lab7

May 16, 2018

1 Lab 7

First we import data from the City of Chicago Data Portal's API.

In [1]:
```
import pandas as pd
import json
import requests
from urllib.parse import quote
%matplotlib inline

# Get crime data
soql = "https://data.cityofchicago.org/resource/6zsd-86xi.json?$query="
soql += quote("SELECT community_area AS Community, count(*)/5 AS No_Crimes, primary_ty
soql += quote("WHERE 2007 < year AND year < 2013 ")
soql += quote("GROUP BY Community, Primary_Type LIMIT 10000")

resp = requests.get(soql).json()

crime_type = pd.DataFrame(resp)
```

In [2]:
```
# Get rid of the redundant decimal places
crime_type.No_Crimes = crime_type.No_Crimes.astype(float)
crime_type.round(1)

crime_type.dropna(axis = 0, how = "any", inplace = True)
```

In [3]:
```
# Get socioeconomics data
socioeconomics = pd.read_csv("./../data/Census_Data_-_Selected_socioeconomic_indicator
# rename "community"
crime_type.rename(columns={'Community':'Community Area Number'}, inplace = True)
crime_type['Community Area Number'] = crime_type['Community Area Number'].astype(float
# crime_type.head(15)
```

In [4]:
```
# Merging two dataset
crime_df = pd.DataFrame.merge(crime_type, socioeconomics, on = 'Community Area Number'
```

In [5]:
```
# geopandas starts from here

import matplotlib.pyplot as plt
import geopandas as gpd
```

Importing Chicago Community Area's Shapefile

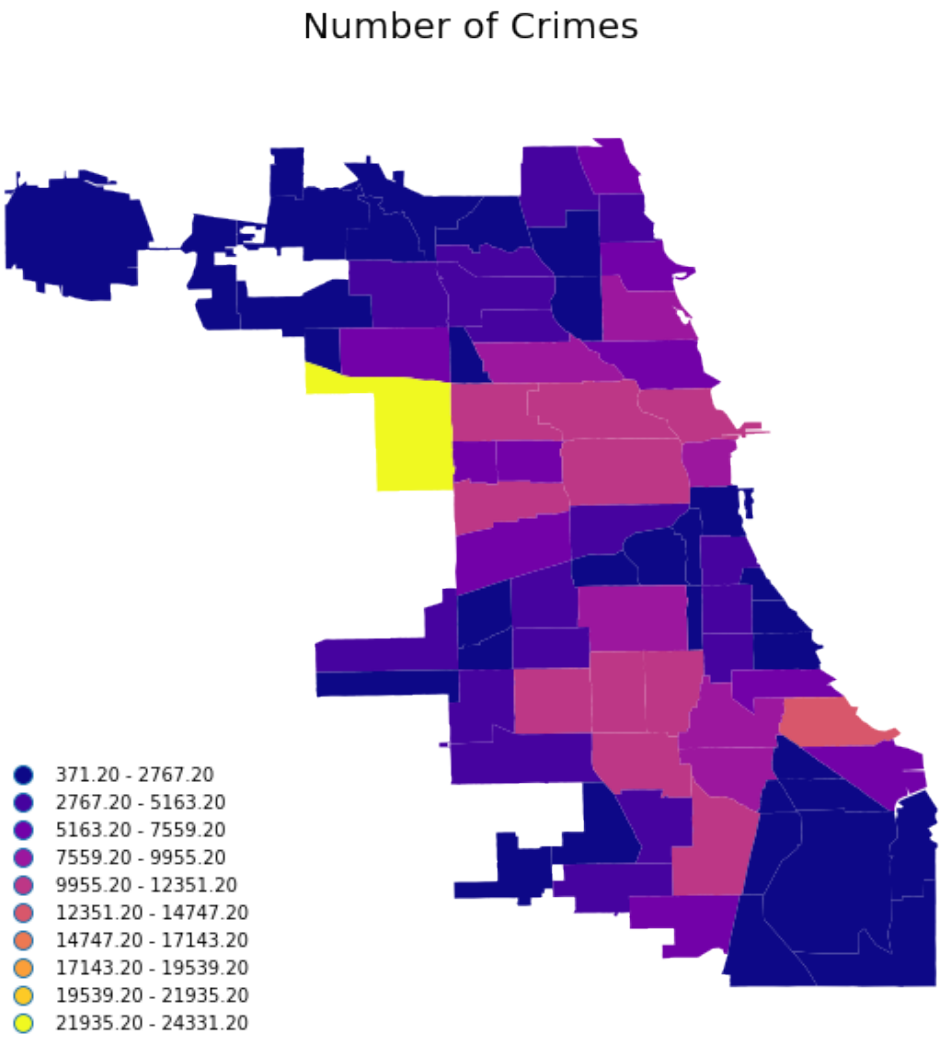
```
In [6]: chicomm_df = gpd.read_file("../data/chicomm/chicomm.shp")
        chicomm_df.rename(columns = {'CHICOMNO':'Community Area Number'}, inplace = True)
        chicomm_df.set_index('Community Area Number', inplace = True)
        chicomm_df.index = chicomm_df.index.astype(int)
        chicomm_df.sort_index(inplace = True)
```

Now merge the data with shapefile for plotting after aggregating

```
In [7]: # plotting crimes total
        crime_sum = crime_df.groupby('Community Area Number').sum()
        crime_sum.index = crime_sum.index.astype(int)
        crime_sum_geo = chicomm_df.join(crime_sum)
```

Here shows the sum of all kinds of crimes in different community areas

```
In [8]: ax_sum = crime_sum_geo.plot(column = 'No_Crimes', cmap = 'plasma', scheme = 'equal_inte
                                    k = 10, linewidth = 2,
                                    legend = True, figsize = (10, 10),
                                    )
        ax_sum.set_title("Number of Crimes", fontsize = 20, y = 1.05)
        ax_sum.get_legend().set_bbox_to_anchor((0.3, 0.3))
        ax_sum.get_legend().get_frame().set_linewidth(0)
        ax_sum.set_axis_off()
        # ax_sum.figure.savefig('output/maps/Number_of_Crimes.png')
```



1.1 Interactive Web Maps

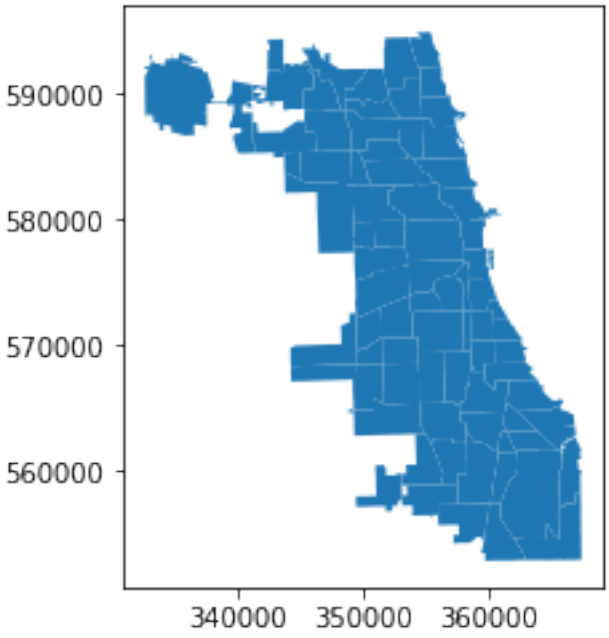
```
In [9]: # building interactive map
        import folium

        m = folium.Map([41.8, -87.8],
                       tiles='cartodbpositron',
                       zoom_start=9.3, max_zoom=17, min_zoom=4)


        colormap = folium.LinearColormap(("red", "orange", "yellow", "green",
                                          "blue", "purple"),
                                         vmin = crime_sum_geo['No_Crimes'].min(),
                                         vmax = crime_sum_geo['No_Crimes'].max(),
```

```
                                        caption = 'Number of Crimes')

colormap.add_to(m)

folium.GeoJson(crime_sum_geo,
               style_function = lambda feature: {
                  'fillColor': colormap(feature['properties']['No_Crimes']),
                  "color" : "black", "weight" : 1, "fillOpacity" : 0.55
               }).add_to(m)

# m.save("output/maps/Number_of_Crimes.html")

m
```

Out[9]: <folium.folium.Map at 0x7f019c309550>

1.2 Looking at the Jails in the City of Chicago

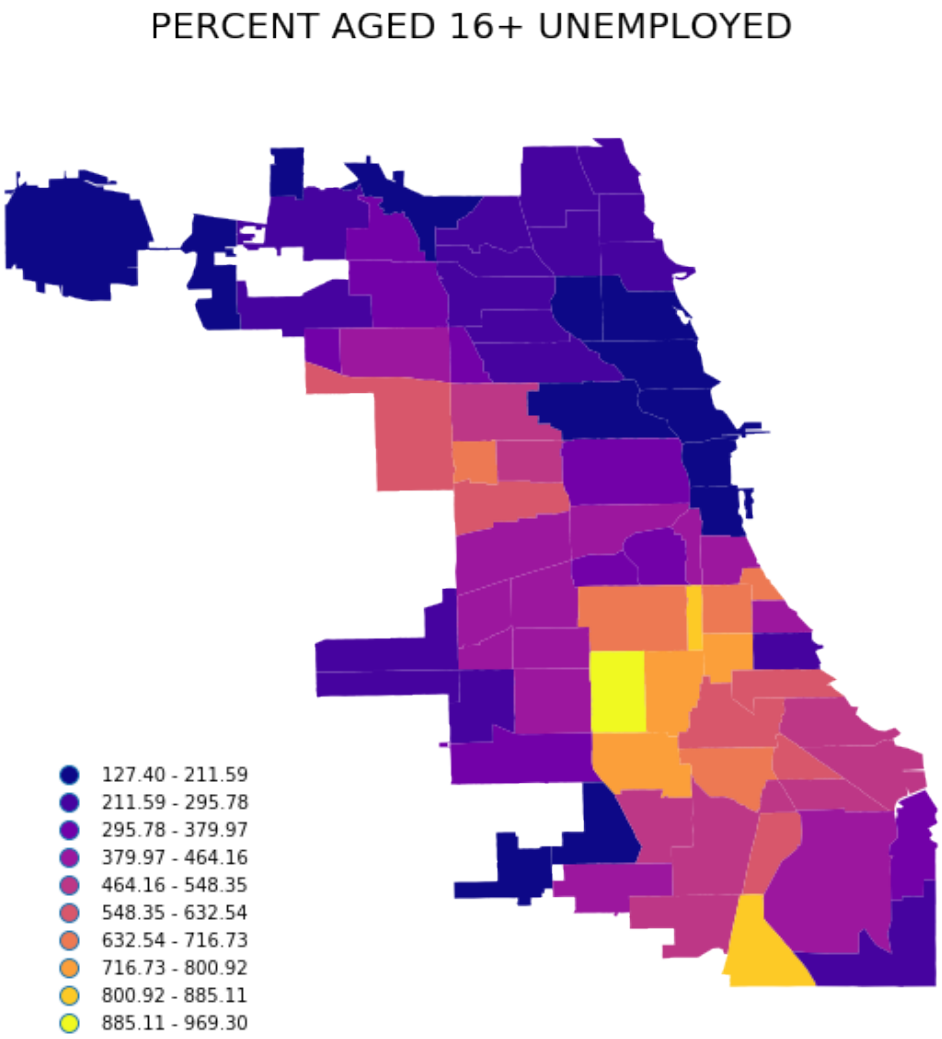
```
In [10]: df_4 = gpd.read_file('advanced_gis/inclassdata_unit4.gdb')

In [11]: crime_sum_geo.crs

Out[11]: {'init': 'epsg:4269'}

In [12]: df_4.crs

Out[12]: {'init': 'epsg:26971'}

In [13]: # reproject the maps
         df_4_proj = df_4.to_crs({'init': 'epsg:4269'})
         crime_sum_geo_proj = crime_sum_geo.to_crs({'init': 'epsg:26971'})

In [14]: crime_sum_geo_proj.plot()

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7f019a1ede48>
```




```
In [15]: crime_sum_geo.columns
```

```
Out[15]: Index(['DISTNAME', 'DISTITLE', 'FAMINC', 'HOUSINC', 'PERCAPINC', 'MEDVALOOH',
                'MEDRENT', 'geometry', 'No_Crimes', 'PERCENT OF HOUSING CROWDED',
                'PERCENT HOUSEHOLDS BELOW POVERTY', 'PERCENT AGED 16+ UNEMPLOYED',
                'PERCENT AGED 25+ WITHOUT HIGH SCHOOL DIPLOMA',
                'PERCENT AGED UNDER 18 OR OVER 64', 'PER CAPITA INCOME ',
                'HARDSHIP INDEX'],
               dtype='object')
```

```
In [16]: # plot the unemployment rate
         ax_sum = crime_sum_geo.plot(column = 'PERCENT AGED 16+ UNEMPLOYED', cmap = 'plasma', s
                                     k = 10, linewidth = 2,
                                     legend = True, figsize = (10, 10),
                                     )
         ax_sum.set_title("PERCENT AGED 16+ UNEMPLOYED", fontsize = 20, y = 1.05)
         ax_sum.get_legend().set_bbox_to_anchor((0.3, 0.3))
         ax_sum.get_legend().get_frame().set_linewidth(0)
         ax_sum.set_axis_off()
         # ax_sum.figure.savefig('output/maps/Number_of_Crimes.png')
```



In [17]:
```
# spatial join
Jails_within_community_area = gpd.sjoin(df_4, crime_sum_geo_proj, how="inner", op='int
```

In [18]:
```
# Number of Jails within each community area
Jails_within_community_area.DISTITLE.value_counts()
```

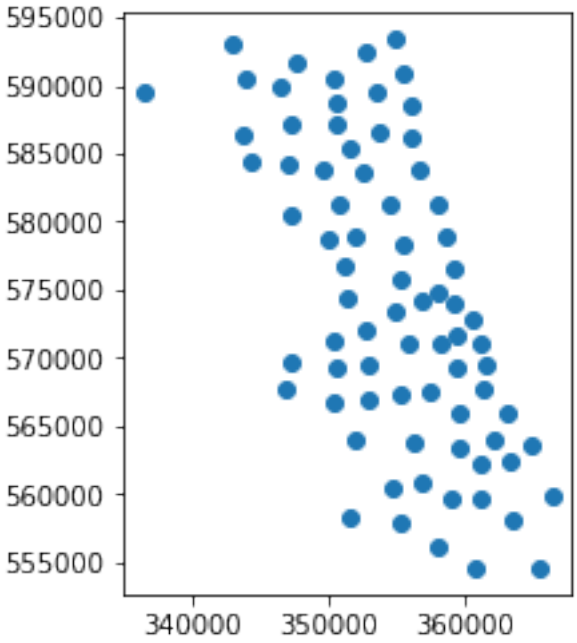
Out[18]:
```
Austin                 582
North Lawndale         228
Humboldt Park          221
West Englewood         218
Englewood              171
Auburn Gresham         168
South Shore            165
Roseland               160
East Garfield Park     153
```

New City	142
West Pullman	133
West Garfield Park	130
Chicago Lawn	110
Grand Boulevard	108
Near West Side	108
Greater Grand Crossing	98
West Town	98
South Chicago	88
Near North Side	88
Woodlawn	87
Chatham	86
Rogers Park	86
Logan Square	84
Douglas	79
Washington Heights	71
Uptown	69
Washington Park	60
Belmont Cragin	55
Irving Park	47
South Lawndale	45
	...
Fuller Park	22
Hyde Park	21
Hermosa	20
East Side	19
Lower West Side	19
North Center	17
Ashburn	17
Calumet Heights	17
Dunning	16
Pullman	15
Avalon Park	14
West Elsdon	13
Lincoln Square	13
West Lawn	12
Jefferson Park	11
Armour Square	10
Clearing	10
Burnside	8
Norwood Park	7
Beverly	6
McKinley Park	6
Hegewisch	5
North Park	5
Forest Glen	5
Montclare	4
O'Hare	4

```
Archer Heights              4
Mount Greenwood             3
Loop                        3
Edison Park                 1
Name: DISTITLE, Length: 77, dtype: int64
```

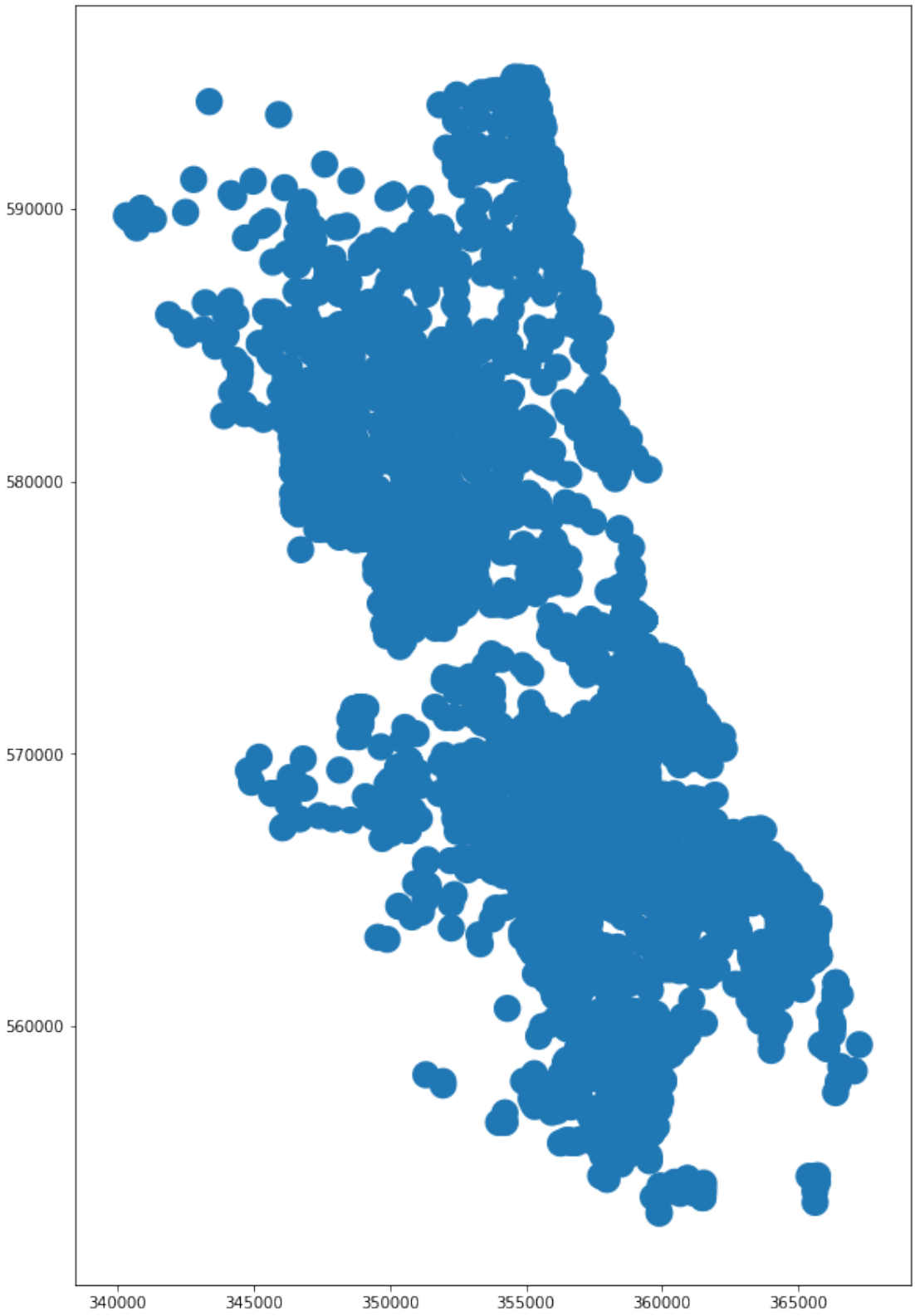
In [19]:
```
# plot the centroid of community areas
crime_sum_geo_proj.centroid.plot()
```

Out[19]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f019a065198>
```



In [20]:
```
# buffer for 500 meters
df_4_buffer = df_4.buffer(500)
df_4_buffer.plot(figsize = (15, 15))
```

Out[20]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0199c89b00>
```

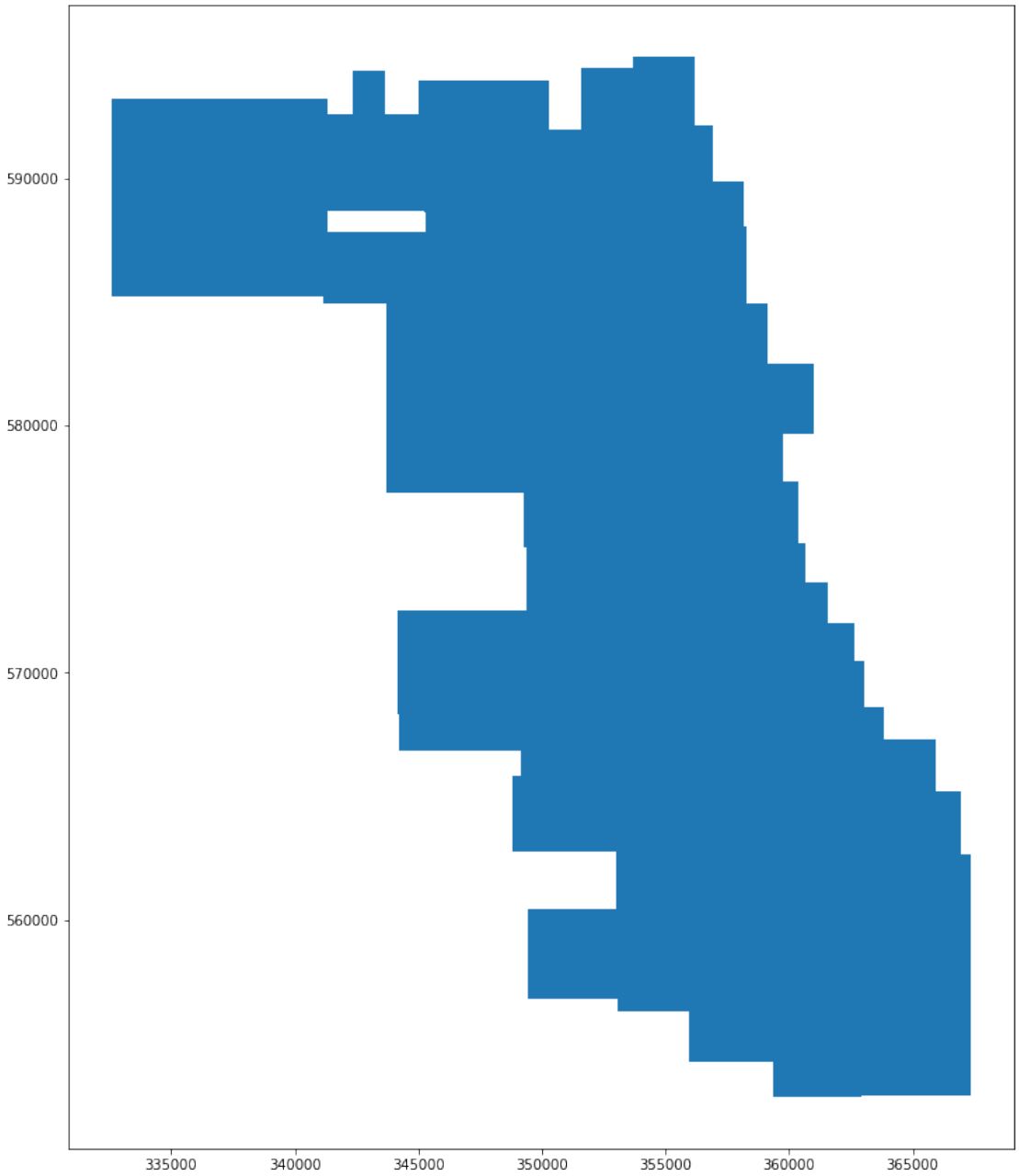

590000
580000
570000
560000
340000
345000
350000
355000
360000
365000

geopandas's overlay functions simply does not work for me

In [21]:
```
# df_4_buffer = gpd.GeoDataFrame({'geometry': df_4_buffer})
# no_jail = gpd.overlay(chicomm_df_proj, df_4_buffer, how='difference')
# this should be working but for some reason it just takes forever
```

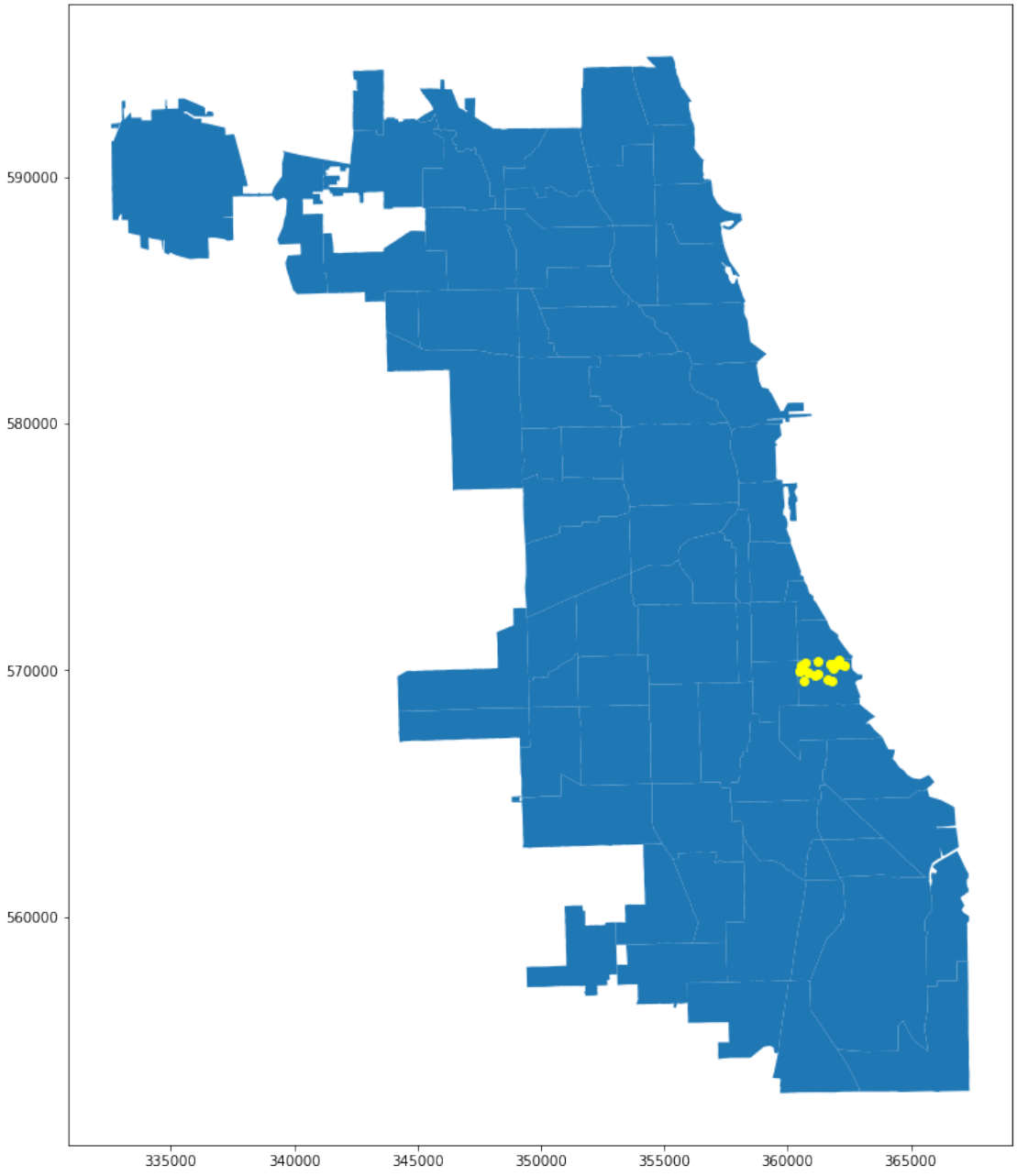
In [22]:
```
# envelope
# Returns a GeoSeries of geometries representing the point or smallest rectangular po
crime_sum_geo_proj.envelope.plot(figsize = (15, 15))
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0199c6ad30>



In [23]:
```
# find out the jails in hyde park with a for loop
hyde_park = []
for i in range(len(Jails_within_community_area)):
    if Jails_within_community_area.iloc[i, 6] == 'Hyde Park':
        hyde_park.append(Jails_within_community_area.iloc[i, 3])

hyde_park = gpd.GeoDataFrame({'geometry': hyde_park})
hyde_park.crs = {'init': 'epsg:26971'}
chicomm_df_proj = chicomm_df.to_crs({'init': 'epsg:26971'})
hyde_park_context = gpd.sjoin(hyde_park, chicomm_df_proj, how="inner", op='intersects

ax = chicomm_df_proj.plot(figsize = (15, 15))
hyde_park_context.plot(ax = ax, color = "yellow")
```

Out[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0199bdc080>
```


590000
580000
570000
560000
335000
340000
345000
350000
355000
360000
365000