]:

```
drop_index = df_2[df_2['Distance from NYU (miles)']>3].index # To get t
df_2.drop(drop_index , inplace=True)
df_2
```

```
/opt/conda/envs/DSX-Python35/lib/python3.5/site-packages/ipykernel/
__main__.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy
(http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing
-view-versus-copy)
  from ipykernel import kernelapp as app
```

Out[19]:

	Name	Address	Type	Price	Latitude	Longitude	Distance from NYU (miles)
7	174 Mulberry St 2	174 Mulberry St New York, NY 10013	Studio • 1 Bath	$1,650.00	40.719998	-73.996902	0.650935
8	510 E 5th St B	510 E 5th St New York, NY 10009	Studio • 1 Bath	$1,975.00	40.724452	-73.983861	0.780736
9	226 E 26th St B-E	226 E 26th St New York, NY 10010	Studio • 1 Bath	$1,825.00	40.739994	-73.980549	1.138868
10	152 E 35th St 6-H	152 E 35th St New York, NY 10016	Studio • 1 Bath	$1,975.00	40.746509	-73.978469	1.535379
11	5 Tudor City Pl 306	5 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,800.00	40.748640	-73.970494	1.929948
12	25 Tudor City Pl 1206	25 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,950.00	40.748636	-73.970505	1.929326
24	349 W 45th St BW	349 W 45th St New York, NY 10036	1 Bed • 1 Bath	$1,750.00	40.760182	-73.990272	2.153141
34	297 Broadway 3F	297 Broadway Brooklyn, NY 11211	1 Bed • 1 Bath	$2,000.00	40.708554	-73.958006	2.512521
48	81 Waverly Ave 3	81 Waverly Ave Brooklyn, NY 11205	Studio • 1 Bath	$1,850.00	40.695231	-73.968025	2.813850
50	177 Congress St 4	177 Congress St Brooklyn, NY 11201	Studio • 1 Bath	$2,000.00	40.688475	-73.995646	2.827077

Let's sort the data by price and distance from NYU (miles) in ascending order.

```
In [20]: df_3 = df_2.sort_values(by=['Price', 'Distance from NYU (miles)'])
         df_3
```

Out[20]:

	Name	Address	Type	Price	Latitude	Longitude	Distance from NYU (miles)
7	174 Mulberry St 2	174 Mulberry St New York, NY 10013	Studio • 1 Bath	$1,650.00	40.719998	-73.996902	0.650935
24	349 W 45th St BW	349 W 45th St New York, NY 10036	1 Bed • 1 Bath	$1,750.00	40.760182	-73.990272	2.153141
11	5 Tudor City Pl 306	5 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,800.00	40.748640	-73.970494	1.929948
9	226 E 26th St B-E	226 E 26th St New York, NY 10010	Studio • 1 Bath	$1,825.00	40.739994	-73.980549	1.138868
48	81 Waverly Ave 3	81 Waverly Ave Brooklyn, NY 11205	Studio • 1 Bath	$1,850.00	40.695231	-73.968025	2.813850
12	25 Tudor City Pl 1206	25 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,950.00	40.748636	-73.970505	1.929326
8	510 E 5th St B	510 E 5th St New York, NY 10009	Studio • 1 Bath	$1,975.00	40.724452	-73.983861	0.780736
10	152 E 35th St 6-H	152 E 35th St New York, NY 10016	Studio • 1 Bath	$1,975.00	40.746509	-73.978469	1.535379
34	297 Broadway 3F	297 Broadway Brooklyn, NY 11211	1 Bed • 1 Bath	$2,000.00	40.708554	-73.958006	2.512521
50	177 Congress St 4	177 Congress St Brooklyn, NY 11201	Studio • 1 Bath	$2,000.00	40.688475	-73.995646	2.827077

In [21]:
```
df_3 = df_3.reset_index(drop=True)
df_3
```

Out[21]:

	Name	Address	Type	Price	Latitude	Longitude	Distance from NYU (miles)
0	174 Mulberry St 2	174 Mulberry St New York, NY 10013	Studio • 1 Bath	$1,650.00	40.719998	-73.996902	0.650935
1	349 W 45th St BW	349 W 45th St New York, NY 10036	1 Bed • 1 Bath	$1,750.00	40.760182	-73.990272	2.153141
2	5 Tudor City Pl 306	5 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,800.00	40.748640	-73.970494	1.929948
3	226 E 26th St B-E	226 E 26th St New York, NY 10010	Studio • 1 Bath	$1,825.00	40.739994	-73.980549	1.138868
4	81 Waverly Ave 3	81 Waverly Ave Brooklyn, NY 11205	Studio • 1 Bath	$1,850.00	40.695231	-73.968025	2.813850
5	25 Tudor City Pl 1206	25 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,950.00	40.748636	-73.970505	1.929326
6	510 E 5th St B	510 E 5th St New York, NY 10009	Studio • 1 Bath	$1,975.00	40.724452	-73.983861	0.780736
7	152 E 35th St 6-H	152 E 35th St New York, NY 10016	Studio • 1 Bath	$1,975.00	40.746509	-73.978469	1.535379
8	297 Broadway 3F	297 Broadway Brooklyn, NY 11211	1 Bed • 1 Bath	$2,000.00	40.708554	-73.958006	2.512521
9	177 Congress St 4	177 Congress St Brooklyn, NY 11201	Studio • 1 Bath	$2,000.00	40.688475	-73.995646	2.827077

ZipCode needs to be extracted and saved to the dataframe for further use.

In [22]:
```
df_3['ZipCode'] = df_3['Address'].str.extract('(\d{5})', expand=True)
df_3
```

Out[22]:

	Name	Address	Type	Price	Latitude	Longitude	Distance from NYU (miles)	ZipCode
0	174 Mulberry St 2	174 Mulberry St New York, NY 10013	Studio • 1 Bath	$1,650.00	40.719998	-73.996902	0.650935	10013
1	349 W 45th St BW	349 W 45th St New York, NY 10036	1 Bed • 1 Bath	$1,750.00	40.760182	-73.990272	2.153141	10036
2	5 Tudor City Pl 306	5 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,800.00	40.748640	-73.970494	1.929948	10017
3	226 E 26th St B-E	226 E 26th St New York, NY 10010	Studio • 1 Bath	$1,825.00	40.739994	-73.980549	1.138868	10010
4	81 Waverly Ave 3	81 Waverly Ave Brooklyn, NY 11205	Studio • 1 Bath	$1,850.00	40.695231	-73.968025	2.813850	11205
5	25 Tudor City Pl 1206	25 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,950.00	40.748636	-73.970505	1.929326	10017
6	510 E 5th St B	510 E 5th St New York, NY 10009	Studio • 1 Bath	$1,975.00	40.724452	-73.983861	0.780736	10009
7	152 E 35th St 6-H	152 E 35th St New York, NY 10016	Studio • 1 Bath	$1,975.00	40.746509	-73.978469	1.535379	10016
8	297 Broadway 3F	297 Broadway Brooklyn, NY 11211	1 Bed • 1 Bath	$2,000.00	40.708554	-73.958006	2.512521	11211
9	177 Congress St 4	177 Congress St Brooklyn, NY 11201	Studio • 1 Bath	$2,000.00	40.688475	-73.995646	2.827077	11201

In [23]:
```
df_3.to_csv('rental_distance_zipcode.csv')
```

Now let's visualize the filtered rental places on the map.

In [24]:

```
rental_filtered_map = folium.Map(location=[NYU_latitude, NYU_longitude]

# add a red circle marker to represent the New York University
folium.features.CircleMarker(
    [NYU_latitude, NYU_longitude],
    radius=10,
    color='red',
    popup='New York University',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(rental_filtered_map)

# add the Italian restaurants as blue circle markers
for address, lat, lng in zip(df_3.Address, df_3.Latitude, df_3.Longitud
    folium.features.CircleMarker(
        [lat, lng],
        radius=5,
        color='blue',
        popup=address,
        fill = True,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(rental_filtered_map)

# display map
rental_filtered_map
```

Out[24]:

3.2 Crime rate data scraping

Safety is a critical issue that I would like to consider when deciding where to live. So I need to get the both **violent crime rate** and **property crime rate** for those potential rental places. I will use the following website https://www.bestplaces.net/crime/zip-code/new_york/new_york/11210 (https://www.bestplaces.net/crime/zip-code/new_york/new_york/11210) and zip code of those rental places to scrape the crime rates. To do so, I choose to use **beautifulsoup4**.

```
In [25]: import bs4 # beautifulsoup4 to scrap webpage
         import urllib.request
```

After importing the necessary libraries, zip code from dataframe will be saved as **zipcode_list** for scrapping webpages.

```
In [26]: zipcode_list=df_3['ZipCode'].tolist()
         zipcode_list
```

```
Out[26]: ['10013',
          '10036',
          '10017',
          '10010',
          '11205',
          '10017',
          '10009',
          '10016',
          '11211',
          '11201']
```

In [27]:

```
import urllib.parse
url = 'https://www.bestplaces.net/crime/zip-code/new_york/new_york/1121
parts = urllib.parse.urlparse(url) # To parse the components of url.
#print(parts)

path_list = ['/crime/zip-code/new_york/new_york/' + zipcode for zipcode
#print(path_list)
url_list=[]
for p in path_list:
    parts = parts._replace(path=p) # Replace the path with new path wit
    url_list.append(parts.geturl())# Get urls and save them into a new
print(url_list)
```

```
['https://www.bestplaces.net/crime/zip-code/new_york/new_york/10013
', 'https://www.bestplaces.net/crime/zip-code/new_york/new_york/100
36', 'https://www.bestplaces.net/crime/zip-code/new_york/new_york/1
0017', 'https://www.bestplaces.net/crime/zip-code/new_york/new_york
/10010', 'https://www.bestplaces.net/crime/zip-code/new_york/new_yo
rk/11205', 'https://www.bestplaces.net/crime/zip-code/new_york/new_
york/10017', 'https://www.bestplaces.net/crime/zip-code/new_york/ne
w_york/10009', 'https://www.bestplaces.net/crime/zip-code/new_york/
new_york/10016', 'https://www.bestplaces.net/crime/zip-code/new_yor
k/new_york/11211', 'https://www.bestplaces.net/crime/zip-code/new_y
ork/new_york/11201']
```

Using the url_list, I scrape the webpage respectively to get violent crime rate and property crime rate.

In [28]:

```
from bs4 import BeautifulSoup
vc_rate=[]
pc_rate=[]
for zip_url in url_list:
    website_url = requests.get(zip_url).text
    soup = BeautifulSoup(website_url,'lxml')
#print(soup.prettify())

    rate = soup.find_all('h5')
    #print(rate[1],rate[2])
    v_rate = re.findall("\d+\.\d+", str(rate[1]))
    p_rate = re.findall("\d+\.\d+", str(rate[2]))
    vc_rate.append(v_rate[0])
    pc_rate.append(p_rate[0])
print(vc_rate)
print(pc_rate)
df_3['Violent Crime Rate']=vc_rate # Write into dataframe
df_3['Property Crime Rate']=pc_rate
df_3
```

```
['36.4', '49.9', '28.7', '33.5', '61.7', '28.7', '63.3', '34.7', '5
1.5', '32.7']
['44.7', '58.2', '37.8', '43.0', '60.3', '37.8', '67.0', '44.1', '5
0.1', '28.5']
```

Out[28]:

	Name	Address	Type	Price	Latitude	Longitude	Distance from NYU (miles)	ZipCode	Violent Crime Rate
0	174 Mulberry St 2	174 Mulberry St New York, NY 10013	Studio • 1 Bath	$1,650.00	40.719998	-73.996902	0.650935	10013	36.4
1	349 W 45th St BW	349 W 45th St New York, NY 10036	1 Bed • 1 Bath	$1,750.00	40.760182	-73.990272	2.153141	10036	49.9
2	5 Tudor City Pl 306	5 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,800.00	40.748640	-73.970494	1.929948	10017	28.7
3	226 E 26th St B-E	226 E 26th St New York, NY 10010	Studio • 1 Bath	$1,825.00	40.739994	-73.980549	1.138868	10010	33.5
4	81 Waverly Ave 3	81 Waverly Ave Brooklyn, NY 11205	Studio • 1 Bath	$1,850.00	40.695231	-73.968025	2.813850	11205	61.7
5	25 Tudor City Pl 1206	25 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,950.00	40.748636	-73.970505	1.929326	10017	28.7
6	510 E 5th St B	510 E 5th St New York, NY 10009	Studio • 1 Bath	$1,975.00	40.724452	-73.983861	0.780736	10009	63.3
7	152 E 35th St 6-H	152 E 35th St New York, NY 10016	Studio • 1 Bath	$1,975.00	40.746509	-73.978469	1.535379	10016	34.7
8	297 Broadway 3F	297 Broadway Brooklyn, NY 11211	1 Bed • 1 Bath	$2,000.00	40.708554	-73.958006	2.512521	11211	51.5
9	177 Congress St 4	177 Congress St Brooklyn, NY 11201	Studio • 1 Bath	$2,000.00	40.688475	-73.995646	2.827077	11201	32.7

For both violent crime rate and property crime rate, the lower the number is, the safer this place will be. So I would like to sort the remaining places based on both crime rates in an ascending manner.

In [29]:
```
df_4 = df_3.sort_values(by=['Violent Crime Rate', 'Property Crime Rate'
df_4
```

Out[29]:

	Name	Address	Type	Price	Latitude	Longitude	Distance from NYU (miles)	ZipCode	Violent Crime Rate
2	5 Tudor City Pl 306	5 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,800.00	40.748640	-73.970494	1.929948	10017	28.7
5	25 Tudor City Pl 1206	25 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,950.00	40.748636	-73.970505	1.929326	10017	28.7
9	177 Congress St 4	177 Congress St Brooklyn, NY 11201	Studio • 1 Bath	$2,000.00	40.688475	-73.995646	2.827077	11201	32.7
3	226 E 26th St B-E	226 E 26th St New York, NY 10010	Studio • 1 Bath	$1,825.00	40.739994	-73.980549	1.138868	10010	33.5
7	152 E 35th St 6-H	152 E 35th St New York, NY 10016	Studio • 1 Bath	$1,975.00	40.746509	-73.978469	1.535379	10016	34.7
0	174 Mulberry St 2	174 Mulberry St New York, NY 10013	Studio • 1 Bath	$1,650.00	40.719998	-73.996902	0.650935	10013	36.4
1	349 W 45th St BW	349 W 45th St New York, NY 10036	1 Bed • 1 Bath	$1,750.00	40.760182	-73.990272	2.153141	10036	49.9
8	297 Broadway 3F	297 Broadway Brooklyn, NY 11211	1 Bed • 1 Bath	$2,000.00	40.708554	-73.958006	2.512521	11211	51.5
	81	81 Waverly	Studio						

4	Waverly Ave 3	Ave Brooklyn, NY 11205	• 1 Bath	$1,850.00	40.695231	-73.968025	2.813850	11205	61.7
6	510 E 5th St B	510 E 5th St New York, NY 10009	Studio • 1 Bath	$1,975.00	40.724452	-73.983861	0.780736	10009	63.3

```
In [30]: df_4.to_csv('crime_rate.csv')
```

Later, I would like to show both address and crime rates information on popups, so first, I need to combine these information into one single string for each place.

```
In [31]: pops = []
         for address, vcrime_rate, pcrime_rate in zip(df_4['Address'], df_4['Vic
             my_string = 'address: {}, violent crime rate: {}, property crime ra
             pops.append(my_string)
         pops
```

```
Out[31]: ['address: 5 Tudor City Pl New York, NY 10017, violent crime rate:
         28.7, property crime rate: 37.8',
          'address: 25 Tudor City Pl New York, NY 10017, violent crime rate:
         28.7, property crime rate: 37.8',
          'address: 177 Congress St Brooklyn, NY 11201, violent crime rate:
         32.7, property crime rate: 28.5',
          'address: 226 E 26th St New York, NY 10010, violent crime rate: 33
         .5, property crime rate: 43.0',
          'address: 152 E 35th St New York, NY 10016, violent crime rate: 34
         .7, property crime rate: 44.1',
          'address: 174 Mulberry St New York, NY 10013, violent crime rate:
         36.4, property crime rate: 44.7',
          'address: 349 W 45th St New York, NY 10036, violent crime rate: 49
         .9, property crime rate: 58.2',
          'address: 297 Broadway Brooklyn, NY 11211, violent crime rate: 51.
         5, property crime rate: 50.1',
          'address: 81 Waverly Ave Brooklyn, NY 11205, violent crime rate: 6
         1.7, property crime rate: 60.3',
          'address: 510 E 5th St New York, NY 10009, violent crime rate: 63.
         3, property crime rate: 67.0']
```

Let's visualize the map again.

In [32]:

```
filtered_map = folium.Map(location=[NYU_latitude, NYU_longitude], zoom_

# add a red circle marker to represent the New York University
folium.features.CircleMarker(
    [NYU_latitude, NYU_longitude],
    radius=10,
    color='red',
    popup='New York University',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(filtered_map)

# add the Italian restaurants as blue circle markers
for pop, lat, lng in zip(pops, df_4['Latitude'], df_4['Longitude']):
    folium.features.CircleMarker(
        [lat, lng],
        radius=5,
        color='blue',
        popup= pop,
        fill = True,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(filtered_map)

# display map
filtered_map
```

Out[32]:

3.3 Explore the restaurant venues nearby

Next, let's explore the venues for each place and figure out whether they have my favorite Chinese food and Janpanese food nearby.

In [33]:

```
address_list = df_4['Address'].tolist()
latitude_list = df_4['Latitude'].tolist()
longitude_list = df_4['Longitude'].tolist()
#print(address_list)
#print(latitude_list)
#print(longitude_list)
```

In [34]:

```
# @hidden_cell
CLIENT_ID = 'UOQC5VBGMXKL0MWLS******************' # your Foursquare ID
CLIENT_SECRET = 'HH4WDD4VPMN0********************' # your Foursquare Se
VERSION = '201906013'
limit=200
#print('Your credentails:')
#print('CLIENT_ID: ' + CLIENT_ID)
#print('CLIENT_SECRET:' + CLIENT_SECRET)
```

In [35]:
```
def getNearbyVenues(addresses, latitudes, longitudes, radius=500):

    venues_list=[]
    for address, lat, lng in zip(addresses, latitudes, longitudes):
        #print(address)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id=
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            query,
            radius,
            limit)

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

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

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

    return(nearby_venues)
```

In [36]:
```
query = 'Restaurant'
food_venues = getNearbyVenues(addresses=address_list, latitudes=latitud
food_venues
```

Out[36]:

	Address	Latitude	Longitude	Venue	Venue ID	Venue Category
0	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Sai Gon Dep	5c1a6dd6121384002ce93aff	Vietnamese Restaurant
	5 Tudor					

1	City Pl New York, NY 10017	40.748640	-73.970494	Pietro's	4b2042fff964a5208f2f24e3	Italian Restaurant
2	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Patsy's Pizzeria	50f9918de4b06a4e33f7bc7b	Pizza Place
3	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Sushi Yasuda	450972f5f964a5203d391fe3	Sushi Restaurant
4	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	El Pote	4ba19d8bf964a52005c437e3	Spanish Restaurant
5	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Alcala	4a8489f0f964a520eafc1fe3	Spanish Restaurant
6	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Pleroma	5b55d92bb5cb76002c7a013c	Deli / Bodega
7	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	696 Gourmet Deli	4b86d45cf964a52032a031e3	Restaurant
8	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Mighty Bowl	59723de1033693233be0f8fe	Asian Restaurant
9	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Tempura Matsui	559cbaa6498eaa4e8d884811	Japanese Restaurant
10	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Soba Totto	49ef3cedf964a5208b681fe3	Soba Restaurant
11	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Sushi Ryusei	5b88b22f16ef67002ce49afe	Sushi Restaurant
	5 Tudor City Pl					

12	New York, NY 10017	40.748640	-73.970494	Tudor Cafe	50ca02be245f2d4aa8c2acf5	Deli / Bodega
13	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Ben & Jack's Steakhouse	5b3e9929ca18ea002c8ba1f2	Steakhouse
14	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Luke's Lobster	53c5b7ad498e0b3a75ca97d0	Seafood Restaurant
15	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Kiosku	51789d26e4b060cad8760e22	Deli / Bodega
16	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Sons of Thunder	563c0261cd102a339785c7db	Hawaiian Restaurant
17	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Shake Shack	5460fb3b498e94400cc7fd29	Burger Joint
18	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	The Kati Roll Company	59fcd48c464d6567ed2f5e37	Indian Restaurant
19	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	The Palm Too	4b61c759f964a5203a222ae3	Steakhouse
20	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Aburiya Kinnosuke	49db8b67f964a520d85e1fe3	Japanese Restaurant
21	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Naya Express	4b4cb455f964a520fbba26e3	Lebanese Restaurant
22	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Tsushima	536ee0fb498e265dd8cb4a93	Japanese Restaurant
	5 Tudor City Pl			Sachi Asian		Asian

23	New York, NY 10017	40.748640	-73.970494	Bistro	540b8829498ef527c96a6973	Restaurant
24	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Herbs Thai Bistro	548cb1a2498e7cd8fd9f98ce	Thai Restaurant
25	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	KaoruMC	577445a4cd100bd4db96ed96	Sushi Restaurant
26	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	The Capital Grille	49e79ffdf964a520e3641fe3	American Restaurant
27	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	John's Coffee Shop	4b310484f964a5204dfe24e3	Diner
28	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Tacos Grand Central	59650bb0531593755b6b940c	Taco Place
29	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Zengo New York	4bb7d3f77421a593c120c140	Latin American Restaurant
...	...	...	...	...	...	...
836	510 E 5th St New York, NY 10009	40.724452	-73.983861	Poco	49dbf8f0f964a520305f1fe3	Tapas Restaurant
837	510 E 5th St New York, NY 10009	40.724452	-73.983861	Mayanoki	5928aa5d780eee6f1f1aebc3	Sushi Restaurant
838	510 E 5th St New York, NY 10009	40.724452	-73.983861	Eats Khao Man Gai	5b7619b2237dee002ca790fe	Thai Restaurant
839	510 E 5th St New York, NY 10009	40.724452	-73.983861	Lucien	3fd66200f964a5202de51ee3	French Restaurant
	510 E					

840	5th St New York, NY 10009	40.724452	-73.983861	Cornerstone Cafe	4dea657bfa76cc1b8ae3bb30	Café
841	510 E 5th St New York, NY 10009	40.724452	-73.983861	Boulton & Watt	50d4d50ee4b0260dfc1efe72	Gastropub
842	510 E 5th St New York, NY 10009	40.724452	-73.983861	MONO+MONO	4c78103edf08a1cd411fd65d	Wings Joint
843	510 E 5th St New York, NY 10009	40.724452	-73.983861	Tramonti Pizza	5924f135e97dfb49b5f28473	Italian Restaurant
844	510 E 5th St New York, NY 10009	40.724452	-73.983861	Sweet Chick	537b5a29498ec121cf9fa1f4	Southern / Soul Food Restaurant
845	510 E 5th St New York, NY 10009	40.724452	-73.983861	Sushi Dojo NYC	51bb39d77dd2ab9f2ed9f601	Sushi Restaurant
846	510 E 5th St New York, NY 10009	40.724452	-73.983861	Cafe Himalaya	4a85ffcdf964a520060020e3	Himalayan Restaurant
847	510 E 5th St New York, NY 10009	40.724452	-73.983861	Spiegel	5324e729498e6444d5ee8271	Café
848	510 E 5th St New York, NY 10009	40.724452	-73.983861	Whitmans	4c0d0b440e9e9521c7d13cbd	Burger Joint
849	510 E 5th St New York, NY 10009	40.724452	-73.983861	Spicy Moon	5c4a63b11987ec002cddaece	Vegetarian / Vegan Restaurant
850	510 E 5th St New York, NY 10009	40.724452	-73.983861	Arepa Factory	560ef205498e7c96491fa974	Arepa Restaurant
	510 E 5th St					

851	New York, NY 10009	40.724452	-73.983861	Dirty French	53b8c811498ee296db5cfaed	French Restaurant
852	510 E 5th St New York, NY 10009	40.724452	-73.983861	The Organic Grill	4eaca19730f855f5b2feed5a	Vegetarian / Vegan Restaurant
853	510 E 5th St New York, NY 10009	40.724452	-73.983861	Zum Schneider	3fd66200f964a52022e51ee3	German Restaurant
854	510 E 5th St New York, NY 10009	40.724452	-73.983861	Tompkins Square Bagels	4d38f531a558a1cde4c1cd43	Bagel Shop
855	510 E 5th St New York, NY 10009	40.724452	-73.983861	Dumpling Man	41575800f964a5202f1d1fe3	Dumpling Restaurant
856	510 E 5th St New York, NY 10009	40.724452	-73.983861	noreetuh	5504c48e498e2f599c554792	Hawaiian Restaurant
857	510 E 5th St New York, NY 10009	40.724452	-73.983861	V-Nam Cafe	4c9e5229ca44236af6b32999	Vietnamese Restaurant
858	510 E 5th St New York, NY 10009	40.724452	-73.983861	Caravan of Dreams	42dc4280f964a52036261fe3	Vegetarian / Vegan Restaurant
859	510 E 5th St New York, NY 10009	40.724452	-73.983861	Le French Diner	5303f306498ebbca8f7e1158	French Restaurant
860	510 E 5th St New York, NY 10009	40.724452	-73.983861	Izakaya	5452c0ad498ef95447ead60c	Japanese Restaurant
861	510 E 5th St New York, NY 10009	40.724452	-73.983861	Awash	49f7a827f964a520cd6c1fe3	Ethiopian Restaurant
	510 E 5th St					

862	New York, NY 10009	40.724452	-73.983861	Balvanera	53e02d67498e78c091589343	Argentinian Restaurant
863	510 E 5th St New York, NY 10009	40.724452	-73.983861	Malai Marke	510431dde4b0f2b954d00297	Indian Restaurant
864	510 E 5th St New York, NY 10009	40.724452	-73.983861	Benemon	565e3a9d498e679c06db9915	Japanese Restaurant
865	510 E 5th St New York, NY 10009	40.724452	-73.983861	The MasalaWala	4eaf0d987beb08c1a60c130f	Indian Restaurant

866 rows × 6 columns

Since I would like to figure out whether the rental places have either Chinese or Japanese restaurants nearby, next, I would like to filter the food_venues data.

In [37]:
```
food_venues_1 = food_venues[food_venues['Venue Category'].str.contains(
food_venues_1
```

Out[37]:

	Address	Latitude	Longitude	Venue	Venue ID	Venue Category
9	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Tempura Matsui	559cbaa6498eaa4e8d884811	Japanese Restaurant
20	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Aburiya Kinnosuke	49db8b67f964a520d85e1fe3	Japanese Restaurant
22	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Tsushima	536ee0fb498e265dd8cb4a93	Japanese Restaurant
45	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Curry-Ya	59a0500ae679bc4d9c6a0de1	Japanese Curry Restaurant
66	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Ming's Delight	4ad12033f964a520e8dc20e3	Chinese Restaurant
68	5 Tudor City Pl New York,	40.748640	-73.970494	Six Happiness	4b5a5efff964a520b1c028e3	Chinese Restaurant

	NY 10017					
93	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Tempura Matsui	559cbaa6498eaa4e8d884811	Japanese Restaurant
104	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Aburiya Kinnosuke	49db8b67f964a520d85e1fe3	Japanese Restaurant
106	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Tsushima	536ee0fb498e265dd8cb4a93	Japanese Restaurant
129	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Curry-Ya	59a0500ae679bc4d9c6a0de1	Japanese Curry Restaurant
150	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Ming's Delight	4ad12033f964a520e8dc20e3	Chinese Restaurant
152	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Six Happiness	4b5a5efff964a520b1c028e3	Chinese Restaurant
168	177 Congress St Brooklyn, NY 11201	40.688475	-73.995646	Hibino	48a41073f964a52091511fe3	Japanese Restaurant
184	177 Congress St Brooklyn, NY 11201	40.688475	-73.995646	Mitoushi Sushi	4bafd791f964a5203c253ce3	Japanese Restaurant
246	177 Congress St Brooklyn, NY 11201	40.688475	-73.995646	Chan's Golden City	4b8c86bdf964a52028d532e3	Chinese Restaurant
247	177 Congress St Brooklyn, NY 11201	40.688475	-73.995646	Happy Garden	4ee168e38231cd14a23f4197	Chinese Restaurant
279	226 E 26th St New York, NY 10010	40.739994	-73.980549	Natsumi Tapas	56e8c670cd1017cb5460dc3f	Japanese Restaurant
284	226 E 26th St New York, NY 10010	40.739994	-73.980549	o ya	555fdf00498ee79eb2118581	Japanese Restaurant
	226 E					

303	26th St New York, NY 10010	40.739994	-73.980549	Burp Bowl Café	56b3fa1f498e61baecfcf541	Chinese Restaurant
310	226 E 26th St New York, NY 10010	40.739994	-73.980549	Oishi Bay Japanese Cuisine	52e55531498e7a42cd0c25b1	Japanese Restaurant
326	226 E 26th St New York, NY 10010	40.739994	-73.980549	Oka	5997a9d8123a1921c379e14c	Japanese Restaurant
363	152 E 35th St New York, NY 10016	40.746509	-73.978469	Dun Huang	5bd497a9f427de003989e9d8	Chinese Restaurant
366	152 E 35th St New York, NY 10016	40.746509	-73.978469	Kajitsu	4a99b4f4f964a520f62f20e3	Japanese Restaurant
370	152 E 35th St New York, NY 10016	40.746509	-73.978469	Tempura Matsui	559cbaa6498eaa4e8d884811	Japanese Restaurant
391	152 E 35th St New York, NY 10016	40.746509	-73.978469	Wokuni	59e005d00d8a0f671ecd2a68	Japanese Restaurant
393	152 E 35th St New York, NY 10016	40.746509	-73.978469	Café China	4e3484038877beb5e9a22a0b	Chinese Restaurant
395	152 E 35th St New York, NY 10016	40.746509	-73.978469	Zuma New York	540dc46d498e86f0e5059dd2	Japanese Restaurant
413	152 E 35th St New York, NY 10016	40.746509	-73.978469	Oka	5997a9d8123a1921c379e14c	Japanese Restaurant
416	152 E 35th St New York, NY 10016	40.746509	-73.978469	Hakubai	4bbe8652ba9776b075e8fdc8	Japanese Restaurant
487	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Yi Ji Shi Mo Noodle Corp	5c965dad5455b2002c058659	Chinese Restaurant
...	...	...	...	...	...	...
525	174 Mulberry St New York, NY	40.719998	-73.996902	May Wah Pork Chop Fast Food	3fd66200f964a52099e61ee3	Chinese Restaurant

	10013					
530	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Joe’s Steam Rice Roll	5bf4559c65cdf8002c6be196	Chinese Restaurant
540	349 W 45th St New York, NY 10036	40.760182	-73.990272	Gyu-Kaku Japanese BBQ	4ef0d81fbe7ba3ed7c2296d2	Japanese Restaurant
545	349 W 45th St New York, NY 10036	40.760182	-73.990272	Hakkasan	4ea832fe8b8154b19ffa31ad	Chinese Restaurant
555	349 W 45th St New York, NY 10036	40.760182	-73.990272	Sake Bar Hagi 46	56397ed1cd104946e8ef4ed9	Japanese Restaurant
576	349 W 45th St New York, NY 10036	40.760182	-73.990272	Panda Express	57bdfbb4498e68c4b71697da	Chinese Restaurant
614	349 W 45th St New York, NY 10036	40.760182	-73.990272	Kung Fu Little Steamed Buns Ramen	52b782a6498ee2a8438f4150	Chinese Restaurant
643	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	Shalom Japan	51f9b7b3498eefe896caeb23	Japanese Restaurant
665	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	Wei Williamsburg	54b318a1498eb29e9541053a	Chinese Restaurant
668	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	M Shanghai Bistro	3fd66200f964a5206bf11ee3	Chinese Restaurant
688	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	Good Friend	4bfb378265fbc9b61233916c	Chinese Restaurant
692	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	Sumo Sushi	4b70a754f964a520ca272de3	Japanese Restaurant
709	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	Kum Kau	4a71ff01f964a52016da1fe3	Chinese Restaurant
714	81 Waverly Ave	40.695231	-73.968025	U-Gu	591f7c5f35f9836382455eca	Japanese

	Brooklyn, NY 11205					Restaurant
728	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	Waza	4e3cadcb1f6e844231dbe017	Japanese Restaurant
740	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	New Grace Chinese Kitchen	4a9efbcbf964a5200c3c20e3	Chinese Restaurant
745	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	Yummy Yummy	4b5b8ae1f964a520dc0529e3	Chinese Restaurant
755	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	Park Ave Chinese	4d0d1b2c257d6dcbb5269a67	Chinese Restaurant
756	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	66S Fusion	5aa09a586eda026624b25ecb	Japanese Restaurant
758	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	Hardee	521bf3b711d2dce4f1b42195	Chinese Restaurant
762	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	Chung Chun Kitchen	4b10b7a6f964a520117523e3	Chinese Restaurant
763	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	E-bite 東	5c14068db399f7002c1986a2	Japanese Restaurant
771	510 E 5th St New York, NY 10009	40.724452	-73.983861	Secchu Yokota	57a3bda338fa6612ad9c17f2	Japanese Restaurant
783	510 E 5th St New York, NY 10009	40.724452	-73.983861	Kura	510c85e7e4b0056826b88297	Japanese Restaurant
807	510 E 5th St New York, NY 10009	40.724452	-73.983861	Málà Project	5647ee82498e8bfc0ddef53d	Chinese Restaurant

818	510 E 5th St New York, NY 10009	40.724452	-73.983861	Suki Curry	5b3805910dd31f002409405a	Japanese Restaurant
828	510 E 5th St New York, NY 10009	40.724452	-73.983861	Xi'an Famous Foods	4c2a81398abca59393c8fe1f	Chinese Restaurant
829	510 E 5th St New York, NY 10009	40.724452	-73.983861	Umi No Ie	446b2f7cf964a52058331fe3	Japanese Restaurant
860	510 E 5th St New York, NY 10009	40.724452	-73.983861	Izakaya	5452c0ad498ef95447ead60c	Japanese Restaurant
864	510 E 5th St New York, NY 10009	40.724452	-73.983861	Benemon	565e3a9d498e679c06db9915	Japanese Restaurant

65 rows × 6 columns

Venue ID will be needed for scraping rating information of each restaurant.

In [38]:
```
venue_id_list = food_venues_1['Venue ID'].tolist()
venue_id_list
```

Out[38]:
```
['559cbaa6498eaa4e8d884811',
 '49db8b67f964a520d85e1fe3',
 '536ee0fb498e265dd8cb4a93',
 '59a0500ae679bc4d9c6a0de1',
 '4ad12033f964a520e8dc20e3',
 '4b5a5efff964a520b1c028e3',
 '559cbaa6498eaa4e8d884811',
 '49db8b67f964a520d85e1fe3',
 '536ee0fb498e265dd8cb4a93',
 '59a0500ae679bc4d9c6a0de1',
 '4ad12033f964a520e8dc20e3',
 '4b5a5efff964a520b1c028e3',
 '48a41073f964a52091511fe3',
 '4bafd791f964a5203c253ce3',
 '4b8c86bdf964a52028d532e3',
 '4ee168e38231cd14a23f4197',
 '56e8c670cd1017cb5460dc3f',
 '555fdf00498ee79eb2118581',
 '56b3fa1f498e61baecfcf541',
 '52e55531498e7a42cd0c25b1',
 '5997a9d8123a1921c379e14c',
 '5bd497a9f427de003989e9d8',
 '4a99b4f4f964a520f62f20e3',
 '559cbaa6498eaa4e8d884811',
 '59e005d00d8a0f671ecd2a68',
 '4e3484038877beb5e9a22a0b',
 '540dc46d498e86f0e5059dd2',
 '5997a9d8123a1921c379e14c',
 '4bbe8652ba9776b075e8fdc8',
 '5c965dad5455b2002c058659',
 '4cf9655334c1a09374cb390e',
 '542074d8498e537b40b198ac',
 '59ad625a2619ee5cdded881f',
 '56ddc1f8cd1028ec415bf35a',
 '54ef6f6d498e47c1606c63fd',
 '3fd66200f964a52099e61ee3',
 '5bf4559c65cdf8002c6be196',
 '4ef0d81fbe7ba3ed7c2296d2',
 '4ea832fe8b8154b19ffa31ad',
 '56397ed1cd104946e8ef4ed9',
 '57bdfbb4498e68c4b71697da',
 '52b782a6498ee2a8438f4150',
 '51f9b7b3498eefe896caeb23',
 '54b318a1498eb29e9541053a',
 '3fd66200f964a5206bf11ee3',
 '4bfb378265fbc9b61233916c',
 '4b70a754f964a520ca272de3',
 '4a71ff01f964a52016da1fe3',
 '591f7c5f35f9836382455eca',
 '4e3cadcb1f6e844231dbe017',
 '4a9efbcbf964a5200c3c20e3',
 '4b5b8ae1f964a520dc0529e3',
 '4d0d1b2c257d6dcbb5269a67',
 '5aa09a586eda026624b25ecb',
 '521bf3b711d2dce4f1b42195',
 '4b10b7a6f964a520117523e3',
 '5c14068db399f7002c1986a2',
 '57a3bda338fa6612ad9c17f2',
 '510c85e7e4b0056826b88297',
 '5647ee82498e8bfc0ddef53d',
 '5b3805910dd31f002409405a',
 '4c2a81398abca59393c8fe1f',
 '446b2f7cf964a52058331fe3',
 '5452c0ad498ef95447ead60c',
 '565e3a9d498e679c06db9915']
```

To get the rating of each restaurant, I have to explore each venue. https://api.foursquare.com/v2/venues/VENUE_ID?client_id=CLIENT_ID&client_secret=CLIENT_SECRET&v=VERSION (https://api.foursquare.com/v2/venues/VENUE_ID?client_id=CLIENT_ID&client_secret=CLIENT_SECRET&v=VERSION)

In [39]:

```python
Rating=[]
for venue_id in venue_id_list:
    url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_
    result = requests.get(url).json()
    try:
        r = result['response']['venue']['rating']
        #print(r)
    except:
        r = 'NAN'
        #print(r)
    Rating.append(r)
print(Rating)
```

```
[8.6, 8.9, 8.1, 7.2, 6.3, 6.0, 8.6, 8.9, 8.1, 7.2, 6.3, 6.0, 9.2, 8
.0, 'NAN', 'NAN', 7.7, 8.8, 7.6, 7.4, 7.5, 8.3, 8.7, 8.6, 7.7, 8.9,
8.6, 7.5, 7.4, 8.3, 9.1, 7.9, 8.1, 7.8, 7.7, 7.7, 8.7, 9.0, 8.7, 8.
3, 8.0, 8.2, 8.2, 7.7, 8.1, 'NAN', 5.8, 7.8, 8.7, 6.9, 6.2, 'NAN',
'NAN', 'NAN', 'NAN', 'NAN', 'NAN', 'NAN', 'NAN', 'NAN', 'NAN', 'NAN
', 'NAN', 'NAN', 'NAN']
```

In [40]:

```
food_venues_1['Rating']=Rating
food_venues_1.head()
```

```
/opt/conda/envs/DSX-Python35/lib/python3.5/site-packages/ipykernel/
__main__.py:1: 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: http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy
(http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing
-view-versus-copy)
  if __name__ == '__main__':
```

Out[40]:

	Address	Latitude	Longitude	Venue	Venue ID	Venue Category	Rating
9	5 Tudor City Pl New York, NY 10017	40.74864	-73.970494	Tempura Matsui	559cbaa6498eaa4e8d884811	Japanese Restaurant	8.6
20	5 Tudor City Pl New York, NY 10017	40.74864	-73.970494	Aburiya Kinnosuke	49db8b67f964a520d85e1fe3	Japanese Restaurant	8.9
22	5 Tudor City Pl New York, NY 10017	40.74864	-73.970494	Tsushima	536ee0fb498e265dd8cb4a93	Japanese Restaurant	8.1
45	5 Tudor City Pl New York, NY 10017	40.74864	-73.970494	Curry-Ya	59a0500ae679bc4d9c6a0de1	Japanese Curry Restaurant	7.2
66	5 Tudor City Pl New York, NY 10017	40.74864	-73.970494	Ming's Delight	4ad12033f964a520e8dc20e3	Chinese Restaurant	6.3

After getting the rating, I would like to drop those haven't been rated yet.

In [41]:

```
food_venues_2 = food_venues_1[food_venues_1['Rating']!='NAN']
food_venues_2
```

Out[41]:

	Address	Latitude	Longitude	Venue	Venue ID	Venue Category	R
	5 Tudor City Pl			Tempura		Japanese	

9	New York, NY 10017	40.748640	-73.970494	Matsui	559cbaa6498eaa4e8d884811	Restaurant
20	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Aburiya Kinnosuke	49db8b67f964a520d85e1fe3	Japanese Restaurant
22	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Tsushima	536ee0fb498e265dd8cb4a93	Japanese Restaurant
45	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Curry-Ya	59a0500ae679bc4d9c6a0de1	Japanese Curry Restaurant
66	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Ming's Delight	4ad12033f964a520e8dc20e3	Chinese Restaurant
68	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Six Happiness	4b5a5efff964a520b1c028e3	Chinese Restaurant
93	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Tempura Matsui	559cbaa6498eaa4e8d884811	Japanese Restaurant
104	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Aburiya Kinnosuke	49db8b67f964a520d85e1fe3	Japanese Restaurant
106	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Tsushima	536ee0fb498e265dd8cb4a93	Japanese Restaurant
129	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Curry-Ya	59a0500ae679bc4d9c6a0de1	Japanese Curry Restaurant
150	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Ming's Delight	4ad12033f964a520e8dc20e3	Chinese Restaurant
152	25 Tudor City Pl New	40.748636	-73.970505	Six	4b5a5efff964a520b1c028e3	Chinese

	York, NY 10017			Happiness		Restaurant
168	177 Congress St Brooklyn, NY 11201	40.688475	-73.995646	Hibino	48a41073f964a52091511fe3	Japanese Restaurant
184	177 Congress St Brooklyn, NY 11201	40.688475	-73.995646	Mitoushi Sushi	4bafd791f964a5203c253ce3	Japanese Restaurant
279	226 E 26th St New York, NY 10010	40.739994	-73.980549	Natsumi Tapas	56e8c670cd1017cb5460dc3f	Japanese Restaurant
284	226 E 26th St New York, NY 10010	40.739994	-73.980549	o ya	555fdf00498ee79eb2118581	Japanese Restaurant
303	226 E 26th St New York, NY 10010	40.739994	-73.980549	Burp Bowl Café	56b3fa1f498e61baecfcf541	Chinese Restaurant
310	226 E 26th St New York, NY 10010	40.739994	-73.980549	Oishi Bay Japanese Cuisine	52e55531498e7a42cd0c25b1	Japanese Restaurant
326	226 E 26th St New York, NY 10010	40.739994	-73.980549	Oka	5997a9d8123a1921c379e14c	Japanese Restaurant
363	152 E 35th St New York, NY 10016	40.746509	-73.978469	Dun Huang	5bd497a9f427de003989e9d8	Chinese Restaurant
366	152 E 35th St New York, NY 10016	40.746509	-73.978469	Kajitsu	4a99b4f4f964a520f62f20e3	Japanese Restaurant
370	152 E 35th St New York, NY 10016	40.746509	-73.978469	Tempura Matsui	559cbaa6498eaa4e8d884811	Japanese Restaurant
391	152 E 35th St New	40.746509	-73.978469	Wokuni	59e005d00d8a0f671ecd2a68	Japanese Restaurant

	York, NY 10016					
393	152 E 35th St New York, NY 10016	40.746509	-73.978469	Café China	4e3484038877beb5e9a22a0b	Chinese Restaurant
395	152 E 35th St New York, NY 10016	40.746509	-73.978469	Zuma New York	540dc46d498e86f0e5059dd2	Japanese Restaurant
413	152 E 35th St New York, NY 10016	40.746509	-73.978469	Oka	5997a9d8123a1921c379e14c	Japanese Restaurant
416	152 E 35th St New York, NY 10016	40.746509	-73.978469	Hakubai	4bbe8652ba9776b075e8fdc8	Japanese Restaurant
487	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Yi Ji Shi Mo Noodle Corp	5c965dad5455b2002c058659	Chinese Restaurant
503	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Cocoron	4cf9655334c1a09374cb390e	Japanese Restaurant
506	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Yakitori Tora	542074d8498e537b40b198ac	Japanese Restaurant
507	174 Mulberry St New York, NY 10013	40.719998	-73.996902	House Special 甘來飯店	59ad625a2619ee5cdded881f	Chinese Restaurant
513	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Mimi Cheng's	56ddc1f8cd1028ec415bf35a	Chinese Restaurant
515	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Goemon	54ef6f6d498e47c1606c63fd	Japanese Curry Restaurant
525	174 Mulberry St New York, NY	40.719998	-73.996902	May Wah Pork Chop Fast Food	3fd66200f964a52099e61ee3	Chinese Restaurant

	10013					
530	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Joe’s Steam Rice Roll	5bf4559c65cdf8002c6be196	Chinese Restaurant
540	349 W 45th St New York, NY 10036	40.760182	-73.990272	Gyu-Kaku Japanese BBQ	4ef0d81fbe7ba3ed7c2296d2	Japanese Restaurant
545	349 W 45th St New York, NY 10036	40.760182	-73.990272	Hakkasan	4ea832fe8b8154b19ffa31ad	Chinese Restaurant
555	349 W 45th St New York, NY 10036	40.760182	-73.990272	Sake Bar Hagi 46	56397ed1cd104946e8ef4ed9	Japanese Restaurant
576	349 W 45th St New York, NY 10036	40.760182	-73.990272	Panda Express	57bdfbb4498e68c4b71697da	Chinese Restaurant
614	349 W 45th St New York, NY 10036	40.760182	-73.990272	Kung Fu Little Steamed Buns Ramen	52b782a6498ee2a8438f4150	Chinese Restaurant
643	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	Shalom Japan	51f9b7b3498eefe896caeb23	Japanese Restaurant
665	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	Wei Williamsburg	54b318a1498eb29e9541053a	Chinese Restaurant
668	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	M Shanghai Bistro	3fd66200f964a5206bf11ee3	Chinese Restaurant
692	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	Sumo Sushi	4b70a754f964a520ca272de3	Japanese Restaurant
709	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	Kum Kau	4a71ff01f964a52016da1fe3	Chinese Restaurant
	81 Waverly					

714	Ave Brooklyn, NY 11205	40.695231	-73.968025	U-Gu	591f7c5f35f9836382455eca	Japanese Restaurant
728	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	Waza	4e3cadcb1f6e844231dbe017	Japanese Restaurant
740	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	New Grace Chinese Kitchen	4a9efbcbf964a5200c3c20e3	Chinese Restaurant

Also drop those whose rating is less than 7.

In [42]:
```
food_venues_3 = food_venues_2[food_venues_2['Rating']>7]
food_venues_3
```

Out[42]:

	Address	Latitude	Longitude	Venue	Venue ID	Venue Category	R
9	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Tempura Matsui	559cbaa6498eaa4e8d884811	Japanese Restaurant	
20	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Aburiya Kinnosuke	49db8b67f964a520d85e1fe3	Japanese Restaurant	
22	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Tsushima	536ee0fb498e265dd8cb4a93	Japanese Restaurant	
45	5 Tudor City Pl New York, NY 10017	40.748640	-73.970494	Curry-Ya	59a0500ae679bc4d9c6a0de1	Japanese Curry Restaurant	
93	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Tempura Matsui	559cbaa6498eaa4e8d884811	Japanese Restaurant	
104	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Aburiya Kinnosuke	49db8b67f964a520d85e1fe3	Japanese Restaurant	
	25 Tudor City Pl						

106	New York, NY 10017	40.748636	-73.970505	Tsushima	536ee0fb498e265dd8cb4a93	Japanese Restaurant
129	25 Tudor City Pl New York, NY 10017	40.748636	-73.970505	Curry-Ya	59a0500ae679bc4d9c6a0de1	Japanese Curry Restaurant
168	177 Congress St Brooklyn, NY 11201	40.688475	-73.995646	Hibino	48a41073f964a52091511fe3	Japanese Restaurant
184	177 Congress St Brooklyn, NY 11201	40.688475	-73.995646	Mitoushi Sushi	4bafd791f964a5203c253ce3	Japanese Restaurant
279	226 E 26th St New York, NY 10010	40.739994	-73.980549	Natsumi Tapas	56e8c670cd1017cb5460dc3f	Japanese Restaurant
284	226 E 26th St New York, NY 10010	40.739994	-73.980549	o ya	555fdf00498ee79eb2118581	Japanese Restaurant
303	226 E 26th St New York, NY 10010	40.739994	-73.980549	Burp Bowl Café	56b3fa1f498e61baecfcf541	Chinese Restaurant
310	226 E 26th St New York, NY 10010	40.739994	-73.980549	Oishi Bay Japanese Cuisine	52e55531498e7a42cd0c25b1	Japanese Restaurant
326	226 E 26th St New York, NY 10010	40.739994	-73.980549	Oka	5997a9d8123a1921c379e14c	Japanese Restaurant
363	152 E 35th St New York, NY 10016	40.746509	-73.978469	Dun Huang	5bd497a9f427de003989e9d8	Chinese Restaurant
366	152 E 35th St New York, NY 10016	40.746509	-73.978469	Kajitsu	4a99b4f4f964a520f62f20e3	Japanese Restaurant
	152 E 35th St					

370	New York, NY 10016	40.746509	-73.978469	Tempura Matsui	559cbaa6498eaa4e8d884811	Japanese Restaurant
391	152 E 35th St New York, NY 10016	40.746509	-73.978469	Wokuni	59e005d00d8a0f671ecd2a68	Japanese Restaurant
393	152 E 35th St New York, NY 10016	40.746509	-73.978469	Café China	4e3484038877beb5e9a22a0b	Chinese Restaurant
395	152 E 35th St New York, NY 10016	40.746509	-73.978469	Zuma New York	540dc46d498e86f0e5059dd2	Japanese Restaurant
413	152 E 35th St New York, NY 10016	40.746509	-73.978469	Oka	5997a9d8123a1921c379e14c	Japanese Restaurant
416	152 E 35th St New York, NY 10016	40.746509	-73.978469	Hakubai	4bbe8652ba9776b075e8fdc8	Japanese Restaurant
487	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Yi Ji Shi Mo Noodle Corp	5c965dad5455b2002c058659	Chinese Restaurant
503	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Cocoron	4cf9655334c1a09374cb390e	Japanese Restaurant
506	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Yakitori Tora	542074d8498e537b40b198ac	Japanese Restaurant
507	174 Mulberry St New York, NY 10013	40.719998	-73.996902	House Special 甘來飯店	59ad625a2619ee5cdded881f	Chinese Restaurant
513	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Mimi Cheng's	56ddc1f8cd1028ec415bf35a	Chinese Restaurant
515	174 Mulberry	40.719998	-73.996902	Goemon	54ef6f6d498e47c1606c63fd	Japanese

	St New York, NY 10013					Curry Restaurant
525	174 Mulberry St New York, NY 10013	40.719998	-73.996902	May Wah Pork Chop Fast Food	3fd66200f964a52099e61ee3	Chinese Restaurant
530	174 Mulberry St New York, NY 10013	40.719998	-73.996902	Joe’s Steam Rice Roll	5bf4559c65cdf8002c6be196	Chinese Restaurant
540	349 W 45th St New York, NY 10036	40.760182	-73.990272	Gyu-Kaku Japanese BBQ	4ef0d81fbe7ba3ed7c2296d2	Japanese Restaurant
545	349 W 45th St New York, NY 10036	40.760182	-73.990272	Hakkasan	4ea832fe8b8154b19ffa31ad	Chinese Restaurant
555	349 W 45th St New York, NY 10036	40.760182	-73.990272	Sake Bar Hagi 46	56397ed1cd104946e8ef4ed9	Japanese Restaurant
576	349 W 45th St New York, NY 10036	40.760182	-73.990272	Panda Express	57bdfbb4498e68c4b71697da	Chinese Restaurant
614	349 W 45th St New York, NY 10036	40.760182	-73.990272	Kung Fu Little Steamed Buns Ramen	52b782a6498ee2a8438f4150	Chinese Restaurant
643	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	Shalom Japan	51f9b7b3498eefe896caeb23	Japanese Restaurant
665	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	Wei Williamsburg	54b318a1498eb29e9541053a	Chinese Restaurant
668	297 Broadway Brooklyn, NY 11211	40.708554	-73.958006	M Shanghai Bistro	3fd66200f964a5206bf11ee3	Chinese Restaurant
709	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	Kum Kau	4a71ff01f964a52016da1fe3	Chinese Restaurant

714	81 Waverly Ave Brooklyn, NY 11205	40.695231	-73.968025	U-Gu	591f7c5f35f9836382455eca	Japanese Restaurant

In [43]:
```
food_venues_3.to_csv('food_venues_rating.csv')
```

In [44]:
```
print('There are {} places having good-rating Chinese and Japanese rest
```

```
There are 9 places having good-rating Chinese and Japanese restaura
nts.
```

In [45]:
```
address_food_filtered = food_venues_3['Address'].unique()
list(address_food_filtered)
```

Out[45]:
```
['5 Tudor City Pl New York, NY 10017',
 '25 Tudor City Pl New York, NY 10017',
 '177 Congress St Brooklyn, NY 11201',
 '226 E 26th St New York, NY 10010',
 '152 E 35th St New York, NY 10016',
 '174 Mulberry St New York, NY 10013',
 '349 W 45th St New York, NY 10036',
 '297 Broadway Brooklyn, NY 11211',
 '81 Waverly Ave Brooklyn, NY 11205']
```

Let's see how many qualified restaurants does each place have.

In [46]:

```
food_venues_3.groupby('Address').count()
```

Out[46]:

Address	Latitude	Longitude	Venue	Venue ID	Venue Category	Rating
152 E 35th St New York, NY 10016	8	8	8	8	8	8
174 Mulberry St New York, NY 10013	8	8	8	8	8	8
177 Congress St Brooklyn, NY 11201	2	2	2	2	2	2
226 E 26th St New York, NY 10010	5	5	5	5	5	5
25 Tudor City Pl New York, NY 10017	4	4	4	4	4	4
297 Broadway Brooklyn, NY 11211	3	3	3	3	3	3
349 W 45th St New York, NY 10036	5	5	5	5	5	5
5 Tudor City Pl New York, NY 10017	4	4	4	4	4	4
81 Waverly Ave Brooklyn, NY 11205	2	2	2	2	2	2

Not bad, for all of places, I have at least 2 restaurants to go. So these places can be my targeted rental places. However, I need to check their crime rates to figure out better choices.

In [47]:

```
df_5 = df_4[df_4['Address'].isin(list(address_food_filtered))]
df_5
```

Out[47]:

	Name	Address	Type	Price	Latitude	Longitude	Distance from NYU (miles)	ZipCode	Violent Crime Rate
2	5 Tudor City Pl 306	5 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,800.00	40.748640	-73.970494	1.929948	10017	28.7
5	25 Tudor City Pl 1206	25 Tudor City Pl New York, NY 10017	Studio • 1 Bath	$1,950.00	40.748636	-73.970505	1.929326	10017	28.7
9	177 Congress St 4	177 Congress St Brooklyn, NY 11201	Studio • 1 Bath	$2,000.00	40.688475	-73.995646	2.827077	11201	32.7
3	226 E 26th St B-E	226 E 26th St New York, NY 10010	Studio • 1 Bath	$1,825.00	40.739994	-73.980549	1.138868	10010	33.5
7	152 E 35th St 6-H	152 E 35th St New York, NY 10016	Studio • 1 Bath	$1,975.00	40.746509	-73.978469	1.535379	10016	34.7
0	174 Mulberry St 2	174 Mulberry St New York, NY 10013	Studio • 1 Bath	$1,650.00	40.719998	-73.996902	0.650935	10013	36.4
1	349 W 45th St BW	349 W 45th St New York, NY 10036	1 Bed • 1 Bath	$1,750.00	40.760182	-73.990272	2.153141	10036	49.9
8	297 Broadway 3F	297 Broadway Brooklyn, NY 11211	1 Bed • 1 Bath	$2,000.00	40.708554	-73.958006	2.512521	11211	51.5
4	81 Waverly Ave 3	81 Waverly Ave Brooklyn, NY 11205	Studio • 1 Bath	$1,850.00	40.695231	-73.968025	2.813850	11205	61.7

Apparently, **"5 Tudor City Pl 306"** and **"25 Tudo City Pl 1206"** have lower violent crime rate and property crime rate. Meanwhile, they also meet other requirements such as monthly rent less than US $2000, distance from NYU is less than 3 miles, with 4 good_rating restaurants nearby. So I would choose from these two places.

4. Results

A quick recap of the requirements that I have when choosing potential rental places.

1. The monthly rent should not be higher than US $2000.
2. The distance between the potential rental place and New York University should not exceed 3 miles.
3. The crime rates (both violent crime rate and property crime rate) in potential rental places should be relatively low.
4. It will be better to have some good-rating Chinese or Japanese food restaurants nearby.

In order to make better choice, I cleaned the rental information that I scraped from www.rent.com (http://www.rent.com) according to requirements 1 and 2, and narrowed down to 10 places for rent. Later I scraped violent crime rate and property crime rate from https://www.bestplaces.net/crime/zip-code/new_york/new_york/11210 (https://www.bestplaces.net/crime/zip-code/new_york/new_york/11210) for each place by replacing the zipcode. So at this stage, I know the crime rates for these places. For these 10 places, I explored the venues nearby, more specifically, restaurants nearby. Using Foursquare API, I got venues information (name, id) and venue category, which then I filtered out those belong to Chinese or Japanese restaurants. In order to find those restaurants with good-ratings, I used the venue id to and foursquare API again to get the ratings. Those either without rating or raitng less than or equals to 7 were removed from the dataframe. In the end, there are still 9 places left. Subsequently, I went back to check the crime rates for these 9 places and figured out that **"5 Tudor City Pl 306"** and **"25 Tudo City Pl 1206"** have lower crime rates than others. They both have 4 qualified restaurants nearby, and the monthly rent is 1800 and 1950 respectively. The distance to New York University are also within desired range. So I will choose between these two places.

5. Discussion and conclusion

In this project, I would like to find a place to rent near New York University. I set requirements for the monthly rent, type of rental place, distance from New York University, relatively safe environment (lower crime rates) and good-rating Chinese or Japanese food restaurants nearby. Eventually, I narrowed down my choice to two places: **"5 Tudor City Pl 306"** and **"25 Tudo City Pl 1206"**. By using these data and analysis, I could find those places that meet my preferences. It will also be useful for others who will move to a new city to start a new job, because most of the analysis methods are applicable to other cases, what they need to do is just to change to their requirements.