The Battle of Neighborhoods week1 [Part 2]

Popular venue and neighborhood crime in Uptown Chicago

Data

2.1 Data Description

In this section you will get the information about type and source of data and how the data help to solve the problem.

- I use open source Chicago Crime data to provide the user with additional crime data. The data can be found through the following link: https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2

- Chicago uptown neighborhood and community data is scraped from the wiki page. You can found the wiki page here: https://en.wikipedia.org/wiki/List_of_neighborhoods_in_Chicago

- Latitude and longitude of the all the neighborhood was searched in here: https://www.latlong.net/

- I use FourSquare API to get supplemental geographical data about the neighborhood venue

2.2 Data Cleaning

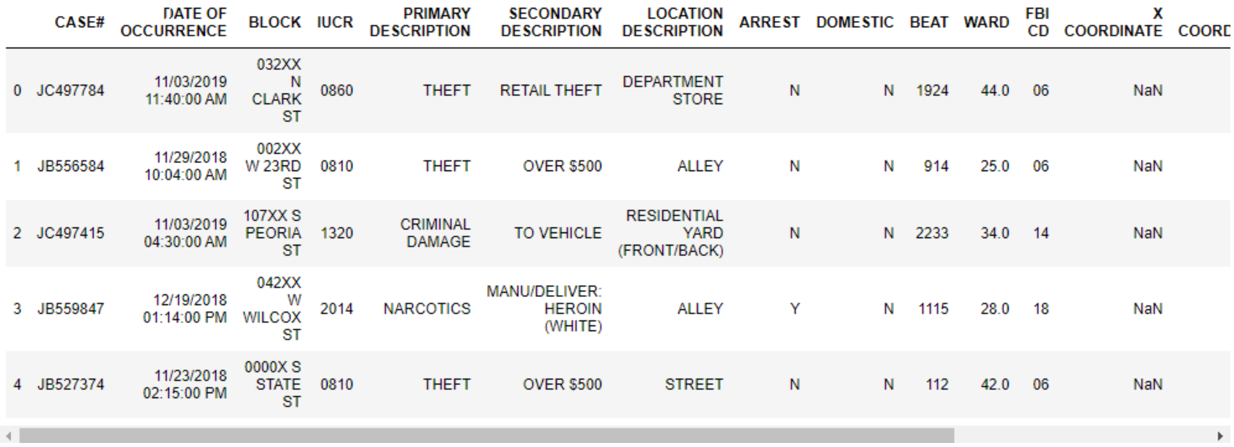
Chicago Crime Data

This dataset reflects reported incidents of crime (with the exception of murders where data exists for each victim) that occurred in the City of Chicago from 2001 to present, minus the most recent seven days. Data is extracted from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system. Due to limitation of computational capabilities we downloads only 2019 data.

Next page you will get crime data column information.

Column Name	**Type**	**Description**
CASE#	Plain Text	The Chicago Police Department RD Number
DATE OF OCCURRENCE	Date & Time	Date when the incident occurred.
BLOCK	Plain Text	The partially redacted address where the incident occurred, placing it on the same block as the actual address.
IUCR	Plain Text	The Illinois Unifrom Crime Reporting code. This is directly linked to the Primary Type and Description.
PRIMARY DESCRIPTION	Plain Text	The primary description of the IUCR code.
SECONDARY DESCRIPTION	Plain Text	The secondary description of the IUCR code, a subcategory of the primary description.
LOCATION DESCRIPTION	Plain Text	Description of the location where the incident occurred.
ARREST	Plain Text	Indicates whether an arrest was made.
DOMESTIC	Plain Text	Indicates whether the incident was domestic-related as defined by The Illinois Domestic Violence Act.
BEAT	Plain Text	Indicates the beat where the incident occurred. A beat is the smallest police geographic area – each beat has a dedicated police beat car. Three to five beats make up a police sector, and three sectors make up a police district. The Chicago Police Department has 22 police districts. See the beats at https://data.cityofchicago.org/d/aerh-rz74.
WARD	Number	The ward (City Council district) where the incident occurred. See the wards at https://data.cityofchicago.org/d/sp34-6z76.
FBI CD	Plain Text	Indicates the crime classification as outlined in the FBI's National Incident-Base

		.
X COORDINATE	Plain Text	The x coordinate of the location where the incident occurred in State Plane Illinois East NAD 1983 projection. This location is shifted from the actual location for partial redaction but falls on the same block.
Y COORDINATE	Plain Text	The y coordinate of the location where the incident occurred in State Plane Illinois East NAD 1983 projection. This location is shifted from the actual location for partial redaction but falls on the same block.
LATITUDE	Number	The latitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block.
LONGITUDE	Number	The longitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block.
LOCATION	Location	The location where the incident occurred in a format that allows for creation of maps and other geographic operations on this data portal. This location is shifted from the actual location for partial redaction but falls on the same block.



	CASE#	DATE OF OCCURRENCE	BLOCK	IUCR	PRIMARY DESCRIPTION	SECONDARY DESCRIPTION	LOCATION DESCRIPTION	ARREST	DOMESTIC	BEAT	WARD	FBI CD	X COORDINATE	COORD
0	JC497784	11/03/2019 11:40:00 AM	032XX N CLARK ST	0860	THEFT	RETAIL THEFT	DEPARTMENT STORE	N	N	1924	44.0	06	NaN	
1	JB556584	11/29/2018 10:04:00 AM	002XX W 23RD ST	0810	THEFT	OVER $500	ALLEY	N	N	914	25.0	06	NaN	
2	JC497415	11/03/2019 04:30:00 AM	107XX S PEORIA ST	1320	CRIMINAL DAMAGE	TO VEHICLE	RESIDENTIAL YARD (FRONT/BACK)	N	N	2233	34.0	14	NaN	
3	JB559847	12/19/2018 01:14:00 PM	042XX W WILCOX ST	2014	NARCOTICS	MANU/DELIVER: HEROIN (WHITE)	ALLEY	Y	N	1115	28.0	18	NaN	
4	JB527374	11/23/2018 02:15:00 PM	0000X S STATE ST	0810	THEFT	OVER $500	STREET	N	N	112	42.0	06	NaN	

Now only the attributes listed below are required for our analysis:

- Date of Occurrence
- Primary Description
- Ward
- Latitude
- Longitude

Uptown is divided into multiple wards, which are the districts from which aldermen in the Chicago City Council are drawn. Most of the community area lies in the 46th and 48th wards, with small portions of the neighborhood's west side located in the 47th and 40th wards. So for this project data for ward 40, 46, 47, 48 are used.

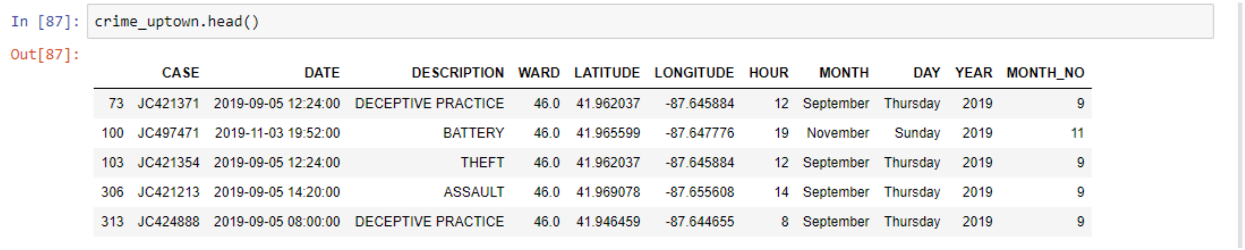
Then the data is processed below:

- Cleanup column name
- Change the date of occurrence field to a date / time object
- Add new columns for:

    1. Hour
    2. Day
    3. Month
    4. Month_NO
    5. Year

- Get data frame for ward 40,46,47,48 and then concat to get the crime data for Uptown.

.



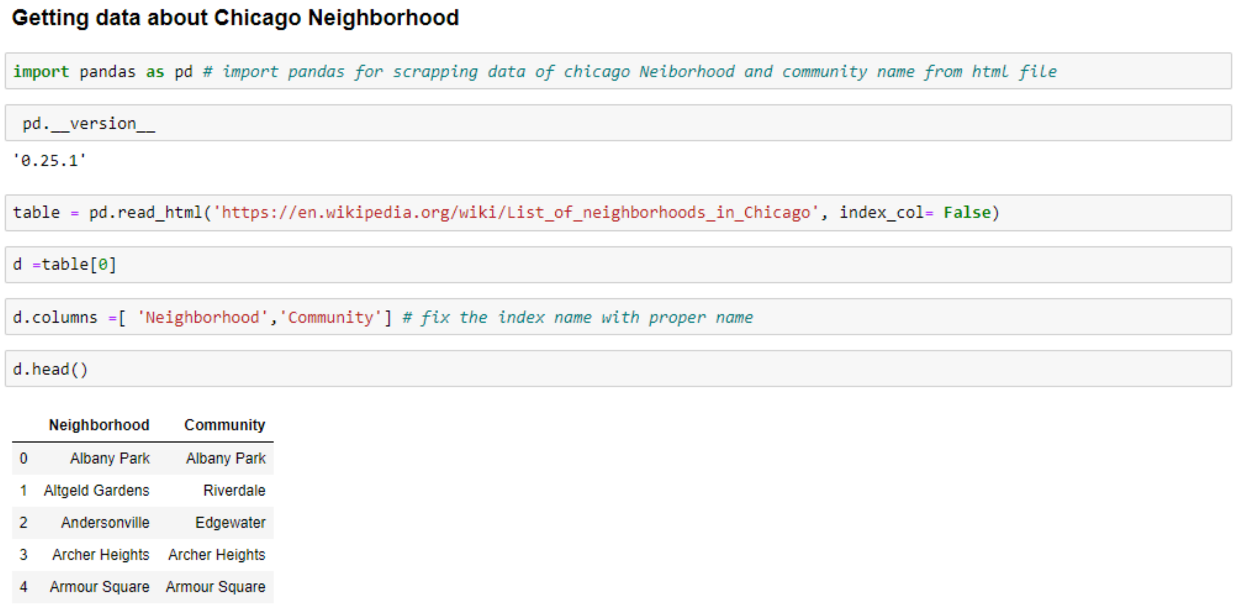
```
In [87]: crime_uptown.head()
```

Out[87]:

	CASE	DATE	DESCRIPTION	WARD	LATITUDE	LONGITUDE	HOUR	MONTH	DAY	YEAR	MONTH_NO
73	JC421371	2019-09-05 12:24:00	DECEPTIVE PRACTICE	46.0	41.962037	-87.645884	12	September	Thursday	2019	9
100	JC497471	2019-11-03 19:52:00	BATTERY	46.0	41.965599	-87.647776	19	November	Sunday	2019	11
103	JC421354	2019-09-05 12:24:00	THEFT	46.0	41.962037	-87.645884	12	September	Thursday	2019	9
306	JC421213	2019-09-05 14:20:00	ASSAULT	46.0	41.969078	-87.655608	14	September	Thursday	2019	9
313	JC424888	2019-09-05 08:00:00	DECEPTIVE PRACTICE	46.0	41.946459	-87.644655	8	September	Thursday	2019	9

Chicago Neighborhood and community Data

Chicago neighborhood and community data table is read as read_htm. This convert html table to data frame object. Here all the table are in a list. Then we get the list element in which our desired table contained.



Getting data about Chicago Neighborhood

```
import pandas as pd # import pandas for scrapping data of chicago Neiborhood and community name from html file
```

```
 pd.__version__
```

```
'0.25.1'
```

```
table = pd.read_html('https://en.wikipedia.org/wiki/List_of_neighborhoods_in_Chicago', index_col= False)
```

```
d =table[0]
```

```
d.columns =[ 'Neighborhood','Community'] # fix the index name with proper name
```

```
d.head()
```

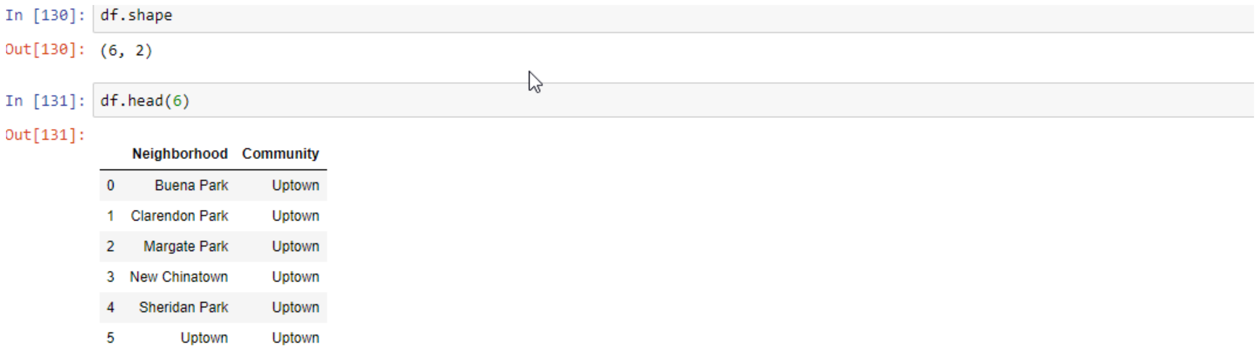
	Neighborhood	Community
0	Albany Park	Albany Park
1	Altgeld Gardens	Riverdale
2	Andersonville	Edgewater
3	Archer Heights	Archer Heights
4	Armour Square	Armour Square

Chicago neighborhood:

We are getting name of all the community of uptown by this code and replacing name Argyle Park with its new name New Chinatown

df.replace({'argyle park':'New Chinatown'},inplace = True)

df =d[d['Community'] =='Uptown'].reset_index(drop=True)

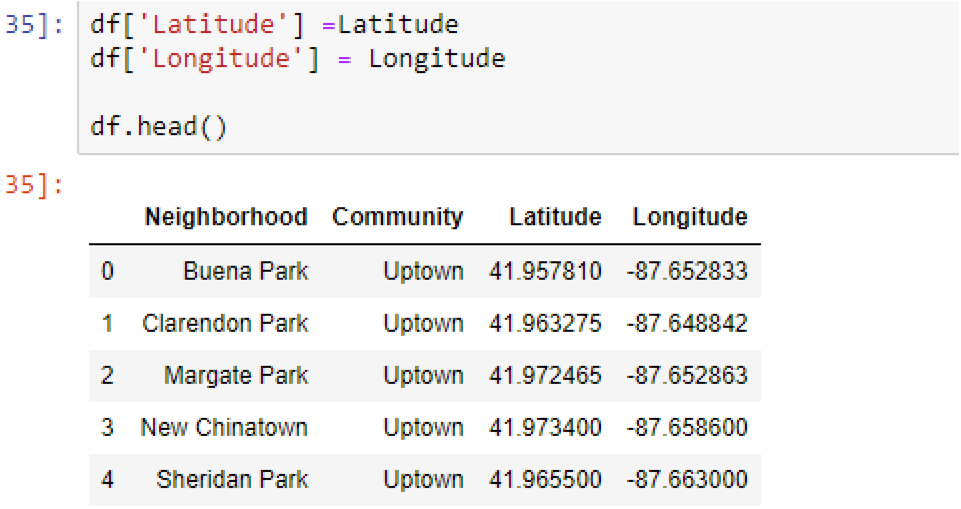


```
In [130]: df.shape
Out[130]: (6, 2)

In [131]: df.head(6)
Out[131]:
```

	Neighborhood	Community
0	Buena Park	Uptown
1	Clarendon Park	Uptown
2	Margate Park	Uptown
3	New Chinatown	Uptown
4	Sheridan Park	Uptown
5	Uptown	Uptown

Figure: Neighborhood name of uptown community

Longitude latitude for each neighborhood are added as separate column of data frame.



```
35]: df['Latitude'] =Latitude
     df['Longitude'] = Longitude

     df.head()
```

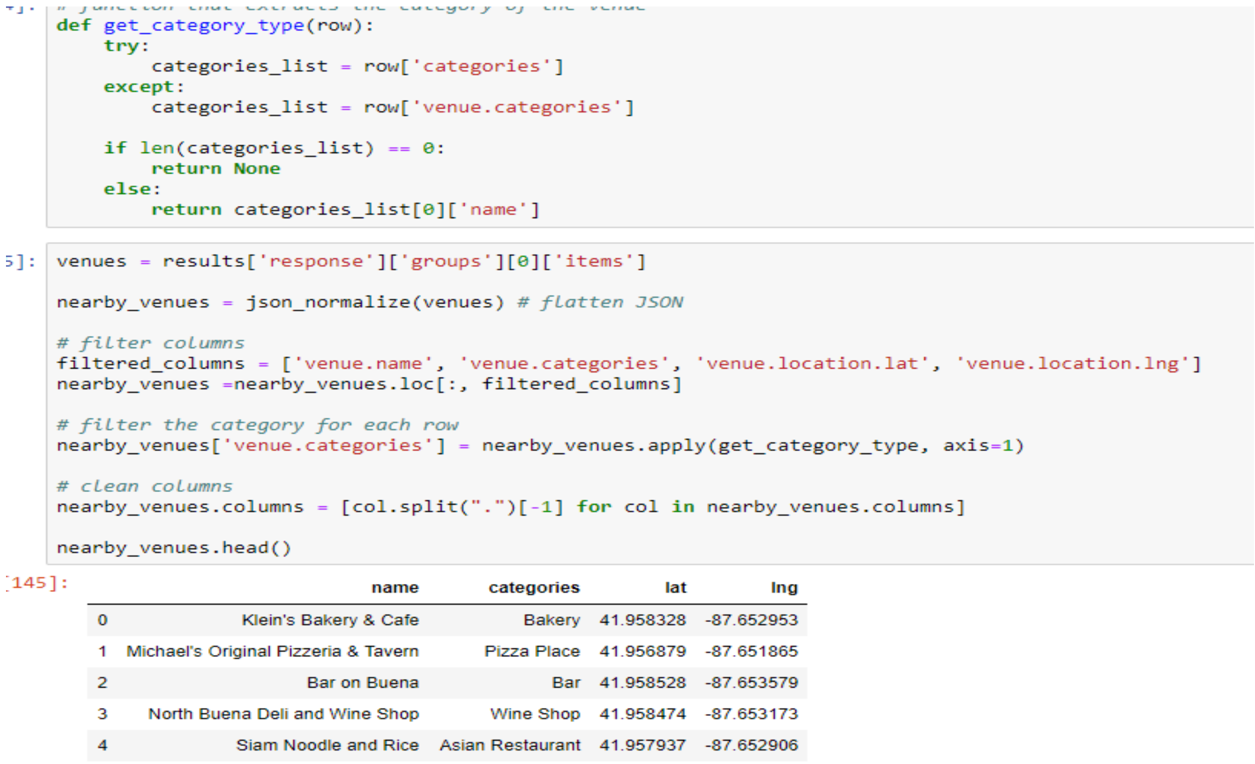
35]:

	Neighborhood	Community	Latitude	Longitude
0	Buena Park	Uptown	41.957810	-87.652833
1	Clarendon Park	Uptown	41.963275	-87.648842
2	Margate Park	Uptown	41.972465	-87.652863
3	New Chinatown	Uptown	41.973400	-87.658600
4	Sheridan Park	Uptown	41.965500	-87.663000

Data from foursquare

We construct the url with our CLIENT_id, CLIENT_secret and VERSION for all the 6 neighborhood longitude and latitude and get top 100 venue within 500 meter radius. Get the request of constructed url in .JSON file.

By the get_catagory_type function we get all venue in each neighborhood

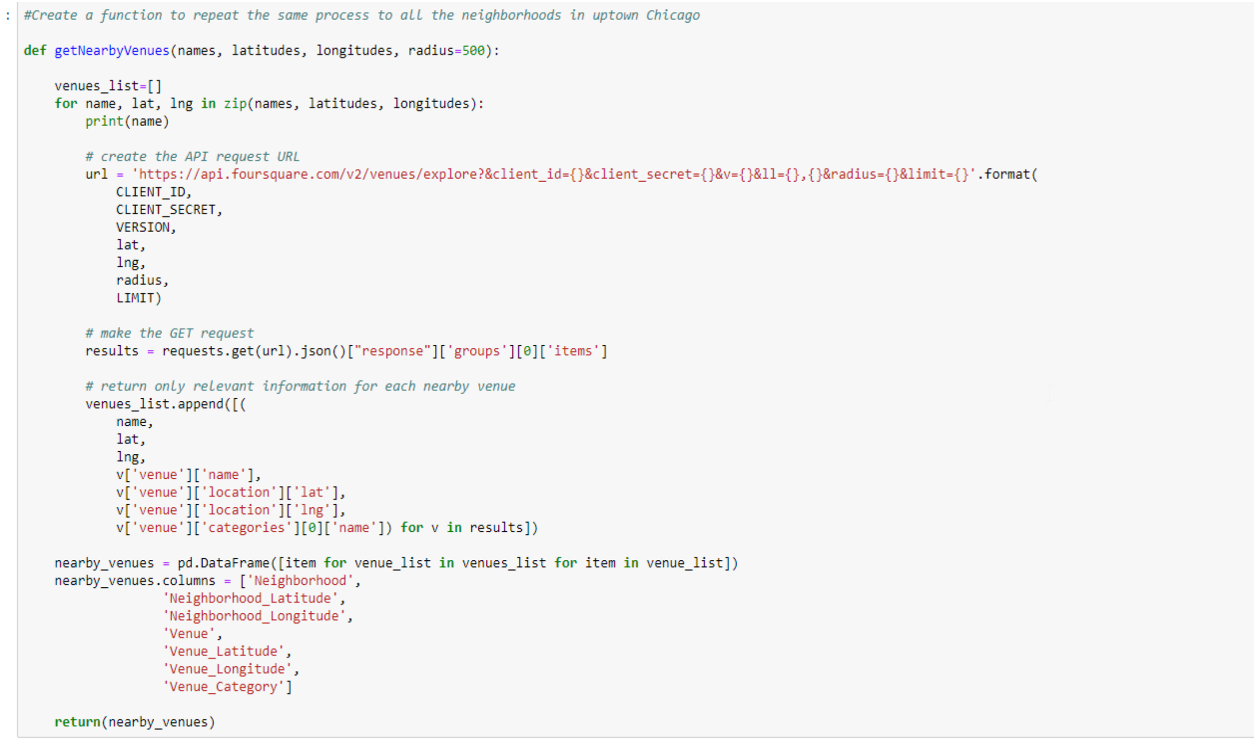


```
4]: # function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

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

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

nearby_venues = json_normalize(venues) # flatten JSON

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

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

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

nearby_venues.head()
```

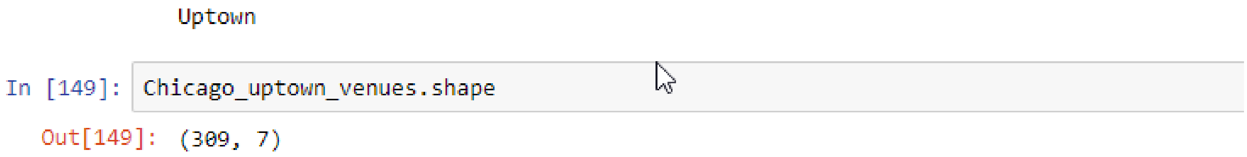
[145]:

	name	categories	lat	lng
0	Klein's Bakery & Cafe	Bakery	41.958328	-87.652953
1	Michael's Original Pizzeria & Tavern	Pizza Place	41.956879	-87.651865
2	Bar on Buena	Bar	41.958528	-87.653579
3	North Buena Deli and Wine Shop	Wine Shop	41.958474	-87.653173
4	Siam Noodle and Rice	Asian Restaurant	41.957937	-87.652906



```
#Create a function to repeat the same process to all the neighborhoods in uptown Chicago

def getNearbyVenues(names, latitudes, longitudes, radius=500):

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

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

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

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

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood_Latitude',
                  'Neighborhood_Longitude',
                  'Venue',
                  'Venue_Latitude',
                  'Venue_Longitude',
                  'Venue_Category']

    return(nearby_venues)
```

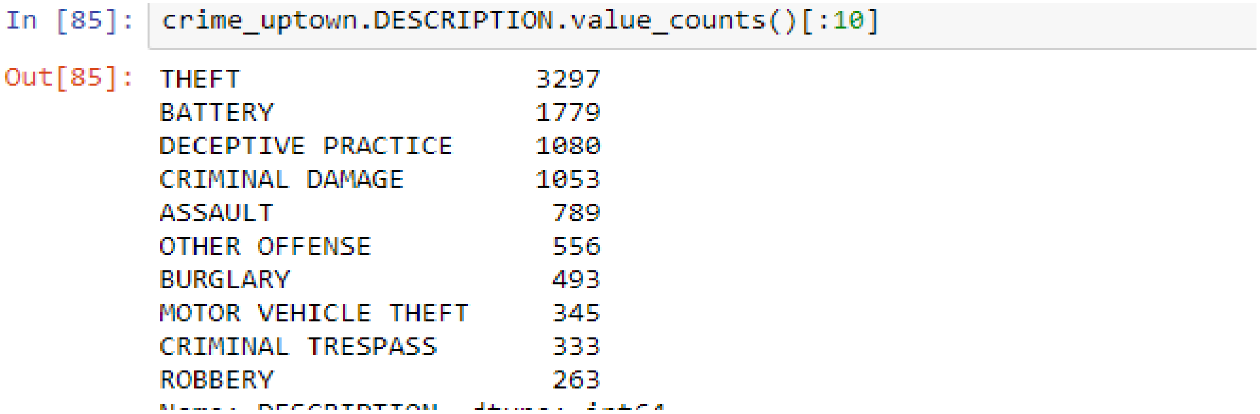
Our nearby_venues function returns 309 venue in all the neighborhood.



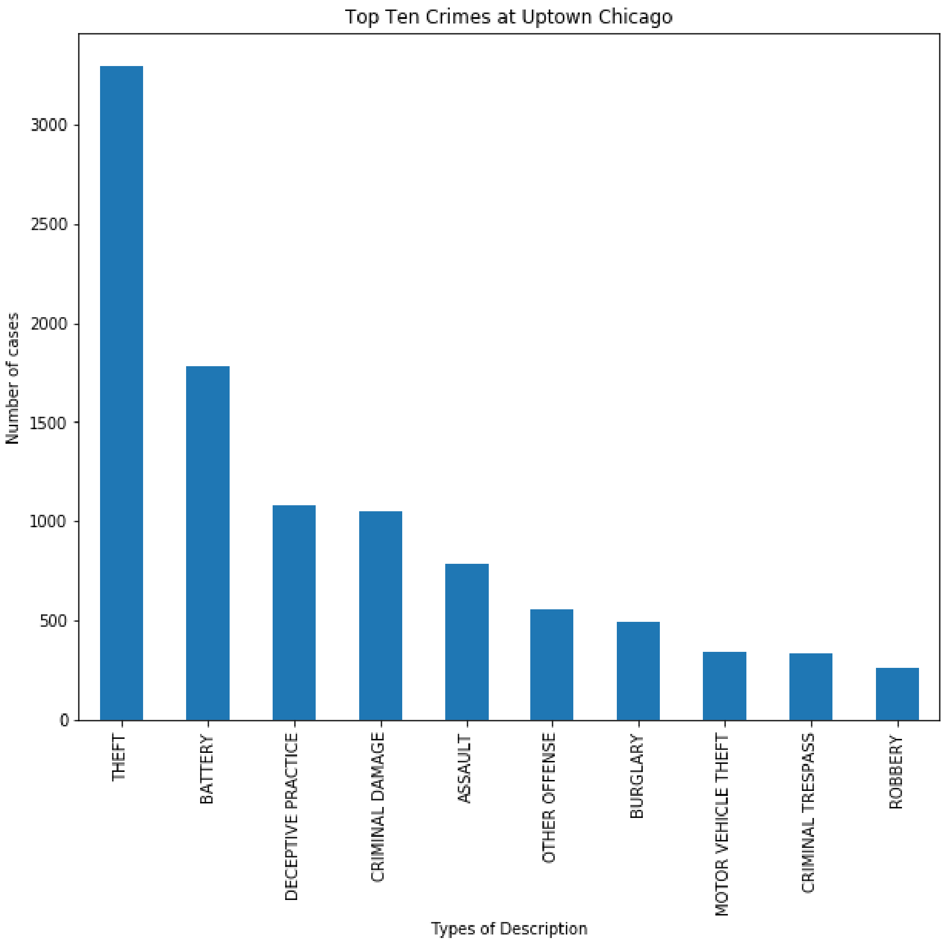
```
            Uptown

In [149]: Chicago_uptown_venues.shape

 Out[149]: (309, 7)
```

2.3 Data Analysis and Visualizations

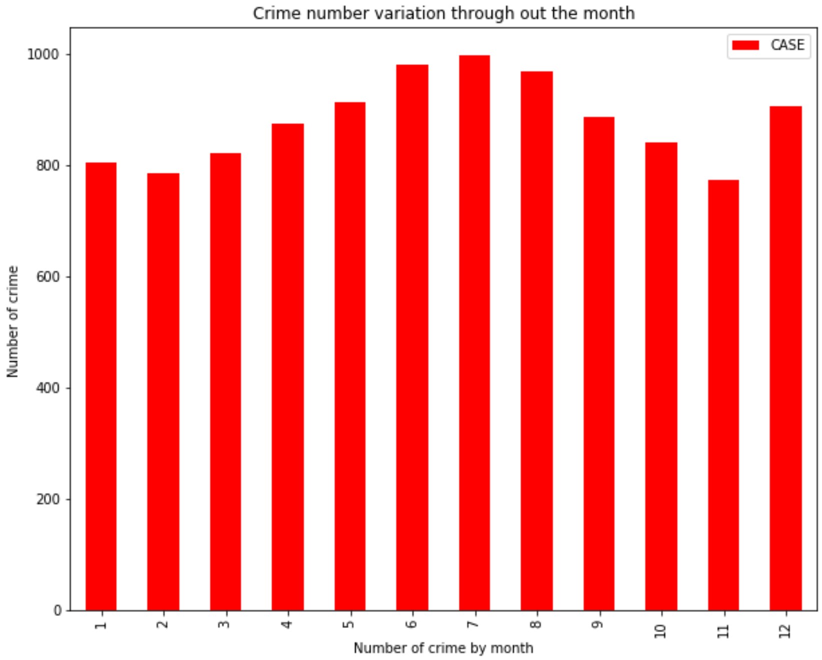
Now look at some statistics at uptown_chicago crime data



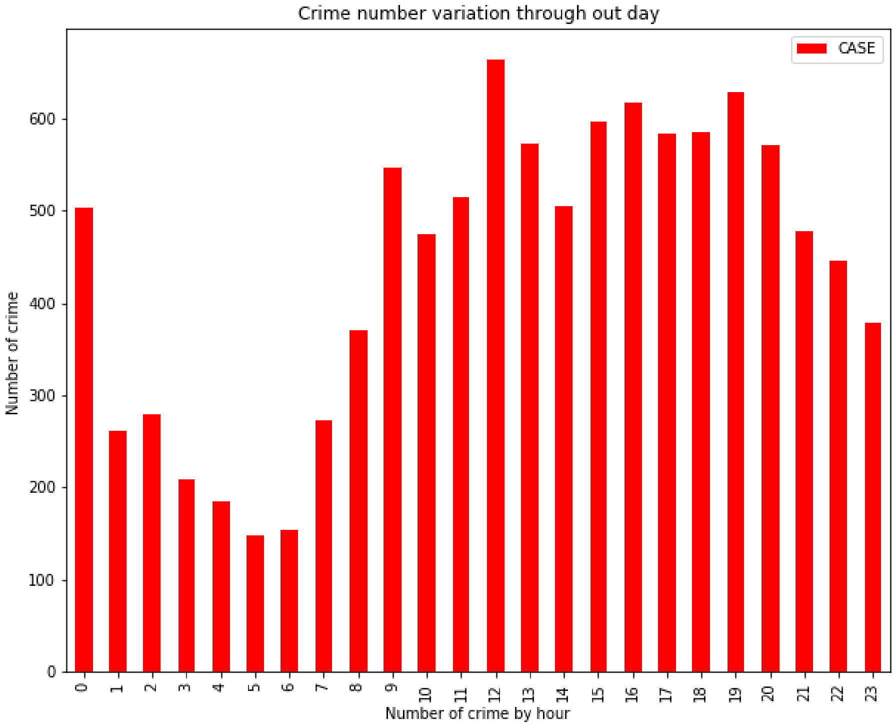
```
In [85]: crime_uptown.DESCRIPTION.value_counts()[:10]

Out[85]: THEFT                     3297
         BATTERY                   1779
         DECEPTIVE PRACTICE        1080
         CRIMINAL DAMAGE           1053
         ASSAULT                    789
         OTHER OFFENSE              556
         BURGLARY                   493
         MOTOR VEHICLE THEFT        345
         CRIMINAL TRESPASS          333
         ROBBERY                    263
         Name: DESCRIPTION, dtype: int64
```



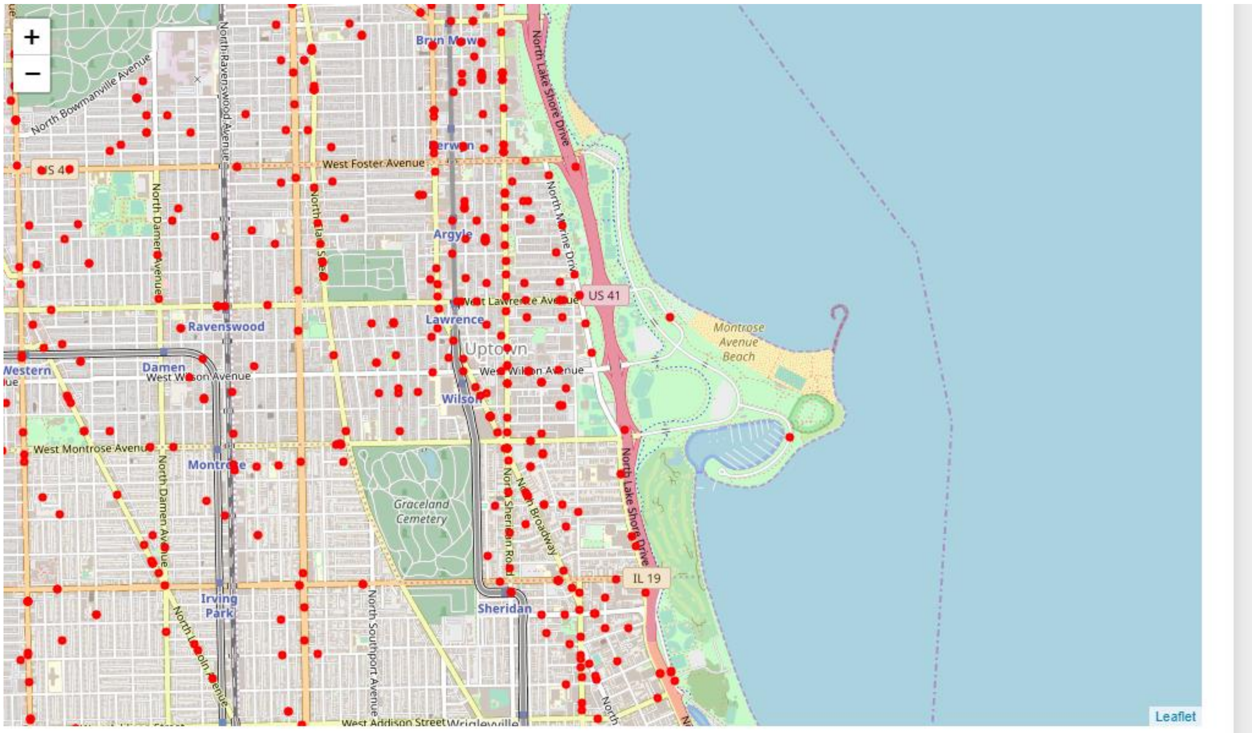
Looking at top 10 crime at uptown Chicago, it is obvious that theft frequency is almost twice and third than respective 2nd and 3rd most frequent crime.



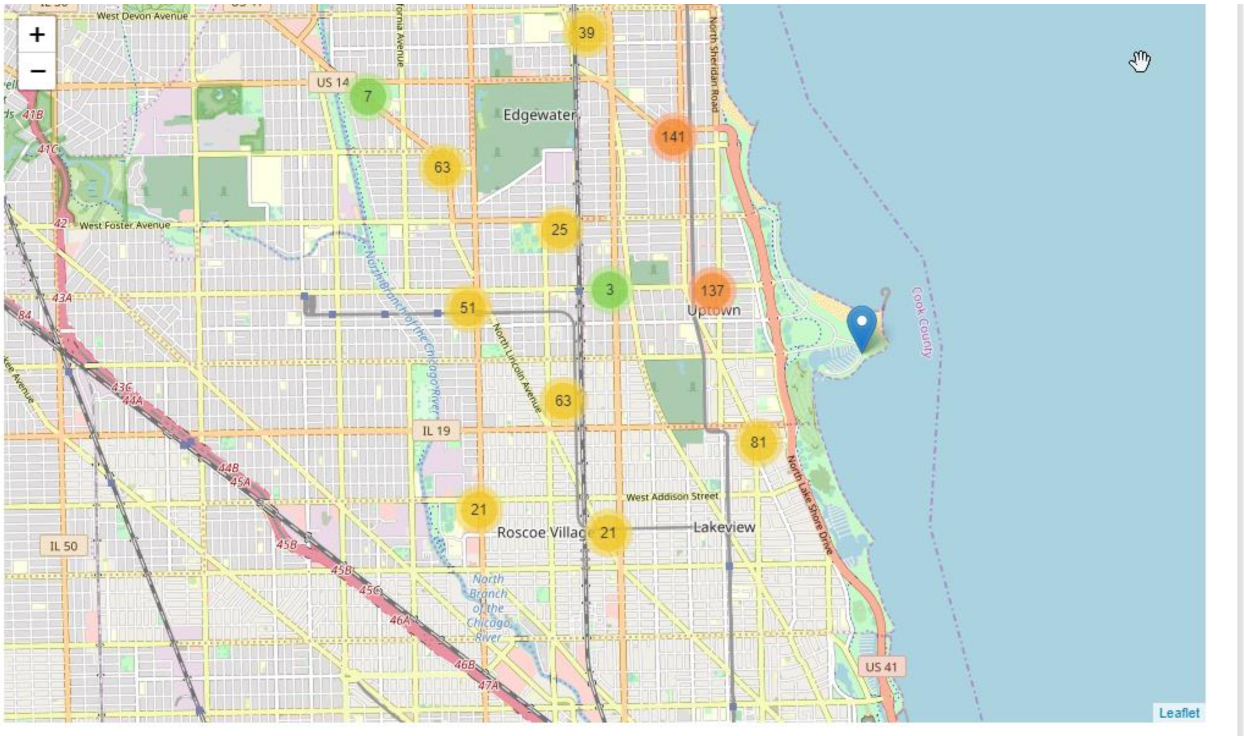
Number of increased in the middle of year especially in June, July, August it is pic.



Number of crime is at its peak during noon. It is higher from 3pm to 9pm.



Crime distributions for 2019 until November15 is superimpose on map created by folium. Then a clustered map is created. It shows center of uptown community has high crime rate.


+
−
West Devon Avenue
US 14
39
7
Edgewater
141
63
West Foster Avenue
25
3
137
Uptown
51
North Branch of the Chicago River
North Lincoln Avenue
Cook County
63
IL 19
81
North Sheridan Road
West Addison Street
21
Roscoe Village
21
Lakeview
IL 50
North Lake Shore Drive
North Branch of the Chicago River
US 41
Leaflet