MSDS420

Assignment 3: Chicago Crimes

Author : Atef Bader, PhD

Last Edit : 3/30/2019

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

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)](https://postgis.net/docs/manual-1.4/ch04.html)

In [13]:
```
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 [14]:
```
# To run the script on the complete dataset takes roughly 35 minutes to omplete.
# 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 s
ubmit your assignment on this dataset

# db_connection = psycopg2.connect(host='129.105.208.229',dbname="chicago_crimes",
user="YourNetID" , password="YourPassword")




# Use the following dataset for unit testing purposes only. It takes roughly 5 minu
tes to omplete.
# Comment the following line when you are done with your unit testing and ready to
run your assignment on the complete dataset and submit your Assignment

db_connection = psycopg2.connect(host='129.105.208.229',dbname="chicago_crimes", us
er="" , password="")


cursor = db_connection.cursor()
```

Requirements

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

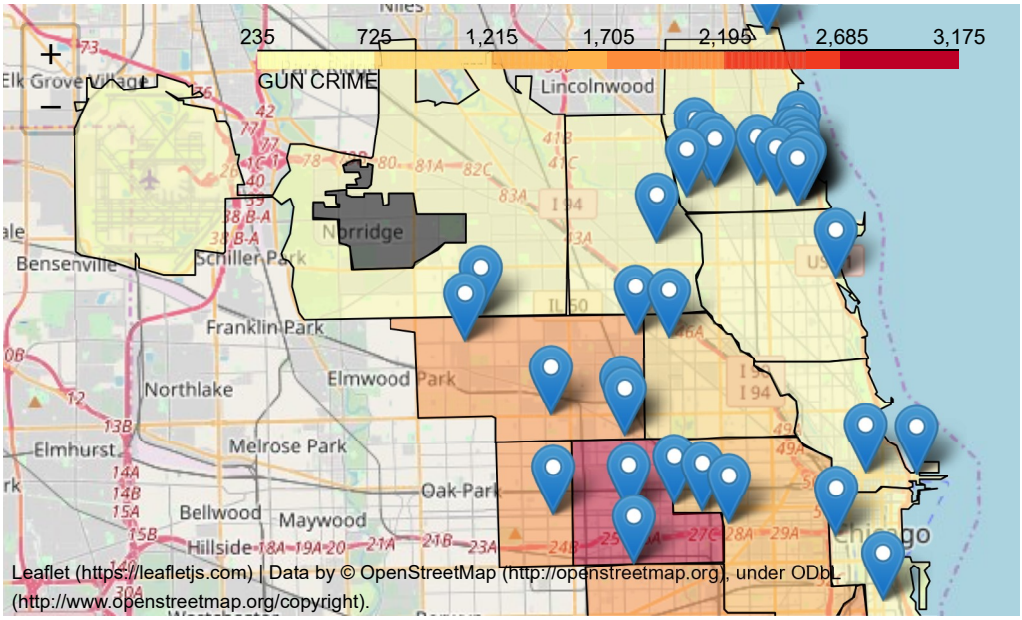
Requirement #1:

- 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 [15]:
```
#My understanding is that I am requested to disply the block(s) with highest number
of gun crime within each district.


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

highest_block_gun_crime_map = folium.Map(location =(41.8781, -87.6298),zoom_start=1
1)
highest_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"
              )

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 district,block, where_is, count(block)
    FROM crimes
    WHERE district=%s and DESCRIPTION::text LIKE %s
    GROUP BY block, district, where_is
    HAVING count(block)=(
    SELECT MAX(count)
    FROM (SELECT district,block, where_is, count(block) as count
    FROM crimes WHERE district=%s and DESCRIPTION::text LIKE %s GROUP BY block, dis
trict, where_is) AS f)""",[police_station[2],gun,police_station[2],gun])

    highest_block_gun_crime = cursor.fetchall()
    for i in highest_block_gun_crime:
        cursor.execute("SELECT ST_X(ST_AsText(%s)), ST_Y(ST_AsText(%s))",(i[2],i
[2]))
        highest_block_gun_crime_location = cursor.fetchall()
        folium.Marker(highest_block_gun_crime_location[0],popup=folium.Popup(html="
District No.:%s <br> Block:%s <br> Total gun crimes:%s"%(i[0],i[1],i[3]))).add_to(h
ighest_block_gun_crime_map)
```

In [ ]:
```
highest_block_gun_crime_map # Blocks with a tie in highest number of gun crimes are all shown in each district.
```

Out[ ]:



Requirement #2:

- Calculate the gun crimes density in every district

In [ ]:
```
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, districts_gun_violent_crimes_df, on='dist_num', how='inner
')
final_data['guncrime_density'] = round(final_data['gun_crimes']/(final_data['distri
ct_area_inHectares']/100))
final_data
```

Out[ ]:

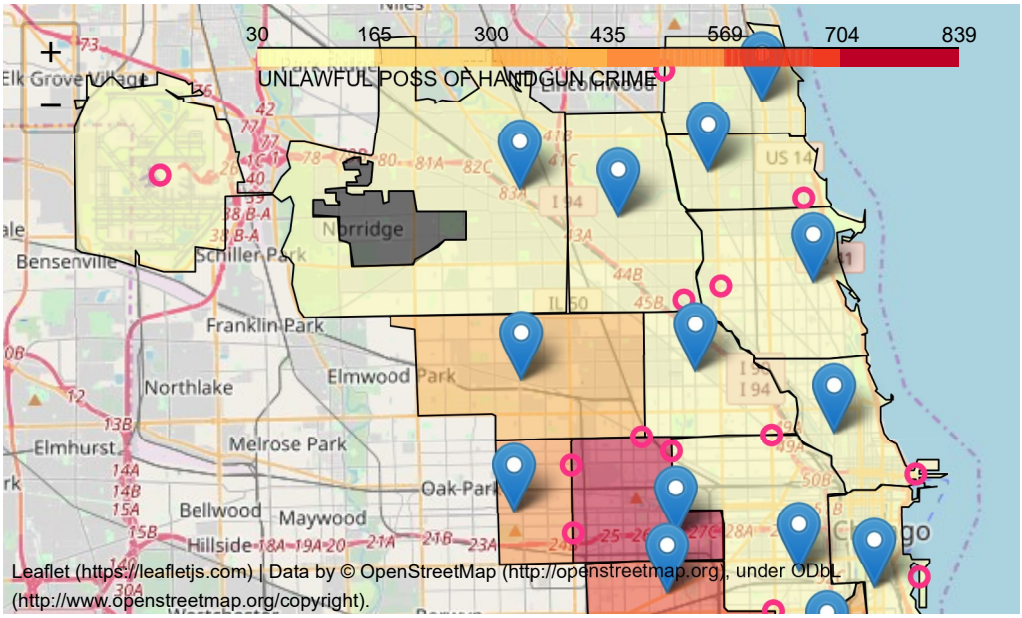
	dist_num	district_area_inHectares	gun_crimes	guncrime_density
0	17	2492.727155	574	23.0
1	20	1132.170216	235	21.0
2	19	2225.035732	501	23.0
3	25	2827.989237	1738	61.0
4	14	1555.869965	822	53.0
5	7	1688.670732	2739	162.0
6	3	1576.063931	1928	122.0
7	4	7068.152865	1960	28.0
8	6	2099.682124	2598	124.0
9	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

In [ ]:
```
posgun='UNLAWFUL POSS OF HANDGUN'
cursor.execute("SELECT district, count(district) from crimes where DESCRIPTION::tex
t LIKE %s GROUP BY district",[posgun])
districts_posgun_violent_crimes = cursor.fetchall()
districts_posgun_violent_crimes_df = pd.DataFrame(districts_posgun_violent_crimes,
columns=['dist_num','posgun_crimes'])
districts_posgun_violent_crimes_df['dist_num'] = districts_posgun_violent_crimes_df
['dist_num'].astype(str)
districts_posgun_violent_crimes_df

farthest_block_posgun_crime_map = folium.Map(location =(41.8781, -87.6298),zoom_sta
rt=11)
farthest_block_posgun_crime_map.choropleth(geo_data="Boundaries.geojson",
              fill_color='YlOrRd',
              fill_opacity=0.5,
              line_opacity=1,
              data = districts_posgun_violent_crimes_df,
              key_on='feature.properties.dist_num',
              columns = ['dist_num', 'posgun_crimes'],
              legend_name="UNLAWFUL POSS OF HANDGUN CRIME"
              )

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 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],
posgun,police_station[2]])
    farthest_block_posgun_crime = cursor.fetchall()
    cursor.execute("SELECT ST_X(ST_AsText(%s)), ST_Y(ST_AsText(%s))",(farthest_bloc
k_posgun_crime[0][2],farthest_block_posgun_crime[0][2]))
    farthest_block_posgun_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_posgun_crime[0][0],farthest_block_posgun_crime[0][1]))).add_to(farthest_b
lock_posgun_crime_map)
    folium.CircleMarker(farthest_block_posgun_crime_location[0],radius=5,color='#ff
3187',popup=folium.Popup(html="District No.:%s <br> Block:%s"%(farthest_block_posgu
n_crime[0][0],farthest_block_posgun_crime[0][1]))).add_to(farthest_block_posgun_cri
me_map)
farthest_block_posgun_crime_map
```

Out[ ]:

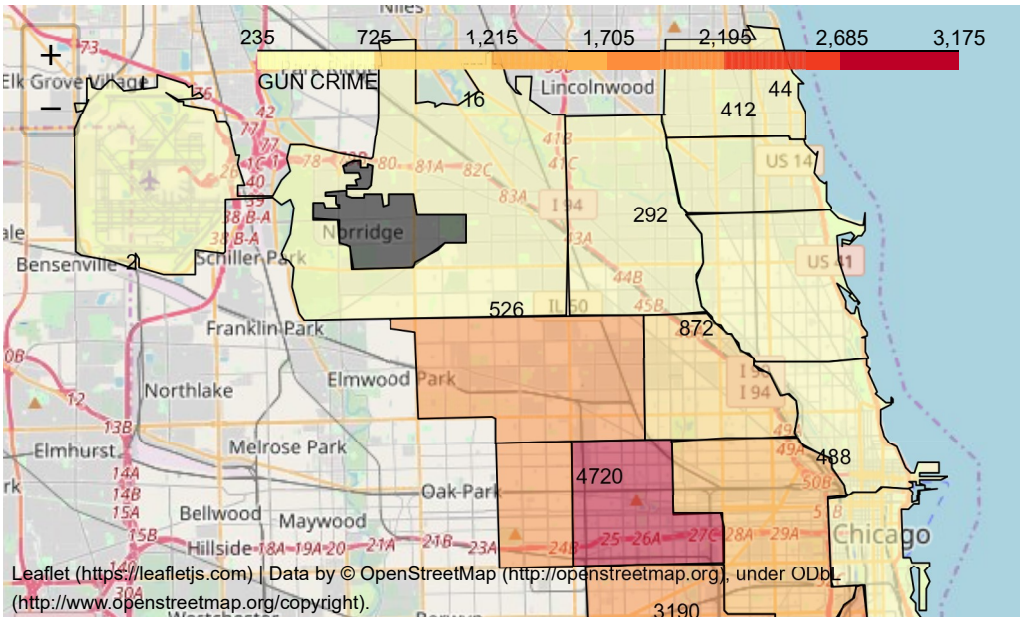


Requirement #4:

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

In [17]:
```
gun='%GUN%'
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"
              )

marker_cluster = MarkerCluster().add_to(districts_gun_violent_crimes_map)

for police_station in police_stations:
    police_station_location = (police_station[0],police_station[1])
    cursor.execute("""SELECT DISTINCT ON(caseno) caseno, Location_Description,latit
ude, longitude from crimes where district=%s and Description::text LIKE %s and Loca
tion_Description='RESIDENCE' GROUP BY caseno, Location_Description,latitude, longit
ude""",[police_station[2],gun])
    crime_residence = cursor.fetchall()
    cursor.execute("""SELECT DISTINCT ON(caseno) caseno, Location_Description,latit
ude, longitude from crimes where district=%s and Description::text LIKE %s and Loca
tion_Description='STREET' GROUP BY caseno, Location_Description,latitude, longitud
e""",[police_station[2],gun])
    crime_street = cursor.fetchall()
    for crime in crime_residence:
        folium.Marker(location=(crime[2],crime[3]),popup=folium.Popup(html="Distric
t No: %s <br> Description: %s" %(police_station[2],crime[1])),icon=folium.Icon(colo
r='green')).add_to(marker_cluster)
    for crime in crime_street:
        folium.Marker(location=(crime[2],crime[3]),popup=folium.Popup(html="Distric
t No: %s <br> Description: %s" %(police_station[2],crime[1])),icon=folium.Icon(colo
r='red')).add_to(marker_cluster)

districts_gun_violent_crimes_map
```

Out[17]:



In [ ]: