07/22/19

Deliverables:

- Submit the following two files:

  1. Your **html 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 Station 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 :

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. Make sure to include your name as part of the IPYNB file name.
3. From the File menu select Download As -> HTML (.html) to create an HTML of the IPYNB file.

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)** , 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/) issues and publishes on daily basis on its website crime alerts, and press releases for the different districts (https://home.chicagopolice.org/community/districts/) .

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)

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 boundaries
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 Union 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 Choropleteh map those districts that have highest gun crimes.

Here are examples of those types of queries:

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 station** 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

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]:
```
!pip install psycopg2
```

```
Requirement already satisfied: psycopg2 in /Library/Frameworks/Python.framework/
Versions/3.7/lib/python3.7/site-packages (2.8.4)
WARNING: You are using pip version 19.2.3, however version 19.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

Collecting psycopg2 Downloading https://files.pythonhosted.org/packages/5c/1c/6997288da181277a0c29bc39a5f9143ff20b8c99f2a7d059cfb55163e165/psycopg2-2.8.3.tar.gz (https://files.pythonhosted.org/packages/5c/1c/6997288da181277a0c29bc39a5f9143ff20b8c99f2a7d059cfb55163e165/psycopg2-2.8.3.tar.gz) (377kB) 100% |████████████████████████████████| 378kB 6.6MB/s eta 0:00:01 Building wheels for collected packages: psycopg2 Building wheel for psycopg2 (setup.py) ... done Stored in directory: /Users/EdwardArroyo/Library/Caches/pip/wheels/48/06/67/475967017d99b988421b87bf7ee5fad0dad789dc349561786b Successfully built psycopg2 Installing collected packages: psycopg2 Successfully installed psycopg2-2.8.3

In [2]:
```
!pip install area
```

```
Requirement already satisfied: area in /Library/Frameworks/Python.framework/Vers
ions/3.7/lib/python3.7/site-packages (1.1.1)
WARNING: You are using pip version 19.2.3, however version 19.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

In [3]:
```
!pip install folium
```

```
Requirement already satisfied: folium in /Library/Frameworks/Python.framework/Ve
rsions/3.7/lib/python3.7/site-packages (0.10.0)
Requirement already satisfied: requests in /Library/Frameworks/Python.framework/
Versions/3.7/lib/python3.7/site-packages (from folium) (2.22.0)
Requirement already satisfied: jinja2>=2.9 in /Library/Frameworks/Python.framewo
rk/Versions/3.7/lib/python3.7/site-packages (from folium) (2.10.3)
Requirement already satisfied: numpy in /Library/Frameworks/Python.framework/Ver
sions/3.7/lib/python3.7/site-packages (from folium) (1.17.2)
Requirement already satisfied: branca>=0.3.0 in /Library/Frameworks/Python.frame
work/Versions/3.7/lib/python3.7/site-packages (from folium) (0.3.1)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /Library/Frameworks/Pyth
on.framework/Versions/3.7/lib/python3.7/site-packages (from requests->folium)
(3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /Library/Frameworks/Python.
framework/Versions/3.7/lib/python3.7/site-packages (from requests->folium) (201
9.9.11)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /Libra
ry/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages (from re
quests->folium) (1.25.3)
Requirement already satisfied: idna<2.9,>=2.5 in /Library/Frameworks/Python.fram
ework/Versions/3.7/lib/python3.7/site-packages (from requests->folium) (2.8)
Requirement already satisfied: MarkupSafe>=0.23 in /Library/Frameworks/Python.fr
amework/Versions/3.7/lib/python3.7/site-packages (from jinja2>=2.9->folium) (1.
1.1)
Requirement already satisfied: six in /Library/Frameworks/Python.framework/Versi
ons/3.7/lib/python3.7/site-packages (from branca>=0.3.0->folium) (1.12.0)
WARNING: You are using pip version 19.2.3, however version 19.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

In [4]:
```
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
```

In [5]:
```
import getpass

pw = getpass.getpass("Password: ")
```

```
Password: ········
```

In [6]:
```
db_connection = psycopg2.connect(host='129.105.208.229',dbname="chicago_crimes", u
ser="agc4488" , password=pw)


cursor = db_connection.cursor()
```

In [7]:
```
db_connection.commit()
```

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 resources 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 performance significantly specially 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(*[Math Processing Error]*) and O(*[Math Processing Error]*) where *[Math Processing Error]* is the size of the number
- **Search algorithms** used by the database engines vary in performance between O(*[Math Processing Error]*) and O(*[Math Processing Error]*) where *[Math Processing Error]* 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 other words, create a Choropleth map where the districts are shaded in proportion to the number of **crimes** in that district.

In [8]:
```
cursor.execute("SELECT district, count(district) from crimes GROUP BY district")
rows=cursor.fetchall()
```

In [9]:
```
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[9]:

	dist_num	number_of_crimes
0	1	31994
1	2	25160
2	3	27306
3	4	32434
4	5	25618

In [10]:
```
crimes_per_district_map = folium.Map(location =(41.8781, -87.6298),zoom_start=11)
```

In [11]:
```
## This will yield a warning about choropleth being deprecated in future versio
n...
# 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'],
#                legend_name="CRIME MAP"
#                )

# I pasted a screenshot of this warning in the next (markdown cell)
```

```
/Users/EdwardArroyo/anaconda3/lib/python3.7/site-packages/folium/folium.py:415: FutureWarning: The choropleth  method has been deprecated. Instead use the new Choropleth class, which has the same arguments. See the example notebook 'GeoJSON_and_choropleth' for how to do this.
  FutureWarning
```

In [ ]:
```
## Use this this instead of the code in last cell. Note "choropleth" is now "Choro
pletth".
folium.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'],
               legend_name="CRIMES PER DISTRICT"
               ).add_to(crimes_per_district_map)
crimes_per_district_map
```

In [13]:
```
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:
    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()
    #print(districts_crime_numbers)
    for district in districts_crime_numbers:
        folium.Marker(location =
                      police_station_location,
                      popup=folium.Popup(
                          html="District No : %s  has   Total Number of Crimes:%s"
%district ,max_width=450)).add_to(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 [14]:
```
#  db_connection.commit()
```

In [ ]:
```
crimes_per_district_map
```

Query #2:

- Calculate the total number of **violent crimes** in every district and plot that in a **table** on Choropleth map. In other words, create a Choropleth map where the districts are shaded in proportion to the number of **violent crimes** in that district.

N.B: A crime is considered a **violent crime** if the **PRIMARY_TYPE** of the crimes is `THEFT` , `ASSAULT` , `ROBBER` , `KIDNAPPING` , `CRIM SEXUAL ASSAULT` , `BATTERY` , or `MURDER` .

- Then find the total number of crimes in the district for each of these primary types of violent crime and add a popup marker (located at that district's police headquarter) that displays a DataFrame containing this data.

District Number 1 - Violent Crimes

	Description	Number of Violent Crimes
0	ASSAULT	1532
1	BATTERY	3305
2	CRIM SEXUAL ASSAULT	143
3	KIDNAPPING	12
4	ROBBERY	1195
5	THEFT	15607

In [16]:
```
violent_crime_categories='THEFT','ASSAULT','ROBBERY','KIDNAPPING','CRIM SEXUAL ASS
AULT','BATTERY','MURDER'
```

In [17]:
```
cursor.execute('''
    SELECT district, count(district)
    FROM crimes
    WHERE PRIMARY_TYPE in %s
    GROUP BY district
''',[violent_crime_categories])
rows=cursor.fetchall()
violent_crimes_per_district=pd.DataFrame(rows, columns=['dist_num','number_of_viol
ent_crimes'])
violent_crimes_per_district['dist_num'] = violent_crimes_per_district['dist_num'].
astype(str)
violent_crimes_per_district
```

Out[17]:

	dist_num	number_of_violent_crimes
0	1	21794
1	2	14517
2	3	14988
3	4	17128
4	5	13112
5	6	19772
6	7	15931
7	8	18530
8	9	14030
9	10	13988
10	11	18279
11	12	17472
12	14	13198
13	15	11927
14	16	9602
15	17	9418
16	18	20653
17	19	15429
18	20	5533
19	22	9326
20	24	9453
21	25	16032
22	31	3

In [18]:
```
# violent_crimes_per_district_map= folium.Map(location =(41.8781, -87.6298),zoom_s
tart=11)
# violent_crimes_per_district_map.choropleth(geo_data="Boundaries.geojson",
#               fill_color='YlOrRd',
#               fill_opacity=0.5,
#               line_opacity=1,
#               data = violent_crime_per_district,
#               key_on='feature.properties.dist_num',
#               columns = ['district_num', 'number_of_violent_crimes'],
#               legend_name="VOILENT CRIME MAP"
#               )
```

In [ ]:
```
violent_crimes_per_district_map= folium.Map(location =(41.8781, -87.6298),zoom_sta
rt=11)
folium.Choropleth(geo_data="Boundaries.geojson",
              fill_color='OrRd',
              fill_opacity=0.5,
              line_opacity=1,
              data = violent_crimes_per_district,
              key_on='feature.properties.dist_num',
              columns = ['dist_num', 'number_of_violent_crimes'],
              legend_name="VIOLENT CRIMES PER DISTRICT"
              ).add_to(violent_crimes_per_district_map)
violent_crimes_per_district_map
```

In addition, for each district find the block(s) that has the highest number of gun crimes in that district. Note that there might be a tie for the highest number of gun crimes. You need to find **all** such blocks. Add a popup marker (located at that district's police headquarter) that displays a DataFrame containing all such block along with the number of gun crimes for that block (i.e. the highest number of crimes for a district).

In [20]:
```
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:
    police_station_location =(police_station[0],police_station[1])
    cursor.execute('''
        SELECT PRIMARY_TYPE, count(PRIMARY_TYPE)
        FROM crimes
        WHERE district =%s AND PRIMARY_TYPE in %s
        GROUP BY PRIMARY_TYPE
    ''',[police_station[2],violent_crime_categories])
    data = cursor.fetchall()
#     print(data)
    violent_crimes_per_district = pd.DataFrame(data, columns=['Description', 'Numb
er of Violent Crimes'])
    header = violent_crimes_per_district.to_html(classes='table table-striped tabl
e-hover table-condensed table-responsive')
    folium.Marker(location=police_station_location, popup=folium.Popup(html="Distr
ict Number %s - Violent Crimes %s" %(police_station[2],header))).add_to(violent_cr
imes_per_district_map)
```

In [21]:
```
#  db_connection.commit()
```

In [ ]:
```
violent_crimes_per_district_map
```

Query #3:

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

N.B: A crime is considered a **gun related violent crime** if the word "gun" is contained in the **DESCRIPTION** and the **PRIMARY_TYPE** of the crimes is `THEFT` , `ASSAULT` , `ROBBER` , `KIDNAPPING` , `CRIM SEXUAL ASSAULT` , `BATTERY` , or `MURDER` .

- Then find the total number of crimes in the district for the different DESCRIPTIONs containing the word "gun" and add a popup marker (located at that district's police headquarter) that displays a DataFrame containing this data.

Lets first create a dataframe of **violent gun crimes per district**.

In [23]:
```
gun='%GUN%'
cursor.execute('''
    SELECT district, count(district)
    FROM crimes
    WHERE DESCRIPTION::text LIKE %s and PRIMARY_TYPE in %s
    GROUP BY district
''',[gun,violent_crime_categories])
rows = cursor.fetchall()
violent_gun_crimes_per_district = pd.DataFrame(rows, columns=['dist_num','number_of_violent_gun_crimes'])
violent_gun_crimes_per_district['dist_num'] = violent_gun_crimes_per_district['dist_num'].astype(str)
violent_gun_crimes_per_district
```

Out[23]:

	dist_num	number_of_violent_gun_crimes
0	1	354
1	2	1101
2	3	1454
3	4	1351
4	5	1225
5	6	1837
6	7	1646
7	8	1476
8	9	1335
9	10	1426
10	11	2234
11	12	1147
12	14	723
13	15	1452
14	16	269
15	17	514
16	18	351
17	19	450
18	20	177
19	22	760
20	24	427
21	25	1324

In [24]:
```
# violent_gun_crimes_per_district_map= folium.Map(location =(41.8781, -87.6298),zoom_start=11)
# violent_gun_crimes_per_district_map.choropleth(geo_data="Boundaries.geojson",
#                fill_color='YlOrRd',
#                fill_opacity=0.5,
#                line_opacity=1,
#                data = violent_gun_crimes_per_district,
#                key_on='feature.properties.dist_num',
#                columns = ['dist_num', 'number_of_violent_gun_crimes'],
#                legend_name="VIOLENT GUN CRIMES PER DISTRICT"
#                )
```

In [ ]:
```
violent_gun_crimes_per_district_map= folium.Map(location =(41.8781, -87.6298),zoom
_start=11)
folium.Choropleth(geo_data="Boundaries.geojson",
                  fill_color='OrRd',
                  fill_opacity=0.5,
                  line_opacity=1,
                  data = violent_gun_crimes_per_district,
                  key_on='feature.properties.dist_num',
                  columns = ['dist_num', 'number_of_violent_gun_crimes'],
                  legend_name="VIOLENT GUN CRIMES PER DISTRICT"
                  ).add_to(violent_gun_crimes_per_district_map)
violent_gun_crimes_per_district_map
```

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

In [26]:
```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), dis
trict 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 district=%s and DESCRIPTION::text LIKE %s and PRIMARY_TYPE in %s
        GROUP BY DESCRIPTION
    ''',[police_station[2],gun, violent_crime_categories])
    data=cursor.fetchall()
    df = pd.DataFrame(data, columns=['Description', 'Number of Violent Gun Crimes
'])
    header = df.to_html(classes='table table-striped table-hover table-condensed t
able-responsive')
    folium.Marker(location=police_station_location,popup=folium.Popup(html="Distri
ct No: %s GUN_Crime: %s" %(police_station[2],header) )).add_to(violent_gun_crimes_
per_district_map)
```

In [27]:
```
#  db_connection.commit()
```

In [ ]:
```
violent_gun_crimes_per_district_map
```

Query #4:

- Calculate the crime density per district

In [29]:
```
district=[]
tarea=[]

with open('Boundaries.geojson') as f:
    data = json.load(f)
    a = data['features'] # a is a list of district data (dictionaries)
    for i in range(len(a)):# a[i] is the dictionary for ith district in Boundarie
s.geojson
        obj=a[i]['geometry'] # list of coordinates for ith district
        n= a[i]['properties'] # district number and district label for ith distric
t
        district.append(n['dist_num']) # add district number to the district list
        # 1 square meter = 1/10000 hectares. area(obj) is in hectares
        tarea.append(area(obj)/10000) # add the the area (in hectares) to area lis
t

af=pd.DataFrame({'dist_num': district,'district_area_in_Hectares':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['di
strict_area_in_Hectares']/100))
final_data
```

Out[29]:

	dist_num	district_area_in_Hectares	number_of_crimes	crime_density
0	17	2492.727155	17165	689.0
1	20	1132.170216	9820	867.0
2	31	51.045317	16	31.0
3	31	799.507694	16	2.0
4	31	32.407658	16	49.0
5	19	2225.035732	26135	1175.0
6	25	2827.989237	31477	1113.0
7	14	1555.869965	21793	1401.0
8	7	1688.670732	30748	1821.0
9	3	1576.063931	27306	1733.0
10	4	7068.152865	32434	459.0
11	6	2099.682124	35855	1708.0
12	22	3490.416073	18582	532.0
13	5	3318.613379	25618	772.0
14	24	1406.081387	16946	1205.0
15	16	8171.776367	19972	244.0
16	8	5992.169760	37275	622.0
17	18	1215.520046	31003	2551.0
18	12	2509.453028	29678	1183.0
19	11	1582.727274	40386	2552.0
20	15	989.631393	23615	2386.0
21	10	2038.988883	27513	1349.0
22	1	1214.818895	31994	2634.0
23	9	3505.216898	26584	758.0
24	2	1949.690970	25160	1290.0

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 [30]:
```
# 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 = violent_gun_crimes_per_district,
#                key_on='feature.properties.dist_num',
#                columns = ['dist_num', 'number_of_violent_gun_crimes'],
#                legend_name="GUN CRIME"
#                )
```

In [ ]:
```
gun_crime_arrests_map = folium.Map(location =(41.8781, -87.6298),zoom_start=11)
folium.Choropleth(geo_data="Boundaries.geojson",
                  fill_color='OrRd',
                  fill_opacity=0.5,
                  line_opacity=1,
                  data = violent_gun_crimes_per_district,
                  key_on='feature.properties.dist_num',
                  columns = ['dist_num', 'number_of_violent_gun_crimes'],
                  legend_name="VIOLENT GUN CRIMES PER DISTRICT"
                  ).add_to(gun_crime_arrests_map)
gun_crime_arrests_map
```

In [32]:
```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), dis
trict 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(arrest), arrest,latitude, longitude
        FROM crimes
        WHERE district=%s and DESCRIPTION::text LIKE %s GROUP BY caseno,block,
              DESCRIPTION,arrest, latitude, longitude

    ''',[police_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="Di
strict No: %s <br> Description: %s <br> Block: %s" %(police_station[2],crime[2],cr
ime[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="Di
strict No: %s <br> Description: %s<br> Block: %s" %(police_station[2],crime[2],cri
me[1])),icon=folium.Icon(color='red', icon='remove-sign'),).add_to(marker_cluster)
```

In [43]:
```
db_connection.commit()
```

In [34]:
```
def embed_map(m):
    from IPython.display import IFrame

    m.save('index.html')
    return IFrame('index.html', width='100%', height='750px')
```

In [ ]:
```
embed_map(gun_crime_arrests_map)
```

Query #6:

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

Locate the **farthest** gun crime from the police station in every district. Create a Choropleth map where the districts are shaded in proportion to the number of **gun crimes in that district. For each district, find the gun crime that was** farthest **from police station. Add a pop-up on the Choropleth map to display the district number and the Block where the** farthest **gun crime occurred. Also add circle marker (of radius 5) at the location of the** farthest** gun crime.

In [36]:
```
gun='%GUN%'

cursor.execute('''

    SELECT district, count(district)
    FROM crimes
    WHERE DESCRIPTION::text LIKE %s
    GROUP BY district

''',[gun])
rows=cursor.fetchall()
gun_crimes_per_district = pd.DataFrame(rows, columns=['dist_num','number_of_gun_crimes'])
gun_crimes_per_district['dist_num'] = gun_crimes_per_district['dist_num'].astype(str)

gun_crimes_per_district.head()
```

Out[36]:

	dist_num	number_of_gun_crimes
0	1	410
1	2	1348
2	3	1928
3	4	1960
4	5	1925

In [37]:
```
# 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 = gun_crimes_per_district,
#                 key_on='feature.properties.dist_num',
#                 columns = ['dist_num', 'number_of_gun_crimes'],
#                 legend_name="GUN CRIMES PER DISTRICT"
#                 )
# farthest_block_gun_crime_map
```

In [ ]:
```
farthest_block_gun_crime_map = folium.Map(location =(41.8781, -87.6298),zoom_star
t=11)
folium.Choropleth(geo_data="Boundaries.geojson",
                  fill_color='OrRd',
                  fill_opacity=0.5,
                  line_opacity=1,
                  data = gun_crimes_per_district,
                  key_on='feature.properties.dist_num',
                  columns = ['dist_num', 'number_of_gun_crimes'],
                  legend_name="GUN CRIMES PER DISTRICT"
                  ).add_to(farthest_block_gun_crime_map )
farthest_block_gun_crime_map
```

In [39]:
```
cursor.execute("""SELECT DISTINCT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_
IS)), district 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_i
s,
                                                    ST_Distance(A.where_is,B.where_
is)
                        FROM crimes as A, police_stations as B
                        WHERE
                              A.district=%s and DESCRIPTION::text LIKE %s and B.dis
trict= %s
                              and 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_stations as B
          WHERE A.district=%s  and DESCRIPTION::text LIKE %s and B.district= %s )
as f)



    """,[police_station[2],gun,police_station[2],police_station[2],gun, police_sta
tion[2]])


    farthest_block_gun_crime = cursor.fetchall()

    print(farthest_block_gun_crime)
    print((farthest_block_gun_crime[0][2],farthest_block_gun_crime[0][2]))

    if not farthest_block_gun_crime:
        continue

    cursor.execute('''
    SELECT ST_X(ST_AsText(%s)), ST_Y(ST_AsText(%s))
    ''',(farthest_block_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.Popu
p(html="Police Station <br> District No.:%s <br> Farthest Gun_Crime Block:%s"%(far
thest_block_gun_crime[0][0],farthest_block_gun_crime[0][1]))).add_to(farthest_bloc
k_gun_crime_map)
    folium.CircleMarker(farthest_block_gun_crime_location[0],radius=5,color='#ff31
87',popup=folium.Popup(html="District No.:%s <br> Block:%s"%(farthest_block_gun_cr
ime[0][0],farthest_block_gun_crime[0][1]))).add_to(farthest_block_gun_crime_map)
```

```
[(8, '054XX S HARLEM AVE', '0101000020E6100000131BC6B99AE544406E3A0C2A4DF355C0',
10356.40924857)]
('0101000020E6100000131BC6B99AE544406E3A0C2A4DF355C0', '0101000020E6100000131BC6
B99AE544406E3A0C2A4DF355C0')
[(17, '032XX N TALMAN AVE', '0101000020E610000070A8C6E84CF844402B4272B170EC55C0
', 3769.89527319)]
('0101000020E610000070A8C6E84CF844402B4272B170EC55C0', '0101000020E610000070A8C6
E84CF844402B4272B170EC55C0')
[(9, '035XX W 38TH PL', '0101000020E61000009AE3E49764E94440AF39CD239BED55C0', 73
92.75745613)]
('0101000020E61000009AE3E49764E94440AF39CD239BED55C0', '0101000020E61000009AE3E4
9764E94440AF39CD239BED55C0')
[(3, '027XX E 75TH ST', '0101000020E61000001A6D8A0469E14440A13D94F19AE355C0', 55
19.41242301)]
('0101000020E61000001A6D8A0469E14440A13D94F19AE355C0', '0101000020E61000001A6D8A
0469E14440A13D94F19AE355C0')
[(10, '024XX S ASHLAND AVE', '0101000020E6100000CBC506758BEC444088CAEC179FEA55C0
', 4738.32815077)]
('0101000020E6100000CBC506758BEC444088CAEC179FEA55C0', '0101000020E6100000CBC506
758BEC444088CAEC179FEA55C0')
[(24, '032XX W DEVON AVE', '0101000020E610000088E18D46AAFF44405A45B2596BED55C0',
4283.72628994)]
('0101000020E610000088E18D46AAFF44405A45B2596BED55C0', '0101000020E610000088E18D
46AAFF44405A45B2596BED55C0')
[(4, '107XX S STATE LINE RD', '0101000020E61000001CC20D05A7D944404CDBA61592E155C
0', 4893.11189547)]
('0101000020E61000001CC20D05A7D944404CDBA61592E155C0', '0101000020E61000001CC20D
05A7D944404CDBA61592E155C0')
[(20, '006XX W FOSTER DR', '0101000020E61000009353F4ED08FD44409D70F3E978E955C0',
5008.23515645)]
('0101000020E61000009353F4ED08FD44409D70F3E978E955C0', '0101000020E61000009353F4
ED08FD44409D70F3E978E955C0')
[(7, '070XX S DAN RYAN EXPY OB', '0101000020E61000003211719E2AE24440088A7DD40FE8
55C0', 3900.8163863)]
('0101000020E61000003211719E2AE24440088A7DD40FE855C0', '0101000020E6100000321171
9E2AE24440088A7DD40FE855C0')
[(18, '006XX E GRAND AVE', '0101000020E610000031D0ABBD2CF2444099766C2F22E755C0',
3562.31044254)]
('0101000020E610000031D0ABBD2CF2444099766C2F22E755C0', '0101000020E610000031D0AB
BD2CF2444099766C2F22E755C0')
[(16, '117XX W IRVING PARK RD', '0101000020E6100000C64B65C12CFA4440B03D1B08B5FA5
5C0', 16882.55492555)]
('0101000020E6100000C64B65C12CFA4440B03D1B08B5FA55C0', '0101000020E6100000C64B65
C12CFA4440B03D1B08B5FA55C0')
[(25, '013XX N CENTRAL PARK AVE', '0101000020E61000003F75DC8BD0F34440DF989BF8D5E
D55C0', 5516.7825042)]
('0101000020E61000003F75DC8BD0F34440DF989BF8D5ED55C0', '0101000020E61000003F75DC
8BD0F34440DF989BF8D5ED55C0')
[(6, '075XX S SOUTH CHICAGO AVE', '0101000020E61000004B62B39D1BE1444039E22BE21DE
655C0', 5434.45624182)]
('0101000020E61000004B62B39D1BE1444039E22BE21DE655C0', '0101000020E61000004B62B3
9D1BE1444039E22BE21DE655C0')
[(22, '040XX W 115TH ST', '0101000020E61000003FC7D5358BD74440D4CDBE2532EE55C0',
5952.2111397)]
('0101000020E61000003FC7D5358BD74440D4CDBE2532EE55C0', '0101000020E61000003FC7D5
358BD74440D4CDBE2532EE55C0')
[(14, '012XX W NORTH AVE', '0101000020E61000002E0E73F193F44440E7E732A71EEA55C0',
4393.2011984)]
('0101000020E61000002E0E73F193F44440E7E732A71EEA55C0', '0101000020E61000002E0E73
F193F44440E7E732A71EEA55C0')
[(1, '030XX S FORT DEARBORN', '0101000020E61000006AB81C3D6BEB444095F2A1DEE8E655C
0', 2167.86247749)]
('0101000020E61000006AB81C3D6BEB444095F2A1DEE8E655C0', '0101000020E61000006AB81C
3D6BEB444095F2A1DEE8E655C0')
[(2, '055XX S LAKE SHORE DR SB', '0101000020E6100000CF54F7E4BBE54440E6E04B4114E5
```

In [40]:
```
print(farthest_block_gun_crime)
print((farthest_block_gun_crime[0][2],farthest_block_gun_crime[0][2]))
```

```
[(19, '047XX N VIRGINIA AVE', '0101000020E6100000CB569E0DE7FB4440769FDCE9C0EC55C
0', 5335.50876621)]
('0101000020E6100000CB569E0DE7FB4440769FDCE9C0EC55C0', '0101000020E6100000CB569E
0DE7FB4440769FDCE9C0EC55C0')
```

In [41]:
```
db_connection.commit()
```

In [ ]:
```
farthest_block_gun_crime_map
```

Requirements

The HTML document your are submitting along with this notebook file must have the source code and the output for the following requirements

Requirement #1:

- Locate the **Block(s)** that has the **highest number of gun crimes**. Create a Choropleth map where the districts are shaded in proportion to the number of **gun crimes** in that district. In addition, for each district find the block(s) that has the highest number of gun crimes in that district. Note that there might be a tie for the highest number of gun crimes. You need to find **all** such blocks. Add a popup marker (located at that district's police headquarter) that displays a DataFrame containing all such block along with the number of gun crimes for that block (i.e. the highest number of crimes for a district).

In [112]:
```
blocks_gun_crime_map2 = folium.Map(location =(41.8781, -87.6298),zoom_start=11)
blocks_gun_crime_map2.choropleth(geo_data="Boundaries.geojson",
              fill_color='YlOrRd',
              fill_opacity=0.5,
              line_opacity=1,
              data = gun_crimes_per_district,
              key_on='feature.properties.dist_num',
              columns = ['dist_num', 'number_of_gun_crimes'],
              legend_name="GUN CRIME"
              )
```

In [111]:
```
#cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), di
strict from police_stations where district!='Headquarters'""")

# find highest gun crime block in each district
gun='%GUN%'


blocks_gun_violent_crimes2 = cursor.fetchall()
blocks_gun_violent_crimes_df2 = pd.DataFrame(blocks_gun_violent_crimes2, column
s=['block','district','gun_crimes'])
df3 = pd.DataFrame(blocks_gun_violent_crimes2, columns=['block','district','gun_cr
imes'])
df4 = pd.DataFrame(blocks_gun_violent_crimes2, columns=['block','district','gun_cr
imes'])

for district in [x for x in range(1,26) if x not in [13,21,23]]:
    cursor.execute("SELECT block, district, count(block) from crimes where DESCRIP
TION::text LIKE %s GROUP BY block, district"
,[gun])
    df2 = blocks_gun_violent_crimes_df2.query(f'district=={district}')
    block_gun_violent_crimes_df3 = df2[df2.gun_crimes == df2.gun_crimes.max()]
    header = block_gun_violent_crimes_df3.to_html(classes='table table-striped tab
le-hover table-condensed table-responsive')
    folium.Marker(location=police_station_location,popup=folium.Popup(html="Distri
ct No: %s Block with Highest Gun Crimes: %s" %(police_station[2],header))).add_to
(blocks_gun_crime_map2)
    print(block_gun_violent_crimes_df3)
    df4 = df3.append(block_gun_violent_crimes_df3)
```

```
                    block  district  gun_crimes
9026  0000X E ROOSEVELT RD         1          10
                 block  district  gun_crimes
2336  003XX E 47TH ST         2          15
                                  block  district  gun_crimes
3883  064XX S DR MARTIN LUTHER KING JR DR         3          39
                   block  district  gun_crimes
5479  078XX S ESSEX AVE         4          17
                  block  district  gun_crimes
9808   130XX S EVANS AVE         5          13
11805   102XX S STATE ST         5          13
                        block  district  gun_crimes
9268  083XX S COTTAGE GROVE AVE         6          23
                  block  district  gun_crimes
5077  066XX S HALSTED ST         7          21
                    block  district  gun_crimes
2592  063XX S ARTESIAN AVE         8          18
                      block  district  gun_crimes
583           045XX S WOOD ST         9          13
13069  054XX S WINCHESTER AVE         9          13
                 block  district  gun_crimes
1122  012XX S AVERS AVE        10          27
                      block  district  gun_crimes
6314  008XX N MONTICELLO AVE        11          19
                     block  district  gun_crimes
5575   023XX W JACKSON BLVD        12          11
11459   022XX W MAYPOLE AVE        12          11
                  block  district  gun_crimes
10159  033XX W BEACH AVE        14          10
                     block  district  gun_crimes
7399  0000X S LEAMINGTON AVE        15          23
                   block  district  gun_crimes
7415  062XX W BELMONT AVE        16           5
                    block  district  gun_crimes
10228   043XX N KIMBALL AVE        17           7
10663  037XX W MONTROSE AVE        17           7
                  block  district  gun_crimes
1184  013XX N HUDSON AVE        18           8
                  block  district  gun_crimes
1892    035XX N CLARK ST        19           7
8624  011XX W WILSON AVE        19           7
                   block  district  gun_crimes
339  050XX N WINTHROP AVE        20           6
                   block  district  gun_crimes
11205  103XX S HALSTED ST        22          10
                    block  district  gun_crimes
9498  076XX N BOSWORTH AVE        24          13
                 block  district  gun_crimes
1847  055XX W NORTH AVE        25          26
```

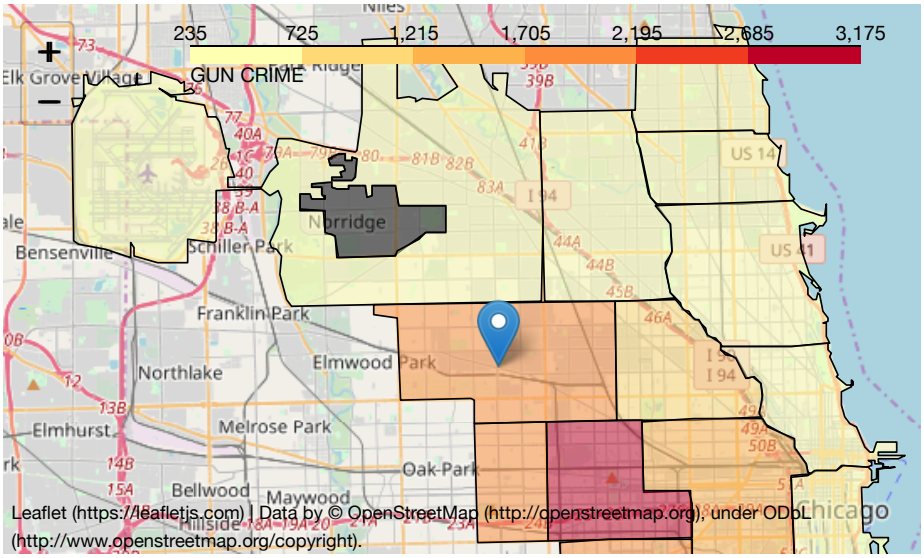
In [113]:
```
df3.head()
```

Out[113]:

	block	district	gun_crimes
0	015XX N RIDGEWAY AVE	25	2
1	090XX S KINGSTON AVE	4	1
2	040XX W WRIGHTWOOD AVE	25	1
3	022XX S WENTWORTH AVE	9	1
4	050XX S ELIZABETH ST	9	2

In [114]:
```
df4 = pd.DataFrame(blocks_gun_violent_crimes2, columns=['block','district','gun_crimes'])

df5 = df3[(df3['district'] == 25) & (df3['gun_crimes'] == 26)]

df5

header = df5.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 Block with Highest Gun Crimes: %s" %(police_station[2],header))).add_to(blocks_gun_crime_map2)

blocks_gun_crime_map2
```

Out[114]:



Requirement #2:

- Calculate the gun crimes density in every district **(See Query 4.)**

In [45]:
```
gun='%GUN%'
cursor.execute('''
    SELECT district, count(district)
    FROM crimes
    WHERE DESCRIPTION::text LIKE %s
    GROUP BY district
''',[gun])

rows = cursor.fetchall()
gun_crimes_per_district = pd.DataFrame(rows, columns=['dist_num','number_of_gun_cr
imes'])
gun_crimes_per_district['dist_num'] = gun_crimes_per_district['dist_num'].astype(s
tr)

district=[]
tarea=[]

with open('Boundaries.geojson') as f:
    data = json.load(f)
    a = data['features'] # a is a list of district data (dictionaries)
    for i in range(len(a)):# a[i] is the dictionary for ith district in Boundarie
s.geojson
        obj=a[i]['geometry'] # list of coordinates for ith district
        n= a[i]['properties'] # district number and district label for ith distric
t
        district.append(n['dist_num']) # add district number to the district list
        # 1 square meter = 1/10000 hectares. area(obj) is in hectares
        tarea.append(area(obj)/10000) # add the the area (in hectares) to area lis
t

af=pd.DataFrame({'dist_num': district,'district_area_in_Hectares':tarea})
af['dist_num'] = af['dist_num'].astype(str)
final_data2= pd.merge(af, gun_crimes_per_district, on='dist_num', how='inner')
final_data2['crime_density'] = round(final_data2['number_of_gun_crimes']/(final_da
ta2['district_area_in_Hectares']/100))
final_data2
```

Out[45]:

	dist_num	district_area_in_Hectares	number_of_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. Create a Choropleth map where the districts are shaded in proportion to the number of UNLAWFUL POSS OF HANDGUN crimes in that district. For each district, find the UNLAWFUL POSS OF HANDGUN crime that was **farthest** from police station. Add a popup on the Choropleth map to display the district number and the Block where the **farthest** UNLAWFUL POSS OF HANDGUN crime occured. Also add circle marker (of radius 5) at the location of the **farthest** UNLAWFUL POSS OF HANDGUN crime. (**SEE Query 6.**)

In [46]:
```
handguns='UNLAWFUL POSS OF HANDGUN'

cursor.execute("SELECT district, count(district) from crimes where DESCRIPTION::te
xt LIKE %s GROUP BY district",[handguns])
unlawful_poss_handguns = cursor.fetchall()
unlawful_poss_handguns_df = pd.DataFrame(unlawful_poss_handguns, columns=['dist_nu
m','unlawful_poss_handguns'])
unlawful_poss_handguns_df['dist_num'] = unlawful_poss_handguns_df['dist_num'].asty
pe(str)

unlawful_poss_handguns_df
```

Out[46]:

	dist_num	unlawful_poss_handguns
0	1	48
1	2	193
2	3	400
3	4	478
4	5	570
5	6	591
6	7	829
7	8	339
8	9	376
9	10	694
10	11	839
11	12	157
12	14	66
13	15	478
14	16	44
15	17	34
16	18	51
17	19	30
18	20	30
19	22	237
20	24	74
21	25	321

In [47]:
```
farthest_block_unlawful_poss_handguns_map = folium.Map(location =(41.8781, -87.629
8),zoom_start=11)
folium.Choropleth(geo_data="Boundaries.geojson",
                  fill_color='YlOrRd',
                  fill_opacity=0.5,
                  line_opacity=1,
                  data = unlawful_poss_handguns_df,
                  key_on='feature.properties.dist_num',
                  columns = ['dist_num', 'unlawful_poss_handguns'],
                  legend_name="UNLAWFUL POSS OF HANDGUN PER DISTRICT"
                  ).add_to(farthest_block_unlawful_poss_handguns_map )
```

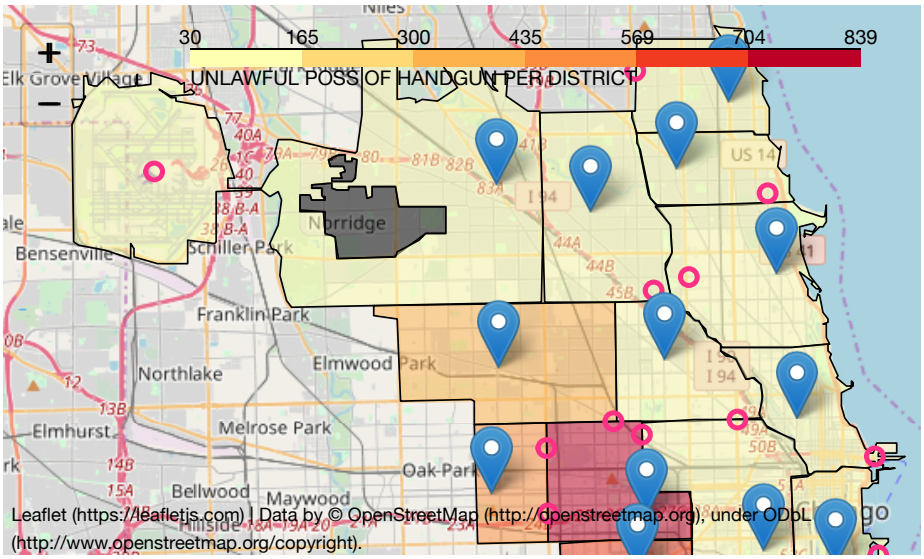
Out[47]: <folium.features.Choropleth at 0x15329add8>

In [48]:
```
cursor.execute("""SELECT DISTINCT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_
IS)), district from police_stations where district!='Headquarters'""")
police_stations = cursor.fetchall()

gun='%UNLAWFUL POSS OF HANDGUN%'

for police_station in police_stations:

    cursor.execute("""SELECT DISTINCT on (A.block) A.district, A.block, A.where_i
s,
                                                    ST_Distance(A.where_is,B.where_
is)
                        FROM crimes as A, police_stations as B
                        WHERE
                              A.district=%s and DESCRIPTION::text LIKE %s and B.dis
trict= %s
                              and 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_stations as B
          WHERE A.district=%s  and DESCRIPTION::text LIKE %s and B.district= %s )
as f)



    """,[police_station[2],gun,police_station[2],police_station[2],gun, police_sta
tion[2]])


    farthest_block_unlawful_poss_handguns = cursor.fetchall()

    print(farthest_block_unlawful_poss_handguns)

    if not farthest_block_unlawful_poss_handguns:
        continue

    cursor.execute('''
    SELECT ST_X(ST_AsText(%s)), ST_Y(ST_AsText(%s))
    ''',(farthest_block_unlawful_poss_handguns[0][2],farthest_block_unlawful_poss_
handguns[0][2]))

    farthest_block_unlawful_poss_handguns_location = cursor.fetchall()

    folium.Marker(location=(police_station[0],police_station[1]),popup=folium.Popu
p(html="Police Station <br> District No.:%s <br> Farthest Unlawful Handgun Possion
Block:%s"%(farthest_block_unlawful_poss_handguns[0][0],farthest_block_unlawful_pos
s_handguns[0][1]))).add_to(farthest_block_unlawful_poss_handguns_map)
    folium.CircleMarker(farthest_block_unlawful_poss_handguns_location[0],radius=
5,color='#ff3187',popup=folium.Popup(html="District No.:%s <br> Block:%s"%(farthes
t_block_unlawful_poss_handguns[0][0],farthest_block_unlawful_poss_handguns
[0][1]))).add_to(farthest_block_unlawful_poss_handguns_map)
```

```
[(8, '069XX W ARCHER AVE', '0101000020E6100000D0ED67A963E5444068C7C130D6F255C0',
9545.3168851)]
[(17, '029XX W ROSCOE ST', '0101000020E6100000AB183AB8ACF84440E046A44FE6EC55C0',
2968.07531325)]
[(9, '035XX W 38TH PL', '0101000020E61000006A7CE79664E9444013187BC79AED55C0', 73
90.29909706)]
[(3, '027XX E 75TH ST', '0101000020E610000058F8C0AF4DE144404626EF02B6E355C0', 53
34.90988188)]
[(10, '016XX W CERMAK RD', '0101000020E610000091FCC87A18ED444083643E76B0EA55C0',
4619.80562291)]
[(24, '070XX N KEDZIE AVE', '0101000020E61000005C0E17602F0145406D596DE763ED55C0
', 4233.16567589)]
[(4, '095XX S EDWARD BARRON DR', '0101000020E610000070D3464581DC4440F42C361DA9E1
55C0', 4736.50993999)]
[(20, '049XX N SHERIDAN RD', '0101000020E610000060EEA2CE6DFC44403047E3DEE9E955C0
', 4238.31185545)]
[(7, '071XX S LAFAYETTE AVE', '0101000020E6100000814E4DF1D4E14440437AD3F218E855C
0', 3838.86251642)]
[(18, '006XX E GRAND AVE', '0101000020E610000031D0ABBD2CF2444099766C2F22E755C0',
3562.31044254)]
[(16, '0000X W TERMINAL ST', '0101000020E6100000B6180E1450FD44404170097E03FA55C0
', 15672.05199393)]
[(25, '036XX W DIVISION ST', '0101000020E6100000FFCAFE048DF344406E3CF1FFF7ED55C0
', 5284.93050531)]
[(6, '079XX S GREENWOOD AVE', '0101000020E6100000D963C9EC11E04440715413E943E655C
0', 5175.14227991)]
[(22, '040XX W 115TH ST', '0101000020E61000008068674A8BD7444040A6523F31EE55C0',
5946.07514383)]
[(14, '012XX N ASHLAND AVE', '0101000020E6100000FE87CB5CA6F3444019C7644BB8EA55C0
', 3346.49947608)]
[(1, '014XX S LYNN WHITE DR', '0101000020E6100000DF8F827757EE4440EA78553203E755C
0', 1986.63468184)]
[(2, '055XX S LAKE SHORE DR SB', '0101000020E6100000813ACC34ADE54440E8B58BC918E5
55C0', 5687.71878513)]
[(12, '009XX N KEDZIE AVE', '0101000020E610000076CD02F207F3444005CB4BDD39ED55C0
', 5551.68392132)]
[(15, '004XX S CICERO AVE', '0101000020E610000073D918D6F5EF4440DE672AFEAEEF55C0
', 2585.13252336)]
[(11, '007XX N CICERO AVE', '0101000020E61000007532CE657EF244402C6F6F00BBEF55C0
', 4502.21004824)]
[(5, '015XX W 119TH ST', '0101000020E61000001BFAEF0DBAD6444002E3167C45EA55C0', 6
253.48700573)]
[(19, '023XX W ADDISON ST', '0101000020E6100000506CDACC2EF94440BFFD60B000EC55C0
', 4024.27150353)]
```

In [49]:
```
farthest_block_unlawful_poss_handguns_map
```

Out[49]:



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)**. (**SEE Query 5.**)

In [50]:
```
final_data2['dist_num'] = final_data2['dist_num'].astype(str)

final_data2.head()
```

Out[50]:

	dist_num	district_area_in_Hectares	number_of_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

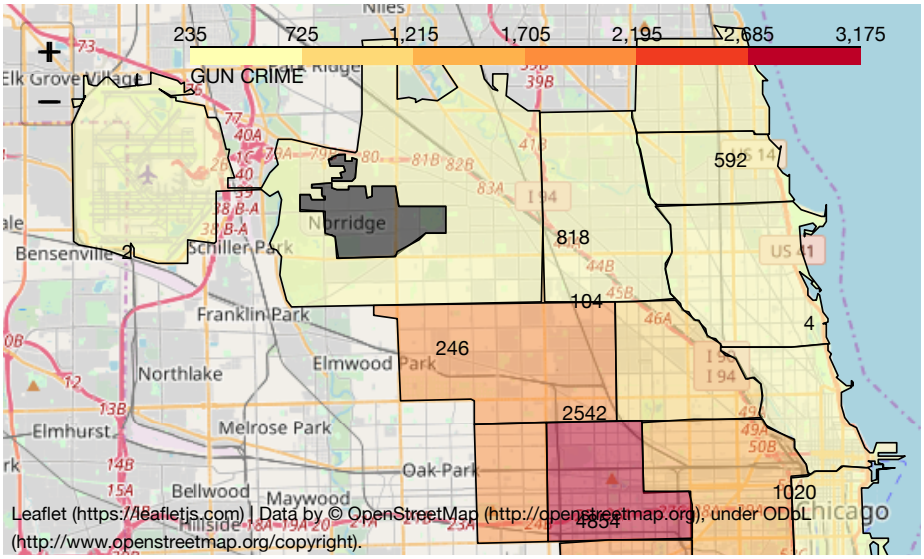
In [51]:
```
gun_crime_location_map = folium.Map(location =(41.8781, -87.6298),zoom_start=11)
folium.Choropleth(geo_data="Boundaries.geojson",
                  fill_color='YlOrRd',
                  fill_opacity=0.5,
                  line_opacity=1,
                  data = final_data2,
                  key_on='feature.properties.dist_num',
                  columns = ['dist_num', 'number_of_gun_crimes'],
                  legend_name="GUN CRIME"
                  ).add_to(gun_crime_location_map )
```

Out[51]:
```
<folium.features.Choropleth at 0x11701f6a0>
```

In [52]:
```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), dis
trict from police_stations where district!='Headquarters'""")
gun='%GUN%'

police_stations = cursor.fetchall()

marker_cluster = MarkerCluster().add_to(gun_crime_location_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
(description), location_description, latitude, longitude from crimes where distric
t=%s and DESCRIPTION::text LIKE %s and (location_description = 'RESIDENCE' or loca
tion_description = 'STREET') GROUP BY caseno,block, DESCRIPTION, location_descript
ion, latitude, longitude""",[police_station[2],gun])
    crimes_per_area = cursor.fetchall()
    for crime in crimes_per_area:
        if crime[4]=='RESIDENCE':
            folium.Marker(location=(crime[5],crime[6]),popup=folium.Popup(html="Di
strict No: %s <br> Description: %s <br> Block: %s" %(police_station[2],crime[2],cr
ime[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="Di
strict No: %s <br> Description: %s<br> Block: %s" %(police_station[2],crime[2],cri
me[1])),icon=folium.Icon(color='red', icon='ok-sign'),).add_to(marker_cluster)
```

In [52]:
```
gun_crime_location_map
```

Out[52]:



In [ ]:
```

```