MSDS420

Assignment 3: Chicago Crimes

Author : Atef Bader, PhD

Last Edit : 3/30/2019

Trump on Chicago: 'A disaster' 'Out of control,' 'Not good!' ... 'It's a great city'

10 times Trump has talked about Chicago violence. Quotes since 2016.

Deliverables:

- Submit a single zip-compressed file that has the name: YourLastName_Assignment_5 that has the following files:

    1. Your **PDF document** that has your Source code and output
    2. Your **ipynb script** that has your Source code and output

Objectives:

- Use SQL to execute different queries to retrieve data from Chicago Crime dataset and Police statins dataset
- Use Geospatial queries to locate **police stations** and **gun** related crimes (with arrest or no arrest) in every district on **Choropleth** map
- Use Geospatial queries to provide **descriptive stat** for every **district** on Choropleth map
- Use Geospatial queries to locate the **Block** that is the furthest (Maximum Distance) from the police station that has gun related crime resulted in arrest

Submission Formats :

Create a folder or directory with all supplementary files with your last name at the beginning of the folder name, compress that folder with zip compression, and post the zip-archived folder under the assignment link in Canvas. The following files should be included in an archive folder/directory that is uploaded as a single zip-compressed file. (Use zip, not StuffIt or any 7z or any other compression method.)

1. Complete IPYNB script that has the source code in Python used to access and analyze the data. The code should be submitted as an IPYNB script that can be be loaded and run in Jupyter Notebook for Python
2. Output from the program, such as console listing/logs, text files, and graphics output for visualizations. If you use the Data Science Computing Cluster or School of Professional Studies database servers or systems, include Linux logs of your sessions as plain text files. Linux logs may be generated by using the script process at the beginning of your session, as demonstrated in tutorial handouts for the DSCC servers.
3. List file names and descriptions of files in the zip-compressed folder/directory.

Formatting Python Code When programming in Python, refer to Kenneth Reitz' PEP 8: The Style Guide for Python Code: http://pep8.org/ (http://pep8.org/) (Links to an external site.)Links to an external site. There is the Google style guide for Python at https://google.github.io/styleguide/pyguide.html (https://google.github.io/styleguide/pyguide.html) (Links to an external site.)Links to an external site. Comment often and in detail.

Descriptions and Requirement Specifications

Chicago Crimes

In his first state of the uniion address , president Trump mentioned Chicago violance 10 times **Trump's State of the Union Address (http://www.chicagotribune.com/news/local/breaking/ct-trump-tweets-quotes-chicago-htmlstory.html)**

Chicago has more homicides than New York and Los Angeles combined

Columnist Clarence Page wrote an [article (http://www.chicagotribune.com/news/opinion/page/ct-perspec-page-trump-murder-rate-jeff-sessions-0103-20180102-story.html)](http://www.chicagotribune.com/news/opinion/page/ct-perspec-page-trump-murder-rate-jeff-sessions-0103-20180102-story.html) , published by the Chicago Tribune stated that the city of Chicago had **more homicides in the past two years than New York and Los Angeles combined**

Chicago Police Department

Chicago police department [CPD (https://home.chicagopolice.org/community/districts/11th-district-harrison/)](https://home.chicagopolice.org/community/districts/11th-district-harrison/) issues and publishes on daily basis on its website crime alerts, and press releases for the different [districts (https://home.chicagopolice.org/community/districts/)](https://home.chicagopolice.org/community/districts/) .

10 times Trump has talked about Chicago violence. Quotes since 2016.

The CPD categoizes the crimes into 8 categories as follows:

Trump on Chicago: 'A disaster' 'Out of control,' 'Not good!' ... 'It's a great city'

10 times Trump has talked about Chicago violence. Quotes since 2016.

Chicago Crimes Dataset

The CSV file for crimes dataset for the city of Chicago is obtained from the data portal for the city of Chicago. Here is the link for the city of Chicago data portal [City of Chicago Data Portal (https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2)](https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2)

Trump on Chicago: 'A disaster' 'Out of control,' 'Not good!' ... 'It's a great city'

10 times Trump has talked about Chicago violence. Quotes since 2016.

Loading the Dataset CSV file

Three datasets are need for this assignment:

1. The Chicago police stations in every district
2. The Boundaries.geojson data for district boundries
3. The Crimes dataset

Lets load the CSV file into a DataFrame object and see the nature of the data that we have.

Complete description of the dataset can be found on Chicago city data portal.

Based on Trumps State of the Uniion Address and the article written by columnist Clarence Page and published by the Chicago Tribune, we are interested to retrieve the data for the past two years and perform different types of spatial queries.

There are few of these queries that we are interested in to help CPD and city of Chicago to plot on a Choroplteh map those districts that have highest gun crimes.

Here are examples of those types of quereis:

1. Plot on **Choropleth map** the **districts** and their **Violent Crimes**
2. Plot on Choropleth map the districts and their **Gun** related crimes
3. Which district is the **crime capital** of **Chicago districts**?
4. What the **crime density** per **district**?
5. Plot on Choropleth map those **gun related crimes** that resulted in **arrests**
6. Plot on Choropleth map the gun related crime that is in the **farthest Block** from the **policy stattion** for every **district**

Packages you need to Connect **PostgreSQL** server to load and retrieve Crhicago Crime dataset from the database:

1. **psycopg2:** for PostgreSQL driver
2. **area:** to calculate the area inside of any GeoJSON geometry
3. **Folium:** for Choropleth maps

Execute the **pip install** command from the command window to install the packages listed bove

Trump on Chicago: 'A disaster' 'Out of control,' 'Not good!' ... 'It's a great city'

10 times Trump has talked about Chicago violence. Quotes since 2016.

Execute the following **pip show** command from the command line to get info about any package you installed. Make sure that the packages got installed into Anaconda library since we are using Anaconda tool; please see below the pakcage info

Trump on Chicago: 'A disaster' 'Out of control,' 'Not good!' ... 'It's a great city'

10 times Trump has talked about Chicago violence. Quotes since 2016.

Since we are using PostGIS in our work, please read and bookmark **Chapter 4. Using PostGIS: Data Management and Queries (https://postgis.net/docs/manual-1.4/ch04.html)**

In [1]:
```
import folium
from folium import plugins
from folium.plugins import MarkerCluster
import psycopg2
import csv
import pandas as pd
import json
from area import area

from psycopg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT
```

IMPORTANT NOTES:

- Use this version of the assignment if you don't want to install PostgreSQL server on your laptop/computer locally to experiment with datbase and tables creation
- In this version of the assignment you will be connecting to DSCC PostgreSQL server that has the database and tables already created on the server.
- You need to use your NetID and password for login and passward to connect PostgreSQL server hosted on DSCC
- For the psycopg2.connect statements listed below, you must provide your NetID and password in order to connect to PostgreSQL server hosted on DSCC

In [118]:
```
# To run the script on the complete dataset takes roughly 35 minutes to complete.
# Use this data set for your final submission of your Assignment 3
# Uncomment the following line after you unit test your code and ready to run and
submit your assignment on this dataset

db_connection = psycopg2.connect(                                    ame="chicago_crimes", u
ser="" , password="")
```

[redacted]

```
cursor = db_connection.cursor()
```

Chicago Crimes Dataset

The Crimes_2001_to_present.csv is downloaded from Chicago data portal and it has roughly 6.5 million records.

While working on this dataset, It is prudent to make a note of the following:

1. Geospatial queries are very demanding for system resouces like CPU, Memory, and DISK
2. We are interested in the data set of the past 2 years, and when you execute Geospatial type queries, please be advised that these queries slow down your machine.
3. Running this script to work on the data of the past 2 years will require roughly 25 minutes to complete. And requires roughly 40 minutes to complete using the dataset of the past 5 years. And requires hours to complete on the entire dataset with at least 16GB memory.
4. It is a good idea to take a slice (past two years) of the dataset and store it, that will help improve perfoamnce significantly especialy for SEARCH and SORT algorithms that are utilized by the database engine.

Algorithm Performance

- **Sort algorithms** used by the database engines vary in performance between O($NlogN$) and O(N^2) where N is the size of the number
- **Search algorithms** used by the database engines vary in performance between O($logN$) and O(N) where N is the size of the number

Lets start executing different Queries

Query #1:

- Calculate the total number of crimes in every district and plot that on Choropleth map

In [29]:
```
cursor.execute("SELECT district, count(district) from crimes GROUP BY district")
rows=cursor.fetchall()
```

In [30]:
```
crimes_per_district = pd.DataFrame(rows, columns=['dist_num','number_of_crimes'])
crimes_per_district['dist_num'] = crimes_per_district['dist_num'].astype(str)

crimes_per_district.head()
```

Out[30]:

	dist_num	number_of_crimes
0	24	412
1	11	1113
2	8	886
3	19	608
4	25	726

In [31]:
```
crimes_dataset = pd.DataFrame(rows)
```

In [32]:
```
crimes_dataset.head()
```

Out[32]:

	0	1
0	24	412
1	11	1113
2	8	886
3	19	608
4	25	726

In [33]:
```
total_number_of_crimes_per_district_map = folium.Map(location =(41.8781, -87.6298),
zoom_start=11)
```

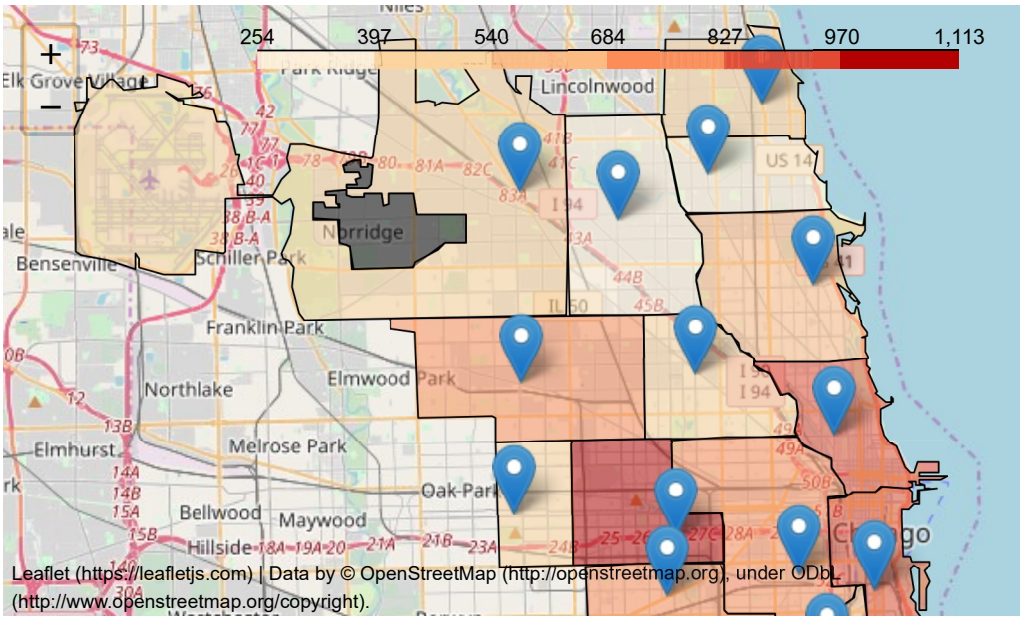
In [34]:
```
total_number_of_crimes_per_district_map.choropleth(geo_data="Boundaries.geojson",
              fill_color='OrRd',
              fill_opacity=0.5,
              line_opacity=1,
              data = crimes_per_district,
              key_on='feature.properties.dist_num',
              columns = ['dist_num', 'number_of_crimes']
              )
```

In [35]:
```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), dist
rict from police_stations where district!='Headquarters'""")
police_stations = cursor.fetchall()

for police_station in police_stations:
    police_station_location = (police_station[0],police_station[1])
    cursor.execute("SELECT district, count(district) from crimes where district= %s
GROUP BY district",[police_station[2]])
    districts_crime_numbers = cursor.fetchall()
    for district in districts_crime_numbers:
        folium.Marker(location = police_station_location,popup=folium.Popup(html="D
istrict No : %s  has   Total Number of Crimes:%s" %district ,max_width=450)).add_to
(total_number_of_crimes_per_district_map)
```

- **Lets plot the Choropleth map and notice the intensity of color on the different districts**
- **The Blue POPUP represents the location of police station in the different districts in the map**

In [36]:
```
total_number_of_crimes_per_district_map
```

Out[36]:



Query #2:

- Calculate the total number of **violent crimes** in every district and plot that in a table on Choropleth map

Well, we really need only the violent crimes per district, so we will filter only those crimes that we are interested in. Please note that we are not interested to plot property crimes, we are really after violent crimes and in particular **Gun** related crimes.

So for now, lets plot violent crimes on Choropleth map and later on we will filter only Gun related crimes

In [37]:
```
violent_crime_categories='THEFT','ASSAULT','ROBBERY','KIDNAPPING','CRIM SEXUAL ASSA
ULT','BATTERY','MURDER'
```

In [38]:
```
cursor.execute("SELECT district, count(district) from crimes where PRIMARY_TYPE in
%s GROUP BY district",[violent_crime_categories])
rows=cursor.fetchall()
violent_crime_data=pd.DataFrame(rows, columns=['district_num','number_of_violent_cr
imes'])
violent_crime_data['district_num'] = violent_crime_data['district_num'].astype(str)
violent_crime_data
```

Out[38]:

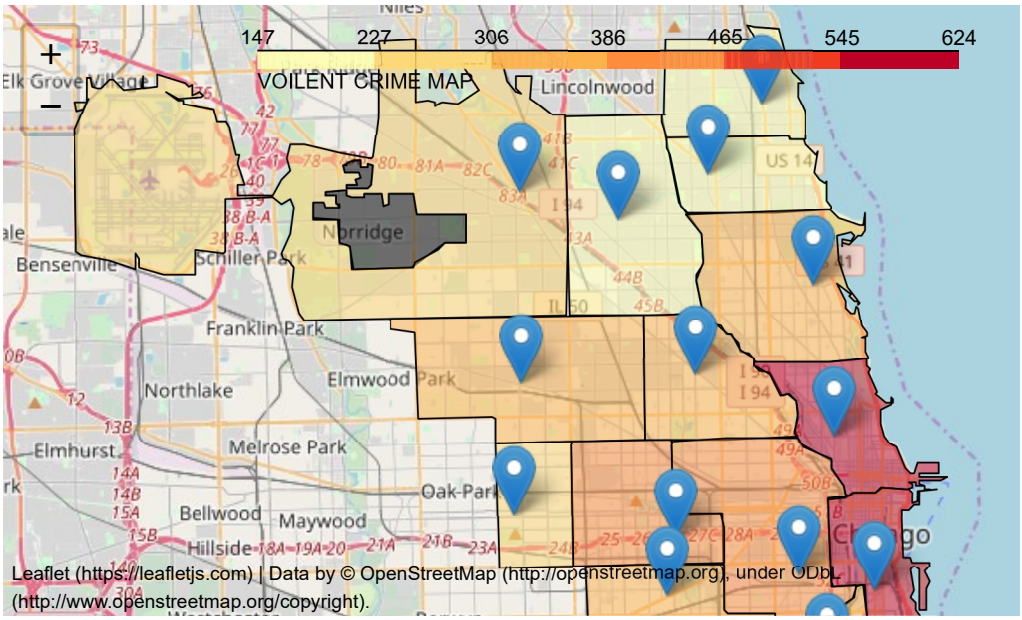
	district_num	number_of_violent_crimes
0	8	488
1	11	407
2	24	216
3	19	368
4	25	362
5	4	400
6	14	309
7	3	387
8	17	220
9	20	147
10	22	217
11	9	312
12	7	367
13	10	362
14	1	624
15	5	293
16	18	605
17	2	349
18	16	252
19	15	291
20	6	457
21	12	430

In [39]:
```
violent_crimes_per_district_map= folium.Map(location =(41.8781, -87.6298),zoom_star
t=11)
violent_crimes_per_district_map.choropleth(geo_data="Boundaries.geojson",
               fill_color='YlOrRd',
               fill_opacity=0.5,
               line_opacity=1,
               data = violent_crime_data,
               key_on='feature.properties.dist_num',
               columns = ['district_num', 'number_of_violent_crimes'],
               legend_name="VOILENT CRIME MAP"
               )
```

In [40]:
```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), dist
rict from police_stations where district!='Headquarters'""")
police_stations = cursor.fetchall()

for police_station in police_stations:
    police_station_location =(police_station[0],police_station[1])
    cursor.execute("SELECT PRIMARY_TYPE, count(PRIMARY_TYPE) from crimes where dist
rict =%s AND PRIMARY_TYPE in %s GROUP BY PRIMARY_TYPE",[police_station[2],violent_c
rime_categories])
    data = cursor.fetchall()
    violent_crimes_per_district_df = pd.DataFrame(data, columns=['Description', 'Nu
mber of Violent Crimes'])
    header = violent_crimes_per_district_df.to_html(classes='table table-striped ta
ble-hover table-condensed table-responsive')
    folium.Marker(location=police_station_location, popup=folium.Popup(html="Distri
ct Number %s - Violent Crimes %s" %(police_station[2],header))).add_to(violent_crim
es_per_district_map)
```

In [41]:
```
violent_crimes_per_district_map
```

Out[41]:



Query #3:

- Calculate the total number of **gun related violent crimes** in every district and plot that in a table on Choropleth map

Lets first create a dataframe of gun crimes per district first to get an idea about the number of gun crimes per district

In [42]:
```
gun='%GUN%'
cursor.execute("SELECT district, count(district) from crimes where DESCRIPTION::tex
t LIKE %s GROUP BY district",[gun])
districts_gun_violent_crimes = cursor.fetchall()
districts_gun_violent_crimes_df = pd.DataFrame(districts_gun_violent_crimes, column
s=['dist_num','gun_crimes'])
districts_gun_violent_crimes_df['dist_num'] = districts_gun_violent_crimes_df['dist
_num'].astype(str)
districts_gun_violent_crimes_df
```

Out[42]:

	dist_num	gun_crimes
0	24	9
1	8	49
2	11	69
3	19	11
4	25	43
5	4	50
6	14	12
7	3	57
8	17	7
9	20	5
10	22	24
11	10	58
12	9	39
13	7	67
14	1	6
15	5	44
16	18	14
17	2	34
18	16	12
19	15	34
20	6	63
21	12	38

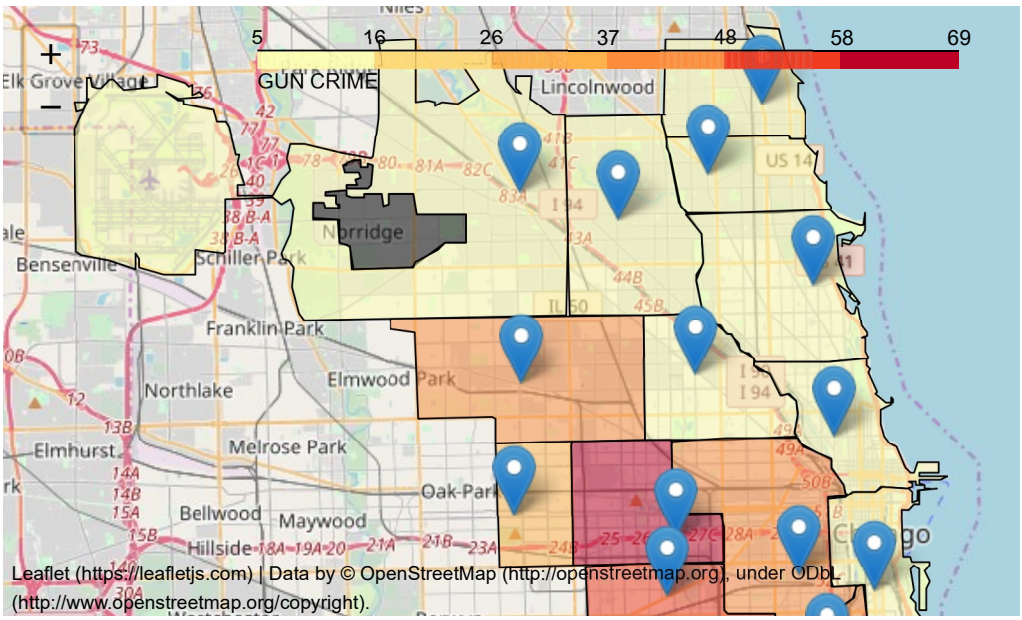
In [43]:
```
districts_gun_violent_crimes_map = folium.Map(location =(41.8781, -87.6298),zoom_st
art=11)
districts_gun_violent_crimes_map.choropleth(geo_data="Boundaries.geojson",
              fill_color='YlOrRd',
              fill_opacity=0.5,
              line_opacity=1,
              data = districts_gun_violent_crimes_df,
              key_on='feature.properties.dist_num',
              columns = ['dist_num', 'gun_crimes'],
              legend_name="GUN CRIME"
              )
```

Now, lets create a dataframe of the **different types of gun crimes for every district** and then plot it on Choropleth map

In [44]:
```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), dist
rict from police_stations where district!='Headquarters'""")

gun='%GUN%'
police_stations = cursor.fetchall()

for police_station in police_stations:
    police_station_location = (police_station[0],police_station[1])
    cursor.execute("""SELECT DESCRIPTION, count(DESCRIPTION) from crimes where dist
rict=%s and DESCRIPTION::text LIKE %s GROUP BY DESCRIPTION""",[police_station[2],gu
n])
    district_gun_violent_crimes=cursor.fetchall()
    district_gun_violent_crimes_df=pd.DataFrame(district_gun_violent_crimes, column
s=['Description', 'Number of Gun Crime'])
    header = district_gun_violent_crimes_df.to_html(classes='table table-striped ta
ble-hover table-condensed table-responsive')
    folium.Marker(location=police_station_location,popup=folium.Popup(html="Distric
t No: %s GUN_Crime: %s" %(police_station[2],header) )).add_to(districts_gun_violent
_crimes_map)
```

In [45]:
```
districts_gun_violent_crimes_map
```

Out[45]:



Query #4:

- Calculate the crime density per district

In [46]:
```
district=[]
tarea=[]

with open('Boundaries.geojson') as f:
    data = json.load(f)
    a = data['features']
    for i in range(len(a)):
        obj=a[i]['geometry']
        n= a[i]['properties']
        district.append(n['dist_num'])
        tarea.append(area(obj)/10000)

af=pd.DataFrame({'dist_num': district,'district_area_inHectares':tarea})
af['dist_num'] = af['dist_num'].astype(str)
final_data= pd.merge(af, crimes_per_district, on='dist_num', how='inner')
final_data['crime_density'] = round(final_data['number_of_crimes']/(final_data['dis
trict_area_inHectares']/100))
final_data
```

Out[46]:

	dist_num	district_area_inHectares	number_of_crimes	crime_density
0	17	2492.727155	386	15.0
1	20	1132.170216	254	22.0
2	19	2225.035732	608	27.0
3	25	2827.989237	726	26.0
4	14	1555.869965	508	33.0
5	7	1688.670732	745	44.0
6	3	1576.063931	695	44.0
7	4	7068.152865	790	11.0
8	6	2099.682124	844	40.0
9	22	3490.416073	442	13.0
10	5	3318.613379	635	19.0
11	24	1406.081387	412	29.0
12	16	8171.776367	473	6.0
13	8	5992.169760	886	15.0
14	18	1215.520046	869	71.0
15	12	2509.453028	700	28.0
16	11	1582.727274	1113	70.0
17	15	989.631393	530	54.0
18	10	2038.988883	779	38.0
19	1	1214.818895	854	70.0
20	9	3505.216898	585	17.0
21	2	1949.690970	602	31.0

Query #5:

- Create **Marker Clusters** on Choropleth map for those **gun related violent crimes** that resulted in **arrest (green icon)** and those that **didn't (red icon)**

In [47]:
```
gun_crime_arrests_map = folium.Map(location =(41.8781, -87.6298),zoom_start=11)
gun_crime_arrests_map.choropleth(geo_data="Boundaries.geojson",
              fill_color='YlOrRd',
              fill_opacity=0.5,
              line_opacity=1,
              data = districts_gun_violent_crimes_df,
              key_on='feature.properties.dist_num',
              columns = ['dist_num', 'gun_crimes'],
              legend_name="GUN CRIME"
              )
```

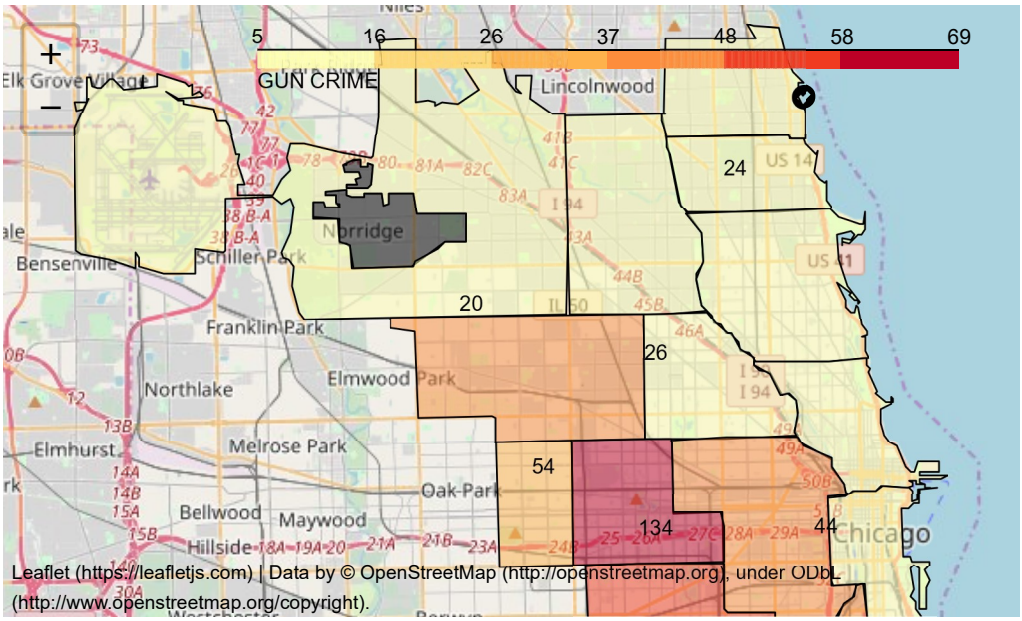
In [48]:
```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), dist
rict from police_stations where district!='Headquarters'""")
gun='%GUN%'

police_stations = cursor.fetchall()

marker_cluster = MarkerCluster().add_to(gun_crime_arrests_map)

for police_station in police_stations:
    police_station_location = (police_station[0],police_station[1])
    cursor.execute("""SELECT DISTINCT ON(caseno) caseno, block,DESCRIPTION, count(a
rrest), arrest,latitude, longitude from crimes where district=%s and DESCRIPTION::t
ext LIKE %s GROUP BY caseno,block, DESCRIPTION,arrest, latitude, longitude""",[poli
ce_station[2],gun])
    crimes_per_district = cursor.fetchall()
    for crime in crimes_per_district:
        if crime[4]==True:
            folium.Marker(location=(crime[5],crime[6]),popup=folium.Popup(html="Dis
trict No: %s <br> Description: %s <br> Block: %s" %(police_station[2],crime[2],crim
e[1])),icon=folium.Icon(color='green', icon='ok-sign'),).add_to(marker_cluster)
        else:
            folium.Marker(location=(crime[5],crime[6]),popup=folium.Popup(html="Dis
trict No: %s <br> Description: %s<br> Block: %s" %(police_station[2],crime[2],crime
[1])),icon=folium.Icon(color='red', icon='remove-sign'),).add_to(marker_cluster)
```

In [49]:
```
crimes_per_district[:3]
```

Out[49]:
```
[('JC100027',
  '017XX N MAYFIELD AVE',
  'UNLAWFUL POSS OF HANDGUN',
  1,
  True,
  41.91205157,
  -87.7728073),
 ('JC100191',
  '017XX N NEWCASTLE AVE',
  'UNLAWFUL POSS OF HANDGUN',
  1,
  True,
  41.91174566,
  -87.7960732),
 ('JC103615',
  '019XX N LA CROSSE AVE',
  'ARMED: HANDGUN',
  1,
  False,
  41.91638097,
  -87.74732207)]
```

In [50]:
```
gun_crime_arrests_map
```

Out[50]:



Query #6:

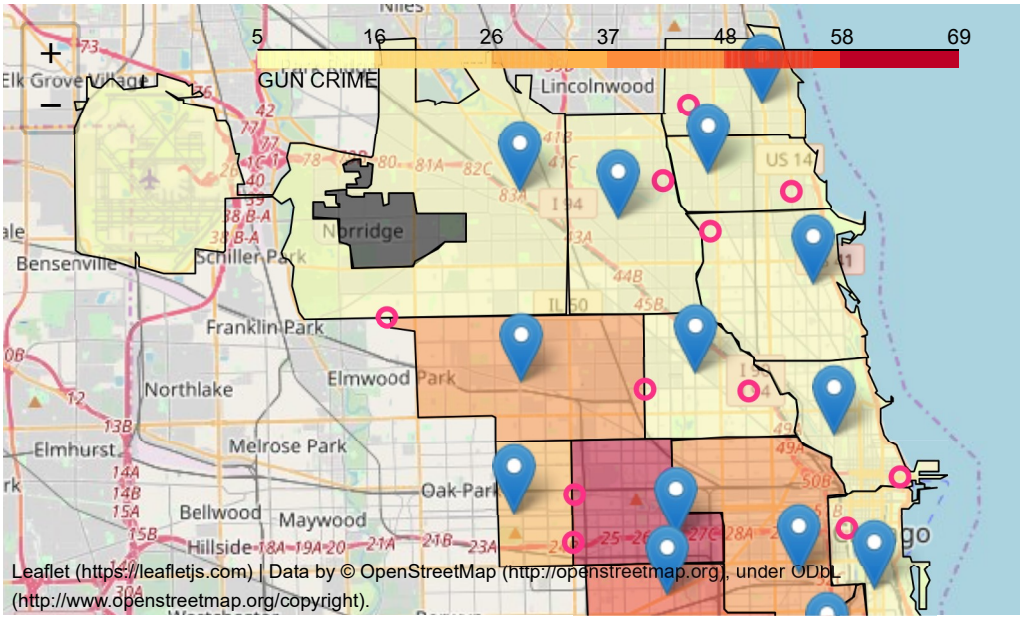
- Plot on Choropleth map the **farthest Block** that has a gun crime from every police station in every district

In [51]:
```
farthest_block_gun_crime_map = folium.Map(location =(41.8781, -87.6298),zoom_start=
11)
farthest_block_gun_crime_map.choropleth(geo_data="Boundaries.geojson",
              fill_color='YlOrRd',
              fill_opacity=0.5,
              line_opacity=1,
              data = districts_gun_violent_crimes_df,
              key_on='feature.properties.dist_num',
              columns = ['dist_num', 'gun_crimes'],
              legend_name="GUN CRIME"
              )
```

In [52]:
```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), dist
rict from police_stations where district!='Headquarters'""")
police_stations = cursor.fetchall()

gun='%GUN%'

for police_station in police_stations:

    cursor.execute("""SELECT DISTINCT on (A.block) A.district,A.block, A.where_is,S
T_Distance(A.where_is,B.where_is) from crimes as A, police_stations as B
    where ST_Distance(A.where_is,B.where_is) in ( SELECT max(dist) FROM
    (SELECT ST_Distance(A.where_is,B.where_is) as dist from crimes as A, police_sta
tions as B where A.district=%s
    and DESCRIPTION::text LIKE %s and B.district= %s ) as f)""",[police_station[2],
gun,police_station[2]])

    farthest_block_gun_crime = cursor.fetchall()

    cursor.execute("SELECT ST_X(ST_AsText(%s)), ST_Y(ST_AsText(%s))",(farthest_bloc
k_gun_crime[0][2],farthest_block_gun_crime[0][2]))
    farthest_block_gun_crime_location = cursor.fetchall()
    folium.Marker(location=(police_station[0],police_station[1]),popup=folium.Popup
(html="Police Station <br> District No.:%s <br> Farthest Gun_Crime Block:%s"%(farth
est_block_gun_crime[0][0],farthest_block_gun_crime[0][1]))).add_to(farthest_block_g
un_crime_map)
    folium.CircleMarker(farthest_block_gun_crime_location[0],radius=5,color='#ff318
7',popup=folium.Popup(html="District No.:%s <br> Block:%s"%(farthest_block_gun_crim
e[0][0],farthest_block_gun_crime[0][1]))).add_to(farthest_block_gun_crime_map)
```

In [53]:
```
farthest_block_gun_crime_map
```

Out[53]:



Requirements

The PDF document your are submitting must have the source code and the output for the following requirements

- ***Full data set used for requirements***
- ***Full data set not used for examples***

Requirement #1:

- Locate the **Block** that has the **higest number of gun crimes**. The popup on Choropleth map shall display the Block in every district along with the total number of gun crimes for that block

In [80]:
```
#Highest gun crimes per block
gun='%GUN%'
cursor.execute("SELECT district, block, gun_crimes FROM (SELECT district, block, gu
n_crimes, RANK() OVER (PARTITION BY district ORDER BY gun_crimes DESC) AS rn FROM(S
ELECT district, block, COUNT(block) AS gun_crimes FROM crimes WHERE DESCRIPTION::te
xt LIKE %s GROUP BY district, block) t) s WHERE s.rn = 1",[gun])
blockmax_gun_violent_crimes = cursor.fetchall()
blockmax_gun_violent_crimes_df = pd.DataFrame(blockmax_gun_violent_crimes, column
s=['dist_num', 'block_num','gun_crimes'])
blockmax_gun_violent_crimes_df['dist_num'] = blockmax_gun_violent_crimes_df['dist_n
um'].astype(str)
blockmax_gun_violent_crimes_df['block_num'] = blockmax_gun_violent_crimes_df['block
_num'].astype(str)
blockmax_gun_violent_crimes_df
```

Out[80]:

	dist_num	block_num	gun_crimes
0	1	0000X E ROOSEVELT RD	10
1	2	003XX E 47TH ST	15
2	3	064XX S DR MARTIN LUTHER KING JR DR	39
3	4	078XX S ESSEX AVE	17
4	5	102XX S STATE ST	13
5	5	130XX S EVANS AVE	13
6	6	083XX S COTTAGE GROVE AVE	23
7	7	066XX S HALSTED ST	21
8	8	063XX S ARTESIAN AVE	18
9	9	054XX S WINCHESTER AVE	13
10	9	045XX S WOOD ST	13
11	10	012XX S AVERS AVE	27
12	11	008XX N MONTICELLO AVE	19
13	12	023XX W JACKSON BLVD	11
14	12	022XX W MAYPOLE AVE	11
15	14	033XX W BEACH AVE	10
16	15	0000X S LEAMINGTON AVE	23
17	16	062XX W BELMONT AVE	5
18	17	037XX W MONTROSE AVE	7
19	17	043XX N KIMBALL AVE	7
20	18	013XX N HUDSON AVE	8
21	19	011XX W WILSON AVE	7
22	19	035XX N CLARK ST	7
23	20	050XX N WINTHROP AVE	6
24	22	103XX S HALSTED ST	10
25	24	076XX N BOSWORTH AVE	13
26	25	055XX W NORTH AVE	26

In [81]:
```
#Block with highest total out of all
bkmax = blockmax_gun_violent_crimes_df.sort_values(by=['gun_crimes'], ascending=False)
bkmax.iloc[0]
```

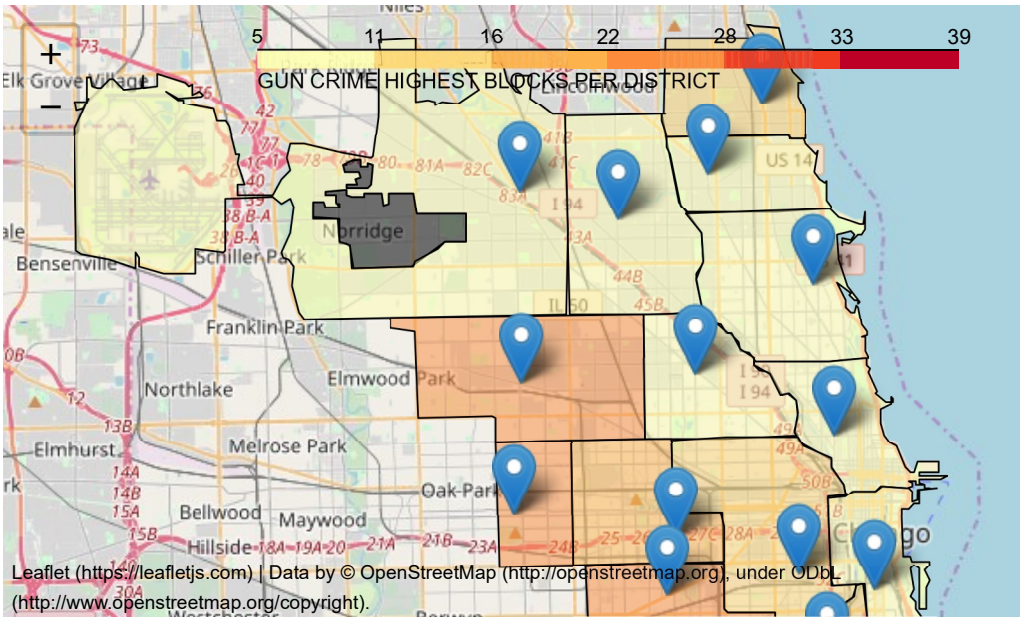
Out[81]:
```
dist_num                                        3
block_num     064XX S DR MARTIN LUTHER KING JR DR
gun_crimes                                     39
Name: 2, dtype: object
```

In [83]:
```
#Map of highest by block per district
blockmax_gun_violent_crimes_map = folium.Map(location =(41.8781, -87.6298),zoom_start=11)
blockmax_gun_violent_crimes_map.choropleth(geo_data="Boundaries.geojson",
              fill_color='YlOrRd',
              fill_opacity=0.5,
              line_opacity=1,
              data = blockmax_gun_violent_crimes_df,
              key_on= 'feature.properties.dist_num',
              columns = ['dist_num', 'gun_crimes'],
              legend_name="GUN CRIME HIGHEST BLOCKS PER DISTRICT"
              )
```

In [84]:
```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), district from police_stations where district!='Headquarters'""")

gun='%GUN%'
police_stations = cursor.fetchall()

for police_station in police_stations:
    police_station_location = (police_station[0],police_station[1])
    cursor.execute("""SELECT district, block, gun_crimes FROM (SELECT district, block, gun_crimes, RANK() OVER (PARTITION BY district ORDER BY gun_crimes DESC) AS rn FROM(SELECT district, block, COUNT(block) AS gun_crimes FROM crimes WHERE district=%s AND DESCRIPTION::text LIKE %s GROUP BY district, block) t) s WHERE s.rn = 1""",[police_station[2],gun])
    blockmax_gun_violent_crimes=cursor.fetchall()
    blockmax_gun_violent_crimes_df=pd.DataFrame(blockmax_gun_violent_crimes, columns=['District No.', 'Block', 'Number of Gun Crimes'])
    header = blockmax_gun_violent_crimes_df.to_html(classes='table table-striped table-hover table-condensed table-responsive')
    folium.Marker(location=police_station_location,popup=folium.Popup(html="District No: %s GUN_Crime: %s" %(police_station[2],header) )).add_to(blockmax_gun_violent_crimes_map)
```

In [85]:
```
blockmax_gun_violent_crimes_map
```

Out[85]:



Requirement #2:

- Calculate the gun crimes density in every district

In [86]:
```
gun='%GUN%'
cursor.execute("SELECT district, count(district) from crimes where DESCRIPTION::tex
t LIKE %s GROUP BY district",[gun])
districts_gun_violent_crimes = cursor.fetchall()
districts_gun_violent_crimes_df = pd.DataFrame(districts_gun_violent_crimes, column
s=['dist_num','gun_crimes'])
districts_gun_violent_crimes_df['dist_num'] = districts_gun_violent_crimes_df['dist
_num'].astype(str)
districts_gun_violent_crimes_df['gun_crimes'].sum()
```

Out[86]:
```
31473
```

In [87]:
```
district=[]
tarea=[]

with open('Boundaries.geojson') as f:
    data = json.load(f)
    a = data['features']
    for i in range(len(a)):
        obj=a[i]['geometry']
        n= a[i]['properties']
        district.append(n['dist_num'])
        tarea.append(area(obj)/10000)

den=pd.DataFrame({'dist_num': district,'district_area_inHectares':tarea})
den['dist_num'] = den['dist_num'].astype(str)
final_den= pd.merge(af, districts_gun_violent_crimes_df, on='dist_num', how='inner
')
final_den['crime_density'] = round(final_den['gun_crimes']/(final_den['district_are
a_inHectares']/100))
final_den
```

Out[87]:

	dist_num	district_area_inHectares	gun_crimes	crime_density
0	17	2492.727155	574	23.0
1	20	1132.170216	235	21.0
2	19	2225.035732	501	23.0
3	25	2827.989237	1738	61.0
4	14	1555.869965	822	53.0
5	7	1688.670732	2739	162.0
6	3	1576.063931	1928	122.0
7	4	7068.152865	1960	28.0
8	6	2099.682124	2598	124.0
9	22	3490.416073	1059	30.0
10	5	3318.613379	1925	58.0
11	24	1406.081387	540	38.0
12	16	8171.776367	326	4.0
13	8	5992.169760	1853	31.0
14	18	1215.520046	411	34.0
15	12	2509.453028	1334	53.0
16	11	1582.727274	3175	201.0
17	15	989.631393	2015	204.0
18	10	2038.988883	2188	107.0
19	1	1214.818895	410	34.0
20	9	3505.216898	1794	51.0
21	2	1949.690970	1348	69.0

Requirement #3:

- Locate the **farthest** UNLAWFUL POSS OF HANDGUN crime from the police station in every district. The popup on Choropleth map shall display the district number and the block

NOTE:
I initially had issues with the code provided as the search variable gun = 'UNLAWFUL POSS OF HANDGUN' would not work. I tried many alternatives which gave me an error and the only query that would work was if I set gun to '%POSS%, which is incorrect. I attempted many hours to resolve this and the only alternative that worked was pulling all the data and putting it into dataframe objects and finding the furthest by utilizing groupby and join. I then mapped the data.

However, did get the first method to work using the Geospatial query which is in the ***following*** first set of code and map.

Since I spent a lot of time on pulling the data and reworking it into dataframe objects I have included that work as well after the "Correct" map. The second map I believe is slightly inaccurate.

The second set/map is very close to the first map, though there are differences. Therefore, please take the ***first map*** as the ***correct*** one. In addition, I did run some queries based of the search words 'UNLAWFUL POSS OF HANDGUN' and other descriptions were included in the queries outside of the search words so map 1 might also be suspect.

However, ***MAP 1*** uses the correct geospatial query so I would take that as my answer, yet please also consider the ***second*** one.

Correct MAP below

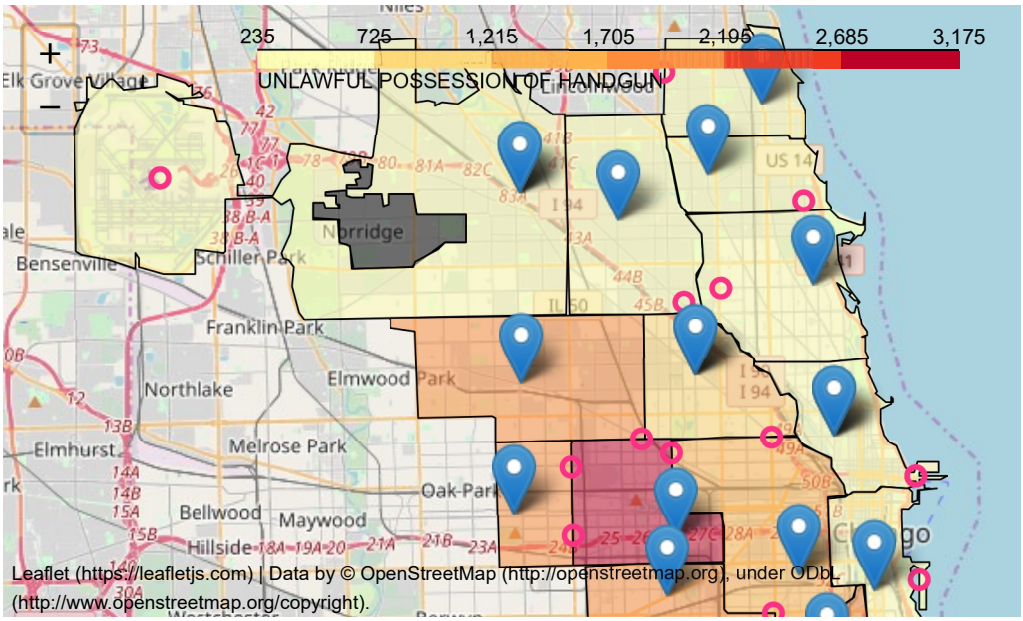
In [124]:
```
#Map using 'UNLAWFUL POSS OF HANGUN' as search variable
farthest_unlaw_gun_crime_map = folium.Map(location =(41.8781, -87.6298),zoom_star
t=11)
farthest_unlaw_gun_crime_map.choropleth(geo_data="Boundaries.geojson",
              fill_color='YlOrRd',
              fill_opacity=0.5,
              line_opacity=1,
              data = districts_gun_violent_crimes_df,
              key_on='feature.properties.dist_num',
              columns = ['dist_num', 'gun_crimes'],
              legend_name="UNLAWFUL POSSESSION OF HANDGUN"
              )
```

In [121]:
```
gun='UNLAWFUL POSS OF HANDGUN'
```

In [122]:
```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), dis
trict from police_stations where district!='Headquarters'""")
police_stations = cursor.fetchall()

for police_station in police_stations:

    cursor.execute("""SELECT DISTINCT on (A.block) A.district,A.block, A.where_is,
ST_Distance(A.where_is,B.where_is) from crimes as A, police_stations as B
    where ST_Distance(A.where_is,B.where_is) in ( SELECT max(dist) FROM
    (SELECT ST_Distance(A.where_is,B.where_is) as dist from crimes as A, police_st
ations as B where A.district=%s
    and DESCRIPTION::text like %s and B.district= %s ) as f)""",[police_station
[2],gun,police_station[2]])

    farthest_unlaw_gun_crime = cursor.fetchall()

    for i in farthest_unlaw_gun_crime:
        if cursor.execute("SELECT ST_X(ST_AsText(%s)), ST_Y(ST_AsText(%s))",(farth
est_unlaw_gun_crime[0][2],farthest_unlaw_gun_crime[0][2])) == []:
            continue
        elif cursor.execute("SELECT ST_X(ST_AsText(%s)), ST_Y(ST_AsText(%s))",(far
thest_unlaw_gun_crime[0][2],farthest_unlaw_gun_crime[0][2])) != []:
            cursor.execute("SELECT ST_X(ST_AsText(%s)), ST_Y(ST_AsText(%s))",(fart
hest_unlaw_gun_crime[0][2],farthest_unlaw_gun_crime[0][2]))

        farthest_unlaw_gun_crime_location = cursor.fetchall()
        folium.Marker(location=(police_station[0],police_station[1]),popup=folium.
Popup(html="Police Station <br> District No.:%s <br> Farthest Unlawful Poss of Han
dgun Block:%s"%(farthest_unlaw_gun_crime[0][0],farthest_unlaw_gun_crime[0][1]))).a
dd_to(farthest_unlaw_gun_crime_map)
        folium.CircleMarker(farthest_unlaw_gun_crime_location[0],radius=5,color='#
ff3187',popup=folium.Popup(html="District No.:%s <br> Block:%s"%(farthest_unlaw_gu
n_crime[0][0],farthest_unlaw_gun_crime[0][1]))).add_to(farthest_unlaw_gun_crime_ma
p)
```

In [123]:
```
farthest_unlaw_gun_crime_map
```

Out[123]:



Second MAP (inaccurate)

In [36]:
```
# Map built off dataframe finding farthest with groupby and joining data. All data
was pulled as DISTINCT was inaccurate.
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), dist
rict from police_stations where district!='Headquarters'""")
police_stations = cursor.fetchall()

for police_station in police_stations:

    cursor.execute("""SELECT A.district, A.block, A.description, A.latitude, A.long
itude, A.where_is,ST_Distance(A.where_is,B.where_is) from crimes as A, police_stati
ons as B
    where ST_Distance(A.where_is,B.where_is) in (SELECT (dist) FROM
    (SELECT ST_Distance(A.where_is,B.where_is) as dist from crimes as A, police_sta
tions as B where DESCRIPTION::text LIKE '%POSS%' and
    iucr = '143A' order by dist DESC) as f)""")

    farthest_unlaw_gun_poss_df = cursor.fetchall()
```

In [53]:
```
#Pulled data was put into dataframe objects where furthest distances are located by
block for each district. Query did not
#fully filter by 'UNLAWFUL POSSESSION OF HANDGUN', therefore they are filtered out
in the data frame created as well.
#A tuple is then created for mapping.

df1 = pd.DataFrame(farthest_unlaw_gun_poss_df, columns = ['dist_num','block','descr
iption', 'latitude', 'longitude', 'location','distance'])
df2 = df1.loc[df1['description']=='UNLAWFUL POSS OF HANDGUN']
df3 = pd.DataFrame(df2)
df4 = df3['distance'].groupby([df3['dist_num']]).max().reset_index()
df5 = pd.merge(df3, df4, how='right')
df6 = df5.sort_values(by=['dist_num'])
poss_df7 = df6.drop_duplicates()
poss_unlaw = list(poss_df7.itertuples(index=False, name=None))
poss_df7
```

Out[53]:

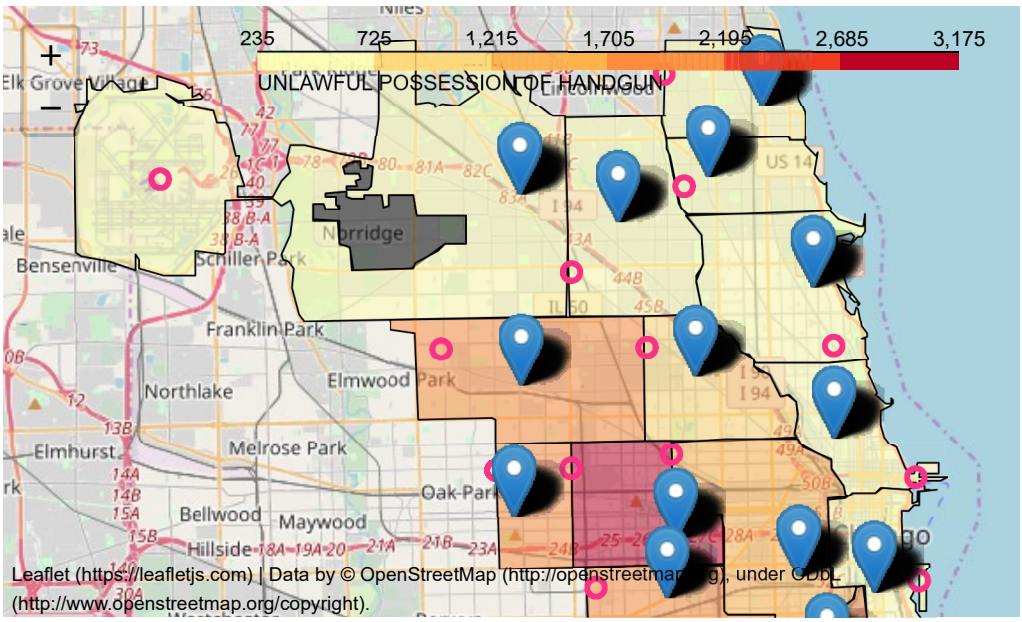
	dist_num	block	description	latitude	longitude	
22	1	014XX S LYNN WHITE DR	UNLAWFUL POSS OF HANDGUN	41.862044	-87.609570	0101000020E6100000DF8F827757EE4440EA785
43	2	055XX S LAKE SHORE DR SB	UNLAWFUL POSS OF HANDGUN	41.794348	-87.579638	0101000020E6100000813ACC34ADE54440E8B58
16	3	027XX E 75TH ST	UNLAWFUL POSS OF HANDGUN	41.760183	-87.557984	0101000020E610000058F8C0AF4DE144404626E
36	4	095XX S EDWARD BARRON DR	UNLAWFUL POSS OF HANDGUN	41.722695	-87.525947	0101000020E610000070D3464581DC4440F42C3
38	5	014XX E 103RD ST	UNLAWFUL POSS OF HANDGUN	41.707736	-87.588543	0101000020E6100000425A081797DA44408EE878
12	6	079XX S GREENWOOD AVE	UNLAWFUL POSS OF HANDGUN	41.750547	-87.597895	0101000020E6100000D963C9EC11E0444071541
34	7	071XX S LAFAYETTE AVE	UNLAWFUL POSS OF HANDGUN	41.764311	-87.626523	0101000020E6100000814E4DF1D4E14440437AD
27	8	069XX W ARCHER AVE	UNLAWFUL POSS OF HANDGUN	41.792104	-87.794323	0101000020E6100000D0ED67A963E5444068C7C
25	9	035XX W 38TH PL	UNLAWFUL POSS OF HANDGUN	41.823382	-87.712572	0101000020E61000006A7CE79664E9444013187E
0	10	015XX S KENNETH AVE	UNLAWFUL POSS OF HANDGUN	41.859970	-87.735995	0101000020E61000007F913A7B13EE4440B2418
28	11	007XX N CICERO AVE	UNLAWFUL POSS OF HANDGUN	41.894482	-87.745789	0101000020E61000007532CE657EF244402C6F6
32	12	009XX N KEDZIE AVE	UNLAWFUL POSS OF HANDGUN	41.898680	-87.706657	0101000020E610000076CD02F207F3444005CB4E
9	14	035XX W SCHUBERT AVE	UNLAWFUL POSS OF HANDGUN	41.930092	-87.716492	0101000020E6100000B2208D3D0DF7444039A47E
19	15	007XX N AUSTIN BLVD	UNLAWFUL POSS OF HANDGUN	41.894126	-87.775109	0101000020E6100000661417BC72F24440101DE
15	16	0000X W TERMINAL ST	UNLAWFUL POSS OF HANDGUN	41.979006	-87.906463	0101000020E6100000B6180E1450FD444041700
2	17	047XX W BYRON ST	UNLAWFUL POSS OF HANDGUN	41.951586	-87.746048	0101000020E61000005F1D9A8ECDF94440AB8F5
4	18	006XX E GRAND AVE	UNLAWFUL POSS OF HANDGUN	41.891990	-87.611462	0101000020E610000031D0ABBD2CF2444099766
21	19	026XX N CLARK ST	UNLAWFUL POSS OF HANDGUN	41.930391	-87.643714	0101000020E6100000E6A97E0D17F74440F6CAB
30	20	029XX W FARRAGUT AVE	UNLAWFUL POSS OF HANDGUN	41.976714	-87.701983	0101000020E6100000E37177F404FD4440B1C5FC
		040XX W	UNLAWFUL			

In [55]:
```
farthest_unlaw_poss_map = folium.Map(location =(41.8781, -87.6298),zoom_start=11)
farthest_unlaw_poss_map.choropleth(geo_data="Boundaries.geojson",
                fill_color='YlOrRd',
                fill_opacity=0.5,
                line_opacity=1,
                data = districts_gun_violent_crimes_df,
                key_on='feature.properties.dist_num',
                columns = ['dist_num', 'gun_crimes'],
                legend_name="UNLAWFUL POSSESSION OF HANDGUN"
                )
```

In [58]:
```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), dist
rict from police_stations where district!='Headquarters'""")
police_stations = cursor.fetchall()

count=0
for unlaw in poss_unlaw:
    for police_station in police_stations:

        folium.Marker(location=(police_station[0],police_station[1]),popup=folium.P
opup(html="Police Station <br> District No.:%s"%(police_station[2]))).add_to(farthe
st_unlaw_poss_map)
        folium.CircleMarker(location = [poss_unlaw[count][3], poss_unlaw[coun
t][4]],radius=5,color='#ff3187',popup=('District No:',poss_unlaw[count][0],'Block N
o:',poss_unlaw[count][1])).add_to(farthest_unlaw_poss_map)

    count = count + 1
```

In [59]:
```
farthest_unlaw_poss_map
```

Out[59]:



Requirement #4:

- Create **Marker Clusters** on Choropleth map for those **gun related violent crimes** that have Location Desciption as RESIDENCE in **(green icon)** and those that have Location Desciption as STREET in **(red icon)**

In [77]:
```
gun_crime_rslocation_map = folium.Map(location =(41.8781, -87.6298),zoom_start=11)
gun_crime_rslocation_map.choropleth(geo_data="Boundaries.geojson",
              fill_color='YlOrRd',
              fill_opacity=0.5,
              line_opacity=1,
              data = districts_gun_violent_crimes_df,
              key_on='feature.properties.dist_num',
              columns = ['dist_num', 'gun_crimes'],
              legend_name="GUN CRIME RESIDENCE OR STREET"
              )
```

In [78]:
```
cursor.execute("""SELECT DISTINCT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_I
S)), district from police_stations where district!='Headquarters'""")
gun='%GUN%'

police_stations = cursor.fetchall()

marker_cluster = MarkerCluster().add_to(gun_crime_rslocation_map)

for police_station in police_stations:
    police_station_location = (police_station[0],police_station[1])
    cursor.execute("""SELECT DISTINCT ON(caseno) caseno, block,DESCRIPTION, count(b
lock), location_description, latitude, longitude from crimes where district=%s and
DESCRIPTION::text LIKE %s GROUP BY caseno,block, DESCRIPTION, location_description,
latitude, longitude""",[police_station[2],gun])
    crimes_per_district = cursor.fetchall()
    for crime in crimes_per_district:
        if crime[4]=="RESIDENCE":
            folium.Marker(location=(crime[5],crime[6]),popup=folium.Popup(html="Dis
trict No: %s <br> Location Description: %s <br> Description: %s <br> Block: %s" %(p
olice_station[2],crime[4],crime[2],crime[1])),icon=folium.Icon(color='green', ico
n='ok-sign'),).add_to(marker_cluster)
        elif crime[4]=="STREET":
            folium.Marker(location=(crime[5],crime[6]),popup=folium.Popup(html="Dis
trict No: %s <br> Location Description: %s <br> Description: %s<br> Block: %s" %(po
lice_station[2],crime[4],crime[2],crime[1])),icon=folium.Icon(color='red', icon='re
move-sign'),).add_to(marker_cluster)
```

In [79]:
```
gun_crime_rslocation_map
```

Out[79]: