Assignment 3

PART 2

Query 4

Objective: Calculate the crime density per district

First, get the area of each district:

```
district=[]
tarea=[]

with open('Boundaries.geojson') as f:
    data = json.load(f)
    a = data['features']                  # Here a is a list of district data (dictionaries)
    for i in range(len(a)):                # Here a[i] is the dictionary for the ith district in Boundaries.geojson
        obj=a[i]['geometry']               # get list of coordinates for that list
        n= a[i]['properties']              # contains district number and district label for ith district
        district.append(n['dist_num'])    # add district number to the district list
        tarea.append(area(obj)/10000)   # add the  the area (in hectares to the tarea list
```

Use area() to find the district areas

Get the lists of district numbers and the area of the corresponding district (in hectares).

```
print(district)
```

```
['17', '20', '31', '31', '19', '25', '14', '7', '3', '4', '6', '31', '22', '5', '24', '16', '8', '18', '12', '11', '1
5', '10', '1', '9', '2']
```

```
print(tarea)
```

```
[2492.72715468015, 1132.170216395556, 51.04531707516082, 799.5076939092202, 2225.03573186162, 2827.9892370365283, 155
5.8699645555637, 1688.6707319518882, 1576.0639305027323, 7068.152865339773, 2099.6821240524123, 32.40765831251095, 34
90.416072723304, 3318.6133789289916, 1406.0813872233591, 8171.776366782006, 5992.169759986729, 1215.5200458486174, 25
09.4530275753145, 1582.7272740554295, 989.631393290842, 2038.9888834066373, 1214.8188948522854, 3505.216898246742, 19
49.6909700659628]
```

```
data # Dictionary containing content of Boundaries.geojson
```

```
{'features': [{'geometry': {'coordinates': [[[[-87.71067089391354,
         41.997365655369435],
        [-87.71066884721016, 41.99729359357709],
        [-87.7106653080999, 41.997225765680135],
        [-87.7106465308153, 41.99711482608049],
        [-87.71063495001478, 41.99707637166497 4],
        [-87.71063126951208, 41.99703491309021],
        [-87.71063173430669, 41.99702111355831],
        [-87.71064107245383, 41.99701057121908],
        [-87.71064185160014, 41.99700613033399],
        [-87.7106289441022, 41.996790558671314],
        [-87.71062465241883, 41.996702446053355],
```

```
: data # Dictionary containing content of Boundaries.geojson
```

```
        [-87.71520773719217, 41.93924278726396],
        [-87.7158417568539, 41.939235167067004],
        ...]]],
      'type': 'MultiPolygon'},
     'properties': {'dist_label': '17TH', 'dist_num': '17'},
     'type': 'Feature'},
    {'geometry': {'coordinates': [[[[-87.66029423572358, 41.990916338539776],
        [-87.66029047841705, 41.99079098951434],
        [-87.66028253832803, 41.9905006613094],
        [-87.66027046884024, 41.99015194133787],
        [-87.66026648601695, 41.99000181125436],
        [-87.65999648827015, 41.99001704881578],
        [-87.65969970282563, 41.9900286731309],
        [-87.65950156451294, 41.990033433451025],
        [-87.65945009593533, 41.990034672050655],
        [-87.65926276620326, 41.990037113997715],
        [-87.65916228563397, 41.99003873890808],
        [-87.65900792917716, 41.99004052701838],
        [-87.6589430171194, 41.99004206251163],
        [-87.65891629200512, 41.990042694254065],
```

Creating the DataFrame

```
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['district_area_inHectares']/100))final_data.head()
```

```
af.head()
```

	dist_num	district_area_inHectares
0	17	2492.727155
1	20	1132.170216
2	31	51.045317
3	31	799.507694
4	19	2225.035732

```
crimes_per_district.head()
```

	dist_num	number_of_crimes
0	14	21793
1	25	31477
2	8	37275
3	12	29678
4	17	17165

```
final_data.head()   # merge the two DataFrames--equivalent of SQL join
```

	dist_num	district_area_inHectares	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

Query 5

Objective: 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)**

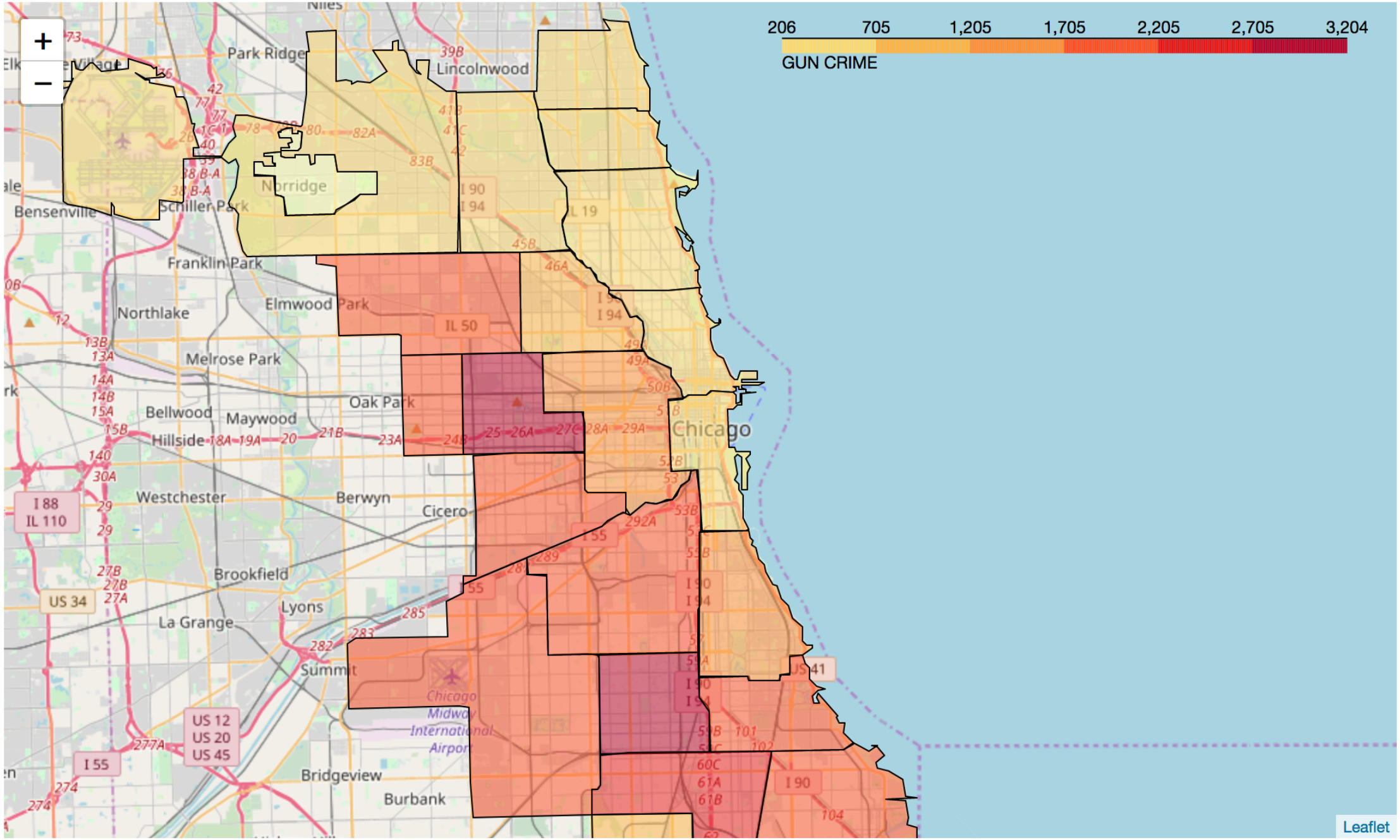
Create the choropleth map

Use **districts_gun_violent_crimes_df** from Query 3:

```
# base map
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"       ).
```

```
gun_crime_arrests_map
```



Getting the marker locations

cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), district from police_stations where district!='Headquarters'""")

'police_stations = cursor.fetchall()

Adding Markers

```
cursor.execute("""SELECT ST_X(ST_AsText(Where_IS)), ST_Y(ST_AsText(Where_IS)), district from police_stations where distr
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 fr
    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="District No: %s <br> Description: %s <br
        else:
            folium.Marker(location=(crime[5],crime[6]),popup=folium.Popup(html="District No: %s <br> Description: %s<br>
```

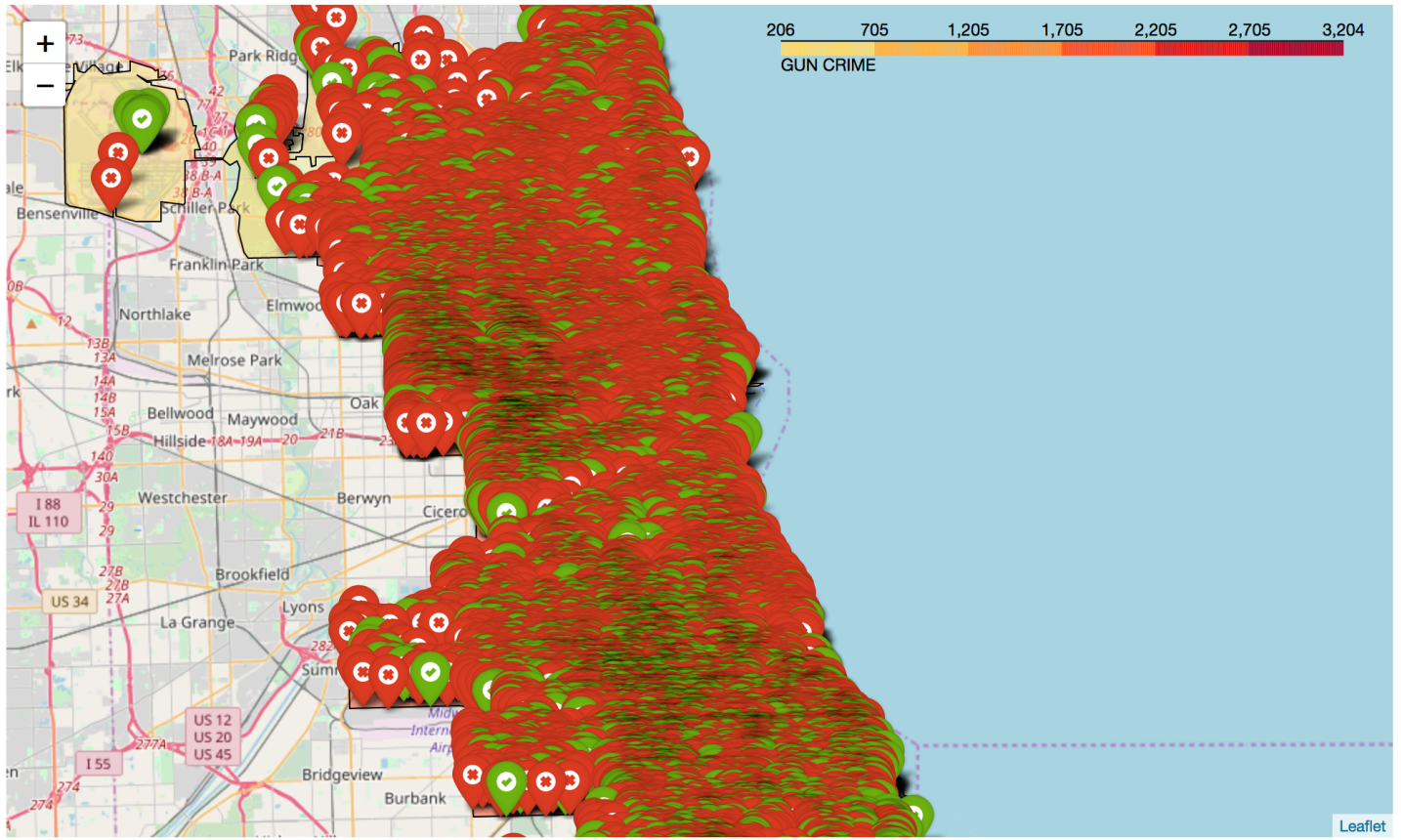
Marker Clusters

“Marker clusters can be a good way to simply a map containing many markers. When the map is zoomed out nearby markers are combined together into a cluster, which is separated out when the map zoom level is closer.”

https://deparkes.co.uk/2016/06/24/folium-marker-clusters/

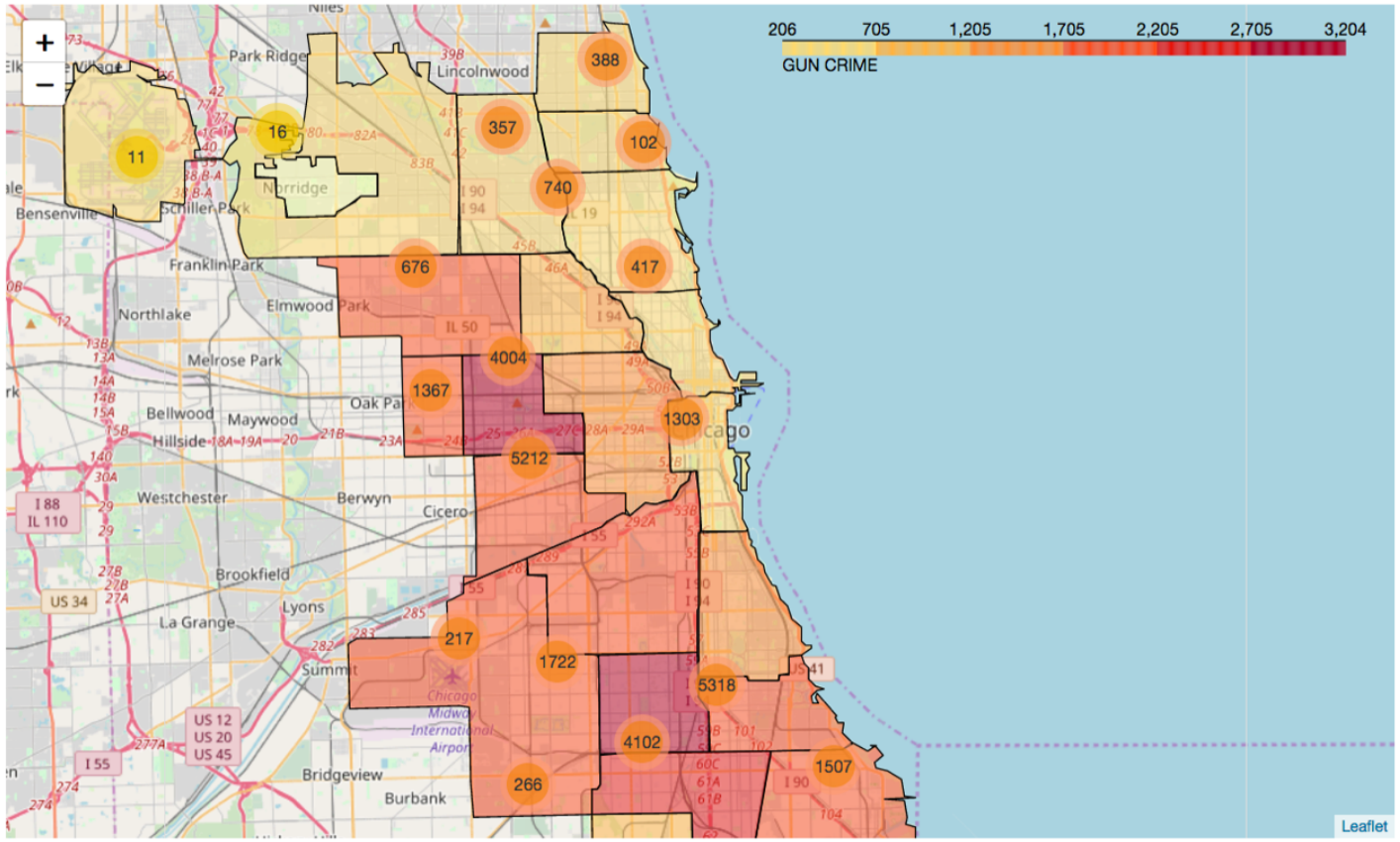
Adding markers directly to the map

```
gun_crime_arrests_map
```



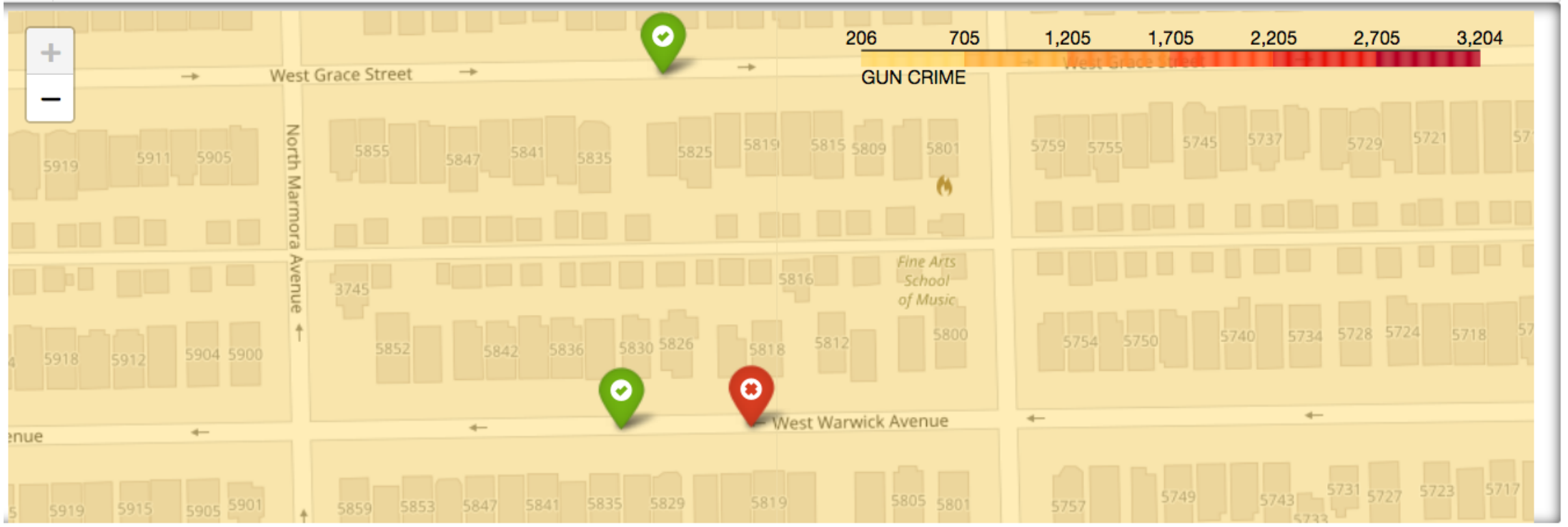
Adding markers to the marker cluster

```
1 gun_crime_arrests_map
```



Green and red marker icons

```
1 gun_crime_arrests_map
```



```
crimes_per_district  # caseno, block, DESCRIPTION, count(arrest), arrest, latitute, longitude
```

```
[('HZ100175',
  '041XX W HIRSCH ST',
  'AGGRAVATED: HANDGUN',
  1,
  False,
  41.906240189,
  -87.729678708),
 ('HZ102198',
  '027XX N OAK PARK AVE',
  'UNLAWFUL POSS OF HANDGUN',
  1,
  True,
  41.929964397,
  -87.795438924),
```

```
crime # caseno, block, DESCRIPTION, count(arrest), arrest, latitute, longitude
```

```
('JB208800',
 '028XX N LOTUS AVE',
 'GUN OFFENDER: ANNUAL REGISTRATION',
 1,
 False,
 41.931798069,
 -87.762579482)
```

Two different marker icons and color

```
if crime[4]==True:  # arrest was made
    folium.Marker(location=(crime[5],crime[6]),popup=folium.Popup(html="District No: %s <br>
        Description: %s <br> Block: %s" %(police_station[2],crime[2],crime[1])),
        icon=folium.Icon(color='green', icon='ok-sign'),).add_to(marker_cluster) \
else:   # arrest is false, i.e. arrest was not made
        folium.Marker(location=(crime[5],crime[6]),popup=folium.Popup(html="District No: %s
            Description: %s<br> Block: %s" %(police_station[2],crime[2],crime[1])),
            icon=folium.Icon(color='red', icon='remove-sign'),).add_to(marker_cluster)
```

Query 6

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

First create the choropleth...

```
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"
                )
```

Finding the furthest Block from HQ

```
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_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]])

farthest_block_gun_crime = cursor.fetchall()
```

First get the maximum distance

For a fixed district police station get all the all the distances to each block in the distance where there was a gun crime (according to the description of the crime).

```
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]])
```

Then get the largest of these distances:

```
SELECT max(dist) FROM (SELECT....)
```

The get the furthest block

Finally, find the blocks with these distances and use DISTINCT to pick the first (?) one in case of a tie....

```
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  ...
```

The query result data

We get the district number, block, Where_IS (for block), and distance from this block to the police headquarter in the district:

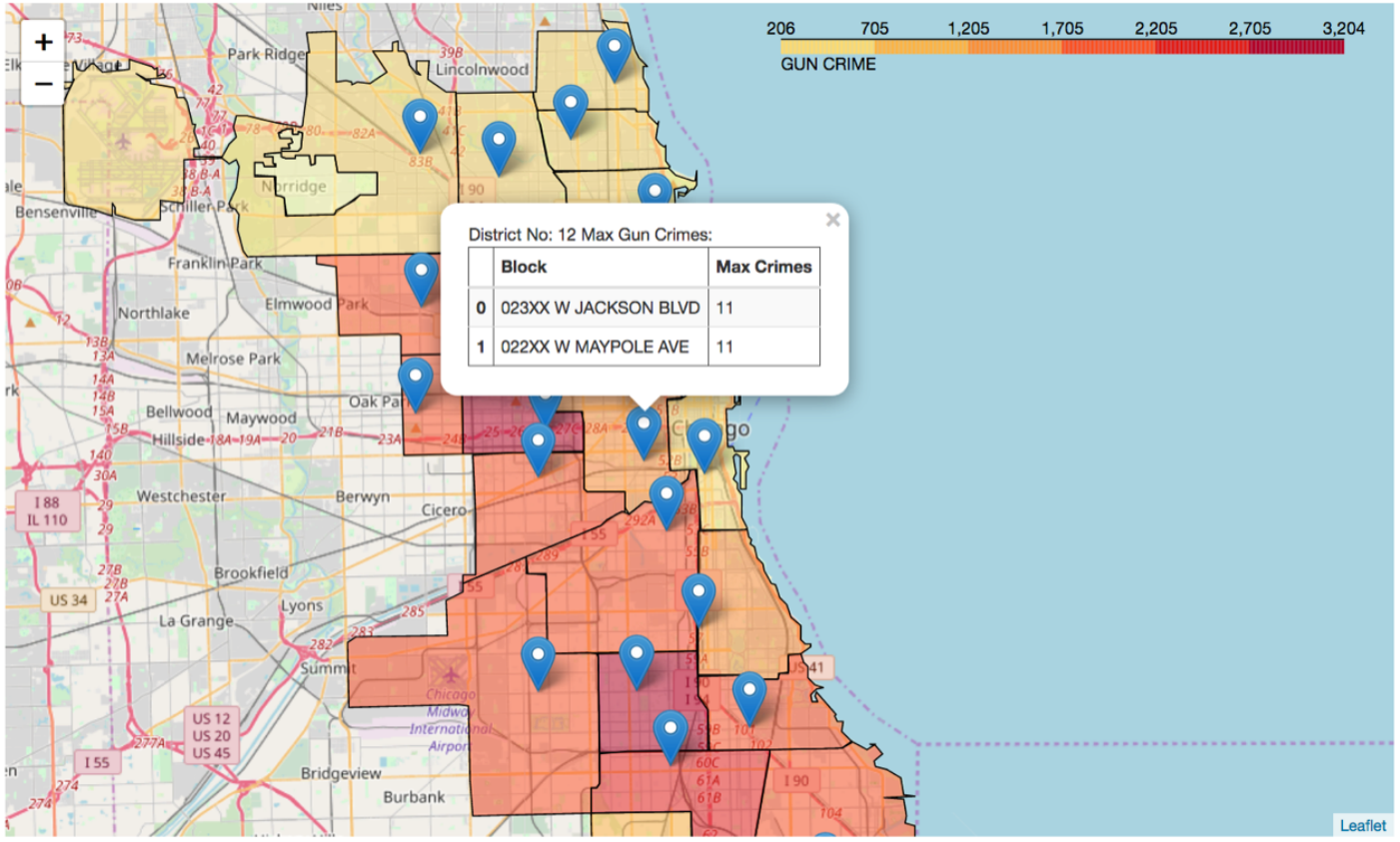
```
farthest_block_gun_crime
```

```
[(1,
  '030XX S FORT DEARBORN',
  '0101000020E61000006AB81C3D6BEB444095F2A1DEE8E655C0',
  2167.862477495)]
```

This information is used to locate the circles on the map at the blocks but also for popup info at leaflets and circles

Requirement 1 Output

```
1 districts_gun_crimes_map
```



Requirement 2 Sample Output

	dist_num	district_area_inHectares	gun_crimes	gun_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 Sample Output

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

Requirement 4 Sample Output

```
1 district_handgun_crime_map
```