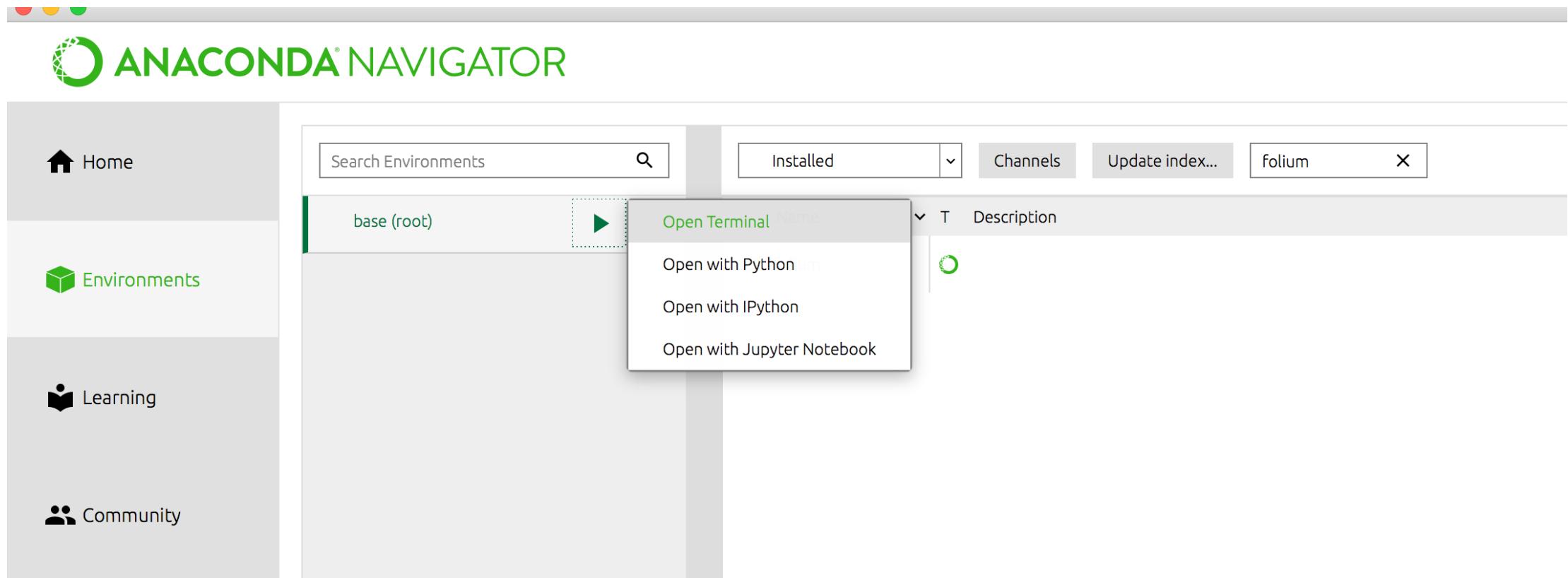


Assignment 3

PART 1

The Anaconda terminal



Installing folium

```
EdwardArroyo — a.tool — bash --init-file /dev/fd/63 — 94x12
Last login: Mon Jul 30 18:03:49 on ttys000
/Users/EdwardArroyo/.anaconda/navigator/a.tool ; exit;
vpn-165-124-161-146:~ EdwardArroyo$ /Users/EdwardArroyo/.anaconda/navigator/a.tool ; exit;
(base) bash-3.2$ conda install -c conda-forge folium
Solving environment: done

# All requested packages already installed.

(base) bash-3.2$ ]
```

Installing psycopg2

```
EdwardArroyo — a.tool — conda install -c conda-forge psycopg2 — 89x22
(base) bash-3.2$ conda install -c conda-forge psycopg2
Solving environment: done

## Package Plan ##

environment location: /anaconda3

added / updated specs:
- psycopg2

The following packages will be downloaded:

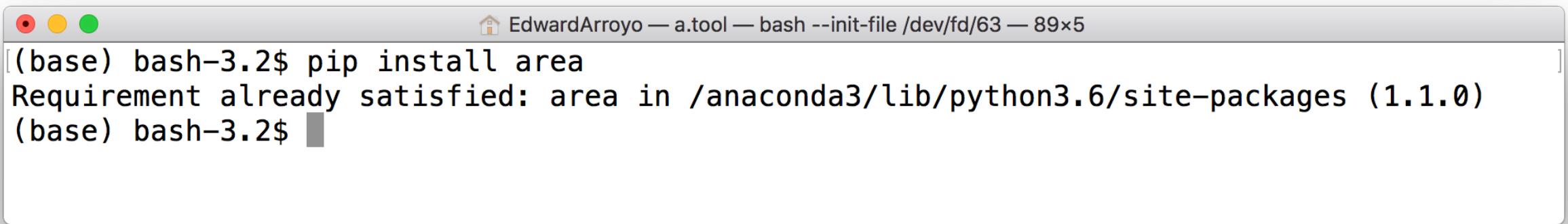
  package          |      build
  --              | -----
  psycopg2-2.7.5  | py36hdffb7b8_1    286 KB  conda-forge

The following packages will be UPDATED:

  psycopg2: 2.7.5-py36hdbc3d79_0 --> 2.7.5-py36hdffb7b8_1  conda-forge

Proceed ([y]/n)?
```

Installing the area package



A screenshot of a terminal window titled "EdwardArroyo — a.tool — bash --init-file /dev/fd/63 — 89x5". The window shows the command "[base] bash-3.2\$ pip install area" followed by the response "Requirement already satisfied: area in /anaconda3/lib/python3.6/site-packages (1.1.0)". The terminal has a standard OS X look with red, yellow, and green window control buttons.

```
[base] bash-3.2$ pip install area
Requirement already satisfied: area in /anaconda3/lib/python3.6/site-packages (1.1.0)
(base) bash-3.2$
```

The Crimes table

The *crimes* table has 587,075 rows and the following 23 columns..

```
['id', 'caseno', 'date_of_occurrence', 'block', 'iucr', 'primary_type', 'description',  
'location_description', 'arrest', 'domestic', 'beat', 'district', 'ward', 'community_area', 'fbi_cd',  
'xcoordinate', 'ycoordinate', 'year', 'updated_on', 'latitude', 'longitude', 'location', 'where_is']
```

Getting the column names and data types

```
query= "select column_name, data_type from information_schema.columns  
where table_name = 'crimes';"  
cursor.execute(query)  
rows=cursor.fetchall()  
print(rows)
```

Getting the column names and data types(2)

```
[('district', 'text'),
 ('district_name', 'text'),
 ('address', 'text'),
 ('city', 'text'),
 ('state', 'text'),
 ('zip', 'integer'),
 ('website', 'text'),
 ('phone', 'text'),
 ('fax', 'text'),
 ('tty', 'text'),
 ('x_coordinate', 'double precision'),
 ('y_coordinate', 'double precision'),
 ('latitude', 'double precision'),
 ('longitude', 'double precision'),
 ('location', 'text'),
 ('where_is', 'USER-DEFINED')]
```

Getting the first few rows...

```
crimes_cols =\['id', 'caseno', 'date_of_occurrence', 'block', 'iucr', 'primary_type', 'description',  
'location_description', 'arrest', 'domestic', 'beat', 'district', 'ward', 'community_area','fbi_cd',  
'xcoordinate', 'ycoordinate', 'year', 'updated_on', 'latitude', 'longitude', 'location', 'where_is']
```

```
cursor.execute("select * from crimes")  
rows=cursor.fetchall()  
crimes_table = pd.DataFrame(rows,columns=crimes_cols)  
crimes_table.head()
```

The first few rows

	<code>id</code>	<code>caseno</code>	<code>date_of_occurrence</code>	<code>block</code>	<code>iucr</code>	<code>primary_type</code>	<code>description</code>	<code>location_description</code>	<code>arrest</code>	<code>domestic</code>	<code>...</code>
0	11019838	JA350419	2017-07-16 14:51:00	072XX S LOOMIS BLVD	143A	WEAPONS VIOLATION	UNLAWFUL POSS OF HANDGUN	STREET	True	False	...
1	11020542	JA351222	2017-07-17 08:15:00	057XX S CICERO AVE	1478	CONCEALED CARRY LICENSE VIOLATION	PROHIBITED PLACES	AIRPORT TERMINAL UPPER LEVEL - SECURE AREA	True	False	...
2	11019560	JA350223	2017-07-16 12:00:00	035XX N KOSTNER AVE	1320	CRIMINAL DAMAGE	TO VEHICLE	STREET	False	False	...
3	11019369	JA349879	2017-07-16 03:30:00	067XX S LOOMIS BLVD	0420	BATTERY	AGGRAVATED:KNIFE/CUTTING INSTR	RESIDENCE PORCH/HALLWAY	False	False	...
4	11020293	JA351075	2017-07-17 02:30:00	078XX S JEFFERY BLVD	0560	ASSAULT	SIMPLE	RESIDENCE	True	True	...

....and the last few columns

	beat	district	ward	community_area	fbi_cd	xcoordinate	ycoordinate	year	updated_on	latitude	longitude	location
0	734	7	17.0	67	15	1168245.0	1856926.0	2017	2018-02-10 15:50:01	41.762936	-87.658914	(41.762935962, -87.658914303) 0101000020E6100000BCDDB6
1	813	8	23.0	56	15	1145654.0	1866253.0	2017	2018-02-10 15:50:01	41.788987	-87.741480	(41.788987036, -87.74147999) 0101000020E6100000434BF6
2	1731	17	30.0	16	14	1146426.0	1923123.0	2017	2018-02-10 15:50:01	41.945031	-87.737201	(41.945030728, -87.737201232) 0101000020E6100000CF3C5
3	724	7	17.0	67	04B	1168160.0	1859963.0	2017	2018-02-10 15:50:01	41.771272	-87.659139	(41.771271712, -87.659138593) 0101000020E61000005AAF0D
4	414	4	8.0	43	08A	1190896.0	1853388.0	2017	2018-02-10 15:50:01	41.752710	-87.576010	(41.752710067, -87.576009627) 0101000020E61000004691B0

Dealing with postgis

```
cursor.execute("select latitude, longitude, location, ST_AsText(Where_IS) from crimes")
rows=cursor.fetchall()
crimes_table2 = pd.DataFrame(rows,columns=['latitude', 'longitude', 'location', "Where_IS"])
crimes_table2.head()
```

https://postgis.net/docs/manual-2.4/ST_AsText.html

Dealing with postgis

	latitude	longitude	location	Where_IS
0	41.797818	-87.581919	(41.797818324, -87.581918691)	POINT(41.797818324 -87.581918691)
1	41.880612	-87.666782	(41.880612385, -87.666781505)	POINT(41.880612385 -87.666781505)
2	41.895547	-87.658487	(41.895547498, -87.658486918)	POINT(41.895547498 -87.658486918)
3	41.820254	-87.600389	(41.820254095, -87.600388638)	POINT(41.820254095 -87.600388638)
4	41.754593	-87.741529	(41.754592961, -87.741528537)	POINT(41.754592961 -87.741528537)

Frequency of primary type

THEFT	137593
BATTERY	110240
CRIMINAL DAMAGE	65857
ASSAULT	42366
OTHER OFFENSE	38017
DECEPTIVE PRACTICE	37762
BURGLARY	29640
NARCOTICS	27687
ROBBERY	26126
MOTOR VEHICLE THEFT	25072
CRIMINAL TRESPASS	14829
WEAPONS VIOLATION	9246
OFFENSE INVOLVING CHILDREN	4908
PUBLIC PEACE VIOLATION	3383
CRIM SEXUAL ASSAULT	3351
INTERFERENCE WITH PUBLIC OFFICER	2320
SEX OFFENSE	2053
PROSTITUTION	1622
HOMICIDE	1520
ARSON	1028
LIQUOR LAW VIOLATION	471

Frequency of primary type (2)

KIDNAPPING	439
GAMBLING	396
STALKING	391
INTIMIDATION	311
OBScenity	158
CONCEALED CARRY LICENSE VIOLATION	122
NON-CRIMINAL	101
HUMAN TRAFFICKING	23
PUBLIC INDECENCY	20
OTHER NARCOTIC VIOLATION	15
NON - CRIMINAL	5
NON-CRIMINAL (SUBJECT SPECIFIED)	3

The police_stations table

The *police_stations* has 23 rows and the following 16 columns:

```
['district', 'district_name', 'address', 'city', 'state', 'zip', 'website', 'phone', 'fax', 'tty',  
'x_ccordinate', 'y_coordinate', 'latitude', 'longitude', 'location', 'where_is']
```

Note that *where_is* has a 'USER-DEFINED' type.

The police_stations table

```
query=\n"select district,x_coordinate,y_coordinate,latitude,longitude,location,where_is from police_stations"\n\ncursor.execute(query)\nrows=cursor.fetchall()\npst_table2 = pd.DataFrame(rows,columns=\n['district', 'x_coordinate','y_coordinate','latitude','longitude','location','where_is'])\npst_table2.head()
```

The police_stations table

	district	x_coordinate	y_coordinate	latitude	longitude	location	where_is
0	1	1176569.052	1891771.704	41.858373	-87.627356	(41.8583725929, -87.627356171)	0101000020E61000009AEA2C27DFED444001467E9A26E8...
1	2	1175864.837	1871153.753	41.801811	-87.630560	(41.8018110912, -87.6305601801)	0101000020E6100000608FECBEA1E64440A2D015195BE8...
2	3	1182739.183	1858317.732	41.766431	-87.605748	(41.7664308925, -87.6057478606)	0101000020E6100000DB984B681AE244409B14AC92C4E6...
3	4	1193131.299	1837090.265	41.707933	-87.568349	(41.7079332906, -87.5683491228)	0101000020E61000005C26DC8E9DDA444010C7FCD45FE4...
4	5	1183305.427	1831462.313	41.692723	-87.604506	(41.6927233639, -87.6045058667)	0101000020E61000000830B828ABD84440F6786339B0E6...

The police_stations table

```
query="select  
district,x_coordinate,y_coordinate,latitude,longitude,location,\ST_X(ST_ASTEXT(where_is)),ST_Y(S  
T_ASTEXT(where_is))\from police_stations"cursor.execute(query)rows=cursor.fetchall()pst_table3  
= pd.DataFrame(rows,columns=['district',  
'x_coordiate','y_coordinate','latitude','longitude','location','x','y'])  
pst_table3.head()
```

The police_station table

	district	x_coordinate	y_coordinate	latitude	longitude	location	x	y
0	1	1176569.052	1891771.704	41.858373	-87.627356	(41.8583725929, -87.627356171)	41.858373	-87.627356
1	2	1175864.837	1871153.753	41.801811	-87.630560	(41.8018110912, -87.6305601801)	41.801811	-87.630560
2	3	1182739.183	1858317.732	41.766431	-87.605748	(41.7664308925, -87.6057478606)	41.766431	-87.605748
3	4	1193131.299	1837090.265	41.707933	-87.568349	(41.7079332906, -87.5683491228)	41.707933	-87.568349
4	5	1183305.427	1831462.313	41.692723	-87.604506	(41.6927233639, -87.6045058667)	41.692723	-87.604506

Query 1 simplification

Objective: Calculate the total number of crimes in every district and plot that on Choropleth map.

```
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()

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()

# don't need a for loop since districts_crime_numbers only contains one ordered pair
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(total_number_of_crimes_per_district_map)
    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(total_number_of_crimes_per_district_map)
```

Query 1 simplification(2)

Objective: Calculate the total number of crimes in every district and plot that on Choropleth map.

```
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()
```

```
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() # we just need districts_crime_numbers[0]
```

```
folium.Marker(location =
    police_station_location,popup=folium.Popup(html="District No : %s has Total Number of Crimes:%s"
    % districts_crime_numbers[0], max_width=450)).add_to(total_number_of_crimes_per_district_map)
```

Getting total crimes per district

Objective: Calculate the total number of crimes in every district and plot that on Choropleth map.

The following select statement gets the total number of crimes per district. In the previous page we used this query (with a WHERE clause) to get the data for just one district but for now lets load the data for all districts into a DataFrame.

```
cursor.execute("SELECT district, count(district) from crimes GROUP BY district")
```

```
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()
```

Total Crime per District

dist_num	number_of_crimes
0	14
1	25
2	12
3	8
4	17

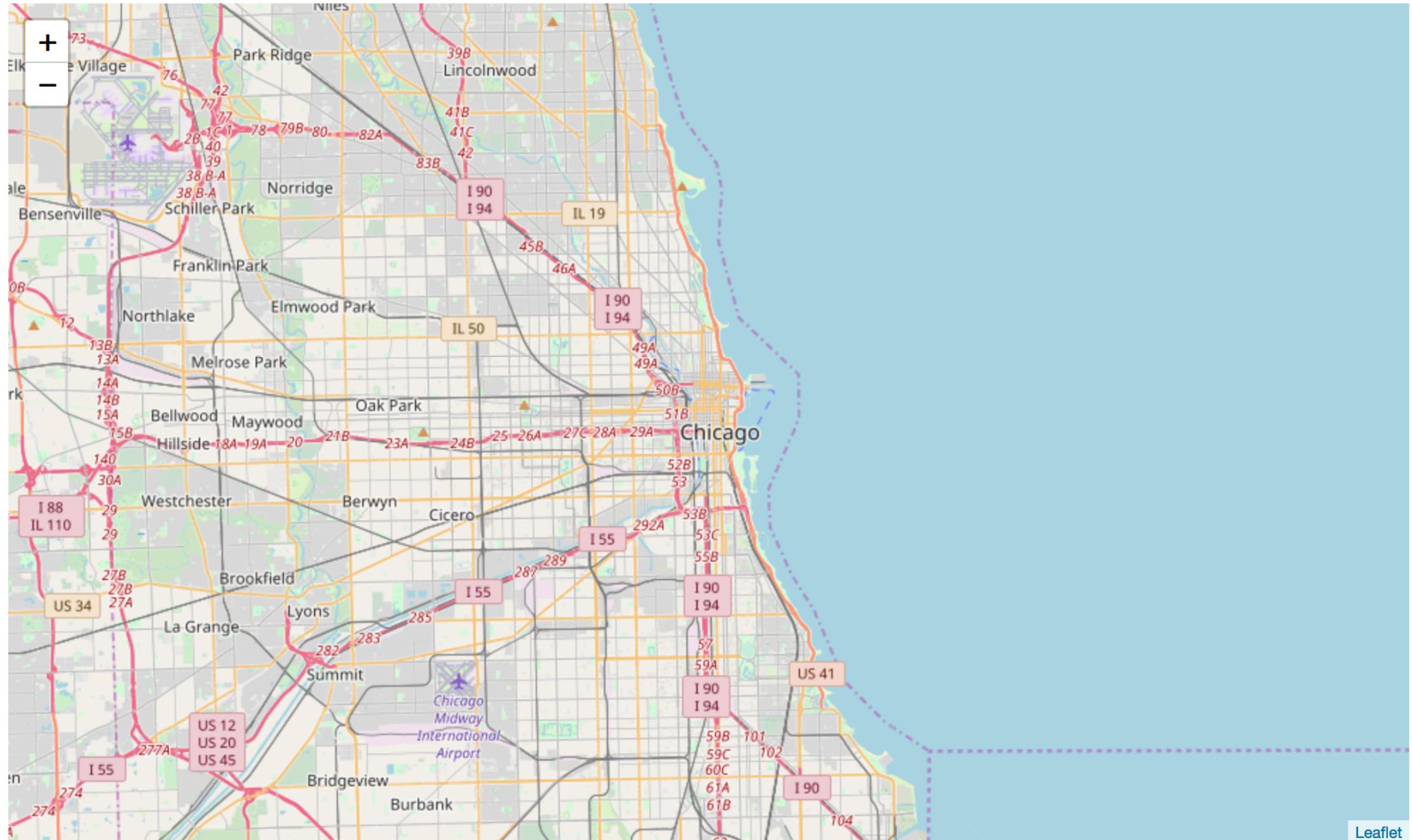
Creating the base map

We first create a base to which we add a layer with the district information...

```
# https://python-visualization.github.io/folium/modules.html
total_number_of_crimes_per_district_map = \
folium.Map(location =(41.8781, -87.6298), zoom_start=11)
```

Note: Try Googling “Chicago coordinates”

total_number_of_crimes_per_district_map

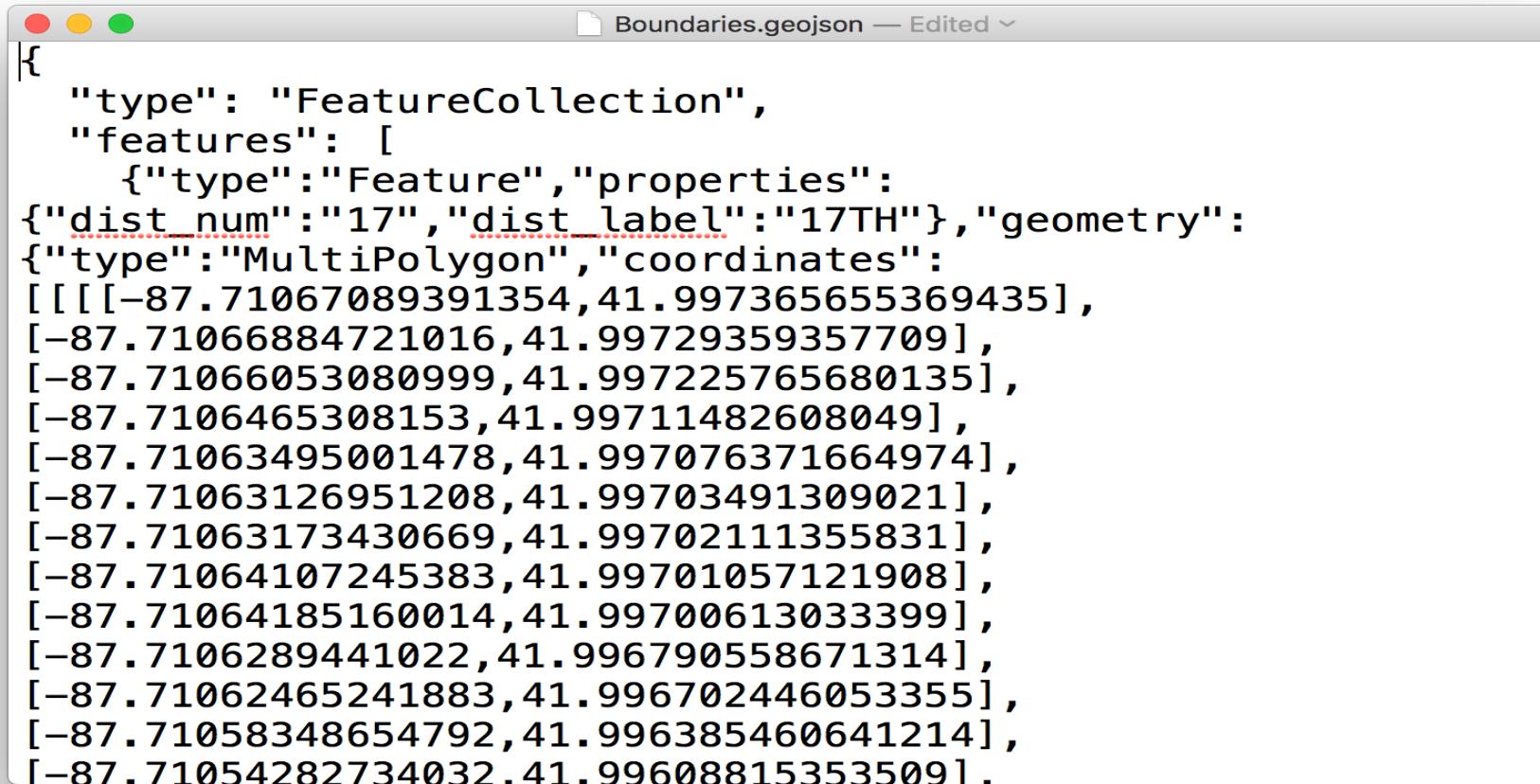


Plot a GeoJSON overlay on the base map

Use `Boundaries.geojson` to get the (coordinates of the) boundaries of each district and `crimes_per_district` to determine the `number_of_crimes` for each district number '`dist_num` (represented by intensity of color for each district of the choropleth map).

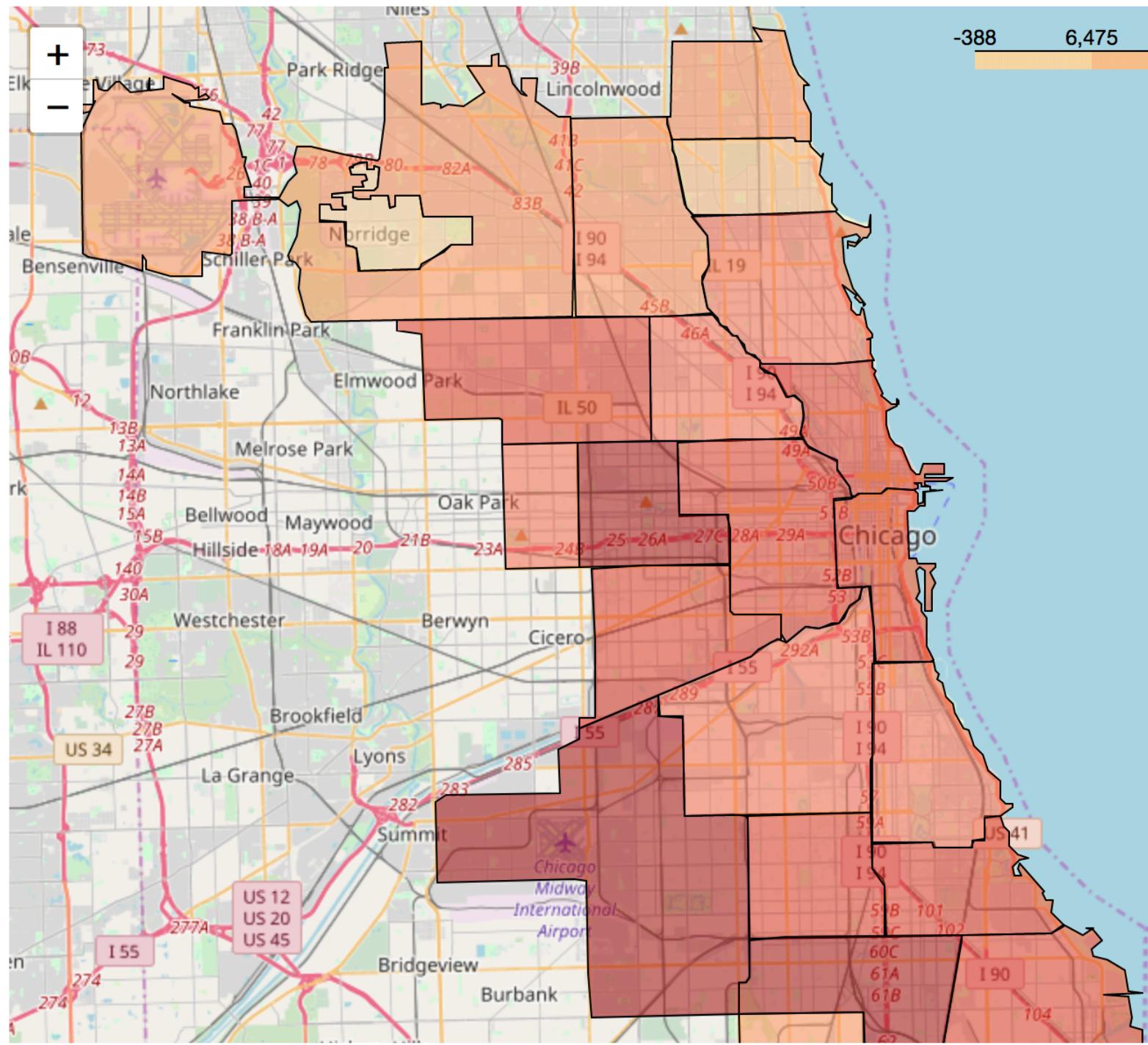
```
# https://python-visualization.github.io/folium/modules.html#choropleth
total_number_of_crimes_per_district_map.choropleth(geo_data="Boundaries.geojson",
    fill_color='OrRd',
    fill_opacity=0.5,
    line_opacity=1,
    data = crimes_per_district,
    key_on='feature.properties.dist_num',
    columns = ['dist_num', 'number_of_crimes'] )
```

Boundaries.geojson



A screenshot of a Mac OS X desktop environment showing a text editor window. The window title is "Boundaries.geojson — Edited". The content of the file is a JSON object representing a FeatureCollection with one feature, which is a MultiPolygon. The geometry consists of approximately 14 coordinates, each represented by a pair of longitude and latitude values.

```
{
  "type": "FeatureCollection",
  "features": [
    {"type": "Feature", "properties": {
      "dist_num": "17", "dist_label": "17TH"}, "geometry": {
        "type": "MultiPolygon", "coordinates": [
          [[[[-87.71067089391354, 41.997365655369435], [-87.71066884721016, 41.99729359357709], [-87.71066053080999, 41.997225765680135], [-87.7106465308153, 41.99711482608049], [-87.71063495001478, 41.997076371664974], [-87.71063126951208, 41.99703491309021], [-87.71063173430669, 41.99702111355831], [-87.71064107245383, 41.99701057121908], [-87.71064185160014, 41.99700613033399], [-87.7106289441022, 41.996790558671314], [-87.71062465241883, 41.996702446053355], [-87.71058348654792, 41.996385460641214], [-87.71054282734032, 41.996088153535091]]]}]}]
```



-388 6,475 13,338 20,201 27,064 33,927 40,790

Getting coordinates of the police stations

Next we want to create a marker at the location of each police station (with the crime data as well.) So we need to first get the location (coordinates) of each district police station.

```
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()
```

For example, here are the coordinates of first police district:

```
police_stations[0]  
(41.85837259, -87.62735617, '1')
```

Finding number of crimes per district

```
# Note that there is one police station per district so each districts_crime_numbers is data for a single district.  
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()  
....
```

To repeat each districts_crime_numbers is list consisting of a single ordered pair with coordinates the *district* and the. *total # of crimes* for a particular district. These values are inserted into the select statement. (<https://pyformat.info/>).

Note: Here police_station[2] is the current district number for the current police station (in the for loop).

Adding district markers to the map

for police_station in police_stations:

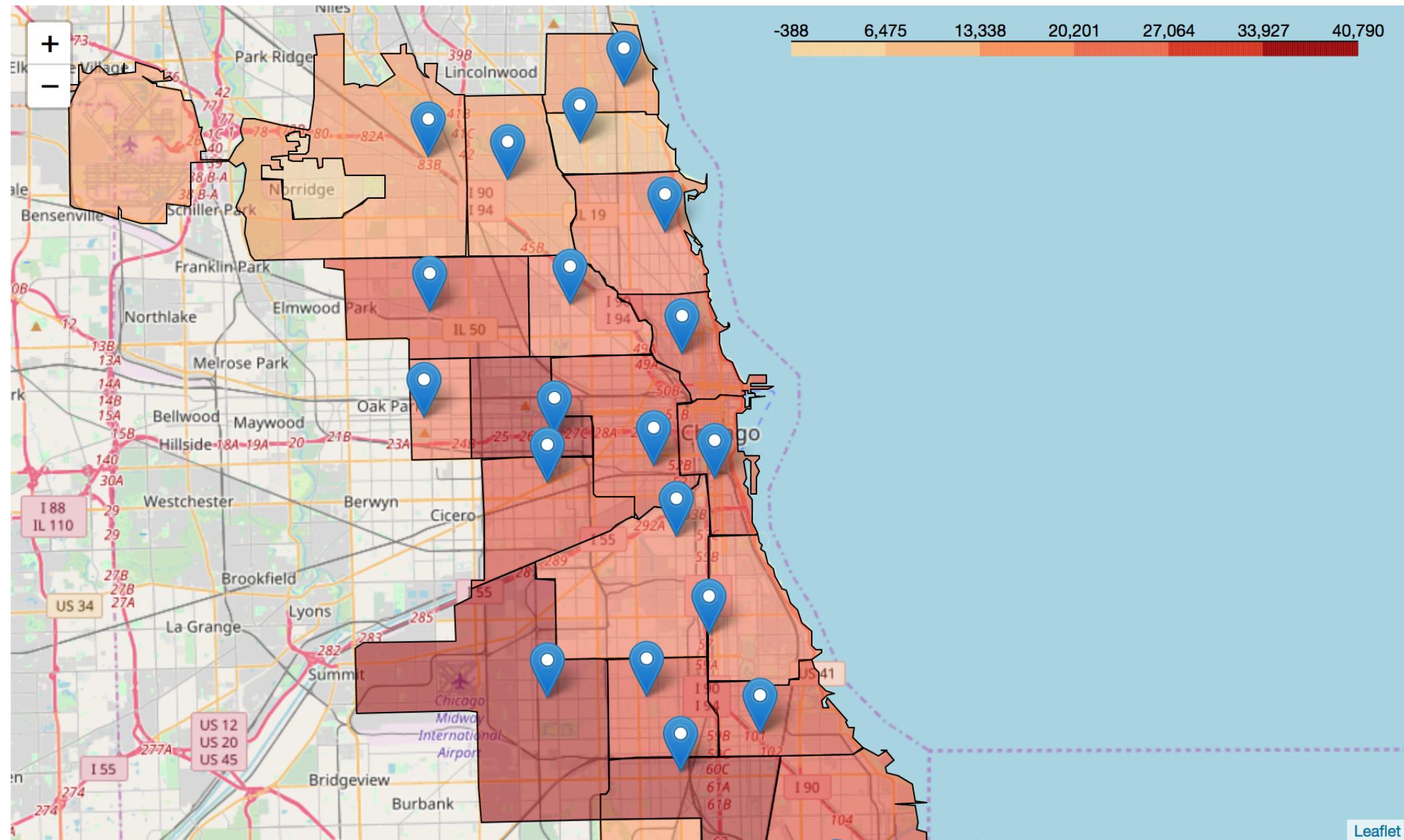
.....

```
# You don't really need a for loop since the districts_crime_numbers list contains just one ordered pair  
# 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" % districts_crime_numbers[0]  
max_width=450)).add_to(total_number_of_crimes_per_district_map)
```

Note: This works because *district* is an ordered pair consisting of the district # and the # of crimes in that district.

total_number_of_crimes_per_district_map



Data for marker popup

Data for *last* district marker generated in the “for police_station in police_stations” loop:

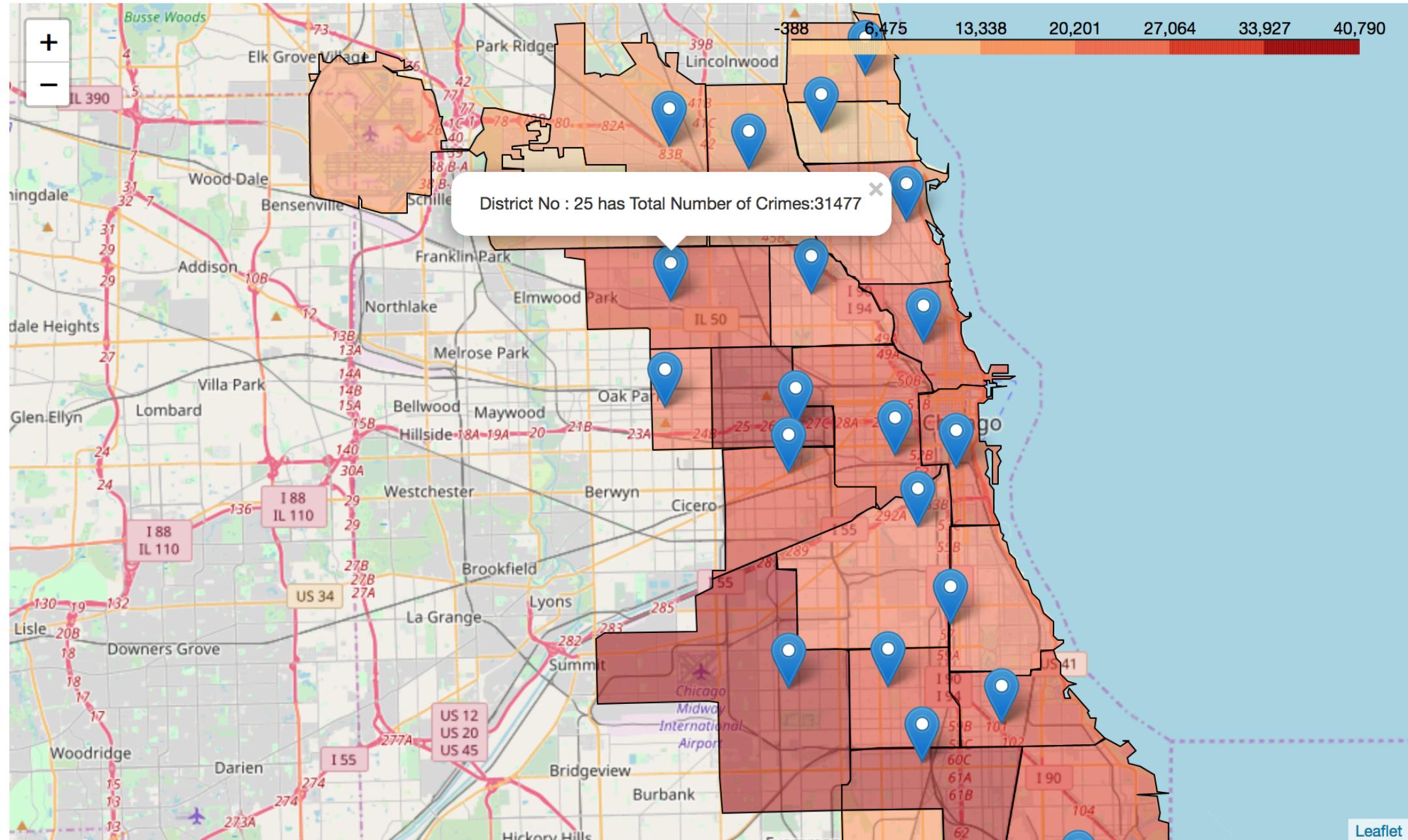
```
districts_crime_numbers
```

```
[ (25, 31477) ]
```

```
districts_crime_numbers[0]
```

```
(25, 31477)
```

total_number_of_crimes_per_district_map # Marker for District No. 25



Query 2

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

Number of violent crimes by district

```
violent_crime_categories='THEFT','ASSAULT','ROBBERY','KIDNAPPING','CRIM SEXUAL  
ASSAULT','BATTERY','MURDER' # This is a tuple
```

```
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_crimes'])  
violent_crime_data['district_num'] = violent_crime_data['district_num'].astype(str)  
violent_crime_data
```

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

Query 2 (cont'd)

```
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 ])
```

```
...
```

Here we select the rows where the value of PRIMARY_TYPE is in the violent_crime_categories and find the total number of crimes for each crime type.

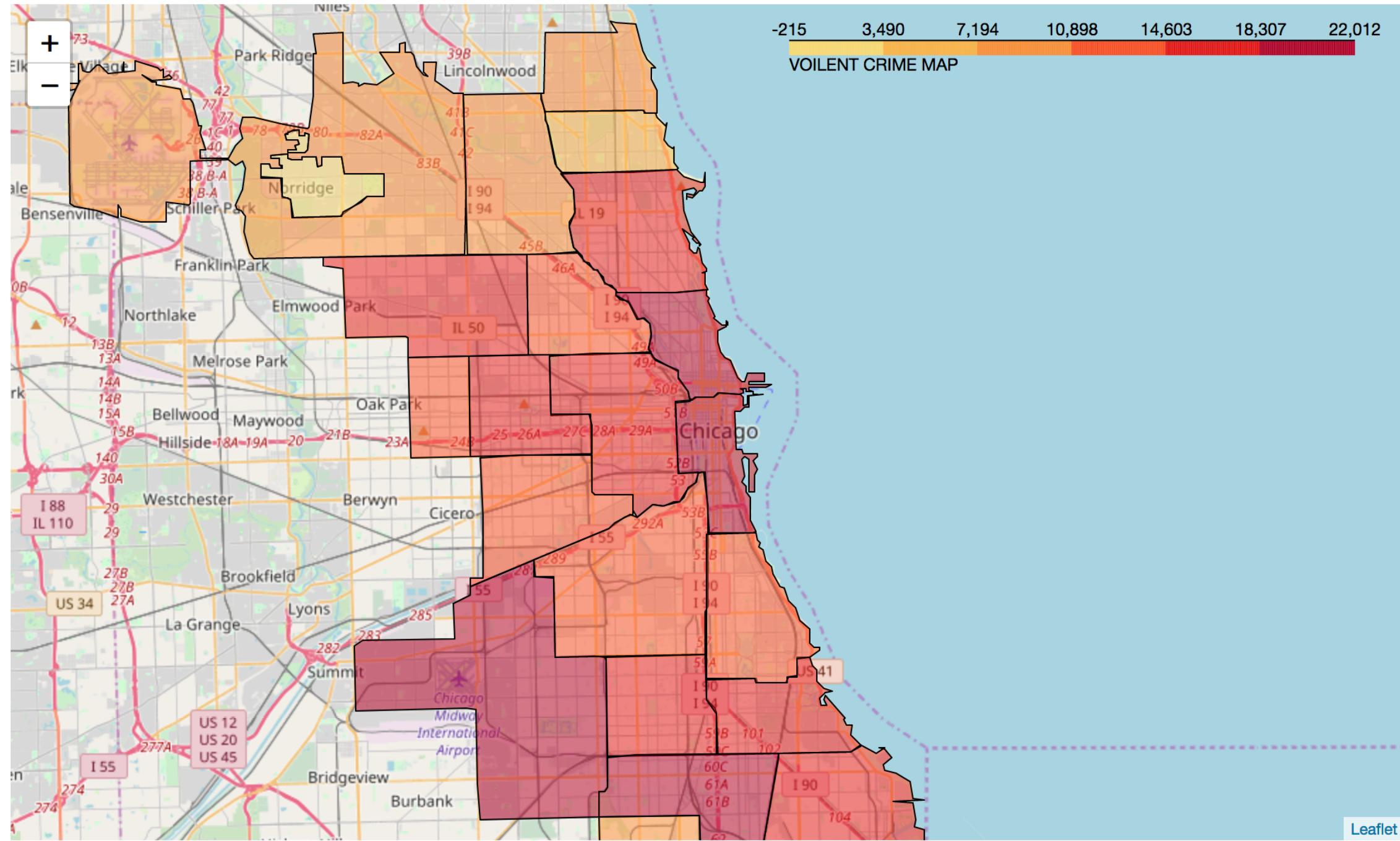
Plot a GeoJSON overlay on the base map

But this time the color intensity represents the number of violent crime (not crimes in general).

```
violent_crimes_per_district_map= folium.Map(location =(41.8781, -87.6298),zoom_start=11)
```

```
violent_crimes_per_district_map.choropleth(geo_data="Boundaries.geojson",
fill_color='YIOrRd', 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" )
```

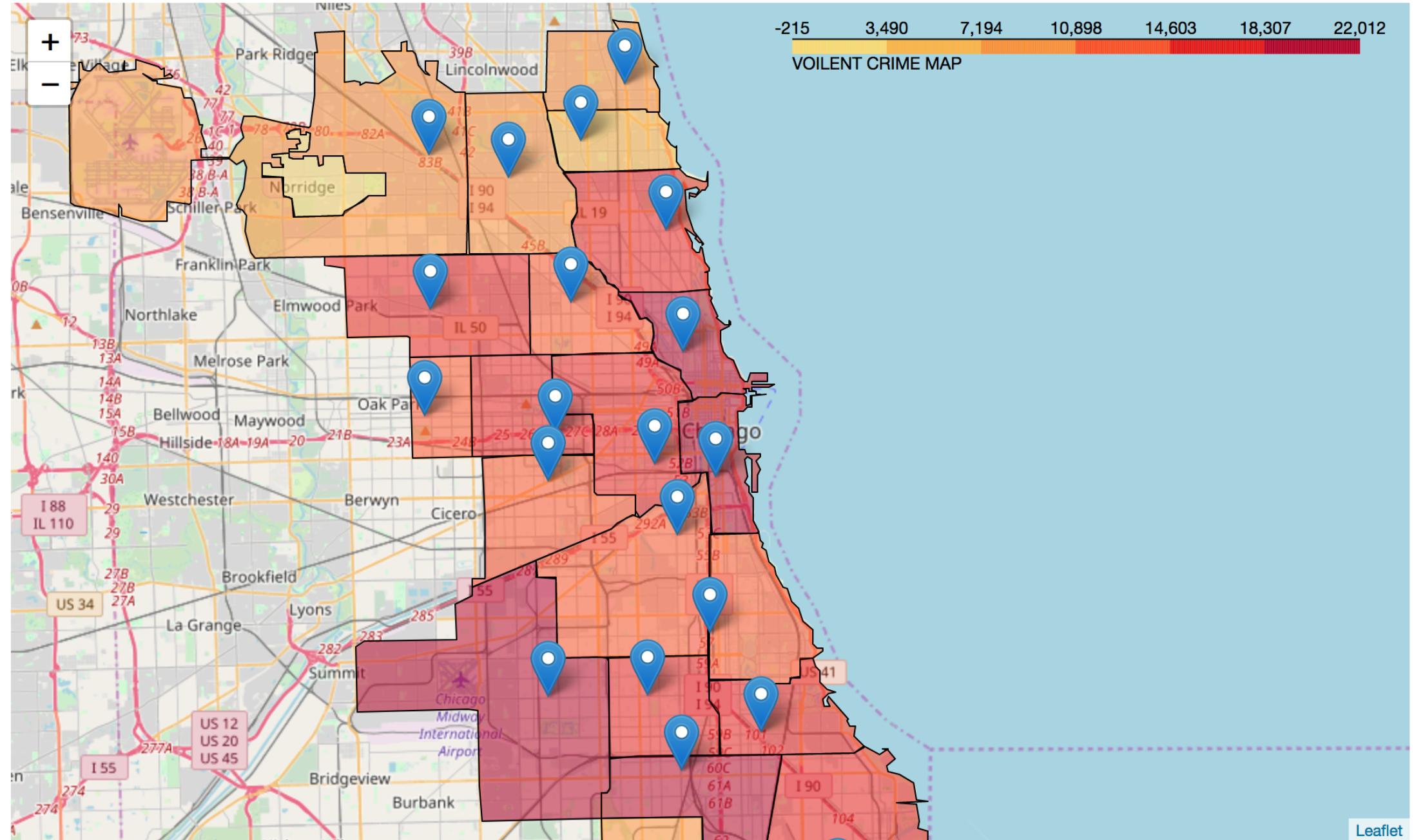
violent_crimes_per_district_map



Adding markers to the map

```
for police_station in police_stations:  
    police_station_location =(police_station[0],police_station[1])  
  
    # get break down of crimes in the current district by PRIMARY_TYPE  
    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()  
  
    violent_crimes_per_district_df = pd.DataFrame(data, columns=['Description', 'Number of Violent Crimes'])  
  
    # this time we add a DataFrame (instead of ordered pair) to each marker but first convert it to html...  
    header = violent_crimes_per_district_df.to_html(classes='table table-striped table-hover table-  
    condensed table-responsive')  
    folium.Marker(location=police_station_location, popup=folium.Popup(html="District Number %s -  
    Violent Crimes %s" %(police_station[2],header))).add_to(violent_crimes_per_district_map)
```

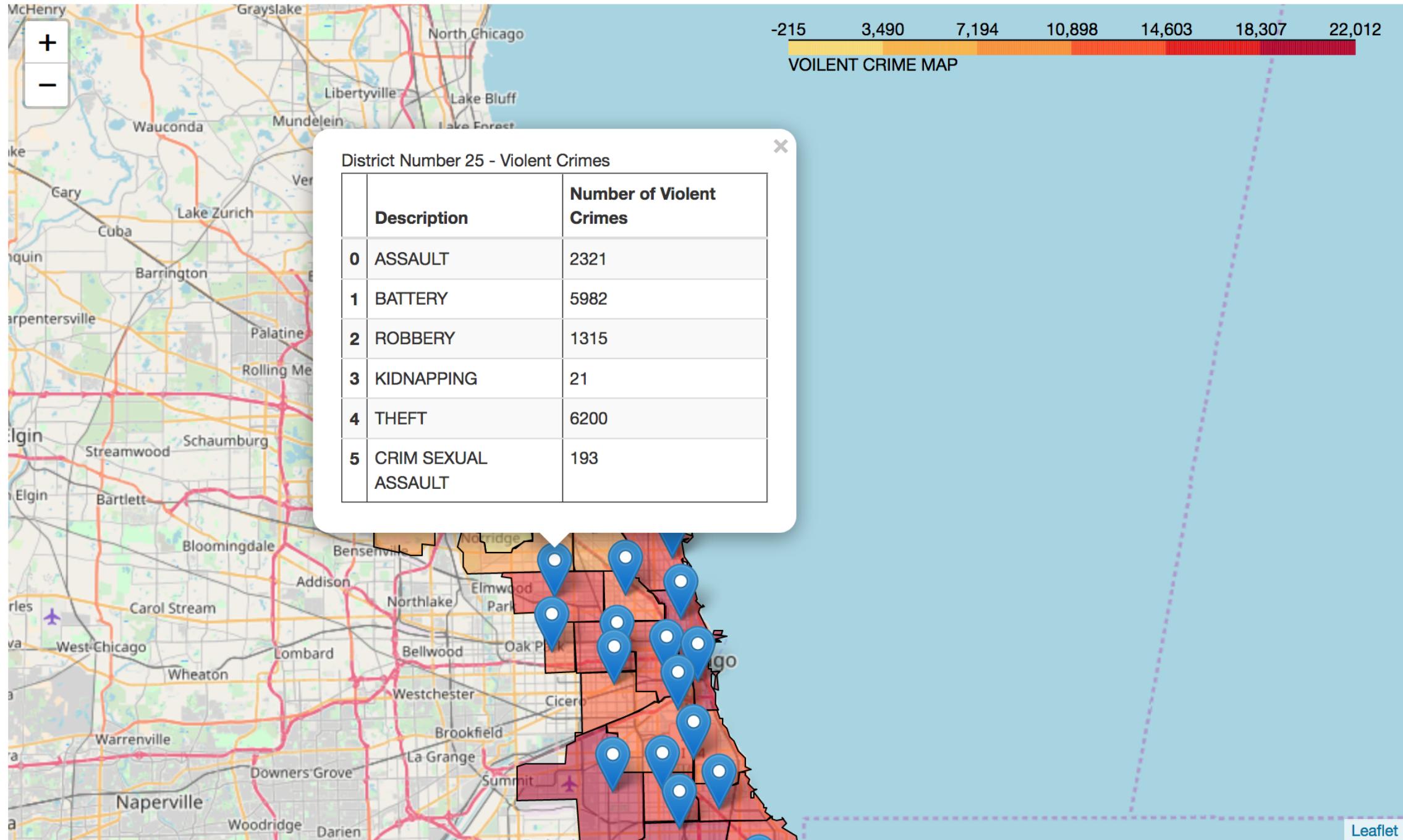
violent_crimes_per_district_map



```
violent_crimes_per_district_df # Last DataFrame in the loop (for District Number 25)|
```

	Description	Number of Violent Crimes
0	ASSAULT	2321
1	BATTERY	5982
2	ROBBERY	1315
3	KIDNAPPING	21
4	THEFT	6200
5	CRIM SEXUAL ASSAULT	193

```
violent_crimes_per_district_map # Last DataFrame in the loop (for District Number 25)
```



Query 3

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

The only substantive difference between this and the last is that we start with the DataFrame containing the total number of violent crimes per distance instead of the total number of crimes per district defined by this tuple

```
violent_crime_categories='THEFT','ASSAULT','ROBBERY','KIDNAPPING','CRIM SEXUAL  
ASSAULT','BATTERY','MURDER'
```

Query 3

First we get the number of gun crimes per district to generate the choropleth map...

```
gun=%GUN%
```

```
cursor.execute("SELECT district, count(district) from crimes where DESCRIPTION::text LIKE %s GROUP BY district", [gun])
```

```
districts_gun_violent_crimes = cursor.fetchall()
```

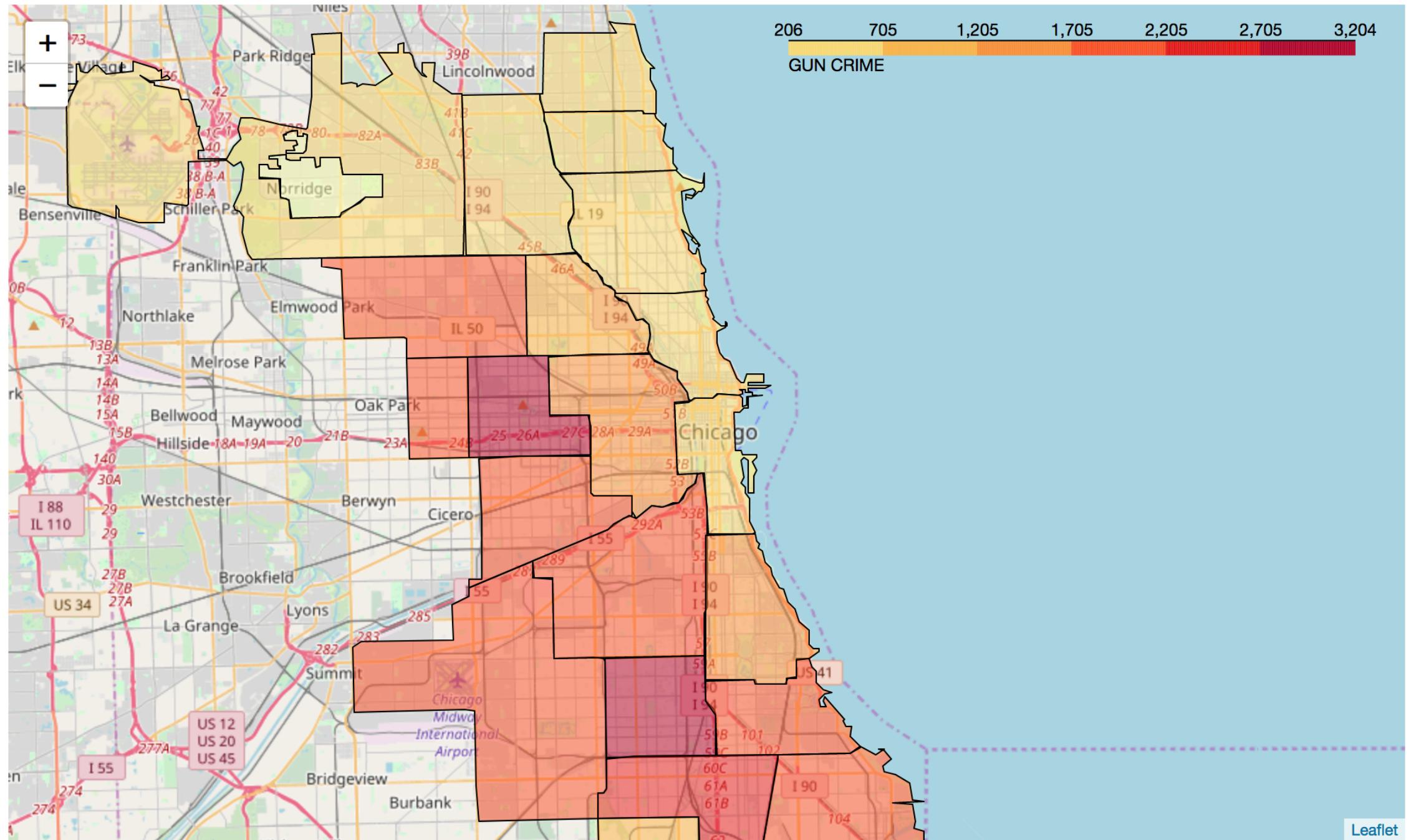
```
districts_gun_violent_crimes_df  
= pd.DataFrame(districts_gun_violent_crimes,  
columns=['dist_num','gun_crimes'])districts_gun_violent_crimes_
```

```
df['dist_num'] = districts_gun_violent_crimes_df['dist_num'].astype(str)  
districts_gun_violent_crimes_df
```

dist_num	gun_crimes
0	14
1	25
2	12
3	8
4	17
5	15
6	1
7	10
8	3
9	11
10	4
11	18
12	20
13	5
14	22
15	9
16	7
17	24
18	16
19	6
20	19
21	2
	1348

Plot a GeoJSON overlay on the base map

districts_gun_violent_crimes_map



Adding the markers(1)

```
# get the locations for the markers as usual
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()

gun='%GUN%'

# For each district get the total number of crimes that include "gun" as part of the discription
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 %
GROUP BY DESCRIPTION""",[police_station[2],gun])

district_gun_violent_crimes=cursor.fetchall()

.....
```

Adding the markers(2)

```
for police_station in police_stations:
```

```
.....
```

```
# create the frequency DataFrame for gun-related crimes
```

```
district_gun_violent_crimes_df=pd.DataFrame(district_gun_violent_crimes,  
                                              columns=['Description', 'Number of Gun Crime'])
```

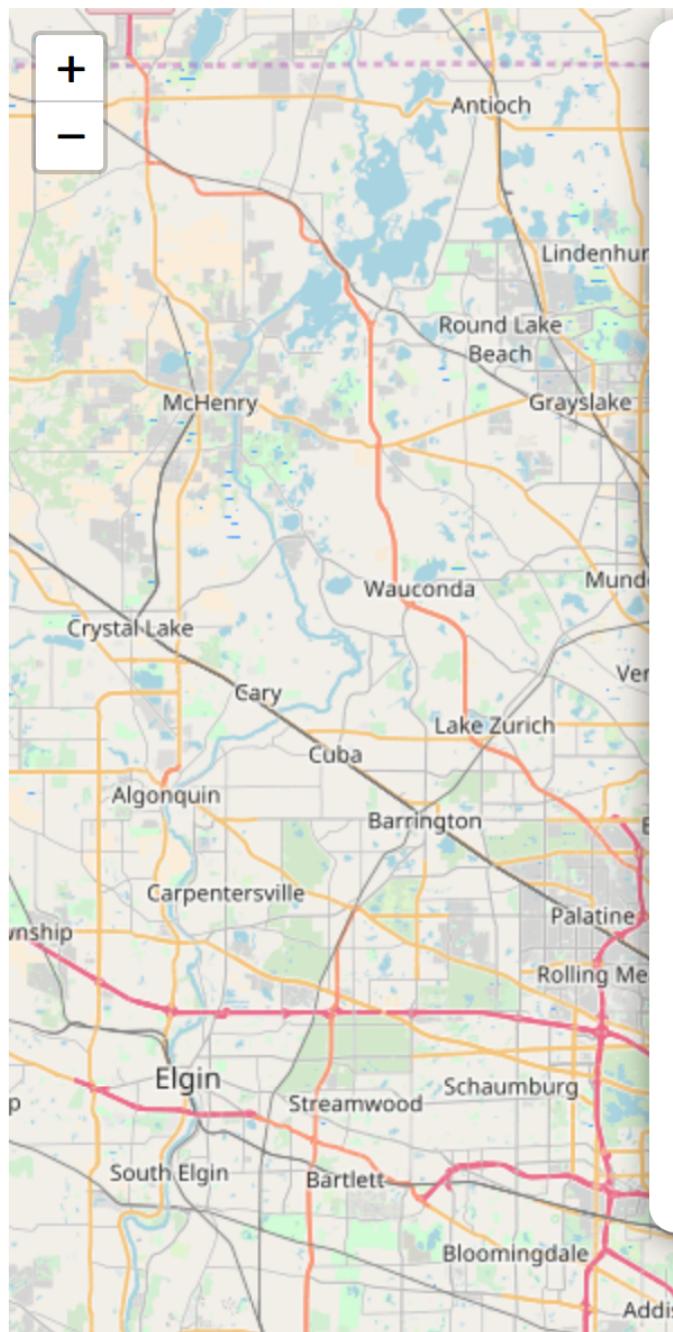
```
# convert the DataFrame to html
```

```
header = district_gun_violent_crimes_df.to_html(classes='table table-striped table-hover table-condensed table-responsive')
```

```
# add the marker
```

```
folium.Marker(location=police_station_location, popup=folium.Popup(html="District No: %s  
GUN_Crime: %s" %(police_station[2],header))).add_to(districts_gun_violent_crimes_map)
```

districts_gun_violent_crimes_map



District No: 25 GUN_Crime:

206 705 1,205 1,705 2,205 2,705 3,204

	Description	GUN CRIME Number of Gun Crime
0	ARMED: HANDGUN	525
1	GUN OFFENDER: DUTY TO REGISTER	11
2	UNLAWFUL USE HANDGUN	68
3	AGGRAVATED: HANDGUN	741
4	UNLAWFUL SALE HANDGUN	1
5	ATTEMPT: ARMED-HANDGUN	48
6	UNLAWFUL POSS OF HANDGUN	321
7	GUN OFFENDER: ANNUAL REGISTRATION	10
8	AGG PRO.EMP: HANDGUN	2
9	AGGRAVATED PO: HANDGUN	7
10	GUN OFFENDER: DUTY TO REPORT CHANGE OF INFORMATION	3
11	AGGRAVATED DOMESTIC BATTERY: HANDGUN	1

