

data_explore

February 6, 2022

- 0.1 Data collection, Preprocessing, Exploratory Data Analysis, Feature Engineering**
- 0.2 Topic: Identifying Vacant and Uninhabitable Properties using Publicly Available Data in Philadelphia**

Name: Akshay Srivastava, Prianka Ball and Reina Carissa

After researching different papers on lots, we decided to work on predicting vacants in Philadelphia. We accumulated different types of data after reading papers connected to vacant lots and made a list of data sources that might be useful while we are trying to predict and access vacant lots. Some of the data we collected might not be used in the final model.

Important packages that we used were geopandas and censusdata. geopandas package was used to plot maps and it assisted us while working with location data. Censusdata package was used to pull American Community Survey data easily.

The data laid out on this notebook are from the following sources:

1. **Philadelphia Shape Files:** Shape file is a data storage format for storing location, shape and attributes of geo. Shape files assist us to create maps and we also used it to join with other types of datasets. For this project we focused specifically on Philadelphia.
 - Neighbourhoods: Neighbourhoods act like social communities where people are expected to have more face-to-face interactions. There is no official list of neighbourhoods.
 - Source: https://github.com/azavea/geo-data/tree/master/Neighborhoods_Philadelphia
 - Census Block Groups: Census block group is a geographic unit used by United States Census Bureau. It is between Census Tract and Census Block. It is the smallest geographical unit for which the bureau publishes sample data. Typically Block Groups have a population of 600 to 3000 people. Data collected from American Community Survey had data at census block group level, so this shape file was important to analyze this dataset.
 - Source: <https://www.opendataphilly.org/dataset/census-block-groups>
 - Zip Code: Zip Code is a postal code used by United States Postal Service(USPS). The basic format normally consists of 5 digits. Some of the datasets collected from the city government has zip code level data, we used it to plot them.
 - Source: <https://www.opendataphilly.org/dataset/zip-codes>
2. **American Community Survey(5 year):** ACE is conducted by Census Bureau. But unlike the Decennial Census which is conducted every 10 years, the American Community Survey

is conducted more frequently. The census tries to count every person, whereas as the ACS is sent to sample addresses. For our purposes, we used the 5-year ACS data where the data has been collected over 5 years between 2015- 2019. Data in ACS is at the block group level. Censusdata package was used to pull the data

- Occupancy Status(Table Code:B25002): Data on whether the property was vacant or occupied.
- Vacancy Status(Table Code:B25004): Vacant properties could be further broken down according to their housing market classification. For our purposes, we will be focusing on “other” vacancy status. Vacant status is classified as other when it does not fall in any of the year round category
- Population

3. Philadelphia City: The Open Data Program of City of Philadelphia helps departments share data from the city government with the Public on [OpenDataPhilly](#)

- Crime: Data is collected from the Philadelphia Police Department. It has data from 2006-Present and is being updated everyday. We decided to use city crime data as a lot of papers mentioned that areas with vacant lots tend to have high crime rate.
 - Source: <https://metadata.phila.gov/#home/datasetdetails/5543868920583086178c4f8e/representations>
- Property Assessment: Data is collected by the Philadelphia Properties and Assessment History. It includes property characteristics and assessment information from the Office of Property Assessment. This dataset includes data of properties that are already known as vacant lands by the city government.
 - Source: <https://metadata.phila.gov/#home/datasetdetails/5543865f20583086178c4ee5/representations>
- 311 Data: Contains data about 311 Service Requests. This represents all service and information requests since December 8th, 2014 submitted to Philly311 via the 311 mobile application, calls, walk-ins, emails, the 311 website or social media. We incorporated this data set as vacant lots tends to get more 311 calls to reports about things like illegal dumping, maintenance services, graffiti removal, etc
 - Source: <https://metadata.phila.gov/#home/datasetdetails/5543864d20583086178c4e98/representations>
- Property Tax Delinquency: This is a dataset that shows the Philadelphia properties with tax delinquencies, including those that are in payment agreements. An account is delinquent when Real Estate Tax is still unpaid on January 1 the following year the tax was due. Data is from 1972 - 2018. Vacant lots tend to have unpaid taxes and this is good indicator that properties might be vacant soon.
 - Source: <https://metadata.phila.gov/#home/datasetdetails/57d9643afab162fe2708224e/representations>
- Property Code Violations: Data contains violations issued by the Department of License and Inspection. We downloaded datasets for 2013-2015, 2016-2018, 2019-now. Data contains where the violation was occurred and reason for violation. Some of the violations are vacant lot related.
 - Source: <https://www.opendataphilly.org/dataset/licenses-and-inspections-violations>

4. Predicted Vacant Lots: The office of Innovation and Technology of City of Philadelphia aggregated multiple city administrative and geographic data source to come up with a model that can identify building or land vacancy in each tax parcel boundary in the city. We will be using this dataset to measure our model's performance and accuracy rate

- Source: <https://metadata.phila.gov/#home/datasetdetails/58078697d414285d25b14e3c/representation>

The notebook also contains datasets that seemed important but we are still assessing if we should use them. These datasets are not yet explored properly. These are

1. Open Street Map: The Open Street Map is a collaborative project that created free editable geographic database of the world. The dataset contains different types of data such as public places, hospitals, restaurants, main roads, museums etc

- Source: <https://download.geofabrik.de/north-america/us/pennsylvania.html>
- Details: <http://download.geofabrik.de/osm-data-in-gis-formats-free.pdf>

2. American Community Survey:

- Race (Table Code:B03002)
- Age and Sex (Table Code:B01001)
- Poverty Status (Table Code:B17001)
- Household Income (Table Code:B19001)
- Education (Table Code:B15001)
- ZCTAs: ZIP Code Tabulation Areas (ZCTAs) are generalized areal representations of United States Postal Service (USPS) ZIP Code service areas. The ZCTAs were created by first examining all of the addresses within each census block and then the most frequently occurring zip code within each block was assigned to the entire census block. This dataset might be used to connect ACS data with other types of data.
 - Source: <https://www.census.gov/programs-surveys/geography/guidance/geographic-areas/zctas.html>

We have recently added information on how these different datasets were joined together given that they are using very different spatially. The Property Assessment file was considered as the file based on which everything else was joined together. Some of the files like the property assessment, property code violation and property tax delinquency had parcel number in them. So these were joined based on parcel number. Other files like crime and 311 datasets did not have parcel number so we tried to find out the number of crimes and 311 calls that happened within 50m of these parcel number. Others like the ACS data had only block group level data so these were joined with the rest of the dataset spatially i.e if the lat/lng of the parcel number was within the block group then it was assigned that specific block group.

We have created the labelled column using the property assessment dataset. The property assessment column included data about what type of property it was. Based on this information, we have created a new column that identifies if it is a vacant lot or not. We will be using this labelled column for the ML model. Details of the process if included in the rest of the notebook.

[]: #Libraries used

```
import pandas as pd
import numpy as np
import geopandas as gpd # for mapping
import matplotlib.pyplot as plt
```

```

import descartes # for mapping
from shapely.geometry import Point, Polygon #for mapping
import seaborn as sns
import censusdata # to pull data from census
from datetime import datetime
import folium
from folium.plugins import HeatMap

%matplotlib inline

```

0.2.1 Philly Shape File: ZCTAs

<https://www.census.gov/programs-surveys/geography/guidance/geo-areas/zctas.html>

Dataset contains ZACTs of all the US. We are still deciding if we need to use it to connect city data with ACS data.

```
[ ]: census_zip = gpd.read_file("data/census_shape/cb_2018_us_zcta510_500k/
→cb_2018_us_zcta510_500k.shp") # loading dataset
census_zip.head()
#ZCTA5CE10 -- zip code tabulation area
```

```
[ ]:   ZCTA5CE10      AFFGEOID10  GEOID10      ALAND10  AWATER10  \
0      36083  8600000US36083      36083  659750662  5522919
1      35441  8600000US35441      35441  172850429  8749105
2      35051  8600000US35051      35051  280236456  5427285
3      35121  8600000US35121      35121  372736030  5349303
4      35058  8600000US35058      35058  178039922  3109259
```

```
                                geometry
0  MULTIPOLYGON (((-85.63225 32.28098, -85.62439 ...
1  MULTIPOLYGON (((-87.83287 32.84437, -87.83184 ...
2  POLYGON ((-86.74384 33.25002, -86.73802 33.251...
3  POLYGON ((-86.58527 33.94743, -86.58033 33.948...
4  MULTIPOLYGON (((-86.87884 34.21196, -86.87649 ...
```

```
[ ]: census_zip.shape #size of data
```

```
[ ]: (33144, 6)
```

```
[ ]: census_zip.nunique()#unique values in dataset
```

```
[ ]: ZCTA5CE10      33144
AFFGEOID10      33144
GEOID10         33144
ALAND10         33138
AWATER10        28425
```

```
geometry      33144
dtype: int64
```

0.2.2 Philly Shape File: Census Block Group

<https://www.opendataphilly.org/dataset/census-block-groups> <https://metadata.phila.gov/#home/datasetdetails/>

Dataset was used later while plotting ACS dataset

```
[ ]: census_blockgroups = gpd.read_file("data/census_shape/
    ↳Census_Block_Groups_2010-shp/Census_Block_Groups_2010.shp")#loading dataset
census_blockgroups.head()
```

```
[ ]:   OBJECTID STATEFP10 COUNTYFP10 TRACTCE10 BLKGRPCE10           GEOID10 \
0          1        42       101  010800          1 421010108001
1          2        42       101  010800          2 421010108002
2          3        42       101  010900          2 421010109002
3          4        42       101  011000          2 421010110002
4          5        42       101  011000          1 421010110001

      NAMELSA10 MTFCC10 FUNCSTAT10 ALAND10 AWATER10     INTPTLAT10 \
0  Block Group 1  G5030             S  161887          0 +39.9687580
1  Block Group 2  G5030             S  103778          0 +39.9665475
2  Block Group 2  G5030             S   43724          0 +39.9642929
3  Block Group 2  G5030             S  108966          0 +39.9753585
4  Block Group 1  G5030             S  142244          0 +39.9724202

      INTPTLON10     Shape__Are     Shape__Len \
0 -075.1997251  1.742508e+06  8200.327170
1 -075.2004455  1.117026e+06  4364.980144
2 -075.1896435  4.706347e+05  3048.109084
3 -075.2113476  1.172871e+06  5169.004282
4 -075.2051689  1.531076e+06  10476.574129

      geometry
0  POLYGON ((-75.19851 39.96945, -75.19744 39.969...
1  POLYGON ((-75.19783 39.96571, -75.20006 39.965...
2  POLYGON ((-75.18766 39.96450, -75.18755 39.963...
3  POLYGON ((-75.20984 39.97351, -75.21221 39.973...
4  POLYGON ((-75.19855 39.97330, -75.19854 39.973...
```

```
[ ]: census_blockgroups.nunique() #number of unique values in each column.
#important columns to consider is the TRACTCE10 and BLKGRPCE10. The TRACTCE10 ↳
#is tracts and BLKGRPCE10 stands for block group.
# The number of tract and blockgroup is equal to what was pulled from ACS
```

```
[ ]: OBJECTID      1336
STATEFP10        1
COUNTYFP10       1
TRACTCE10        384
BLKGRPCE10       8
GEOID10          1336
NAMELSAD10       8
MTFCC10          1
FUNCSTAT10       1
ALAND10          1332
AWATER10          121
INTPTLAT10       1336
INTPTLON10       1336
Shape__Are        1336
Shape__Len        1336
geometry          1336
dtype: int64
```

```
[ ]: census_blockgroups['BLKGRPCE10'].value_counts()
```

```
[ ]: 1    384
2    349
3    271
4    176
5    88
6    45
7    21
8    2
Name: BLKGRPCE10, dtype: int64
```

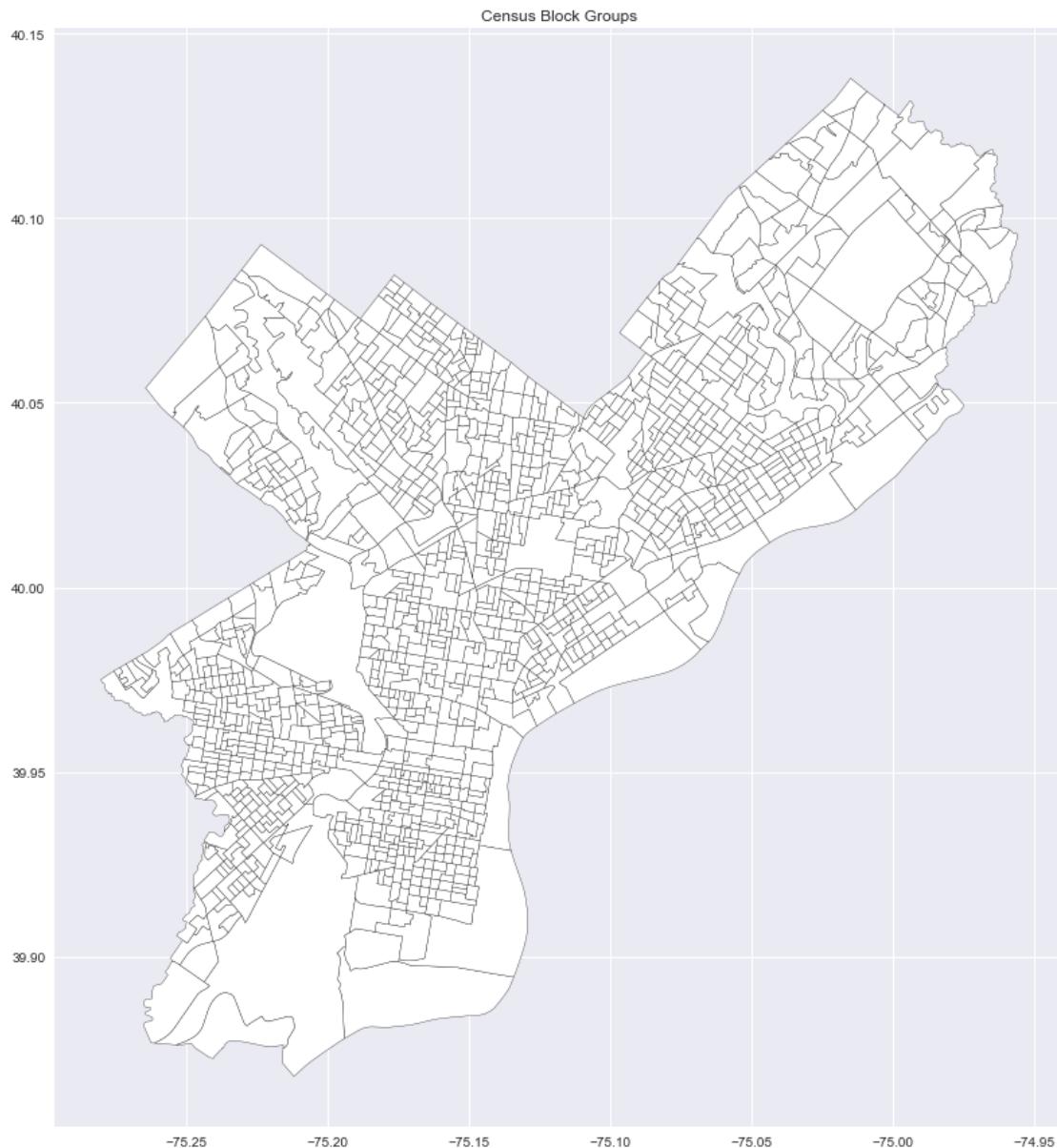
```
[ ]: census_blockgroups['TRACTCE10'].value_counts()
```

```
[ ]: 026600     8
039000     8
018800     7
008200     7
031600     7
..
980500     1
980400     1
023500     1
000200     1
005600     1
Name: TRACTCE10, Length: 384, dtype: int64
```

```
[ ]: #plotting map using the census blockgroup
fig, ax = plt.subplots(figsize=(15,15))
```

```
plt.style.use('seaborn')
plt.title("Census Block Groups")
census_blockgroups.to_crs("EPSG:4269").plot(ax=ax, color='white', u
→edgecolor='black')# epsg is the coordinate reference system(crs).
#CRS tells python how these coordinates related to places on the Earth
#"EPSG:4269" is for latitude, longitude projection
```

```
[ ]: <AxesSubplot:title={'center':'Census Block Groups'}>
```



0.2.3 Philly Shape file: Neighbourhoods

<https://github.com/azavea/geo-data>

```
[ ]: street_map = gpd.read_file("data/geo_shape/Neighborhoods_Philadelphia.
→shp")#loading dataset
#Download shape file from here. Download all files under folder ↳
→"Neighborhoods_philadelphia" and keep in the same folder https://github.com/
→azavea/geo-data
```

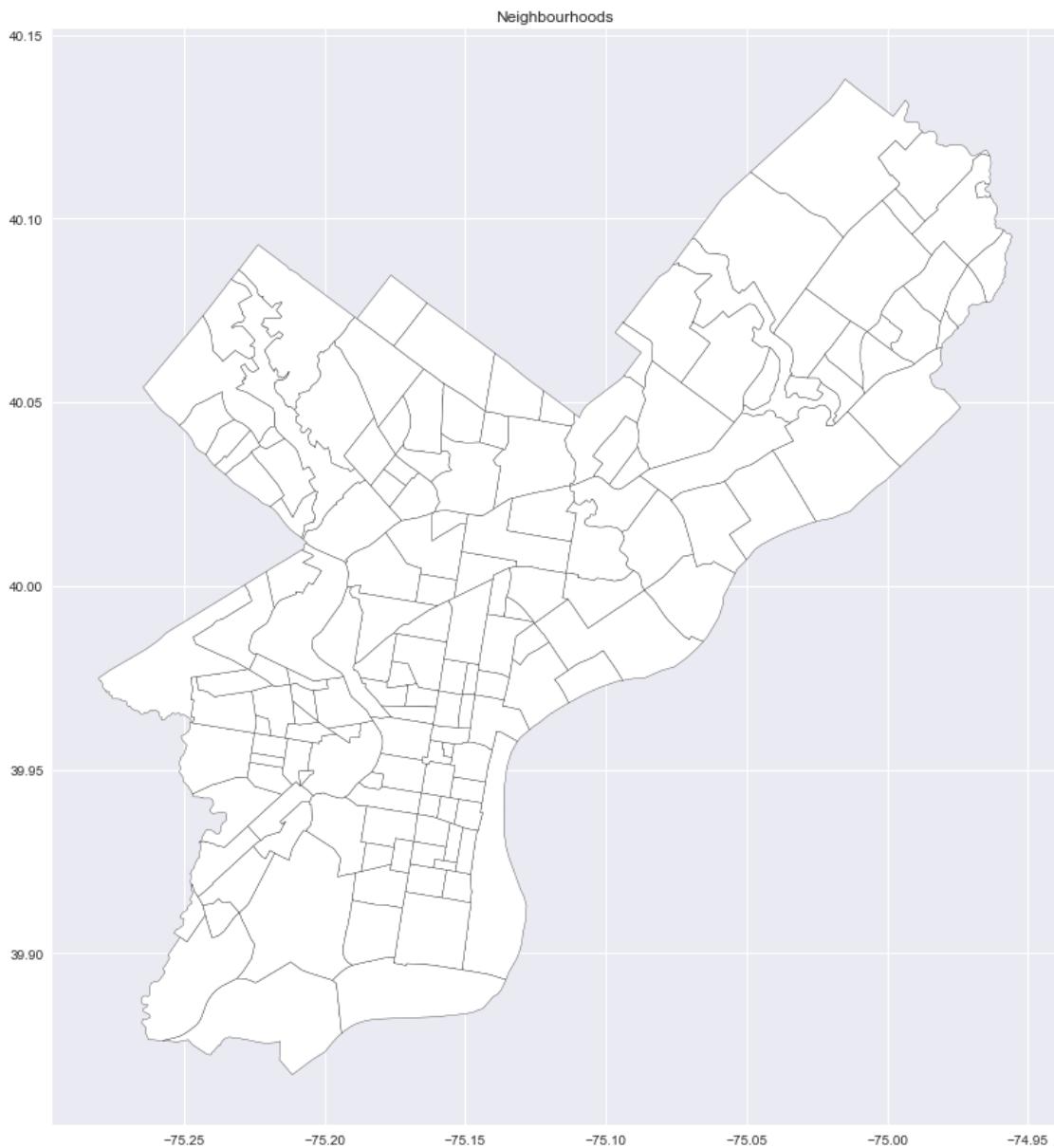
```
[ ]: street_map.head()
```

```
[ ]:      NAME    LISTNAME    MAPNAME    Shape_Leng    Shape_Area \
0    BRIDESBURG    Bridesburg    Bridesburg    27814.546521    4.458626e+07
1    BUSTLETON    Bustleton    Bustleton    48868.458365    1.140504e+08
2    CEDARBROOK    Cedarbrook    Cedarbrook    20021.415802    2.487174e+07
3  CHESTNUT_HILL    Chestnut Hill    Chestnut Hill    56394.297195    7.966498e+07
4    EAST_FALLS    East Falls    East Falls    27400.776417    4.057689e+07

                           geometry
0  POLYGON ((2719789.837 256235.538, 2719814.855 ...
1  POLYGON ((2733378.171 289259.945, 2732818.985 ...
2  POLYGON ((2685267.950 279747.336, 2685272.265 ...
3  POLYGON ((2678490.151 284400.400, 2678518.732 ...
4  POLYGON ((2686769.727 263625.367, 2686921.108 ...
```

```
[ ]: #Plotting with neighbourhoods
fig,ax = plt.subplots(figsize =(15,15))
plt.title("Neighbourhoods")
street_map.to_crs(epsg = 4326).plot(ax = ax, color = "white", ↳
→edgecolor='black')# converting axis to coordinate with longitude and latitude
```

```
[ ]: <AxesSubplot:title={'center':'Neighbourhoods'}>
```



```
[ ]: street_map.total_bounds# exact city boundary
```

```
[ ]: array([2660586.2010556, 204650.55486186, 2750109.00494927,  
304965.32339202])
```

```
[ ]: street_map.centroid# center coordinate of the shape
```

```
[ ]: 0      POINT (2719422.233 253264.287)  
1      POINT (2725947.795 288491.804)  
2      POINT (2688745.576 280652.166)
```

```
3      POINT (2679098.697 279137.188)
4      POINT (2685458.776 259484.374)
...
153     POINT (2688489.596 218958.968)
154     POINT (2697705.388 227294.296)
155     POINT (2691305.087 226663.440)
156     POINT (2688805.843 226518.573)
157     POINT (2693761.573 226871.685)
Length: 158, dtype: geometry
```

0.2.4 Philly Shape Files: Zip Codes

Source: <https://www.opendataphilly.org/dataset/zip-codes>

```
[ ]: poly_zip = gpd.read_file("data/zip_shape/Zipcodes_Poly-shp/Zipcodes_Poly.shp") #_
    ↪uploading dataset
poly_zip.head()
```

```
[ ]:   OBJECTID    CODE    COD    Shape__Are    Shape__Len  \
0          1  19120    20  9.177970e+07  49921.544063
1          2  19121    21  6.959879e+07  39534.887217
2          3  19122    22  3.591632e+07  24124.645221
3          4  19123    23  3.585175e+07  26421.728982
4          5  19124    24  1.448080e+08  63658.770420

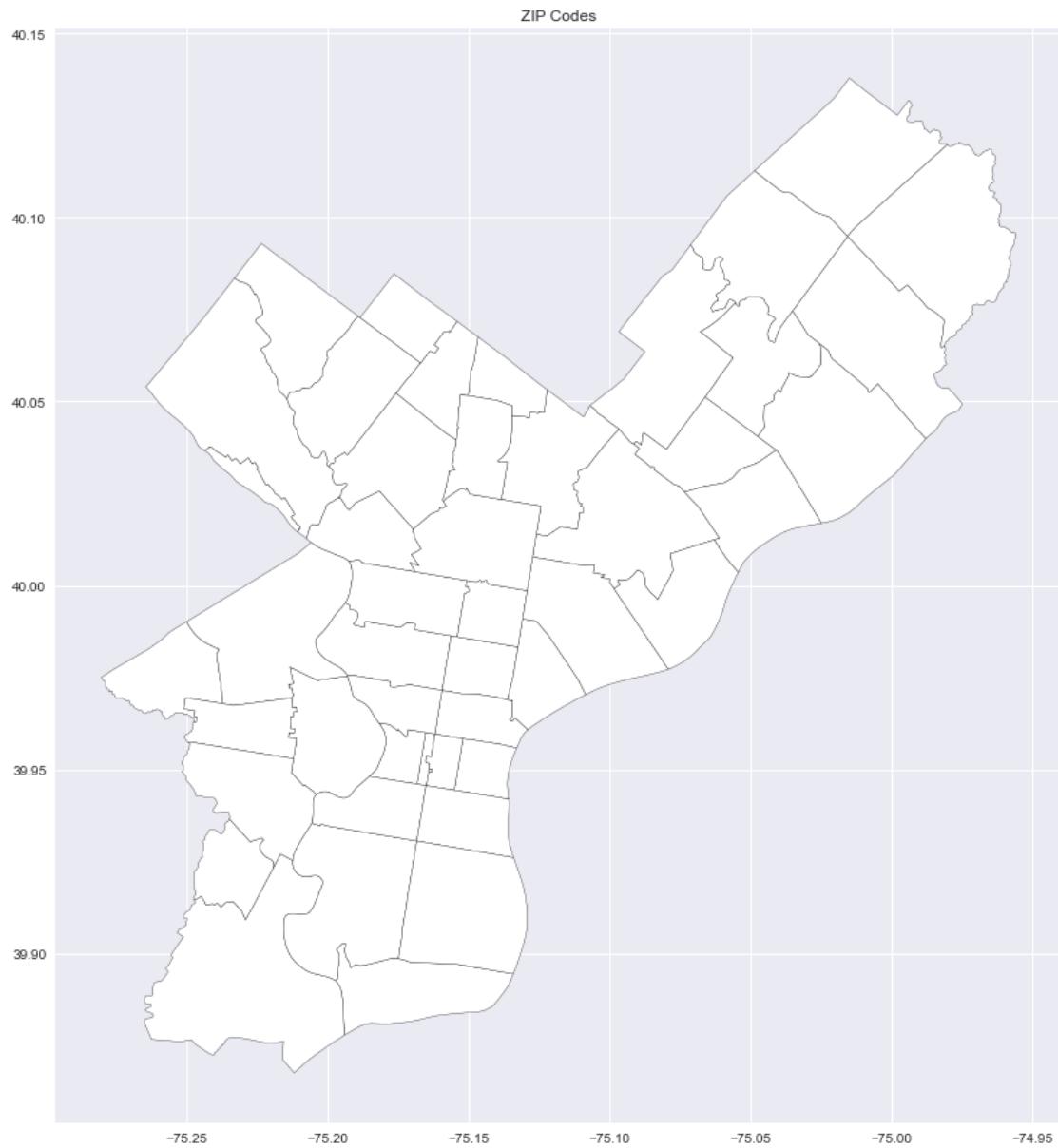
                                                geometry
0  POLYGON ((-75.11107 40.04682, -75.10943 40.045...
1  POLYGON ((-75.19227 39.99463, -75.19205 39.994...
2  POLYGON ((-75.15406 39.98601, -75.15328 39.985...
3  POLYGON ((-75.15190 39.97056, -75.15150 39.970...
4  POLYGON ((-75.09660 40.04249, -75.09281 40.039...
```

```
[ ]: poly_zip.dtypes
```

```
[ ]: OBJECTID        int64
CODE            object
COD            int64
Shape__Are      float64
Shape__Len      float64
geometry        geometry
dtype: object
```

```
[ ]: #plotting with zip codes
fig,ax = plt.subplots(figsize =(15,15))
plt.title("ZIP Codes")
poly_zip.to_crs(epsg = 4326).plot(ax = ax, color = "white", edgecolor='black')
```

```
[ ]: <AxesSubplot:title={'center':'ZIP Codes'}>
```



0.2.5 Open Street Map Files

<https://download.geofabrik.de/north-america/us/pennsylvania.htm>

<http://download.geofabrik.de/osm-data-in-gis-formats-free.pdf>

We are still deciding how and where to use this dataset

```
[ ]: #download pennsylvania open street map data
roads_path = "data/osm/gis_osm_roads_free_1.shp" #loading the roads file from
    ↵open map
roads = gpd.read_file(roads_path, encoding='utf-8')
```

```
[ ]: roads_new = roads.to_crs(epsg = 4326)
street_map_new = street_map.to_crs(epsg = 4326) # neighbourhood shape file
```

```
[ ]: street_map_new.head()
```

	NAME	LISTNAME	MAPNAME	Shape_Leng	Shape_Area	\
0	BRIDESBURG	Bridesburg	Bridesburg	27814.546521	4.458626e+07	
1	BUSTLETON	Bustleton	Bustleton	48868.458365	1.140504e+08	
2	CEDARBROOK	Cedarbrook	Cedarbrook	20021.415802	2.487174e+07	
3	CHESTNUT_HILL	Chestnut Hill	Chestnut Hill	56394.297195	7.966498e+07	
4	EAST_FALLS	East Falls	East Falls	27400.776417	4.057689e+07	

	geometry						
0	POLYGON ((-75.06773 40.00540, -75.06765 40.005...						
1	POLYGON ((-75.01560 40.09487, -75.01768 40.092...						
2	POLYGON ((-75.18848 40.07273, -75.18846 40.072...						
3	POLYGON ((-75.21221 40.08604, -75.21210 40.086...						
4	POLYGON ((-75.18478 40.02837, -75.18426 40.027...						

```
[ ]: roads_new.head()
```

	osm_id	code	fclass	name	ref	oneway	\
0	368034	5115	tertiary	Seaport Drive	None	F	
1	368041	5113	primary	Industrial Highway	US 13;PA 291	B	
2	368043	5115	tertiary	Bullens Lane	None	F	
3	368044	5113	primary	Chester Road	PA 320	B	
4	418185	5113	primary	East 9th Street	US 13 Business	B	

	maxspeed	layer	bridge	tunnel	\
0	0	0	F	F	
1	0	0	F	F	
2	56	0	F	F	
3	64	1	T	F	
4	56	0	F	F	

	geometry						
0	LINESTRING (-75.38773 39.82798, -75.38600 39.8...						
1	LINESTRING (-75.35786 39.84750, -75.35676 39.8...						
2	LINESTRING (-75.35060 39.86874, -75.35050 39.8...						
3	LINESTRING (-75.36147 39.87190, -75.36118 39.8...						
4	LINESTRING (-75.35941 39.85319, -75.35874 39.8...						

```
[ ]: #filtering only for philadelphia shape using neighbourhood shape file
roads = gpd.sjoin(roads_new, street_map_new, predicate ='intersects') # joining
    ↳both datasets based on their locations
```

```
[ ]: roads.head()
```

```
osm_id code fclass name ref oneway maxspeed \
235570 12108955 5122 residential Brunner Street None F 0
235571 12108958 5122 residential Brunner Street None F 0
238204 12119360 5122 residential Gratz Street None F 0
239953 12133630 5122 residential Staub Street None B 0
239955 12133635 5122 residential Staub Street None F 0

layer bridge tunnel \
235570 0 F F
235571 0 F F
238204 0 F F
239953 0 F F
239955 0 F F

geometry index_right \
235570 LINESTRING (-75.15542 40.01863, -75.15717 40.0... 61
235571 LINESTRING (-75.15710 40.01773, -75.15856 40.0... 61
238204 LINESTRING (-75.15635 40.02050, -75.15634 40.0... 61
239953 LINESTRING (-75.15447 40.01708, -75.15510 40.0... 61
239955 LINESTRING (-75.15458 40.01727, -75.15440 40.0... 61

NAME LISTNAME MAPNAME Shape_Leng Shape_Area
235570 NICETOWN Nicetown Nicetown 11237.318154 6.587596e+06
235571 NICETOWN Nicetown Nicetown 11237.318154 6.587596e+06
238204 NICETOWN Nicetown Nicetown 11237.318154 6.587596e+06
239953 NICETOWN Nicetown Nicetown 11237.318154 6.587596e+06
239955 NICETOWN Nicetown Nicetown 11237.318154 6.587596e+06
```

```
[ ]: roads.shape
```

```
(72827, 17)
```

```
[ ]: roads.fclass.value_counts()# this shows the type of roads. Most of the roads
    ↳are ones that are used for service, foorway and residential
```

```
service 26354
footway 19962
residential 13744
primary 3392
tertiary 2338
secondary 2009
path 1132
```

```
trunk           688
motorway_link   649
motorway        625
steps           569
cycleway        351
trunk_link      190
pedestrian      190
unclassified    153
primary_link    150
track            123
secondary_link   87
tertiary_link    62
bridleway       41
track_grade2     5
living_street    5
unknown          4
track_grade5     2
track_grade1     2
Name: fclass, dtype: int64
```

```
[ ]: #selecting roads where car travel
car_roads = roads[(roads.fclass == 'tertiary') |
                   (roads.fclass == 'tertiary_link') |
                   (roads.fclass == 'secondary') |
                   (roads.fclass == 'secondary_link') |
                   (roads.fclass == 'primary') |
                   (roads.fclass == 'primary_link') |
                   (roads.fclass == 'motorway') |
                   (roads.fclass == 'motorway_linkt'))]

car_roads.shape
```

```
(8663, 17)
```

```
[ ]: #plotting with different type of roads
fig, ax = plt.subplots(figsize =(15,15))
car_roads.plot(ax = ax, markersize=0.01, column='fclass', figsize=(5, 5), cmap = magma)
plt.axis('off');
plt.title("Car Roads")
```

```
Text(0.5, 1.0, 'Car Roads')
```

Car Roads



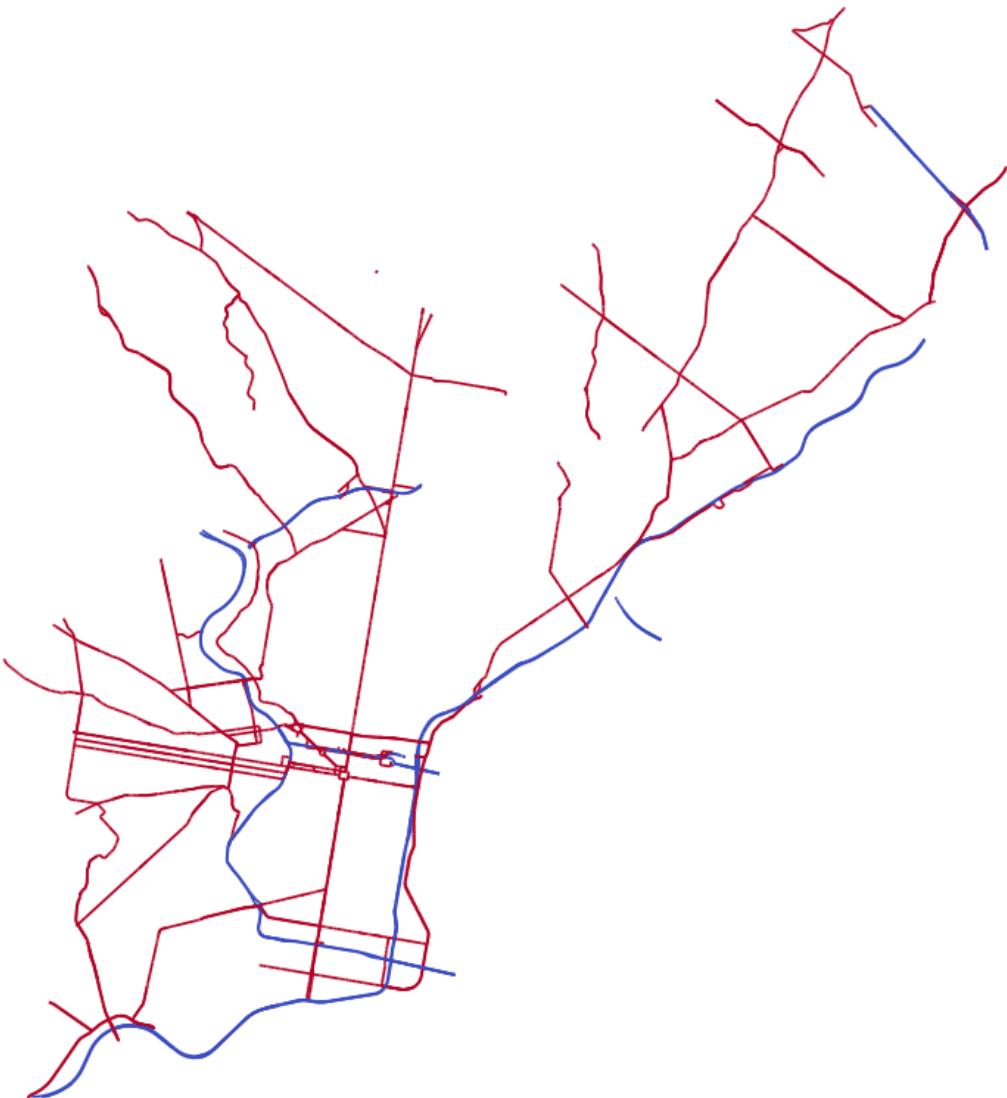
```
[ ]: #plotting with main roads only
main_roads = car_roads[(car_roads.fclass == 'primary') |
                      (car_roads.fclass == 'motorway')]
                      ]#selectng only primary and motorways

fig, ax = plt.subplots(figsize =(15,15))
```

```
main_roads.plot(ax = ax, column='fclass', cmap = 'coolwarm')
plt.axis('off')
plt.title("Main Roads - Philadelphia")
```

Text(0.5, 1.0, 'Main Roads - Philadelphia')

Main Roads - Philadelphia



0.2.6 City of Philadelphia: Predicted Vacant Lot

<https://metadata.phila.gov/#home/datasetdetails/58078697d414285d25b14e3c/representationdetails/59c154f1c8>

```
[ ]: philly_points = pd.read_csv('data/city/Vacant_Indicators_Points.csv') #loading ↴the dataset  
philly_points.head()
```

```
          X      Y   OBJECTID      ADDRESS \
0 -75.178904 39.934505       1    2041 REED ST
1 -75.164548 39.988160       2   2233 N UBER ST
2 -75.180480 39.978561       3  1460 N MARSTON ST
3 -75.186579 40.006852       4  3241 SUGDENS ROW
4 -75.238794 39.954902       5   5816 PINE ST

          OWNER1      OWNER2      BLDG_DESC \
0 GREATER DELIVERANCE TEMPL      NaN  VAC LAND COMM. < ACRE
1           CITY OF PHILA      NaN  VAC LAND RES < ACRE
2 PHILADELPHIA HOUSING AUTH      NaN  VAC LAND RES < ACRE
3           JORDAN MARIA      NaN  VAC LAND RES < ACRE
4        WALSH JAMES  LUBLIN WILLIAM H  ROW 2 STY MASONRY

      OPA_ID  LNIADDRESSKEY  COUNCILDISTRICT  ZONINGBASEDISTRICT  ZIPCODE \
0  885396760.0        498086             2            CMX-2  19146.0
1  162113701.0        581713             5            RSA-5  19132.0
2  292083110.0        415511             5            RSA-5  19121.0
3  382209500.0        557317             4            RSA-5  19129.0
4  604178400.0        485528             3            RM-1  19143.0

  LAND_RANK  BUILD_RANK  VACANT_FLAG  VACANT_RANK
0      0.67        0.0     Land        0.67
1      0.50        0.0     Land        0.50
2      0.50        0.0     Land        0.50
3      1.00        0.0     Land        1.00
4      0.00        1.0 Building        1.00
```

```
[ ]: philly_points.shape
```

```
[ ]: (36917, 16)
```

```
[ ]: philly_points.dtypes #Type of data
```

```
X      float64
Y      float64
OBJECTID      int64
ADDRESS      object
OWNER1      object
OWNER2      object
BLDG_DESC      object
OPA_ID      float64
LNIADDRESSKEY      object
COUNCILDISTRICT      int64
```

```
ZONINGBASEDISTRICT      object
ZIPCODE                  float64
LAND_RANK                 float64
BUILD_RANK                 float64
VACANT_FLAG                object
VACANT_RANK                 float64
dtype: object
```

```
[ ]: philly_points.isna().sum() # sum of null values in each column
```

X	0
Y	0
OBJECTID	0
ADDRESS	1
OWNER1	1
OWNER2	28556
BLDG_DESC	70
OPA_ID	21
LNIADDRESSKEY	336
COUNCILDISTRICT	0
ZONINGBASEDISTRICT	40
ZIPCODE	322
LAND_RANK	2
BUILD_RANK	0
VACANT_FLAG	2
VACANT_RANK	2
dtype: int64	

```
[ ]: philly_points.describe(include = 'all') # Vacant flag column indicates if the ↪property is likely to be a vacant building or vacant land
```

	X	Y	OBJECTID	ADDRESS	\
count	36917.000000	36917.000000	36917.000000	36916	
unique	NaN	NaN	NaN	36855	
top	NaN	NaN	NaN	4923R-47 N 16TH ST	
freq	NaN	NaN	NaN	27	
mean	-75.167337	39.985244	18459.00000	NaN	
std	0.041322	0.031944	10657.16428	NaN	
min	-75.269183	39.883301	1.00000	NaN	
25%	-75.189216	39.968623	9230.00000	NaN	
50%	-75.165435	39.986638	18459.00000	NaN	
75%	-75.144508	39.999227	27688.00000	NaN	
max	-74.964149	40.135042	36917.00000	NaN	
	OWNER1	OWNER2	BLDG_DESC	OPA_ID	\
count	36916	8361	36847	3.689600e+04	
unique	20842	4550	311	NaN	

top	CITY OF PHILA	OF PHILADELPHIA	VAC LAND RES < ACRE		NaN
freq		2412	642	23752	NaN
mean		NaN	NaN	NaN	3.431197e+08
std		NaN	NaN	NaN	2.276140e+08
min		NaN	NaN	NaN	1.100490e+07
25%		NaN	NaN	NaN	1.831285e+08
50%		NaN	NaN	NaN	3.110167e+08
75%		NaN	NaN	NaN	4.320889e+08
max		NaN	NaN	NaN	8.886000e+08

	LNIADDRESSKEY	COUNCILDISTRICT	ZONINGBASEDISTRICT	ZIPCODE	\
count	36581	36917.000000	36877	36595.000000	
unique	36523	NaN	34	NaN	
top	749746	NaN	RSA-5	NaN	
freq	27	NaN	20941	NaN	
mean	NaN	4.913211	NaN	19131.573330	
std	NaN	2.175462	NaN	11.154188	
min	NaN	1.000000	NaN	19102.000000	
25%	NaN	3.000000	NaN	19122.000000	
50%	NaN	5.000000	NaN	19133.000000	
75%	NaN	7.000000	NaN	19140.000000	
max	NaN	10.000000	NaN	19154.000000	

	LAND_RANK	BUILD_RANK	VACANT_FLAG	VACANT_RANK
count	36915.000000	36917.000000	36915	36915.000000
unique	NaN	NaN	2	NaN
top	NaN	NaN	Land	NaN
freq	NaN	NaN	27613	NaN
mean	0.525169	0.150244	NaN	0.665391
std	0.343984	0.252557	NaN	0.184587
min	0.000000	0.000000	NaN	0.500000
25%	0.415000	0.000000	NaN	0.500000
50%	0.500000	0.000000	NaN	0.670000
75%	0.670000	0.500000	NaN	0.670000
max	1.000000	1.000000	NaN	1.000000

```
[ ]: fig,ax = plt.subplots(figsize =(15,15))
philly_points.hist(ax = ax)#plotting histogram of the dataset to make sure all data is fine
```

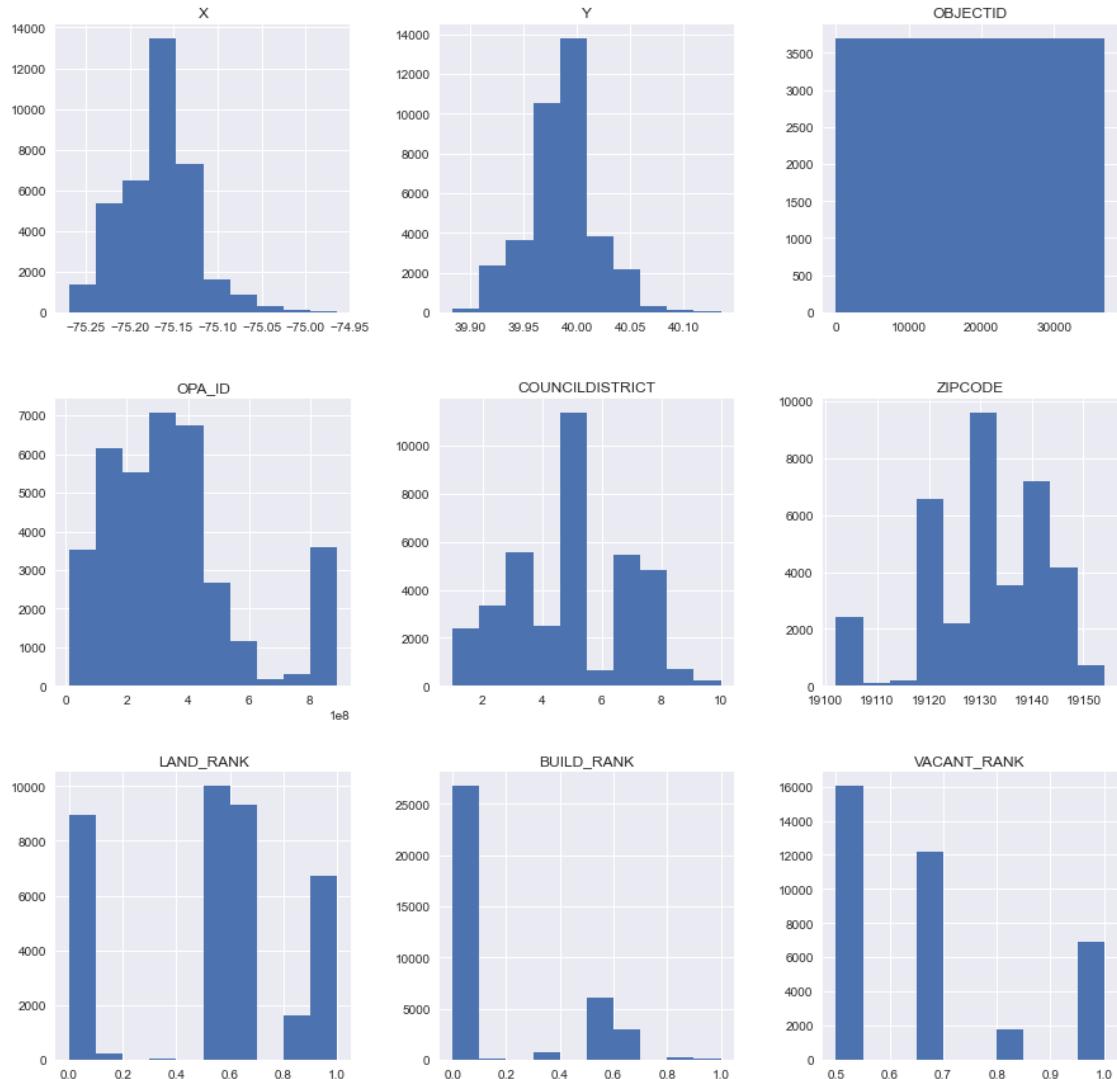
```
/var/folders/6p/wpw9qm157530xkxqkkhprrf40000gn/T/ipykernel_74088/469468503.py:2:
UserWarning: To output multiple subplots, the figure containing the passed axes
is being cleared
    philly_points.hist(ax = ax)
```

```
[ ]: array([ [,
<AxesSubplot:title={'center':'Y'}>,
```

```

<AxesSubplot:title={'center':'OBJECTID'}>,
[<AxesSubplot:title={'center':'OPA_ID'}>,
<AxesSubplot:title={'center':'COUNCILDISTRICT'}>,
<AxesSubplot:title={'center':'ZIPCODE'}>,
[<AxesSubplot:title={'center':'LAND_RANK'}>,
<AxesSubplot:title={'center':'BUILD_RANK'}>,
<AxesSubplot:title={'center':'VACANT_RANK'}>]] , dtype=object)

```



```

[ ]: #X and Y columns are latitude and longitude columns. Both columns need to be combined for into a geometry column for geo pandas to read and plot the data
      ↵crs = {'init': 'epsg:4326'}
      ↵geometry = [Point(xy) for xy in zip(philly_points["X"], philly_points["Y"])]
      ↵geometry[:3]

```

```
[ ]: [<shapely.geometry.point.Point at 0x2e24cba30>,
       <shapely.geometry.point.Point at 0x2e99f90f0>,
       <shapely.geometry.point.Point at 0x2e24e1d50>]

[ ]: philly_points = gpd.GeoDataFrame(philly_points,
                                      crs = crs,
                                      geometry = geometry)

philly_points.head()

/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/pyproj/crs/crs.py:131: FutureWarning: '+init=<authority>:<code>' syntax
is deprecated. '<authority>:<code>' is the preferred initialization method. When
making the change, be mindful of axis order changes:
https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-proj-6
    in_crs_string = _prepare_from_proj_string(in_crs_string)

[ ]:          X            Y   OBJECTID      ADDRESS \
0 -75.178904  39.934505        1    2041 REED ST
1 -75.164548  39.988160        2    2233 N UBER ST
2 -75.180480  39.978561        3   1460 N MARSTON ST
3 -75.186579  40.006852        4   3241 SUGDEN'S ROW
4 -75.238794  39.954902        5    5816 PINE ST

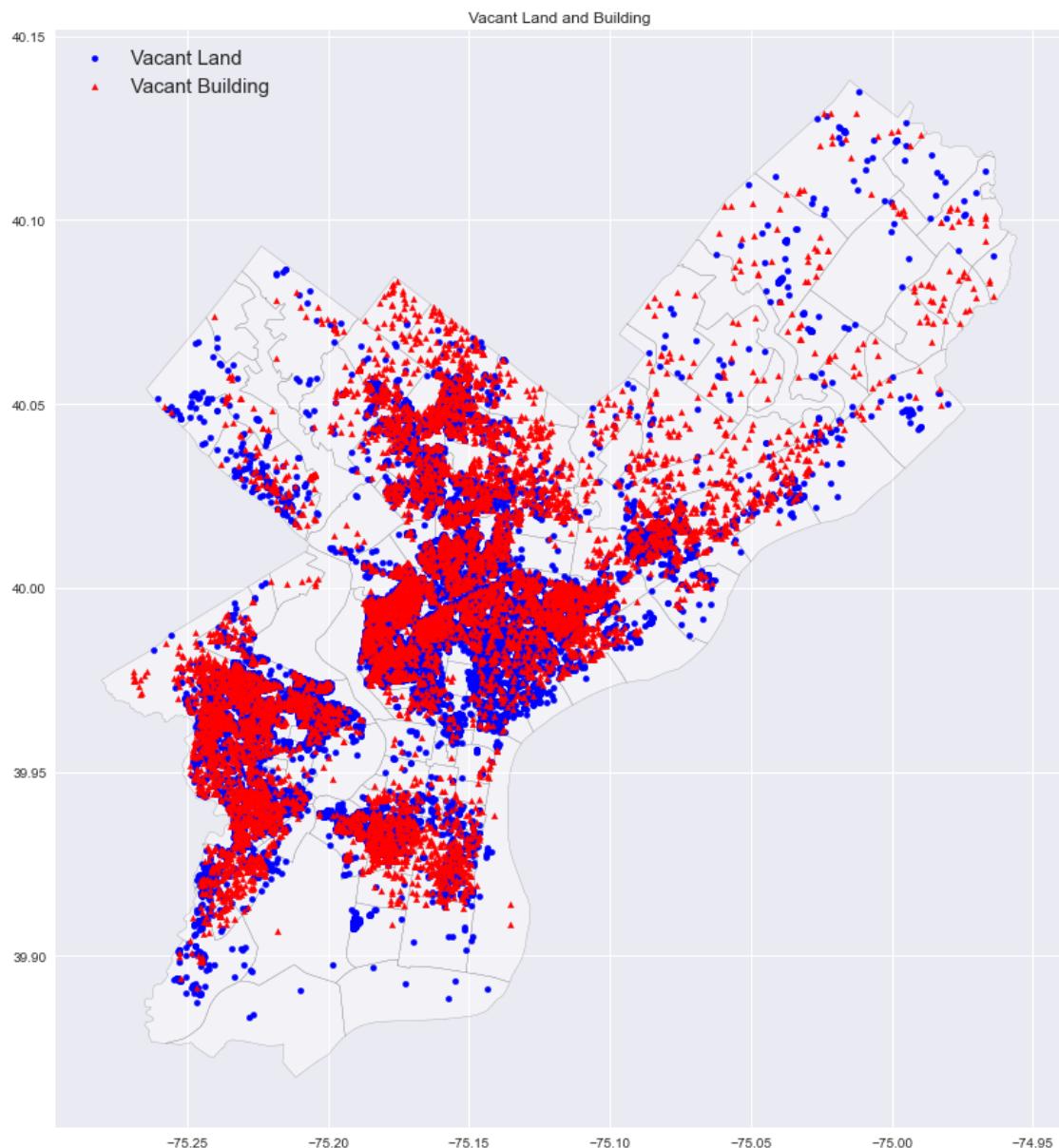
          OWNER1      OWNER2      BLDG_DESC \
0  GREATER DELIVERANCE TEMPL      NaN  VAC LAND COMM. < ACRE
1           CITY OF PHILA      NaN  VAC LAND RES < ACRE
2  PHILADELPHIA HOUSING AUTH      NaN  VAC LAND RES < ACRE
3           JORDAN MARIA      NaN  VAC LAND RES < ACRE
4        WALSH JAMES  LUBLIN WILLIAM H  ROW 2 STY MASONRY

      OPA_ID LNIADDRESSKEY COUNCILDISTRICT ZONINGBASEDISTRICT ZIPCODE \
0  885396760.0        498086            2        CMX-2  19146.0
1  162113701.0        581713            5        RSA-5  19132.0
2  292083110.0        415511            5        RSA-5  19121.0
3  382209500.0        557317            4        RSA-5  19129.0
4  604178400.0        485528            3        RM-1  19143.0

  LAND_RANK BUILD_RANK VACANT_FLAG VACANT_RANK      geometry
0      0.67        0.0     Land      0.67  POINT (-75.17890 39.93451)
1      0.50        0.0     Land      0.50  POINT (-75.16455 39.98816)
2      0.50        0.0     Land      0.50  POINT (-75.18048 39.97856)
3      1.00        0.0     Land      1.00  POINT (-75.18658 40.00685)
4      0.00        1.0  Building      1.00  POINT (-75.23879 39.95490)
```

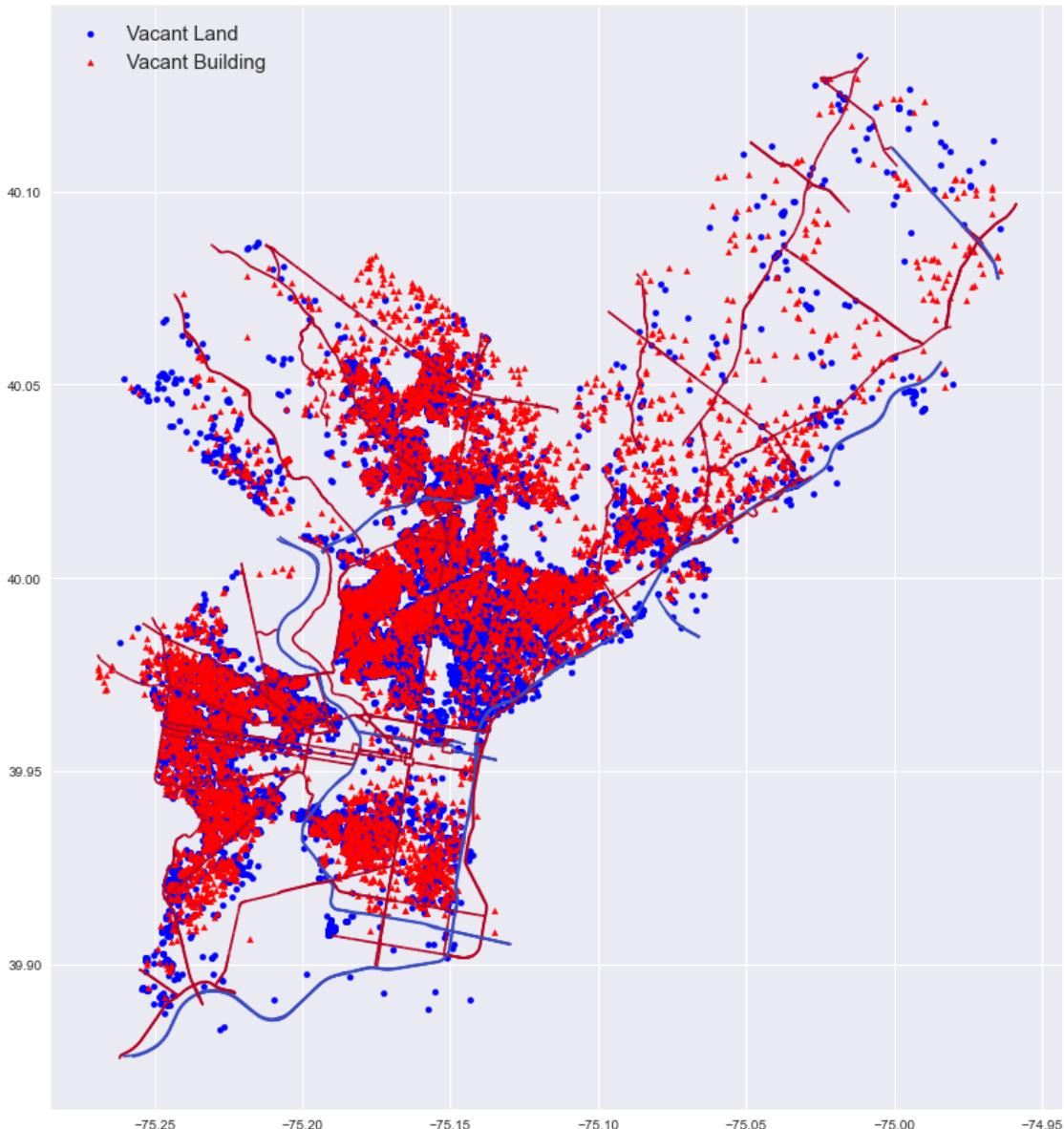
```
[ ]: #Plotting Vacant land and Vacant buildings
fig, ax = plt.subplots(figsize =(15,15))
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white", ↴
    ↪edgecolor='black')
plt.title('Vacant Land and Building')
philly_points[philly_points['VACANT_FLAG'] == 'Land'].plot(ax = ax, markersize= ↪
    ↪= 20, color = "blue", marker = "o", label = "Vacant Land")
philly_points[philly_points['VACANT_FLAG'] == 'Building'].plot(ax = ax, ↪
    ↪markersize = 20, color = "red", marker = "^", label = "Vacant Building")
plt.legend(prop = {'size' : 15})
```

[]: <matplotlib.legend.Legend at 0x2df0e3f10>



```
[ ]: #vacant land and vacant building layer on top of main roads in philadelphia
fig, ax = plt.subplots(figsize =(15,15))
plt.title("Predicted Vacant Land & Building with Main Roads")
main_roads.plot(ax = ax, column='fclass', cmap = 'coolwarm')
philly_points[philly_points['VACANT_FLAG'] == 'Land'].plot(ax = ax, markersize=20, color = "blue", marker = "o", label = "Vacant Land")
philly_points[philly_points['VACANT_FLAG'] == 'Building'].plot(ax = ax, markersize = 20, color = "red", marker = "^", label = "Vacant Building")
plt.legend(prop = {'size' : 15})
```

```
[ ]: <matplotlib.legend.Legend at 0x1d9b1b280>
```



0.2.7 American Community Survey(ACS)

Census Bureau has data that is in the following levels: states > counties > tracts > blockgroups > blocks

Tracts are fairly homogenous, when tract is beyond 800 people the tract is split up Blockgroup contains blocks. Block groups have between 250 and 550 housing units. Census block is the smalest geographic census unit. Blocks can be bounded by visible features—such as streets—or by invisible boundaries, such as city limits.Census blocks are often the same as ordinary city blocks. Census blocks change every decade.

ACS Occupancy, ACS Vacancy Type, ACS Population were used here. The major takeaways are:
- Percentage vacant lots are skewed right - There are around 12% vacant lots - Percentage of vacant lot graph is different from number of vacant lots - “Other” type of vacant lot is the highest percentage of vacant lots - Around 6.7% “other” vacant of the total places

Feature Engineering used on this dataset were: - percentage vacant lots of total places - Percentage “other” vacant lots of total places - population density in each block group in km²

```
[ ]: censusdata.search('acs5', 2019, 'label', 'vacant')# finding 5 year ACS estimates  
→from 2015 with vacant in the concept
```

```
[ ]: [('B25002_003E', 'OCCUPANCY STATUS', 'Estimate!!Total:!!Vacant'),  
      ('B25004_008E', 'VACANCY STATUS', 'Estimate!!Total:!!Other vacant'),  
      ('B25005_002E',  
       'VACANT - CURRENT RESIDENCE ELSEWHERE',  
       'Estimate!!Total:!!Vacant - current residence elsewhere'),  
      ('B25005_003E',  
       'VACANT - CURRENT RESIDENCE ELSEWHERE',  
       'Estimate!!Total:!!All other vacant units')]
```

```
[ ]: states = censusdata.geographies(censusdata.censusgeo([('state', '*'))], 'acs5',  
→2015)#printing name of state and code in the dataset  
print(states['Pennsylvania']) # pennsylvania is code 42
```

Summary level: 040, state:42

```
[ ]: counties = censusdata.geographies(censusdata.censusgeo([('state', '42'),  
→('county', '*'))], 'acs5', 2015)#all counties in Pennsylvania.  
#We will be using Philadelphia county with the county code 101
```

```
[ ]: censusdata.geographies(censusdata.censusgeo([('state', '42'), ('county',  
→'101')]), 'acs5', 2015)# selecting philadelphia county only  
# we will be using these location lode to pull data from teh census package
```

```
[ ]: {'Philadelphia County, Pennsylvania': censusgeo([('state', '42'), ('county', '101'))]}
```

```
[ ]: list_blockgroup = censusdata.geographies(censusdata.censusgeo([('state', '42'), ('county', '101'), ('block group', '*')]), 'acs5', 2019)
# all block group in philadelphia county
```

```
[ ]: list_blockgroup
```

```
[ ]: {'Block Group 1, Census Tract 9807, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '980700'), ('block
group', '1'))),
'Block Group 3, Census Tract 27.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '002701'), ('block
group', '3'))),
'Block Group 2, Census Tract 337.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033701'), ('block
group', '2'))),
'Block Group 3, Census Tract 337.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033701'), ('block
group', '3'))),
'Block Group 2, Census Tract 205, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '020500'), ('block
group', '2'))),
'Block Group 1, Census Tract 218, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '021800'), ('block
group', '1'))),
'Block Group 2, Census Tract 218, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '021800'), ('block
group', '2'))),
'Block Group 1, Census Tract 219, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '021900'), ('block
group', '1'))),
'Block Group 4, Census Tract 349, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '034900'), ('block
group', '4'))),
'Block Group 2, Census Tract 349, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '034900'), ('block
group', '2'))),
'Block Group 1, Census Tract 349, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '034900'), ('block
group', '1'))),
'Block Group 2, Census Tract 269, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '026900'), ('block
group', '2'))),
'Block Group 1, Census Tract 269, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '026900'), ('block
```

```

group', '1'))),
'Block Group 3, Census Tract 292, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '029200'), ('block
group', '3'))),
'Block Group 1, Census Tract 292, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '029200'), ('block
group', '1'))),
'Block Group 6, Census Tract 149, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014900'), ('block
group', '6'))),
'Block Group 2, Census Tract 149, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014900'), ('block
group', '2'))),
'Block Group 4, Census Tract 149, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014900'), ('block
group', '4'))),
'Block Group 1, Census Tract 149, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014900'), ('block
group', '1'))),
'Block Group 2, Census Tract 158, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '015800'), ('block
group', '2'))),
'Block Group 1, Census Tract 81.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '008102'), ('block
group', '1'))),
'Block Group 5, Census Tract 81.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '008102'), ('block
group', '5'))),
'Block Group 3, Census Tract 81.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '008102'), ('block
group', '3'))),
'Block Group 1, Census Tract 151.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '015102'), ('block
group', '1'))),
'Block Group 3, Census Tract 151.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '015102'), ('block
group', '3'))),
'Block Group 3, Census Tract 365.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '036501'), ('block
group', '3'))),
'Block Group 2, Census Tract 365.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '036501'), ('block
group', '2'))),
'Block Group 1, Census Tract 9804, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '980400'), ('block
group', '1'))),
'Block Group 2, Census Tract 134.02, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '013402'), ('block
group', '2'))),
'Block Group 1, Census Tract 134.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '013402'), ('block
group', '1'))),
'Block Group 1, Census Tract 134.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '013401'), ('block
group', '1'))),
'Block Group 2, Census Tract 134.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '013401'), ('block
group', '2'))),
'Block Group 2, Census Tract 136.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '013601'), ('block
group', '2'))),
'Block Group 3, Census Tract 379, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '037900'), ('block
group', '3'))),
'Block Group 1, Census Tract 379, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '037900'), ('block
group', '1'))),
'Block Group 6, Census Tract 379, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '037900'), ('block
group', '6'))),
'Block Group 2, Census Tract 382, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '038200'), ('block
group', '2'))),
'Block Group 4, Census Tract 158, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '015800'), ('block
group', '4'))),
'Block Group 6, Census Tract 158, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '015800'), ('block
group', '6'))),
'Block Group 5, Census Tract 158, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '015800'), ('block
group', '5'))),
'Block Group 3, Census Tract 158, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '015800'), ('block
group', '3'))),
'Block Group 1, Census Tract 158, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '015800'), ('block
group', '1'))),
'Block Group 4, Census Tract 164, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '016400'), ('block
group', '4'))),
'Block Group 1, Census Tract 164, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '016400'), ('block
group', '1'))),

```

```

'Block Group 2, Census Tract 164, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016400'), ('block
group', '2'))),
'Block Group 3, Census Tract 164, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016400'), ('block
group', '3'))),
'Block Group 3, Census Tract 338, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033800'), ('block
group', '3'))),
'Block Group 2, Census Tract 338, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033800'), ('block
group', '2'))),
'Block Group 1, Census Tract 338, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033800'), ('block
group', '1'))),
'Block Group 3, Census Tract 344, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '034400'), ('block
group', '3'))),
'Block Group 4, Census Tract 344, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '034400'), ('block
group', '4'))),
'Block Group 2, Census Tract 169.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016902'), ('block
group', '2'))),
'Block Group 1, Census Tract 169.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016902'), ('block
group', '1'))),
'Block Group 3, Census Tract 169.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016902'), ('block
group', '3'))),
'Block Group 3, Census Tract 171, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017100'), ('block
group', '3'))),
'Block Group 1, Census Tract 171, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017100'), ('block
group', '1'))),
'Block Group 2, Census Tract 171, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017100'), ('block
group', '2'))),
'Block Group 3, Census Tract 382, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038200'), ('block
group', '3'))),
'Block Group 1, Census Tract 380, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038000'), ('block
group', '1'))),
'Block Group 1, Census Tract 381, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038100'), ('block
group', '1'))]

```

```

group', '1'))),
'Block Group 2, Census Tract 381, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038100'), ('block
group', '2'))),
'Block Group 2, Census Tract 384, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038400'), ('block
group', '2'))),
'Block Group 1, Census Tract 31, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '003100'), ('block
group', '1'))),
'Block Group 2, Census Tract 31, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '003100'), ('block
group', '2'))),
'Block Group 4, Census Tract 31, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '003100'), ('block
group', '4'))),
'Block Group 2, Census Tract 344, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '034400'), ('block
group', '2'))),
'Block Group 2, Census Tract 347.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '034701'), ('block
group', '2'))),
'Block Group 2, Census Tract 166, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016600'), ('block
group', '2'))),
'Block Group 1, Census Tract 166, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016600'), ('block
group', '1'))),
'Block Group 1, Census Tract 173, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017300'), ('block
group', '1'))),
'Block Group 2, Census Tract 173, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017300'), ('block
group', '2'))),
'Block Group 2, Census Tract 175, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017500'), ('block
group', '2'))),
'Block Group 6, Census Tract 175, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017500'), ('block
group', '6'))),
'Block Group 5, Census Tract 175, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017500'), ('block
group', '5'))),
'Block Group 4, Census Tract 175, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017500'), ('block
group', '4'))),
'Block Group 1, Census Tract 175, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '017500'), ('block
group', '1'))),
'Block Group 2, Census Tract 176.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017602'), ('block
group', '2'))),
'Block Group 3, Census Tract 176.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017602'), ('block
group', '3'))),
'Block Group 2, Census Tract 253, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '025300'), ('block
group', '2'))),
'Block Group 1, Census Tract 253, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '025300'), ('block
group', '1'))),
'Block Group 2, Census Tract 292, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '029200'), ('block
group', '2'))),
'Block Group 1, Census Tract 313, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031300'), ('block
group', '1'))),
'Block Group 2, Census Tract 313, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031300'), ('block
group', '2'))),
'Block Group 3, Census Tract 313, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031300'), ('block
group', '3'))),
'Block Group 6, Census Tract 313, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031300'), ('block
group', '6'))),
'Block Group 5, Census Tract 238, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '023800'), ('block
group', '5'))),
'Block Group 1, Census Tract 238, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '023800'), ('block
group', '1'))),
'Block Group 1, Census Tract 113, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011300'), ('block
group', '1'))),
'Block Group 1, Census Tract 274.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '027402'), ('block
group', '1'))),
'Block Group 4, Census Tract 274.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '027402'), ('block
group', '4'))),
'Block Group 5, Census Tract 274.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '027402'), ('block
group', '5'))),

```

```

'Block Group 2, Census Tract 86.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008601'), ('block
group', '2'))),
'Block Group 1, Census Tract 86.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008601'), ('block
group', '1'))),
'Block Group 2, Census Tract 365.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '036502'), ('block
group', '2'))),
'Block Group 1, Census Tract 365.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '036502'), ('block
group', '1'))),
'Block Group 1, Census Tract 367, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '036700'), ('block
group', '1'))),
'Block Group 2, Census Tract 367, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '036700'), ('block
group', '2'))),
'Block Group 2, Census Tract 376, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '037600'), ('block
group', '2'))),
'Block Group 1, Census Tract 376, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '037600'), ('block
group', '1'))),
'Block Group 1, Census Tract 386, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038600'), ('block
group', '1'))),
'Block Group 2, Census Tract 385, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038500'), ('block
group', '2'))),
'Block Group 3, Census Tract 388, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038800'), ('block
group', '3'))),
'Block Group 1, Census Tract 385, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038500'), ('block
group', '1'))),
'Block Group 2, Census Tract 387, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038700'), ('block
group', '2'))),
'Block Group 4, Census Tract 388, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038800'), ('block
group', '4'))),
'Block Group 2, Census Tract 388, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038800'), ('block
group', '2'))),
'Block Group 1, Census Tract 9803, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '980300'), ('block
group', '1'))]

```

```

group', '1'))),
'Block Group 2, Census Tract 74, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '007400'), ('block
group', '2'))),
'Block Group 1, Census Tract 74, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '007400'), ('block
group', '1'))),
'Block Group 3, Census Tract 74, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '007400'), ('block
group', '3'))),
'Block Group 4, Census Tract 74, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '007400'), ('block
group', '4'))),
'Block Group 5, Census Tract 74, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '007400'), ('block
group', '5'))),
'Block Group 1, Census Tract 92, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '009200'), ('block
group', '1'))),
'Block Group 2, Census Tract 92, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '009200'), ('block
group', '2'))),
'Block Group 2, Census Tract 22, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002200'), ('block
group', '2'))),
'Block Group 3, Census Tract 22, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002200'), ('block
group', '3'))),
'Block Group 1, Census Tract 22, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002200'), ('block
group', '1'))),
'Block Group 1, Census Tract 23, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002300'), ('block
group', '1'))),
'Block Group 2, Census Tract 23, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002300'), ('block
group', '2'))),
'Block Group 1, Census Tract 41.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '004101'), ('block
group', '1'))),
'Block Group 3, Census Tract 41.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '004101'), ('block
group', '3'))),
'Block Group 4, Census Tract 42.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '004202'), ('block
group', '4'))),
'Block Group 2, Census Tract 42.02, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '004202'), ('block
group', '2'))),
'Block Group 3, Census Tract 42.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '004202'), ('block
group', '3'))),
'Block Group 3, Census Tract 55, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '005500'), ('block
group', '3'))),
'Block Group 1, Census Tract 69, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '006900'), ('block
group', '1'))),
'Block Group 2, Census Tract 69, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '006900'), ('block
group', '2'))),
'Block Group 3, Census Tract 70, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '007000'), ('block
group', '3'))),
'Block Group 1, Census Tract 70, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '007000'), ('block
group', '1'))),
'Block Group 5, Census Tract 70, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '007000'), ('block
group', '5'))),
'Block Group 4, Census Tract 80, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008000'), ('block
group', '4'))),
'Block Group 3, Census Tract 80, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008000'), ('block
group', '3'))),
'Block Group 1, Census Tract 94, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009400'), ('block
group', '1'))),
'Block Group 2, Census Tract 94, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009400'), ('block
group', '2'))),
'Block Group 3, Census Tract 94, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009400'), ('block
group', '3'))),
'Block Group 3, Census Tract 95, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009500'), ('block
group', '3'))),
'Block Group 4, Census Tract 95, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009500'), ('block
group', '4'))),
'Block Group 1, Census Tract 91, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009100'), ('block
group', '1'))),

```

```

'Block Group 1, Census Tract 103, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '010300'), ('block
group', '1'))),
'Block Group 3, Census Tract 105, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '010500'), ('block
group', '3'))),
'Block Group 1, Census Tract 105, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '010500'), ('block
group', '1'))),
'Block Group 2, Census Tract 3, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '000300'), ('block
group', '2'))),
'Block Group 1, Census Tract 3, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '000300'), ('block
group', '1'))),
'Block Group 1, Census Tract 24, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '002400'), ('block
group', '1'))),
'Block Group 4, Census Tract 24, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '002400'), ('block
group', '4'))),
'Block Group 1, Census Tract 28.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '002801'), ('block
group', '1'))),
'Block Group 3, Census Tract 28.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '002801'), ('block
group', '3'))),
'Block Group 1, Census Tract 8.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '000801'), ('block
group', '1'))),
'Block Group 2, Census Tract 11.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001101'), ('block
group', '2'))),
'Block Group 1, Census Tract 11.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001101'), ('block
group', '1'))),
'Block Group 3, Census Tract 11.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001101'), ('block
group', '3'))),
'Block Group 3, Census Tract 274.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '027402'), ('block
group', '3'))),
'Block Group 2, Census Tract 113, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '011300'), ('block
group', '2'))),
'Block Group 1, Census Tract 114, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '011400'), ('block
group', '1'))]

```

```

group', '1'))),
'Block Group 4, Census Tract 114, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '011400'), ('block
group', '4'))),
'Block Group 5, Census Tract 114, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '011400'), ('block
group', '5'))),
'Block Group 3, Census Tract 114, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '011400'), ('block
group', '3'))),
'Block Group 2, Census Tract 325, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '032500'), ('block
group', '2'))),
'Block Group 4, Census Tract 325, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '032500'), ('block
group', '4'))),
'Block Group 3, Census Tract 325, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '032500'), ('block
group', '3'))),
'Block Group 1, Census Tract 325, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '032500'), ('block
group', '1'))),
'Block Group 1, Census Tract 14, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '001400'), ('block
group', '1'))),
'Block Group 2, Census Tract 14, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '001400'), ('block
group', '2'))),
'Block Group 1, Census Tract 37.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003701'), ('block
group', '1'))),
'Block Group 4, Census Tract 37.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003701'), ('block
group', '4'))),
'Block Group 5, Census Tract 37.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003701'), ('block
group', '5'))),
'Block Group 2, Census Tract 37.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003701'), ('block
group', '2'))),
'Block Group 1, Census Tract 287, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028700'), ('block
group', '1'))),
'Block Group 2, Census Tract 287, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028700'), ('block
group', '2'))),
'Block Group 1, Census Tract 9891, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '989100'), ('block
group', '1'))),
'Block Group 3, Census Tract 291, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '029100'), ('block
group', '3'))),
'Block Group 2, Census Tract 290, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '029000'), ('block
group', '2'))),
'Block Group 1, Census Tract 301, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030100'), ('block
group', '1'))),
'Block Group 5, Census Tract 301, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030100'), ('block
group', '5'))),
'Block Group 3, Census Tract 301, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030100'), ('block
group', '3'))),
'Block Group 1, Census Tract 137, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '013700'), ('block
group', '1'))),
'Block Group 3, Census Tract 137, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '013700'), ('block
group', '3'))),
'Block Group 2, Census Tract 137, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '013700'), ('block
group', '2'))),
'Block Group 3, Census Tract 115, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011500'), ('block
group', '3'))),
'Block Group 1, Census Tract 115, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011500'), ('block
group', '1'))),
'Block Group 3, Census Tract 259, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '025900'), ('block
group', '3'))),
'Block Group 5, Census Tract 259, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '025900'), ('block
group', '5'))),
'Block Group 1, Census Tract 290, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '029000'), ('block
group', '1'))),
'Block Group 3, Census Tract 290, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '029000'), ('block
group', '3'))),
'Block Group 3, Census Tract 369, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '036900'), ('block
group', '3'))),

```

```

'Block Group 1, Census Tract 369, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '036900'), ('block
group', '1'))),
'Block Group 3, Census Tract 373, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '037300'), ('block
group', '3'))),
'Block Group 1, Census Tract 373, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '037300'), ('block
group', '1'))),
'Block Group 2, Census Tract 372, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '037200'), ('block
group', '2'))),
'Block Group 1, Census Tract 372, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '037200'), ('block
group', '1'))),
'Block Group 1, Census Tract 383, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038300'), ('block
group', '1'))),
'Block Group 2, Census Tract 383, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '038300'), ('block
group', '2'))),
'Block Group 6, Census Tract 390, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '039000'), ('block
group', '6'))),
'Block Group 5, Census Tract 24, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '002400'), ('block
group', '5'))),
'Block Group 2, Census Tract 24, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '002400'), ('block
group', '2'))),
'Block Group 2, Census Tract 358, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '035800'), ('block
group', '2'))),
'Block Group 3, Census Tract 358, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '035800'), ('block
group', '3'))),
'Block Group 1, Census Tract 358, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '035800'), ('block
group', '1'))),
'Block Group 1, Census Tract 259, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '025900'), ('block
group', '1'))),
'Block Group 2, Census Tract 259, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '025900'), ('block
group', '2'))),
'Block Group 6, Census Tract 259, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '025900'), ('block
group', '6'))),

```

```

group', '6'))),
'Block Group 4, Census Tract 259, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '025900'), ('block
group', '4'))),
'Block Group 2, Census Tract 208, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '020800'), ('block
group', '2'))),
'Block Group 3, Census Tract 218, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '021800'), ('block
group', '3'))),
'Block Group 1, Census Tract 9801, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '980100'), ('block
group', '1'))),
'Block Group 1, Census Tract 100, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010000'), ('block
group', '1'))),
'Block Group 2, Census Tract 100, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010000'), ('block
group', '2'))),
'Block Group 3, Census Tract 100, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010000'), ('block
group', '3'))),
'Block Group 4, Census Tract 100, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010000'), ('block
group', '4'))),
'Block Group 2, Census Tract 101, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010100'), ('block
group', '2'))),
'Block Group 6, Census Tract 101, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010100'), ('block
group', '6'))),
'Block Group 3, Census Tract 101, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010100'), ('block
group', '3'))),
'Block Group 5, Census Tract 101, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010100'), ('block
group', '5'))),
'Block Group 7, Census Tract 101, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010100'), ('block
group', '7'))),
'Block Group 4, Census Tract 107, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010700'), ('block
group', '4'))),
'Block Group 2, Census Tract 133, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '013300'), ('block
group', '2'))),
'Block Group 1, Census Tract 133, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '013300'), ('block
group', '1'))),
'Block Group 1, Census Tract 77, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '007700'), ('block
group', '1'))),
'Block Group 3, Census Tract 110, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011000'), ('block
group', '3'))),
'Block Group 1, Census Tract 110, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011000'), ('block
group', '1'))),
'Block Group 4, Census Tract 112, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011200'), ('block
group', '4'))),
'Block Group 3, Census Tract 112, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011200'), ('block
group', '3'))),
'Block Group 2, Census Tract 112, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011200'), ('block
group', '2'))),
'Block Group 5, Census Tract 112, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011200'), ('block
group', '5'))),
'Block Group 6, Census Tract 112, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011200'), ('block
group', '6'))),
'Block Group 1, Census Tract 41.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '004102'), ('block
group', '1'))),
'Block Group 2, Census Tract 41.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '004102'), ('block
group', '2'))),
'Block Group 4, Census Tract 41.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '004102'), ('block
group', '4'))),
'Block Group 1, Census Tract 363.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '036301'), ('block
group', '1'))),
'Block Group 2, Census Tract 353.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '035302'), ('block
group', '2'))),
'Block Group 1, Census Tract 355, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '035500'), ('block
group', '1'))),
'Block Group 1, Census Tract 362.03, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '036203'), ('block
group', '1'))),

```

```

'Block Group 2, Census Tract 362.03, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '036203'), ('block
group', '2'))),
'Block Group 1, Census Tract 364, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '036400'), ('block
group', '1'))),
'Block Group 1, Census Tract 205, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '020500'), ('block
group', '1'))),
'Block Group 2, Census Tract 87.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008701'), ('block
group', '2'))),
'Block Group 4, Census Tract 81.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008102'), ('block
group', '4'))),
'Block Group 2, Census Tract 81.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008102'), ('block
group', '2'))),
'Block Group 1, Census Tract 167.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016701'), ('block
group', '1'))),
'Block Group 3, Census Tract 167.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016701'), ('block
group', '3'))),
'Block Group 3, Census Tract 361, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '036100'), ('block
group', '3'))),
'Block Group 1, Census Tract 167.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016702'), ('block
group', '1'))),
'Block Group 2, Census Tract 167.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016702'), ('block
group', '2'))),
'Block Group 3, Census Tract 279.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '027901'), ('block
group', '3'))),
'Block Group 4, Census Tract 279.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '027901'), ('block
group', '4'))),
'Block Group 1, Census Tract 279.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '027902'), ('block
group', '1'))),
'Block Group 1, Census Tract 315.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '031502'), ('block
group', '1'))),
'Block Group 2, Census Tract 315.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '031502'), ('block
group', '2'))]

```

```

group', '2'))),
'Block Group 1, Census Tract 314.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031401'), ('block
group', '1'))),
'Block Group 3, Census Tract 314.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031401'), ('block
group', '3'))),
'Block Group 6, Census Tract 67, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '006700'), ('block
group', '6'))),
'Block Group 1, Census Tract 170, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '017000'), ('block
group', '1'))),
'Block Group 3, Census Tract 243, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '024300'), ('block
group', '3'))),
'Block Group 2, Census Tract 243, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '024300'), ('block
group', '2'))),
'Block Group 4, Census Tract 243, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '024300'), ('block
group', '4'))),
'Block Group 1, Census Tract 243, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '024300'), ('block
group', '1'))),
'Block Group 3, Census Tract 276, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '027600'), ('block
group', '3'))),
'Block Group 3, Census Tract 104, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010400'), ('block
group', '3'))),
'Block Group 2, Census Tract 104, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010400'), ('block
group', '2'))),
'Block Group 4, Census Tract 104, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '010400'), ('block
group', '4'))),
'Block Group 1, Census Tract 132, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '013200'), ('block
group', '1'))),
'Block Group 3, Census Tract 170, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '017000'), ('block
group', '3'))),
'Block Group 3, Census Tract 111, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '011100'), ('block
group', '3'))),
'Block Group 1, Census Tract 111, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '011100'), ('block
group', '1'))),
'Block Group 4, Census Tract 111, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011100'), ('block
group', '4'))),
'Block Group 5, Census Tract 111, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '011100'), ('block
group', '5'))),
'Block Group 1, Census Tract 82, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008200'), ('block
group', '1'))),
'Block Group 6, Census Tract 82, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008200'), ('block
group', '6'))),
'Block Group 2, Census Tract 82, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008200'), ('block
group', '2'))),
'Block Group 4, Census Tract 82, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008200'), ('block
group', '4'))),
'Block Group 5, Census Tract 82, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008200'), ('block
group', '5'))),
'Block Group 3, Census Tract 82, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008200'), ('block
group', '3'))),
'Block Group 7, Census Tract 82, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008200'), ('block
group', '7'))),
'Block Group 2, Census Tract 83.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008301'), ('block
group', '2'))),
'Block Group 1, Census Tract 83.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008301'), ('block
group', '1'))),
'Block Group 3, Census Tract 83.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008301'), ('block
group', '3'))),
'Block Group 2, Census Tract 83.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008302'), ('block
group', '2'))),
'Block Group 3, Census Tract 83.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008302'), ('block
group', '3'))),
'Block Group 1, Census Tract 147, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '014700'), ('block
group', '1'))),

```

```

'Block Group 2, Census Tract 156, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '015600'), ('block
group', '2'))),
'Block Group 1, Census Tract 163, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016300'), ('block
group', '1'))),
'Block Group 3, Census Tract 163, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016300'), ('block
group', '3'))),
'Block Group 4, Census Tract 163, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016300'), ('block
group', '4'))),
'Block Group 2, Census Tract 178, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017800'), ('block
group', '2'))),
'Block Group 2, Census Tract 4.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '000402'), ('block
group', '2'))),
'Block Group 1, Census Tract 4.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '000401'), ('block
group', '1'))),
'Block Group 2, Census Tract 114, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '011400'), ('block
group', '2'))),
'Block Group 2, Census Tract 7, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '000700'), ('block
group', '2'))),
'Block Group 3, Census Tract 7, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '000700'), ('block
group', '3'))),
'Block Group 1, Census Tract 18, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001800'), ('block
group', '1'))),
'Block Group 2, Census Tract 18, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001800'), ('block
group', '2'))),
'Block Group 1, Census Tract 21, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '002100'), ('block
group', '1'))),
'Block Group 1, Census Tract 257, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '025700'), ('block
group', '1'))),
'Block Group 2, Census Tract 247, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '024700'), ('block
group', '2'))),
'Block Group 4, Census Tract 38, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '003800'), ('block
group', '4')))

```

```

group', '4'))),
'Block Group 1, Census Tract 38, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003800'), ('block
group', '1'))),
'Block Group 2, Census Tract 38, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003800'), ('block
group', '2'))),
'Block Group 1, Census Tract 125, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '012500'), ('block
group', '1'))),
'Block Group 4, Census Tract 210, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '021000'), ('block
group', '4'))),
'Block Group 2, Census Tract 210, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '021000'), ('block
group', '2'))),
'Block Group 5, Census Tract 210, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '021000'), ('block
group', '5'))),
'Block Group 1, Census Tract 347.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034702'), ('block
group', '1'))),
'Block Group 4, Census Tract 337.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '033702'), ('block
group', '4'))),
'Block Group 3, Census Tract 337.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '033702'), ('block
group', '3'))),
'Block Group 3, Census Tract 315.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031502'), ('block
group', '3'))),
'Block Group 1, Census Tract 357.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035701'), ('block
group', '1'))),
'Block Group 2, Census Tract 40.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '004002'), ('block
group', '2'))),
'Block Group 2, Census Tract 42.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '004201'), ('block
group', '2'))),
'Block Group 3, Census Tract 42.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '004201'), ('block
group', '3'))),
'Block Group 4, Census Tract 42.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '004201'), ('block
group', '4'))),
'Block Group 1, Census Tract 42.01, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '004201'), ('block
group', '1'))),
'Block Group 1, Census Tract 78, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '007800'), ('block
group', '1'))),
'Block Group 2, Census Tract 78, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '007800'), ('block
group', '2'))),
'Block Group 2, Census Tract 93, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009300'), ('block
group', '2'))),
'Block Group 1, Census Tract 93, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009300'), ('block
group', '1'))),
'Block Group 4, Census Tract 93, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009300'), ('block
group', '4'))),
'Block Group 3, Census Tract 93, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009300'), ('block
group', '3'))),
'Block Group 5, Census Tract 93, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009300'), ('block
group', '5'))),
'Block Group 2, Census Tract 321, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '032100'), ('block
group', '2'))),
'Block Group 1, Census Tract 321, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '032100'), ('block
group', '1'))),
'Block Group 1, Census Tract 214, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '021400'), ('block
group', '1'))),
'Block Group 5, Census Tract 214, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '021400'), ('block
group', '5'))),
'Block Group 2, Census Tract 214, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '021400'), ('block
group', '2'))),
'Block Group 3, Census Tract 214, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '021400'), ('block
group', '3'))),
'Block Group 3, Census Tract 8.04, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '000804'), ('block
group', '3'))),
'Block Group 2, Census Tract 8.04, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '000804'), ('block
group', '2'))),

```

```
'Block Group 1, Census Tract 8.04, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '000804'), ('block  
group', '1'))),  
'Block Group 2, Census Tract 172.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '017202'), ('block  
group', '2'))),  
'Block Group 4, Census Tract 172.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '017202'), ('block  
group', '4'))),  
'Block Group 1, Census Tract 172.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '017202'), ('block  
group', '1'))),  
'Block Group 3, Census Tract 274.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '027401'), ('block  
group', '3'))),  
'Block Group 1, Census Tract 274.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '027401'), ('block  
group', '1'))),  
'Block Group 1, Census Tract 201.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '020101'), ('block  
group', '1'))),  
'Block Group 3, Census Tract 201.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '020101'), ('block  
group', '3'))),  
'Block Group 3, Census Tract 201.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '020102'), ('block  
group', '3'))),  
'Block Group 1, Census Tract 201.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '020102'), ('block  
group', '1'))),  
'Block Group 2, Census Tract 201.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '020102'), ('block  
group', '2'))),  
'Block Group 4, Census Tract 311.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '031101'), ('block  
group', '4'))),  
'Block Group 1, Census Tract 311.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '031101'), ('block  
group', '1'))),  
'Block Group 3, Census Tract 10.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '001002'), ('block  
group', '3'))),  
'Block Group 2, Census Tract 10.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '001002'), ('block  
group', '2'))),  
'Block Group 2, Census Tract 10.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '001001'), ('block
```

```

group', '2'))),
'Block Group 2, Census Tract 359, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035900'), ('block
group', '2'))),
'Block Group 1, Census Tract 359, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035900'), ('block
group', '1'))),
'Block Group 2, Census Tract 362.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '036201'), ('block
group', '2'))),
'Block Group 2, Census Tract 357.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035701'), ('block
group', '2'))),
'Block Group 1, Census Tract 357.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035702'), ('block
group', '1'))),
'Block Group 2, Census Tract 357.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035702'), ('block
group', '2'))),
'Block Group 4, Census Tract 356.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035601'), ('block
group', '4'))),
'Block Group 3, Census Tract 356.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035601'), ('block
group', '3'))),
'Block Group 1, Census Tract 356.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035601'), ('block
group', '1'))),
'Block Group 2, Census Tract 180.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '018002'), ('block
group', '2'))),
'Block Group 4, Census Tract 180.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '018002'), ('block
group', '4'))),
'Block Group 2, Census Tract 88.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '008801'), ('block
group', '2'))),
'Block Group 2, Census Tract 122.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '012201'), ('block
group', '2'))),
'Block Group 1, Census Tract 98.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '009801'), ('block
group', '1'))),
'Block Group 2, Census Tract 98.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '009801'), ('block
group', '2'))),
'Block Group 2, Census Tract 71.01, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '007101'), ('block
group', '2'))),
'Block Group 3, Census Tract 71.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '007101'), ('block
group', '3'))),
'Block Group 1, Census Tract 326, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '032600'), ('block
group', '1'))),
'Block Group 1, Census Tract 245, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '024500'), ('block
group', '1'))),
'Block Group 4, Census Tract 245, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '024500'), ('block
group', '4'))),
'Block Group 2, Census Tract 245, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '024500'), ('block
group', '2'))),
'Block Group 4, Census Tract 254, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '025400'), ('block
group', '4'))),
'Block Group 1, Census Tract 254, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '025400'), ('block
group', '1'))),
'Block Group 2, Census Tract 311.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031102'), ('block
group', '2'))),
'Block Group 1, Census Tract 311.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031102'), ('block
group', '1'))),
'Block Group 1, Census Tract 195.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '019502'), ('block
group', '1'))),
'Block Group 2, Census Tract 195.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '019502'), ('block
group', '2'))),
'Block Group 1, Census Tract 195.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '019501'), ('block
group', '1'))),
'Block Group 3, Census Tract 195.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '019501'), ('block
group', '3'))),
'Block Group 2, Census Tract 289.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '028901'), ('block
group', '2'))),
'Block Group 1, Census Tract 289.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '028901'), ('block
group', '1'))),

```

'Block Group 1, Census Tract 289.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028902'), ('block group', '1'))),

'Block Group 3, Census Tract 289.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028902'), ('block group', '3'))),

'Block Group 4, Census Tract 289.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028902'), ('block group', '4'))),

'Block Group 5, Census Tract 289.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028902'), ('block group', '5'))),

'Block Group 4, Census Tract 331.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033101'), ('block group', '4'))),

'Block Group 2, Census Tract 331.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033101'), ('block group', '2'))),

'Block Group 4, Census Tract 331.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033102'), ('block group', '4'))),

'Block Group 2, Census Tract 331.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033102'), ('block group', '2'))),

'Block Group 4, Census Tract 326, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '032600'), ('block group', '4'))),

'Block Group 7, Census Tract 326, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '032600'), ('block group', '7'))),

'Block Group 6, Census Tract 326, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '032600'), ('block group', '6'))),

'Block Group 2, Census Tract 326, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '032600'), ('block group', '2'))),

'Block Group 3, Census Tract 326, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '032600'), ('block group', '3'))),

'Block Group 1, Census Tract 332, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033200'), ('block group', '1'))),

'Block Group 2, Census Tract 332, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033200'), ('block group', '2'))),

'Block Group 2, Census Tract 120, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '012000'), ('block

```

group', '2'))),
'Block Group 1, Census Tract 121, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '012100'), ('block
group', '1'))),
'Block Group 4, Census Tract 135, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '013500'), ('block
group', '4'))),
'Block Group 2, Census Tract 143, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014300'), ('block
group', '2'))),
'Block Group 1, Census Tract 143, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014300'), ('block
group', '1'))),
'Block Group 3, Census Tract 146, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014600'), ('block
group', '3'))),
'Block Group 2, Census Tract 146, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014600'), ('block
group', '2'))),
'Block Group 5, Census Tract 149, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014900'), ('block
group', '5'))),
'Block Group 3, Census Tract 149, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014900'), ('block
group', '3'))),
'Block Group 4, Census Tract 329, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '032900'), ('block
group', '4'))),
'Block Group 1, Census Tract 329, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '032900'), ('block
group', '1'))),
'Block Group 2, Census Tract 329, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '032900'), ('block
group', '2'))),
'Block Group 2, Census Tract 348.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034802'), ('block
group', '2'))),
'Block Group 3, Census Tract 348.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034802'), ('block
group', '3'))),
'Block Group 3, Census Tract 348.03, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034803'), ('block
group', '3'))),
'Block Group 1, Census Tract 361, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '036100'), ('block
group', '1'))),
'Block Group 2, Census Tract 254, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '025400'), ('block
group', '2'))),
'Block Group 3, Census Tract 254, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '025400'), ('block
group', '3'))),
'Block Group 2, Census Tract 260, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026000'), ('block
group', '2'))),
'Block Group 3, Census Tract 260, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026000'), ('block
group', '3'))),
'Block Group 1, Census Tract 260, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026000'), ('block
group', '1'))),
'Block Group 1, Census Tract 263.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026302'), ('block
group', '1'))),
'Block Group 2, Census Tract 263.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026302'), ('block
group', '2'))),
'Block Group 4, Census Tract 263.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026302'), ('block
group', '4'))),
'Block Group 4, Census Tract 264, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026400'), ('block
group', '4'))),
'Block Group 6, Census Tract 264, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026400'), ('block
group', '6'))),
'Block Group 2, Census Tract 264, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026400'), ('block
group', '2'))),
'Block Group 7, Census Tract 264, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026400'), ('block
group', '7'))),
'Block Group 3, Census Tract 264, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026400'), ('block
group', '3'))),
'Block Group 3, Census Tract 271, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '027100'), ('block
group', '3'))),
'Block Group 2, Census Tract 15, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '001500'), ('block
group', '2'))),
'Block Group 2, Census Tract 81.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008101'), ('block
group', '2'))),

```

'Block Group 1, Census Tract 81.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008101'), ('block group', '1'))),

'Block Group 3, Census Tract 81.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008101'), ('block group', '3'))),

'Block Group 2, Census Tract 71.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '007102'), ('block group', '2'))),

'Block Group 3, Census Tract 71.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '007102'), ('block group', '3'))),

'Block Group 1, Census Tract 71.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '007102'), ('block group', '1'))),

'Block Group 4, Census Tract 71.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '007102'), ('block group', '4'))),

'Block Group 5, Census Tract 71.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '007102'), ('block group', '5'))),

'Block Group 1, Census Tract 88.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008802'), ('block group', '1'))),

'Block Group 2, Census Tract 88.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008802'), ('block group', '2'))),

'Block Group 3, Census Tract 88.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008802'), ('block group', '3'))),

'Block Group 4, Census Tract 88.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008802'), ('block group', '4'))),

'Block Group 1, Census Tract 87.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008702'), ('block group', '1'))),

'Block Group 2, Census Tract 87.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008702'), ('block group', '2'))),

'Block Group 1, Census Tract 87.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008701'), ('block group', '1'))),

'Block Group 1, Census Tract 122.04, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '012204'), ('block group', '1'))),

'Block Group 1, Census Tract 86.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '008602'), ('block

```

group', '1'))),
'Block Group 1, Census Tract 335, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '033500'), ('block
group', '1'))),
'Block Group 2, Census Tract 335, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '033500'), ('block
group', '2'))),
'Block Group 3, Census Tract 341, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034100'), ('block
group', '3'))),
'Block Group 2, Census Tract 341, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034100'), ('block
group', '2'))),
'Block Group 6, Census Tract 341, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034100'), ('block
group', '6'))),
'Block Group 2, Census Tract 16, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '001600'), ('block
group', '2'))),
'Block Group 1, Census Tract 39.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003902'), ('block
group', '1'))),
'Block Group 4, Census Tract 39.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003902'), ('block
group', '4'))),
'Block Group 2, Census Tract 39.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003902'), ('block
group', '2'))),
'Block Group 3, Census Tract 39.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003902'), ('block
group', '3'))),
'Block Group 3, Census Tract 78, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '007800'), ('block
group', '3'))),
'Block Group 4, Census Tract 78, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '007800'), ('block
group', '4'))),
'Block Group 5, Census Tract 78, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '007800'), ('block
group', '5'))),
'Block Group 2, Census Tract 236, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '023600'), ('block
group', '2'))),
'Block Group 1, Census Tract 239, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '023900'), ('block
group', '1'))),
'Block Group 2, Census Tract 242, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '024200'), ('block
group', '2'))),
'Block Group 1, Census Tract 242, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '024200'), ('block
group', '1'))),
'Block Group 4, Census Tract 242, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '024200'), ('block
group', '4'))),
'Block Group 1, Census Tract 122.03, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '012203'), ('block
group', '1'))),
'Block Group 2, Census Tract 136.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '013602'), ('block
group', '2'))),
'Block Group 1, Census Tract 136.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '013602'), ('block
group', '1'))),
'Block Group 4, Census Tract 136.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '013602'), ('block
group', '4'))),
'Block Group 1, Census Tract 345.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '034501'), ('block
group', '1'))),
'Block Group 2, Census Tract 216, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '021600'), ('block
group', '2'))),
'Block Group 1, Census Tract 220, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '022000'), ('block
group', '1'))),
'Block Group 1, Census Tract 2, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '000200'), ('block
group', '1'))),
'Block Group 1, Census Tract 6, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '000600'), ('block
group', '1'))),
'Block Group 1, Census Tract 241, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '024100'), ('block
group', '1'))),
'Block Group 2, Census Tract 248, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '024800'), ('block
group', '2'))),
'Block Group 1, Census Tract 248, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '024800'), ('block
group', '1'))),
'Block Group 5, Census Tract 202, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '020200'), ('block
group', '5'))),

```

```
'Block Group 2, Census Tract 202, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '020200'), ('block  
group', '2'))),  
'Block Group 6, Census Tract 202, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '020200'), ('block  
group', '6'))),  
'Block Group 3, Census Tract 202, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '020200'), ('block  
group', '3'))),  
'Block Group 4, Census Tract 202, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '020200'), ('block  
group', '4'))),  
'Block Group 1, Census Tract 202, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '020200'), ('block  
group', '1'))),  
'Block Group 1, Census Tract 106, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '010600'), ('block  
group', '1'))),  
'Block Group 2, Census Tract 244, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '024400'), ('block  
group', '2'))),  
'Block Group 1, Census Tract 244, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '024400'), ('block  
group', '1'))),  
'Block Group 3, Census Tract 244, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '024400'), ('block  
group', '3'))),  
'Block Group 2, Census Tract 246, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '024600'), ('block  
group', '2'))),  
'Block Group 1, Census Tract 246, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '024600'), ('block  
group', '1'))),  
'Block Group 3, Census Tract 246, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '024600'), ('block  
group', '3'))),  
'Block Group 3, Census Tract 249, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '024900'), ('block  
group', '3'))),  
'Block Group 2, Census Tract 249, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '024900'), ('block  
group', '2'))),  
'Block Group 1, Census Tract 249, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '024900'), ('block  
group', '1'))),  
'Block Group 1, Census Tract 265, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '026500'), ('block
```

```

group', '1'))),
'Block Group 3, Census Tract 265, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026500'), ('block
group', '3'))),
'Block Group 5, Census Tract 265, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026500'), ('block
group', '5'))),
'Block Group 3, Census Tract 315.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031501'), ('block
group', '3'))),
'Block Group 4, Census Tract 315.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031501'), ('block
group', '4'))),
'Block Group 5, Census Tract 315.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031501'), ('block
group', '5'))),
'Block Group 1, Census Tract 389, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '038900'), ('block
group', '1'))),
'Block Group 2, Census Tract 389, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '038900'), ('block
group', '2'))),
'Block Group 3, Census Tract 389, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '038900'), ('block
group', '3'))),
'Block Group 1, Census Tract 9800, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '980000'), ('block
group', '1'))),
'Block Group 2, Census Tract 265, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026500'), ('block
group', '2'))),
'Block Group 6, Census Tract 267, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026700'), ('block
group', '6'))),
'Block Group 7, Census Tract 267, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026700'), ('block
group', '7'))),
'Block Group 4, Census Tract 267, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026700'), ('block
group', '4'))),
'Block Group 5, Census Tract 267, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026700'), ('block
group', '5'))),
'Block Group 2, Census Tract 286, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028600'), ('block
group', '2'))),
'Block Group 4, Census Tract 286, Philadelphia County, Pennsylvania':

```

```

censusgeo([('state', '42'), ('county', '101'), ('tract', '028600'), ('block
group', '4'))),
'Block Group 5, Census Tract 286, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028600'), ('block
group', '5'))),
'Block Group 1, Census Tract 286, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028600'), ('block
group', '1'))),
'Block Group 1, Census Tract 1, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '000100'), ('block
group', '1'))),
'Block Group 4, Census Tract 13, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001300'), ('block
group', '4'))),
'Block Group 1, Census Tract 13, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001300'), ('block
group', '1'))),
'Block Group 5, Census Tract 13, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001300'), ('block
group', '5'))),
'Block Group 3, Census Tract 13, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001300'), ('block
group', '3'))),
'Block Group 6, Census Tract 191, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019100'), ('block
group', '6'))),
'Block Group 3, Census Tract 191, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019100'), ('block
group', '3'))),
'Block Group 6, Census Tract 192, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019200'), ('block
group', '6'))),
'Block Group 3, Census Tract 192, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019200'), ('block
group', '3'))),
'Block Group 1, Census Tract 192, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019200'), ('block
group', '1'))),
'Block Group 1, Census Tract 17, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001700'), ('block
group', '1'))),
'Block Group 2, Census Tract 17, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001700'), ('block
group', '2'))),
'Block Group 4, Census Tract 25, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '002500'), ('block
group', '4'))),

```

```
'Block Group 3, Census Tract 25, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '002500'), ('block  
group', '3'))),  
'Block Group 5, Census Tract 33, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003300'), ('block  
group', '5'))),  
'Block Group 1, Census Tract 176.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '017601'), ('block  
group', '1'))),  
'Block Group 2, Census Tract 176.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '017601'), ('block  
group', '2'))),  
'Block Group 4, Census Tract 176.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '017601'), ('block  
group', '4'))),  
'Block Group 1, Census Tract 29, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '002900'), ('block  
group', '1'))),  
'Block Group 3, Census Tract 31, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003100'), ('block  
group', '3'))),  
'Block Group 4, Census Tract 32, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003200'), ('block  
group', '4'))),  
'Block Group 3, Census Tract 32, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003200'), ('block  
group', '3'))),  
'Block Group 1, Census Tract 32, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003200'), ('block  
group', '1'))),  
'Block Group 2, Census Tract 32, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003200'), ('block  
group', '2'))),  
'Block Group 3, Census Tract 40.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '004001'), ('block  
group', '3'))),  
'Block Group 2, Census Tract 40.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '004001'), ('block  
group', '2'))),  
'Block Group 4, Census Tract 40.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '004001'), ('block  
group', '4'))),  
'Block Group 2, Census Tract 62, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '006200'), ('block  
group', '2'))),  
'Block Group 4, Census Tract 62, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '006200'), ('block
```

```

group', '4'))),
'Block Group 1, Census Tract 72, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '007200'), ('block
group', '1'))),
'Block Group 6, Census Tract 72, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '007200'), ('block
group', '6'))),
'Block Group 5, Census Tract 72, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '007200'), ('block
group', '5'))),
'Block Group 4, Census Tract 183, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '018300'), ('block
group', '4'))),
'Block Group 4, Census Tract 190, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '019000'), ('block
group', '4'))),
'Block Group 3, Census Tract 190, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '019000'), ('block
group', '3'))),
'Block Group 2, Census Tract 190, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '019000'), ('block
group', '2'))),
'Block Group 1, Census Tract 206, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '020600'), ('block
group', '1'))),
'Block Group 3, Census Tract 215, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '021500'), ('block
group', '3'))),
'Block Group 2, Census Tract 215, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '021500'), ('block
group', '2'))),
'Block Group 4, Census Tract 215, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '021500'), ('block
group', '4'))),
'Block Group 4, Census Tract 317, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031700'), ('block
group', '4'))),
'Block Group 3, Census Tract 317, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031700'), ('block
group', '3'))),
'Block Group 1, Census Tract 317, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031700'), ('block
group', '1'))),
'Block Group 2, Census Tract 317, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031700'), ('block
group', '2'))),
'Block Group 4, Census Tract 318, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '031800'), ('block
group', '4'))),
'Block Group 1, Census Tract 318, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031800'), ('block
group', '1'))),
'Block Group 1, Census Tract 323, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '032300'), ('block
group', '1'))),
'Block Group 1, Census Tract 39.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '003901'), ('block
group', '1'))),
'Block Group 2, Census Tract 39.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '003901'), ('block
group', '2'))),
'Block Group 3, Census Tract 39.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '003901'), ('block
group', '3'))),
'Block Group 1, Census Tract 50, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '005000'), ('block
group', '1'))),
'Block Group 3, Census Tract 63, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '006300'), ('block
group', '3'))),
'Block Group 1, Census Tract 63, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '006300'), ('block
group', '1'))),
'Block Group 1, Census Tract 73, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '007300'), ('block
group', '1'))),
'Block Group 2, Census Tract 73, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '007300'), ('block
group', '2'))),
'Block Group 2, Census Tract 337.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '033702'), ('block
group', '2'))),
'Block Group 4, Census Tract 278, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '027800'), ('block
group', '4'))),
'Block Group 1, Census Tract 278, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '027800'), ('block
group', '1'))),
'Block Group 2, Census Tract 278, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '027800'), ('block
group', '2'))),
'Block Group 3, Census Tract 278, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '027800'), ('block
group', '3'))),

```

```

'Block Group 2, Census Tract 282, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028200'), ('block
group', '2'))),
'Block Group 3, Census Tract 282, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028200'), ('block
group', '3'))),
'Block Group 1, Census Tract 282, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028200'), ('block
group', '1'))),
'Block Group 3, Census Tract 288, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028800'), ('block
group', '3'))),
'Block Group 2, Census Tract 288, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028800'), ('block
group', '2'))),
'Block Group 2, Census Tract 298, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '029800'), ('block
group', '2'))),
'Block Group 3, Census Tract 298, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '029800'), ('block
group', '3'))),
'Block Group 1, Census Tract 298, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '029800'), ('block
group', '1'))),
'Block Group 4, Census Tract 153, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '015300'), ('block
group', '4'))),
'Block Group 2, Census Tract 153, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '015300'), ('block
group', '2'))),
'Block Group 3, Census Tract 157, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '015700'), ('block
group', '3'))),
'Block Group 1, Census Tract 157, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '015700'), ('block
group', '1'))),
'Block Group 3, Census Tract 160, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016000'), ('block
group', '3'))),
'Block Group 4, Census Tract 160, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016000'), ('block
group', '4'))),
'Block Group 6, Census Tract 160, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016000'), ('block
group', '6'))),
'Block Group 1, Census Tract 160, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '016000'), ('block
group', '1'))]

```

```

group', '1'))),
'Block Group 5, Census Tract 160, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '016000'), ('block
group', '5'))),
'Block Group 2, Census Tract 168, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '016800'), ('block
group', '2'))),
'Block Group 1, Census Tract 168, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '016800'), ('block
group', '1'))),
'Block Group 4, Census Tract 168, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '016800'), ('block
group', '4'))),
'Block Group 5, Census Tract 168, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '016800'), ('block
group', '5'))),
'Block Group 1, Census Tract 306, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '030600'), ('block
group', '1'))),
'Block Group 5, Census Tract 341, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034100'), ('block
group', '5'))),
'Block Group 4, Census Tract 341, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034100'), ('block
group', '4'))),
'Block Group 1, Census Tract 351, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035100'), ('block
group', '1'))),
'Block Group 2, Census Tract 351, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035100'), ('block
group', '2'))),
'Block Group 1, Census Tract 277, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '027700'), ('block
group', '1'))),
'Block Group 1, Census Tract 285, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028500'), ('block
group', '1'))),
'Block Group 3, Census Tract 345.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034502'), ('block
group', '3'))),
'Block Group 1, Census Tract 345.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034502'), ('block
group', '1'))),
'Block Group 2, Census Tract 345.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034502'), ('block
group', '2'))),
'Block Group 2, Census Tract 11.02, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '001102'), ('block
group', '2'))),
'Block Group 1, Census Tract 11.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '001102'), ('block
group', '1'))),
'Block Group 1, Census Tract 9.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '000902'), ('block
group', '1'))),
'Block Group 2, Census Tract 9.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '000902'), ('block
group', '2'))),
'Block Group 2, Census Tract 9.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '000901'), ('block
group', '2'))),
'Block Group 1, Census Tract 9.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '000901'), ('block
group', '1'))),
'Block Group 1, Census Tract 172.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017201'), ('block
group', '1'))),
'Block Group 2, Census Tract 172.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017201'), ('block
group', '2'))),
'Block Group 3, Census Tract 172.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017201'), ('block
group', '3'))),
'Block Group 3, Census Tract 172.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017202'), ('block
group', '3'))),
'Block Group 3, Census Tract 348.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '034801'), ('block
group', '3'))),
'Block Group 1, Census Tract 12.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '001202'), ('block
group', '1'))),
'Block Group 5, Census Tract 12.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '001202'), ('block
group', '5'))),
'Block Group 3, Census Tract 12.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '001202'), ('block
group', '3'))),
'Block Group 2, Census Tract 12.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '001202'), ('block
group', '2'))),
'Block Group 2, Census Tract 12.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '001201'), ('block
group', '2'))),

```

```

'Block Group 4, Census Tract 12.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '001201'), ('block
group', '4'))),
'Block Group 4, Census Tract 8.03, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '000803'), ('block
group', '4'))),
'Block Group 2, Census Tract 8.03, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '000803'), ('block
group', '2'))),
'Block Group 3, Census Tract 306, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '030600'), ('block
group', '3'))),
'Block Group 7, Census Tract 306, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '030600'), ('block
group', '7'))),
'Block Group 2, Census Tract 306, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '030600'), ('block
group', '2'))),
'Block Group 7, Census Tract 316, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '031600'), ('block
group', '7'))),
'Block Group 5, Census Tract 316, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '031600'), ('block
group', '5'))),
'Block Group 3, Census Tract 316, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '031600'), ('block
group', '3'))),
'Block Group 4, Census Tract 316, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '031600'), ('block
group', '4'))),
'Block Group 3, Census Tract 353.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '035301'), ('block
group', '3'))),
'Block Group 1, Census Tract 353.01, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '035301'), ('block
group', '1'))),
'Block Group 1, Census Tract 199, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019900'), ('block
group', '1'))),
'Block Group 2, Census Tract 199, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019900'), ('block
group', '2'))),
'Block Group 1, Census Tract 131, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '013100'), ('block
group', '1'))),
'Block Group 2, Census Tract 131, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '013100'), ('block
group', '2'))]

```

```

group', '2'))),
'Block Group 2, Census Tract 138, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '013800'), ('block
group', '2'))),
'Block Group 1, Census Tract 162, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '016200'), ('block
group', '1'))),
'Block Group 1, Census Tract 293, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '029300'), ('block
group', '1'))),
'Block Group 4, Census Tract 302, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '030200'), ('block
group', '4'))),
'Block Group 3, Census Tract 302, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '030200'), ('block
group', '3'))),
'Block Group 2, Census Tract 302, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '030200'), ('block
group', '2'))),
'Block Group 1, Census Tract 315.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031501'), ('block
group', '1'))),
'Block Group 1, Census Tract 347.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034701'), ('block
group', '1'))),
'Block Group 3, Census Tract 352, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035200'), ('block
group', '3'))),
'Block Group 1, Census Tract 352, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035200'), ('block
group', '1'))),
'Block Group 2, Census Tract 390, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '039000'), ('block
group', '2'))),
'Block Group 3, Census Tract 390, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '039000'), ('block
group', '3'))),
'Block Group 7, Census Tract 390, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '039000'), ('block
group', '7'))),
'Block Group 8, Census Tract 390, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '039000'), ('block
group', '8'))),
'Block Group 1, Census Tract 390, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '039000'), ('block
group', '1'))),
'Block Group 4, Census Tract 390, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '039000'), ('block
group', '4'))),
'Block Group 5, Census Tract 390, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '039000'), ('block
group', '5'))),
'Block Group 3, Census Tract 378, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '037800'), ('block
group', '3'))),
'Block Group 1, Census Tract 378, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '037800'), ('block
group', '1'))),
'Block Group 3, Census Tract 377, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '037700'), ('block
group', '3'))),
'Block Group 1, Census Tract 377, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '037700'), ('block
group', '1'))),
'Block Group 3, Census Tract 86.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008602'), ('block
group', '3'))),
'Block Group 2, Census Tract 375, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '037500'), ('block
group', '2'))),
'Block Group 1, Census Tract 9808, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '980800'), ('block
group', '1'))),
'Block Group 1, Census Tract 180.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '018002'), ('block
group', '1'))),
'Block Group 5, Census Tract 314.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031401'), ('block
group', '5'))),
'Block Group 2, Census Tract 314.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031401'), ('block
group', '2'))),
'Block Group 2, Census Tract 314.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031402'), ('block
group', '2'))),
'Block Group 3, Census Tract 314.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031402'), ('block
group', '3'))),
'Block Group 2, Census Tract 162, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '016200'), ('block
group', '2'))),
'Block Group 2, Census Tract 179, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017900'), ('block
group', '2'))),

```

```

'Block Group 1, Census Tract 179, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017900'), ('block
group', '1'))),
'Block Group 4, Census Tract 179, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017900'), ('block
group', '4'))),
'Block Group 3, Census Tract 179, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017900'), ('block
group', '3'))),
'Block Group 2, Census Tract 183, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '018300'), ('block
group', '2'))),
'Block Group 2, Census Tract 294, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '029400'), ('block
group', '2'))),
'Block Group 7, Census Tract 300, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '030000'), ('block
group', '7'))),
'Block Group 4, Census Tract 300, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '030000'), ('block
group', '4'))),
'Block Group 6, Census Tract 300, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '030000'), ('block
group', '6'))),
'Block Group 2, Census Tract 300, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '030000'), ('block
group', '2'))),
'Block Group 1, Census Tract 300, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '030000'), ('block
group', '1'))),
'Block Group 7, Census Tract 188, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '018800'), ('block
group', '7'))),
'Block Group 2, Census Tract 188, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '018800'), ('block
group', '2'))),
'Block Group 3, Census Tract 188, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '018800'), ('block
group', '3'))),
'Block Group 6, Census Tract 188, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '018800'), ('block
group', '6'))),
'Block Group 2, Census Tract 191, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019100'), ('block
group', '2'))),
'Block Group 1, Census Tract 191, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019100'), ('block
group', '1'))]

```

```

group', '1'))),
'Block Group 6, Census Tract 266, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026600'), ('block
group', '6'))),
'Block Group 2, Census Tract 266, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026600'), ('block
group', '2'))),
'Block Group 3, Census Tract 333, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '033300'), ('block
group', '3'))),
'Block Group 2, Census Tract 333, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '033300'), ('block
group', '2'))),
'Block Group 3, Census Tract 336, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '033600'), ('block
group', '3'))),
'Block Group 1, Census Tract 336, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '033600'), ('block
group', '1'))),
'Block Group 4, Census Tract 240, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '024000'), ('block
group', '4'))),
'Block Group 3, Census Tract 240, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '024000'), ('block
group', '3'))),
'Block Group 2, Census Tract 240, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '024000'), ('block
group', '2'))),
'Block Group 2, Census Tract 281, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028100'), ('block
group', '2'))),
'Block Group 1, Census Tract 283, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028300'), ('block
group', '1'))),
'Block Group 5, Census Tract 283, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028300'), ('block
group', '5'))),
'Block Group 6, Census Tract 283, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028300'), ('block
group', '6'))),
'Block Group 2, Census Tract 283, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028300'), ('block
group', '2'))),
'Block Group 4, Census Tract 283, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028300'), ('block
group', '4'))),
'Block Group 3, Census Tract 330, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '033000'), ('block
group', '3'))),
'Block Group 1, Census Tract 330, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '033000'), ('block
group', '1'))),
'Block Group 4, Census Tract 330, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '033000'), ('block
group', '4'))),
'Block Group 5, Census Tract 330, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '033000'), ('block
group', '5'))),
'Block Group 6, Census Tract 330, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '033000'), ('block
group', '6'))),
'Block Group 2, Census Tract 330, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '033000'), ('block
group', '2'))),
'Block Group 2, Census Tract 340, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '034000'), ('block
group', '2'))),
'Block Group 1, Census Tract 305.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030501'), ('block
group', '1'))),
'Block Group 2, Census Tract 305.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030501'), ('block
group', '2'))),
'Block Group 3, Census Tract 305.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030502'), ('block
group', '3'))),
'Block Group 2, Census Tract 305.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030502'), ('block
group', '2'))),
'Block Group 4, Census Tract 305.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030502'), ('block
group', '4'))),
'Block Group 3, Census Tract 98.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009802'), ('block
group', '3'))),
'Block Group 2, Census Tract 98.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009802'), ('block
group', '2'))),
'Block Group 2, Census Tract 180.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '018001'), ('block
group', '2'))),
'Block Group 5, Census Tract 177.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017702'), ('block
group', '5'))),

```

```

'Block Group 4, Census Tract 177.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017702'), ('block
group', '4'))),
'Block Group 1, Census Tract 177.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '017702'), ('block
group', '1'))),
'Block Group 2, Census Tract 36, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '003600'), ('block
group', '2'))),
'Block Group 1, Census Tract 36, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '003600'), ('block
group', '1'))),
'Block Group 4, Census Tract 36, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '003600'), ('block
group', '4'))),
'Block Group 3, Census Tract 36, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '003600'), ('block
group', '3'))),
'Block Group 1, Census Tract 64, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006400'), ('block
group', '1'))),
'Block Group 2, Census Tract 64, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006400'), ('block
group', '2'))),
'Block Group 3, Census Tract 64, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006400'), ('block
group', '3'))),
'Block Group 6, Census Tract 65, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006500'), ('block
group', '6'))),
'Block Group 1, Census Tract 65, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006500'), ('block
group', '1'))),
'Block Group 3, Census Tract 65, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006500'), ('block
group', '3'))),
'Block Group 2, Census Tract 65, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006500'), ('block
group', '2'))),
'Block Group 4, Census Tract 65, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006500'), ('block
group', '4'))),
'Block Group 4, Census Tract 66, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006600'), ('block
group', '4'))),
'Block Group 1, Census Tract 66, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006600'), ('block
group', '1'))]

```

```

group', '1'))),
'Block Group 3, Census Tract 66, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '006600'), ('block
group', '3'))),
'Block Group 1, Census Tract 142, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014200'), ('block
group', '1'))),
'Block Group 2, Census Tract 142, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '014200'), ('block
group', '2'))),
'Block Group 2, Census Tract 161, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '016100'), ('block
group', '2'))),
'Block Group 3, Census Tract 161, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '016100'), ('block
group', '3'))),
'Block Group 2, Census Tract 169.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '016901'), ('block
group', '2'))),
'Block Group 1, Census Tract 169.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '016901'), ('block
group', '1'))),
'Block Group 1, Census Tract 174, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '017400'), ('block
group', '1'))),
'Block Group 2, Census Tract 174, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '017400'), ('block
group', '2'))),
'Block Group 3, Census Tract 340, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034000'), ('block
group', '3'))),
'Block Group 1, Census Tract 340, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '034000'), ('block
group', '1'))),
'Block Group 1, Census Tract 198, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '019800'), ('block
group', '1'))),
'Block Group 6, Census Tract 198, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '019800'), ('block
group', '6'))),
'Block Group 1, Census Tract 212, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '021200'), ('block
group', '1'))),
'Block Group 2, Census Tract 212, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '021200'), ('block
group', '2'))),
'Block Group 1, Census Tract 213, Philadelphia County, Pennsylvania':

```

```
censusgeo(((('state', '42'), ('county', '101'), ('tract', '021300'), ('block group', '1'))),  
    'Block Group 4, Census Tract 213, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '021300'), ('block group', '4'))),  
    'Block Group 5, Census Tract 213, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '021300'), ('block group', '5'))),  
    'Block Group 3, Census Tract 217, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '021700'), ('block group', '3'))),  
    'Block Group 4, Census Tract 217, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '021700'), ('block group', '4'))),  
    'Block Group 4, Census Tract 84, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008400'), ('block group', '4'))),  
    'Block Group 1, Census Tract 84, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008400'), ('block group', '1'))),  
    'Block Group 2, Census Tract 84, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008400'), ('block group', '2'))),  
    'Block Group 3, Census Tract 84, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '008400'), ('block group', '3'))),  
    'Block Group 1, Census Tract 90, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009000'), ('block group', '1'))),  
    'Block Group 2, Census Tract 90, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009000'), ('block group', '2'))),  
    'Block Group 4, Census Tract 90, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009000'), ('block group', '4'))),  
    'Block Group 2, Census Tract 102, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '010200'), ('block group', '2'))),  
    'Block Group 3, Census Tract 102, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '010200'), ('block group', '3'))),  
    'Block Group 1, Census Tract 109, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '010900'), ('block group', '1'))),  
    'Block Group 3, Census Tract 109, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '010900'), ('block group', '3'))),
```

```
'Block Group 1, Census Tract 348.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '034801'), ('block  
group', '1'))),  
'Block Group 2, Census Tract 348.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '034801'), ('block  
group', '2'))),  
'Block Group 1, Census Tract 61, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '006100'), ('block  
group', '1'))),  
'Block Group 4, Census Tract 33, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003300'), ('block  
group', '4'))),  
'Block Group 3, Census Tract 33, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003300'), ('block  
group', '3'))),  
'Block Group 1, Census Tract 33, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003300'), ('block  
group', '1'))),  
'Block Group 3, Census Tract 40.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '004002'), ('block  
group', '3'))),  
'Block Group 1, Census Tract 40.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '004002'), ('block  
group', '1'))),  
'Block Group 4, Census Tract 73, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '007300'), ('block  
group', '4'))),  
'Block Group 1, Census Tract 79, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '007900'), ('block  
group', '1'))),  
'Block Group 3, Census Tract 79, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '007900'), ('block  
group', '3'))),  
'Block Group 2, Census Tract 79, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '007900'), ('block  
group', '2'))),  
'Block Group 4, Census Tract 79, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '007900'), ('block  
group', '4'))),  
'Block Group 2, Census Tract 85, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '008500'), ('block  
group', '2'))),  
'Block Group 5, Census Tract 85, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '008500'), ('block  
group', '5'))),  
'Block Group 6, Census Tract 85, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '008500'), ('block
```

```

group', '6'))),
'Block Group 3, Census Tract 85, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '008500'), ('block
group', '3'))),
'Block Group 4, Census Tract 85, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '008500'), ('block
group', '4'))),
'Block Group 7, Census Tract 85, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '008500'), ('block
group', '7'))),
'Block Group 1, Census Tract 85, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '008500'), ('block
group', '1'))),
'Block Group 3, Census Tract 204, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '020400'), ('block
group', '3'))),
'Block Group 3, Census Tract 362.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '036201'), ('block
group', '3'))),
'Block Group 1, Census Tract 362.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '036202'), ('block
group', '1'))),
'Block Group 3, Census Tract 362.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '036202'), ('block
group', '3'))),
'Block Group 4, Census Tract 362.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '036202'), ('block
group', '4'))),
'Block Group 3, Census Tract 238, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '023800'), ('block
group', '3'))),
'Block Group 4, Census Tract 238, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '023800'), ('block
group', '4'))),
'Block Group 7, Census Tract 252, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '025200'), ('block
group', '7'))),
'Block Group 3, Census Tract 252, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '025200'), ('block
group', '3'))),
'Block Group 4, Census Tract 252, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '025200'), ('block
group', '4'))),
'Block Group 4, Census Tract 262, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026200'), ('block
group', '4'))),
'Block Group 2, Census Tract 262, Philadelphia County, Pennsylvania':

```

```
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026200'), ('block group', '2'))),  
    'Block Group 3, Census Tract 255, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '025500'), ('block group', '3'))),  
    'Block Group 1, Census Tract 309, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030900'), ('block group', '1'))),  
    'Block Group 2, Census Tract 309, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030900'), ('block group', '2'))),  
    'Block Group 6, Census Tract 320, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '032000'), ('block group', '6'))),  
    'Block Group 5, Census Tract 320, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '032000'), ('block group', '5'))),  
    'Block Group 4, Census Tract 320, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '032000'), ('block group', '4'))),  
    'Block Group 7, Census Tract 178, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017800'), ('block group', '7'))),  
    'Block Group 4, Census Tract 178, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017800'), ('block group', '4'))),  
    'Block Group 6, Census Tract 178, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017800'), ('block group', '6'))),  
    'Block Group 1, Census Tract 178, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017800'), ('block group', '1'))),  
    'Block Group 5, Census Tract 178, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017800'), ('block group', '5'))),  
    'Block Group 4, Census Tract 197, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '019700'), ('block group', '4'))),  
    'Block Group 1, Census Tract 197, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '019700'), ('block group', '1'))),  
    'Block Group 5, Census Tract 197, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '019700'), ('block group', '5'))),  
    'Block Group 6, Census Tract 197, Philadelphia County, Pennsylvania':  
censusgeo(((('state', '42'), ('county', '101'), ('tract', '019700'), ('block group', '6'))),
```

```

'Block Group 2, Census Tract 197, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019700'), ('block
group', '2'))),
'Block Group 3, Census Tract 197, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019700'), ('block
group', '3'))),
'Block Group 2, Census Tract 200, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '020000'), ('block
group', '2'))),
'Block Group 1, Census Tract 200, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '020000'), ('block
group', '1'))),
'Block Group 2, Census Tract 108, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '010800'), ('block
group', '2'))),
'Block Group 4, Census Tract 108, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '010800'), ('block
group', '4'))),
'Block Group 5, Census Tract 108, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '010800'), ('block
group', '5'))),
'Block Group 3, Census Tract 199, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019900'), ('block
group', '3'))),
'Block Group 4, Census Tract 199, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019900'), ('block
group', '4'))),
'Block Group 1, Census Tract 334, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033400'), ('block
group', '1'))),
'Block Group 4, Census Tract 334, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033400'), ('block
group', '4'))),
'Block Group 2, Census Tract 334, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033400'), ('block
group', '2'))),
'Block Group 3, Census Tract 334, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033400'), ('block
group', '3'))),
'Block Group 1, Census Tract 339, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033900'), ('block
group', '1'))),
'Block Group 2, Census Tract 339, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033900'), ('block
group', '2'))),
'Block Group 3, Census Tract 339, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '033900'), ('block
group', '3'))]

```

```

group', '3'))),
'Block Group 2, Census Tract 361, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '036100'), ('block
group', '2'))),
'Block Group 1, Census Tract 273, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '027300'), ('block
group', '1'))),
'Block Group 3, Census Tract 273, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '027300'), ('block
group', '3'))),
'Block Group 5, Census Tract 273, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '027300'), ('block
group', '5'))),
'Block Group 2, Census Tract 284, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028400'), ('block
group', '2'))),
'Block Group 3, Census Tract 284, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '028400'), ('block
group', '3'))),
'Block Group 1, Census Tract 8.03, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '000803'), ('block
group', '1'))),
'Block Group 3, Census Tract 8.03, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '000803'), ('block
group', '3'))),
'Block Group 4, Census Tract 27.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002702'), ('block
group', '4'))),
'Block Group 3, Census Tract 27.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002702'), ('block
group', '3'))),
'Block Group 1, Census Tract 27.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002702'), ('block
group', '1'))),
'Block Group 3, Census Tract 30.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003002'), ('block
group', '3'))),
'Block Group 1, Census Tract 30.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003002'), ('block
group', '1'))),
'Block Group 4, Census Tract 30.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003002'), ('block
group', '4'))),
'Block Group 3, Census Tract 30.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '003001'), ('block
group', '3'))),
'Block Group 4, Census Tract 30.01, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '003001'), ('block
group', '4'))),
'Block Group 1, Census Tract 204, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '020400'), ('block
group', '1'))),
'Block Group 2, Census Tract 204, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '020400'), ('block
group', '2'))),
'Block Group 4, Census Tract 204, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '020400'), ('block
group', '4'))),
'Block Group 1, Census Tract 237, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '023700'), ('block
group', '1'))),
'Block Group 4, Census Tract 237, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '023700'), ('block
group', '4'))),
'Block Group 3, Census Tract 237, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '023700'), ('block
group', '3'))),
'Block Group 2, Census Tract 63, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '006300'), ('block
group', '2'))),
'Block Group 1, Census Tract 145, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '014500'), ('block
group', '1'))),
'Block Group 3, Census Tract 165, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '016500'), ('block
group', '3'))),
'Block Group 2, Census Tract 336, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '033600'), ('block
group', '2'))),
'Block Group 2, Census Tract 363.03, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '036303'), ('block
group', '2'))),
'Block Group 1, Census Tract 363.03, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '036303'), ('block
group', '1'))),
'Block Group 3, Census Tract 363.03, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '036303'), ('block
group', '3'))),
'Block Group 3, Census Tract 299, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '029900'), ('block
group', '3'))),
'Block Group 1, Census Tract 299, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '029900'), ('block
group', '1'))),

```

```
'Block Group 4, Census Tract 299, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '029900'), ('block  
group', '4'))),  
'Block Group 3, Census Tract 277, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '027700'), ('block  
group', '3'))),  
'Block Group 6, Census Tract 277, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '027700'), ('block  
group', '6'))),  
'Block Group 2, Census Tract 277, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '027700'), ('block  
group', '2'))),  
'Block Group 1, Census Tract 140, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '014000'), ('block  
group', '1'))),  
'Block Group 2, Census Tract 141, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '014100'), ('block  
group', '2'))),  
'Block Group 1, Census Tract 144, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '014400'), ('block  
group', '1'))),  
'Block Group 2, Census Tract 144, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '014400'), ('block  
group', '2'))),  
'Block Group 1, Census Tract 152, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '015200'), ('block  
group', '1'))),  
'Block Group 3, Census Tract 152, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '015200'), ('block  
group', '3'))),  
'Block Group 5, Census Tract 152, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '015200'), ('block  
group', '5'))),  
'Block Group 1, Census Tract 60, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '006000'), ('block  
group', '1'))),  
'Block Group 2, Census Tract 60, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '006000'), ('block  
group', '2'))),  
'Block Group 4, Census Tract 60, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '006000'), ('block  
group', '4'))),  
'Block Group 3, Census Tract 60, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '006000'), ('block  
group', '3'))),  
'Block Group 3, Census Tract 266, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '026600'), ('block
```

```

group', '3'))),
'Block Group 8, Census Tract 266, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026600'), ('block
group', '8'))),
'Block Group 4, Census Tract 266, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026600'), ('block
group', '4'))),
'Block Group 5, Census Tract 266, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026600'), ('block
group', '5'))),
'Block Group 7, Census Tract 266, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026600'), ('block
group', '7'))),
'Block Group 2, Census Tract 315.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '031501'), ('block
group', '2'))),
'Block Group 5, Census Tract 28.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002802'), ('block
group', '5'))),
'Block Group 3, Census Tract 28.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002802'), ('block
group', '3'))),
'Block Group 2, Census Tract 28.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002802'), ('block
group', '2'))),
'Block Group 1, Census Tract 28.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '002802'), ('block
group', '1'))),
'Block Group 3, Census Tract 177.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '017702'), ('block
group', '3'))),
'Block Group 2, Census Tract 177.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '017702'), ('block
group', '2'))),
'Block Group 1, Census Tract 177.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '017701'), ('block
group', '1'))),
'Block Group 2, Census Tract 177.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '017701'), ('block
group', '2'))),
'Block Group 1, Census Tract 356.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035602'), ('block
group', '1'))),
'Block Group 2, Census Tract 356.02, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '035602'), ('block
group', '2'))),
'Block Group 2, Census Tract 203, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '020300'), ('block
group', '2'))),
'Block Group 1, Census Tract 208, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '020800'), ('block
group', '1'))),
'Block Group 3, Census Tract 209, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '020900'), ('block
group', '3'))),
'Block Group 4, Census Tract 209, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '020900'), ('block
group', '4'))),
'Block Group 2, Census Tract 209, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '020900'), ('block
group', '2'))),
'Block Group 1, Census Tract 207, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '020700'), ('block
group', '1'))),
'Block Group 4, Census Tract 207, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '020700'), ('block
group', '4'))),
'Block Group 2, Census Tract 184, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '018400'), ('block
group', '2'))),
'Block Group 2, Census Tract 268, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026800'), ('block
group', '2'))),
'Block Group 4, Census Tract 268, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026800'), ('block
group', '4'))),
'Block Group 1, Census Tract 319, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031900'), ('block
group', '1'))),
'Block Group 4, Census Tract 319, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031900'), ('block
group', '4'))),
'Block Group 3, Census Tract 319, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031900'), ('block
group', '3'))),
'Block Group 2, Census Tract 319, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031900'), ('block
group', '2'))),
'Block Group 2, Census Tract 308, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030800'), ('block
group', '2'))),
'Block Group 3, Census Tract 308, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030800'), ('block
group', '3'))),

```

```
'Block Group 1, Census Tract 308, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '030800'), ('block  
group', '1'))),  
'Block Group 6, Census Tract 310, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '031000'), ('block  
group', '6'))),  
'Block Group 5, Census Tract 310, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '031000'), ('block  
group', '5'))),  
'Block Group 7, Census Tract 310, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '031000'), ('block  
group', '7'))),  
'Block Group 1, Census Tract 310, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '031000'), ('block  
group', '1'))),  
'Block Group 3, Census Tract 310, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '031000'), ('block  
group', '3'))),  
'Block Group 4, Census Tract 310, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '031000'), ('block  
group', '4'))),  
'Block Group 2, Census Tract 310, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '031000'), ('block  
group', '2'))),  
'Block Group 3, Census Tract 96, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '009600'), ('block  
group', '3'))),  
'Block Group 2, Census Tract 96, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '009600'), ('block  
group', '2'))),  
'Block Group 5, Census Tract 96, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '009600'), ('block  
group', '5'))),  
'Block Group 4, Census Tract 96, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '009600'), ('block  
group', '4'))),  
'Block Group 6, Census Tract 118, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '011800'), ('block  
group', '6'))),  
'Block Group 5, Census Tract 118, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '011800'), ('block  
group', '5'))),  
'Block Group 3, Census Tract 118, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '011800'), ('block  
group', '3'))),  
'Block Group 4, Census Tract 118, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '011800'), ('block  
group', '4'))),
```

```

group', '4'))),
'Block Group 5, Census Tract 119, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '011900'), ('block
group', '5'))),
'Block Group 4, Census Tract 119, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '011900'), ('block
group', '4'))),
'Block Group 3, Census Tract 119, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '011900'), ('block
group', '3'))),
'Block Group 3, Census Tract 139, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '013900'), ('block
group', '3'))),
'Block Group 2, Census Tract 255, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '025500'), ('block
group', '2'))),
'Block Group 1, Census Tract 255, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '025500'), ('block
group', '1'))),
'Block Group 2, Census Tract 256, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '025600'), ('block
group', '2'))),
'Block Group 3, Census Tract 256, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '025600'), ('block
group', '3'))),
'Block Group 1, Census Tract 256, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '025600'), ('block
group', '1'))),
'Block Group 4, Census Tract 263.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026301'), ('block
group', '4'))),
'Block Group 1, Census Tract 263.01, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '026301'), ('block
group', '1'))),
'Block Group 2, Census Tract 270, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '027000'), ('block
group', '2'))),
'Block Group 1, Census Tract 270, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '027000'), ('block
group', '1'))),
'Block Group 4, Census Tract 247, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '024700'), ('block
group', '4'))),
'Block Group 7, Census Tract 320, Philadelphia County, Pennsylvania':
censusgeo((('state', '42'), ('county', '101'), ('tract', '032000'), ('block
group', '7'))),
'Block Group 1, Census Tract 320, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '032000'), ('block
group', '1'))),
'Block Group 2, Census Tract 320, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '032000'), ('block
group', '2'))),
'Block Group 1, Census Tract 342, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '034200'), ('block
group', '1'))),
'Block Group 1, Census Tract 20, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '002000'), ('block
group', '1'))),
'Block Group 2, Census Tract 20, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '002000'), ('block
group', '2'))),
'Block Group 3, Census Tract 29, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '002900'), ('block
group', '3'))),
'Block Group 2, Census Tract 29, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '002900'), ('block
group', '2'))),
'Block Group 3, Census Tract 37.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '003702'), ('block
group', '3'))),
'Block Group 1, Census Tract 37.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '003702'), ('block
group', '1'))),
'Block Group 2, Census Tract 37.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '003702'), ('block
group', '2'))),
'Block Group 3, Census Tract 307, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030700'), ('block
group', '3'))),
'Block Group 2, Census Tract 307, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '030700'), ('block
group', '2'))),
'Block Group 1, Census Tract 312, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031200'), ('block
group', '1'))),
'Block Group 3, Census Tract 312, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '031200'), ('block
group', '3'))),
'Block Group 1, Census Tract 231, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '023100'), ('block
group', '1'))),
'Block Group 3, Census Tract 19, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '001900'), ('block
group', '3'))),

```

```

'Block Group 6, Census Tract 31, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '003100'), ('block
group', '6'))),
'Block Group 2, Census Tract 198, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019800'), ('block
group', '2'))),
'Block Group 3, Census Tract 198, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019800'), ('block
group', '3'))),
'Block Group 5, Census Tract 198, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '019800'), ('block
group', '5'))),
'Block Group 4, Census Tract 284, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028400'), ('block
group', '4'))),
'Block Group 3, Census Tract 257, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '025700'), ('block
group', '3'))),
'Block Group 2, Census Tract 257, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '025700'), ('block
group', '2'))),
'Block Group 2, Census Tract 211, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '021100'), ('block
group', '2'))),
'Block Group 1, Census Tract 211, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '021100'), ('block
group', '1'))),
'Block Group 1, Census Tract 261, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '026100'), ('block
group', '1'))),
'Block Group 3, Census Tract 261, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '026100'), ('block
group', '3'))),
'Block Group 2, Census Tract 261, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '026100'), ('block
group', '2'))),
'Block Group 3, Census Tract 272, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '027200'), ('block
group', '3'))),
'Block Group 1, Census Tract 272, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '027200'), ('block
group', '1'))),
'Block Group 2, Census Tract 272, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '027200'), ('block
group', '2'))),
'Block Group 1, Census Tract 275, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '027500'), ('block
group', '1'))),

```

```

group', '1'))),
'Block Group 3, Census Tract 275, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '027500'), ('block
group', '3'))),
'Block Group 2, Census Tract 275, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '027500'), ('block
group', '2'))),
'Block Group 4, Census Tract 275, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '027500'), ('block
group', '4'))),
'Block Group 1, Census Tract 280, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028000'), ('block
group', '1'))),
'Block Group 3, Census Tract 280, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '028000'), ('block
group', '3'))),
'Block Group 1, Census Tract 294, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '029400'), ('block
group', '1'))),
'Block Group 3, Census Tract 247, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '024700'), ('block
group', '3'))),
'Block Group 1, Census Tract 363.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '036302'), ('block
group', '1'))),
'Block Group 2, Census Tract 363.02, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '036302'), ('block
group', '2'))),
'Block Group 1, Census Tract 56, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '005600'), ('block
group', '1'))),
'Block Group 3, Census Tract 67, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006700'), ('block
group', '3'))),
'Block Group 2, Census Tract 67, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006700'), ('block
group', '2'))),
'Block Group 1, Census Tract 67, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006700'), ('block
group', '1'))),
'Block Group 4, Census Tract 67, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006700'), ('block
group', '4'))),
'Block Group 7, Census Tract 67, Philadelphia County, Pennsylvania':
censusgeo([('state', '42'), ('county', '101'), ('tract', '006700'), ('block
group', '7'))),
'Block Group 5, Census Tract 67, Philadelphia County, Pennsylvania':

```

```

censusgeo(((('state', '42'), ('county', '101'), ('tract', '006700'), ('block
group', '5'))),
'Block Group 2, Census Tract 66, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '006600'), ('block
group', '2'))),
'Block Group 2, Census Tract 91, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '009100'), ('block
group', '2'))),
'Block Group 1, Census Tract 161, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '016100'), ('block
group', '1'))),
'Block Group 4, Census Tract 161, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '016100'), ('block
group', '4'))),
'Block Group 3, Census Tract 169.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '016901'), ('block
group', '3'))),
'Block Group 1, Census Tract 151.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '015101'), ('block
group', '1'))),
'Block Group 2, Census Tract 151.02, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '015102'), ('block
group', '2'))),
'Block Group 1, Census Tract 365.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '036501'), ('block
group', '1'))),
'Block Group 3, Census Tract 177.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '017701'), ('block
group', '3'))),
'Block Group 2, Census Tract 263.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026301'), ('block
group', '2'))),
'Block Group 3, Census Tract 263.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '026301'), ('block
group', '3'))),
'Block Group 1, Census Tract 247, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '024700'), ('block
group', '1'))),
'Block Group 3, Census Tract 14, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '001400'), ('block
group', '3'))),
'Block Group 4, Census Tract 362.01, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '036201'), ('block
group', '4'))),
'Block Group 2, Census Tract 369, Philadelphia County, Pennsylvania':
censusgeo(((('state', '42'), ('county', '101'), ('tract', '036900'), ('block
group', '2'))),

```

```
'Block Group 2, Census Tract 373, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '037300'), ('block  
group', '2'))),  
'Block Group 4, Census Tract 372, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '037200'), ('block  
group', '4'))),  
'Block Group 3, Census Tract 383, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '038300'), ('block  
group', '3'))),  
'Block Group 1, Census Tract 9809, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '980900'), ('block  
group', '1'))),  
'Block Group 2, Census Tract 27.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '002702'), ('block  
group', '2'))),  
'Block Group 2, Census Tract 30.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003002'), ('block  
group', '2'))),  
'Block Group 1, Census Tract 30.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003001'), ('block  
group', '1'))),  
'Block Group 2, Census Tract 30.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '003001'), ('block  
group', '2'))),  
'Block Group 4, Census Tract 28.02, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '002802'), ('block  
group', '4'))),  
'Block Group 3, Census Tract 87.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '008701'), ('block  
group', '3'))),  
'Block Group 2, Census Tract 167.01, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '016701'), ('block  
group', '2'))),  
'Block Group 3, Census Tract 286, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '028600'), ('block  
group', '3'))),  
'Block Group 6, Census Tract 286, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '028600'), ('block  
group', '6'))),  
'Block Group 2, Census Tract 1, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '000100'), ('block  
group', '2'))),  
'Block Group 2, Census Tract 13, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '001300'), ('block  
group', '2'))),  
'Block Group 4, Census Tract 191, Philadelphia County, Pennsylvania':  
censusgeo([('state', '42'), ('county', '101'), ('tract', '019100'), ('block
```

```
group', '4'))),  
...}
```

0.2.8 ACS: Occupancy Status

Data shows places that are occupied or vacant

```
[ ]: censusdata.printtable(censusdata.censustable('acs5', 2019, 'B25002'))  
↳#selecting the occupancy table and it shows the columns available
```

Variable	Table	Label
Type		
B25002_001E	OCCUPANCY STATUS	!! Estimate Total:
int		
B25002_002E	OCCUPANCY STATUS	!!! Estimate Total: Occupied
int		
B25002_003E	OCCUPANCY STATUS	!!! Estimate Total: Vacant
int		

```
[ ]: acs_occupancy = censusdata.download('acs5', 2019,  
censusdata.censusgeo([('state', '42'), # PS State  
('county', '101'), # philadelphia county  
('block group', '*')]), #all blockgroups  
['B25002_001E', 'B25002_002E', 'B25002_003E'])  
↳selecting which columns I would like to use  
  
acs_occupancy.rename(columns = {'B25002_001E': 'total',  
'B25002_002E': 'occupied',  
'B25002_003E' : 'vacant'}, inplace = True)  
↳renaming all columns  
  
acs_occupancy.to_csv('data/acs/occupancy.csv') # download the data in my  
↳computer  
acs_occupancy.head()
```

```
[ ]:  
total    occupied    vacant  
Block Group 1, Census Tract 9807, Philadelphia ...      0        0        0  
Block Group 3, Census Tract 27.01, Philadelphia...     707       616       91  
Block Group 2, Census Tract 337.01, Philadelphi...    400       400        0  
Block Group 3, Census Tract 337.01, Philadelphi...   1451      1340      111  
Block Group 2, Census Tract 205, Philadelphia C...    774       668      106
```

```
[ ]: #converting the first column into different columns for county, census tract and census blockgroup
acs_occupancy = acs_occupancy.reset_index() #resetting the index
acs_occupancy['index'] = acs_occupancy['index'].astype(str) # turning all values into string
acs_occupancy[['census_info','county', 'census_tract', 'census_blockgroup']] = acs_occupancy['index'].str.split('>', expand = True) #splitting the column based on '>'
acs_occupancy.head()
```

```
[ ]:                                     index  total  occupied  vacant  \
0  Block Group 1, Census Tract 9807, Philadelphia...      0       0       0
1  Block Group 3, Census Tract 27.01, Philadelphia...    707     616      91
2  Block Group 2, Census Tract 337.01, Philadelphia...   400     400       0
3  Block Group 3, Census Tract 337.01, Philadelphia...  1451    1340     111
4  Block Group 2, Census Tract 205, Philadelphia ...   774     668     106

                                     census_info      county  \
0  Block Group 1, Census Tract 9807, Philadelphia...  county:101
1  Block Group 3, Census Tract 27.01, Philadelphia...  county:101
2  Block Group 2, Census Tract 337.01, Philadelphia...  county:101
3  Block Group 3, Census Tract 337.01, Philadelphia...  county:101
4  Block Group 2, Census Tract 205, Philadelphia ...  county:101

           census_tract  census_blockgroup
0        tract:980700    block group:1
1        tract:002701    block group:3
2        tract:033701    block group:2
3        tract:033701    block group:3
4        tract:020500    block group:2
```

```
[ ]: #removing unnecessary words from the new columns eg county, tract, blockgroup
acs_occupancy['county'] = acs_occupancy['county'].str.replace('county:', '', regex = False)
acs_occupancy['census_tract'] = acs_occupancy['census_tract'].str.replace('tract:', '', regex = False)
acs_occupancy['census_blockgroup'] = acs_occupancy['census_blockgroup'].str.replace('block group:', '', regex = False)
```

```
[ ]: acs_occupancy.head()
```

```
[ ]:                                     index  total  occupied  vacant  \
0  Block Group 1, Census Tract 9807, Philadelphia...      0       0       0
1  Block Group 3, Census Tract 27.01, Philadelphia...    707     616      91
2  Block Group 2, Census Tract 337.01, Philadelphia...   400     400       0
3  Block Group 3, Census Tract 337.01, Philadelphia...  1451    1340     111
4  Block Group 2, Census Tract 205, Philadelphia ...   774     668     106
```

```

census_info county census_tract \
0 Block Group 1, Census Tract 9807, Philadelphia... 101 980700
1 Block Group 3, Census Tract 27.01, Philadelphia... 101 002701
2 Block Group 2, Census Tract 337.01, Philadelphia... 101 033701
3 Block Group 3, Census Tract 337.01, Philadelphia... 101 033701
4 Block Group 2, Census Tract 205, Philadelphia ... 101 020500

census_blockgroup
0 1
1 3
2 2
3 3
4 2

```

[]: acs_occupancy.dtypes # datatype of columns

```

[ ]: index          object
total           int64
occupied        int64
vacant          int64
census_info     object
county          object
census_tract    object
census_blockgroup object
dtype: object

```

[]: acs_occupancy['perc_vacant'] = acs_occupancy['vacant']/
→acs_occupancy['total'] #creating new column which has percentage of vacant lot

[]: acs_occupancy['perc_vacant'].mean() #Mean percentage vacant lots for each block
→group is 0.12578075679635622

0.12578075679635622

[]: acs_occupancy['perc_vacant'].median() #Median percentage vacant lots for each
→block group is 0.12578075679635622

0.10602569473537216

[]: acs_occupancy.describe(include = 'all') # some block groups have 54% vacant
→places

	index	total
count	1336	1336.000000
unique	1336	NaN
top	Block Group 1, Census Tract 9807, Philadelphia...	NaN

freq		1	NaN
mean		NaN	513.440120
std		NaN	254.862223
min		NaN	0.000000
25%		NaN	348.000000
50%		NaN	460.500000
75%		NaN	619.500000
max		NaN	2043.000000

	occupied	vacant	\
count	1336.000000	1336.000000	
unique	NaN	NaN	
top	NaN	NaN	
freq	NaN	NaN	
mean	450.102545	63.337575	
std	232.346504	58.294697	
min	0.000000	0.000000	
25%	303.000000	22.750000	
50%	397.000000	51.000000	
75%	554.250000	92.000000	
max	1850.000000	460.000000	

	census_info	county	census_tract	\
count		1336	1336	1336
unique		1336	1	384
top	Block Group 1, Census Tract 9807, Philadelphia...	101	026600	
freq		1	1336	8
mean		NaN	NaN	NaN
std		NaN	NaN	NaN
min		NaN	NaN	NaN
25%		NaN	NaN	NaN
50%		NaN	NaN	NaN
75%		NaN	NaN	NaN
max		NaN	NaN	NaN

	census_blockgroup	perc_vacant
count	1336	1326.000000
unique	8	NaN
top	1	NaN
freq	384	NaN
mean	NaN	0.125781
std	NaN	0.103121
min	NaN	0.000000
25%	NaN	0.049860
50%	NaN	0.106026
75%	NaN	0.185122
max	NaN	0.542645

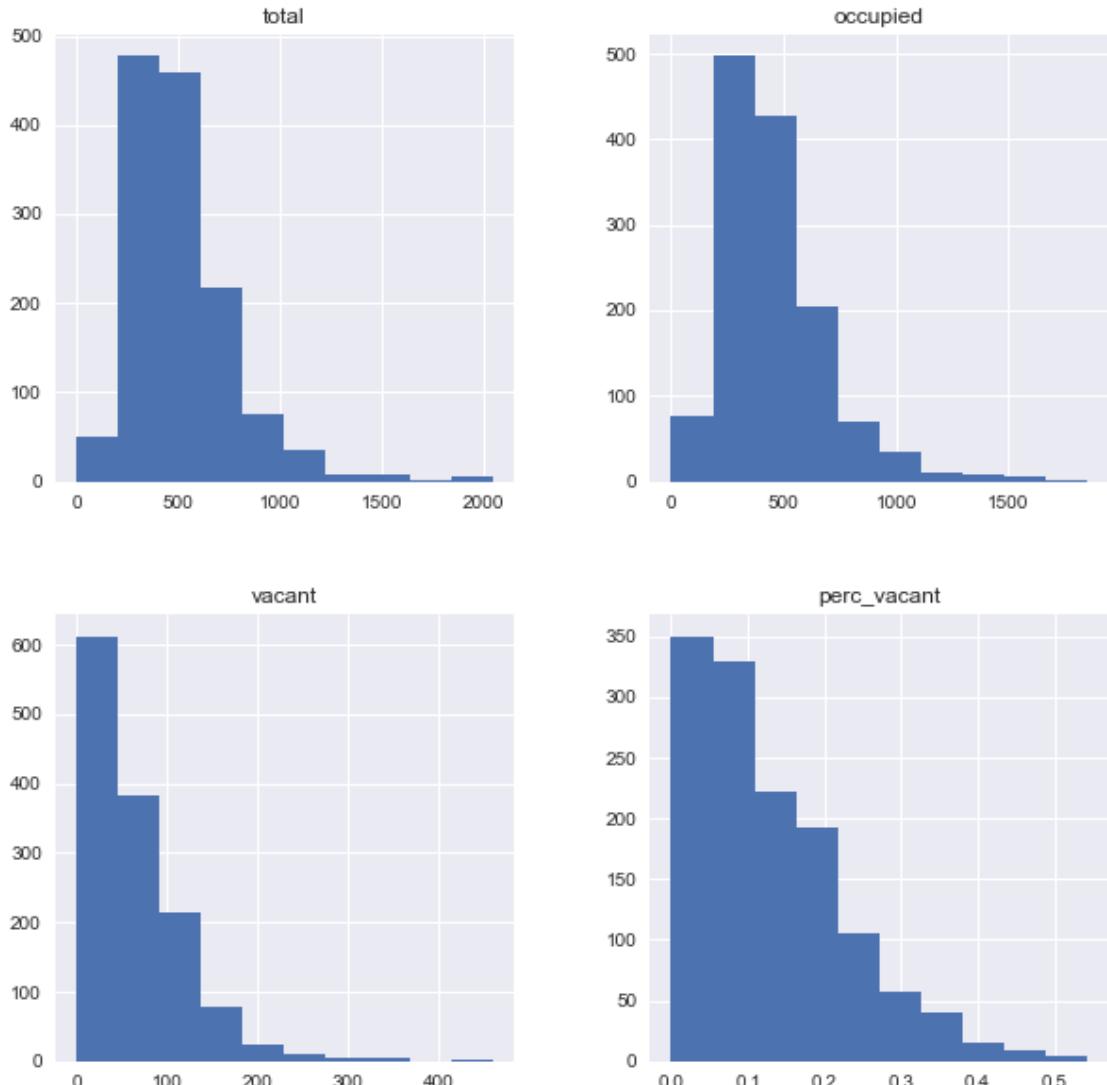
```
[ ]: acs_occupancy.sum() #there are less vacant lots than occupies lots
```

```
[ ]: index           Block Group 1, Census Tract 9807, Philadelphia...
      total            685956
      occupied          601337
      vacant            84619
      census_info       Block Group 1, Census Tract 9807, Philadelphia...
      county             101 101 101 101 101 101 101 101 101 101 1...
      census_tract        980700 002701 033701 033701 020500 021800 021...
      census_blockgroup   1 3 2 3 2 1 2 1 4 2 1 2 1 3 1 6 2 4 1 2 1 5 3...
      perc_vacant         166.785284
      dtype: object
```

```
[ ]: fig, ax = plt.subplots(figsize =(10,10))
acs_occupancy.hist(ax = ax) # vacant places are skewed
```

```
/var/folders/6p/wpw9qml157530xkxqkkhprrf40000gn/T/ipykernel_21232/2571989370.py:2
: UserWarning: To output multiple subplots, the figure containing the passed
axes is being cleared
    acs_occupancy.hist(ax = ax) # vacant places are skewed
```

```
[ ]: array([ [,
              <AxesSubplot:title={'center':'occupied'}>],
             [,
              <AxesSubplot:title={'center':'perc_vacant'}>]], dtype=object)
```



```
[ ]: #converting tract and blockgroups into integer so that we can join dataset
      ↵easily
acs_occupancy['census_tract'] = acs_occupancy['census_tract'].astype(int)
acs_occupancy['census_blockgroup'] = acs_occupancy['census_blockgroup'].
      ↵astype(int)

census_blockgroups['TRACTCE10'] = census_blockgroups['TRACTCE10'].astype(int)
census_blockgroups['BLKGRPCE10'] = census_blockgroups['BLKGRPCE10'].astype(int)

[ ]: #merging occupancy acs dataset with blockgroup shape file
occupancy = census_blockgroups.merge(acs_occupancy, how='left',
      ↵left_on=["TRACTCE10", "BLKGRPCE10"],
      ↵right_on=["census_tract", "census_blockgroup"])
```

```
[ ]: occupancy.head()

[ ]:   OBJECTID STATEFP10 COUNTYFP10  TRACTCE10  BLKGRPCE10      GEOID10 \
0          1        42       101     10800           1 421010108001
1          2        42       101     10800           2 421010108002
2          3        42       101     10900           2 421010109002
3          4        42       101     11000           2 421010110002
4          5        42       101     11000           1 421010110001

      NAMELSAD10 MTFCC10 FUNCSTAT10  ALAND10 ... \
0  Block Group 1    G5030            S  161887 ...
1  Block Group 2    G5030            S  103778 ...
2  Block Group 2    G5030            S   43724 ...
3  Block Group 2    G5030            S  108966 ...
4  Block Group 1    G5030            S  142244 ...

                           geometry \
0  POLYGON ((-75.19851 39.96945, -75.19744 39.969...
1  POLYGON ((-75.19783 39.96571, -75.20006 39.965...
2  POLYGON ((-75.18766 39.96450, -75.18755 39.963...
3  POLYGON ((-75.20984 39.97351, -75.21221 39.973...
4  POLYGON ((-75.19855 39.97330, -75.19854 39.973...

                           index total occupied vacant \
0  Block Group 1, Census Tract 108, Philadelphia ...  243     202     41
1  Block Group 2, Census Tract 108, Philadelphia ...  360     239    121
2  Block Group 2, Census Tract 109, Philadelphia ...  236     221     15
3  Block Group 2, Census Tract 110, Philadelphia ...  478     348    130
4  Block Group 1, Census Tract 110, Philadelphia ...  240     187     53

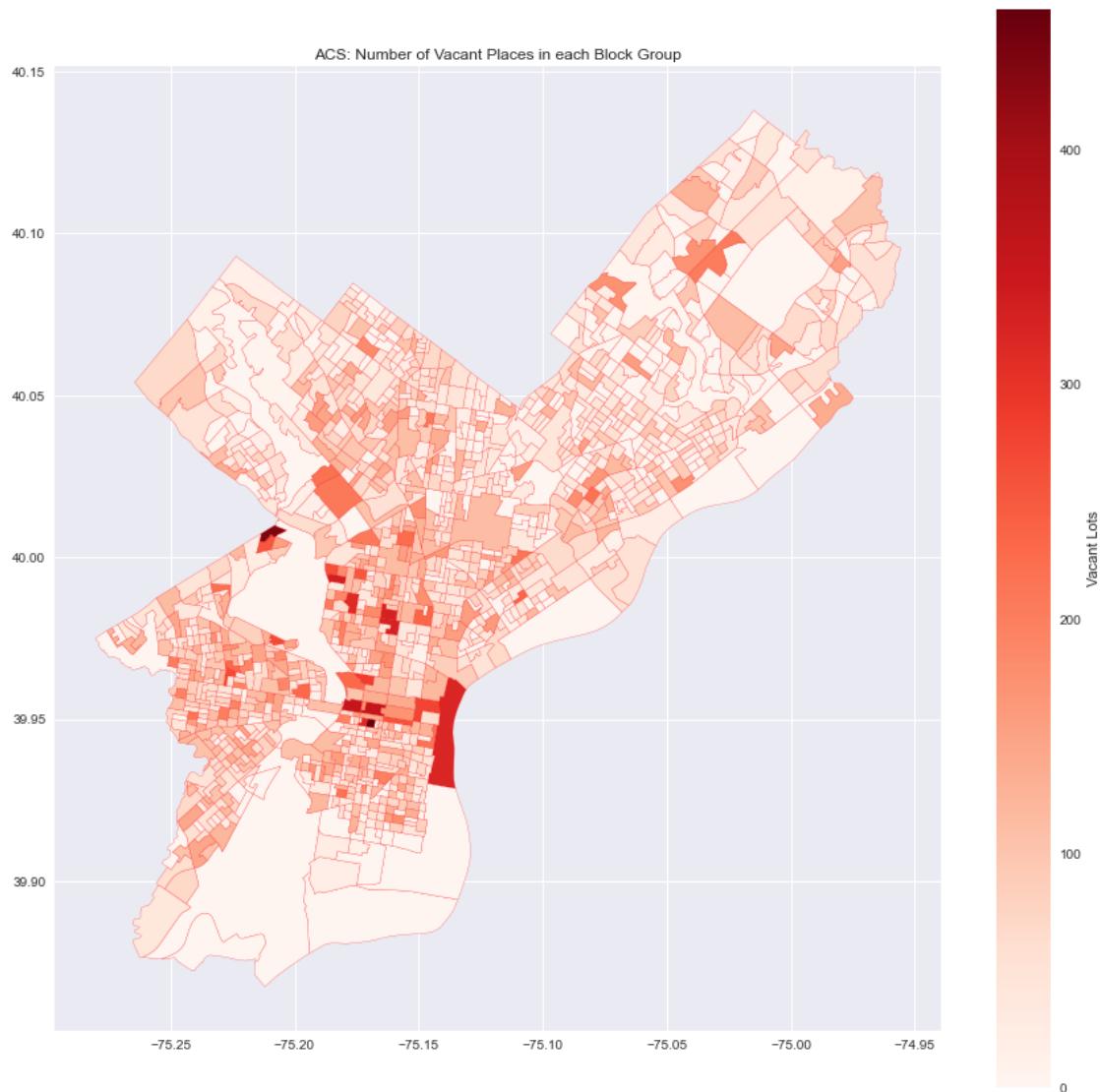
                           census_info county  census_tract \
0  Block Group 1, Census Tract 108, Philadelphia ...    101      10800
1  Block Group 2, Census Tract 108, Philadelphia ...    101      10800
2  Block Group 2, Census Tract 109, Philadelphia ...    101      10900
3  Block Group 2, Census Tract 110, Philadelphia ...    101      11000
4  Block Group 1, Census Tract 110, Philadelphia ...    101      11000

  census_blockgroup  perc_vacant
0                  1    0.168724
1                  2    0.336111
2                  2    0.063559
3                  2    0.271967
4                  1    0.220833

[5 rows x 25 columns]
```

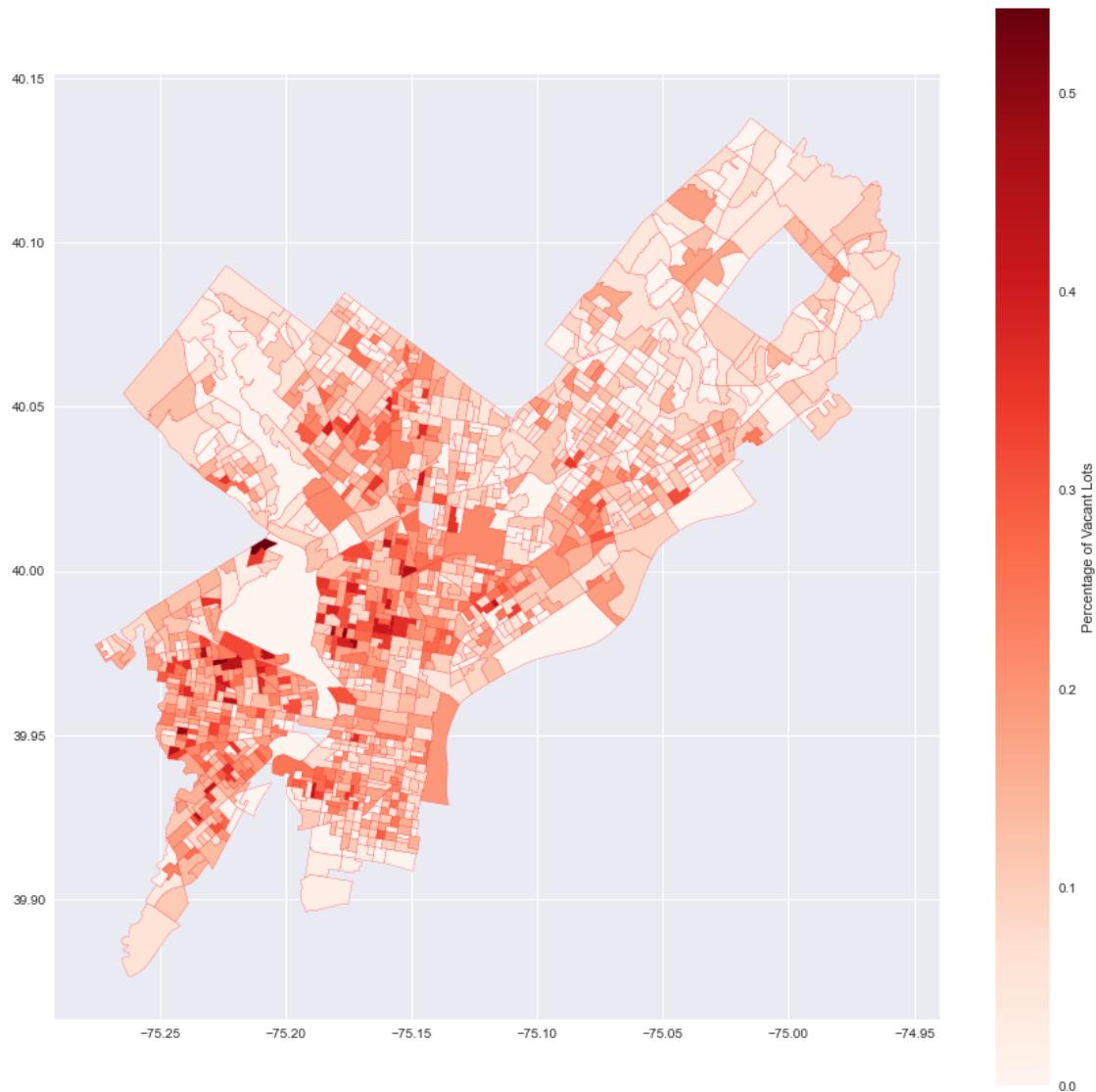
```
[ ]: #plotting vacant lots on each block group
fig, ax = plt.subplots(figsize=(15,15))
plt.style.use('seaborn')
plt.title('ACS: Number of Vacant Places in each Block Group')
#census_blockgroups.to_crs("EPSG:4269").plot(ax=ax, color='white', ↴
    ↴edgecolor='black')
occupancy.plot(ax=ax, column='vacant',
    edgecolor='red', linewidth=.2,
    cmap='Reds', legend=True,
    legend_kwds={'label': 'Vacant Lots'})
```

```
[ ]: <AxesSubplot:title={'center':'ACS: Number of Vacant Places in each Block Group'}>
```



```
[ ]: #plotting percentage of vacant lots on each block group
#this shows that when you do percentage, the plot changes completely. ↴
→ Think would be a better variable to consider than just looking at vacant lots
fig, ax = plt.subplots(figsize=(15,15))
plt.style.use('seaborn')
plt.title('ACS: Percentage of Vacant Places in each Block Group')
#census_blockgroups.to_crs("EPSG:4269").plot(ax=ax, color='white', ↴
→ edgecolor='black')
occupancy.plot(ax=ax, column='perc_vacant',
                edgecolor='red', linewidth=.2,
                cmap='Reds', legend=True,
                legend_kwds={'label': 'Percentage of Vacant Lots'})
```

[]: <AxesSubplot:>



0.2.9 ACS: Vacancy Type

```
[ ]: censusdata.printtable(censusdata.censustable('acs5', 2019, 'B25004'))#_
    ↳selecting type of vacancy table and the columns included
```

Variable	Table	Label
	Type	
<hr/>		
B25004_001E	VACANCY STATUS	!! Estimate Total:
int		
B25004_002E	VACANCY STATUS	!!! Estimate Total: For rent
int		
B25004_003E	VACANCY STATUS not occupied	!!! Estimate Total: Rented, int
B25004_004E	VACANCY STATUS only	!!! Estimate Total: For sale int
B25004_005E	VACANCY STATUS occupied	!!! Estimate Total: Sold, not int
B25004_006E	VACANCY STATUS seasonal, recreational, or occ	!!! Estimate Total: For int
B25004_007E	VACANCY STATUS migrant workers	!!! Estimate Total: For int
B25004_008E	VACANCY STATUS vacant	!!! Estimate Total: Other int
<hr/>		
<hr/>		

```
[ ]: acs_vacant_type = censusdata.download('acs5', 2019,
    censusdata.censusgeo([('state', '42'),
        ('county', '101'), # philadelphia county
        ('block group', '*')]),# all blockgroups
        ['B25004_001E', 'B25004_002E', 'B25004_003E', 'B25004_004E', 'B25004_005E', 'B25004_006E', 'B25004_007E', 'B25004_008E'])#selecting all
    ↳columns

acs_vacant_type.rename(columns = {'B25004_001E': 'total',
    'B25004_002E': 'for_rent',
    'B25004_003E' : 'rented_not_occupied',
    'B25004_004E' : 'for_sale_only',
    'B25004_005E' : 'sold_not_occupied',
    'B25004_006E' : 'seasonal_recreational',
    'B25004_007E' : 'migrant_workers',
```

```
'B25004_008E' : 'other'}, inplace = ↵True) #renaming columns
```

```
acs_vacant_type.to_csv('data/acs/vacant_type.csv') #downloading dataset
acs_vacant_type.head()
```

[]:

	total	for_rent	\
Block Group 1, Census Tract 9807, Philadelphia ...	0	0	
Block Group 3, Census Tract 27.01, Philadelphia...	91	0	
Block Group 2, Census Tract 337.01, Philadelphi...	0	0	
Block Group 3, Census Tract 337.01, Philadelphi...	111	73	
Block Group 2, Census Tract 205, Philadelphia C...	106	0	
			rented_not_occupied \
Block Group 1, Census Tract 9807, Philadelphia ...	0		0
Block Group 3, Census Tract 27.01, Philadelphia...	0		0
Block Group 2, Census Tract 337.01, Philadelphi...	0		0
Block Group 3, Census Tract 337.01, Philadelphi...	0		0
Block Group 2, Census Tract 205, Philadelphia C...	0		0
			for_sale_only \
Block Group 1, Census Tract 9807, Philadelphia ...	0		0
Block Group 3, Census Tract 27.01, Philadelphia...	48		48
Block Group 2, Census Tract 337.01, Philadelphi...	0		0
Block Group 3, Census Tract 337.01, Philadelphi...	0		0
Block Group 2, Census Tract 205, Philadelphia C...	0		0
			sold_not_occupied \
Block Group 1, Census Tract 9807, Philadelphia ...	0		0
Block Group 3, Census Tract 27.01, Philadelphia...	0		0
Block Group 2, Census Tract 337.01, Philadelphi...	0		0
Block Group 3, Census Tract 337.01, Philadelphi...	38		38
Block Group 2, Census Tract 205, Philadelphia C...	0		0
			seasonal_recreational \
Block Group 1, Census Tract 9807, Philadelphia ...	0		0
Block Group 3, Census Tract 27.01, Philadelphia...	0		0
Block Group 2, Census Tract 337.01, Philadelphi...	0		0
Block Group 3, Census Tract 337.01, Philadelphi...	0		0
Block Group 2, Census Tract 205, Philadelphia C...	0		0
			migrant_workers other
Block Group 1, Census Tract 9807, Philadelphia ...	0	0	
Block Group 3, Census Tract 27.01, Philadelphia...	0	43	
Block Group 2, Census Tract 337.01, Philadelphi...	0	0	
Block Group 3, Census Tract 337.01, Philadelphi...	0	0	
Block Group 2, Census Tract 205, Philadelphia C...	0	106	

```
[ ]: acs_vacant_type.describe(include = 'all')#describing data
```

	total	for_rent	rented_not_occupied	for_sale_only	\
count	1336.000000	1336.000000	1336.000000	1336.000000	
mean	63.337575	14.039671	3.464072	4.755240	
std	58.294697	25.349141	11.301529	13.120158	
min	0.000000	0.000000	0.000000	0.000000	
25%	22.750000	0.000000	0.000000	0.000000	
50%	51.000000	0.000000	0.000000	0.000000	
75%	92.000000	22.000000	0.000000	0.000000	
max	460.000000	231.000000	105.000000	135.000000	
	sold_not_occupied	seasonal_recreational	migrant_workers	other	
count	1336.000000	1336.000000	1336.000000	1336.000000	
mean	3.907186	2.559880	0.142964	34.468563	
std	11.882524	11.040043	3.346382	40.548789	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	25.000000	
75%	0.000000	0.000000	0.000000	52.000000	
max	147.000000	144.000000	114.000000	334.000000	

```
[ ]: acs_vacant_type.sum() #most of the vacant places are "other"
```

	total	for_rent	rented_not_occupied	for_sale_only	\
count	1336.000000	1336.000000	1336.000000	1336.000000	
mean	63.337575	14.039671	3.464072	4.755240	
std	58.294697	25.349141	11.301529	13.120158	
min	0.000000	0.000000	0.000000	0.000000	
25%	22.750000	0.000000	0.000000	0.000000	
50%	51.000000	0.000000	0.000000	0.000000	
75%	92.000000	22.000000	0.000000	0.000000	
max	460.000000	231.000000	105.000000	135.000000	

```
[ ]: migrant_workers          0.002257
      seasonal_recreational  0.040416
      rented_not_occupied   0.054692
      sold_not_occupied     0.061688
      for_sale_only         0.075078
      for_rent              0.221664
      other                 0.544204
      total                 1.000000
      dtype: float64
```

```
[ ]: #splitting column to have separate columns for blockgroup and tract.
acs_vacant_type = acs_vacant_type.reset_index() # reset index
acs_vacant_type['index'] = acs_vacant_type['index'].astype(str) # turning to
→string
acs_vacant_type[['census_info', 'county', 'census_tract', 'census_blockgroup']] ↴
→= acs_vacant_type['index'].str.split('>', expand = True)
acs_vacant_type.head()
```

```
[ ]:
index total for_rent \
0 Block Group 1, Census Tract 9807, Philadelphia... 0 0
1 Block Group 3, Census Tract 27.01, Philadelphia... 91 0
2 Block Group 2, Census Tract 337.01, Philadelphia... 0 0
3 Block Group 3, Census Tract 337.01, Philadelphia... 111 73
4 Block Group 2, Census Tract 205, Philadelphia ... 106 0

rented_not_occupied for_sale_only sold_not_occupied \
0 0 0 0
1 0 48 0
2 0 0 0
3 0 0 38
4 0 0 0

seasonal_recreational migrant_workers other \
0 0 0 0
1 0 0 43
2 0 0 0
3 0 0 0
4 0 0 106

census_info county \
0 Block Group 1, Census Tract 9807, Philadelphia... county:101
1 Block Group 3, Census Tract 27.01, Philadelphia... county:101
2 Block Group 2, Census Tract 337.01, Philadelphia... county:101
3 Block Group 3, Census Tract 337.01, Philadelphia... county:101
4 Block Group 2, Census Tract 205, Philadelphia ... county:101

census_tract census_blockgroup
0 tract:980700 block group:1
1 tract:002701 block group:3
2 tract:033701 block group:2
3 tract:033701 block group:3
4 tract:020500 block group:2
```

```
[ ]: #removing unnecessary words from the columns
acs_vacant_type['county'] = acs_vacant_type['county'].str.replace('county:', ↴
→'', regex = False)
```

```

acs_vacant_type['census_tract'] = acs_vacant_type['census_tract'].str.
    ↪replace('tract:', '', regex = False)
acs_vacant_type['census_blockgroup'] = acs_vacant_type['census_blockgroup'].str.
    ↪replace('block group:', '', regex = False)

#converting to integer
acs_vacant_type['census_tract'] = acs_vacant_type['census_tract'].astype(int)
acs_vacant_type['census_blockgroup'] = acs_vacant_type['census_blockgroup'].
    ↪astype(int)

#converting to integer
census_blockgroups['TRACTCE10'] = census_blockgroups['TRACTCE10'].astype(int)
census_blockgroups['BLKGRPCE10'] = census_blockgroups['BLKGRPCE10'].astype(int)

acs_vacant_type.head()

```

```
[ ]:          index  total  for_rent  \
0  Block Group 1, Census Tract 9807, Philadelphia...      0      0
1  Block Group 3, Census Tract 27.01, Philadelphia...     91      0
2  Block Group 2, Census Tract 337.01, Philadelphia...      0      0
3  Block Group 3, Census Tract 337.01, Philadelphia...    111     73
4  Block Group 2, Census Tract 205, Philadelphia ...     106      0

      rented_not_occupied  for_sale_only  sold_not_occupied  \
0                  0              0                  0
1                  0              48                  0
2                  0              0                  0
3                  0              0                  38
4                  0              0                  0

      seasonal_recreational  migrant_workers  other  \
0                      0              0              0
1                      0              0             43
2                      0              0              0
3                      0              0              0
4                      0              0            106

      census_info  county  census_tract  \
0  Block Group 1, Census Tract 9807, Philadelphia...    101    980700
1  Block Group 3, Census Tract 27.01, Philadelphia...    101      2701
2  Block Group 2, Census Tract 337.01, Philadelphia...    101    33701
3  Block Group 3, Census Tract 337.01, Philadelphia...    101    33701
4  Block Group 2, Census Tract 205, Philadelphia ...    101    20500

      census_blockgroup
0                  1
1                  3
```

```

2          2
3          3
4          2

[ ]: #merging vacant type file with shape file of blockgroup based on tract and
      ↪blockgroup
vacant_type = census_blockgroups.merge(acs_vacant_type, how='left', ↪
      ↪left_on=["TRACTCE10", "BLKGRPCE10"], ↪
      ↪right_on=["census_tract", "census_blockgroup"])
vacant_type.head()

```

```

[ ]:   OBJECTID STATEFP10 COUNTYFP10  TRACTCE10  BLKGRPCE10       GEOID10 \
0           1        42       101     10800           1 421010108001
1           2        42       101     10800           2 421010108002
2           3        42       101     10900           2 421010109002
3           4        42       101     11000           2 421010110002
4           5        42       101     11000           1 421010110001

      NAMELSAD10 MTFCC10 FUNCSTAT10    ALAND10 ... rented_not_occupied \
0  Block Group 1    G5030            S  161887 ...                      0
1  Block Group 2    G5030            S  103778 ...                      0
2  Block Group 2    G5030            S   43724 ...                      8
3  Block Group 2    G5030            S  108966 ...                     32
4  Block Group 1    G5030            S  142244 ...                     33

      for_sale_only sold_not_occupied seasonal_recreational migrant_workers \
0                  0                 0                   0                   0
1                  0                 21                  0                   0
2                  0                 0                   0                   0
3                 25                 0                   0                   0
4                  0                 0                   0                   0

      other                         census_info county \
0    41  Block Group 1, Census Tract 108, Philadelphia ...      101
1    82  Block Group 2, Census Tract 108, Philadelphia ...      101
2     7  Block Group 2, Census Tract 109, Philadelphia ...      101
3    73  Block Group 2, Census Tract 110, Philadelphia ...      101
4    20  Block Group 1, Census Tract 110, Philadelphia ...      101

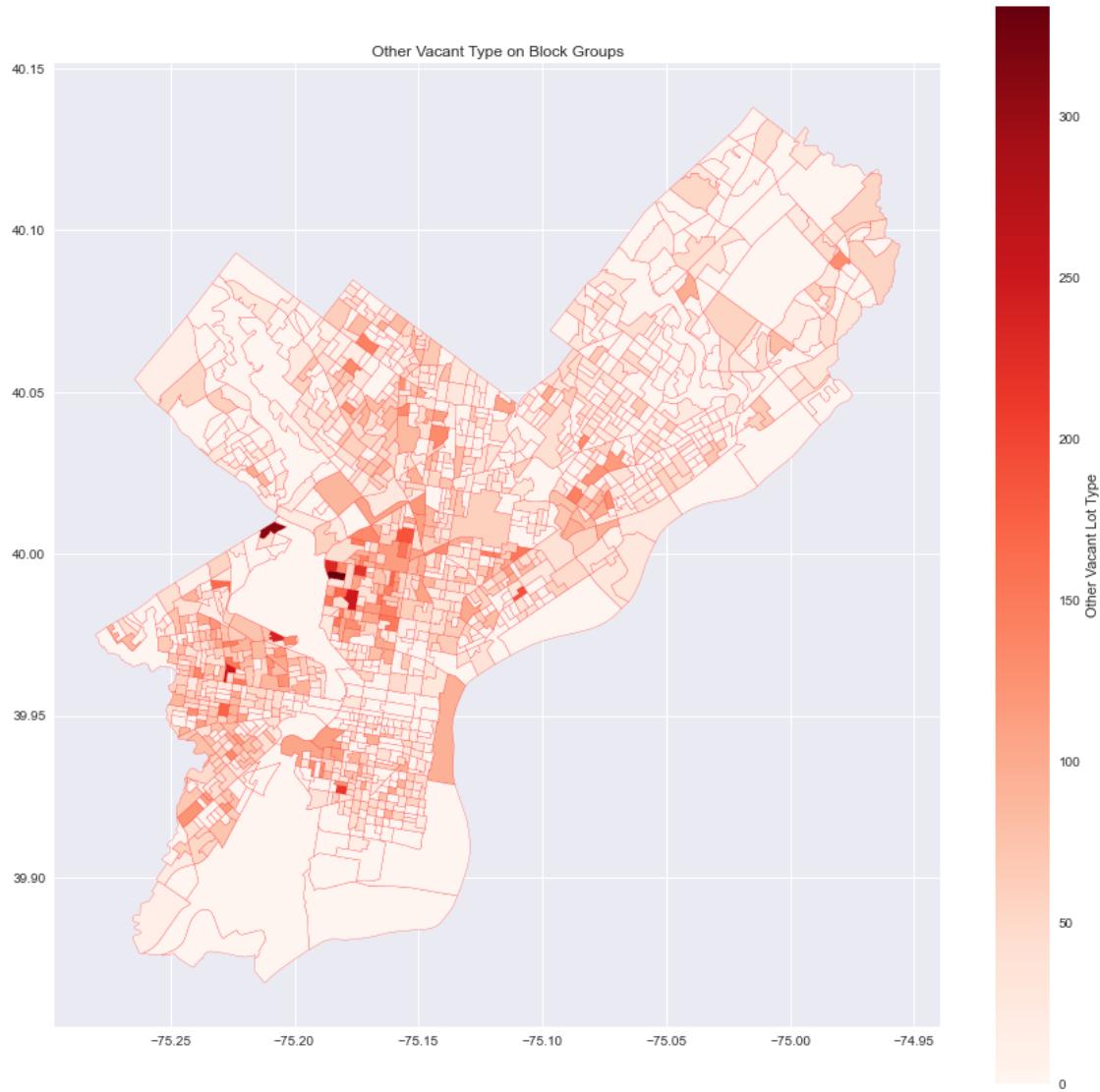
      census_tract  census_blockgroup
0           10800             1
1           10800             2
2           10900             2
3           11000             2
4           11000             1

```

[5 rows x 29 columns]

```
[ ]: fig, ax = plt.subplots(figsize=(15,15))
plt.style.use('seaborn')
plt.title("Other Vacant Type on Block Groups"
vacant_type.plot(ax=ax, column='other',
                  edgecolor='red', linewidth=.2,
                  cmap='Reds', legend=True,
                  legend_kwds={'label': 'Other Vacant Lot Type'})
```

```
[ ]: <AxesSubplot:title={'center':'Other Vacant Type on Block Groups'}>
```



0.2.10 ACS : Gender

Dataset was mostly used to find population in each blockgroup

```
[ ]: censusdata.printtable(censusdata.censustable('acs5', 2019, 'B01001'))#age and
      ↵sex
```

Variable	Table	Label
	Type	
<hr/>		
B01001_001E	SEX BY AGE	!! Estimate Total:
int		
B01001_002E	SEX BY AGE	!!! Estimate Total: Male:
int		
B01001_003E	SEX BY AGE	!!!!! Estimate Total: Male:
Under 5 years	int	!!!!! Estimate Total: Male: 5
to 9 years	int	
B01001_005E	SEX BY AGE	!!!!! Estimate Total: Male:
10 to 14 years	int	!!!!! Estimate Total: Male:
B01001_006E	SEX BY AGE	!!!!! Estimate Total: Male:
15 to 17 years	int	!!!!! Estimate Total: Male:
B01001_007E	SEX BY AGE	!!!!! Estimate Total: Male:
18 and 19 years	int	!!!!! Estimate Total: Male:
B01001_008E	SEX BY AGE	!!!!! Estimate Total: Male:
20 years	int	!!!!! Estimate Total: Male:
B01001_009E	SEX BY AGE	!!!!! Estimate Total: Male:
21 years	int	!!!!! Estimate Total: Male:
B01001_010E	SEX BY AGE	!!!!! Estimate Total: Male:
22 to 24 years	int	!!!!! Estimate Total: Male:
B01001_011E	SEX BY AGE	!!!!! Estimate Total: Male:
25 to 29 years	int	!!!!! Estimate Total: Male:
B01001_012E	SEX BY AGE	!!!!! Estimate Total: Male:
30 to 34 years	int	!!!!! Estimate Total: Male:
B01001_013E	SEX BY AGE	!!!!! Estimate Total: Male:
35 to 39 years	int	!!!!! Estimate Total: Male:
B01001_014E	SEX BY AGE	!!!!! Estimate Total: Male:
40 to 44 years	int	!!!!! Estimate Total: Male:
B01001_015E	SEX BY AGE	!!!!! Estimate Total: Male:
45 to 49 years	int	!!!!! Estimate Total: Male:
B01001_016E	SEX BY AGE	!!!!! Estimate Total: Male:
50 to 54 years	int	!!!!! Estimate Total: Male:
B01001_017E	SEX BY AGE	!!!!! Estimate Total: Male:
55 to 59 years	int	!!!!! Estimate Total: Male:
B01001_018E	SEX BY AGE	!!!!! Estimate Total: Male:
60 and 61 years	int	!!!!! Estimate Total: Male:
B01001_019E	SEX BY AGE	!!!!! Estimate Total: Male:
62 to 64 years	int	!!!!! Estimate Total: Male:
B01001_020E	SEX BY AGE	!!!!! Estimate Total: Male:
65 and 66 years	int	

B01001_021E SEX BY AGE	!!! !! Estimate Total: Male:
67 to 69 years int	
B01001_022E SEX BY AGE	!!! !! Estimate Total: Male:
70 to 74 years int	
B01001_023E SEX BY AGE	!!! !! Estimate Total: Male:
75 to 79 years int	
B01001_024E SEX BY AGE	!!! !! Estimate Total: Male:
80 to 84 years int	
B01001_025E SEX BY AGE	!!! !! Estimate Total: Male:
85 years and over int	
B01001_026E SEX BY AGE	!!! !! Estimate Total: Female:
int	
B01001_027E SEX BY AGE	!!! !! Estimate Total: Female:
Under 5 years int	
B01001_028E SEX BY AGE	!!! !! Estimate Total: Female:
5 to 9 years int	
B01001_029E SEX BY AGE	!!! !! Estimate Total: Female:
10 to 14 years int	
B01001_030E SEX BY AGE	!!! !! Estimate Total: Female:
15 to 17 years int	
B01001_031E SEX BY AGE	!!! !! Estimate Total: Female:
18 and 19 years int	
B01001_032E SEX BY AGE	!!! !! Estimate Total: Female:
20 years int	
B01001_033E SEX BY AGE	!!! !! Estimate Total: Female:
21 years int	
B01001_034E SEX BY AGE	!!! !! Estimate Total: Female:
22 to 24 years int	
B01001_035E SEX BY AGE	!!! !! Estimate Total: Female:
25 to 29 years int	
B01001_036E SEX BY AGE	!!! !! Estimate Total: Female:
30 to 34 years int	
B01001_037E SEX BY AGE	!!! !! Estimate Total: Female:
35 to 39 years int	
B01001_038E SEX BY AGE	!!! !! Estimate Total: Female:
40 to 44 years int	
B01001_039E SEX BY AGE	!!! !! Estimate Total: Female:
45 to 49 years int	
B01001_040E SEX BY AGE	!!! !! Estimate Total: Female:
50 to 54 years int	
B01001_041E SEX BY AGE	!!! !! Estimate Total: Female:
55 to 59 years int	
B01001_042E SEX BY AGE	!!! !! Estimate Total: Female:
60 and 61 years int	
B01001_043E SEX BY AGE	!!! !! Estimate Total: Female:
62 to 64 years int	
B01001_044E SEX BY AGE	!!! !! Estimate Total: Female:
65 and 66 years int	

```

B01001_045E | SEX BY AGE           | !!! !! Estimate Total: Female:
67 to 69 years      | int
B01001_046E | SEX BY AGE           | !!! !! Estimate Total: Female:
70 to 74 years      | int
B01001_047E | SEX BY AGE           | !!! !! Estimate Total: Female:
75 to 79 years      | int
B01001_048E | SEX BY AGE           | !!! !! Estimate Total: Female:
80 to 84 years      | int
B01001_049E | SEX BY AGE           | !!! !! Estimate Total: Female:
85 years and over   | int
-----
```

```

[ ]: acs_gender = censusdata.download('acs5', 2019,
                                   censusdata.censusgeo([('state', '42'),
                                         ('county', '101'), # philadelphia county
                                         ('block group', '*')]),
                                   ['B01001_001E', 'B01001_002E', 'B01001_026E'])

acs_gender.rename(columns = {'B01001_001E': 'total',
                            'B01001_002E': 'Male',
                            'B01001_026E': 'Female'}, inplace = True)
→#include age here, group by under 18, 18-64, 65 and over

acs_gender.to_csv('data/acs/gender.csv')
acs_gender.head()
```

```

[ ]:                                     total  Male  Female
Block Group 1, Census Tract 9807, Philadelphia ...     0    0     0
Block Group 3, Census Tract 27.01, Philadelphia...  1955  1023    932
Block Group 2, Census Tract 337.01, Philadelphi...  976   541    435
Block Group 3, Census Tract 337.01, Philadelphi...  3859  1969   1890
Block Group 2, Census Tract 205, Philadelphia C...  1017   553    464
```

```

[ ]: #splitting column to have seperate columns for blockgroup and tract.
acs_gender = acs_gender.reset_index() # reset index
acs_gender['index'] = acs_gender['index'].astype(str)# turning to string
acs_gender[['census_info','county', 'census_tract', 'census_blockgroup']] =
→acs_gender['index'].str.split('>', expand = True)
acs_gender.head()
```

```

[ ]:                                     index  total  Male  Female \
0  Block Group 1, Census Tract 9807, Philadelphia...     0    0     0
1  Block Group 3, Census Tract 27.01, Philadelphia...  1955  1023    932
2  Block Group 2, Census Tract 337.01, Philadelph...  976   541    435
3  Block Group 3, Census Tract 337.01, Philadelph...  3859  1969   1890
4  Block Group 2, Census Tract 205, Philadelphia ...  1017   553    464
```

```

            census_info      county \
0  Block Group 1, Census Tract 9807, Philadelphia...  county:101
1  Block Group 3, Census Tract 27.01, Philadelphi...  county:101
2  Block Group 2, Census Tract 337.01, Philadelph...  county:101
3  Block Group 3, Census Tract 337.01, Philadelph...  county:101
4  Block Group 2, Census Tract 205, Philadelphia ...  county:101

    census_tract census_blockgroup
0      tract:980700      block group:1
1      tract:002701      block group:3
2      tract:033701      block group:2
3      tract:033701      block group:3
4      tract:020500      block group:2

[ ]: #removing unnecessary words from the columns
acs_gender['county'] = acs_gender['county'].str.replace('county:', '', regex = False)
acs_gender['census_tract'] = acs_gender['census_tract'].str.replace('tract:', '', regex = False)
acs_gender['census_blockgroup'] = acs_gender['census_blockgroup'].str.replace('block group:', '', regex = False)

#converting to integer
acs_gender['census_tract'] = acs_gender['census_tract'].astype(int)
acs_gender['census_blockgroup'] = acs_gender['census_blockgroup'].astype(int)

[ ]: acs_pop = acs_gender.drop(['index', 'census_info', 'county','Male', 'Female'],axis = 1)
acs_pop = acs_pop.rename(columns = {"total" : "total_pop"})#only including population dataset as we will be using this later
acs_pop.head()

[ ]:   total_pop  census_tract  census_blockgroup
0          0        980700              1
1        1955         2701              3
2         976        33701              2
3        3859        33701              3
4        1017        20500              2

```

0.2.11 Feature Engineering: ACS Occupancy, ACS Vacancy Type and Population Dataset

Occupancy, Vacancy and Population dataset is joined togather here based on census track and census block number. These files were also joined with the block group shape file to know the shape and area of each block group. Things that were calculated in the ACS dataset are percentage vacancy, percentage “other” vacancy, population density in km².

```
[ ]: #picking this dataset first as it was already joined with blockgroup shape file  
vacant_type.head()
```

```
[ ]:   OBJECTID STATEFP10 COUNTYFP10 TRACTCE10 BLKGRPCE10      GEOID10 \
0          1        42       101    10800           1 421010108001
1          2        42       101    10800           2 421010108002
2          3        42       101    10900           2 421010109002
3          4        42       101    11000           2 421010110002
4          5        42       101    11000           1 421010110001

      NAMELSAD10 MTFCC10 FUNCSTAT10 ALAND10 ... rented_not_occupied \
0  Block Group 1  G5030          S  161887 ... 0
1  Block Group 2  G5030          S  103778 ... 0
2  Block Group 2  G5030          S  43724 ... 8
3  Block Group 2  G5030          S  108966 ... 32
4  Block Group 1  G5030          S  142244 ... 33

      for_sale_only sold_not_occupied seasonal_recreational migrant_workers \
0            0                0                  0            0
1            0                21                 0            0
2            0                0                  0            0
3            25               0                  0            0
4            0                0                  0            0

      other             census_info county \
0    41  Block Group 1, Census Tract 108, Philadelphia ... 101
1    82  Block Group 2, Census Tract 108, Philadelphia ... 101
2     7  Block Group 2, Census Tract 109, Philadelphia ... 101
3    73  Block Group 2, Census Tract 110, Philadelphia ... 101
4    20  Block Group 1, Census Tract 110, Philadelphia ... 101

      census_tract  census_blockgroup
0        10800            1
1        10800            2
2        10900            2
3        11000            2
4        11000            1

[5 rows x 29 columns]
```

```
[ ]: vacant_type.columns
```

```
[ ]: Index(['OBJECTID', 'STATEFP10', 'COUNTYFP10', 'TRACTCE10', 'BLKGRPCE10',
       'GEOID10', 'NAMELSAD10', 'MTFCC10', 'FUNCSTAT10', 'ALAND10', 'AWATER10',
       'INTPTLAT10', 'INTPTLON10', 'Shape_Are', 'Shape_Len', 'geometry',
       'index', 'total', 'for_rent', 'rented_not_occupied', 'for_sale_only',
       'sold_not_occupied', 'seasonal_recreational', 'migrant_workers',
```

```
'other', 'census_info', 'county', 'census_tract', 'census_blockgroup'],
dtype='object')
```

```
[ ]: vacant_type1 = vacant_type.drop(['OBJECTID', 'STATEFP10', 'COUNTYFP10',
→'NAMELSAD10', 'MTFCC10', 'FUNCSTAT10', 'ALAND10', 'AWATER10',
→'INTPTLAT10', 'INTPTLON10','index', 'total', 'for_rent',
→'rented_not_occupied', 'for_sale_only',
→'sold_not_occupied', 'seasonal_recreational', 'migrant_workers',
→'census_info', 'county', 'census_tract', 'census_blockgroup'], axis =1)
vacant_type1.head() # dropping all other vacancy type and only keeping "other"
→vacancy type
```

```
[ ]:   TRACTCE10    BLKGRPCE10      GEOID10      Shape__Are      Shape__Len \
0       10800           1  421010108001  1.742508e+06  8200.327170
1       10800           2  421010108002  1.117026e+06  4364.980144
2       10900           2  421010109002  4.706347e+05  3048.109084
3       11000           2  421010110002  1.172871e+06  5169.004282
4       11000           1  421010110001  1.531076e+06  10476.574129

                                         geometry      other
0  POLYGON ((-75.19851 39.96945, -75.19744 39.969...
1  POLYGON ((-75.19783 39.96571, -75.20006 39.965...
2  POLYGON ((-75.18766 39.96450, -75.18755 39.963...
3  POLYGON ((-75.20984 39.97351, -75.21221 39.973...
4  POLYGON ((-75.19855 39.97330, -75.19854 39.973...
```

```
[ ]: #joining vacant type with occupancy based on tract and block
vacant_occupancy = vacant_type1.merge(acs_occupancy, how='left',
→left_on=["TRACTCE10", "BLKGRPCE10"],
→right_on=["census_tract","census_blockgroup"])
vacant_occupancy = vacant_occupancy.drop(["index", 'census_info', 'county',
→'census_tract', 'census_blockgroup'], axis = 1)
vacant_occupancy['perc_other_vacant'] = vacant_occupancy['other']/
→vacant_occupancy['total']
vacant_occupancy = vacant_occupancy.rename(
    columns = {
        "total": "total_housing",
        "other": "other_vacant"
    }
)
```

```
[ ]: #Percentage of "other" vacancy type of total
sum(vacant_occupancy['other_vacant'])/ sum(vacant_occupancy['total_housing'])
```

```
[ ]: 0.06713258576351835
```

```
[ ]: vacant_occupancy['perc_other_vacant'].describe()
```

```
[ ]: count      1326.000000
      mean       0.073894
      std        0.085544
      min        0.000000
      25%       0.000000
      50%       0.054011
      75%       0.111872
      max        0.489933
      Name: perc_other_vacant, dtype: float64
```

```
[ ]: vacant_occupancy.head()
```

```
[ ]:   TRACTCE10  BLKGRPCE10      GEOID10      Shape__Are      Shape__Len \
0      10800          1  421010108001  1.742508e+06  8200.327170
1      10800          2  421010108002  1.117026e+06  4364.980144
2      10900          2  421010109002  4.706347e+05  3048.109084
3      11000          2  421010110002  1.172871e+06  5169.004282
4      11000          1  421010110001  1.531076e+06  10476.574129

                                              geometry  other_vacant \
0  POLYGON ((-75.19851 39.96945, -75.19744 39.969...           41
1  POLYGON ((-75.19783 39.96571, -75.20006 39.965...           82
2  POLYGON ((-75.18766 39.96450, -75.18755 39.963...            7
3  POLYGON ((-75.20984 39.97351, -75.21221 39.973...           73
4  POLYGON ((-75.19855 39.97330, -75.19854 39.973...           20

      total_housing  occupied  vacant  perc_vacant  perc_other_vacant
0            243       202      41      0.168724      0.168724
1            360       239     121      0.336111      0.227778
2            236       221      15      0.063559      0.029661
3            478       348     130      0.271967      0.152720
4            240       187      53      0.220833      0.083333
```

```
[ ]: #joining vacant, occupancy and total population
vacant_occupancy = vacant_occupancy.merge(acs_pop, how='left',
                                         left_on=["TRACTCE10", "BLKGRPCE10"],
                                         right_on=["census_tract", "census_blockgroup"])
vacant_occupancy = vacant_occupancy.drop(['census_tract', 'census_blockgroup'],
                                         axis=1)
vacant_occupancy.head()
```

```
[ ]:   TRACTCE10  BLKGRPCE10      GEOID10      Shape__Are      Shape__Len \
0      10800          1  421010108001  1.742508e+06  8200.327170
1      10800          2  421010108002  1.117026e+06  4364.980144
2      10900          2  421010109002  4.706347e+05  3048.109084
3      11000          2  421010110002  1.172871e+06  5169.004282
4      11000          1  421010110001  1.531076e+06  10476.574129
```

```

                                geometry  other_vacant \
0  POLYGON ((-75.19851 39.96945, -75.19744 39.969...      41
1  POLYGON ((-75.19783 39.96571, -75.20006 39.965...      82
2  POLYGON ((-75.18766 39.96450, -75.18755 39.963...       7
3  POLYGON ((-75.20984 39.97351, -75.21221 39.973...      73
4  POLYGON ((-75.19855 39.97330, -75.19854 39.973...      20

   total_housing  occupied  vacant  perc_vacant  perc_other_vacant  total_pop
0            243        202      41    0.168724          0.168724      702
1            360        239     121    0.336111          0.227778      605
2            236        221      15    0.063559          0.029661      457
3            478        348     130    0.271967          0.152720      709
4            240        187      53    0.220833          0.083333      631

```

[]: vacant_occupancy['geometry'].crs #currently in terms of lat and longitude

[]: <Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich

[]: vacant_occupancy['area_m'] = vacant_occupancy['geometry'].to_crs(epsg = 3311) #converting area in m2
vacant_occupancy['pop_density_km'] = vacant_occupancy['total_pop']/
((vacant_occupancy['geometry'].to_crs(epsg=3311).area)/
(1000*1000)) #population density in km2
vacant_occupancy.head()

	TRACTCE10	BLKGRPCE10	GEOID10	Shape__Are	Shape__Len	geometry	other_vacant
0	10800	1	421010108001	1.742508e+06	8200.327170	POLYGON ((-75.19851 39.96945, -75.19744 39.969...	41
1	10800	2	421010108002	1.117026e+06	4364.980144	POLYGON ((-75.19783 39.96571, -75.20006 39.965...	82
2	10900	2	421010109002	4.706347e+05	3048.109084	POLYGON ((-75.18766 39.96450, -75.18755 39.963...	7
3	11000	2	421010110002	1.172871e+06	5169.004282		
4	11000	1	421010110001	1.531076e+06	10476.574129		

```

3 POLYGON ((-75.20984 39.97351, -75.21221 39.973...           73
4 POLYGON ((-75.19855 39.97330, -75.19854 39.973...           20

      total_housing   occupied    vacant  perc_vacant  perc_other_vacant  total_pop \
0            243        202       41     0.168724          0.168724        702
1            360        239      121     0.336111          0.227778        605
2            236        221       15     0.063559          0.029661        457
3            478        348      130     0.271967          0.152720        709
4            240        187       53     0.220833          0.083333        631

                           area_m  pop_density_km
0  POLYGON ((3684773.964 1104249.065, 3684870.739...  4336.368995
1  POLYGON ((3685015.052 1103905.194, 3684857.753...  5829.834621
2  POLYGON ((3685850.037 1104181.114, 3685886.064...  10451.930347
3  POLYGON ((3683707.743 1104211.137, 3683540.308...  6506.669922
4  POLYGON ((3684576.459 1104629.062, 3684590.061...  4436.045393

```

[]: vacant_occupancy.describe()

```

[ ]:      TRACTCE10  BLKGRPCE10  Shape_Are  Shape_Len  other_vacant \
count  1336.000000  1336.000000  1.336000e+03  1336.000000  1336.000000
mean   27668.919910  2.598802  2.972072e+06  6869.044968  34.468563
std    87647.462929  1.496302  8.974322e+06  6447.886531  40.548789
min    100.000000  1.000000  1.803535e+05  1794.706669  0.000000
25%   9500.000000  1.000000  8.783481e+05  4071.874360  0.000000
50%   19800.000000  2.000000  1.342269e+06  5223.795807  25.000000
75%   30200.000000  3.000000  2.449106e+06  7538.593623  52.000000
max   989100.000000  8.000000  2.213137e+08  100965.278313  334.000000

      total_housing   occupied    vacant  perc_vacant \
count  1336.000000  1336.000000  1336.000000  1326.000000
mean   513.440120  450.102545  63.337575  0.125781
std    254.862223  232.346504  58.294697  0.103121
min    0.000000  0.000000  0.000000  0.000000
25%   348.000000  303.000000  22.750000  0.049860
50%   460.500000  397.000000  51.000000  0.106026
75%   619.500000  554.250000  92.000000  0.185122
max   2043.000000  1850.000000  460.000000  0.542645

      perc_other_vacant  total_pop  pop_density_km
count  1326.000000  1336.000000  1336.000000
mean   0.073894  1181.942365  8959.780737
std    0.085544  586.264514  5651.084498
min    0.000000  0.000000  0.000000
25%   0.000000  775.250000  4766.378944
50%   0.054011  1073.500000  7923.688573
75%   0.111872  1473.000000  11953.482935

```

```
max          0.489933  4099.000000   44772.034568

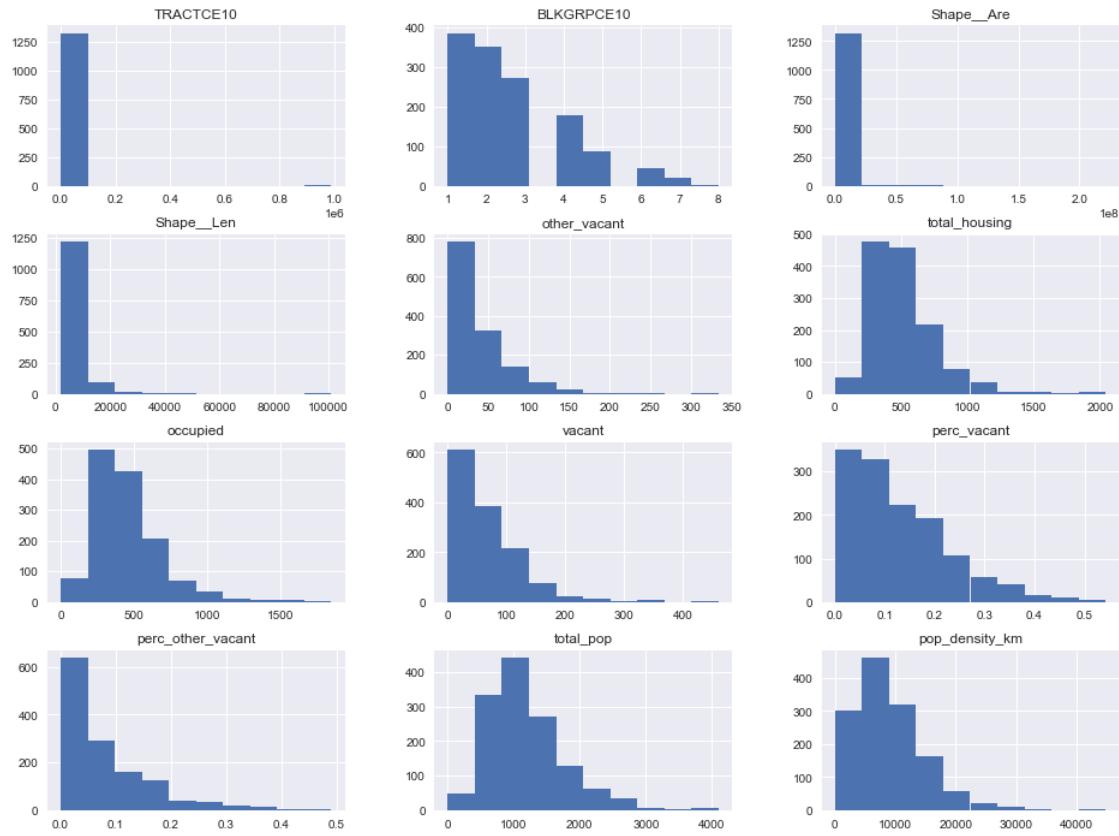
[ ]: vacant_occupancy.shape

[ ]: (1336, 15)

[ ]: fig, ax = plt.subplots(figsize=(16,12))
vacant_occupancy.hist(ax = ax)

/var/folders/6p/wpw9qml57530xkxqkkhprrf40000gn/T/ipykernel_21232/405393172.py:2:
UserWarning: To output multiple subplots, the figure containing the passed axes
is being cleared
vacant_occupancy.hist(ax = ax)

[ ]: array([[<AxesSubplot:title={'center':'TRACTCE10'}>,
   <AxesSubplot:title={'center':'BLKGRPCE10'}>,
   <AxesSubplot:title={'center':'Shape_Are'}>],
  [<AxesSubplot:title={'center':'Shape_Len'}>,
   <AxesSubplot:title={'center':'other_vacant'}>,
   <AxesSubplot:title={'center':'total_housing'}>],
  [<AxesSubplot:title={'center':'occupied'}>,
   <AxesSubplot:title={'center':'vacant'}>,
   <AxesSubplot:title={'center':'perc_vacant'}>],
  [<AxesSubplot:title={'center':'perc_other_vacant'}>,
   <AxesSubplot:title={'center':'total_pop'}>,
   <AxesSubplot:title={'center':'pop_density_km'}>]], dtype=object)
```



```
[ ]: acs = vacant_occupancy.copy()
```

```
[ ]: #vacant_occupancy.to_csv("acs.csv")
```

0.2.12 Other Important ACS datasets

Datasets in this section are uploaded but not explored a lot at the moment. We decided not to use this because of time constraint

ACS: Race

```
[ ]: censusdata.printtable(censusdata.censustable('acs5', 2019, 'B03002'))#race
#censusdata.censustable('acs5', 2019, 'B03002')
```

Variable	Table	Label
Type		
<hr/>		
B03002_001E	HISPANIC OR LATINO ORIGIN BY R	!! Estimate Total:
int		
B03002_002E	HISPANIC OR LATINO ORIGIN BY R	!!! Estimate Total: Not
Hispanic or Latino:	int	

B03002_003E | HISPANIC OR LATINO ORIGIN BY R | !!! !! Estimate Total: Not Hispanic or Latino: White a | int
B03002_004E | HISPANIC OR LATINO ORIGIN BY R | !!! !! Estimate Total: Not Hispanic or Latino: Black o | int
B03002_005E | HISPANIC OR LATINO ORIGIN BY R | !!! !! Estimate Total: Not Hispanic or Latino: America | int
B03002_006E | HISPANIC OR LATINO ORIGIN BY R | !!! !! Estimate Total: Not Hispanic or Latino: Asian a | int
B03002_007E | HISPANIC OR LATINO ORIGIN BY R | !!! !! Estimate Total: Not Hispanic or Latino: Native | int
B03002_008E | HISPANIC OR LATINO ORIGIN BY R | !!! !! Estimate Total: Not Hispanic or Latino: Some ot | int
B03002_009E | HISPANIC OR LATINO ORIGIN BY R | !!! !! Estimate Total: Not Hispanic or Latino: Two or | int
B03002_010E | HISPANIC OR LATINO ORIGIN BY R | !!! !! !! Estimate Total: Not Hispanic or Latino: Two | int
B03002_011E | HISPANIC OR LATINO ORIGIN BY R | !!! !! !! Estimate Total: Not Hispanic or Latino: Two | int
B03002_012E | HISPANIC OR LATINO ORIGIN BY R | !!! !! Estimate Total: Hispanic or Latino: | int
B03002_013E | HISPANIC OR LATINO ORIGIN BY R | !!! !! !! Estimate Total: Hispanic or Latino: White alone | int
B03002_014E | HISPANIC OR LATINO ORIGIN BY R | !!! !! !! Estimate Total: Hispanic or Latino: Black or Af | int
B03002_015E | HISPANIC OR LATINO ORIGIN BY R | !!! !! !! Estimate Total: Hispanic or Latino: American In | int
B03002_016E | HISPANIC OR LATINO ORIGIN BY R | !!! !! !! Estimate Total: Hispanic or Latino: Asian alone | int
B03002_017E | HISPANIC OR LATINO ORIGIN BY R | !!! !! !! Estimate Total: Hispanic or Latino: Native Hawa | int
B03002_018E | HISPANIC OR LATINO ORIGIN BY R | !!! !! !! Estimate Total: Hispanic or Latino: Some other | int
B03002_019E | HISPANIC OR LATINO ORIGIN BY R | !!! !! !! Estimate Total: Hispanic or Latino: Two or more | int
B03002_020E | HISPANIC OR LATINO ORIGIN BY R | !!! !! !! !! Estimate Total: Hispanic or Latino: Two or m | int
B03002_021E | HISPANIC OR LATINO ORIGIN BY R | !!! !! !! !! Estimate Total: Hispanic or Latino: Two or m | int

```
[ ]: acs_race = censusdata.download('acs5', 2019,
    censusdata.censusgeo([('state', '42'),
        ('county', '101'), # philadelphia county
        ('block group', '*')]),
    ['B03002_001E', 'B03002_003E', 'B03002_004E', ↴
     'B03002_006E', 'B03002_012E'])
```

```

acs_race.rename(columns = {'B03002_001E': 'race_total',
                           'B03002_003E': 'race_white',
                           'B03002_004E': 'race_black',
                           'B03002_006E': 'race_asian',
                           'B03002_012E': 'race_hispanic'}, inplace = True)

acs_race.to_csv('data/acs/race.csv')
acs_race.head()

```

[]:

	race_total	race_white	\	
Block Group 1, Census Tract 9807, Philadelphia ...	0	0		
Block Group 3, Census Tract 27.01, Philadelphia...	1955	904		
Block Group 2, Census Tract 337.01, Philadelphi...	976	654		
Block Group 3, Census Tract 337.01, Philadelphi...	3859	1594		
Block Group 2, Census Tract 205, Philadelphia C...	1017	36		
			race_black	
			race_asian	\
Block Group 1, Census Tract 9807, Philadelphia ...	0	0		
Block Group 3, Census Tract 27.01, Philadelphia...	262	502		
Block Group 2, Census Tract 337.01, Philadelphi...	0	99		
Block Group 3, Census Tract 337.01, Philadelphi...	477	305		
Block Group 2, Census Tract 205, Philadelphia C...	796	124		
				race_hispanic
Block Group 1, Census Tract 9807, Philadelphia ...	0			
Block Group 3, Census Tract 27.01, Philadelphia...	251			
Block Group 2, Census Tract 337.01, Philadelphi...	79			
Block Group 3, Census Tract 337.01, Philadelphi...	1301			
Block Group 2, Census Tract 205, Philadelphia C...	61			

ACS: Education

[]:

```
censusdata.printtable(censusdata.censustable('acs5', 2015, 'B15002'))#education
```

Variable	Table	Label
Type		
<hr/>		
B15002_001E	SEX BY EDUCATIONAL ATTAINMENT	!! Estimate Total
int		
B15002_002E	SEX BY EDUCATIONAL ATTAINMENT	!! !! Estimate Total Male
int		
B15002_003E	SEX BY EDUCATIONAL ATTAINMENT	!!! !!! Estimate Total Male No
schooling completed	int	
B15002_004E	SEX BY EDUCATIONAL ATTAINMENT	!!! !!! Estimate Total Male
Nursery to 4th grade	int	
B15002_005E	SEX BY EDUCATIONAL ATTAINMENT	!!! !!! Estimate Total Male 5th

and 6th grade	int	
B15002_006E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	7th
and 8th grade	int	
B15002_007E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	9th
grade	int	
B15002_008E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	
10th grade	int	
B15002_009E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	
11th grade	int	
B15002_010E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	
12th grade, no diploma	int	
B15002_011E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	
High school graduate (inclu	int	
B15002_012E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	
Some college, less than 1 y	int	
B15002_013E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	
Some college, 1 or more yea	int	
B15002_014E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	
Associate's degree	int	
B15002_015E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	
Bachelor's degree	int	
B15002_016E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	
Master's degree	int	
B15002_017E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	
Professional school degree	int	
B15002_018E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Male	
Doctorate degree	int	
B15002_019E SEX BY EDUCATIONAL ATTAINMENT	! ! ! Estimate Total Female	
int		
B15002_020E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Female	
No schooling completed	int	
B15002_021E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Female	
Nursery to 4th grade	int	
B15002_022E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Female	
5th and 6th grade	int	
B15002_023E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Female	
7th and 8th grade	int	
B15002_024E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Female	
9th grade	int	
B15002_025E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Female	
10th grade	int	
B15002_026E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Female	
11th grade	int	
B15002_027E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Female	
12th grade, no diploma	int	
B15002_028E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Female	
High school graduate (inc	int	
B15002_029E SEX BY EDUCATIONAL ATTAINMENT	! ! ! ! ! Estimate Total Female	

```

Some college, less than 1 | int
B15002_030E | SEX BY EDUCATIONAL ATTAINMENT | !!! !! Estimate Total Female
Some college, 1 or more y | int
B15002_031E | SEX BY EDUCATIONAL ATTAINMENT | !!! !! Estimate Total Female
Associate's degree | int
B15002_032E | SEX BY EDUCATIONAL ATTAINMENT | !!! !! Estimate Total Female
Bachelor's degree | int
B15002_033E | SEX BY EDUCATIONAL ATTAINMENT | !!! !! Estimate Total Female
Master's degree | int
B15002_034E | SEX BY EDUCATIONAL ATTAINMENT | !!! !! Estimate Total Female
Professional school degre | int
B15002_035E | SEX BY EDUCATIONAL ATTAINMENT | !!! !! Estimate Total Female
Doctorate degree | int
-----
```

ACS: Household Income

```
[ ]: censusdata.printtable(censusdata.censustable('acs5', 2015, 'B19001'))#household
    ↳income
```

Variable	Table	Label
	Type	
B19001_001E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total
	int	
B19001_002E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total Less than
	\$10,000	int
B19001_003E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total \$10,000 to
	\$14,999	int
B19001_004E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total \$15,000 to
	\$19,999	int
B19001_005E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total \$20,000 to
	\$24,999	int
B19001_006E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total \$25,000 to
	\$29,999	int
B19001_007E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total \$30,000 to
	\$34,999	int
B19001_008E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total \$35,000 to
	\$39,999	int
B19001_009E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total \$40,000 to
	\$44,999	int
B19001_010E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total \$45,000 to
	\$49,999	int
B19001_011E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total \$50,000 to
	\$59,999	int
B19001_012E	HOUSEHOLD INCOME IN THE PAST 1	!!! Estimate Total \$60,000 to

```

$74,999 | int
B19001_013E | HOUSEHOLD INCOME IN THE PAST 1 | !!! Estimate Total $75,000 to
$99,999 | int
B19001_014E | HOUSEHOLD INCOME IN THE PAST 1 | !!! Estimate Total $100,000 to
$124,999 | int
B19001_015E | HOUSEHOLD INCOME IN THE PAST 1 | !!! Estimate Total $125,000 to
$149,999 | int
B19001_016E | HOUSEHOLD INCOME IN THE PAST 1 | !!! Estimate Total $150,000 to
$199,999 | int
B19001_017E | HOUSEHOLD INCOME IN THE PAST 1 | !!! Estimate Total $200,000 or
more | int
-----
-----
```

ACS: Age and Sex

```
[ ]: censusdata.printtable(censusdata.censustable('acs5', 2019, 'B01001'))#age and
    ↪sex
```

Variable	Table	Label
Type		
B01001_001E	SEX BY AGE	!!! Estimate Total:
int		
B01001_002E	SEX BY AGE	!!! Estimate Total: Male:
int		
B01001_003E	SEX BY AGE	!!! !!! Estimate Total: Male:
Under 5 years	int	
B01001_004E	SEX BY AGE	!!! !!! Estimate Total: Male: 5
to 9 years	int	
B01001_005E	SEX BY AGE	!!! !!! Estimate Total: Male:
10 to 14 years	int	
B01001_006E	SEX BY AGE	!!! !!! Estimate Total: Male:
15 to 17 years	int	
B01001_007E	SEX BY AGE	!!! !!! Estimate Total: Male:
18 and 19 years	int	
B01001_008E	SEX BY AGE	!!! !!! Estimate Total: Male:
20 years	int	
B01001_009E	SEX BY AGE	!!! !!! Estimate Total: Male:
21 years	int	
B01001_010E	SEX BY AGE	!!! !!! Estimate Total: Male:
22 to 24 years	int	
B01001_011E	SEX BY AGE	!!! !!! Estimate Total: Male:
25 to 29 years	int	
B01001_012E	SEX BY AGE	!!! !!! Estimate Total: Male:
30 to 34 years	int	
B01001_013E	SEX BY AGE	!!! !!! Estimate Total: Male:

35 to 39 years	int	
B01001_014E SEX BY AGE		!!! !! Estimate Total: Male:
40 to 44 years	int	
B01001_015E SEX BY AGE		!!! !! Estimate Total: Male:
45 to 49 years	int	
B01001_016E SEX BY AGE		!!! !! Estimate Total: Male:
50 to 54 years	int	
B01001_017E SEX BY AGE		!!! !! Estimate Total: Male:
55 to 59 years	int	
B01001_018E SEX BY AGE		!!! !! Estimate Total: Male:
60 and 61 years	int	
B01001_019E SEX BY AGE		!!! !! Estimate Total: Male:
62 to 64 years	int	
B01001_020E SEX BY AGE		!!! !! Estimate Total: Male:
65 and 66 years	int	
B01001_021E SEX BY AGE		!!! !! Estimate Total: Male:
67 to 69 years	int	
B01001_022E SEX BY AGE		!!! !! Estimate Total: Male:
70 to 74 years	int	
B01001_023E SEX BY AGE		!!! !! Estimate Total: Male:
75 to 79 years	int	
B01001_024E SEX BY AGE		!!! !! Estimate Total: Male:
80 to 84 years	int	
B01001_025E SEX BY AGE		!!! !! Estimate Total: Male:
85 years and over	int	
B01001_026E SEX BY AGE		!!! !! Estimate Total: Female:
int		
B01001_027E SEX BY AGE		!!! !! Estimate Total: Female:
Under 5 years	int	
B01001_028E SEX BY AGE		!!! !! Estimate Total: Female:
5 to 9 years	int	
B01001_029E SEX BY AGE		!!! !! Estimate Total: Female:
10 to 14 years	int	
B01001_030E SEX BY AGE		!!! !! Estimate Total: Female:
15 to 17 years	int	
B01001_031E SEX BY AGE		!!! !! Estimate Total: Female:
18 and 19 years	int	
B01001_032E SEX BY AGE		!!! !! Estimate Total: Female:
20 years	int	
B01001_033E SEX BY AGE		!!! !! Estimate Total: Female:
21 years	int	
B01001_034E SEX BY AGE		!!! !! Estimate Total: Female:
22 to 24 years	int	
B01001_035E SEX BY AGE		!!! !! Estimate Total: Female:
25 to 29 years	int	
B01001_036E SEX BY AGE		!!! !! Estimate Total: Female:
30 to 34 years	int	
B01001_037E SEX BY AGE		!!! !! Estimate Total: Female:

35 to 39 years	int	
B01001_038E SEX BY AGE		!!! !! Estimate Total: Female:
40 to 44 years	int	
B01001_039E SEX BY AGE		!!! !! Estimate Total: Female:
45 to 49 years	int	
B01001_040E SEX BY AGE		!!! !! Estimate Total: Female:
50 to 54 years	int	
B01001_041E SEX BY AGE		!!! !! Estimate Total: Female:
55 to 59 years	int	
B01001_042E SEX BY AGE		!!! !! Estimate Total: Female:
60 and 61 years	int	
B01001_043E SEX BY AGE		!!! !! Estimate Total: Female:
62 to 64 years	int	
B01001_044E SEX BY AGE		!!! !! Estimate Total: Female:
65 and 66 years	int	
B01001_045E SEX BY AGE		!!! !! Estimate Total: Female:
67 to 69 years	int	
B01001_046E SEX BY AGE		!!! !! Estimate Total: Female:
70 to 74 years	int	
B01001_047E SEX BY AGE		!!! !! Estimate Total: Female:
75 to 79 years	int	
B01001_048E SEX BY AGE		!!! !! Estimate Total: Female:
80 to 84 years	int	
B01001_049E SEX BY AGE		!!! !! Estimate Total: Female:
85 years and over	int	

ACS: Poverty Status

```
[ ]: censusdata.printtable(censusdata.censustable('acs5', 2015, 'B17001'))#poverty
      ↪status
```

Variable	Table	Label
	Type	
B17001_001E POVERTY STATUS IN THE PAST 12	!!	Estimate Total
int		
B17001_002E POVERTY STATUS IN THE PAST 12	!!!	Estimate Total Income in
the past 12 months below	int	
B17001_003E POVERTY STATUS IN THE PAST 12	!!! !!	Estimate Total Income
in the past 12 months bel	int	
B17001_004E POVERTY STATUS IN THE PAST 12	!!! !! !!	Estimate Total
Income in the past 12 months	int	
B17001_005E POVERTY STATUS IN THE PAST 12	!!! !! !! !!	Estimate Total
Income in the past 12 months	int	
B17001_006E POVERTY STATUS IN THE PAST 12	!!! !! !! !! !!	Estimate Total

```

Income in the past 12 months | int
B17001_055E | POVERTY STATUS IN THE PAST 12 | !!!!!! Estimate Total
Income in the past 12 months | int
B17001_056E | POVERTY STATUS IN THE PAST 12 | !!!!!! Estimate Total
Income in the past 12 months | int
B17001_057E | POVERTY STATUS IN THE PAST 12 | !!!!!! Estimate Total
Income in the past 12 months | int
B17001_058E | POVERTY STATUS IN THE PAST 12 | !!!!!! Estimate Total
Income in the past 12 months | int
B17001_059E | POVERTY STATUS IN THE PAST 12 | !!!!!! Estimate Total
Income in the past 12 months | int
-----
-----
```

0.2.13 City of Philadelphia: Property Tax Delinquency

<https://metadata.phila.gov/#home/datasetdetails/57d9643afab162fe2708224e/representationdetails/57d9643cfab>

An account is delinquent when Real Estate Tax is still unpaid on January 1 the following year the tax was due

Date Range 1972 - 2018, Updated Monthly

Tax Delinquency(1972 - 2018, but it has data from 2021). The major takeaway are: - Principal due data is skewed - Median for Principal due is around \$2000 - Actionable means that the city is actively working to collect these accounts, non-actional means that the city can't do anything further or they are barred from collection. - Is actionable and not actionable is not that different. - Accounts that are in payment agreement, bankruptcy, or overdue but not yet delinquent are considered "not actionable". Payment agreement is one of the way the city collect debts. - Sheriff sale, Sequestration is actionable. A sheriff's sale is a public auction where mortgage lenders, banks, tax collectors, and other litigants can collect money lost on property. Sequestration is the taking of someones property, voluntarily (by deposit) or involuntarily (by seizure), by court officers or into the possession of a third party, awaiting the outcome of a trial in which ownership of that property is at issue .If the delinquent property is a rental property, the City can take over the rent collection and apply those rental payments to the delinquent Real Estate Tax bill. - Most of the principal due is owned for 1-4 years, then there is 25 years which has the highest principal due - When you do median principal dues is for 18, 23, 22 and 27 years. Principal value is skewed so median is a better measure - Most of the delinquent properties are between owned for 1-6 year , then there is 25 years owned of delinquent properties - Most of the delinquent properties are residential, not commercial. - 89% of residential places are delinquent - Most of the places were assessed in 2021 - Principal due is the most for houses and vacant land. However, the median and mean principal due is not high for house and vacant lots. Utility buildings have the highest median principal due

```
[ ]: tax = pd.read_csv('data/city/real_estate_tax_delinquencies.csv') #uploading
→dataset
```

```
[ ]: tax.head()
```

```
[ ]: objectid    opa_number    street_address    zip_code    zip_4    unit_type    \
0    2556493    493169300.0    6045 N CAMAC ST    19141.0    3227.0    NaN
1    2556494    493179100.0    5620 N CAMAC ST    19141.0    4106.0    NaN
2    2556495    493180700.0    5714 N CAMAC ST    19141.0    4108.0    NaN
3    2556496    493183600.0    5812 N CAMAC ST    19141.0    4123.0    NaN
4    2556497    223166200.0    420 GLEN ECHO RD    19119.0    2914.0    NaN

unit_num    owner    co_owner    principal_due    ...    \
0    NaN    WILLIAMS JACQUELINE    WILLIAMS JACQUELINE    12200.18    ...
1    NaN    RAY MATTIE E    RAY MATTIE E    -0.05    ...
2    NaN    TOMLIN PAULA    TOMLIN PAULA    895.87    ...
3    NaN    BATTS PRINCETON B    BATTS PRINCETON B    4536.94    ...
4    NaN    WHITE CLARENCE    WHITE CLARENCE    4224.60    ...

oldest_bankrupt_year    principal_sum_bankrupt_years    \
0    NaN    NaN
1    NaN    NaN
2    NaN    NaN
3    NaN    NaN
4    NaN    NaN

total_amount_bankrupt_years    sheriff_sale    liens_sold_1990s    \
0    NaN    N    False
1    NaN    N    False
2    NaN    N    False
3    NaN    N    False
4    NaN    N    False

liens_sold_2015    assessment_under_appeal    year_month    lat    lng
0    N    False    202111    -75.140099    40.045081
1    N    False    202111    -75.141930    40.039007
2    N    False    202111    -75.141727    40.039940
3    N    False    202111    -75.141395    40.041404
4    N    False    202111    -75.195309    40.051563
```

[5 rows x 55 columns]

```
[ ]: tax.columns #columns in dataset
```

```
[ ]: Index(['objectid', 'opa_number', 'street_address', 'zip_code', 'zip_4',
       'unit_type', 'unit_num', 'owner', 'co_owner', 'principal_due',
       'penalty_due', 'interest_due', 'other_charges_due', 'total_due',
       'is_actionable', 'payment_agreement', 'num_years_owed',
       'most_recent_year_owed', 'oldest_year_owed', 'most_recent_payment_date',
       'year_of_last_assessment', 'total_assessment', 'taxable_assessment',
       'mailing_address', 'mailing_city', 'mailing_state', 'mailing_zip',
       'return_mail', 'building_code', 'detail_building_description',
```

```
'general_building_description', 'building_category',
'coll_agency_num_years', 'coll_agency_most_recent_year',
'coll_agency_oldest_year', 'coll_agency_principal_owed',
'coll_agency_total_owed', 'exempt_abatement_assessment',
'homestead_value', 'net_tax_value_after_homestead', 'agreement_agency',
'sequestration_enforcement', 'bankruptcy', 'years_in_bankruptcy',
'most_recent_bankrupt_year', 'oldest_bankrupt_year',
'principal_sum_bankrupt_years', 'total_amount_bankrupt_years',
'sheriff_sale', 'liens_sold_1990s', 'liens_sold_2015',
'assessment_under_appeal', 'year_month', 'lat', 'lng'],
dtype='object')
```

[]: tax.dtypes #*type of data included*

objectid	int64
opa_number	float64
street_address	object
zip_code	float64
zip_4	float64
unit_type	object
unit_num	object
owner	object
co_owner	object
principal_due	float64
penalty_due	float64
interest_due	float64
other_charges_due	float64
total_due	float64
is_actionable	bool
payment_agreement	bool
num_years_owed	int64
most_recent_year_owed	int64
oldest_year_owed	int64
most_recent_payment_date	object
year_of_last_assessment	float64
total_assessment	float64
taxable_assessment	float64
mailing_address	object
mailing_city	object
mailing_state	object
mailing_zip	float64
return_mail	object
building_code	object
detail_building_description	object
general_building_description	object
building_category	object
coll_agency_num_years	int64

```

coll_agency_most_recent_year      float64
coll_agency_oldest_year          float64
coll_agency_principal_owed       float64
coll_agency_total_owed           float64
exempt_abatement_assessment     float64
homestead_value                  float64
net_tax_value_after_homestead   float64
agreement_agency                object
sequestration_enforcement        bool
bankruptcy                       bool
years_in_bankruptcy              float64
most_recent_bankrupt_year        float64
oldest_bankrupt_year             float64
principal_sum_bankrupt_years    float64
total_amount_bankrupt_years     float64
sheriff_sale                     object
liens_sold_1990s                 bool
liens_sold_2015                  object
assessment_under_appeal          bool
year_month                        int64
lat                               float64
lng                               float64
dtype: object

```

[]: tax.T# *transposing the dataset*

	0	1	\
objectid	2556493	2556494	
opa_number	493169300.0	493179100.0	
street_address	6045 N CAMAC ST	5620 N CAMAC ST	
zip_code	19141.0	19141.0	
zip_4	3227.0	4106.0	
unit_type	NaN	NaN	
unit_num	NaN	NaN	
owner	WILLIAMS JACQUELINE	RAY MATTIE E	
co_owner	WILLIAMS JACQUELINE	RAY MATTIE E	
principal_due	12200.18	-0.05	
penalty_due	1110.17	0.0	
interest_due	14261.97	41.05	
other_charges_due	3098.66	0.0	
total_due	30670.98	41.0	
is_actionable	False	False	
payment_agreement	True	True	
num_years_owed	23	1	
most_recent_year_owed	2021	2021	
oldest_year_owed	1994	2021	
most_recent_payment_date	2021-09-16 00:00:00	2021-09-17 00:00:00	

year_of_last_assessment	2021.0	2021.0
total_assessment	111400.0	111200.0
taxable_assessment	111400.0	111200.0
mailing_address	NaN	NaN
mailing_city	NaN	NaN
mailing_state	NaN	NaN
mailing_zip	NaN	NaN
return_mail	True	NaN
building_code	R30	R30
detail_building_description	ROW B/GAR 2STY MASON	ROW B/GAR 2STY MASON
general_building_description	house	house
building_category	residential	residential
coll_agency_num_years	0	0
coll_agency_most_recent_year	NaN	NaN
coll_agency_oldest_year	NaN	NaN
coll_agency_principal_owed	0.0	0.0
coll_agency_total_owed	0.0	0.0
exempt_abatement_assessment	0.0	0.0
homestead_value	629.91	629.91
net_tax_value_after_homestead	929.47	926.67
agreement_agency	TIPS	TIPS
sequestration_enforcement	False	False
bankruptcy	False	False
years_in_bankruptcy	NaN	NaN
most_recent_bankrupt_year	NaN	NaN
oldest_bankrupt_year	NaN	NaN
principal_sum_bankrupt_years	NaN	NaN
total_amount_bankrupt_years	NaN	NaN
sheriff_sale	N	N
liens_sold_1990s	False	False
liens_sold_2015	N	N
assessment_under_appeal	False	False
year_month	202111	202111
lat	-75.140099	-75.14193
lng	40.045081	40.039007
	2	3 \
objectid	2556495	2556496
opa_number	493180700.0	493183600.0
street_address	5714 N CAMAC ST	5812 N CAMAC ST
zip_code	19141.0	19141.0
zip_4	4108.0	4123.0
unit_type	NaN	NaN
unit_num	NaN	NaN
owner	TOMLIN PAULA	BATTS PRINCETON B
co_owner	TOMLIN PAULA	BATTS PRINCETON B
principal_due	895.87	4536.94

penalty_due		0.0	283.14
interest_due		120.94	1161.05
other_charges_due		0.0	710.57
total_due		1016.81	6691.7
is_actionable		False	False
payment_agreement		False	True
num_years_owed		1	5
most_recent_year_owed		2021	2021
oldest_year_owed		2021	2017
most_recent_payment_date	2020-02-22 00:00:00	2021-08-26 00:00:00	
year_of_last_assessment		2021.0	2021.0
total_assessment		109000.0	110600.0
taxable_assessment		109000.0	110600.0
mailing_address		NaN	NaN
mailing_city		NaN	NaN
mailing_state		NaN	NaN
mailing_zip		NaN	NaN
return_mail		NaN	True
building_code		R30	R30
detail_building_description	ROW B/GAR 2STY MASON	ROW B/GAR 2STY MASON	
general_building_description	house	house	
building_category	residential	residential	
coll_agency_num_years	1	0	
coll_agency_most_recent_year	2021.0	NaN	
coll_agency_oldest_year	2021.0	NaN	
coll_agency_principal_owed	895.87	0.0	
coll_agency_total_owed	1016.81	0.0	
exempt_abatement_assessment	0.0	0.0	
homestead_value	629.91	629.91	
net_tax_value_after_homestead	895.87	918.27	
agreement_agency	NaN	TIPS	
sequestration_enforcement	False	False	
bankruptcy	False	False	
years_in_bankruptcy	NaN	NaN	
most_recent_bankrupt_year	NaN	NaN	
oldest_bankrupt_year	NaN	NaN	
principal_sum_bankrupt_years	NaN	NaN	
total_amount_bankrupt_years	NaN	NaN	
sheriff_sale	N	N	
liens_sold_1990s	False	False	
liens_sold_2015	N	N	
assessment_under_appeal	False	False	
year_month	202111	202111	
lat	-75.141727	-75.141395	
lng	40.03994	40.041404	

objectid	2556497
opa_number	223166200.0
street_address	420 GLEN ECHO RD
zip_code	19119.0
zip_4	2914.0
unit_type	NaN
unit_num	NaN
owner	WHITE CLARENCE
co_owner	WHITE CLARENCE
principal_due	4224.6
penalty_due	0.0
interest_due	570.32
other_charges_due	0.0
total_due	4794.92
is_actionable	False
payment_agreement	False
num_years_owed	1
most_recent_year_owed	2021
oldest_year_owed	2021
most_recent_payment_date	2020-12-22 00:00:00
year_of_last_assessment	2021.0
total_assessment	346800.0
taxable_assessment	346800.0
mailing_address	NaN
mailing_city	NaN
mailing_state	NaN
mailing_zip	NaN
return_mail	NaN
building_code	D30
detail_building_description	DET W/B GAR 2 STY MA
general_building_description	house
building_category	residential
coll_agency_num_years	1
coll_agency_most_recent_year	2021.0
coll_agency_oldest_year	2021.0
coll_agency_principal_owed	4224.6
coll_agency_total_owed	4794.92
exempt_abatement_assessment	0.0
homestead_value	629.91
net_tax_value_after_homestead	4224.6
agreement_agency	NaN
sequestration_enforcement	False
bankruptcy	False
years_in_bankruptcy	NaN
most_recent_bankrupt_year	NaN
oldest_bankrupt_year	NaN
principal_sum_bankrupt_years	NaN

total_amount_bankrupt_years	NaN
sheriff_sale	N
liens_sold_1990s	False
liens_sold_2015	N
assessment_under_appeal	False
year_month	202111
lat	-75.195309
lng	40.051563
	5 \
objectid	2556498
opa_number	882929915.0
street_address	2312 DUNCAN ST
zip_code	19124.0
zip_4	4110.0
unit_type	NaN
unit_num	NaN
owner	WHITE DIAMOND ATHLETIC ASSN
co_owner	WHITE DIAMOND ATHLETIC ASSN
principal_due	1068.0
penalty_due	131.99
interest_due	286.92
other_charges_due	445.91
total_due	1932.82
is_actionable	True
payment_agreement	False
num_years_owed	3
most_recent_year_owed	2021
oldest_year_owed	2018
most_recent_payment_date	2021-09-30 00:00:00
year_of_last_assessment	2021.0
total_assessment	62900.0
taxable_assessment	62900.0
mailing_address	NaN
mailing_city	NaN
mailing_state	NaN
mailing_zip	NaN
return_mail	NaN
building_code	JE0
detail_building_description	AMUSE CLUB PRIV MASO
general_building_description	theater_stadium_other amuse
building_category	commercial
coll_agency_num_years	3
coll_agency_most_recent_year	2019.0
coll_agency_oldest_year	2018.0
coll_agency_principal_owed	1068.0
coll_agency_total_owed	1893.2

exempt_abatement_assessment	0.0
homestead_value	0.0
net_tax_value_after_homestead	880.47
agreement_agency	NaN
sequestration_enforcement	False
bankruptcy	False
years_in_bankruptcy	NaN
most_recent_bankrupt_year	NaN
oldest_bankrupt_year	NaN
principal_sum_bankrupt_years	NaN
total_amount_bankrupt_years	NaN
sheriff_sale	N
liens_sold_1990s	False
liens_sold_2015	N
assessment_under_appeal	False
year_month	202111
lat	-75.080997
lng	40.006557

	6	7	\
objectid	2556499	2556500	
opa_number	231024100.0	231042300.0	
street_address	4321 MELROSE ST	4540 MILNOR ST	
zip_code	19124.0	19124.0	
zip_4	4100.0	4120.0	
unit_type	NaN	NaN	
unit_num	NaN	NaN	
owner	HOPWOOD WILLIAM D	KARAS HELEN	
co_owner	HOPWOOD WILLIAM D	KARAS HELEN	
principal_due	1493.6	21.92	
penalty_due	102.78	1.53	
interest_due	503.95	1.97	
other_charges_due	741.76	4.58	
total_due	2842.09	30.0	
is_actionable	True	True	
payment_agreement	False	False	
num_years_owed	5	1	
most_recent_year_owed	2021	2020	
oldest_year_owed	2016	2020	
most_recent_payment_date	2021-09-21 00:00:00	2021-01-05 00:00:00	
year_of_last_assessment	2021.0	2021.0	
total_assessment	21800.0	104400.0	
taxable_assessment	21800.0	104400.0	
mailing_address	NaN	NaN	
mailing_city	NaN	NaN	
mailing_state	NaN	NaN	
mailing_zip	NaN	NaN	

return_mail		NaN	NaN
building_code		SR	H31
detail_building_description	VAC LAND RES < ACRE	SEMI/DET 2 STY MAS.+	
general_building_description	vacantLand	house	
building_category	residential	residential	
coll_agency_num_years	4	1	
coll_agency_most_recent_year	2019.0	2020.0	
coll_agency_oldest_year	2016.0	2020.0	
coll_agency_principal_owed	1188.44	21.92	
coll_agency_total_owed	2495.73	30.0	
exempt_abatement_assessment	0.0	0.0	
homestead_value	0.0	0.0	
net_tax_value_after_homestead	305.16	1461.39	
agreement_agency	NaN	NaN	
sequestration_enforcement	False	False	
bankruptcy	False	False	
years_in_bankruptcy	NaN	NaN	
most_recent_bankrupt_year	NaN	NaN	
oldest_bankrupt_year	NaN	NaN	
principal_sum_bankrupt_years	NaN	NaN	
total_amount_bankrupt_years	NaN	NaN	
sheriff_sale	N	N	
liens_sold_1990s	False	False	
liens_sold_2015	N	N	
assessment_under_appeal	False	False	
year_month	202111	202111	
lat	-75.08183	-75.079183	
lng	40.006032	40.005409	

objectid		8	\
opa_number		2556501	
street_address		871569460.0	
zip_code		4920 LANCASTER AVE	
zip_4		19131.0	
unit_type		4519.0	
unit_num		NaN	
owner	MICHAEL EARL DAVIS JR IRREVOCA		
co_owner	MICHAEL EARL DAVIS JR IRREVOCABLE TRUST		
principal_due	1054.91		
penalty_due	45.88		
interest_due	202.88		
other_charges_due	181.5		
total_due	1485.17		
is_actionable	False		
payment_agreement	True		
num_years_owed	2		

most_recent_year_owed		2021
oldest_year_owed		2019
most_recent_payment_date		2021-09-13 00:00:00
year_of_last_assessment		2021.0
total_assessment		82100.0
taxable_assessment		82100.0
mailing_address	3708 SPRING GARDEN ST	
mailing_city	PHILADELPHIA	
mailing_state	PA	
mailing_zip	19104.0	
return_mail	Nan	
building_code	S30	
detail_building_description	ROW W-OFF/STR 2STY M	
general_building_description	mixedUsage	
building_category	commercial	
coll_agency_num_years	0	
coll_agency_most_recent_year	Nan	
coll_agency_oldest_year	Nan	
coll_agency_principal_owed	0.0	
coll_agency_total_owed	0.0	
exempt_abatement_assessment	0.0	
homestead_value	0.0	
net_tax_value_after_homestead	1149.24	
agreement_agency	TIPS	
sequestration_enforcement	False	
bankruptcy	False	
years_in_bankruptcy	Nan	
most_recent_bankrupt_year	Nan	
oldest_bankrupt_year	Nan	
principal_sum_bankrupt_years	Nan	
total_amount_bankrupt_years	Nan	
sheriff_sale	N	
liens_sold_1990s	False	
liens_sold_2015	N	
assessment_under_appeal	False	
year_month	202111	
lat	-75.220452	
lng	39.973931	
objectid	9	72708
opa_number	2556502	...
street_address	442203100.0	...
zip_code	923 N FALCON ST	...
zip_4	19131.0	...
unit_type	5120.0	...
unit_num	NaN	...
	NaN	...

owner	SMITH FRANK	...	WHITING WARNER H
co_owner	SMITH FRANK	...	WHITING WARNER H
principal_due	2009.47	...	1758.15
penalty_due	297.5	...	0.0
interest_due	2647.13	...	237.35
other_charges_due	767.26	...	0.0
total_due	5721.36	...	1995.5
is_actionable	False	...	False
payment_agreement	True	...	False
num_years_owed	12	...	1
most_recent_year_owed	2021	...	2021
oldest_year_owed	2007	...	2021
most_recent_payment_date	2021-09-14 00:00:00	...	2021-08-11 00:00:00
year_of_last_assessment	2021.0	...	2021.0
total_assessment	50700.0	...	125600.0
taxable_assessment	50700.0	...	125600.0
mailing_address	NaN	...	NaN
mailing_city	NaN	...	NaN
mailing_state	NaN	...	NaN
mailing_zip	NaN	...	NaN
return_mail	NaN	...	NaN
building_code	030	...	U30
detail_building_description	ROW 2 STY MASONRY	...	ROW CONV/APT 2STY MA
general_building_description	house	...	apartmentSmall
building_category	residential	...	residential
coll_agency_num_years	0	...	0
coll_agency_most_recent_year	NaN	...	NaN
coll_agency_oldest_year	NaN	...	NaN
coll_agency_principal_owed	0.0	...	0.0
coll_agency_total_owed	0.0	...	0.0
exempt_abatement_assessment	0.0	...	0.0
homestead_value	629.91	...	0.0
net_tax_value_after_homestead	79.79	...	1758.15
agreement_agency	TIPS	...	NaN
sequestration_enforcement	False	...	False
bankruptcy	False	...	False
years_in_bankruptcy	NaN	...	NaN
most_recent_bankrupt_year	NaN	...	NaN
oldest_bankrupt_year	NaN	...	NaN
principal_sum_bankrupt_years	NaN	...	NaN
total_amount_bankrupt_years	NaN	...	NaN
sheriff_sale	N	...	N
liens_sold_1990s	False	...	False
liens_sold_2015	N	...	N
assessment_under_appeal	False	...	False
year_month	202111	...	202111
lat	-75.217579	...	-75.228101

lng	39.969481	...	39.947988
objectid	72709	72710	\
opa_number	2628137	2628138	
street_address	463006800.0	463008600.0	
zip_code	5638 LARCHWOOD AVE	5720 LARCHWOOD AVE	
zip_4	19143.0	19143.0	
unit_type	Nan	Nan	
unit_num	Nan	Nan	
owner	SHEFFIELD ANITA	WILLIAMS DORIS	
co_owner	SHEFFIELD ANITA	WILLIAMS DORIS	
principal_due	1377.13	99.69	
penalty_due	111.59	0.0	
interest_due	308.31	55.35	
other_charges_due	397.64	0.0	
total_due	2194.67	155.04	
is_actionable	True	False	
payment_agreement	False	True	
num_years_owed	4	1	
most_recent_year_owed	2021	2021	
oldest_year_owed	2018	2021	
most_recent_payment_date	2021-09-23 00:00:00	2021-09-10 00:00:00	
year_of_last_assessment	2021.0	2021.0	
total_assessment	82800.0	85500.0	
taxable_assessment	82800.0	85500.0	
mailing_address	Nan	Nan	
mailing_city	Nan	Nan	
mailing_state	Nan	Nan	
mailing_zip	Nan	Nan	
return_mail	Nan	Nan	
building_code	030	030	
detail_building_description	ROW 2 STY MASONRY	ROW 2 STY MASONRY	
general_building_description	house	house	
building_category	residential	residential	
coll_agency_num_years	3	0	
coll_agency_most_recent_year	2020.0	Nan	
coll_agency_oldest_year	2018.0	Nan	
coll_agency_principal_owed	848.01	0.0	
coll_agency_total_owed	1594.12	0.0	
exempt_abatement_assessment	0.0	0.0	
homestead_value	629.91	0.0	
net_tax_value_after_homestead	529.12	1196.83	
agreement_agency	Nan	TIPS	
sequestration_enforcement	False	False	
bankruptcy	False	False	
years_in_bankruptcy	Nan	Nan	

most_recent_bankrupt_year	NaN	NaN
oldest_bankrupt_year	NaN	NaN
principal_sum_bankrupt_years	NaN	NaN
total_amount_bankrupt_years	NaN	NaN
sheriff_sale	N	N
liens_sold_1990s	False	False
liens_sold_2015	N	N
assessment_under_appeal	False	False
year_month	202111	202111
lat	-75.23575	-75.237316
lng	39.95296	39.95315
	72711	72712 \
objectid	2628139	2628140
opa_number	463012500.0	406265800.0
street_address	5541 HAZEL AVE	6434 GARMAN ST
zip_code	19143.0	19142.0
zip_4	1905.0	3023.0
unit_type	NaN	NaN
unit_num	NaN	NaN
owner	PHILSON JANICE	SMITH DAVID A
co_owner	PHILSON JANICE	SMITH DAVID A
principal_due	5869.57	4740.34
penalty_due	753.48	677.16
interest_due	10820.57	4218.91
other_charges_due	1408.16	1886.04
total_due	18851.78	11522.45
is_actionable	False	True
payment_agreement	True	False
num_years_owed	13	12
most_recent_year_owed	2021	2021
oldest_year_owed	2002	2009
most_recent_payment_date	2021-09-10 00:00:00	2020-02-18 00:00:00
year_of_last_assessment	2021.0	2021.0
total_assessment	48100.0	78200.0
taxable_assessment	48100.0	78200.0
mailing_address	NaN	649 HEMLOCK CT
mailing_city	NaN	BENSALEM
mailing_state	NaN	PA
mailing_zip	NaN	19020.0
return_mail	NaN	NaN
building_code	O30	R30
detail_building_description	ROW 2 STY MASONRY	ROW B/GAR 2STY MASON
general_building_description	house	house
building_category	residential	residential
coll_agency_num_years	0	11
coll_agency_most_recent_year	NaN	2020.0

coll_agency_oldest_year	NaN	2009.0
coll_agency_principal_owed	0.0	3645.7
coll_agency_total_owed	0.0	10280.03
exempt_abatement_assessment	0.0	0.0
homestead_value	629.91	0.0
net_tax_value_after_homestead	43.39	1094.64
agreement_agency	TIPS	NaN
sequestration_enforcement	False	False
bankruptcy	False	False
years_in_bankruptcy	NaN	NaN
most_recent_bankrupt_year	NaN	NaN
oldest_bankrupt_year	NaN	NaN
principal_sum_bankrupt_years	NaN	NaN
total_amount_bankrupt_years	NaN	NaN
sheriff_sale	N	N
liens_sold_1990s	False	False
liens_sold_2015	N	N
assessment_under_appeal	False	False
year_month	202111	202111
lat	-75.233814	-75.227927
lng	39.952345	39.920792
	72713	72714 \
objectid	2628141	2628142
opa_number	406297800.0	406299300.0
street_address	6918 DICKS AVE	7009 W PASSYUNK AVE
zip_code	19142.0	19142.0
zip_4	2517.0	1713.0
unit_type	NaN	NaN
unit_num	NaN	NaN
owner	CAO SON THANH	MCMICHAEL RICHARD
co_owner	CAO SON THANH	MCMICHAEL RICHARD
principal_due	71.34	1542.58
penalty_due	0.0	0.0
interest_due	22.48	208.25
other_charges_due	0.0	0.0
total_due	93.82	1750.83
is_actionable	False	False
payment_agreement	True	False
num_years_owed	1	1
most_recent_year_owed	2021	2021
oldest_year_owed	2021	2021
most_recent_payment_date	2021-09-28 00:00:00	2020-06-16 00:00:00
year_of_last_assessment	2021.0	2021.0
total_assessment	75600.0	110200.0
taxable_assessment	75600.0	110200.0
mailing_address	NaN	NaN

mailing_city		NaN	NaN
mailing_state		NaN	NaN
mailing_zip		NaN	NaN
return_mail		NaN	NaN
building_code		R30	030
detail_building_description	ROW B/GAR 2STY MASON	ROW 2 STY MASONRY	
general_building_description	house	house	
building_category	residential	residential	
coll_agency_num_years	0	1	
coll_agency_most_recent_year	NaN	2021.0	
coll_agency_oldest_year	NaN	2021.0	
coll_agency_principal_owed	0.0	1542.58	
coll_agency_total_owed	0.0	1750.83	
exempt_abatement_assessment	0.0	0.0	
homestead_value	629.91	0.0	
net_tax_value_after_homestead	428.34	1542.58	
agreement_agency	TIPS	NaN	
sequestration_enforcement	False	False	
bankruptcy	False	False	
years_in_bankruptcy	NaN	NaN	
most_recent_bankrupt_year	NaN	NaN	
oldest_bankrupt_year	NaN	NaN	
principal_sum_bankrupt_years	NaN	NaN	
total_amount_bankrupt_years	NaN	NaN	
sheriff_sale	N	N	
liens_sold_1990s	False	False	
liens_sold_2015	N	N	
assessment_under_appeal	False	False	
year_month	202111	202111	
lat	-75.233571	-75.23451	
lng	39.915321	39.914297	
	72715	72716	\
objectid	2628143	2628144	
opa_number	406313100.0	406317000.0	
street_address	6737 GUYER AVE	6853 GUYER AVE	
zip_code	19142.0	19142.0	
zip_4	2610.0	2518.0	
unit_type	NaN	NaN	
unit_num	NaN	NaN	
owner	COPPOLA JOHN	JEAN-LOUIS PATRICIA	
co_owner	COPPOLA JOHN	JEAN-LOUIS PATRICIA	
principal_due	2504.38	4079.25	
penalty_due	115.86	219.7	
interest_due	311.99	697.49	
other_charges_due	392.2	1030.92	
total_due	3324.43	6027.36	

is_actionable	False	True
payment_agreement	True	False
num_years_owed	4	4
most_recent_year_owed	2021	2021
oldest_year_owed	2018	2018
most_recent_payment_date	2021-09-13 00:00:00	2018-03-09 00:00:00
year_of_last_assessment	2021.0	2021.0
total_assessment	78200.0	67200.0
taxable_assessment	78200.0	67200.0
mailing_address	1742 DELSEA DR	6853 GUYER AVE
mailing_city	DEPTFORD	PHILA
mailing_state	NJ	PA
mailing_zip	8096.0	19142.0
return_mail	NaN	NaN
building_code	R30	R30
detail_building_description	ROW B/GAR 2STY MASON	ROW B/GAR 2STY MASON
general_building_description	house	house
building_category	residential	residential
coll_agency_num_years	0	3
coll_agency_most_recent_year	NaN	2020.0
coll_agency_oldest_year	NaN	2018.0
coll_agency_principal_owed	0.0	3138.58
coll_agency_total_owed	0.0	4959.7
exempt_abatement_assessment	0.0	0.0
homestead_value	0.0	0.0
net_tax_value_after_homestead	1094.64	940.67
agreement_agency	TIPS	NaN
sequestration_enforcement	False	False
bankruptcy	False	False
years_in_bankruptcy	NaN	NaN
most_recent_bankrupt_year	NaN	NaN
oldest_bankrupt_year	NaN	NaN
principal_sum_bankrupt_years	NaN	NaN
total_amount_bankrupt_years	NaN	NaN
sheriff_sale	N	N
liens_sold_1990s	False	False
liens_sold_2015	N	N
assessment_under_appeal	False	False
year_month	202111	202111
lat	-75.230644	-75.232507
lng	39.917135	39.915844
	72717	
objectid	2628145	
opa_number	406349800.0	
street_address	6716 DOREL ST	
zip_code	19142.0	

zip_4	2607.0
unit_type	NaN
unit_num	NaN
owner	HABIL HOUSSEINI
co_owner	HABIL HOUSSEINI
principal_due	2301.27
penalty_due	84.56
interest_due	256.31
other_charges_due	360.15
total_due	3002.29
is_actionable	True
payment_agreement	False
num_years_owed	2
most_recent_year_owed	2021
oldest_year_owed	2020
most_recent_payment_date	2019-11-18 00:00:00
year_of_last_assessment	2021.0
total_assessment	78100.0
taxable_assessment	78100.0
mailing_address	2527 S CARROLL ST
mailing_city	PHILADELPHIA
mailing_state	PA
mailing_zip	19142.0
return_mail	NaN
building_code	R30
detail_building_description	ROW B/GAR 2STY MASON
general_building_description	house
building_category	residential
coll_agency_num_years	1
coll_agency_most_recent_year	2020.0
coll_agency_oldest_year	2020.0
coll_agency_principal_owed	1208.03
coll_agency_total_owed	1761.46
exempt_abatement_assessment	0.0
homestead_value	0.0
net_tax_value_after_homestead	1093.24
agreement_agency	NaN
sequestration_enforcement	False
bankruptcy	False
years_in_bankruptcy	NaN
most_recent_bankrupt_year	NaN
oldest_bankrupt_year	NaN
principal_sum_bankrupt_years	NaN
total_amount_bankrupt_years	NaN
sheriff_sale	N
liens_sold_1990s	False
liens_sold_2015	N

```
assessment_under_appeal           False
year_month                          202111
lat                                 -75.229525
lng                                39.916847
```

[55 rows x 72718 columns]

```
[ ]: tax.isna().sum()#null values included in the dataset
```

```
[ ]: objectid                  0
opa_number                   3
street_address                10
zip_code                      59
zip_4                         3102
unit_type                     71339
unit_num                      71339
owner                         2
co_owner                       98
principal_due                 0
penalty_due                    0
interest_due                   0
other_charges_due              0
total_due                      0
is_actionable                  0
payment_agreement              0
num_years_owed                  0
most_recent_year_owed            0
oldest_year_owed                  0
most_recent_payment_date        4920
year_of_last_assessment         1482
total_assessment                 1482
taxable_assessment                 1482
mailing_address                  46023
mailing_city                     46022
mailing_state                     46025
mailing_zip                      46024
return_mail                      61696
building_code                     1484
detail_building_description       1487
general_building_description      1487
building_category                  1487
coll_agency_num_years               0
coll_agency_most_recent_year       34121
coll_agency_oldest_year                 34121
coll_agency_principal_owed             0
coll_agency_total_owed                  0
exempt_abatement_assessment          1482
```

```

homestead_value           1482
net_tax_value_after_homestead 1482
agreement_agency          48917
sequestration_enforcement 0
bankruptcy                 0
years_in_bankruptcy        72171
most_recent_bankrupt_year 72171
oldest_bankrupt_year       72171
principal_sum_bankrupt_years 72171
total_amount_bankrupt_years 72171
sheriff_sale                0
liens_sold_1990s             0
liens_sold_2015                0
assessment_under_appeal      0
year_month                   0
lat                           193
lng                           193
dtype: int64

```

[]: tax.shape #size of dataset

[]: (72718, 55)

[]: tax.isna().sum()/tax.shape[0] # remove mailing_address, mailing_city, unit_type, unit_num, mailing_state, mailing_zip, # remove return_mail, coll_agency_most_recent_year, coll_agency.oldest_year, agreement_agency, years_in_bankruptcy , # remove most_recent_bankrupt_year, oldest_bankrupt_year, principal_sum_bankrupt_years, total_amount_bankrupt_years

objectid	0.000000
opa_number	0.000041
street_address	0.000138
zip_code	0.000811
zip_4	0.042658
unit_type	0.981036
unit_num	0.981036
owner	0.000028
co_owner	0.001348
principal_due	0.000000
penalty_due	0.000000
interest_due	0.000000
other_charges_due	0.000000
total_due	0.000000
is_actionable	0.000000
payment_agreement	0.000000
num_years_owed	0.000000

```

most_recent_year_owed           0.000000
oldest_year_owed                0.000000
most_recent_payment_date        0.067659
year_of_last_assessment         0.020380
total_assessment                 0.020380
taxable_assessment               0.020380
mailing_address                  0.632897
mailing_city                     0.632883
mailing_state                    0.632924
mailing_zip                      0.632911
return_mail                      0.848428
building_code                     0.020408
detail_building_description      0.020449
general_building_description     0.020449
building_category                 0.020449
coll_agency_num_years            0.000000
coll_agency_most_recent_year    0.469224
coll_agency_oldest_year          0.469224
coll_agency_principal_owed       0.000000
coll_agency_total_owed           0.000000
exempt_abatement_assessment     0.020380
homestead_value                  0.020380
net_tax_value_after_homestead   0.020380
agreement_agency                 0.672695
sequestration_enforcement        0.000000
bankruptcy                       0.000000
years_in_bankruptcy              0.992478
most_recent_bankrupt_year        0.992478
oldest_bankrupt_year             0.992478
principal_sum_bankrupt_years     0.992478
total_amount_bankrupt_years       0.992478
sheriff_sale                     0.000000
liens_sold_1990s                 0.000000
liens_sold_2015                   0.000000
assessment_under_appeal          0.000000
year_month                        0.000000
lat                               0.002654
lng                               0.002654
dtype: float64

```

```
[ ]: tax.drop(['mailing_address', 'mailing_city', 'unit_type', 'unit_num',  

    ↪'mailing_state', 'mailing_zip',  

    ↪'return_mail', 'coll_agency_most_recent_year',  

    ↪'coll_agency_oldest_year', 'agreement_agency', 'years_in_bankruptcy' ,  

    ↪'most_recent_bankrupt_year', 'oldest_bankrupt_year',  

    ↪'principal_sum_bankrupt_years', 'total_amount_bankrupt_years'],  

    axis=1, inplace=True) # removing columns thta have high null values
```

```
[ ]: tax.columns
```

```
[ ]: Index(['objectid', 'opa_number', 'street_address', 'zip_code', 'zip_4',
       'owner', 'co_owner', 'principal_due', 'penalty_due', 'interest_due',
       'other_charges_due', 'total_due', 'is_actionable', 'payment_agreement',
       'num_years_owed', 'most_recent_year_owed', 'oldest_year_owed',
       'most_recent_payment_date', 'year_of_last_assessment',
       'total_assessment', 'taxable_assessment', 'building_code',
       'detail_building_description', 'general_building_description',
       'building_category', 'coll_agency_num_years',
       'coll_agency_principal_owed', 'coll_agency_total_owed',
       'exempt_abatement_assessment', 'homestead_value',
       'net_tax_value_after_homestead', 'sequestration_enforcement',
       'bankruptcy', 'sheriff_sale', 'liens_sold_1990s', 'liens_sold_2015',
       'assessment_under_appeal', 'year_month', 'lat', 'lng'],
      dtype='object')
```

```
[ ]: (tax.isna().sum()/tax.shape[0]).sort_values(ascending=False)
```

```
[ ]: most_recent_payment_date      0.067659
zip_4                           0.042658
building_category                 0.020449
general_building_description     0.020449
detail_building_description      0.020449
building_code                     0.020408
taxable_assessment                0.020380
net_tax_value_after_homestead    0.020380
homestead_value                  0.020380
exempt_abatement_assessment     0.020380
total_assessment                  0.020380
year_of_last_assessment          0.020380
lat                             0.002654
lng                            0.002654
co_owner                         0.001348
zip_code                          0.000811
street_address                   0.000138
opa_number                       0.000041
owner                            0.000028
num_years_owed                   0.000000
year_month                        0.000000
assessment_under_appeal          0.000000
liens_sold_2015                  0.000000
liens_sold_1990s                 0.000000
sheriff_sale                      0.000000
bankruptcy                        0.000000
sequestration_enforcement        0.000000
coll_agency_total_owed           0.000000
```

```
most_recent_year_owed           0.000000
coll_agency_principal_owed     0.000000
coll_agency_num_years          0.000000
principal_due                  0.000000
penalty_due                    0.000000
interest_due                   0.000000
other_charges_due              0.000000
total_due                      0.000000
is_actionable                  0.000000
payment_agreement              0.000000
oldest_year_owed                0.000000
objectid                       0.000000
dtype: float64
```

```
[ ]: #minimum and maximum latitude
print(tax['lat'].min())
print(tax['lat'].max())
```

```
-75.27396176893846
-74.9593407520959
```

```
[ ]: #minimum and maximum longitude
print(tax['lng'].min())
print(tax['lng'].max())
```

```
39.88640598393346
40.13621377256551
```

```
[ ]: #dropping null values within longitude and latitude
#tax.dropna(subset=['lat'], inplace=True)
#tax.dropna(subset=['lng'], inplace=True)
```

```
[ ]: #combining latitude and longitude data into one column called geometry for
    ↪gropandas to read
crs = {'init': 'epsg:4326'}
geometry = [Point(xy) for xy in zip(tax["lat"], tax["lng"])]
tax = gpd.GeoDataFrame(tax,
                       crs = crs,
                       geometry = geometry)

tax.head()
```

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/pyproj/crs/crs.py:131: FutureWarning: '+init=<authority>:<code>' syntax
is deprecated. '<authority>:<code>' is the preferred initialization method. When
making the change, be mindful of axis order changes:
https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-
```

```

proj-6
    in_crs_string = _prepare_from_proj_string(in_crs_string)

[ ]:   objectid    opa_number    street_address    zip_code    zip_4  \
0    2556493    493169300.0    6045 N CAMAC ST    19141.0    3227.0
1    2556494    493179100.0    5620 N CAMAC ST    19141.0    4106.0
2    2556495    493180700.0    5714 N CAMAC ST    19141.0    4108.0
3    2556496    493183600.0    5812 N CAMAC ST    19141.0    4123.0
4    2556497    223166200.0    420 GLEN ECHO RD    19119.0    2914.0

            owner        co_owner    principal_due    penalty_due  \
0  WILLIAMS JACQUELINE  WILLIAMS JACQUELINE      12200.18     1110.17
1          RAY MATTIE E      RAY MATTIE E       -0.05      0.00
2        TOMLIN PAULA      TOMLIN PAULA      895.87      0.00
3  BATTS PRINCETON B      BATTS PRINCETON B     4536.94     283.14
4    WHITE CLARENCE      WHITE CLARENCE      4224.60      0.00

    interest_due ... sequestration_enforcement    bankruptcy    sheriff_sale  \
0      14261.97 ...                   False      False           N
1        41.05 ...                   False      False           N
2       120.94 ...                   False      False           N
3      1161.05 ...                   False      False           N
4       570.32 ...                   False      False           N

    liens_sold_1990s    liens_sold_2015    assessment_under_appeal    year_month  \
0             False                  N                  False     202111
1             False                  N                  False     202111
2             False                  N                  False     202111
3             False                  N                  False     202111
4             False                  N                  False     202111

        lat      lng            geometry
0 -75.140099  40.045081 POINT (-75.14010 40.04508)
1 -75.141930  40.039007 POINT (-75.14193 40.03901)
2 -75.141727  40.039940 POINT (-75.14173 40.03994)
3 -75.141395  40.041404 POINT (-75.14140 40.04140)
4 -75.195309  40.051563 POINT (-75.19531 40.05156)

[5 rows x 41 columns]

```

```

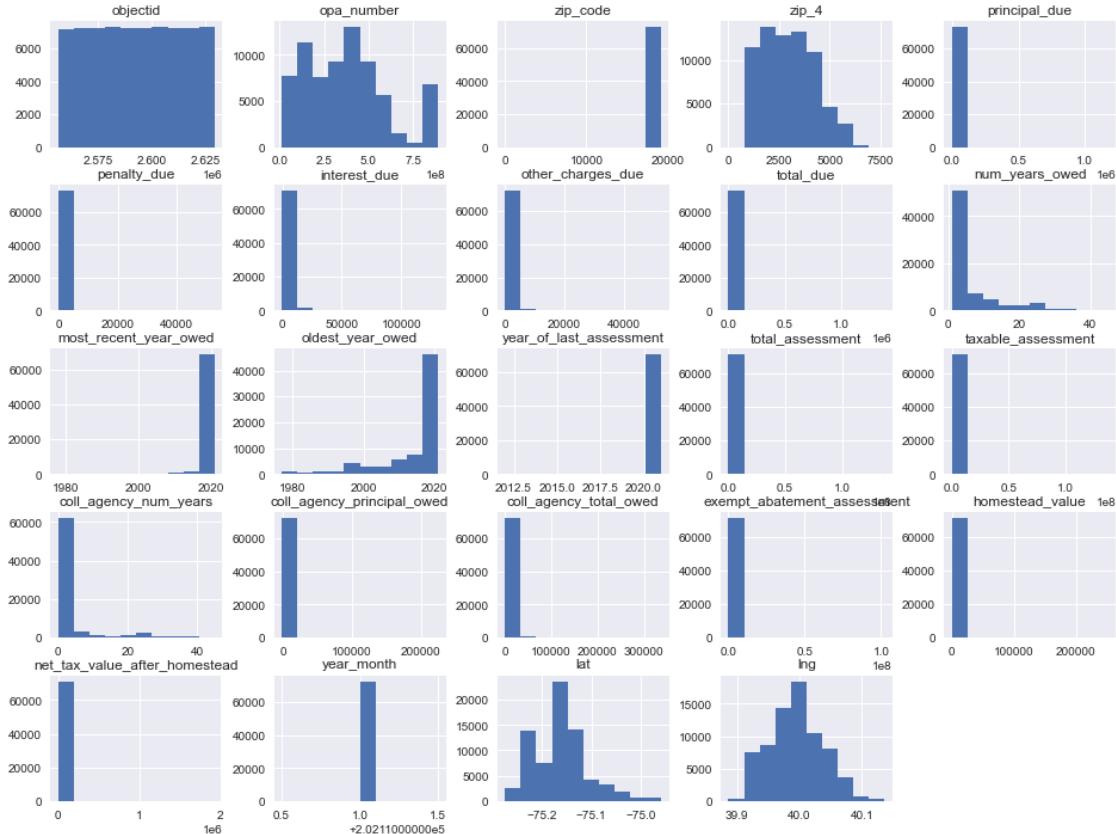
[ ]: #histogram of the dataset
#important column that we will be considering is the principal due column. The data is skewed
fig, ax = plt.subplots(figsize=(16,12))
tax.hist(ax=ax)

```

```
/var/folders/6p/wpw9qml57530xkxqkkhprrf40000gn/T/ipykernel_21232/1960141603.py:4
: UserWarning: To output multiple subplots, the figure containing the passed
```

```
axes is being cleared
tax.hist(ax=ax)

[ ]: array([[<AxesSubplot:title={'center':'objectid'}>,
   <AxesSubplot:title={'center':'opa_number'}>,
   <AxesSubplot:title={'center':'zip_code'}>,
   <AxesSubplot:title={'center':'zip_4'}>,
   <AxesSubplot:title={'center':'principal_due'}>],
  [<AxesSubplot:title={'center':'penalty_due'}>,
   <AxesSubplot:title={'center':'interest_due'}>,
   <AxesSubplot:title={'center':'other_charges_due'}>,
   <AxesSubplot:title={'center':'total_due'}>,
   <AxesSubplot:title={'center':'num_years_owed'}>],
  [<AxesSubplot:title={'center':'most_recent_year_owed'}>,
   <AxesSubplot:title={'center':'oldest_year_owed'}>,
   <AxesSubplot:title={'center':'year_of_last_assessment'}>,
   <AxesSubplot:title={'center':'total_assessment'}>,
   <AxesSubplot:title={'center':'taxable_assessment'}>],
  [<AxesSubplot:title={'center':'coll_agency_num_years'}>,
   <AxesSubplot:title={'center':'coll_agency_principal_owed'}>,
   <AxesSubplot:title={'center':'coll_agency_total_owed'}>,
   <AxesSubplot:title={'center':'exempt_abatement_assessment'}>,
   <AxesSubplot:title={'center':'homestead_value'}>],
  [<AxesSubplot:title={'center':'net_tax_value_after_homestead'}>,
   <AxesSubplot:title={'center':'year_month'}>,
   <AxesSubplot:title={'center':'lat'}>,
   <AxesSubplot:title={'center':'lng'}>, <AxesSubplot:>]],
 dtype=object)
```



```
[ ]: tax.describe().T # median for principal value is around #2000
```

	count	mean	std	\
objectid	72718.0	2.592825e+06	2.099205e+04	
opa_number	72715.0	3.736434e+08	2.359174e+08	
zip_code	72659.0	1.911508e+04	6.062668e+02	
zip_4	69616.0	2.969683e+03	1.271406e+03	
principal_due	72718.0	3.176549e+03	1.046773e+04	
penalty_due	72718.0	2.140727e+02	5.634433e+02	
interest_due	72718.0	1.753979e+03	4.432956e+03	
other_charges_due	72718.0	7.360034e+02	1.324633e+03	
total_due	72718.0	5.881654e+03	1.499406e+04	
num_years_owed	72718.0	6.222710e+00	8.219798e+00	
most_recent_year_owed	72718.0	2.020151e+03	3.533746e+00	
oldest_year_owed	72718.0	2.014279e+03	9.701235e+00	
year_of_last_assessment	71236.0	2.020955e+03	5.007728e-01	
total_assessment	71236.0	1.471699e+05	1.394407e+06	
taxable_assessment	71236.0	1.263154e+05	1.022020e+06	
coll_agency_num_years	72718.0	3.203320e+00	7.340741e+00	
coll_agency_principal_owed	72718.0	1.108075e+03	3.931734e+03	
coll_agency_total_owed	72718.0	2.212659e+03	6.945926e+03	

	min	25%	50%	\
exempt_abatement_assessment	71236.0	2.085454e+04	8.460790e+05	
homestead_value	71236.0	2.219765e+02	9.914458e+02	
net_tax_value_after_homestead	71236.0	1.546778e+03	1.387922e+04	
year_month	72718.0	2.021110e+05	0.000000e+00	
lat	72525.0	-7.516213e+01	5.420265e-02	
lng	72525.0	3.999380e+01	4.274897e-02	
objectid	2.556371e+06	2.574646e+06	2.592826e+06	
opa_number	1.100080e+07	1.810471e+08	3.640104e+08	
zip_code	1.000000e+00	1.912800e+04	1.913500e+04	
zip_4	3.000000e+00	1.907000e+03	2.908000e+03	
principal_due	-1.741060e+03	2.435700e+02	1.194025e+03	
penalty_due	-1.089500e+02	0.000000e+00	5.722000e+01	
interest_due	-4.703100e+02	7.840250e+01	2.661600e+02	
other_charges_due	-4.526000e+01	0.000000e+00	2.733000e+02	
total_due	1.000000e-02	4.607400e+02	2.016135e+03	
num_years_owed	1.000000e+00	1.000000e+00	3.000000e+00	
most_recent_year_owed	1.978000e+03	2.021000e+03	2.021000e+03	
oldest_year_owed	1.977000e+03	2.012000e+03	2.019000e+03	
year_of_last_assessment	2.012000e+03	2.021000e+03	2.021000e+03	
total_assessment	0.000000e+00	3.430000e+04	7.300000e+04	
taxable_assessment	0.000000e+00	3.300000e+04	7.120000e+04	
coll_agency_num_years	0.000000e+00	0.000000e+00	1.000000e+00	
coll_agency_principal_owed	-1.741060e+03	0.000000e+00	0.000000e+00	
coll_agency_total_owed	0.000000e+00	0.000000e+00	5.723000e+01	
exempt_abatement_assessment	0.000000e+00	0.000000e+00	0.000000e+00	
homestead_value	0.000000e+00	0.000000e+00	0.000000e+00	
net_tax_value_after_homestead	0.000000e+00	2.771600e+02	7.509900e+02	
year_month	2.021110e+05	2.021110e+05	2.021110e+05	
lat	-7.527396e+01	-7.519973e+01	-7.516256e+01	
lng	3.988641e+01	3.996484e+01	3.999254e+01	
	75%	max		
objectid	2.611005e+06	2.629184e+06		
opa_number	5.021892e+08	8.888007e+08		
zip_code	1.914300e+04	1.919200e+04		
zip_4	3.920000e+03	7.711000e+03		
principal_due	3.359450e+03	1.174312e+06		
penalty_due	2.471375e+02	5.273024e+04		
interest_due	1.245483e+03	1.307921e+05		
other_charges_due	9.627350e+02	5.252586e+04		
total_due	6.232828e+03	1.391257e+06		
num_years_owed	7.000000e+00	4.500000e+01		
most_recent_year_owed	2.021000e+03	2.021000e+03		
oldest_year_owed	2.021000e+03	2.021000e+03		
year_of_last_assessment	2.021000e+03	2.021000e+03		

total_assessment	1.307000e+05	1.375769e+08
taxable_assessment	1.258000e+05	1.375769e+08
coll_agency_num_years	2.000000e+00	4.500000e+01
coll_agency_principal_owed	1.030368e+03	2.245500e+05
coll_agency_total_owed	1.723820e+03	3.371071e+05
exempt_abatement_assessment	0.000000e+00	1.023300e+08
homestead_value	6.299100e+02	2.519640e+05
net_tax_value_after_homestead	1.455790e+03	1.925801e+06
year_month	2.021110e+05	2.021110e+05
lat	-7.513773e+01	-7.495934e+01
lng	4.002372e+01	4.013621e+01

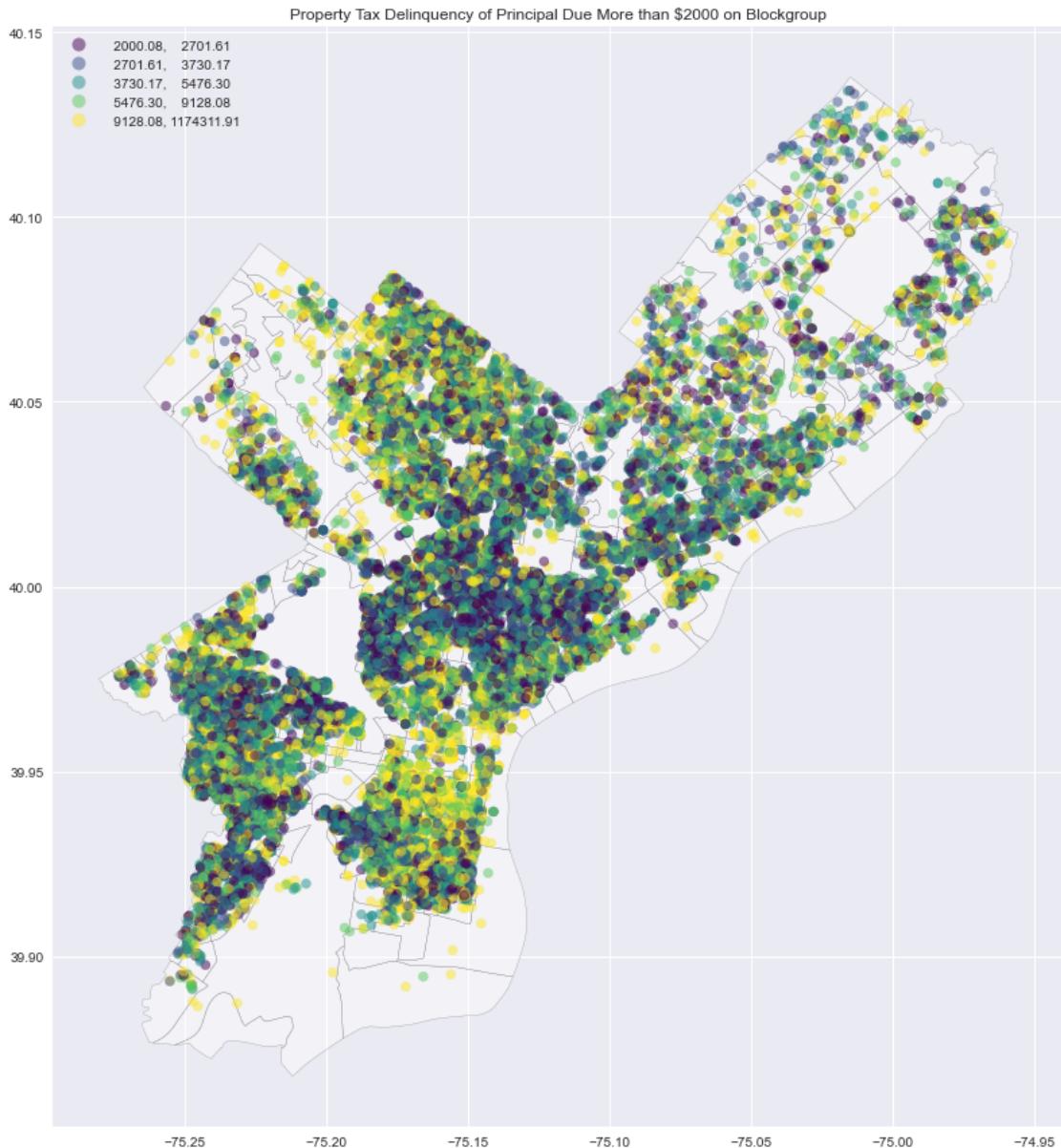
```
[ ]: tax.isna().sum()
```

objectid	0
opa_number	3
street_address	10
zip_code	59
zip_4	3102
owner	2
co_owner	98
principal_due	0
penalty_due	0
interest_due	0
other_charges_due	0
total_due	0
is_actionable	0
payment_agreement	0
num_years_owed	0
most_recent_year_owed	0
oldest_year_owed	0
most_recent_payment_date	4920
year_of_last_assessment	1482
total_assessment	1482
taxable_assessment	1482
building_code	1484
detail_building_description	1487
general_building_description	1487
building_category	1487
coll_agency_num_years	0
coll_agency_principal_owed	0
coll_agency_total_owed	0
exempt_abatement_assessment	1482
homestead_value	1482
net_tax_value_after_homestead	1482
sequestration_enforcement	0
bankruptcy	0

```
sheriff_sale          0
liens_sold_1990s      0
liens_sold_2015        0
assessment_under_appeal 0
year_month             0
lat                     193
lng                     193
geometry                 0
dtype: int64
```

```
[ ]: fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
plt.title("Property Tax Delinquency of Principal Due More than $2000 on
           ↪Blockgroup")
census_blockgroups.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color =
           ↪"white", edgecolor='black') #using shape map of neighbourhood
tax[tax['principal_due'] > 2000].plot(column = "principal_due", ax = ax,
           ↪scheme="quantiles", legend=True, alpha = 0.5) #principal that is more than 2000
```

```
[ ]: <AxesSubplot:title={'center':'Property Tax Delinquency of Principal Due More
than $2000 on Blockgroup'}>
```



```
[ ]: tax1 = tax.copy()
tax1.dropna(subset=['zip_code'], inplace=True) # dropping null values zip code
tax1['zip_code'] = tax1['zip_code'].astype(int).astype(str) # turning number to
→ integer then to object
```

```
[ ]: tax1.dtypes
```

objectid	int64
opa_number	float64
street_address	object

zip_code	object
zip_4	float64
owner	object
co_owner	object
principal_due	float64
penalty_due	float64
interest_due	float64
other_charges_due	float64
total_due	float64
is_actionable	bool
payment_agreement	bool
num_years_owed	int64
most_recent_year_owed	int64
oldest_year_owed	int64
most_recent_payment_date	object
year_of_last_assessment	float64
total_assessment	float64
taxable_assessment	float64
building_code	object
detail_building_description	object
general_building_description	object
building_category	object
coll_agency_num_years	int64
coll_agency_principal_owed	float64
coll_agency_total_owed	float64
exempt_abatement_assessment	float64
homestead_value	float64
net_tax_value_after_homestead	float64
sequestration_enforcement	bool
bankruptcy	bool
sheriff_sale	object
liens_sold_1990s	bool
liens_sold_2015	object
assessment_under_appeal	bool
year_month	int64
lat	float64
lng	float64
geometry	geometry
dtype:	object

```
[ ]: tax_zip_sum = tax1.groupby('zip_code')[['principal_due','total_due',  
    ↪'is_actionable']].sum().reset_index() #grouping by zip code and sum of values  
tax_zip_count = tax1.groupby('zip_code')[['opa_number']].count().  
    ↪reset_index() #group by zip codes and number of delinquent properties
```

```
[ ]:
```

```

poly_zip_tax_count = pd.merge(poly_zip, tax_zip_count, left_on = "CODE",
                             right_on = "zip_code", how = 'left')#combining the tax file with zip code
                             #with zip code shape file
poly_zip_tax_count.head()

```

```

[ ]:   OBJECTID    CODE    COD    Shape__Are    Shape__Len  \
0        1    19120    20  9.177970e+07  49921.544063
1        2    19121    21  6.959879e+07  39534.887217
2        3    19122    22  3.591632e+07  24124.645221
3        4    19123    23  3.585175e+07  26421.728982
4        5    19124    24  1.448080e+08  63658.770420

```

			geometry	zip_code	opa_number
0	POLYGON	((-75.11107 40.04682, -75.10943 40.045...		19120	2039.0
1	POLYGON	((-75.19227 39.99463, -75.19205 39.994...		19121	2952.0
2	POLYGON	((-75.15406 39.98601, -75.15328 39.985...		19122	1334.0
3	POLYGON	((-75.15190 39.97056, -75.15150 39.970...		19123	555.0
4	POLYGON	((-75.09660 40.04249, -75.09281 40.039...		19124	2394.0

```

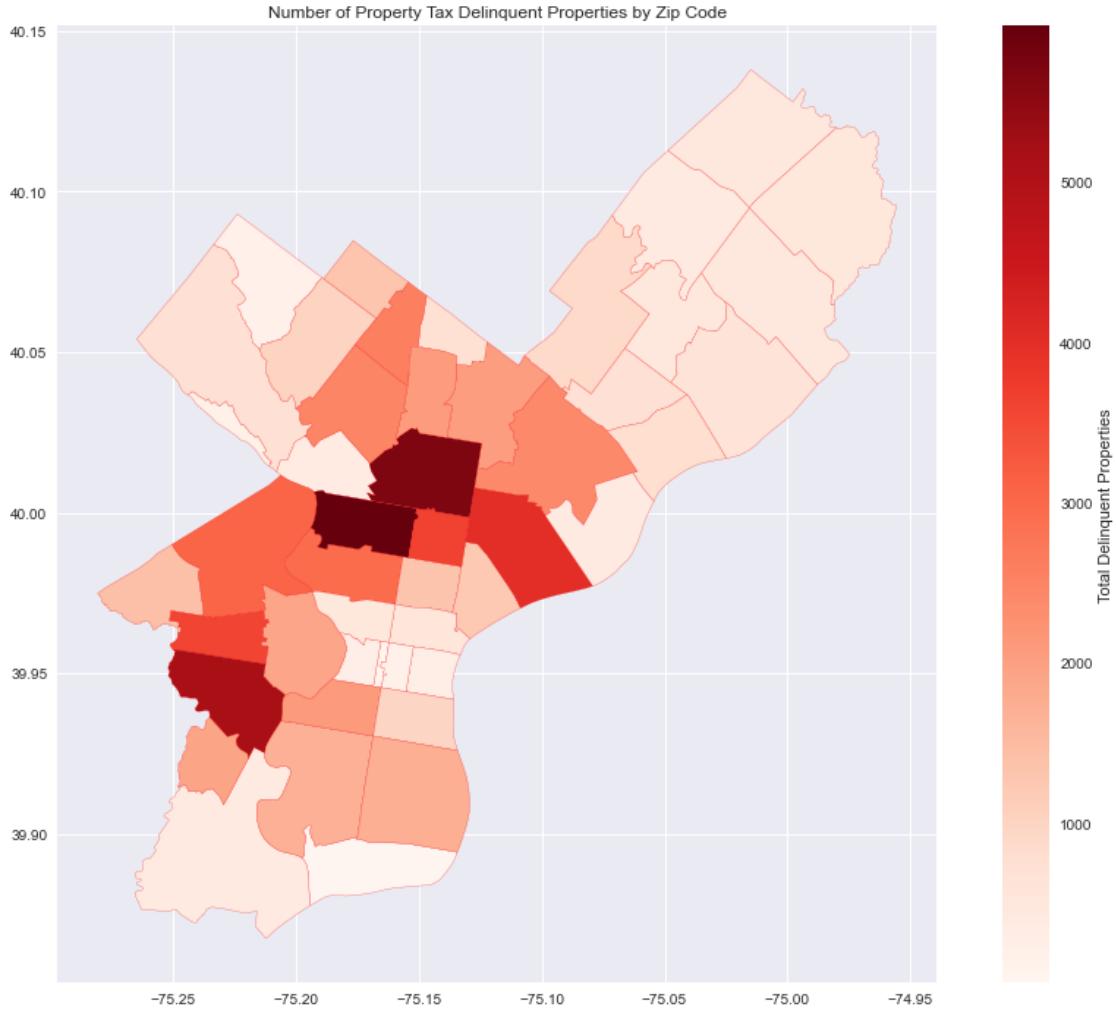
[ ]: fig, ax = plt.subplots(figsize=(16,12))
plt.title("Number of Property Tax Delinquent Properties by Zip Code")
poly_zip_tax_count.plot(ax=ax, column='opa_number',
                        edgecolor='red', linewidth=.2,
                        cmap='Reds', legend=True,
                        legend_kwds={'label': 'Total Delinquent Properties'})
#sum of delinquent properties and the number of delinquent properties seems to
#be around in the same zip codes

```

```

[ ]: <AxesSubplot:title={'center':'Number of Property Tax Delinquent Properties by
Zip Code'}>

```



```
[ ]: poly_tax_zip_sum = pd.merge(poly_zip, tax_zip_sum, left_on = "CODE", right_on = "zip_code", how = 'left')#merging zip code shape file with sum of values dataset
poly_tax_zip_sum.head()
```

```
[ ]:   OBJECTID    CODE    COD      Shape__Are      Shape__Len \
0          1  19120     20  9.177970e+07  49921.544063
1          2  19121     21  6.959879e+07  39534.887217
2          3  19122     22  3.591632e+07  24124.645221
3          4  19123     23  3.585175e+07  26421.728982
4          5  19124     24  1.448080e+08  63658.770420
```

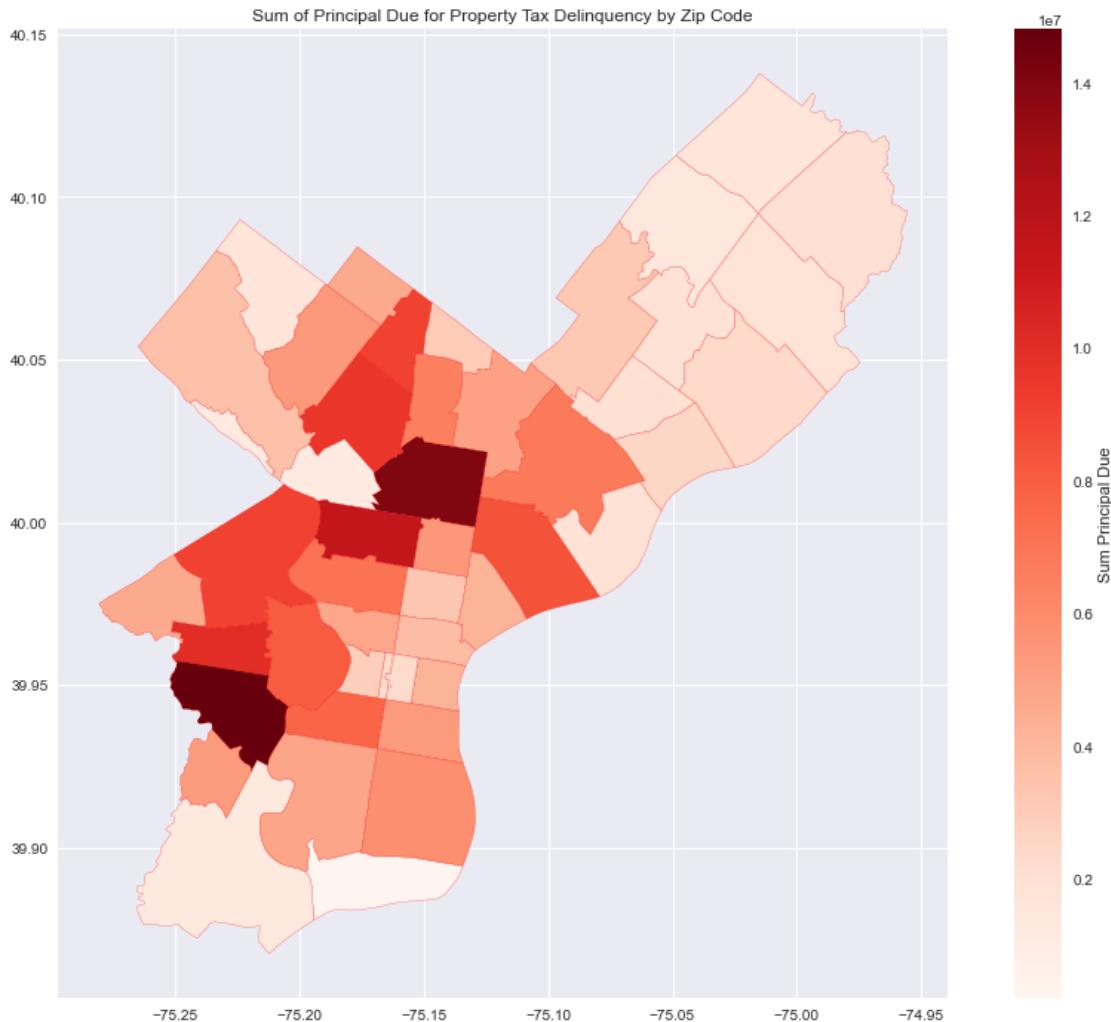
		geometry	zip_code	principal_due	\
0	POLYGON	((-75.11107 40.04682, -75.10943 40.045...	19120	5017812.12	
1	POLYGON	((-75.19227 39.99463, -75.19205 39.994...	19121	7097123.98	
2	POLYGON	((-75.15406 39.98601, -75.15328 39.985...	19122	3346511.40	

```
3  POLYGON ((-75.15190 39.97056, -75.15150 39.970...,      19123      3766900.87
4  POLYGON ((-75.09660 40.04249, -75.09281 40.039...,      19124      6791902.67
```

	total_due	is_actionable
0	9529801.48	880.0
1	12963178.49	1829.0
2	5845794.91	738.0
3	5433204.50	329.0
4	12673441.60	1143.0

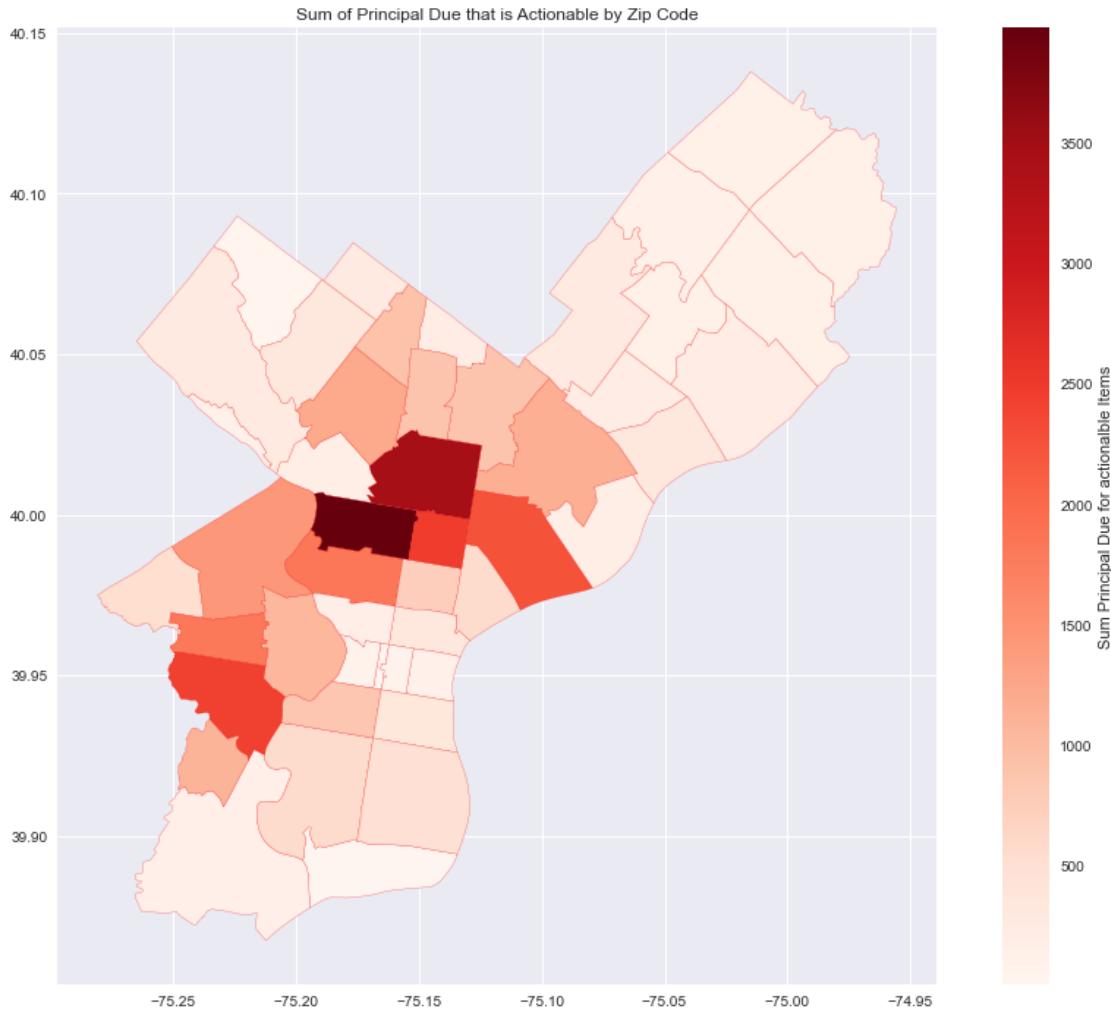
```
[ ]: fig, ax = plt.subplots(figsize=(16,12))
plt.title("Sum of Principal Due for Property Tax Delinquency by Zip Code")
poly_tax_zip_sum.plot(ax=ax, column='principal_due',
                      edgecolor='red', linewidth=.2,
                      cmap='Reds', legend=True,
                      legend_kwds={'label': 'Sum Principal Due'})
```

```
[ ]: <AxesSubplot:title={'center':'Sum of Principal Due for Property Tax Delinquency
by Zip Code'}>
```



```
[ ]: fig, ax = plt.subplots(figsize=(16,12))
plt.title("Sum of Principal Due that is Actionable by Zip Code")
poly_tax_zip_sum.plot(ax=ax, column='is_actionable',
                      edgecolor='red', linewidth=.2,
                      cmap='Reds', legend=True,
                      legend_kwds={'label': 'Sum Principal Due for actionable Items'})
```

```
[ ]: <AxesSubplot:title={'center':'Sum of Principal Due that is Actionable by Zip Code'}>
```



```
[ ]: tax['principal_due'].sum() #sum of principal due
```

```
[ ]: 230181487.98999998
```

```
[ ]: tax['is_actionable'].unique() #this is boolean value
```

```
[ ]: array([False, True])
```

```
[ ]: tax.loc[tax['is_actionable']== True]['principal_due'].sum() #sum of principal due that is actionable
#Actionable means that the city is actively working to collect these accounts,
#non-actional means that the city can't do anything further or they are barred from collection
```

```
[ ]: 121432247.78
```

```
[ ]: tax.groupby(tax['is_actionable'])['principal_due'].sum().reset_index() # is_u  
→actionable and not actionable is not that different  
#Accounts that are in payment agreement, bankruptcy, or overdue but not yet_u  
→delinquent are considered "not actionable".
```

```
[ ]:   is_actionable  principal_due  
0          False  1.087492e+08  
1          True   1.214322e+08
```

```
[ ]: tax.groupby(tax['is_actionable'])['principal_due'].sum()/tax['principal_due'].  
→sum()#percentage of actional and non-actionable principal due is almost the_u  
→same  
#more percentage of principal due for actionaable
```

```
[ ]: is_actionable  
False    0.47245  
True     0.52755  
Name: principal_due, dtype: float64
```

```
[ ]: tax.groupby(tax['num_years_owed'])['principal_due'].sum().reset_index().  
→sort_values(by = ['principal_due'], ascending= False)  
#Most of the principal due is owned for 1-4 years, then there is 25 years which_u  
→has the highest principal due
```

```
[ ]:   num_years_owed  principal_due  
1                  2    27095884.35  
0                  1    26374549.84  
2                  3    22583624.45  
3                  4    16766515.02  
24                 25   13174576.77  
4                  5    12545011.90  
5                  6    11068376.28  
6                  7    8119525.66  
7                  8    7243378.20  
11                 12   6592288.78  
8                  9    6358223.15  
9                  10   6066086.24  
10                 11   5513193.91  
13                 14   5017793.37  
12                 13   4920943.23  
14                 15   4249660.21  
20                 21   3845421.86  
15                 16   3773332.27  
23                 24   3632977.73  
17                 18   3513447.44  
16                 17   3380624.12  
18                 19   3124991.57
```

22	23	3083700.57
21	22	3063217.30
19	20	2910133.65
27	28	1472989.17
25	26	1450242.28
26	27	1429127.75
28	29	1246181.70
32	33	1114820.15
31	32	1077972.33
43	44	1044263.97
30	31	973101.27
29	30	868240.63
33	34	831774.88
35	36	780037.34
34	35	765812.31
36	37	718980.99
37	38	574930.25
38	39	516875.32
42	43	381043.12
39	40	368221.62
41	42	292565.37
40	41	253507.86
44	45	3321.81

```
[ ]: tax.groupby(tax['num_years_owed'])['principal_due'].median().reset_index()
      ↪sort_values(by = ['principal_due'], ascending= False)
#when you do median principal dues, it is 18, 23, 22 and 27 years.
#principal value is skewed so meadian is a better measure
```

	num_years_owed	principal_due
17	18	5777.040
22	23	5735.850
21	22	5725.700
26	27	5666.460
18	19	5284.775
27	28	5131.220
25	26	4909.580
19	20	4787.910
15	16	4720.230
16	17	4675.645
39	40	4592.710
23	24	4565.720
13	14	4516.870
41	42	4448.170
30	31	4270.680
42	43	4204.530
14	15	4095.185

40	41	4017.970
29	30	3820.270
24	25	3720.550
38	39	3698.420
11	12	3592.830
12	13	3572.330
43	44	3497.240
36	37	3478.085
32	33	3436.920
37	38	3414.990
9	10	3324.215
44	45	3321.810
35	36	3274.590
8	9	3240.150
10	11	3221.510
34	35	3158.905
7	8	3075.225
6	7	2882.405
33	34	2866.385
31	32	2805.345
5	6	2588.630
28	29	2411.840
4	5	2272.890
3	4	2005.555
2	3	1703.040
20	21	1406.630
1	2	1082.360
0	1	208.570

```
[ ]: tax.groupby(tax['num_years_owed'])['opa_number'].count().reset_index()
    ↪sort_values(by = ['opa_number'], ascending= False)
#most of the delinquent properties are between owed for 1-6 year , then there
    ↪is 25 years owned of delinquent properties
```

	num_years_owed	opa_number
0	1	26431
1	2	9579
2	3	6675
3	4	4566
4	5	3313
5	6	2635
24	25	2278
6	7	1828
7	8	1592
8	9	1247
9	10	1166
10	11	1081

```
11          12      1017
12          13      863
20          21      800
13          14      760
14          15      666
15          16      528
16          17      488
23          24      456
17          18      446
18          19      422
19          20      402
21          22      400
22          23      378
28          29      224
43          44      219
31          32      194
25          26      191
27          28      190
29          30      166
30          31      165
32          33      164
26          27      159
35          36      151
33          34      148
34          35      142
36          37      116
37          38      106
38          39      95
42          43      57
39          40      56
40          41      52
41          42      46
44          45      1
```

```
[ ]: tax.groupby(tax['building_category'])['opa_number'].count() # most of the ↴  
↳delinquent properties are residential
```

```
[ ]: building_category  
commercial      6037  
residential    65192  
Name: opa_number, dtype: int64
```

```
[ ]: tax.groupby(tax['building_category'])['opa_number'].count()/tax['opa_number']. ↴  
↳count() #89% is residential properties
```

```
[ ]: building_category  
commercial      0.083087
```

```
residential      0.897232  
Name: opa_number, dtype: float64
```

```
[ ]: #converting year_month column to year only  
tax['year_month'] = pd.to_datetime(tax['year_month'], format="%Y%m")  
tax['year'] = pd.DatetimeIndex(tax['year_month']).year
```

```
[ ]: tax['oldest_year_owed'].value_counts()# most of the oldest year owed is from  
→2021
```

```
[ ]: 2021      22224  
2020      9183  
2019      6562  
2018      4716  
2017      3499  
1997      2891  
2016      2653  
2015      1875  
2014      1861  
2010      1312  
2012      1194  
2013      1115  
2011      1037  
2001      1011  
2009      1001  
2008      944  
1978      902  
2007      816  
2006      678  
2005      642  
2003      567  
2004      549  
1998      526  
2002      512  
2000      477  
1999      469  
1996      361  
1995      264  
1992      240  
1994      238  
1989      236  
1988      234  
1990      230  
1993      218  
1991      216  
1986      188  
1987      185
```

```
1984      163
1983      139
1985      132
1982      108
1980       99
1979       97
1981       94
1977        1
Name: oldest_year_owed, dtype: int64
```

```
[ ]: tax['year_of_last_assessment'].value_counts() # most of them were last assessed
      ↪in 2021
```

```
[ ]: 2021.0    70602
2015.0     267
2016.0     102
2017.0      72
2018.0      52
2020.0      40
2019.0      35
2014.0      32
2012.0      26
2013.0      5
Name: year_of_last_assessment, dtype: int64
```

```
[ ]: tax['year'].value_counts()
```

```
[ ]: 2021    72659
Name: year, dtype: int64
```

```
[ ]: #bankruptcy is non-actionable
print(tax.groupby(tax['bankruptcy'])['principal_due'].sum())
print(tax['bankruptcy'].value_counts())
```

```
bankruptcy
False    2.292244e+08
Name: principal_due, dtype: float64
False    72479
Name: bankruptcy, dtype: int64
```

```
[ ]: #payment agreement non-actionable. Payment agreement is one of the way the city
      ↪collect debts
print(tax.groupby(tax['payment_agreement'])['principal_due'].sum())
print(tax['payment_agreement'].value_counts())
```

```
payment_agreement
False    1.482439e+08
True     8.098053e+07
```

```
Name: principal_due, dtype: float64
False    48678
True     23801
Name: payment_agreement, dtype: int64
```

```
[ ]: #sheriff sale is actionable. A sheriff's sale is a public auction where
      ↪mortgage lenders, banks, tax collectors, and other litigants can collect
      ↪money lost on property
print(tax.groupby(tax['sheriff_sale'])['principal_due'].sum())
print(tax['sheriff_sale'].value_counts())
```

```
sheriff_sale
N    2.138628e+08
Y    1.536159e+07
Name: principal_due, dtype: float64
N    70314
Y    2165
Name: sheriff_sale, dtype: int64
```

```
[ ]: # sequestration is actionable
#The taking of someones property, voluntarily (by deposit) or involuntarily (by
      ↪seizure),
# by court officers or into the possession of a third party, awaiting the
      ↪outcome of a trial in which ownership of that property is at issue
#If the delinquent property is a rental property, the City can take over the
      ↪rent collection and apply those rental payments to the delinquent Real
      ↪Estate Tax bill.
print(tax.groupby(tax['sequestration_enforcement'])['principal_due'].sum())
print(tax['sequestration_enforcement'].value_counts())
```

```
sequestration_enforcement
False    2.287520e+08
True     4.724364e+05
Name: principal_due, dtype: float64
False    72381
True      98
Name: sequestration_enforcement, dtype: int64
```

```
[ ]: #The assessment appeal process allows property owners the opportunity to
      ↪dispute the value determined by the Department.
print(tax.groupby(tax['assessment_under_appeal'])['principal_due'].sum())
print(tax['assessment_under_appeal'].value_counts())
#most of the assessment are not under appeal
```

```
assessment_under_appeal
False    2.230154e+08
True     6.208983e+06
Name: principal_due, dtype: float64
```

```
False      71971  
True       508  
Name: assessment_under_appeal, dtype: int64
```

```
[ ]: tax['general_building_description'].unique() #different type of descriptions  
↳ included
```

```
[ ]: array(['house', 'theater_stadium_other amuse', 'vacantLand', 'mixedUsage',  
          'apartmentSmall', 'retail', 'industrial', nan, 'apartmentLarge',  
          'miscCommercial', 'nonProfit', 'parking_garage', 'condo', 'garage',  
          'hotel', 'Restaurant_Bar', 'officeBuilding', 'miscResidential',  
          'parkingLot', 'bank', 'utility'], dtype=object)
```

```
[ ]: tax.groupby(tax['general_building_description'])['principal_due'].sum().  
     ↳sort_values(ascending=False)  
#principal due is also the most for house and vacant land
```

```
[ ]: general_building_description  
house                      1.331961e+08  
vacantLand                  3.026161e+07  
apartmentSmall              1.320392e+07  
mixedUsage                   1.133693e+07  
apartmentLarge              9.104813e+06  
nonProfit                    5.194570e+06  
industrial                   4.105313e+06  
retail                       3.802362e+06  
condo                        3.304997e+06  
miscCommercial               2.440253e+06  
officeBuilding                1.892045e+06  
parkingLot                   1.778640e+06  
theater_stadium_other amuse   1.595614e+06  
Restaurant_Bar                 7.796232e+05  
parking_garage                6.016357e+05  
garage                        5.697422e+05  
hotel                         4.450313e+05  
miscResidential                2.573390e+05  
utility                        5.262612e+04  
bank                           9.603020e+03  
Name: principal_due, dtype: float64
```

```
[ ]: print(tax['general_building_description'].value_counts()) # most of them are  
     ↳houses and vacant lots
```

house	49796
vacantLand	11623
apartmentSmall	3528
mixedUsage	2374
condo	910

```

industrial           560
retail               415
nonProfit            365
apartmentLarge      331
miscCommercial       324
garage                275
parkingLot            235
miscResidential       191
theater_stadium_other amuse    90
officeBuilding        76
Restaurant_Bar         71
hotel                  22
parking_garage          16
bank                   3
utility                 2
Name: general_building_description, dtype: int64

```

```
[ ]: tax.groupby(tax['general_building_description'])['principal_due'].median().
    ↪sort_values(ascending=False)
#however, the median and mean principal due is not high for house and vacant lots
```

```

[ ]: general_building_description
utility           26313.060
officeBuilding     8626.265
apartmentLarge    8141.580
hotel              7714.150
Restaurant_Bar     6149.910
theater_stadium_other amuse 5572.080
nonProfit          4324.050
parking_garage     4072.705
retail              3360.110
miscCommercial      3225.140
bank                2920.940
industrial          2705.660
mixedUsage          2397.415
apartmentSmall     2069.885
condo                1121.630
vacantLand          1101.110
house                1086.830
parkingLot           979.070
garage                911.270
miscResidential      494.970
Name: principal_due, dtype: float64

```

```
[ ]: tax_final = tax[['opa_number','street_address','zip_code','num_years_owed',
    ↪'most_recent_year_owed','oldest_year_owed',
```

```

    'year_of_last_assessment', ↴
    'general_building_description', 'building_category',
    'is_actionable', 'bankruptcy', 'sheriff_sale', 'sequestration_enforcement', ↴
    'payment_agreement',
    'principal_due', 'total_due', 'total_assessment', 'lat', 'lng', ↴
    'geometry'])#selecting only columns that are important and we will be

```

```
tax_final.head()
```

	opa_number	street_address	zip_code	num_years_owed	
0	493169300.0	6045 N CAMAC ST	19141.0		23
1	493179100.0	5620 N CAMAC ST	19141.0		1
2	493180700.0	5714 N CAMAC ST	19141.0		1
3	493183600.0	5812 N CAMAC ST	19141.0		5
4	223166200.0	420 GLEN ECHO RD	19119.0		1
	most_recent_year_owed	oldest_year_owed	year_of_last_assessment		
0		2021	1994		2021.0
1		2021	2021		2021.0
2		2021	2021		2021.0
3		2021	2017		2021.0
4		2021	2021		2021.0
	general_building_description	building_category	is_actionable	bankruptcy	
0	house	residential	False	False	
1	house	residential	False	False	
2	house	residential	False	False	
3	house	residential	False	False	
4	house	residential	False	False	
	sheriff_sale	sequestration_enforcement	payment_agreement	principal_due	
0	N		False	True	12200.18
1	N		False	True	-0.05
2	N		False	False	895.87
3	N		False	True	4536.94
4	N		False	False	4224.60
	total_due	total_assessment	lat	lng	
0	30670.98	111400.0	-75.140099	40.045081	
1	41.00	111200.0	-75.141930	40.039007	
2	1016.81	109000.0	-75.141727	40.039940	
3	6691.70	110600.0	-75.141395	40.041404	
4	4794.92	346800.0	-75.195309	40.051563	
	geometry				
0	POINT (-75.14010 40.04508)				
1	POINT (-75.14193 40.03901)				

```
2 POINT (-75.14173 40.03994)
3 POINT (-75.14140 40.04140)
4 POINT (-75.19531 40.05156)
```

```
[ ]: tax_final.dtypes
```

```
[ ]: opa_number           float64
street_address        object
zip_code              float64
num_years_owed         int64
most_recent_year_owed  int64
oldest_year_owed       int64
year_of_last_assessment float64
general_building_description  object
building_category      object
is_actionable          bool
bankruptcy             bool
sheriff_sale           object
sequestration_enforcement  bool
payment_agreement      bool
principal_due          float64
total_due               float64
total_assessment        float64
lat                     float64
lng                     float64
geometry                geometry
dtype: object
```

```
[ ]: tax_final['general_building_description'].value_counts()
```

```
[ ]: house                  49798
vacantLand              11630
apartmentSmall          3529
mixedUsage                2375
condo                     910
industrial                 561
retail                      415
nonProfit                   365
apartmentLarge            332
miscCommercial              326
garage                      275
parkingLot                  242
miscResidential              191
theater_stadium_other amuse    90
officeBuilding                76
Restaurant_Bar                  71
hotel                      22
```

```
parking_garage           16
bank                   3
utility                 2
Name: general_building_description, dtype: int64
```

```
[ ]: tax_final['building_category'].value_counts()
```

```
[ ]: residential    65192
commercial      6037
Name: building_category, dtype: int64
```

```
[ ]: #tax_final.to_csv("tax_delinquent.csv")
```

0.2.14 City of Philadelphia: Property Code Violations

<https://www.opendataphilly.org/dataset/licenses-and-inspections-violations>

Column description: <https://metadata.phila.gov/#home/datasetdetails/5543ca7a5c4ae4cd66d3ff86/representation>

This dataset was quite big so we had to download three different datasets for different years and combine them together

Property Code Violation(2015 - 2021). The major takeaways are: - There might be some seasonality in code violation - There are around 2000 violation types - Philadelphia Housing Auth and second highest was Philadelphia Land Bank when it comes to Property Assessment ownership - A lot of the violation code titles has vacant lot in the title. - Most of the status were complied

Feature Engineering that occurred using this dataset are: - Create new column called "Vacant_Code" that states if the violation was a vacant code related or not. - Created new column called "Vacant_Code_num" which is based off "Vacant_Code". Where 1 is for Vacant Code and 0 is for Non-Vacant code. - Created casenumber_diff which calculate the number of cases that appeared for each OPA Account Number - Created casecreateddate_year_diff which calculates the the different years that appears for each OPA Account Number - Created violationcode_diff which calculated the number of different violation codes that appears for each OPA Account - Created num_vacant_code which calculated the the number of times vacant related violation code appeared for each OPA Account

```
[ ]: #upload all datasets
violation1 = pd.read_csv('data/city/violations_2019.csv')
violation2 = pd.read_csv('data/city/violations2016-2018.csv')
violation3 = pd.read_csv('data/city/violations2013-2015.csv')

violation = [violation1, violation2, violation3]
violation = pd.concat(violation)#combining all datasets
```

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3457: DtypeWarning: Columns
(2,3,6,10,15,16,20,21) have mixed types. Specify dtype option on import or set
low_memory=False.
```

```

exec(code_obj, self.user_global_ns, self.user_ns)
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3457: DtypeWarning: Columns
(2,3,6,15,16,20) have mixed types.Specify dtype option on import or set
low_memory=False.

exec(code_obj, self.user_global_ns, self.user_ns)
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3457: DtypeWarning: Columns
(3,6,15,16,20) have mixed types.Specify dtype option on import or set
low_memory=False.

exec(code_obj, self.user_global_ns, self.user_ns)

```

[]: violation.head()

	objectid	addressobjectid	parcel_id_num	casenumber	casedatecreated	\
0	22000	156857764.0		NaN	678967	2019-04-05 14:04:19
1	199	15897333.0		475137	568999	2017-01-09 13:23:18
2	20439	131980134.0		NaN	678131	2019-03-29 12:09:52
3	385	15514326.0		326877	569328	2017-01-13 08:41:18
4	386	15514326.0		326877	569328	2017-01-13 08:41:18
	casecompleteddate		casetype	casestatus	\	
0		NaN	NOTICE OF VIOLATION	IN VIOLATION		
1		NaN	NOTICE OF VIOLATION	IN VIOLATION		
2	2020-09-21 12:32:03		NOTICE OF VIOLATION	CLOSED		
3		NaN	NOTICE OF VIOLATION	IN VIOLATION		
4		NaN	NOTICE OF VIOLATION	IN VIOLATION		
	caseresponsibility	caseprioritydesc	...	zip	\	
0	CODE ENFORCEMENT INVESTIGATOR		HAZARDOUS	...	NaN	
1	CSU INVESTIGATOR		UNSAFE	...	19104-1123	
2	BUILDING INVESTIGATOR		STANDARD	...	NaN	
3	BUILDING INVESTIGATOR		STANDARD	...	19138-3051	
4	BUILDING INVESTIGATOR		STANDARD	...	19138-3051	
	censustract	opa_owner	systemofrecord	geocode_x	geocode_y	\
0	NaN	NaN	ECLIPSE	NaN	NaN	
1	110.0	COLEMAN GREGORY	ECLIPSE	2.683078e+06	243246.473027	
2	NaN	NaN	ECLIPSE	NaN	NaN	
3	267.0	JAQUEZ RAMON M	ECLIPSE	2.695877e+06	272847.558740	
4	267.0	JAQUEZ RAMON M	ECLIPSE	2.695877e+06	272847.558740	
	council_district	posse_jobid	lat	lng		
0		NaN	195203950.0	NaN	NaN	
1		3.0	195194705.0	39.972746	-75.200065	
2		NaN	195203407.0	NaN	NaN	
3		9.0	195194755.0	40.052946	-75.151310	

```
4          9.0  195194755.0  40.052946 -75.151310
```

```
[5 rows x 32 columns]
```

```
[ ]: violation.head().T
```

```
[ ]:          0  \
objectid           22000
addressobjectid   156857764.0
parcel_id_num      NaN
casenumber         678967
casecreateddate   2019-04-05 14:04:19
casecompleteddate  NaN
casetype           NOTICE OF VIOLATION
casestatus         IN VIOLATION
caseresponsibility CODE ENFORCEMENT INVESTIGATOR
caseprioritydesc   HAZARDOUS
violationnumber    211959680
violationdate     2019-04-05 00:00:00
violationcode      PM15-301
violationcodetitle VACANT STRUCTURE AND LAND
violationstatus    OPEN
violationresolutiondate  NaN
violationresolutioncode  NaN
mostrecentinvestigation 2020-06-24 12:01:43
opa_account_num    NaN
address            NaN
unit_type          NaN
unit_num           NaN
zip                NaN
censustract        NaN
opa_owner          NaN
systemofrecord     ECLIPSE
geocode_x          NaN
geocode_y          NaN
council_district   NaN
posse_jobid       195203950.0
lat                NaN
lng                NaN

          1  \
objectid           199
addressobjectid   15897333.0
parcel_id_num      475137
casenumber         568999
casecreateddate   2017-01-09 13:23:18
casecompleteddate  NaN
```

	NOTICE OF VIOLATION IN VIOLATION CSU INVESTIGATOR UNSAFE
casetype	
casestatus	
caseresponsibility	
caseprioritydesc	
violationnumber	211935476
violationdate	2019-01-23 00:00:00
violationcode	PM15-304.1G
violationcodetitle	EXTERIOR STRUCT UNSAFE COND 7
violationstatus	OPEN
violationresolutiondate	Nan
violationresolutioncode	Nan
mostrecentinvestigation	2021-09-28 12:01:31
opa_account_num	243186900.0
address	3831 WYALUSING AVE
unit_type	Nan
unit_num	Nan
zip	19104-1123
censustract	110.0
opa_owner	COLEMAN GREGORY
systemofrecord	ECLIPSE
geocode_x	2683077.781269
geocode_y	243246.473027
council_district	3.0
posse_jobid	195194705.0
lat	39.972746
lng	-75.200065

	2 \
objectid	20439
addressobjectid	131980134.0
parcel_id_num	Nan
casenumber	678131
casecreateddate	2019-03-29 12:09:52
casecompleteddate	2020-09-21 12:32:03
casetype	NOTICE OF VIOLATION
casestatus	CLOSED
caseresponsibility	BUILDING INVESTIGATOR
caseprioritydesc	STANDARD
violationnumber	211959511
violationdate	2019-03-27 00:00:00
violationcode	A-303.2/2
violationcodetitle	DEMOL- NOTICE REMOVED TOO SOON
violationstatus	CLOSED
violationresolutiondate	2020-09-18 00:00:00
violationresolutioncode	CLOSED - ADMINISTRATIVELY
mostrecentinvestigation	2020-09-21 12:32:02
opa_account_num	Nan

address		NaN	
unit_type		NaN	
unit_num		NaN	
zip		NaN	
censustract		NaN	
opa_owner		NaN	
systemofrecord		ECLIPSE	
geocode_x		NaN	
geocode_y		NaN	
council_district		NaN	
posse_jobid	195203407.0		
lat		NaN	
lng		NaN	
		3	4
objectid	385		386
addressobjectid	15514326.0		15514326.0
parcel_id_num	326877		326877
casenumber	569328		569328
casecreateddate	2017-01-13 08:41:18		2017-01-13 08:41:18
casecompleteddate	NaN		NaN
casetype	NOTICE OF VIOLATION		NOTICE OF VIOLATION
casestatus	IN VIOLATION		IN VIOLATION
caseresponsibility	BUILDING INVESTIGATOR		BUILDING INVESTIGATOR
caseprioritydesc	STANDARD		STANDARD
violationnumber	211935656		211935657
violationdate	2020-03-14 00:21:37		2020-03-14 00:21:37
violationcode	E-1201.1/217		PM-407.2/4
violationcodetitle	SERVICE HEAD-RAINTIGHT REQ'D	ELEC-CORD DEFECTIVE-RES	
violationstatus	OPEN		OPEN
violationresolutiondate	NaN		NaN
violationresolutioncode	NaN		NaN
mostrecentinvestigation	NaN		NaN
opa_account_num	871515040.0		871515040.0
address	6441 N 20TH ST		6441 N 20TH ST
unit_type	NaN		NaN
unit_num	NaN		NaN
zip	19138-3051		19138-3051
censustract	267.0		267.0
opa_owner	JAQUEZ RAMON M		JAQUEZ RAMON M
systemofrecord	ECLIPSE		ECLIPSE
geocode_x	2695877.169947		2695877.169947
geocode_y	272847.55874		272847.55874
council_district	9.0		9.0
posse_jobid	195194755.0		195194755.0
lat	40.052946		40.052946
lng	-75.15131		-75.15131

```
[ ]: violation.shape# side of dataset
```

```
[ ]: (903633, 32)
```

```
[ ]: violation.columns# different columns included
```

```
[ ]: Index(['objectid', 'addressobjectid', 'parcel_id_num', 'casenumber',
       'casecreateddate', 'casecompleteddate', 'casetype', 'casestatus',
       'caseresponsibility', 'caseprioritydesc', 'violationnumber',
       'violationdate', 'violationcode', 'violationcodetitle',
       'violationstatus', 'violationresolutiondate', 'violationresolutioncode',
       'mostrecentinvestigation', 'opa_account_num', 'address', 'unit_type',
       'unit_num', 'zip', 'censustract', 'opa_owner', 'systemofrecord',
       'geocode_x', 'geocode_y', 'council_district', 'posse_jobid', 'lat',
       'lng'],
      dtype='object')
```

```
[ ]: violation.dtypes #type of dataset
```

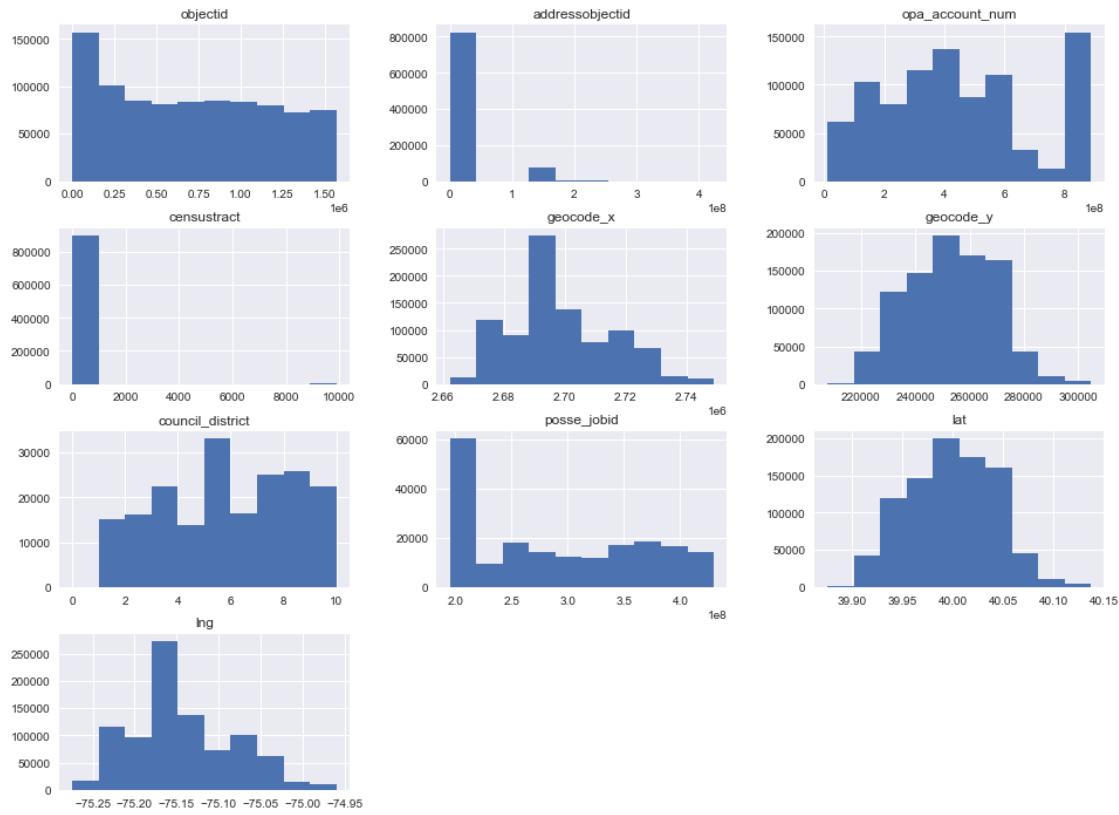
[]:	objectid int64
	addressobjectid float64
	parcel_id_num object
	casenumber object
	casecreateddate object
	casecompleteddate object
	casetype object
	casestatus object
	caseresponsibility object
	caseprioritydesc object
	violationnumber object
	violationdate object
	violationcode object
	violationcodetitle object
	violationstatus object
	violationresolutiondate object
	violationresolutioncode object
	mostrecentinvestigation object
	opa_account_num float64
	address object
	unit_type object
	unit_num object
	zip object
	censustract float64
	opa_owner object
	systemofrecord object
	geocode_x float64
	geocode_y float64

```
council_district           float64
posse_jobid                float64
lat                         float64
lng                         float64
dtype: object
```

```
[ ]: #plotting histogram
fig, ax = plt.subplots(figsize=(16,12))
violation.hist(ax=ax)
```

```
/var/folders/6p/wpw9qml57530xkxqkkhprrf40000gn/T/ipykernel_21232/3257645462.py:3
: UserWarning: To output multiple subplots, the figure containing the passed
axes is being cleared
violation.hist(ax=ax)
```

```
[ ]: array([[<AxesSubplot:title={'center':'objectid'}>,
   <AxesSubplot:title={'center':'addressobjectid'}>,
   <AxesSubplot:title={'center':'opa_account_num'}>],
  [<AxesSubplot:title={'center':'censustract'}>,
   <AxesSubplot:title={'center':'geocode_x'}>,
   <AxesSubplot:title={'center':'geocode_y'}>],
  [<AxesSubplot:title={'center':'council_district'}>,
   <AxesSubplot:title={'center':'posse_jobid'}>,
   <AxesSubplot:title={'center':'lat'}>],
  [<AxesSubplot:title={'center':'lng'}>, <AxesSubplot:>,
   <AxesSubplot:>]], dtype=object)
```



```
[ ]: violation.describe(include = 'all').T# describing dataset
```

	count	unique	\
objectid	903633.0	NaN	
addressobjectid	902429.0	NaN	
parcel_id_num	191017	65323	
casenumber	903633	400338	
casecreateddate	888022	387418	
casecompleteddate	718252	319955	
casetype	192360	5	
casestatus	903633	7	
caseresponsibility	887458	39	
caseprioritydesc	887986	10	
violationnumber	903633	903633	
violationdate	903633	4556	
violationcode	903377	2260	
violationcodetitle	903159	2286	
violationstatus	895070	13	
violationresolutiondate	108619	1376	
violationresolutioncode	108620	14	
mostrecentinvestigation	879738	335052	

opa_account_num	894028.0	NaN
address	902082	163352
unit_type	5827	13
unit_num	11379	613
zip	902075	37162
censustract	901417.0	NaN
opa_owner	896510	132306
systemofrecord	903633	2
geocode_x	901396.0	NaN
geocode_y	901396.0	NaN
council_district	190477.0	NaN
posse_jobid	192575.0	NaN
lat	901396.0	NaN
lng	901396.0	NaN
objectid		top \
addressobjectid		NaN
parcel_id_num		NaN
casenumber		DATA CONVERSION ONLY
casecreateddate		569328
casecompleteddate		2018-07-02 07:43:19
casetype		2016-07-20 08:37:14
casestatus		NOTICE OF VIOLATION
caseresponsibility		CLOSED
caseprioritydesc		CLIP
violationnumber		STANDARD
violationdate		211959680
violationcode		2018-06-21 00:00:00
violationcodetitle		CP-01
violationstatus		CLIP VIOLATION NOTICE
violationresolutiondate		COMPLIED
violationresolutioncode		2021-06-02 00:00:00
mostrecentinvestigation		COMPLIED - OWNER REPAIR
opa_account_num		2018-07-23 00:00:00
address		NaN
unit_type		DATA CONVERSION ONLY 1
unit_num		DATA CONVERSION ONLY MAR...
zip		#
censustract		1
opa_owner		19121-0000
systemofrecord		NaN
geocode_x		PHILADELPHIA HOUSING AUTH
geocode_y		HANSEN
council_district		NaN
posse_jobid		NaN
lat		NaN

lng				NaN
-----	--	--	--	-----

	freq	mean	std	\
objectid	NaN	700253.822956	469527.1076	
addressobjectid	NaN	14387905.594245	39731469.555869	
parcel_id_num	498	NaN	NaN	
casenumber	41	NaN	NaN	
casecreateddate	48	NaN	NaN	
casecompleteddate	8096	NaN	NaN	
casetype	192025	NaN	NaN	
casestatus	803203	NaN	NaN	
caseresponsibility	314813	NaN	NaN	
caseprioritydesc	745654	NaN	NaN	
violationnumber	1	NaN	NaN	
violationdate	823	NaN	NaN	
violationcode	133117	NaN	NaN	
violationcodetitle	133117	NaN	NaN	
violationstatus	699572	NaN	NaN	
violationresolutiondate	459	NaN	NaN	
violationresolutioncode	55298	NaN	NaN	
mostrecentinvestigation	147	NaN	NaN	
opa_account_num	NaN	449880985.849242	258958681.014763	
address	498	NaN	NaN	
unit_type	3885	NaN	NaN	
unit_num	1751	NaN	NaN	
zip	3282	NaN	NaN	
censustract	NaN	225.867819	540.346234	
opa_owner	13011	NaN	NaN	
systemofrecord	711058	NaN	NaN	
geocode_x	NaN	2698469.028863	16675.065162	
geocode_y	NaN	253109.841246	15919.340083	
council_district	NaN	5.381343	2.542162	
posse_jobid	NaN	287458400.794828	79257460.572387	
lat	NaN	39.998561	0.042816	
lng	NaN	-75.144111	0.060601	

	min	25%	50%	\
objectid	1.0	255821.0	680843.0	
addressobjectid	1038.0	322714.0	514315.0	
parcel_id_num	NaN	NaN	NaN	
casenumber	NaN	NaN	NaN	
casecreateddate	NaN	NaN	NaN	
casecompleteddate	NaN	NaN	NaN	
casetype	NaN	NaN	NaN	
casestatus	NaN	NaN	NaN	
caseresponsibility	NaN	NaN	NaN	
caseprioritydesc	NaN	NaN	NaN	

violationnumber		NaN	NaN	NaN
violationdate		NaN	NaN	NaN
violationcode		NaN	NaN	NaN
violationcodetitle		NaN	NaN	NaN
violationstatus		NaN	NaN	NaN
violationresolutiondate		NaN	NaN	NaN
violationresolutioncode		NaN	NaN	NaN
mostrecentinvestigation		NaN	NaN	NaN
opa_account_num	11004900.0	243207800.0	412025650.0	
address		NaN	NaN	NaN
unit_type		NaN	NaN	NaN
unit_num		NaN	NaN	NaN
zip		NaN	NaN	NaN
censustract	1.0	108.0	179.0	
opa_owner		NaN	NaN	NaN
systemofrecord		NaN	NaN	NaN
geocode_x	2662255.079664	2688400.696487	2695435.196684	
geocode_y	207591.375288	241379.31152	252487.560163	
council_district	0.0	3.0	5.0	
posse_jobid	195194617.0	195213867.0	279522734.0	
lat	39.875131	39.967756	39.996909	
lng	-75.274275	-75.180733	-75.155205	
	75%	max		
objectid	1100259.0	1567618.0		
addressobjectid	693697.0	423909631.0		
parcel_id_num		NaN	NaN	
casenumber		NaN	NaN	
casecreateddate		NaN	NaN	
casecompleteddate		NaN	NaN	
casetype		NaN	NaN	
casestatus		NaN	NaN	
caseresponsibility		NaN	NaN	
caseprioritydesc		NaN	NaN	
violationnumber		NaN	NaN	
violationdate		NaN	NaN	
violationcode		NaN	NaN	
violationcodetitle		NaN	NaN	
violationstatus		NaN	NaN	
violationresolutiondate		NaN	NaN	
violationresolutioncode		NaN	NaN	
mostrecentinvestigation		NaN	NaN	
opa_account_num	621072400.0	888800162.0		
address		NaN	NaN	
unit_type		NaN	NaN	
unit_num		NaN	NaN	
zip		NaN	NaN	

```

censustract          298.0      9891.0
opa_owner            NaN        NaN
systemofrecord       NaN        NaN
geocode_x            2709493.522028 2748942.42556
geocode_y            265671.390854 304755.168331
council_district     8.0        10.0
posse_jobid          361314020.0 429848513.0
lat                  40.031826  40.137402
lng                  -75.104109 -74.959958

```

```
[ ]: violation.isna().sum()# null values
```

```

[ ]: objectid          0
addressobjectid      1204
parcel_id_num         712616
casenumber            0
casecreateddate       15611
casecompleteddate    185381
casetype              711273
casestatus            0
caseresponsibility   16175
caseprioritydesc     15647
violationnumber       0
violationdate         0
violationcode         256
violationcodetitle   474
violationstatus       8563
violationresolutiondate 795014
violationresolutioncode 795013
mostrecentinvestigation 23895
opa_account_num       9605
address               1551
unit_type              897806
unit_num               892254
zip                   1558
censustract           2216
opa_owner              7123
systemofrecord          0
geocode_x              2237
geocode_y              2237
council_district        713156
posse_jobid             711058
lat                     2237
lng                     2237
dtype: int64

```

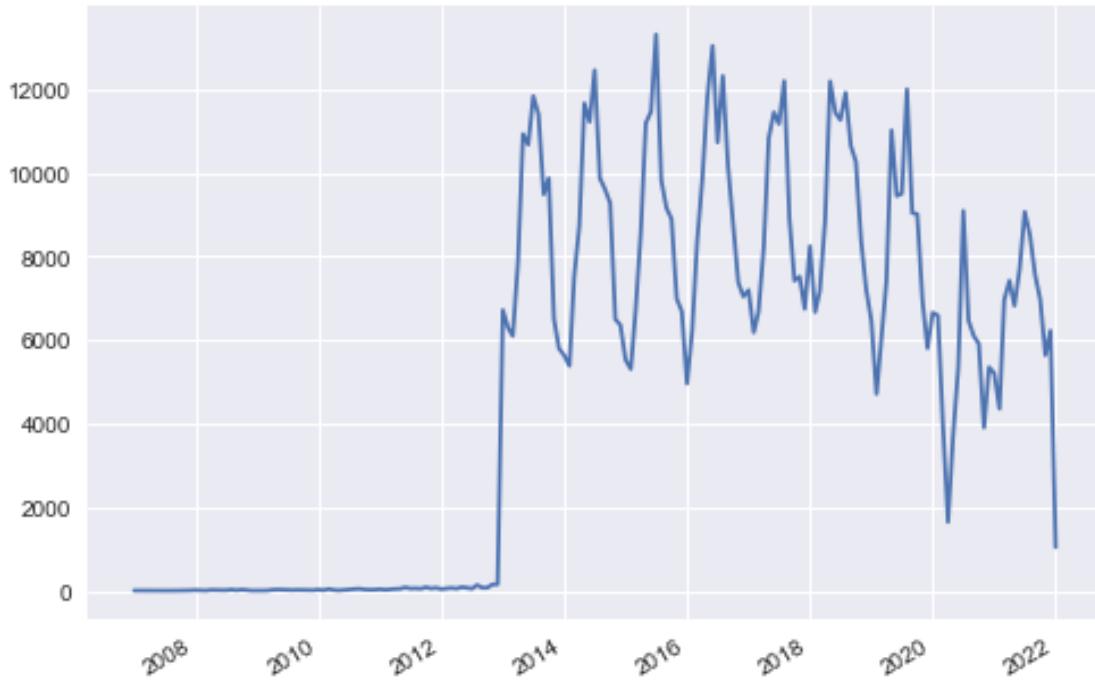
```
[ ]: (violation.isna().sum()/violation.shape[0]).sort_values(ascending=False) #unit type, unit num, has a lot of null values
```

```
[ ]: unit_type          0.993552
unit_num            0.987407
violationresolutiondate  0.879797
violationresolutioncode 0.879796
council_district     0.789210
parcel_id_num        0.788612
casetype             0.787126
posse_jobid          0.786888
casecompleteddate    0.205151
mostrecentinvestigation 0.026443
caseresponsibility   0.017900
caseprioritydesc     0.017316
casecreateddate      0.017276
opa_account_num      0.010629
violationstatus      0.009476
opa_owner             0.007883
geocode_x             0.002476
geocode_y             0.002476
lat                   0.002476
lng                   0.002476
censustract           0.002452
zip                  0.001724
address              0.001716
addressobjectid       0.001332
violationcodetitle   0.000525
violationcode          0.000283
systemofrecord         0.000000
violationdate          0.000000
violationnumber         0.000000
casestatus            0.000000
casenumber             0.000000
objectid              0.000000
dtype: float64
```

```
[ ]: #violation.dropna(subset=['lat', 'lng', 'zip'], inplace = True) #dropping null values within latitude, longitude and zip code data
```

```
[ ]: #converting date columns to datetime
violation['casecreateddate'] = pd.to_datetime(violation['casecreateddate'])
violation['casecompleteddate'] = pd.to_datetime(violation['casecompleteddate'])
violation['violationdate'] = pd.to_datetime(violation['violationdate'])
violation['violationresolutiondate'] = pd.to_datetime(violation['violationresolutiondate'])
```

```
[ ]: violation['casecreateddate_monthyear'] = pd.  
    ↪to_datetime(violation['casecreateddate'].dt.strftime('%Y-%m'))  
violation['casecreateddate_year'] = pd.to_datetime(violation['casecreateddate'].  
    ↪dt.strftime('%Y'))  
  
[ ]: violation['casecreateddate_monthyear'].value_counts().plot() #there seems to be  
    ↪some seasonality in case violations  
  
[ ]: <AxesSubplot:>
```



```
[ ]: #filtering for cases that is between 2015 and 2021 only  
violation = violation.loc[(violation['casecreateddate_year'] > '2014-01-01') &  
    ↪(violation['casecreateddate_year'] < '2021-01-01')].reset_index(drop = True)  
  
[ ]: violation.head()
```

	objectid	addressobjectid	parcel_id_num	casenumber	casecreateddate	\
0	22000	156857764.0		NaN	678967	2019-04-05 14:04:19
1	199	15897333.0		475137	568999	2017-01-09 13:23:18
2	20439	131980134.0		NaN	678131	2019-03-29 12:09:52
3	385	15514326.0		326877	569328	2017-01-13 08:41:18
4	386	15514326.0		326877	569328	2017-01-13 08:41:18

	casecompleteddate	casetype	casestatus	\
--	-------------------	----------	------------	---

```

0          NaT  NOTICE OF VIOLATION  IN VIOLATION
1          NaT  NOTICE OF VIOLATION  IN VIOLATION
2  2020-09-21 12:32:03  NOTICE OF VIOLATION      CLOSED
3          NaT  NOTICE OF VIOLATION  IN VIOLATION
4          NaT  NOTICE OF VIOLATION  IN VIOLATION

           caseresponsibility caseprioritydesc ...      opa_owner \
0  CODE ENFORCEMENT INVESTIGATOR        HAZARDOUS ...        NaN
1          CSU INVESTIGATOR        UNSAFE ...  COLEMAN GREGORY
2  BUILDING INVESTIGATOR        STANDARD ...        NaN
3  BUILDING INVESTIGATOR        STANDARD ...  JAQUEZ RAMON M
4  BUILDING INVESTIGATOR        STANDARD ...  JAQUEZ RAMON M

       systemofrecord      geocode_x      geocode_y council_district posse_jobid \
0      ECLIPSE          NaN          NaN          NaN  195203950.0
1      ECLIPSE  2.683078e+06  243246.473027        3.0  195194705.0
2      ECLIPSE          NaN          NaN          NaN  195203407.0
3      ECLIPSE  2.695877e+06  272847.558740        9.0  195194755.0
4      ECLIPSE  2.695877e+06  272847.558740        9.0  195194755.0

      lat      lng  casecreateddate_monthyear casecreateddate_year
0  NaN      NaN            2019-04-01        2019-01-01
1  39.972746 -75.200065            2017-01-01        2017-01-01
2  NaN      NaN            2019-03-01        2019-01-01
3  40.052946 -75.151310            2017-01-01        2017-01-01
4  40.052946 -75.151310            2017-01-01        2017-01-01

```

[5 rows x 34 columns]

```
[ ]: violation['caseprioritydesc'].value_counts()# most of the cases are standard
```

```

[ ]: STANDARD                  496546
CONSTRUCTION SERVICES          35442
UNSAFE                         33941
HAZARDOUS                      17447
IMMINENTLY DANGEROUS          11295
ACCELERATED REVIEW              27
UNLAWFUL                        21
AIU LICENSING VIOLATION NOTICE    11
UNFIT                           1
5 DAY REVIEW GROUP              1
Name: caseprioritydesc, dtype: int64

```

```
[ ]: violation.shape
```

```
[ ]: (594752, 34)
```

```
[ ]: violation['opa_owner'].value_counts().head(20)# opa_owner is Office of Property  

→Assessment's ownership from the current deed for the property.  

#Most of these were from Philadelphia housing Auth and second highest was  

→philadelphia land bank
```

```
[ ]: PHILADELPHIA HOUSING AUTH 7430  

    PHILADELPHIA LAND BANK 3795  

    REDEVELOPMENT AUTHORITY OF PHILADELPHIA 2824  

    SCHOOL DISTRICT OF PHILA 2394  

    CITY OF PHILADELPHIA 1335  

    REDEVELOPMENT AUTHORITY OF PHILA 1105  

    GEENA LLC 847  

    REDEVELOPMENT AUTHORITY O 690  

    CITY OF PHILA 668  

    EMARCO DREW 492  

    STABLE GENIUS LLC 457  

    REDEVELOPMENT AUTHORITY, OF PHILADELPHIA 431  

    GULLE JEAN PAUL 403  

    PHILADELPHIA REDEVELOPMEN 381  

    ULATOWSKI WALTER 359  

    BID PROPERTIES LLC 356  

    CORESTATES GROUP LLC 350  

    PHILADELPHIA LAND INVESTM 340  

    SOILED LLC 337  

    TTP HOLDINGS LLC 321  

Name: opa_owner, dtype: int64
```

```
[ ]: violation['opa_owner'].nunique()# number of unique opa_owner
```

```
[ ]: 100765
```

```
[ ]: (violation.groupby('violationcodetitle')['objectid'].count())  

→sort_values(ascending=False).reset_index().head(20)  

#included description of the violations. A lot of them are related to vacant  

→lots
```

```
[ ]: violationcodetitle objectid  

0 CLIP VIOLATION NOTICE 102081  

1 EXT A-VACANT LOT CLEAN/MAINTAI 41030  

2 HIGH WEEDS-CUT 39964  

3 RUBBISH/GARBAGE EXTERIOR-OWNER 24250  

4 VACANT STRUCTURE LICENSE 20208  

5 EXTERIOR AREA WEEDS 19969  

6 EXTERIOR AREA SANITATION 13660  

7 UNSAFE STRUCTURE 11725  

8 LICENSE - RENTAL PROPERTY 9668  

9 VACANT STRUCTURE AND LAND 9016
```

```

10          VACANT AND OPEN      9014
11  EXTERIOR STRUCT UNSAFE COND 7      8060
12          INTERIOR SURFACES    7744
13          ONE AND TWO FAMILY (R3) 6738
14  EXTERIOR STRUCT UNSAFE COND 8      5623
15          PERM Z- NEW USE      5238
16  EXTERIOR STRUCT ROOF DRAINAGE    4989
17  ARCHITECT/ENGINEER SERVICES     4447
18          RUBBISH AND GARBAGE   4261
19          VACANT STRUCTURE & LAND 3990

```

```
[ ]: violation['violationcodetitle'].nunique() # around 2 thoughtsad violation types
```

```
[ ]: 1950
```

```
[ ]: violation['violationstatus'].value_counts() # most of the status of violation
→are complied
```

```

[ ]: COMPLIED           471214
CLOSEDCASE           51566
OPEN                46111
CLOSED              8041
DEMOLISH            6025
ERROR               3949
RESOLVE             2515
CVN ISSUED          779
STOP WORK           86
WARNING ISSUED      16
SVN ISSUED          1
Name: violationstatus, dtype: int64

```

```
[ ]: violation.columns # column names in the dataset
```

```

[ ]: Index(['objectid', 'addressobjectid', 'parcel_id_num', 'casenumber',
       'casetreateddate', 'casecompleteddate', 'casetype', 'casestatus',
       'caseresponsibility', 'caseprioritydesc', 'violationnumber',
       'violationdate', 'violationcode', 'violationcodetitle',
       'violationstatus', 'violationresolutiondate', 'violationresolutioncode',
       'mostrecentinvestigation', 'opa_account_num', 'address', 'unit_type',
       'unit_num', 'zip', 'censustract', 'opa_owner', 'systemofrecord',
       'geocode_x', 'geocode_y', 'council_district', 'posse_jobid', 'lat',
       'lng', 'casetreateddate_monthyear', 'casetreateddate_year'],
      dtype='object')

```

```

[ ]: #combining latitude and longitude columns into geometry column for geo pandas to
→read and make maps
crs = {'init': 'epsg:4326'}

```

```

geometry = [Point(xy) for xy in zip(violation["lng"], violation["lat"])]
violation = gpd.GeoDataFrame(violation,
                             crs = crs,
                             geometry = geometry)

violation.head()

```

```

/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/pyproj/crs/crs.py:131: FutureWarning: '+init=<authority>:<code>' syntax
is deprecated. '<authority>:<code>' is the preferred initialization method. When
making the change, be mindful of axis order changes:
https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-
proj-6
    in_crs_string = _prepare_from_proj_string(in_crs_string)

```

	objectid	addressobjectid	parcel_id_num	casenumber	casedate	casecreateddate	casestatus	casetype	caserecord	caseresponsibility	caseprioritydesc	... systemofrecord		
0	22000	156857764.0		NaN	678967	2019-04-05 14:04:19				CODE ENFORCEMENT INVESTIGATOR	HAZARDOUS	...	ECLIPSE	
1	199	15897333.0		475137	568999	2017-01-09 13:23:18				CSU INVESTIGATOR	UNSAFE	...	ECLIPSE	
2	20439	131980134.0		NaN	678131	2019-03-29 12:09:52				BUILDING INVESTIGATOR	STANDARD	...	ECLIPSE	
3	385	15514326.0		326877	569328	2017-01-13 08:41:18				BUILDING INVESTIGATOR	STANDARD	...	ECLIPSE	
4	386	15514326.0		326877	569328	2017-01-13 08:41:18				BUILDING INVESTIGATOR	STANDARD	...	ECLIPSE	
										geocode_x	geocode_y	council_district	posse_jobid	lat
0		NaN		NaN		195203950.0				NaN	243246.473027			NaN
1	2.683078e+06		243246.473027		3.0	195194705.0	39.972746			NaN				
2		NaN		NaN		195203407.0				NaN				
3	2.695877e+06		272847.558740		9.0	195194755.0	40.052946			NaN				
4	2.695877e+06		272847.558740		9.0	195194755.0	40.052946			NaN				
										lng	caserecord	caseprioritydesc	... systemofrecord	caserecord
0		NaN		2019-04-01		2019-01-01				NaN	2019-04-01			2019-01-01
1	-75.200065			2017-01-01		2017-01-01				NaN	2017-01-01			2017-01-01
2		NaN		2019-03-01		2019-01-01				NaN	2019-03-01			2019-01-01

```
3 -75.151310          2017-01-01          2017-01-01
4 -75.151310          2017-01-01          2017-01-01
```

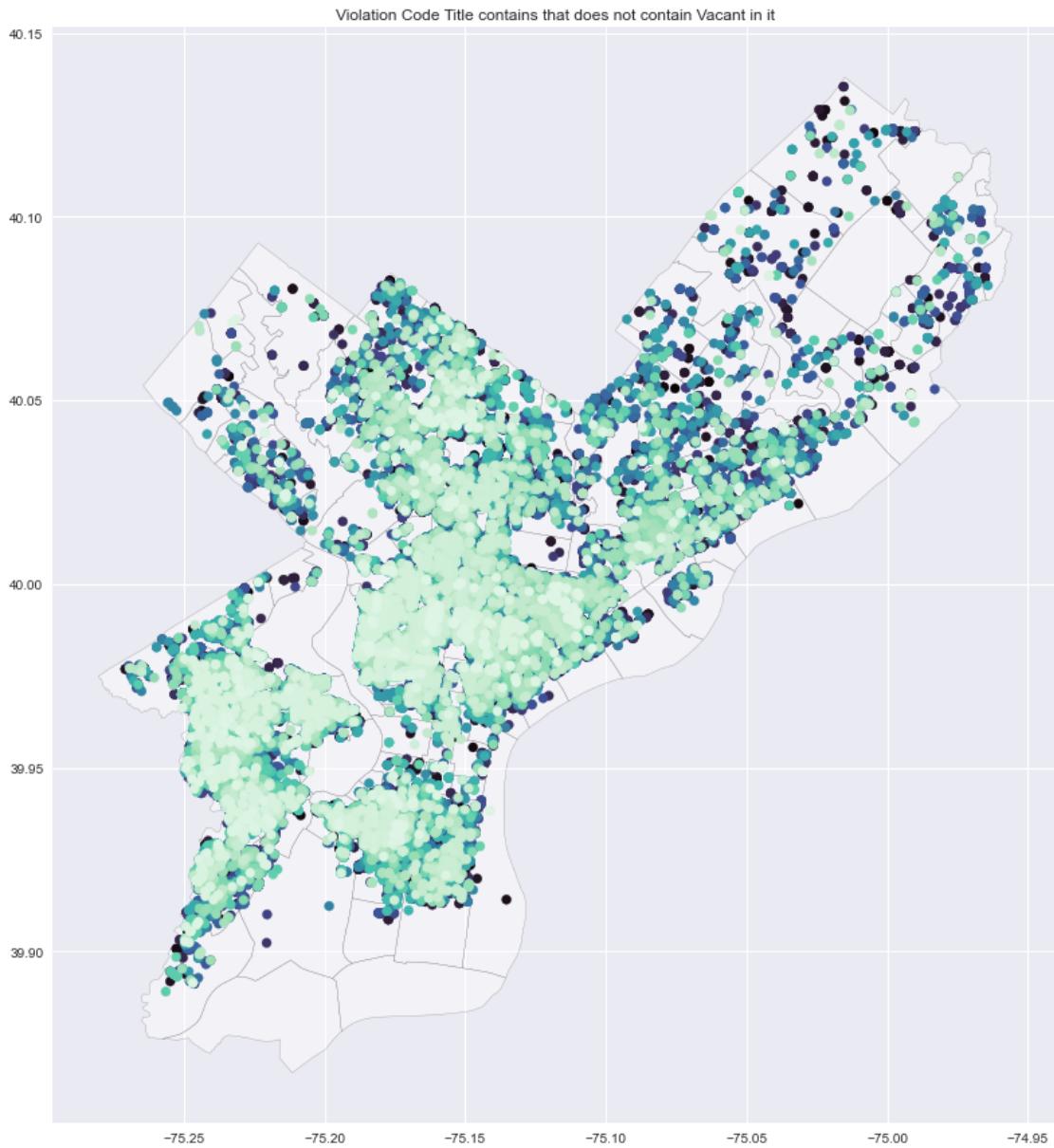
```
      geometry
0      POINT EMPTY
1  POINT (-75.20007 39.97275)
2      POINT EMPTY
3  POINT (-75.15131 40.05295)
4  POINT (-75.15131 40.05295)
```

```
[5 rows x 35 columns]
```

```
[ ]: #creating map that does not include vacant in the code violation title
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
plt.title("Violation Code Title contains that does not contain Vacant in it")
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white",
                                     edgecolor='black')
violation[violation['violationcodetitle'].str.contains('VACANT',na=False)].
    plot(ax = ax, cmap = 'mako', label = "Violation does not cotain Vacant")

#the map shows that the violations are very different from what we see where
#vacant lots tend to be
```

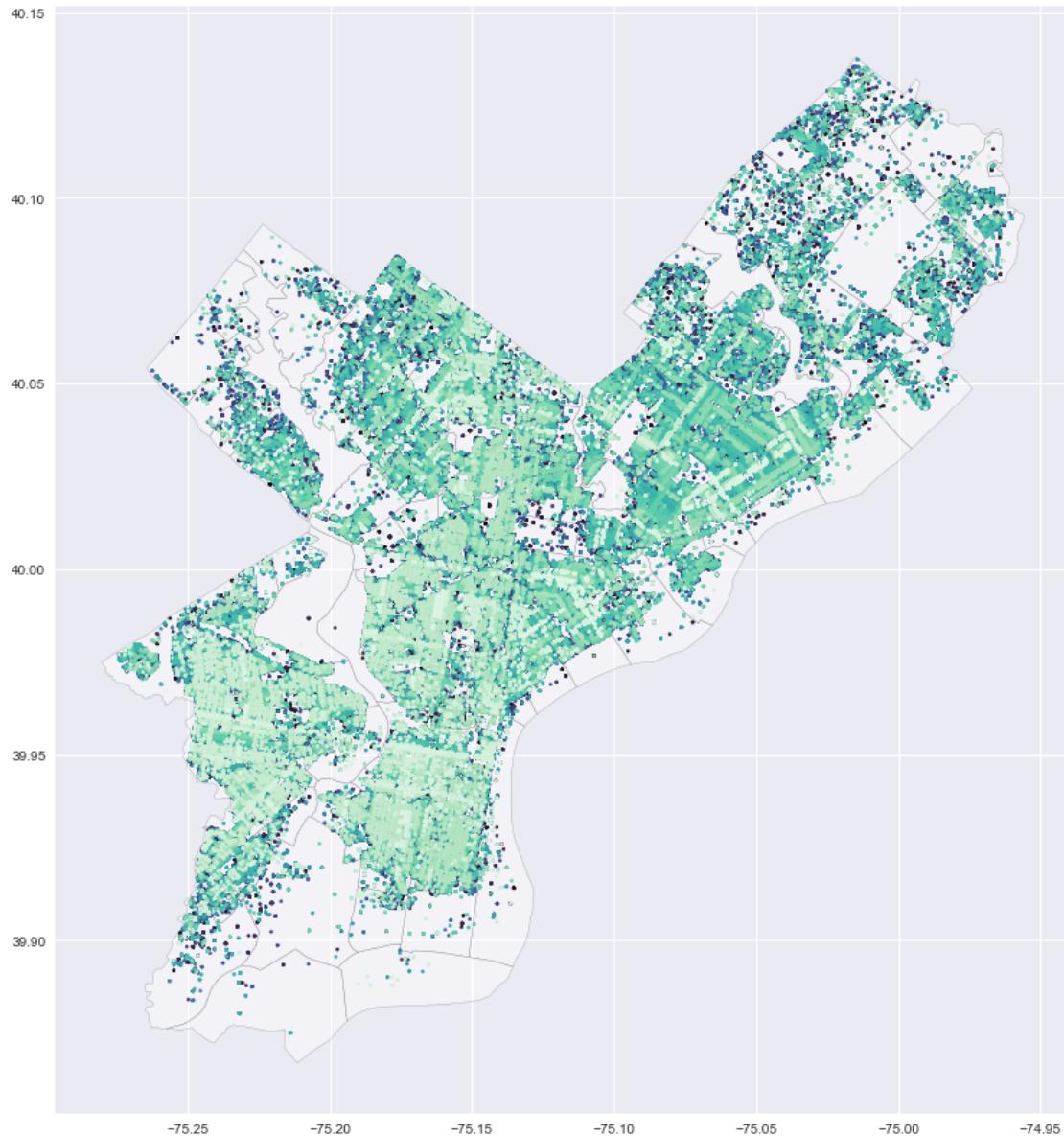
```
[ ]: <AxesSubplot:title={'center':'Violation Code Title contains that does not
contain Vacant in it'}>
```



```
[ ]: #creating map that does not include vacant in the code violation title
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
plt.title("Violation Code Title contains that does not contain Vacant in it")
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white",
edgecolor='black')
violation[~violation['violationcodetitle'].str.contains('VACANT',na=False)].
plot(ax = ax, cmap = 'mako',legend=True, markersize = 5, label = "Violation
does not cotain Vacant")
```

```
#the map shows that the violations are very different from what we see where  
→vacant lots tend to be
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: def plot_point_distribution(**kwargs):  
    ax = plt.gca()  
    context.plot(ax = ax, alpha = 0.4, color = "white", edgecolor='black')  
    color = kwargs.pop('color')  
    geodf = kwargs.pop('data')  
    bounds = kwargs.pop('bounds', None)
```

```

sns.kdeplot(geodf.geometry.x, geodf.geometry.y, **kwargs)

if bounds is not None:
    plt.xlim(bounds[:,0])
    plt.ylim(bounds[:,1])

plt.axis('equal')
plt.axis('off')

def plot_heat_map(data, label, context):
    g = sns.FacetGrid(data=data,
                       height=15,
                       dropna=False
                      )

    g.map_dataframe(plot_point_distribution,
                    shade=True,
                    alpha=0.7,
                    cbar=True,
                    cmap='magma_r',
                    cbar_kws={'orientation': 'horizontal',
                               'label': label,
                               'fraction': 0.02,
                               'shrink': 0.35,
                               'pad': 0}
                   )

```

```

[ ]: crashes= violation[~violation['violationcodetitle'].str.
    ~contains('VACANT',na=False)].reset_index(drop=True)
context = street_map.to_crs(epsg = 4326)
plot_heat_map(crashes, 'Number of Violations with the word VACANT', context)

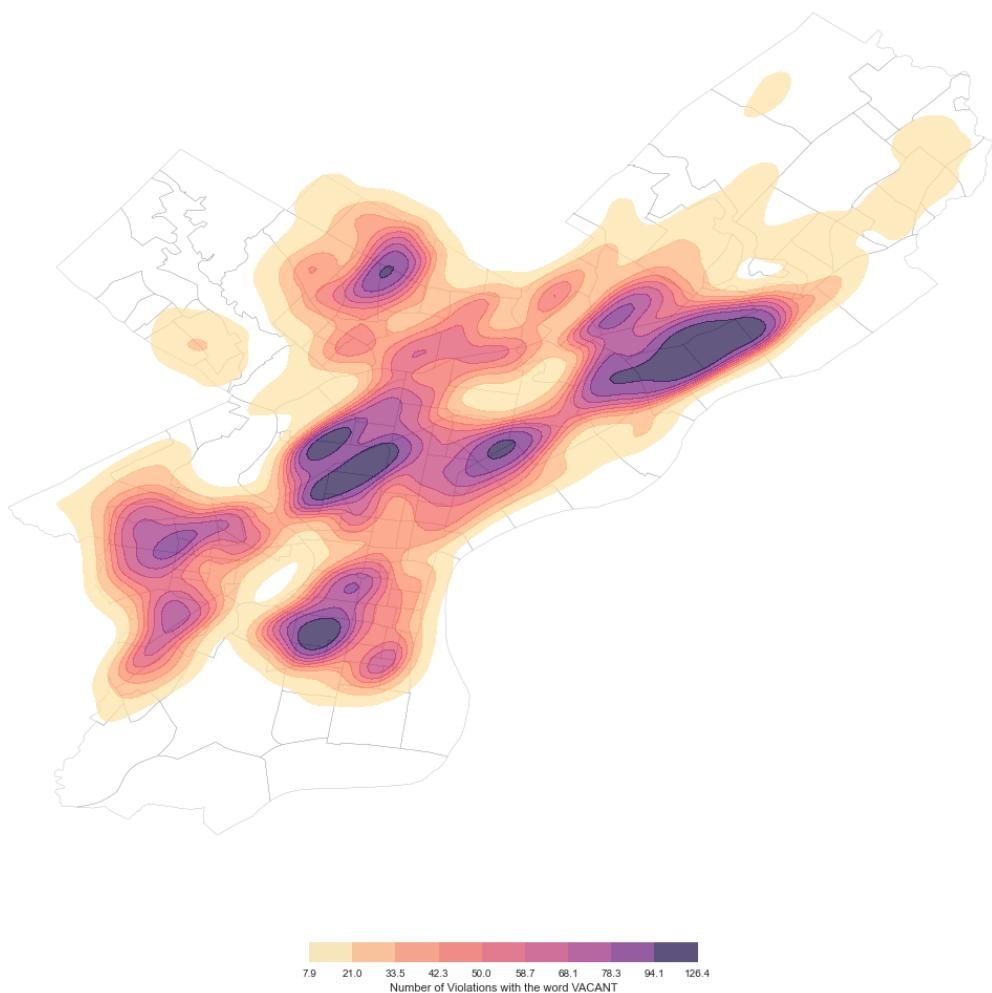
```

/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```

warnings.warn(

```



```
[ ]: (violation.isna().sum()/violation.shape[0]).sort_values(ascending=False)
```

```
[ ]: unit_type          0.991529  
unit_num            0.986959  
violationresolutioncode 0.894005  
violationresolutiondate 0.894005  
council_district     0.819024  
parcel_id_num        0.818170  
casetype             0.816608  
posse_jobid          0.816475  
casecompleteddate    0.160815  
mostrecentinvestigation 0.039778  
opa_account_num      0.008757
```

```
violationstatus          0.007480
opa_owner                0.006149
geocode_x                 0.002700
geocode_y                 0.002700
lat                        0.002700
lng                        0.002700
censustract               0.002692
address                     0.001666
zip                         0.001666
addressobjectid            0.001286
caseresponsibility          0.000933
violationcodetitle          0.000471
violationcode               0.000156
caseprioritydesc             0.000034
casecreateddate_monthyear    0.000000
casecreateddate_year           0.000000
objectid                     0.000000
systemofrecord               0.000000
violationdate                0.000000
violationnumber               0.000000
casestatus                     0.000000
casecreateddate                0.000000
casenumber                     0.000000
geometry                      0.000000
dtype: float64
```

```
[ ]: violation.columns
```

```
[ ]: Index(['objectid', 'addressobjectid', 'parcel_id_num', 'casenumber',
       'casecreateddate', 'casecompleteddate', 'casetype', 'casestatus',
       'caseresponsibility', 'caseprioritydesc', 'violationnumber',
       'violationdate', 'violationcode', 'violationcodetitle',
       'violationstatus', 'violationresolutiondate', 'violationresolutioncode',
       'mostrecentinvestigation', 'opa_account_num', 'address', 'unit_type',
       'unit_num', 'zip', 'censustract', 'opa_owner', 'systemofrecord',
       'geocode_x', 'geocode_y', 'council_district', 'posse_jobid', 'lat',
       'lng', 'casecreateddate_monthyear', 'casecreateddate_year', 'geometry'],
      dtype='object')
```

```
[ ]: violation_final = violation.drop(['unit_type', 'unit_num', 'violationresolutioncode', 'violationresolutiondate', 'council_district', 'parcel_id_num', 'casetype', 'posse_jobid', 'casecompleteddate', 'casecreateddate_monthyear', 'geocode_x', 'geocode_y', 'objectid', 'addressobjectid',
```

```

        'caseresponsibility', u
    ↵'violationnumber', 'violationdate', u
    ↵'mostrecentinvestigation', 'systemofrecord',
          'casestatus', 'violationstatus', u
    ↵'caserecreateddate'], axis = 1)
#casenumber is unique
#everything will be joined by 'opa_account_num'
violation_final.head()

```

```

[ ]: casenumber caseprioritydesc violationcode           violationcodetitle \
0   678967      HAZARDOUS     PM15-301      VACANT STRUCTURE AND LAND
1   568999      UNSAFE       PM15-304.1G    EXTERIOR STRUCT UNSAFE COND 7
2   678131      STANDARD     A-303.2/2    DEMOL- NOTICE REMOVED TOO SOON
3   569328      STANDARD     E-1201.1/217   SERVICE HEAD-RAINTIGHT REQ'D
4   569328      STANDARD     PM-407.2/4    ELEC-CORD DEFECTIVE-RES

      opa_account_num      address      zip censustract \
0           NaN            NaN      NaN      NaN
1  243186900.0  3831 WYALUSING AVE  19104-1123    110.0
2           NaN            NaN      NaN      NaN
3  871515040.0   6441 N 20TH ST  19138-3051    267.0
4  871515040.0   6441 N 20TH ST  19138-3051    267.0

      opa_owner      lat      lng caserecreateddate_year \
0           NaN      NaN      NaN      2019-01-01
1 COLEMAN GREGORY  39.972746 -75.200065    2017-01-01
2           NaN      NaN      NaN      2019-01-01
3 JAQUEZ RAMON M  40.052946 -75.151310    2017-01-01
4 JAQUEZ RAMON M  40.052946 -75.151310    2017-01-01

      geometry
0      POINT EMPTY
1  POINT (-75.20007 39.97275)
2      POINT EMPTY
3  POINT (-75.15131 40.05295)
4  POINT (-75.15131 40.05295)

```

```
[ ]: violation['casenumber'].nunique()
```

```
[ ]: 260500
```

```
[ ]: violation['opa_account_num'].nunique()
```

```
[ ]: 122327
```

```
[ ]: violation['objectid'].nunique()
```

```
[ ]: 572237
[ ]: violation['violationnumber'].nunique()
[ ]: 246304
[ ]: violation_final.columns
[ ]: Index(['casenumber', 'caseprioritydesc', 'violationcode', 'violationcodetitle',
       'opa_account_num', 'address', 'zip', 'censustract', 'opa_owner', 'lat',
       'lng', 'casecreateddate_year', 'geometry'],
       dtype='object')
[ ]: violation_final.shape
[ ]: (594752, 13)
[ ]: violation_final['opa_account_num'].nunique()#as the number of unique opa number
    ↪is less than
    #the violations mean that one location is getting multiple violations
[ ]: 122327
[ ]: #violation_final.to_csv("violation_final.csv")
[ ]: #create new columns to label if the violation was vacany related or not
violation_final['Vacant_Code'] = np.where(violation_final['violationcodetitle'].str.contains('VACANT', regex=False, na = False), 'Vacant_Code', 'Not_Vacant_Code')
violation_final['Vacant_Code_num'] = np.where(violation_final['violationcodetitle'].str.contains('VACANT', regex=False, na = False), 1, 0)
violation_final.head()
[ ]: casenumber caseprioritydesc violationcode violationcodetitle \
0 678967 HAZARDOUS PM15-301 VACANT STRUCTURE AND LAND
1 568999 UNSAFE PM15-304.1G EXTERIOR STRUCT UNSAFE COND 7
2 678131 STANDARD A-303.2/2 DEMOL- NOTICE REMOVED TOO SOON
3 569328 STANDARD E-1201.1/217 SERVICE HEAD-RAINTIGHT REQ'D
4 569328 STANDARD PM-407.2/4 ELEC-CORD DEFECTIVE-RES
[ ]: opa_account_num address zip censustract \
0      NaN      NaN  NaN      NaN
1 243186900.0 3831 WYALUSING AVE 19104-1123 110.0
2      NaN      NaN  NaN      NaN
3 871515040.0   6441 N 20TH ST 19138-3051 267.0
4 871515040.0   6441 N 20TH ST 19138-3051 267.0
```

	opa_owner	lat	lng	casecreateddate_year	\
0	NaN	NaN	NaN	2019-01-01	
1	COLEMAN GREGORY	39.972746	-75.200065	2017-01-01	
2	NaN	NaN	NaN	2019-01-01	
3	JAQUEZ RAMON M	40.052946	-75.151310	2017-01-01	
4	JAQUEZ RAMON M	40.052946	-75.151310	2017-01-01	

	geometry	Vacant_Code	Vacant_Code_num
0	POINT EMPTY	Vacant_Code	1
1	POINT (-75.20007 39.97275)	Not_Vacant_Code	0
2	POINT EMPTY	Not_Vacant_Code	0
3	POINT (-75.15131 40.05295)	Not_Vacant_Code	0
4	POINT (-75.15131 40.05295)	Not_Vacant_Code	0

```
[ ]: #as OPA account number appeared multiple times, we have to group by opa account number and calculate some of the important metrics.
# OPA account will be used later to join datasets
violation_final1 = (violation_final.groupby('opa_account_num')
                     .agg({'casenumber':'nunique', #calculated unique number of casenumbers
                           'casecreateddate_year': 'nunique', # calculated unique number of years
                           'violationcode': 'nunique', #calculated unique number of violation code
                           'Vacant_Code_num': 'sum'}).sort_values(ascending=False, by='violationcode')#calculate number of vacancy related violation
                     .reset_index()
                     .rename(columns={'casenumber':'casenumber_diff',
                                     'casecreateddate_year':'casecreateddate_year_diff',
                                     'violationcode':'violationcode_diff',
                                     'Vacant_Code_num':'num_vacant_code'}))

violation_final1.head()
```

	opa_account_num	casenumber_diff	casecreateddate_year_diff	\
0	881126002.0	34	6	
1	884202000.0	8	3	
2	881074500.0	64	6	
3	884114060.0	9	2	
4	881041000.0	11	6	

	violationcode_diff	num_vacant_code
0	94	0
1	73	1
2	65	0
3	61	0

```
4
```

```
60
```

```
0
```

```
[ ]: #violation_final1.to_csv('violation_final.csv')
```

```
[ ]: violation_final1['opa_account_num'].nunique()
```

```
[ ]: 122327
```

0.2.15 City of Philadelphia: Crime

<https://metadata.phila.gov/#home/datasetdetails/5543868920583086178c4f8e/representationdetails/570e7621c03>

Crime(2015 - 2021). Key takeaways are: - Most fo the crime are all other offenses, assaults is the second highest - After grouping the number of crimes that happened within 50m of each parcel number, we noticed that the data is quite skewed.

Feature Engineering that in this dataset are: - Dropped null values in lat and lng columns as this data was very important for us. - Removed lat and lng data that was outside the Philadelphia lat lng range. - Calculate number of crimes that happened within 50m of each parcel number in the last 6 months (Jul 2021 - Dec 2021) - Calculate number of crimes with 50m of each parcel number in the last 3.5 years(Jul 2018 - Dec 2021) - Calculate number of crimes that happened within 50m of each parcel number using all of the dataset(2015- 2021)

```
[ ]: crime = pd.read_csv('data/city/crime.csv')
```

```
[ ]: crime.columns
```

```
[ ]: Index(['the_geom', 'the_geom_webmercator', 'objectid', 'dc_dist', 'psa',
       'dispatch_date_time', 'dispatch_date', 'dispatch_time', 'hour_',
       'dc_key', 'location_block', 'ucr_general', 'text_general_code',
       'point_x', 'point_y', 'lat', 'lng'],
      dtype='object')
```

```
[ ]: crime.head()
```

```
[ ]: the_geom \
0 0101000020E6100000EA77405D0DC952C016F8ED98F8FA...
1 0101000020E6100000EA77405D0DC952C016F8ED98F8FA...
2 0101000020E6100000EA77405D0DC952C016F8ED98F8FA...
3 0101000020E6100000EA77405D0DC952C016F8ED98F8FA...
4 0101000020E6100000FB79CF5866C552C0942E81847604...
```

	the_geom_webmercator	objectid	dc_dist	psa
0	0101000020110F000080BB90BAA8E85FC1EC88B8A8528A...	107	6	1
1	0101000020110F000080BB90BAA8E85FC1EC88B8A8528A...	108	6	1
2	0101000020110F000080BB90BAA8E85FC1EC88B8A8528A...	109	6	1
3	0101000020110F000080BB90BAA8E85FC1EC88B8A8528A...	110	6	1
4	0101000020110F0000A78BF98174E25FC145F74595D894...	111	2	1

```

      dispatch_date_time dispatch_date dispatch_time hour_ dc_key \
0 2013-05-28 09:43:00    2013-05-28    09:43:00  9.0 201306025636
1 2013-11-26 10:24:00    2013-11-26    10:24:00 10.0 201306061456
2 2013-12-16 13:10:00    2013-12-16   13:10:00 13.0 201306064336
3 2014-01-27 13:12:00    2014-01-27   13:12:00 13.0 201406003790
4 2011-09-08 11:27:00    2011-09-08  11:27:00 11.0 201102059237

      location_block ucr_general text_general_code point_x \
0 N 02ND ST / SPRING GARDEN ST          600 Thefts -75.141441
1 N 02ND ST / SPRING GARDEN ST          300 Robbery No Firearm -75.141441
2 N 02ND ST / SPRING GARDEN ST          600 Thefts -75.141441
3 N 02ND ST / SPRING GARDEN ST          600 Thefts -75.141441
4      5900 BLOCK LORETTO AVE           500 Burglary Residential -75.084372

      point_y      lat      lng
0 39.960712 39.960712 -75.141441
1 39.960712 39.960712 -75.141441
2 39.960712 39.960712 -75.141441
3 39.960712 39.960712 -75.141441
4 40.034867 40.034867 -75.084372

```

[]: crime.isna().sum()# sum of null values

```

[ ]: the_geom            282
the_geom_webmercator     282
objectid                 0
dc_dist                   0
psa                      644
dispatch_date_time        0
dispatch_date              0
dispatch_time              0
hour_                     28
dc_key                    0
location_block             55
ucr_general                0
text_general_code           0
point_x                  1423
point_y                  1423
lat                      282
lng                      282
dtype: int64

```

[]: crime.dtypes#type of data

```

[ ]: the_geom            object
the_geom_webmercator     object

```

```

objectid           int64
dc_dist            int64
psa                object
dispatch_date_time object
dispatch_date      object
dispatch_time      object
hour_              float64
dc_key             int64
location_block     object
ucr_general        int64
text_general_code object
point_x            float64
point_y            float64
lat                float64
lng                float64
dtype: object

```

```
[ ]: #dropping null values of lat and lng values
crime.dropna(subset=['lat'], inplace=True)
crime.dropna(subset=['lng'], inplace=True)
```

```
[ ]: crime['dispatch_date'] = pd.to_datetime(crime['dispatch_date'])# converting
    ↪ dispatch date to time value
```

```
[ ]: crime.shape#size of dataset
```

```
[ ]: (2828248, 17)
```

```
[ ]: crime = crime.loc[(crime["dispatch_date"] >= "2015-01-01") &
    ↪(crime["dispatch_date"] <= "2021-12-31")].reset_index(drop=True)#filtering
    ↪for dataset for 2015 and onwards
crime.shape
```

```
[ ]: (1081115, 17)
```

```
[ ]: crime.info()#dataset description
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1081115 entries, 0 to 1081114
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   the_geom          1081115 non-null   object 
 1   the_geom_webmercator 1081115 non-null   object 
 2   objectid          1081115 non-null   int64  
 3   dc_dist            1081115 non-null   int64  
 4   psa                1080946 non-null   object 

```

```
5  dispatch_date_time    1081115 non-null  object
6  dispatch_date        1081115 non-null  datetime64[ns]
7  dispatch_time        1081115 non-null  object
8  hour_                 1081111 non-null  float64
9  dc_key                1081115 non-null  int64
10 location_block       1081108 non-null  object
11 ucr_general          1081115 non-null  int64
12 text_general_code    1081115 non-null  object
13 point_x              1079974 non-null  float64
14 point_y              1079974 non-null  float64
15 lat                   1081115 non-null  float64
16 lng                   1081115 non-null  float64
dtypes: datetime64[ns](1), float64(5), int64(4), object(7)
memory usage: 140.2+ MB
```

```
[ ]: #minimum of pointx and longitude
print(crime['point_x'].min())
print(crime['lng'].min())
```

```
-81.58137853
-81.58137853
```

```
[ ]: #maximum of point x and longitude
print(crime['point_x'].max())
print(crime['lng'].max())
```

```
2725830.9416288
-74.95758648
```

```
[ ]: print(crime['point_y'].max())
print(crime['lat'].max())
```

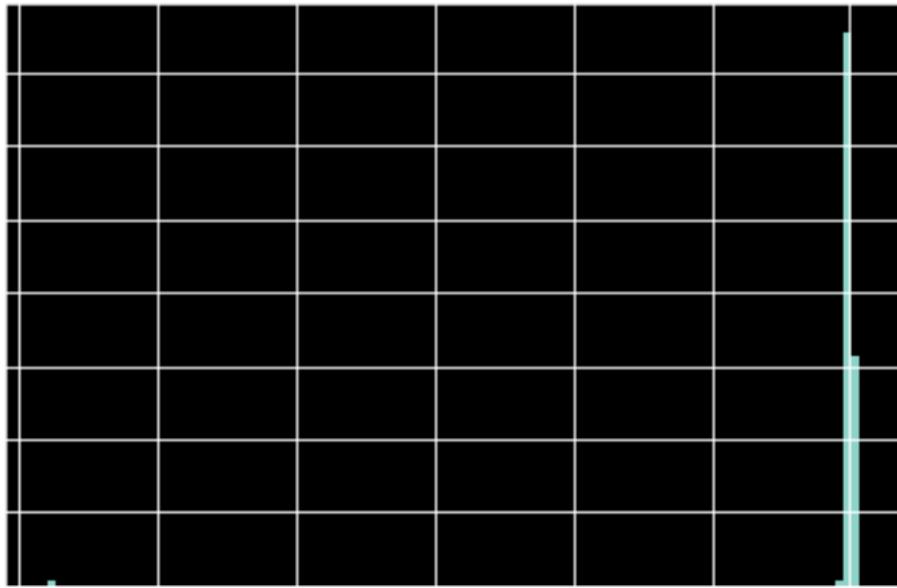
```
278069.04403542
40.13771285
```

```
[ ]: print(crime['point_y'].min())
print(crime['lat'].min())
#After analyzing longitude and latitude and point_x and point_y, we came to the
#conclusion that we should not use point_x and point_y as they have more
#irregular values.
#lat and lng also have values that are beyond the range
```

```
-3975202.88585439
28.41954829
```

```
[ ]: crime['lat'].hist(bins = 100) # histogram shows the irregularities in lat data
```

```
[ ]: <AxesSubplot:>
```

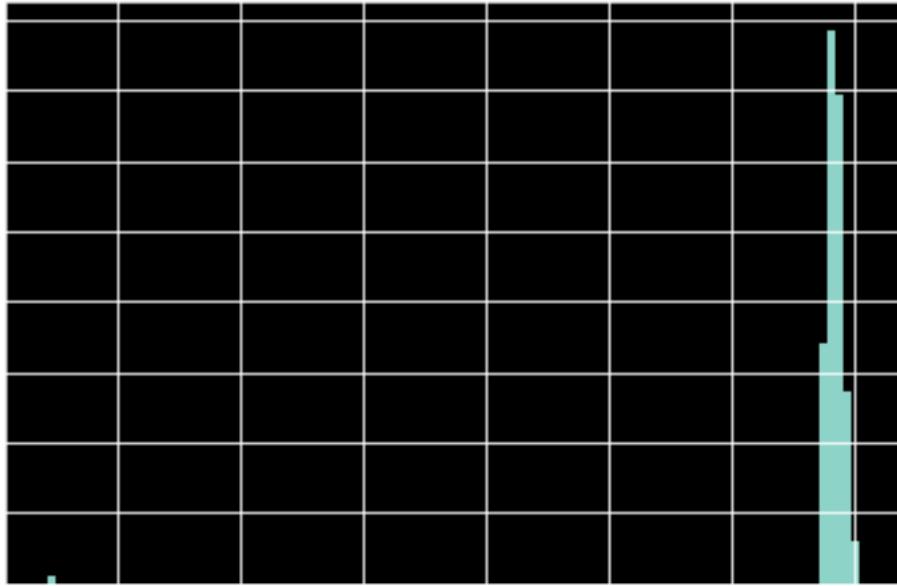


```
[ ]: print(crime.loc[crime['lat']>30].shape) #size of dataset after with just more than 30 lat
      ↵
print(crime.loc[crime['lat']<30].shape) #size of dataset after with less than than 30 lat.
      ↵
#There are just around 5000 rows that are irregular. Thus if we remove them we dont be losing a lot fo data
```

```
(1076153, 17)
(4962, 17)
```

```
[ ]: crime['lng'].hist(bins= 100) # histogram shows the irregularities in lng data
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: print(crime.loc[crime['lng']<-81].shape) #size of dataset after with less than  
→ -81 as longitude. This is the irregular data.  
print(crime.loc[crime['lng']>-81].shape) #size of dataset after with more than  
→ -81 as longitude.
```

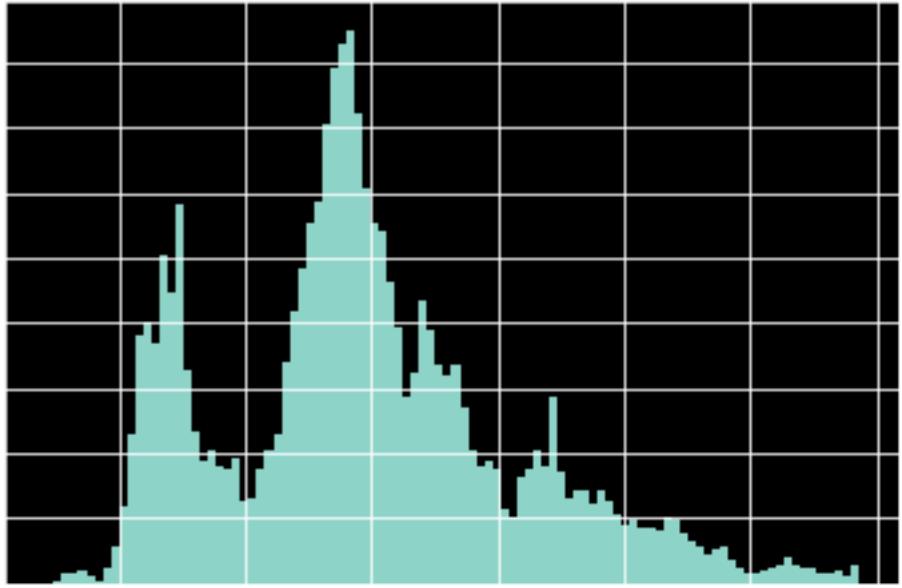
```
(4962, 17)  
(1076153, 17)
```

```
[ ]: crime = crime.loc[crime['lng']>-81].reset_index(drop=True) #dropping all  
→ irregular longitude and latitude  
crime.shape
```

```
[ ]: (1076153, 17)
```

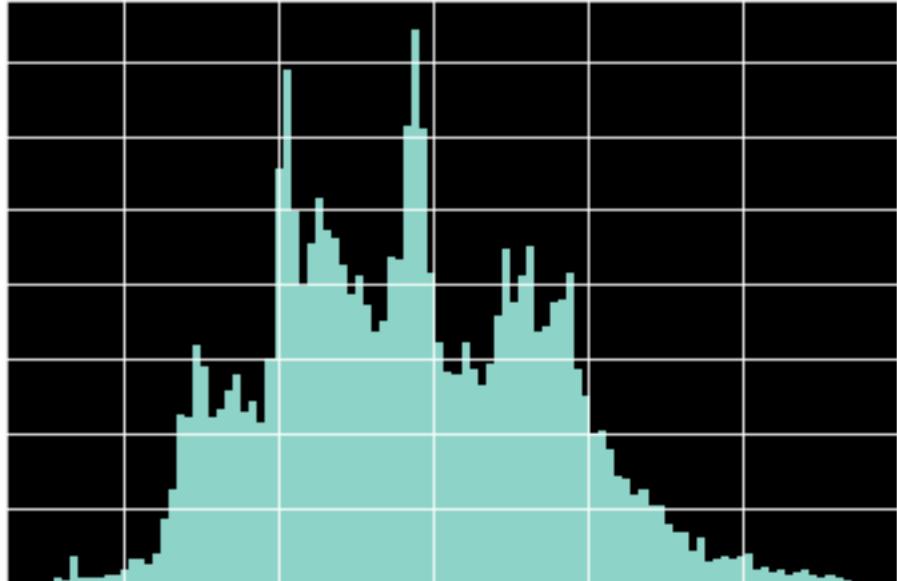
```
[ ]: crime['lng'].hist(bins= 100) #histogram shows that values are within the range  
→ it is supposed to be
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: crime['lat'].hist(bins= 100) #histogram shows that values are within the range ↴ it is supposed to be
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: #combining latitude and longitude data into geometrey column so that we can use it on map
      ↵it on map
crs = {'init': 'epsg:4326'}
geometry = [Point(xy) for xy in zip(crime["lng"], crime["lat"])]
geometry[:3]
```

```
[ ]: [<shapely.geometry.point.Point at 0x180e40160>,
      <shapely.geometry.point.Point at 0x16f827850>,
      <shapely.geometry.point.Point at 0x16f826a10>]
```

```
[ ]: crime = gpd.GeoDataFrame(crime,
                                crs = crs,
                                geometry = geometry)

crime.head()
```

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/pyproj/crs/crs.py:131: FutureWarning: '+init=<authority>:<code>' syntax
is deprecated. '<authority>:<code>' is the preferred initialization method. When
making the change, be mindful of axis order changes:
https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-
proj-6
```

```
    in_crs_string = _prepare_from_proj_string(in_crs_string)
```

```
[ ]:                                     the_geom  \
0  0101000020E61000002FD31F2F1ECE52C07129BE0C0CF5...
1  0101000020E61000002FD31F2F1ECE52C07129BE0C0CF5...
2  0101000020E61000002FD31F2F1ECE52C07129BE0C0CF5...
3  0101000020E61000002FD31F2F1ECE52C07129BE0C0CF5...
4  0101000020E61000002FD31F2F1ECE52C07129BE0C0CF5...

                                     the_geom_webmercator  objectid  dc_dist psa  \
0  0101000020110F0000401FFA8143F15FC1160AD2D2C283...     117      12    1
1  0101000020110F0000401FFA8143F15FC1160AD2D2C283...     118      12    1
2  0101000020110F0000401FFA8143F15FC1160AD2D2C283...     119      12    1
3  0101000020110F0000401FFA8143F15FC1160AD2D2C283...     120      12    1
4  0101000020110F0000401FFA8143F15FC1160AD2D2C283...     121      12    1

      dispatch_date_time dispatch_date dispatch_time hour_          dc_key  \
0  2018-01-06 10:56:00    2018-01-06    10:56:00   10.0  201812001185
1  2018-06-21 22:57:00    2018-06-21    22:57:00   22.0  201812045738
2  2018-07-23 01:58:00    2018-07-23    01:58:00    1.0  201812053606
3  2015-01-15 08:58:00    2015-01-15    08:58:00    8.0  201512003444
4  2015-05-27 20:09:00    2015-05-27    20:09:00   20.0  201512041070

      location_block  ucr_general           text_general_code  \
0  6600 BLOCK ESSINGTON AVE            600                      Theft
```

1	6600 BLOCK ESSINGTON AVE	300	Robbery Firearm
2	6600 BLOCK ESSINGTON AVE	300	Robbery No Firearm
3	6600 BLOCK ESSINGTON AVE	500	Burglary Non-Residential
4	6600 BLOCK ESSINGTON AVE	400	Aggravated Assault No Firearm

	point_x	point_y	lat	lng	geometry
0	-75.220592	39.91443	39.91443	-75.220592	POINT (-75.22059 39.91443)
1	-75.220592	39.91443	39.91443	-75.220592	POINT (-75.22059 39.91443)
2	-75.220592	39.91443	39.91443	-75.220592	POINT (-75.22059 39.91443)
3	-75.220592	39.91443	39.91443	-75.220592	POINT (-75.22059 39.91443)
4	-75.220592	39.91443	39.91443	-75.220592	POINT (-75.22059 39.91443)

```
[ ]: crime['text_general_code'].value_counts()
```

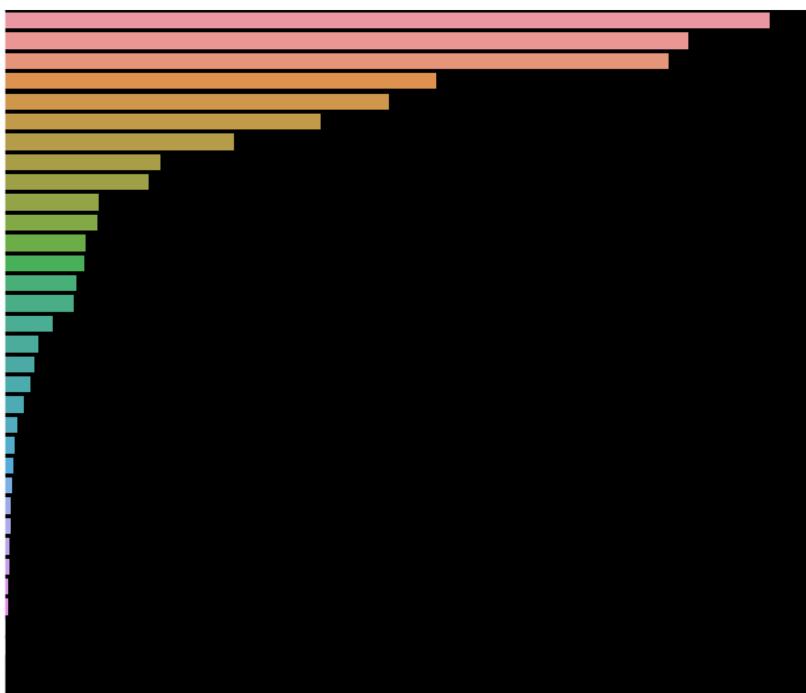
All Other Offenses	183402
Other Assaults	163910
Thefts	159038
Vandalism/Criminal Mischief	103315
Theft from Vehicle	91929
Fraud	75606
Narcotic / Drug Law Violations	55026
Aggravated Assault No Firearm	37394
Burglary Residential	34550
Motor Vehicle Theft	22476
Robbery No Firearm	22182
DRIVING UNDER THE INFLUENCE	19354
Aggravated Assault Firearm	19257
Weapon Violations	17267
Robbery Firearm	16648
Burglary Non-Residential	11719
Disorderly Conduct	7963
Rape	7302
Other Sex Offenses (Not Commercialized)	6353
Prostitution and Commercialized Vice	4709
Arson	3175
Embezzlement	2316
Homicide - Criminal	2059
Offenses Against Family and Children	1730
Public Drunkenness	1403
Forgery and Counterfeiting	1389
Recovered Stolen Motor Vehicle	1137
Vagrancy/Loitering	1120
Liquor Law Violations	916
Receiving Stolen Property	885
Homicide - Criminal	365
Gambling Violations	250
Homicide - Justifiable	7

```
Homicide - Gross Negligence  
Name: text_general_code, dtype: int64
```

1

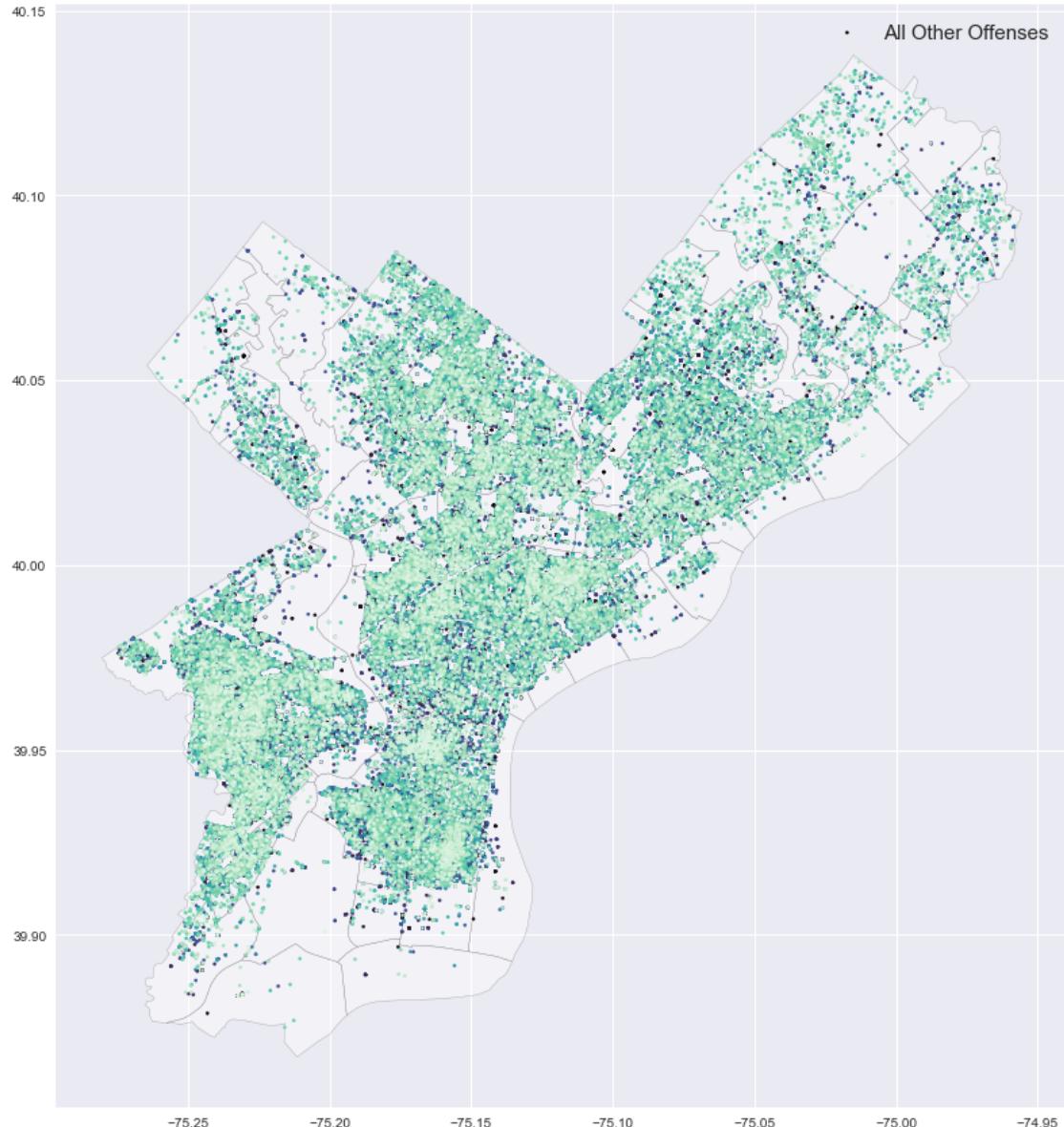
```
[ ]: # Countplot of crime type  
# most fo the crime are all other offenses,assults is the second highest  
sns.catplot(y = 'text_general_code',  
            kind = 'count',  
            height = 8,  
            aspect = 1.5,  
            order = crime.text_general_code.value_counts().index,  
            data = crime)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x180db0310>
```



```
[ ]: #plotting where all other offenses are occurring  
fig, ax = plt.subplots(figsize =(15,15))  
plt.style.use('seaborn')  
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white",  
                                   edgecolor='black')  
#crime[crime['text_general_code'] == 'All Other Offenses'].plot(ax = ax,  
#                                                               markersize = 20, color = "blue", marker = "o", label = "All Other Offenses")  
crime[crime['text_general_code'] == 'All Other Offenses'].plot(ax = ax, cmap =  
#                                                               'mako', legend=True, markersize = 5, label = "All Other Offenses")
```

```
#crime[crime_df['text_general_code'] == 'Thefts'].plot(ax = ax, markersize = 20, color = "red", marker = "^", label = "Thefts")
plt.legend(prop = {'size' : 15})
plt.show()
```

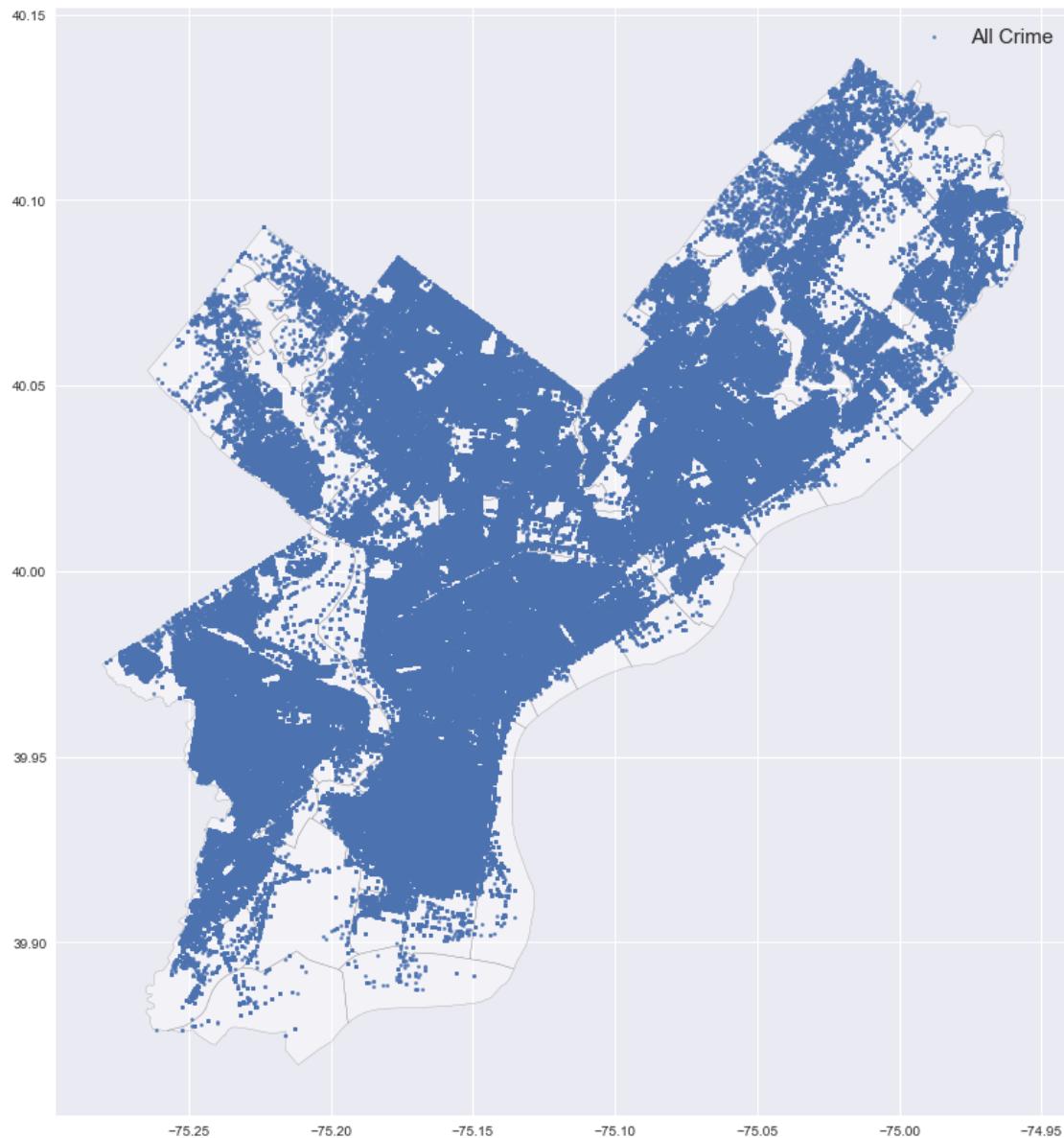


```
[ ]: #plotting where crimes are occurring
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white", edgecolor='black')
```

```

#crime[crime['text_general_code'] == 'All Other Offenses'].plot(ax = ax,
    ↪markersize = 20, color = "blue", marker = "o", label = "All Other Offenses")
crime.plot(ax = ax,legend=True, markersize = 5, label = "All Crime")
#crime[crime_df['text_general_code'] == 'Thefts'].plot(ax = ax, markersize = 20,
    ↪color = "red", marker = "^", label = "Thefts")
plt.legend(prop = {'size' : 15})
plt.show()

```



[]: # Crimes by the hour

```
sns.catplot( x = 'hour_',

```

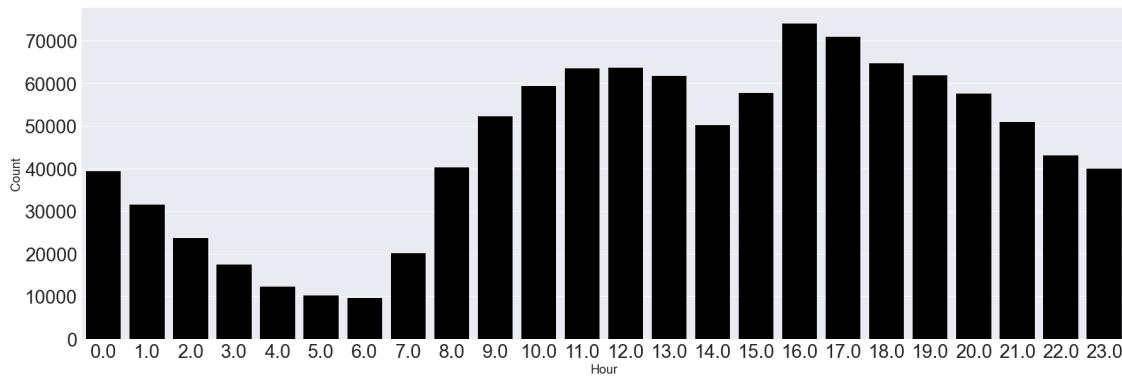
```

        kind = 'count',
        height = 8,
        aspect = 3,
        color = 'black',
        data = crime)

plt.xticks(size=30)
plt.yticks(size=30)
plt.xlabel('Hour', fontsize = 20)
plt.ylabel('Count', fontsize = 20)

```

[]: Text(-7.324999999999974, 0.5, 'Count')



0.2.16 City of Philadelphia: Property Assessment

<https://metadata.phila.gov/#home/datasetdetails/5543865f20583086178c4ee5/>

Property Assessment major takeaways: - Typical house is around 1000 sq feet and has a radius of 6m - Total livable area and total area are very skewed. It did not say anywhere what metric it was using. We are guessing it was square feet - Most of assessment was from 2021 - Vacant Lot has less market value than non-vacant lots - Vacant lot has higher mean depth meaning they were more away from the roads. The depth is measured from the principal street back to the rear property line or secondary street

Feature engineering that occurred in this dataset are: - Dropped all unknown lat and long data - The column "Category_code_description" was used to create the labelled column for vacant lot. If it had "Vacant Land" then labelled as 1, if not then 0. The labelled column is called "vacant". There are 44162 Vacant lot and 537190 non vacant lots. - Removed some rows that had property assessment come up multiple times

[]: `assess = pd.read_csv('data/city/opa_properties_public.csv') #uploading dataset`

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3457: DtypeWarning: Columns
(1,2,4,11,12,21,25,30,34,42,47,53,54,55,60,67,69,71) have mixed types. Specify
```

```
dtype option on import or set low_memory=False.  
exec(code_obj, self.user_global_ns, self.user_ns)
```

```
[ ]: assess.shape
```

```
[ ]: (581404, 78)
```

```
[ ]: assess.head()
```

```
[ ]: objectid      assessment_date basements beginning_point book_and_page  \  
0 55242915 1949-01-01 00:00:00      NaN          NaN 0872170  
1 55242916 1949-01-01 00:00:00      NaN          NaN 2620507  
2 55242917 1949-01-01 00:00:00      NaN          NaN 2677268  
3 55242918 1949-01-01 00:00:00      NaN          NaN 2886779  
4 55242919 1949-01-01 00:00:00      NaN          NaN 2886779  
  
building_code  building_code_description category_code  \  
0      SR      VACANT LAND RESIDE < ACRE 6  
1      SR      VACANT LAND RESIDE < ACRE 6  
2      SR      VACANT LAND RESIDE < ACRE 6  
3      SR      VACANT LAND RESIDE < ACRE 6  
4      SR      VACANT LAND RESIDE < ACRE 6  
  
category_code_description census_tract ... unit utility view_type  \  
0      Vacant Land      142.0 ... CA      NaN      NaN  
1      Vacant Land      379.0 ... NaN      NaN      NaN  
2      Vacant Land      142.0 ... NaN      NaN      NaN  
3      Vacant Land      367.0 ... NaN      NaN      NaN  
4      Vacant Land      367.0 ... NaN      NaN      NaN  
  
year_built  year_built_estimate zip_code zoning      pin      lat  \  
0      0.0            NaN 19123.0 RSA5 1001317719 -75.144757  
1      0.0            NaN 19134.0 RSA5 1001124565 -75.092534  
2      0.0            NaN 19123.0 RSA5 1001430746 -75.146154  
3      0.0            NaN 19123.0 ICMX 1001206446 -75.145586  
4      0.0            NaN 19123.0 ICMX 1001206456 -75.146035  
  
    lng  
0 39.967847  
1 39.991459  
2 39.967067  
3 39.962679  
4 39.962772
```

```
[5 rows x 78 columns]
```

```
[ ]: assess.dtypes#type of data
```

```
[ ]: objectid          int64
      assessment_date    object
      basements          object
      beginning_point    object
      book_and_page      object
      ...
      zip_code           float64
      zoning             object
      pin                int64
      lat                float64
      lng                float64
Length: 78, dtype: object
```

```
[ ]: assess.columns
```

```
[ ]: Index(['objectid', 'assessment_date', 'basements', 'beginning_point',
       'book_and_page', 'building_code', 'building_code_description',
       'category_code', 'category_code_description', 'census_tract',
       'central_air', 'cross_reference', 'date_exterior_condition', 'depth',
       'exempt_building', 'exempt_land', 'exterior_condition', 'fireplaces',
       'frontage', 'fuel', 'garage_spaces', 'garage_type',
       'general_construction', 'geographic_ward', 'homestead_exemption',
       'house_extension', 'house_number', 'interior_condition', 'location',
       'mailing_address_1', 'mailing_address_2', 'mailing_care_of',
       'mailing_city_state', 'mailing_street', 'mailing_zip', 'market_value',
       'market_value_date', 'number_of_bathrooms', 'number_of_bedrooms',
       'number_of_rooms', 'number_stories', 'off_street_open',
       'other_building', 'owner_1', 'owner_2', 'parcel_number', 'parcel_shape',
       'quality_grade', 'recording_date', 'registry_number', 'sale_date',
       'sale_price', 'separate_utilities', 'sewer', 'site_type', 'state_code',
       'street_code', 'street_designation', 'street_direction', 'street_name',
       'suffix', 'taxable_building', 'taxable_land', 'topography',
       'total_area', 'total_livable_area', 'type_heater', 'unfinished', 'unit',
       'utility', 'view_type', 'year_built', 'year_built_estimate', 'zip_code',
       'zoning', 'pin', 'lat', 'lng'],
      dtype='object')
```

```
[ ]: assess.isna().sum() #sum of null values
```

```
[ ]: objectid          0
      assessment_date   547392
      basements         255101
      beginning_point   11174
      book_and_page     2769
      ...
      zip_code          53
      zoning            734
```

```
pin          0
lat         52
lng         52
Length: 78, dtype: int64
```

```
[ ]: assess['parcel_number'].nunique()
```

```
[ ]: 581395
```

```
[ ]: assess['assessment_date'] = pd.to_datetime(assess['assessment_date'])#turning
      ↪ into datetime value
```

```
[ ]: assess.describe(include = 'all')
```

```
/var/folders/6p/wpw9qml57530xkxqkkhprrf40000gn/T/ipykernel_74088/2767216599.py:1
: FutureWarning: Treating datetime data as categorical rather than numeric in
`.describe` is deprecated and will be removed in a future version of pandas.
Specify `datetime_is_numeric=True` to silence this warning and adopt the future
behavior now.
    assess.describe(include = 'all')
```

```
[ ]:      objectid      assessment_date basements \
count  5.814040e+05            34012     326303
unique        NaN             1669       16
top           NaN  2021-10-06 17:51:57        D
freq           NaN             133    119848
first          NaN  1949-01-01 00:00:00       NaN
last            NaN  2022-01-05 13:57:16       NaN
mean   5.553357e+07             NaN       NaN
std    1.678370e+05             NaN       NaN
min    5.524285e+07             NaN       NaN
25%    5.538822e+07             NaN       NaN
50%    5.553357e+07             NaN       NaN
75%    5.567892e+07             NaN       NaN
max    5.582427e+07             NaN       NaN
```

```
beginning_point book_and_page building_code \
count            570230        578635      581400
unique          435488        499527       806
top    57' S BAINBRIDGE ST            0000000        030
freq             862         28720      176692
first          NaN           NaN       NaN
last            NaN           NaN       NaN
mean           NaN           NaN       NaN
std            NaN           NaN       NaN
min            NaN           NaN       NaN
25%           NaN           NaN       NaN
```

50%		NaN		NaN			
75%		NaN		NaN			
max		NaN		NaN			
	building_code_description		category_code	category_code_description	\		
count		581387	581404.000000		581323		
unique		798			6		
top	ROW 2 STY MASONRY			Single Family			
freq		176692			461881		
first		NaN			NaN		
last		NaN			NaN		
mean		NaN	1.606518		NaN		
std		NaN	1.435826		NaN		
min		NaN	1.000000		NaN		
25%		NaN	1.000000		NaN		
50%		NaN	1.000000		NaN		
75%		NaN	1.000000		NaN		
max		NaN	15.000000		NaN		
	census_tract	...	unit	utility	view_type	year_built	\
count	581349.000000	...	38902	9103	560367	578871.0	
unique		NaN	...	8775	5	8	402.0
top		NaN	...	A	A	I	1925.0
freq		NaN	...	2037	8484	521899	113704.0
first		NaN	...	NaN	NaN	NaN	NaN
last		NaN	...	NaN	NaN	NaN	NaN
mean	195.130075	...	NaN	NaN	NaN	NaN	NaN
std	118.747152	...	NaN	NaN	NaN	NaN	NaN
min	1.000000	...	NaN	NaN	NaN	NaN	NaN
25%	93.000000	...	NaN	NaN	NaN	NaN	NaN
50%	188.000000	...	NaN	NaN	NaN	NaN	NaN
75%	302.000000	...	NaN	NaN	NaN	NaN	NaN
max	891.000000	...	NaN	NaN	NaN	NaN	NaN
	year_built_estimate		zip_code	zoning		pin	\
count		438577	581351.000000	580670	5.814040e+05		
unique		3		NaN	36		NaN
top		Y		NaN	RSA5		NaN
freq		438283		NaN	311976		NaN
first		NaN		NaN	NaN		NaN
last		NaN		NaN	NaN		NaN
mean		NaN	19133.376399	NaN	1.001361e+09		
std		NaN	183.463516	NaN	1.805621e+05		
min		NaN	19102.000000	NaN	1.001049e+09		
25%		NaN	19123.000000	NaN	1.001204e+09		
50%		NaN	19134.000000	NaN	1.001361e+09		
75%		NaN	19144.000000	NaN	1.001516e+09		

```
max                      NaN    88888.000000      NaN  1.001682e+09  
  
          lat        lng  
count  581352.000000  581352.000000  
unique           NaN        NaN  
top            NaN        NaN  
freq            NaN        NaN  
first           NaN        NaN  
last            NaN        NaN  
mean       -75.143498    39.999843  
std        0.065909    0.050866  
min       -75.274389    39.875128  
25%       -75.182677    39.957730  
50%       -75.155155    39.996529  
75%       -75.104724    40.039888  
max       -74.958190    40.137705
```

[13 rows x 78 columns]

```
[ ]: assess[['total_livable_area', 'total_area']].isna().sum()
```

```
[ ]: total_livable_area    2529  
total_area                 452  
dtype: int64
```

```
[ ]: assess[['total_livable_area', 'total_area']].describe()  
# we will be using this later while trying to count number of crime and 311  
→calls  
#typical house is around 1000square feet
```

```
[ ]:      total_livable_area    total_area  
count      5.788750e+05  5.809520e+05  
mean      2.266656e+03  5.049853e+03  
std       1.685203e+04  2.966376e+05  
min       0.000000e+00  0.000000e+00  
25%      1.028000e+03  8.824500e+02  
50%      1.230000e+03  1.280000e+03  
75%      1.586000e+03  2.051102e+03  
max      2.500000e+06  2.076941e+08
```

```
[ ]: 1000/10.764 #converting to square meter
```

```
[ ]: 92.9022668153103
```

```
[ ]: (92.90)**0.5 #each side of the house
```

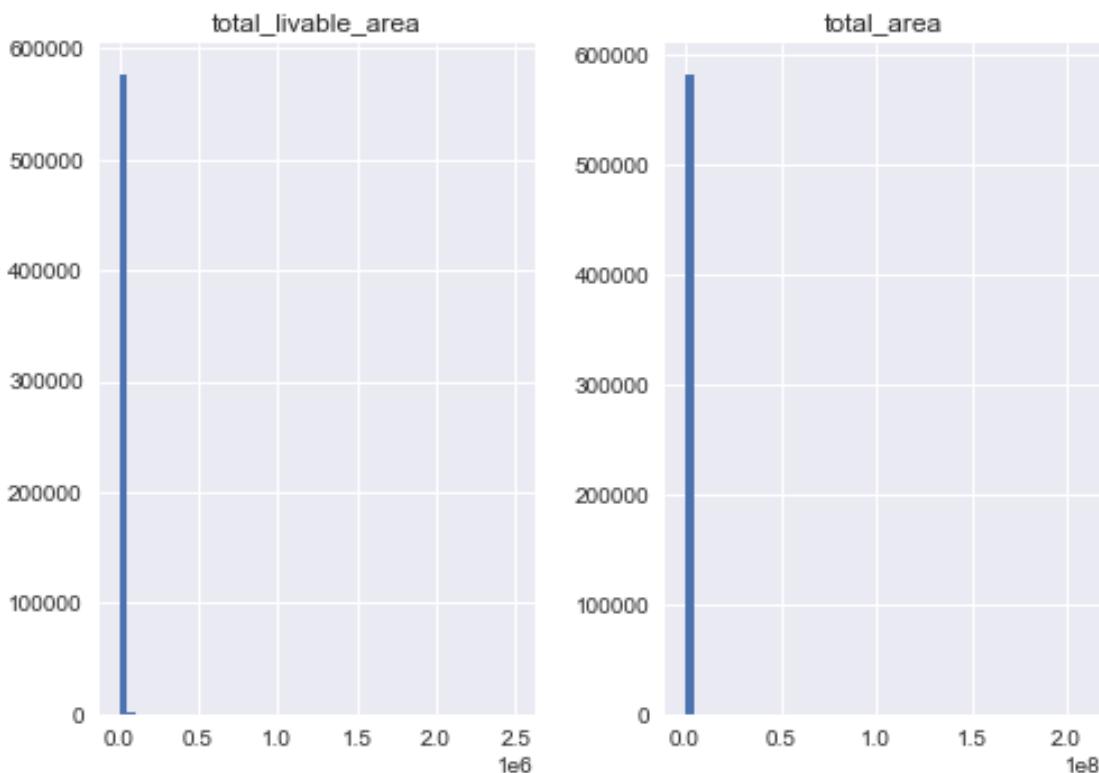
```
[ ]: 9.638464608017193
```

```
[ ]: (((9.638464608017193**2) + (9.638464608017193**2))**0.5)/2 #typical house in philly is with radius of 6m.
#Using hypoteneous formula to find the find side of the triangle and then devidiing by 2 to find the radius
```

```
[ ]: 6.815423684555496
```

```
[ ]: assess[['total_livable_area', 'total_area']].hist(bins = 50)
```

```
[ ]: array([[<AxesSubplot:title={'center':'total_livable_area'}>,
<AxesSubplot:title={'center':'total_area'}>]], dtype=object)
```



```
[ ]: assess['assessment_year'] = pd.to_datetime(assess['assessment_date']).dt.year
assess.groupby('assessment_year')['objectid'].count()
```

```
[ ]: assessment_year
1949.0      124
2020.0      144
2021.0    33736
2022.0       8
Name: objectid, dtype: int64
```

```
[ ]: assess['parcel_number'].nunique()
[ ]: 581395
[ ]: assess.loc[assess['category_code'] == 6].shape# size of data with vacant lot
[ ]: (44190, 79)
[ ]: #dropping null values of lat and lng values
[ ]: assess.dropna(subset=['lat'], inplace=True)
[ ]: assess.dropna(subset=['lng'], inplace=True)
[ ]: assess.shape
[ ]: (581352, 79)
[ ]: assess.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 581352 entries, 0 to 581403
Data columns (total 79 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   objectid        581352 non-null   int64  
 1   assessment_date 34005 non-null    datetime64[ns]
 2   basements       326298 non-null   object  
 3   beginning_point 570185 non-null   object  
 4   book_and_page   578589 non-null   object  
 5   building_code   581348 non-null   object  
 6   building_code_description 581335 non-null   object  
 7   category_code   581352 non-null   int64  
 8   category_code_description 581275 non-null   object  
 9   census_tract    581349 non-null   float64 
 10  central_air     286727 non-null   object  
 11  cross_reference 114843 non-null   object  
 12  date_exterior_condition 334294 non-null   object  
 13  depth          580743 non-null   float64 
 14  exempt_building 581109 non-null   float64 
 15  exempt_land     581109 non-null   float64 
 16  exterior_condition 553554 non-null   float64 
 17  fireplaces      576960 non-null   float64 
 18  frontage        580744 non-null   float64 
 19  fuel            13982 non-null    object  
 20  garage_spaces   576876 non-null   float64 
 21  garage_type     500809 non-null   object  
 22  general_construction 517343 non-null   object  
 23  geographic_ward 581349 non-null   float64
```

24	homestead_exemption	580922	non-null	float64
25	house_extension	26738	non-null	object
26	house_number	581352	non-null	int64
27	interior_condition	552755	non-null	float64
28	location	581352	non-null	object
29	mailing_address_1	56731	non-null	object
30	mailing_address_2	6518	non-null	object
31	mailing_care_of	28372	non-null	object
32	mailing_city_state	232848	non-null	object
33	mailing_street	232359	non-null	object
34	mailing_zip	231711	non-null	object
35	market_value	581109	non-null	float64
36	market_value_date	0	non-null	float64
37	number_of_bathrooms	577028	non-null	float64
38	number_of_bedrooms	577270	non-null	float64
39	number_of_rooms	547825	non-null	float64
40	number_stories	577270	non-null	float64
41	off_street_open	580503	non-null	float64
42	other_building	1312	non-null	object
43	owner_1	581352	non-null	object
44	owner_2	203765	non-null	object
45	parcel_number	581352	non-null	int64
46	parcel_shape	575505	non-null	object
47	quality_grade	56269	non-null	object
48	recording_date	581203	non-null	object
49	registry_number	580649	non-null	object
50	sale_date	581344	non-null	object
51	sale_price	581338	non-null	float64
52	separate_utilities	25671	non-null	object
53	sewer	8855	non-null	object
54	site_type	296357	non-null	object
55	state_code	573060	non-null	object
56	street_code	581352	non-null	int64
57	street_designation	581347	non-null	object
58	street_direction	226495	non-null	object
59	street_name	581352	non-null	object
60	suffix	2918	non-null	object
61	taxable_building	581109	non-null	float64
62	taxable_land	581109	non-null	float64
63	topography	542453	non-null	object
64	total_area	580908	non-null	float64
65	total_livable_area	578832	non-null	float64
66	type_heater	294554	non-null	object
67	unfinished	2706	non-null	object
68	unit	38892	non-null	object
69	utility	9096	non-null	object
70	view_type	560322	non-null	object
71	year_built	578828	non-null	object

```
72 year_built_estimate      438571 non-null  object
73 zip_code                  581351 non-null  float64
74 zoning                   580670 non-null  object
75 pin                      581352 non-null  int64
76 lat                       581352 non-null  float64
77 lng                      581352 non-null  float64
78 assessment_year          34005 non-null  float64
dtypes: datetime64[ns](1), float64(27), int64(6), object(45)
memory usage: 354.8+ MB
```

```
[ ]: print(assess['lng'].min())
```

```
39.87512753045684
```

```
[ ]: print(assess['lng'].max())
```

```
40.137704744433634
```

```
[ ]: print(assess['lat'].max())
```

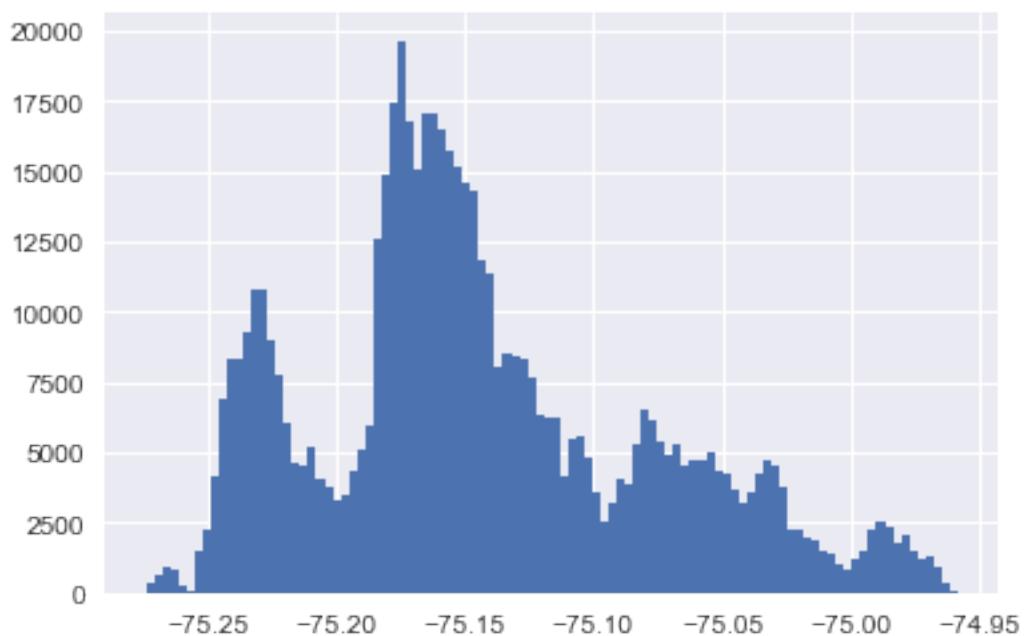
```
-74.9581902396646
```

```
[ ]: print(assess['lat'].min())
```

```
-75.27438935809397
```

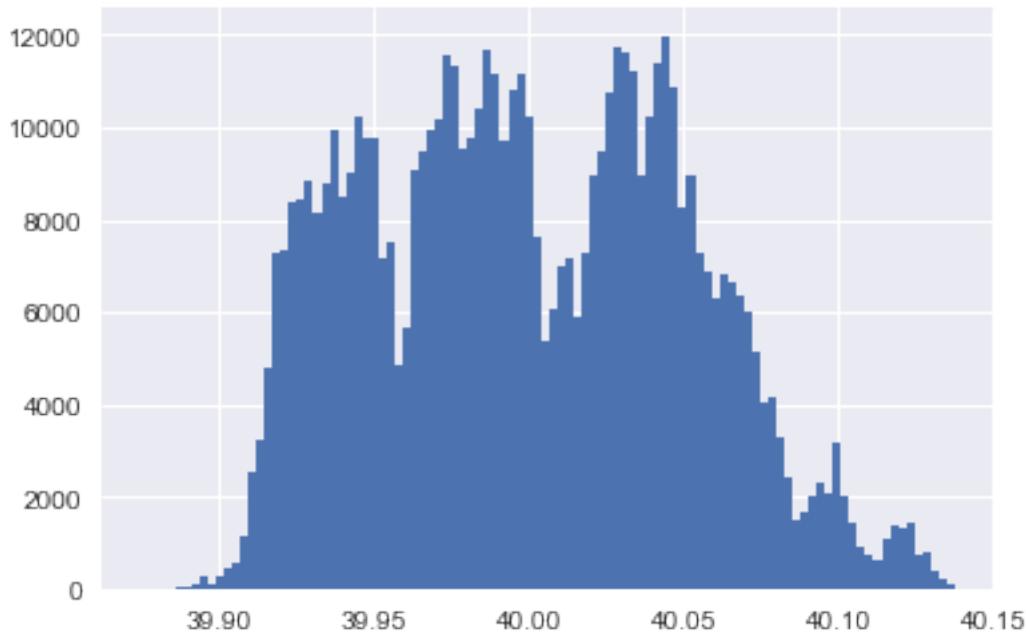
```
[ ]: assess['lat'].hist(bins = 100) #histogram of latitude
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: assess['lng'].hist(bins= 100)#histogram of longitude
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: #combining latitude and longitude data to make geometry column  
crs = {'init': 'epsg:4326'}  
geometry = [Point(xy) for xy in zip(assess["lat"], assess["lng"])]  
geometry[:3]
```

```
[ ]: [<shapely.geometry.point.Point at 0x2b60b6020>,  
<shapely.geometry.point.Point at 0x2b9e8c820>,  
<shapely.geometry.point.Point at 0x2b9e8d780>]
```

```
[ ]: assess = gpd.GeoDataFrame(assess,  
                           crs = crs,  
                           geometry = geometry)  
  
assess.head()
```

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-  
packages/pyproj/crs/crs.py:131: FutureWarning: '+init=<authority>:<code>' syntax  
is deprecated. '<authority>:<code>' is the preferred initialization method. When  
making the change, be mindful of axis order changes:  
https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-
```

```

proj-6
    in_crs_string = _prepare_from_proj_string(in_crs_string)

[ ]:   objectid assessment_date basements beginning_point book_and_page \
0  55242915      1949-01-01      NaN          NaN  0872170
1  55242916      1949-01-01      NaN          NaN  2620507
2  55242917      1949-01-01      NaN          NaN  2677268
3  55242918      1949-01-01      NaN          NaN  2886779
4  55242919      1949-01-01      NaN          NaN  2886779

    building_code  building_code_description category_code \
0        SR        VACANT LAND RESIDE < ACRE          6
1        SR        VACANT LAND RESIDE < ACRE          6
2        SR        VACANT LAND RESIDE < ACRE          6
3        SR        VACANT LAND RESIDE < ACRE          6
4        SR        VACANT LAND RESIDE < ACRE          6

    category_code_description census_tract ... view_type year_built \
0            Vacant Land       142.0 ...      NaN      0.0
1            Vacant Land       379.0 ...      NaN      0.0
2            Vacant Land       142.0 ...      NaN      0.0
3            Vacant Land       367.0 ...      NaN      0.0
4            Vacant Land       367.0 ...      NaN      0.0

    year_built_estimate zip_code zoning           pin      lat      lng \
0             NaN     19123.0    RSA5  1001317719 -75.144757  39.967847
1             NaN     19134.0    RSA5  1001124565 -75.092534  39.991459
2             NaN     19123.0    RSA5  1001430746 -75.146154  39.967067
3             NaN     19123.0    ICMX  1001206446 -75.145586  39.962679
4             NaN     19123.0    ICMX  1001206456 -75.146035  39.962772

    assessment_year           geometry
0      1949.0  POINT (-75.14476 39.96785)
1      1949.0  POINT (-75.09253 39.99146)
2      1949.0  POINT (-75.14615 39.96707)
3      1949.0  POINT (-75.14559 39.96268)
4      1949.0  POINT (-75.14604 39.96277)

```

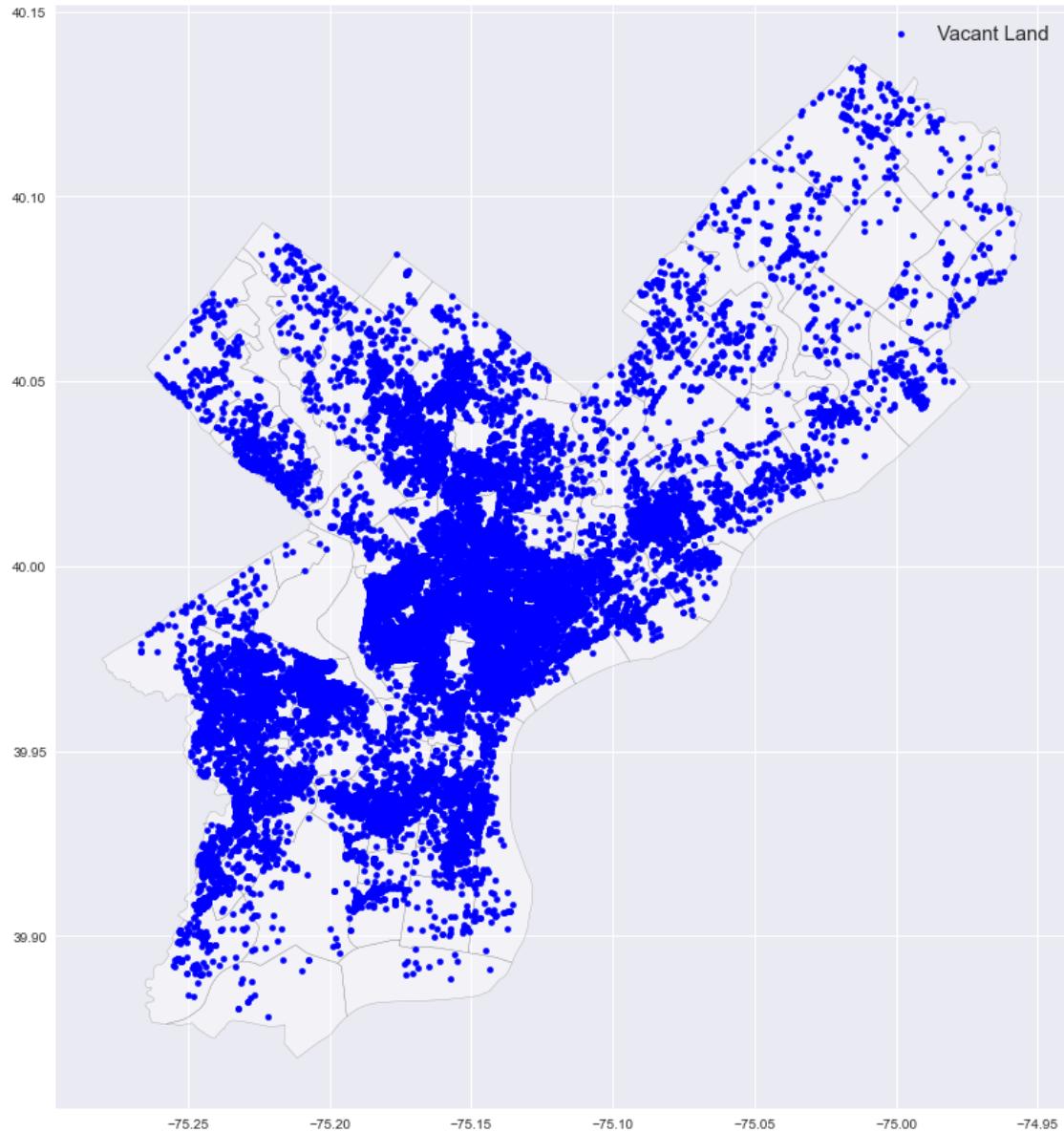
[5 rows x 80 columns]

```

[ ]: #crime_df = crime_df.to_crs(epsg = 4326)
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white", ↴
                                     edgecolor='black')
assess[assess['category_code_description'] == 'Vacant Land'].plot(ax = ax, ↴
                                                               markersize = 20, color = "blue", marker = "o", label = "Vacant Land")

```

```
#crime_df[crime_df['text_general_code'] == 'Thefts'].plot(ax = ax, markersize = 20, color = "red", marker = "^", label = "Thefts")
#geo_df.plot(column = 'BUILD_RANK', ax = ax, alpha = 0.5, legend = True, markersize = 10)
plt.legend(prop = {'size' : 15})
plt.show()
```



```
[ ]: assess1 = assess[['parcel_number', 'objectid',
→'category_code', 'category_code_description',
'census_tract', 'depth', 'location', 'market_value',
'year_built', 'zip_code', 'lat', 'lng', 'geometry']]
```

```
[ ]: assess1.describe()
```

	parcel_number	objectid	category_code	census_tract	\
count	5.813520e+05	5.813520e+05	581352.000000	581349.000000	
mean	4.303274e+08	5.553357e+07	1.606206	195.130075	
std	2.500070e+08	1.678322e+05	1.435171	118.747152	
min	1.100060e+07	5.524285e+07	1.000000	1.000000	
25%	2.322749e+08	5.538822e+07	1.000000	93.000000	
50%	4.058826e+08	5.553358e+07	1.000000	188.000000	
75%	5.831653e+08	5.567892e+07	1.000000	302.000000	
max	8.888006e+08	5.582427e+07	15.000000	891.000000	

	depth	market_value	zip_code	lat	lng
count	5.807430e+05	5.811090e+05	581351.000000	581352.000000	581352.000000
mean	1.094363e+02	2.931600e+05	19133.376399	-75.143498	39.999843
std	4.224082e+03	3.021416e+06	183.463516	0.065909	0.050866
min	0.000000e+00	0.000000e+00	19102.000000	-75.274389	39.875128
25%	5.700000e+01	6.840000e+04	19123.000000	-75.182677	39.957730
50%	7.700000e+01	1.294000e+05	19134.000000	-75.155155	39.996529
75%	1.000000e+02	2.162000e+05	19144.000000	-75.104724	40.039888
max	1.829520e+06	4.541974e+08	88888.000000	-74.958190	40.137705

```
[ ]: def land_type(row):
    if row['category_code'] == 6:
        return 1
    return 0

assess1['vacant'] = assess1.apply(lambda row: land_type(row), axis = 1)
assess1.head()#1 f there is vacant lot
```

/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-packages/geopandas/geodataframe.py:1351: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
super().__setitem__(key, value)

```
[ ]: parcel_number objectid category_code category_code_description \
0      57127281  55242915          6      Vacant Land
1      451106305  55242916          6      Vacant Land
2      57166705   55242917          6      Vacant Land
3      56143925   55242918          6      Vacant Land
4      56143950   55242919          6      Vacant Land

census_tract depth location market_value year_built zip_code \
```

```
0      142.0  439.00  968 N LAWRENCE ST      0.0      0.0  19123.0
1      379.0  256.00  2550 CASTOR AVE      0.0      0.0  19134.0
2      142.0   43.53   513 POPLAR ST      0.0      0.0  19123.0
3      367.0   68.83  428 FAIRMOUNT AVE      0.0      0.0  19123.0
4      367.0   68.83  442 FAIRMOUNT AVE      0.0      0.0  19123.0
```

```
      lat        lng           geometry  vacant
0 -75.144757  39.967847  POINT (-75.14476 39.96785)      1
1 -75.092534  39.991459  POINT (-75.09253 39.99146)      1
2 -75.146154  39.967067  POINT (-75.14615 39.96707)      1
3 -75.145586  39.962679  POINT (-75.14559 39.96268)      1
4 -75.146035  39.962772  POINT (-75.14604 39.96277)      1
```

```
[ ]: assess1['vacant'].value_counts()# there are more vacant lots
```

```
[ ]: 0      537190
1      44162
Name: vacant, dtype: int64
```

```
[ ]: assess1.groupby('vacant')['market_value'].sum()#vacant land is also less money
```

```
[ ]: vacant
0      1.649850e+11
1      5.372963e+09
Name: market_value, dtype: float64
```

```
[ ]: assess1.groupby('vacant')['depth'].mean()#higher mean depth meaning that are
      ↳deeper in
      #The depth is measured from the principal street back to the rear property line
      ↳or secondary street
```

```
[ ]: vacant
0      106.987503
1      139.259543
Name: depth, dtype: float64
```

```
[ ]: fig, ax = plt.subplots(figsize = (15, 15))
assess1.hist(ax = ax)
```

```
/var/folders/6p/wpw9qml57530xkxqkhprrf40000gn/T/ipykernel_74088/2015698519.py:2
: UserWarning: To output multiple subplots, the figure containing the passed
axes is being cleared
    assess1.hist(ax = ax)
```

```
[ ]: array([[<AxesSubplot:title={'center':'parcel_number'}>,
             <AxesSubplot:title={'center':'objectid'}>,
             <AxesSubplot:title={'center':'category_code'}>],
```

```
[<AxesSubplot:title={'center':'census_tract'}>,
 <AxesSubplot:title={'center':'depth'}>,
 <AxesSubplot:title={'center':'market_value'}>],
 [<AxesSubplot:title={'center':'zip_code'}>,
 <AxesSubplot:title={'center':'lat'}>,
 <AxesSubplot:title={'center':'lng'}>],
 [<AxesSubplot:title={'center':'vacant'}>, <AxesSubplot:>,
 <AxesSubplot:>], dtype=object)
```



0.2.17 City: Property Assessment History

<https://metadata.phila.gov/#home/datasetdetails/5543865f20583086178c4ee5/representationdetails/55d62f07ee9>

Property Assessment History(2015 - 2021), key takeaways: - Market value of properties has increased over the years

```
[ ]: assess_his = pd.read_csv('data/city/assessments.csv') #uploading dataset  
assess_his.head()
```

```
[ ]:    parcel_number  year  market_value  taxable_land  taxable_building  \  
0      11004720.0  2017      317300.0      78976.0          0.0  
1      11004720.0  2018      317300.0      78976.0          0.0  
2      11004720.0  2019     430000.0     107000.0          0.0  
3      11004720.0  2020     466000.0     115987.0     350013.0  
4      11004720.0  2021     466000.0     115987.0     350013.0  
  
exempt_land  exempt_building  
0            0.0        238324.0  
1            0.0        238324.0  
2            0.0        323000.0  
3            0.0          0.0  
4            0.0          0.0
```

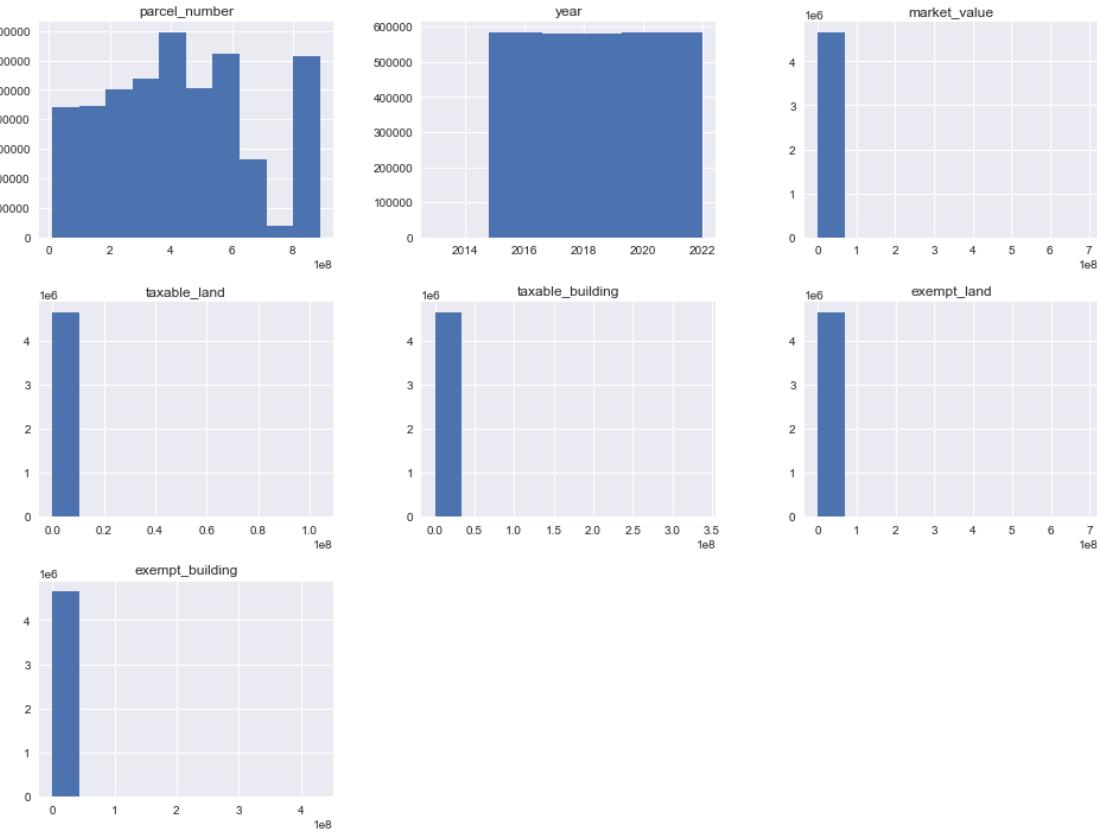
```
[ ]: assess_his.shape
```

```
[ ]: (4655571, 7)
```

```
[ ]: fig,ax = plt.subplots(figsize=(16,12))  
assess_his.hist(ax = ax)
```

```
/var/folders/6p/wpw9qml57530xkxqkkhprrf40000gn/T/ipykernel_74088/450088490.py:2:  
UserWarning: To output multiple subplots, the figure containing the passed axes  
is being cleared  
assess_his.hist(ax = ax)
```

```
[ ]: array([[<AxesSubplot:title={'center':'parcel_number'}>,  
           <AxesSubplot:title={'center':'year'}>,  
           <AxesSubplot:title={'center':'market_value'}>],  
          [<AxesSubplot:title={'center':'taxable_land'}>,  
           <AxesSubplot:title={'center':'taxable_building'}>,  
           <AxesSubplot:title={'center':'exempt_land'}>],  
          [<AxesSubplot:title={'center':'exempt_building'}>, <AxesSubplot:>,  
           <AxesSubplot:>]], dtype=object)
```



```
[ ]: assess_his.describe()
```

```
[ ]:      parcel_number      year  market_value  taxable_land \
count    4.655530e+06  4.655571e+06  4.655571e+06  4.655571e+06
mean    4.302874e+08  2.018500e+03  2.624731e+05  4.577635e+04
std     2.498826e+08  2.292846e+00  2.874228e+06  3.696039e+05
min     1.100060e+07  2.013000e+03  0.000000e+00  0.000000e+00
25%    2.322924e+08  2.016000e+03  6.350000e+04  9.600000e+03
50%    4.058826e+08  2.019000e+03  1.183000e+05  1.888500e+04
75%    5.831633e+08  2.021000e+03  1.970000e+05  3.980000e+04
max    8.888007e+08  2.022000e+03  6.907062e+08  1.040270e+08

      taxable_building  exempt_land  exempt_building
count    4.655568e+06  4.655571e+06  4.655571e+06
mean    1.294052e+05  1.764231e+04  6.964965e+04
std     1.393211e+06  1.253400e+06  1.706136e+06
min     0.000000e+00  0.000000e+00  0.000000e+00
25%    2.905875e+04  0.000000e+00  0.000000e+00
50%    7.076000e+04  0.000000e+00  0.000000e+00
75%   1.202000e+05  0.000000e+00  3.090000e+04
max    3.380388e+08  6.880302e+08  4.304176e+08
```

```
[ ]: assess_his['parcel_number'].nunique()
```

```
[ ]: 588600
```

```
[ ]: assess_his.groupby('year')['parcel_number'].count() # has data from 2013 - 2022
```

```
[ ]: year
2013      7
2014     17
2015  583289
2016  582115
2017  581392
2018  580297
2019  580828
2020  582030
2021  582531
2022  583024
Name: parcel_number, dtype: int64
```

```
[ ]: assess_his.groupby('year')['market_value'].sum() # value increased over the years
```

```
[ ]: year
2013    6.299200e+04
2014    2.586900e+06
2015    1.320876e+11
2016    1.334764e+11
2017    1.362923e+11
2018    1.491600e+11
2019    1.613413e+11
2020    1.680227e+11
2021    1.702639e+11
2022    1.713154e+11
Name: market_value, dtype: float64
```

```
[ ]: assess_his.isna().sum()
```

```
[ ]: parcel_number      41
year                  0
market_value          0
taxable_land          0
taxable_building      3
exempt_land           0
exempt_building       0
dtype: int64
```

```
[ ]: assess_his1 = assess_his[['parcel_number', 'year', 'market_value']]
assess_his1.head()
```

	parcel_number	year	market_value
0	11004720.0	2017	317300.0
1	11004720.0	2018	317300.0
2	11004720.0	2019	430000.0
3	11004720.0	2020	466000.0
4	11004720.0	2021	466000.0


```
[ ]: # making into wide format dataset with years as different columns so that it is easier to join the dataset
assess_his1 = assess_his1.pivot_table('market_value', ['parcel_number'], 'year').reset_index()
assess_his1.head()
```

year	parcel_number	2013	2014	2015	2016	2017	2018	\
0	11000600.0	NaN	NaN	127300.0	127300.0	127300.0	127300.0	
1	11000700.0	NaN	NaN	180400.0	180400.0	180400.0	180400.0	
2	11000800.0	NaN	NaN	179700.0	179700.0	179700.0	179700.0	
3	11000900.0	NaN	NaN	202000.0	202000.0	202000.0	202000.0	
4	11001000.0	NaN	NaN	192200.0	192200.0	192200.0	192200.0	

year	2019	2020	2021	2022
0	172400.0	186800.0	186800.0	186800.0
1	315200.0	341600.0	341600.0	341600.0
2	208800.0	226200.0	226200.0	226200.0
3	222900.0	241500.0	241500.0	241500.0
4	218700.0	237000.0	237000.0	237000.0


```
[ ]: assess_his1 = assess_his1[['parcel_number', 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]]
assess_his1.head()
```

year	parcel_number	2015	2016	2017	2018	2019	\
0	11000600.0	127300.0	127300.0	127300.0	127300.0	172400.0	
1	11000700.0	180400.0	180400.0	180400.0	180400.0	315200.0	
2	11000800.0	179700.0	179700.0	179700.0	179700.0	208800.0	
3	11000900.0	202000.0	202000.0	202000.0	202000.0	222900.0	
4	11001000.0	192200.0	192200.0	192200.0	192200.0	218700.0	

year	2020	2021	2022
0	186800.0	186800.0	186800.0
1	341600.0	341600.0	341600.0
2	226200.0	226200.0	226200.0
3	241500.0	241500.0	241500.0
4	237000.0	237000.0	237000.0

```
[ ]: assess_his1 = assess_his1.rename(columns={2015 : "market_value_2015",
                                             2016 : "market_value_2016",
                                             2017 : "market_value_2017",
                                             2018 : "market_value_2018",
                                             2019 : "market_value_2019",
                                             2020 : "market_value_2020",
                                             2021 : "market_value_2021",
                                             2022 : "market_value_2022"})

[ ]: assess_his1.head()

[ ]: year    parcel_number  market_value_2015  market_value_2016  market_value_2017 \
0        11000600.0          127300.0          127300.0          127300.0
1        11000700.0          180400.0          180400.0          180400.0
2        11000800.0          179700.0          179700.0          179700.0
3        11000900.0          202000.0          202000.0          202000.0
4        11001000.0          192200.0          192200.0          192200.0

year  market_value_2018  market_value_2019  market_value_2020 \
0        127300.0          172400.0          186800.0
1        180400.0          315200.0          341600.0
2        179700.0          208800.0          226200.0
3        202000.0          222900.0          241500.0
4        192200.0          218700.0          237000.0

year  market_value_2021  market_value_2022
0        186800.0          186800.0
1        341600.0          341600.0
2        226200.0          226200.0
3        241500.0          241500.0
4        237000.0          237000.0

[ ]: assess_his1.shape

[ ]: (588600, 9)
```

Joining all Property Assessment datasets Combining recent assessment data with historic market value of the assessment

```
[ ]: assess_full = assess1.merge(assess_his1, left_on = 'parcel_number', right_on = 'parcel_number', how= 'left')
assess_full = assess_full.drop("objectid", axis = 1)
assess_full.head()

[ ]:    parcel_number  category_code category_code_description  census_tract \
0        57127281             6                 Vacant Land      142.0
1        451106305            6                 Vacant Land      379.0
```

```

2      57166705      6          Vacant Land      142.0
3      56143925      6          Vacant Land      367.0
4      56143950      6          Vacant Land      367.0

      depth      location  market_value year_built  zip_code      lat \
0  439.00  968 N LAWRENCE ST        0.0        0.0  19123.0 -75.144757
1  256.00   2550 CASTOR AVE        0.0        0.0  19134.0 -75.092534
2  43.53    513 POPLAR ST        0.0        0.0  19123.0 -75.146154
3  68.83   428 FAIRMOUNT AVE        0.0        0.0  19123.0 -75.145586
4  68.83   442 FAIRMOUNT AVE        0.0        0.0  19123.0 -75.146035

      ...      geometry  vacant  market_value_2015 \
0 ... POINT (-75.14476 39.96785)      1        0.0
1 ... POINT (-75.09253 39.99146)      1        0.0
2 ... POINT (-75.14615 39.96707)      1        0.0
3 ... POINT (-75.14559 39.96268)      1        0.0
4 ... POINT (-75.14604 39.96277)      1        0.0

      market_value_2016  market_value_2017  market_value_2018  market_value_2019 \
0            0.0        0.0        0.0        0.0
1            0.0        0.0        0.0        0.0
2            0.0        0.0        0.0        0.0
3            0.0        0.0        0.0        0.0
4            0.0        0.0        0.0        0.0

      market_value_2020  market_value_2021  market_value_2022
0            0.0        0.0        0.0
1            0.0        0.0        0.0
2            0.0        0.0        0.0
3            0.0        0.0        0.0
4            0.0        0.0        0.0

[5 rows x 21 columns]

```

```
[ ]: assess_full.groupby('category_code_description').sum()
```

```

[ ]:      parcel_number  category_code  census_tract \
category_code_description
Commercial           11890049605332      56100    2761490.0
Industrial          3716250666438       21680    1139556.0
Mixed Use           12111520394823      43134    2256554.0
Multi Family        18864279637865      85010    8148805.0
Single Family       187122105053253     461869    91362135.0
Vacant Land         16412713085973      264972    7760457.0

      depth  market_value      zip_code \
category_code_description

```

Commercial	11873670.59	4.587975e+10	2.684972e+08
Industrial	1437334.85	4.555214e+09	8.296047e+07
Mixed Use	1222484.44	3.544653e+09	2.751066e+08
Multi Family	6960645.90	3.082025e+10	8.131507e+08
Single Family	35921738.13	8.013380e+10	8.837119e+09
Vacant Land	6136611.01	5.372963e+09	8.449007e+08
category_code_description		lat	lng
Commercial	-1.053918e+06	5.609277e+05	0
Industrial	-3.257617e+05	1.734108e+05	0
Mixed Use	-1.080627e+06	5.748748e+05	0
Multi Family	-3.194098e+06	1.700270e+06	0
Single Family	-3.470545e+07	1.847554e+07	0
Vacant Land	-3.319180e+06	1.765889e+06	44162
category_code_description		market_value_2015	market_value_2016
Commercial	3.857786e+10	3.877296e+10	
Industrial	3.284106e+09	3.284635e+09	
Mixed Use	3.072563e+09	3.071235e+09	
Multi Family	1.827870e+10	1.881354e+10	
Single Family	6.391587e+10	6.455961e+10	
Vacant Land	3.002924e+09	3.025281e+09	
category_code_description		market_value_2017	market_value_2018
Commercial	3.925187e+10	4.465314e+10	
Industrial	3.370658e+09	4.140864e+09	
Mixed Use	3.093522e+09	3.389155e+09	
Multi Family	1.974743e+10	2.480522e+10	
Single Family	6.599378e+10	6.689947e+10	
Vacant Land	3.422348e+09	4.265233e+09	
category_code_description		market_value_2019	market_value_2020
Commercial	4.442347e+10	4.508031e+10	
Industrial	4.372664e+09	4.479147e+09	
Mixed Use	3.454385e+09	3.517866e+09	
Multi Family	2.800641e+10	3.009656e+10	
Single Family	7.618227e+10	7.972939e+10	
Vacant Land	4.081793e+09	4.332333e+09	
category_code_description		market_value_2021	market_value_2022
Commercial	4.579015e+10	4.587838e+10	
Industrial	4.557987e+09	4.555214e+09	

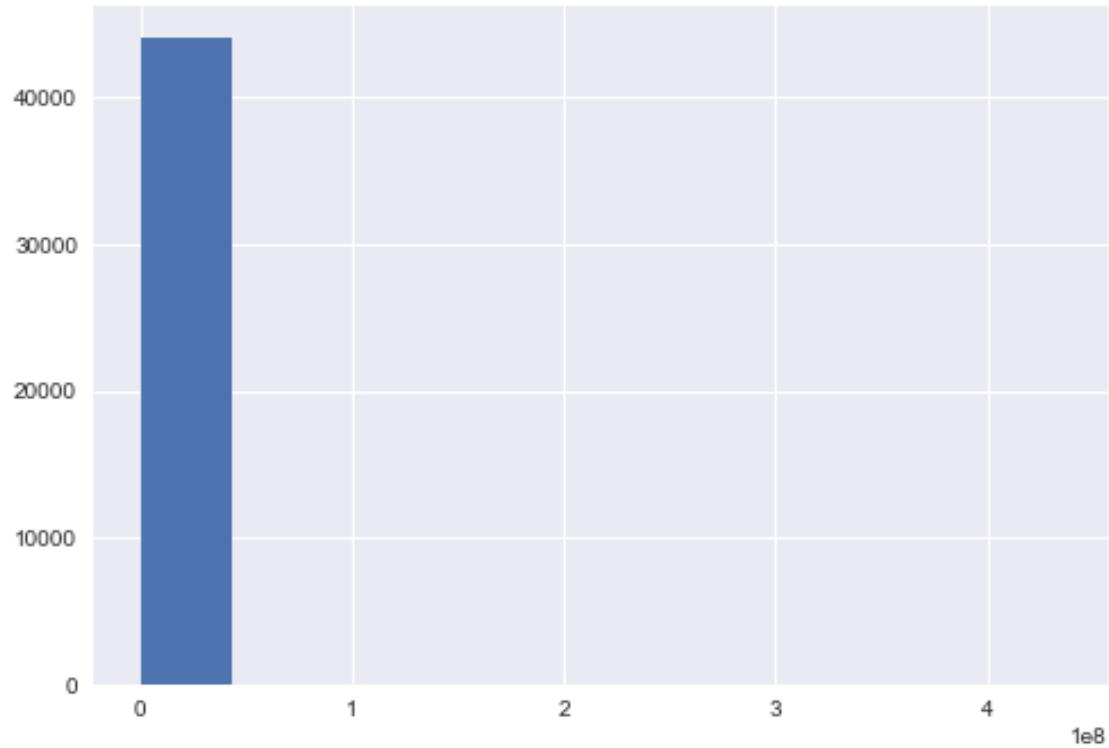
Mixed Use	3.541004e+09	3.544262e+09
Multi Family	3.066398e+10	3.081972e+10
Single Family	8.001237e+10	8.012833e+10
Vacant Land	4.749491e+09	5.368645e+09

```
[ ]: assess_full.groupby('category_code_description')['market_value'].describe() #  
↳market_value data is very skewed
```

	count	mean	std	min	\
category_code_description					
Commercial	14000.0	3.277125e+06	1.621596e+07	0.0	
Industrial	4330.0	1.052013e+06	4.989780e+06	0.0	
Mixed Use	14369.0	2.466875e+05	2.737034e+05	0.0	
Multi Family	42476.0	7.255920e+05	5.113219e+06	0.0	
Single Family	461794.0	1.735272e+05	1.945517e+05	0.0	
Vacant Land	44089.0	1.218663e+05	2.350904e+06	0.0	
	25%	50%	75%	max	
category_code_description					
Commercial	173000.0	422050.0	1370275.0	454197400.0	
Industrial	112225.0	269650.0	780275.0	245769000.0	
Mixed Use	97200.0	157000.0	283100.0	8322400.0	
Multi Family	121700.0	201800.0	351925.0	276892000.0	
Single Family	74600.0	130100.0	209900.0	17680000.0	
Vacant Land	6700.0	12100.0	30900.0	435000000.0	

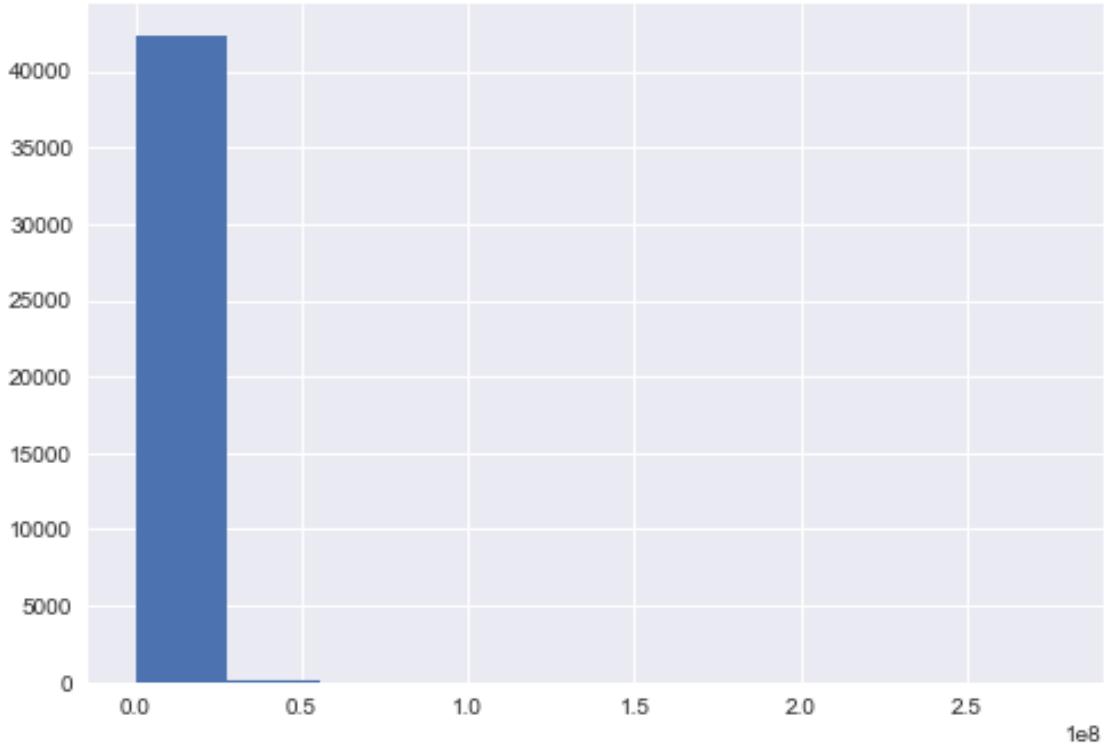
```
[ ]: assess_full.loc[assess_full['category_code_description'] == 'Vacant  
↳Land']['market_value'].hist()
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: assess_full.loc[assess_full['category_code_description'] == 'Multi\u2022Family']['market_value'].hist()
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: #assess_full.to_csv("assess_full.csv")
```

0.2.18 City of Philadelphia: 311 Calls

<https://metadata.phila.gov/#home/datasetdetails/5543864d20583086178c4e98/representationdetails/5762e19fa23>

Key takeaways: - 2021 data is not available on the website - Big percentage of the lat and long data was missing in this dataset - A lot of the service names were “Information Request”. - Most of the calls were for the Streets department - 2018 and 2020 had highest number of 311 calls - After grouping the number of 311 calls that happened within 50m of each parcel number, we noticed that the data is quite skewed. However, this is less skewed than the crime dataset.

Feature Engineering on this dataset: - Removed “Information Request” service as it was not related to vacant lot. Once we removed this, we noticed that the missing lat and lng data decreased a lot. - Removed lat and lng data that was outside of the Philadelphia lat/lng range. - Removed rows that didn’t have lat/lng data - Number of 311 calls that happened within 50m of each parcel number in the last 3.5 years (Jul 2018 - 2020) - Number of all 311 calls that happened within each parcel number in (2015 - 2020)

```
[ ]: #2021 data is not available
philly_311_1 = pd.read_csv("data/city/311Request_2015.csv")
philly_311_2 = pd.read_csv("data/city/311Request_2016.csv")
philly_311_3 = pd.read_csv("data/city/311Request_2017.csv")
philly_311_4 = pd.read_csv("data/city/311Request_2018.csv")
```

```

philly_311_5 = pd.read_csv("data/city/311Request_2019.csv")
philly_311_6 = pd.read_csv("data/city/311Request_2020.csv")

philly_311_all = [philly_311_1, philly_311_2, philly_311_3, philly_311_4, philly_311_5, philly_311_6]
philly_311 = pd.concat(philly_311_all)#combining all datasets

philly_311.head(10)

```

```

/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3457: DtypeWarning: Columns (12) have
mixed types.Specify dtype option on import or set low_memory=False.
exec(code_obj, self.user_global_ns, self.user_ns)

```

```

[ ]:   objectid  service_request_id  status      status_notes  \
0       32          8967043  Closed    Issue Resolved
1       39          8967052  Closed    Issue Resolved
2       40          8967049  Closed  Information Provided
3       41          8967048  Closed    Issue Resolved
4       92          8997221  Closed            NaN
5       93          8997222  Closed  Question Answered
6       94          8997223  Closed  Question Answered
7       95          8997224  Closed  Question Answered
8       96          8997225  Closed  Question Answered
9       97          8997226  Closed  Question Answered

           service_name  service_code  agency_responsible  \
0  Graffiti Removal     SR-CL01  Community Life Improvement Program
1  Graffiti Removal     SR-CL01  Community Life Improvement Program
2  Graffiti Removal     SR-CL01  Community Life Improvement Program
3  Graffiti Removal     SR-CL01  Community Life Improvement Program
4  Information Request    SR-IR01  Licenses & Inspections- L&I
5  Information Request    SR-IR01  Philly311 Contact Center
6  Information Request    SR-IR01  Philly311 Contact Center
7  Information Request    SR-IR01  Philly311 Contact Center
8  Information Request    SR-IR01        Free Library
9  Information Request    SR-IR01  Philly311 Contact Center

     service_notice  requested_datetime  updated_datetime  \
0  7 Business Days  2015-01-11 10:45:10  2015-08-12 03:47:02
1  7 Business Days  2015-01-11 12:15:21  2015-08-12 03:47:02
2  7 Business Days  2015-01-11 12:15:10  2015-08-12 03:47:02
3  7 Business Days  2015-01-11 12:15:06  2015-08-12 03:47:02
4        None  2015-01-30 19:15:02  2015-01-30 19:17:11
5        None  2015-01-30 19:25:35  2015-02-13 06:17:03
6        None  2015-01-30 19:29:00  2015-02-13 06:17:03

```

```

7      None 2015-01-30 19:30:05 2015-01-30 19:30:06
8      None 2015-01-30 19:30:55 2015-01-30 19:33:31
9      None 2015-01-30 19:31:20 2015-01-30 19:31:21

      expected_datetime          address zipcode \
0 2015-01-19 19:00:00      708 S 50TH ST 19143
1 2015-01-19 19:00:00      711 TASKER ST 19147
2 2015-01-19 19:00:00 7223 LANSDOWNE AVE 19151
3 2015-01-19 19:00:00      711 DICKINSON ST 19147
4           NaN             NaN 19111
5           NaN             NaN   NaN
6           NaN             NaN   NaN
7           NaN             NaN   NaN
8           NaN             NaN 19141
9           NaN             NaN   NaN

      media_url      lat      lon
0           NaN 39.947292 -75.222567
1 https://d21tc4b3k3r3vo.cloudfront.net/uploads/... 39.929033 -75.157283
2           NaN 39.973162 -75.259006
3           NaN 39.930267 -75.157078
4           NaN       NaN       NaN
5           NaN       NaN       NaN
6           NaN       NaN       NaN
7           NaN       NaN       NaN
8           NaN       NaN       NaN
9           NaN       NaN       NaN

```

[]: philly_311.columns

```

[ ]: Index(['objectid', 'service_request_id', 'status', 'status_notes',
       'service_name', 'service_code', 'agency_responsible', 'service_notice',
       'requested_datetime', 'updated_datetime', 'expected_datetime',
       'address', 'zipcode', 'media_url', 'lat', 'lon'],
       dtype='object')

```

[]: philly_311.dtypes

```

[ ]: objectid          int64
service_request_id    int64
status               object
status_notes         object
service_name          object
service_code          object
agency_responsible   object
service_notice        object
requested_datetime   object

```

```
updated_datetime      object
expected_datetime    object
address              object
zipcode              object
media_url            object
lat                  float64
lon                  float64
dtype: object
```

```
[ ]: philly_311.isna().sum()
```

```
objectid              0
service_request_id    0
status                0
status_notes          1853014
service_name           0
service_code           390556
agency_responsible   89
service_notice         1828945
requested_datetime    0
updated_datetime       0
expected_datetime     427727
address               2380315
zipcode               3253168
media_url              3548515
lat                   2425451
lon                   2425451
dtype: int64
```

```
[ ]: philly_311.isna().sum()/philly_311.shape[0] # a big percentage of latitude and longitude data is missing
# we can also remove status, status note, service notice, address, zipcode, media url
```

```
objectid              0.000000
service_request_id    0.000000
status                0.000000
status_notes          0.495050
service_name           0.000000
service_code           0.104341
agency_responsible   0.000024
service_notice         0.488620
requested_datetime    0.000000
updated_datetime       0.000000
expected_datetime     0.114271
address               0.635924
zipcode               0.869115
```

```
media_url          0.948020
lat                0.647982
lon                0.647982
dtype: float64
```

```
[ ]: philly_311.describe(include = 'all')
```

	objectid	service_request_id	status	status_notes	\
count	3.743082e+06	3.743082e+06	3743082	1890068	
unique	NaN	NaN	2	36	
top	NaN	NaN	Closed	Question Answered	
freq	NaN	NaN	3484297	1392224	
mean	4.556441e+06	1.171150e+07	NaN	NaN	
std	3.001763e+06	1.373912e+06	NaN	NaN	
min	2.000000e+00	8.955949e+06	NaN	NaN	
25%	1.650116e+06	1.086908e+07	NaN	NaN	
50%	4.583862e+06	1.184317e+07	NaN	NaN	
75%	7.291064e+06	1.284088e+07	NaN	NaN	
max	1.159748e+07	1.384723e+07	NaN	NaN	
	service_name	service_code	agency_responsible	service_notice	\
count	3743082	3352526	3742993	1914137	
unique	74	48	96	39	
top	Information Request	SR-IR01	Streets Department	None	
freq	2017395	2017395	865394	765375	
mean	NaN	NaN	NaN	NaN	
std	NaN	NaN	NaN	NaN	
min	NaN	NaN	NaN	NaN	
25%	NaN	NaN	NaN	NaN	
50%	NaN	NaN	NaN	NaN	
75%	NaN	NaN	NaN	NaN	
max	NaN	NaN	NaN	NaN	
	requested_datetime	updated_datetime	expected_datetime	\	
count	3743082	3743082	3315355		
unique	3631083	2965606	45099		
top	2015-12-08 14:34:26	2018-06-13 11:11:05	2020-07-14 20:00:00		
freq	16	873	2146		
mean	NaN	NaN	NaN		
std	NaN	NaN	NaN		
min	NaN	NaN	NaN		
25%	NaN	NaN	NaN		
50%	NaN	NaN	NaN		
75%	NaN	NaN	NaN		
max	NaN	NaN	NaN		
	address zipcode		media_url	\	

```

count    1362767  489914                               194567
unique   450561    5366                               184193
top          .    19143  https://d17aqltn7cihbm.cloudfront.net/uploads/...
freq      2802    24672                                24
mean        NaN     NaN                               NaN
std         NaN     NaN                               NaN
min         NaN     NaN                               NaN
25%        NaN     NaN                               NaN
50%        NaN     NaN                               NaN
75%        NaN     NaN                               NaN
max        NaN     NaN                               NaN

                    lat          lon
count  1.317631e+06  1.317631e+06
unique   NaN          NaN
top          NaN          NaN
freq          NaN          NaN
mean    3.999077e+01 -7.515214e+01
std     4.883178e-02  8.234499e-02
min     3.179832e+01 -1.230819e+02
25%    3.995091e+01 -7.518467e+01
50%    3.998652e+01 -7.515971e+01
75%    4.002810e+01 -7.512257e+01
max     4.928218e+01 -7.495826e+01

```

[]: philly_311.shape

[]: (3743082, 16)

[]: philly_311['service_name'].nunique()

[]: 74

[]: philly_311['service_name'].unique()

[]: array(['Graffiti Removal', 'Information Request', 'Smoke Detector',
 'Abandoned Vehicle', 'Vacant House or Commercial',
 'Street Light Outage', 'Rubbish/Recyclable Material Collection',
 'Illegal Dumping', 'Maintenance Residential or Commercial',
 'Construction Site Task Force', 'Traffic Signal Emergency',
 'Traffic (Other)', 'License Residential', 'Manhole Cover',
 'Dangerous Sidewalk', 'Inlet Cleaning', 'No Heat (Residential)',
 'Dead Animal in Street', 'Street Defect', 'Building Construction',
 'Infestation Residential', 'Sanitation / Dumpster Violation',
 'Other (Streets)', 'Fire Residential or Commercial',
 'Police Complaint', 'Zoning Residential', 'Zoning Business',
 'Boarding Room House', 'Shoveling', 'Line Striping'],

```
'Building Dangerous', 'Complaints against Fire or EMS',
'Miscellaneous', 'Vacant Lot Clean-Up', 'Directory Assistance',
'Stop Sign Repair', 'Alley Light Outage', 'Street Trees',
'Abandoned Bike', 'Complaint (Streets)', 'Street Paving',
'Tree Dangerous', 'Daycare Residential or Commercial', 'Salting',
'Hydrant Request', 'Other Dangerous', 'Newsstand Outdoor Cafe',
'Parks and Rec Safety and Maintenance',
'Hydrant Knocked Down (No Water)', 'Building Force',
'No Heat Residential', 'Emergency Air Conditioning',
'Newsstand/Outdoor Cafe', 'LI Escalation', 'KB Escalations',
'Right of Way Unit', 'Revenue Escalation', 'Right-of-Way',
'Agency Receivables', 'Service Day', 'Homeless Encampment',
'Complaint against Fire or EMS', 'Maintenance Complaint',
'Construction Complaints', 'License Complaint',
'Fire Safety Complaint', 'Dangerous Building Complaint',
'Eclipse Help', 'Social Distancing Complaint',
'Homeless Encampment Request', 'License_Complaint',
'Fire_Safety_Complaint', 'Social Distancing Compliant',
'Opioid Response Unit'], dtype=object)
```

```
[ ]: philly_311['service_name'].value_counts()
```

Information Request	2017395
Miscellaneous	170424
Rubbish/Recyclable Material Collection	167547
Revenue Escalation	147551
Maintenance Residential or Commercial	130608
...	
Emergency Air Conditioning	3
Newsstand/Outdoor Cafe	2
Fire_Safety_Complaint	2
Social Distancing Compliant	2
Complaint against Fire or EMS	1

Name: service_name, Length: 74, dtype: int64

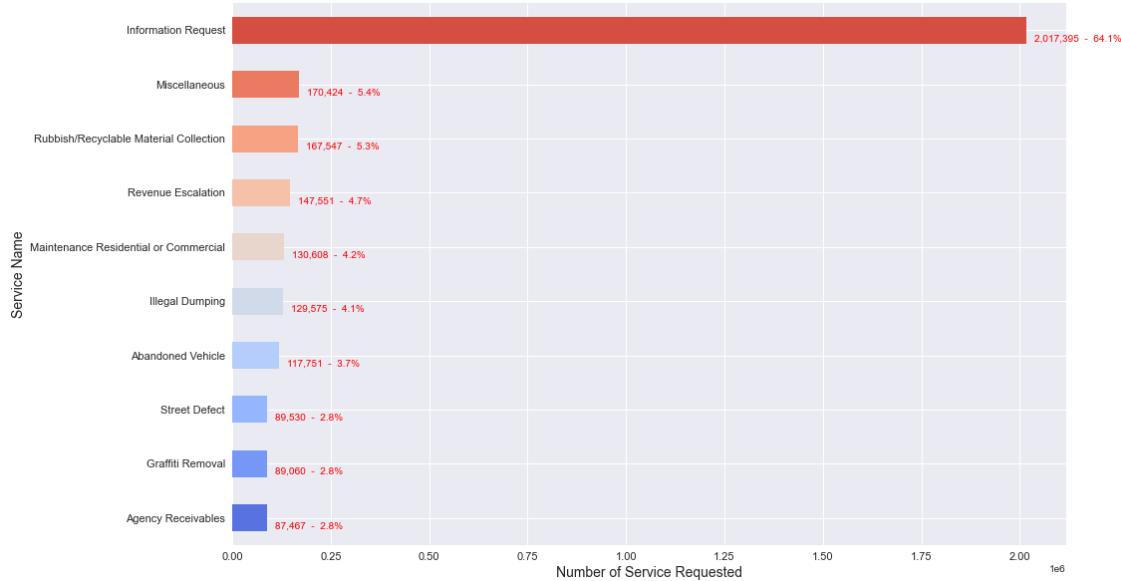
```
[ ]: #plotting bar chart based on service(name) type requested
philly_service = philly_311['service_name'].value_counts().head(10).
    ↪sort_values()
philly_service.plot(kind='barh', figsize=(15,10), fontsize=11, color=sns.
    ↪color_palette('coolwarm', len(philly_service)))
plt.ylabel('Service Name', fontsize = 14)
plt.xlabel('Number of Service Requested', fontsize = 14)
```

```
# Include the number of service name and the corresponding percentage for every
↪type
```

```

for index, value in enumerate(philly_service):
    label = str(format(int(value), ',')) + ' - {}'.format(round( (value/
→philly_service.sum())*100, 1))
    plt.annotate(label, xy = (value + 20000, index - 0.2 ), color = 'red')

```

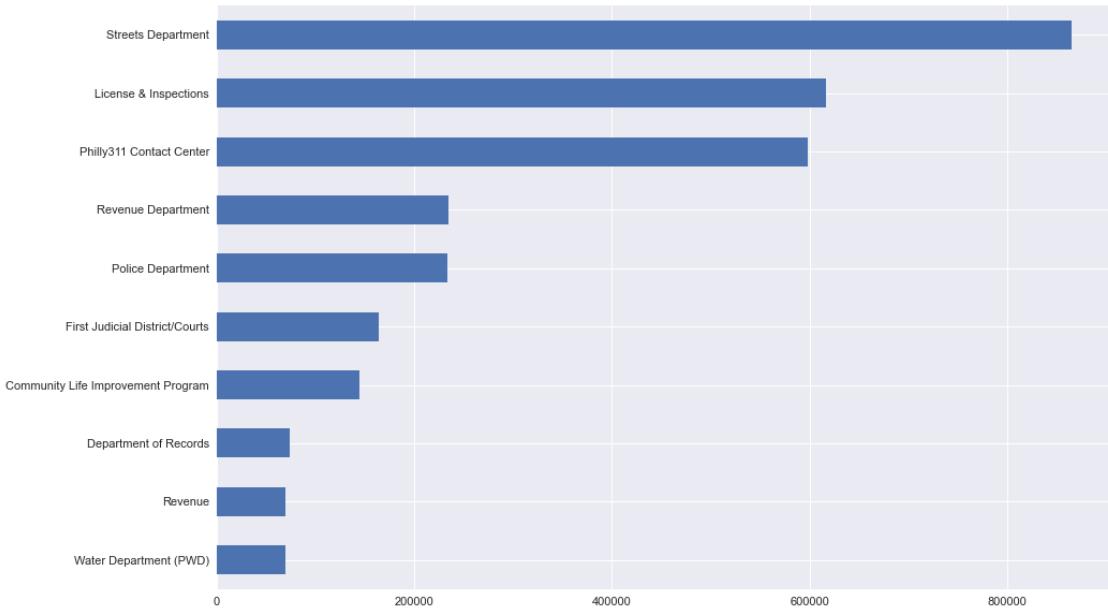


```

[ ]: #Agencies responsible for 311 calls. Most of the calls are for the Street
     ↪Department
philly_agency = philly_311['agency_responsible'].value_counts().head(10).
     ↪sort_values()
philly_agency.plot(kind='barh', figsize=(15,10), fontsize=11)

```

```
[ ]: <AxesSubplot:>
```



```
[ ]: #creating geometry column based on longitude and latitude data
crs= {'init': 'epsg:4326'}
geometry = [Point(xy) for xy in zip(philly_311["lon"],philly_311["lat"])]
```

```
philly_311 = gpd.GeoDataFrame(philly_311,
                               crs = crs,
                               geometry = geometry)
```

```
philly_311.head()
```

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/pyproj/crs/crs.py:131: FutureWarning: '+init=<authority>:<code>' syntax
is deprecated. '<authority>:<code>' is the preferred initialization method. When
making the change, be mindful of axis order changes:
https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-
proj-6
in_crs_string = _prepare_from_proj_string(in_crs_string)
```

```
[ ]:   objectid  service_request_id  status      status_notes \
0        32          8967043  Closed    Issue Resolved
1        39          8967052  Closed    Issue Resolved
2        40          8967049  Closed  Information Provided
3        41          8967048  Closed    Issue Resolved
4        92          8997221  Closed            NaN
```



```
           service_name  service_code  agency_responsible \
0  Graffiti Removal       SR-CL01  Community Life Improvement Program
```

```

1      Graffiti Removal      SR-CL01  Community Life Improvement Program
2      Graffiti Removal      SR-CL01  Community Life Improvement Program
3      Graffiti Removal      SR-CL01  Community Life Improvement Program
4  Information Request      SR-IR01      Licenses & Inspections- L&I

      service_notice  requested_datetime  updated_datetime \
0  7 Business Days  2015-01-11 10:45:10  2015-08-12 03:47:02
1  7 Business Days  2015-01-11 12:15:21  2015-08-12 03:47:02
2  7 Business Days  2015-01-11 12:15:10  2015-08-12 03:47:02
3  7 Business Days  2015-01-11 12:15:06  2015-08-12 03:47:02
4            None  2015-01-30 19:15:02  2015-01-30 19:17:11

      expected_datetime          address zipcode \
0  2015-01-19 19:00:00    708 S 50TH ST  19143
1  2015-01-19 19:00:00    711 TASKER ST  19147
2  2015-01-19 19:00:00  7223 LANSDOWNE AVE  19151
3  2015-01-19 19:00:00    711 DICKINSON ST  19147
4                NaN           NaN  19111

      media_url          lat          lon \
0                  NaN  39.947292 -75.222567
1  https://d21tc4b3k3r3vo.cloudfront.net/uploads/...  39.929033 -75.157283
2                  NaN  39.973162 -75.259006
3                  NaN  39.930267 -75.157078
4                  NaN       NaN       NaN

      geometry
0  POINT (-75.22257 39.94729)
1  POINT (-75.15728 39.92903)
2  POINT (-75.25901 39.97316)
3  POINT (-75.15708 39.93027)
4        POINT EMPTY

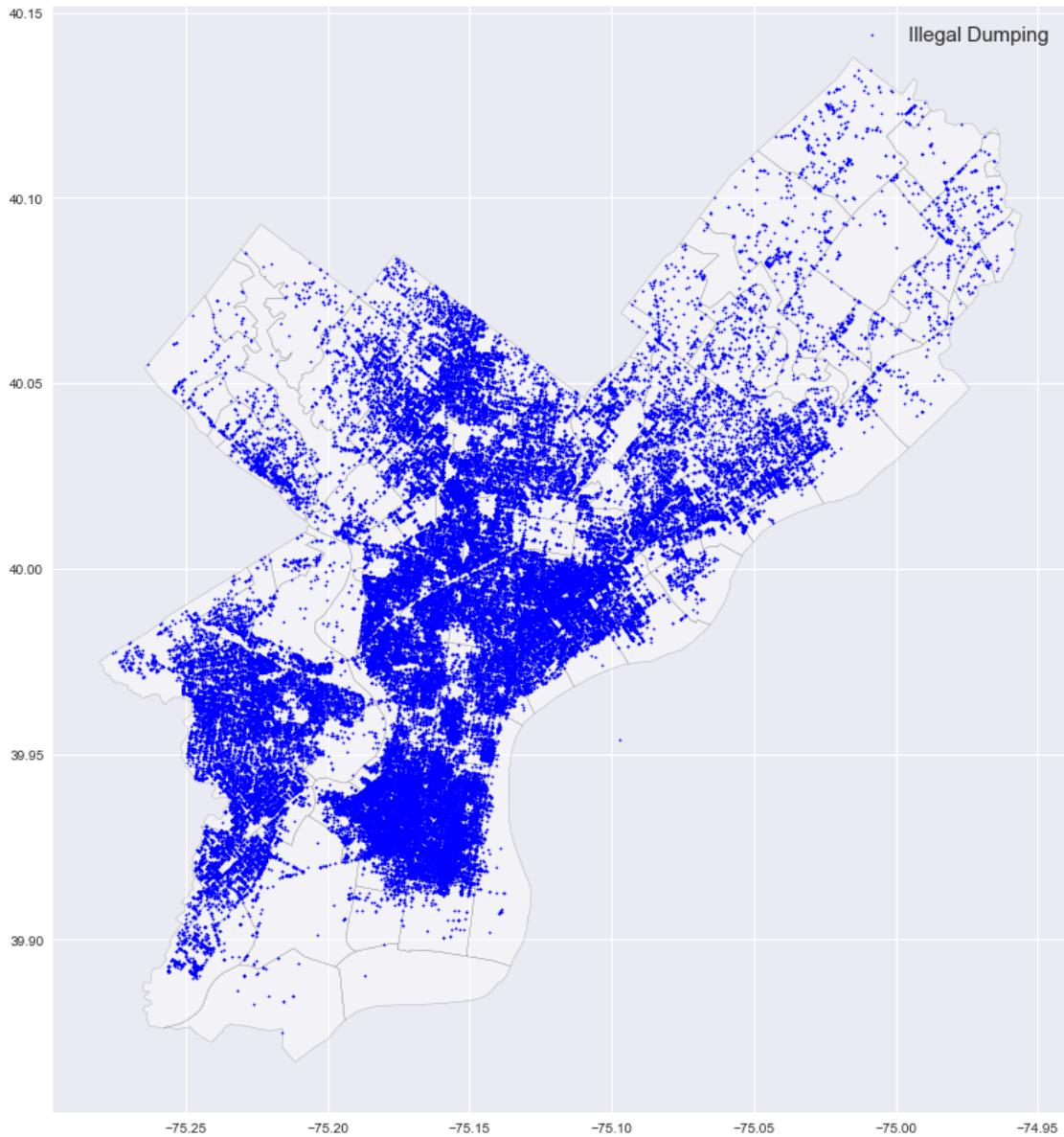
```

```

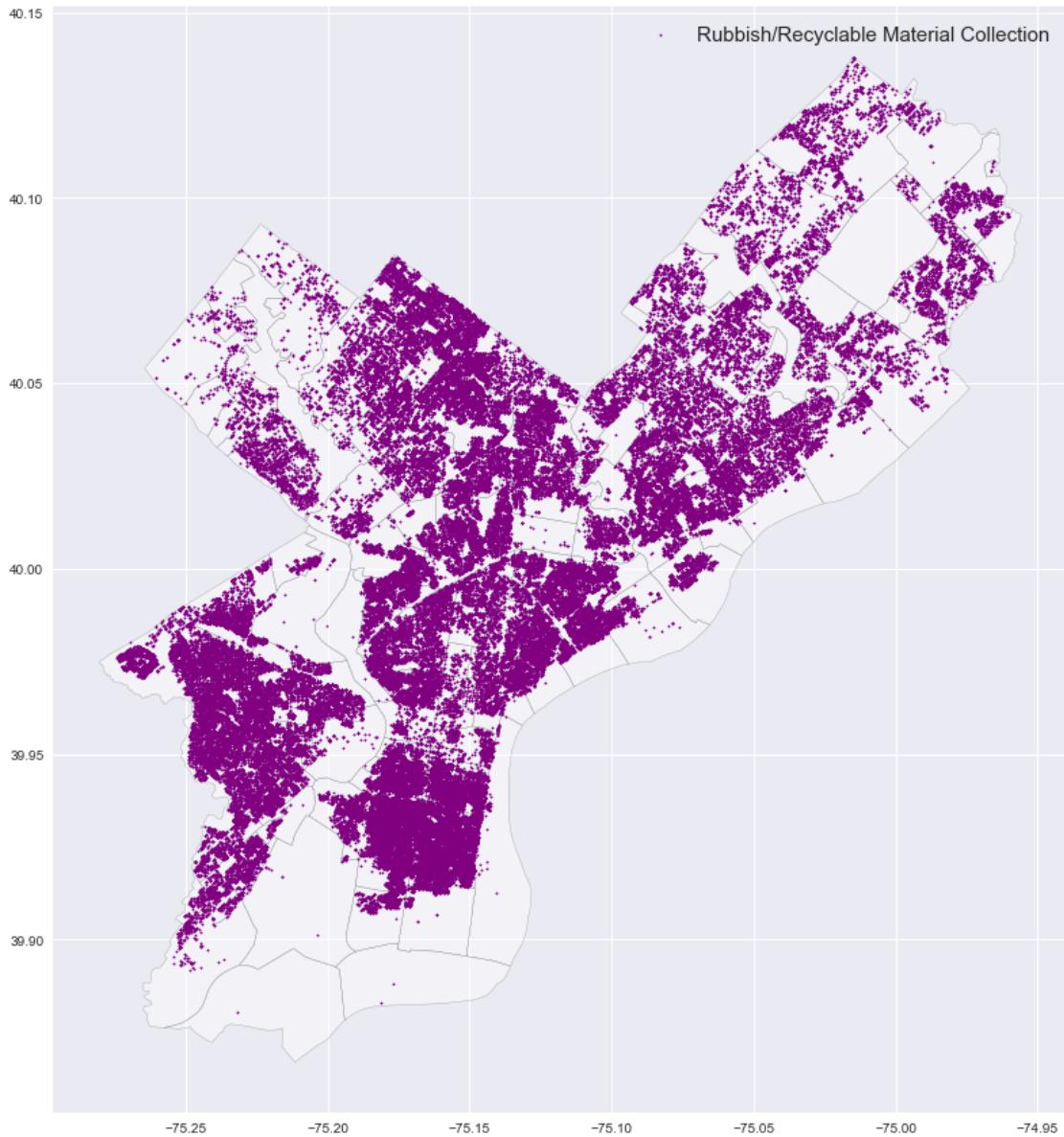
[ ]: #plotting for Illegal Dumping service
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white",
edgecolor='black')
philly_311[philly_311['service_name'] == 'Illegal Dumping'].plot(ax = ax,
markersize = 2, color = "blue", marker = "o", label = "Illegal Dumping")

plt.legend(prop = {'size' : 15})
plt.show()

```

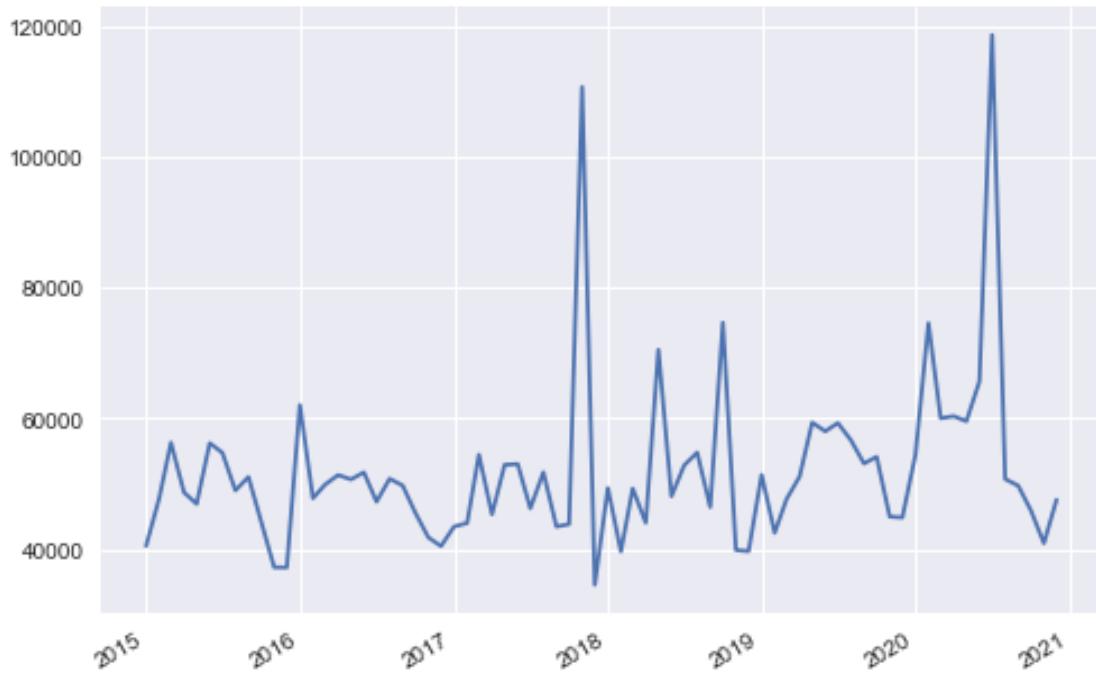


```
[ ]: #plotting for Rubbish/Recyclable Material Collection service
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white",
                                         edgecolor='black')
philly_311[philly_311['service_name'] == 'Rubbish/Recyclable Material
                                         Collection'].plot(ax = ax, markersize = 2, color = "purple", marker = "o",
                                         label = "Rubbish/Recyclable Material Collection")
plt.legend(prop = {'size' : 15})
plt.show()
```



```
[ ]: philly_311['requested_datetime'] = pd.  
    →to_datetime(philly_311['requested_datetime'], format = "%Y-%m")  
philly_311['requested_yeарmonth'] = pd.  
    →to_datetime(philly_311['requested_datetime'].dt.strftime('%Y-%m'))  
philly_311['requested_yeарmonth'].value_counts().plot()# there are some  
    →significant peaks in 311 calls in 2018 and middle of 2020
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: # if I remove Information request then the percentage of unknown latitude and longitude decreases
      ↵philly_311.loc[~philly_311['service_name'].str.contains("Information Request")].
      ↵reset_index(drop = True).isna().sum()/philly_311.
      ↵loc[~philly_311['service_name'].str.contains("Information Request")].
      ↵reset_index(drop = True).shape[0]
```

objectid	0.000000
service_request_id	0.000000
status	0.000000
status_notes	0.711450
service_name	0.000000
service_code	0.226319
agency_responsible	0.000003
service_notice	0.295555
requested_datetime	0.000000
updated_datetime	0.000000
expected_datetime	0.013318
address	0.238014
zipcode	0.847535
media_url	0.893389
lat	0.262308
lon	0.262308
geometry	0.000000
requested_yearmonth	0.000000

```

dtype: float64

[ ]: philly_311.loc[~philly_311['service_name'].str.contains("Information Request")].
    ↪reset_index(drop = True).shape

[ ]: (1725687, 18)

[ ]: philly_311_full = philly_311.loc[~philly_311['service_name'].str.
    ↪contains("Information Request")].reset_index(drop = True)

[ ]: philly_311_full.columns

[ ]: Index(['objectid', 'service_request_id', 'status', 'status_notes',
    'service_name', 'service_code', 'agency_responsible', 'service_notice',
    'requested_datetime', 'updated_datetime', 'expected_datetime',
    'address', 'zipcode', 'media_url', 'lat', 'lon', 'geometry',
    'requested_yeарmonth'],
    dtype='object')

[ ]: philly_311_full['objectid'].nunique()

[ ]: 1725687

[ ]: philly_311_full['service_request_id'].nunique()

[ ]: 1725355

[ ]: philly_311_full['agency_responsible'].nunique()

[ ]: 49

[ ]: philly_311_full = philly_311_full.drop(['status', 'status_notes', ↪
    'service_notice', 'address', 'zipcode',
    'media_url', 'agency_responsible', 'objectid', ↪
    'requested_datetime',
    'updated_datetime', 'expected_datetime', 'service_code'], ↪
    axis = 1)

[ ]: philly_311_full.isna().sum()

[ ]: service_request_id      0
    service_name            0
    lat                      452661
    lon                      452661
    geometry                 0
    requested_yeарmonth     0
    dtype: int64

```

```
[ ]: #remove null values of lat and lng data
philly_311_full = philly_311_full.dropna(how='any',axis=0)

[ ]: philly_311_full.isna().sum()

[ ]: service_request_id      0
    service_name            0
    lat                      0
    lon                      0
    geometry                 0
    requested_yeарmonth     0
    dtype: int64

[ ]: print(philly_311_full.loc[philly_311_full['lon']<-81].shape) #size of dataset
    ↪after with less than -81 as longitude. This is the irregular data.
print(philly_311_full.loc[philly_311_full['lon']>-81].shape) #size of dataset
    ↪after with more than -81 as longitude.
```

(2, 6)
(1273024, 6)

```
[ ]: philly_311_full = philly_311_full.loc[philly_311_full['lon']>-81].
    ↪reset_index(drop=True) #dropping all irregular longitude and latitude
philly_311_full.shape
```

(1273024, 6)

```
[ ]: #philly_311_full.to_csv("philly_311_full.csv", index = False)
```

0.2.19 Combining dataframes that has parcel number and opa account number.

Parcel_number is a unique nine-digit parcel identifier/account number created by the Board of Revision of Taxes Staff to identify a specific property.

This field can be used as a join key for various city datasets. It is also sometimes known as the ‘BRT’ number or within some datasets coming directly from OPA is also called the ‘PARCEL_NUMBER’. (Other departments may use different internal ‘Parcel Numbers’, so please verify before using).

```
[ ]: assess_full = pd.read_csv("assess_full.csv", index_col=0) # property assessment
    ↪and property assessment history file
assess_full = assess_full.add_suffix('_assess') # adding assess in the end of
    ↪column to identify later where they are coming from
assess_full.head()
```

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3457: DtypeWarning: Columns (8) have
mixed types. Specify dtype option on import or set low_memory=False.
exec(code_obj, self.user_global_ns, self.user_ns)
```

```
[ ]:    parcel_number_assess  category_code_assess  \
0          57127281              6
1          451106305             6
2          57166705              6
3          56143925              6
4          56143950              6

category_code_description_assess  census_tract_assess  depth_assess  \
0           Vacant Land        142.0      439.00
1           Vacant Land        379.0      256.00
2           Vacant Land        142.0      43.53
3           Vacant Land        367.0      68.83
4           Vacant Land        367.0      68.83

location_assess  market_value_assess year_built_assess  zip_code_assess  \
0  968 N LAWRENCE ST            0.0          0.0      19123.0
1  2550 CASTOR AVE            0.0          0.0      19134.0
2  513 POPLAR ST              0.0          0.0      19123.0
3  428 FAIRMOUNT AVE          0.0          0.0      19123.0
4  442 FAIRMOUNT AVE          0.0          0.0      19123.0

lat_assess  ...                  geometry_assess  \
0 -75.144757  ...      POINT (-75.144756917366 39.96784709341962)
1 -75.092534  ...      POINT (-75.09253430091161 39.99145897715201)
2 -75.146154  ...      POINT (-75.14615444716026 39.96706685767916)
3 -75.145586  ...      POINT (-75.14558649490823 39.96267918822164)
4 -75.146035  ...      POINT (-75.1460354483123 39.96277166528522)

vacant_assess  market_value_2015_assess  market_value_2016_assess  \
0          1                  0.0          0.0
1          1                  0.0          0.0
2          1                  0.0          0.0
3          1                  0.0          0.0
4          1                  0.0          0.0

market_value_2017_assess  market_value_2018_assess  \
0          0.0                  0.0
1          0.0                  0.0
2          0.0                  0.0
3          0.0                  0.0
4          0.0                  0.0

market_value_2019_assess  market_value_2020_assess  \
0          0.0                  0.0
1          0.0                  0.0
2          0.0                  0.0
3          0.0                  0.0
```

```
4          0.0          0.0  
  
  market_value_2021_assess  market_value_2022_assess  
0          0.0          0.0  
1          0.0          0.0  
2          0.0          0.0  
3          0.0          0.0  
4          0.0          0.0
```

[5 rows x 21 columns]

```
[ ]: assess_full.dtypes
```

```
[ ]: parcel_number_assess           int64  
category_code_assess             int64  
category_code_description_assess object  
census_tract_assess            float64  
depth_assess                   float64  
location_assess                object  
market_value_assess             float64  
year_built_assess              object  
zip_code_assess                float64  
lat_assess                     float64  
lng_assess                     float64  
geometry_assess                object  
vacant_assess                 int64  
market_value_2015_assess        float64  
market_value_2016_assess        float64  
market_value_2017_assess        float64  
market_value_2018_assess        float64  
market_value_2019_assess        float64  
market_value_2020_assess        float64  
market_value_2021_assess        float64  
market_value_2022_assess        float64  
dtype: object
```

```
[ ]: tax_final = pd.read_csv("tax_delinquent.csv", index_col=0) # tax delinquency  
    ↪ file  
tax_final.opa_number = tax_final.opa_number.astype(int) # turning to integer  
    ↪ value  
tax_final = tax_final.add_suffix('_tax') # adding tax in the end of column to  
    ↪ identify later where they are coming from  
tax_final.head()
```

```
[ ]:   opa_number_tax street_address_tax  zip_code_tax  num_years_owed_tax  \  
0      493169300      6045 N CAMAC ST          19141.0            23  
1      493179100      5620 N CAMAC ST          19141.0            1
```

```

2      493180700    5714 N CAMAC ST        19141.0      1
3      493183600    5812 N CAMAC ST        19141.0      5
4      223166200    420 GLEN ECHO RD       19119.0      1

      most_recent_year_owed_tax  oldest_year_owed_tax \
0                  2021           1994
1                  2021           2021
2                  2021           2021
3                  2021           2017
4                  2021           2021

      year_of_last_assessment_tax general_building_description_tax \
0                      2021.0             house
1                      2021.0             house
2                      2021.0             house
3                      2021.0             house
4                      2021.0             house

      building_category_tax  is_actionable_tax  bankruptcy_tax sheriff_sale_tax \
0          residential        False            False            N
1          residential        False            False            N
2          residential        False            False            N
3          residential        False            False            N
4          residential        False            False            N

      sequestration_enforcement_tax  payment_agreement_tax  principal_due_tax \
0                      False              True            12200.18
1                      False              True            -0.05
2                      False             False            895.87
3                      False              True            4536.94
4                      False             False            4224.60

      total_due_tax  total_assessment_tax      lat_tax      lng_tax \
0      30670.98            111400.0 -75.140099  40.045081
1      41.00              111200.0 -75.141930  40.039007
2     1016.81            109000.0 -75.141727  40.039940
3     6691.70            110600.0 -75.141395  40.041404
4     4794.92            346800.0 -75.195309  40.051563

      geometry_tax
0  POINT (-75.14009862350571 40.04508127195381)
1  POINT (-75.1419299030054 40.0390067239218)
2  POINT (-75.14172656588731 40.03993981997084)
3  POINT (-75.14139543574684 40.04140373648085)
4  POINT (-75.19530918746055 40.05156297646704)

```

[]: tax_final.dtypes

```
[ ]: opa_number_tax           int64
street_address_tax          object
zip_code_tax                 float64
num_years_owed_tax          int64
most_recent_year_owed_tax   int64
oldest_year_owed_tax        int64
year_of_last_assessment_tax float64
general_building_description_tax object
building_category_tax       object
is_actionable_tax           bool
bankruptcy_tax               bool
sheriff_sale_tax             object
sequestration_enforcement_tax bool
payment_agreement_tax       bool
principal_due_tax           float64
total_due_tax                float64
total_assessment_tax        float64
lat_tax                      float64
lng_tax                      float64
geometry_tax                 object
dtype: object
```

```
[ ]: print("Unique number of OPA in tax dataset", tax_final['opa_number_tax'].  
        →nunique())
print("Shape of tax dataset", tax_final.shape)
print("Unique number of OPA in assess dataset",  
      →assess_full['parcel_number_assess'].nunique())
print("Shape of assess dataset", assess_full.shape)
#mismatch between number of rows and unique number of opa values in the assess  
→data set so we have to explore the assessment dataset
```

Unique number of OPA in tax dataset 72659
 Shape of tax dataset (72659, 20)
 Unique number of OPA in assess dataset 581343
 Shape of assess dataset (581352, 21)

```
[ ]: assess_full.columns
```

```
[ ]: Index(['parcel_number_assess', 'category_code_assess',
           'category_code_description_assess', 'census_tract_assess',
           'depth_assess', 'location_assess', 'market_value_assess',
           'year_built_assess', 'zip_code_assess', 'lat_assess', 'lng_assess',
           'geometry_assess', 'vacant_assess', 'market_value_2015_assess',
           'market_value_2016_assess', 'market_value_2017_assess',
           'market_value_2018_assess', 'market_value_2019_assess',
           'market_value_2020_assess', 'market_value_2021_assess',
           'market_value_2022_assess'],
```

```

    dtype='object')

[ ]: assess_full.
    ↪groupby(['parcel_number_assess'])['category_code_description_assess'].
    ↪count().sort_values(ascending=False).head(7)
#parcen numbers that are being recorded twice

[ ]: parcel_number_assess
365349205      4
888182184      2
888182180      2
888182144      2
888182142      2
888182188      2
888182186      2
Name: category_code_description_assess, dtype: int64

[ ]: assess_full.loc[assess_full['parcel_number_assess'].isin([365349205, 888182184, ↪
    ↪888182180, 888182144, 888182142, 888182188, 888182186])].sort_values(by=↪
    ↪='parcel_number_assess')
#the values looks like they might be mistake. as this is few rows, we will ↪
    ↪delete them

[ ]:      parcel_number_assess  category_code_assess  \
54433            365349205                  6
565290           365349205                  1
565291           365349205                  1
565292           365349205                  6
571356           888182142                  2
571592           888182142                  2
571355           888182144                  2
571594           888182144                  2
571588           888182180                  2
571589           888182180                  2
571358           888182184                  2
571590           888182184                  2
571359           888182186                  2
571593           888182186                  2
571586           888182188                  2
571591           888182188                  2

      category_code_description_assess  census_tract_assess  depth_assess  \
54433                  Vacant Land              30.0        58.0
565290                 Single Family             30.0        NaN
565291                 Single Family             30.0        NaN
565292                  Vacant Land              30.0        NaN
571356                Multi Family             156.0        NaN

```

571592	Multi Family	156.0	NaN
571355	Multi Family	156.0	NaN
571594	Multi Family	156.0	NaN
571588	Multi Family	156.0	NaN
571589	Multi Family	156.0	NaN
571358	Multi Family	156.0	NaN
571590	Multi Family	156.0	NaN
571359	Multi Family	156.0	NaN
571593	Multi Family	156.0	NaN
571586	Multi Family	156.0	NaN
571591	Multi Family	156.0	NaN

	location_assess	market_value_assess	year_built_assess	\
54433	1521 REED ST	0.0	0	
565290	1521 REED ST	0.0	2020.0	
565291	1521 REED ST	0.0	2020.0	
565292	1521 REED ST	0.0	NaN	
571356	520 W MONTGOMERY AVE	NaN	NaN	
571592	1747 N 6TH ST	NaN	NaN	
571355	520 W MONTGOMERY AVE	NaN	NaN	
571594	522 W MONTGOMERY AVE	NaN	NaN	
571588	516 W MONTGOMERY AVE	NaN	NaN	
571589	520 W MONTGOMERY AVE	NaN	NaN	
571358	516 W MONTGOMERY AVE	NaN	NaN	
571590	520 W MONTGOMERY AVE	NaN	NaN	
571359	516 W MONTGOMERY AVE	NaN	NaN	
571593	520 W MONTGOMERY AVE	NaN	NaN	
571586	516 W MONTGOMERY AVE	NaN	NaN	
571591	520 W MONTGOMERY AVE	NaN	NaN	

	zip_code_assess	lat_assess	...	\
54433	19146.0	-75.170327	...	
565290	19146.0	-75.170327	...	
565291	19146.0	-75.170327	...	
565292	19146.0	-75.170327	...	
571356	19122.0	-75.144497	...	
571592	19122.0	-75.144497	...	
571355	19122.0	-75.144590	...	
571594	19122.0	-75.144590	...	
571588	19122.0	-75.144309	...	
571589	19122.0	-75.144309	...	
571358	19122.0	-75.144309	...	
571590	19122.0	-75.144309	...	
571359	19122.0	-75.144497	...	
571593	19122.0	-75.144497	...	
571586	19122.0	-75.144309	...	
571591	19122.0	-75.144309	...	

		geometry_assess	vacant_assess	\
54433	POINT (-75.17032740502535 39.933403243705634)		1	
565290	POINT (-75.17032740502535 39.933403243705634)		0	
565291	POINT (-75.17032740502535 39.933403243705634)		0	
565292	POINT (-75.17032740502535 39.933403243705634)		1	
571356	POINT (-75.1444966549058 39.97851531814684)		0	
571592	POINT (-75.1444966549058 39.97851531814684)		0	
571355	POINT (-75.14459023951433 39.978527707313056)		0	
571594	POINT (-75.14459023951433 39.978527707313056)		0	
571588	POINT (-75.14430948579115 39.97849053958551)		0	
571589	POINT (-75.14430948579115 39.97849053958551)		0	
571358	POINT (-75.14430948579115 39.97849053958551)		0	
571590	POINT (-75.14430948579115 39.97849053958551)		0	
571359	POINT (-75.1444966549058 39.97851531814684)		0	
571593	POINT (-75.1444966549058 39.97851531814684)		0	
571586	POINT (-75.14430948579115 39.97849053958551)		0	
571591	POINT (-75.14430948579115 39.97849053958551)		0	
		market_value_2015_assess	market_value_2016_assess	\
54433		11400.0	11400.0	
565290		11400.0	11400.0	
565291		11400.0	11400.0	
565292		11400.0	11400.0	
571356		Nan	Nan	
571592		Nan	Nan	
571355		Nan	Nan	
571594		Nan	Nan	
571588		Nan	Nan	
571589		Nan	Nan	
571358		Nan	Nan	
571590		Nan	Nan	
571359		Nan	Nan	
571593		Nan	Nan	
571586		Nan	Nan	
571591		Nan	Nan	
		market_value_2017_assess	market_value_2018_assess	\
54433		47200.0	47200.0	
565290		47200.0	47200.0	
565291		47200.0	47200.0	
565292		47200.0	47200.0	
571356		Nan	Nan	
571592		Nan	Nan	
571355		Nan	Nan	
571594		Nan	Nan	
571588		Nan	Nan	

571589	NaN	NaN
571358	NaN	NaN
571590	NaN	NaN
571359	NaN	NaN
571593	NaN	NaN
571586	NaN	NaN
571591	NaN	NaN
	market_value_2019_assess	market_value_2020_assess \
54433	47200.0	47200.0
565290	47200.0	47200.0
565291	47200.0	47200.0
565292	47200.0	47200.0
571356	NaN	NaN
571592	NaN	NaN
571355	NaN	NaN
571594	NaN	NaN
571588	NaN	NaN
571589	NaN	NaN
571358	NaN	NaN
571590	NaN	NaN
571359	NaN	NaN
571593	NaN	NaN
571586	NaN	NaN
571591	NaN	NaN
	market_value_2021_assess	market_value_2022_assess
54433	145666.666667	145666.666667
565290	145666.666667	145666.666667
565291	145666.666667	145666.666667
565292	145666.666667	145666.666667
571356	NaN	NaN
571592	NaN	NaN
571355	NaN	NaN
571594	NaN	NaN
571588	NaN	NaN
571589	NaN	NaN
571358	NaN	NaN
571590	NaN	NaN
571359	NaN	NaN
571593	NaN	NaN
571586	NaN	NaN
571591	NaN	NaN

[16 rows x 21 columns]

```
[ ]: #removing parcel number that appeared more than once
assess_final = assess_full.loc[~assess_full['parcel_number_assess'].isin([365349205, 888182184, 888182180, 888182144, 888182142, 888182188, 888182186])].reset_index(drop=True)
assess_final.head()

[ ]:    parcel_number_assess  category_code_assess  \
0            57127281                  6
1            451106305                 6
2            57166705                  6
3            56143925                  6
4            56143950                  6

category_code_description_assess  census_tract_assess  depth_assess  \
0                   Vacant Land          142.0      439.00
1                   Vacant Land          379.0      256.00
2                   Vacant Land          142.0      43.53
3                   Vacant Land          367.0      68.83
4                   Vacant Land          367.0      68.83

location_assess  market_value_assess  year_built_assess  zip_code_assess  \
0  968 N LAWRENCE ST              0.0              0.0      19123.0
1  2550 CASTOR AVE              0.0              0.0      19134.0
2   513 POPLAR ST              0.0              0.0      19123.0
3  428 FAIRMOUNT AVE             0.0              0.0      19123.0
4  442 FAIRMOUNT AVE             0.0              0.0      19123.0

lat_assess  ...  geometry_assess  \
0 -75.144757  ...  POINT (-75.144756917366 39.96784709341962)
1 -75.092534  ...  POINT (-75.09253430091161 39.99145897715201)
2 -75.146154  ...  POINT (-75.14615444716026 39.96706685767916)
3 -75.145586  ...  POINT (-75.14558649490823 39.96267918822164)
4 -75.146035  ...  POINT (-75.1460354483123 39.96277166528522)

vacant_assess  market_value_2015_assess  market_value_2016_assess  \
0            1                  0.0                  0.0
1            1                  0.0                  0.0
2            1                  0.0                  0.0
3            1                  0.0                  0.0
4            1                  0.0                  0.0

market_value_2017_assess  market_value_2018_assess  \
0                  0.0                  0.0
1                  0.0                  0.0
2                  0.0                  0.0
3                  0.0                  0.0
4                  0.0                  0.0
```

```

market_value_2019_assess  market_value_2020_assess  \
0                      0.0                      0.0
1                      0.0                      0.0
2                      0.0                      0.0
3                      0.0                      0.0
4                      0.0                      0.0

market_value_2021_assess  market_value_2022_assess
0                      0.0                      0.0
1                      0.0                      0.0
2                      0.0                      0.0
3                      0.0                      0.0
4                      0.0                      0.0

```

[5 rows x 21 columns]

[]: `assess_final.shape`

[]: (581336, 21)

[]: `assess_final['parcel_number_assess'].nunique() #the number of unique opa number ↴ is now the same as the shape`

[]: 581336

[]: `#combining assessment and tax delinquency data`
`tax_assess_comb = assess_final.merge(tax_final, left_on = 'parcel_number_assess', right_on = 'opa_number_tax', how= 'left')`
`tax_assess_comb.head()`

	parcel_number_assess	category_code_assess	category_code_description_assess	census_tract_assess	depth_assess	location_assess	market_value_assess	year_built_assess	zip_code_assess
0	57127281	6	Vacant Land	142.0	439.00	968 N LAWRENCE ST	0.0	0.0	19123.0
1	451106305	6	Vacant Land	379.0	256.00				
2	57166705	6	Vacant Land	142.0	43.53				
3	56143925	6	Vacant Land	367.0	68.83				
4	56143950	6	Vacant Land	367.0	68.83				

```

1    2550 CASTOR AVE           0.0          0.0      19134.0
2    513 POPLAR ST            0.0          0.0      19123.0
3  428 FAIRMOUNT AVE          0.0          0.0      19123.0
4  442 FAIRMOUNT AVE          0.0          0.0      19123.0

lat_assess ... bankruptcy_tax sheriff_sale_tax \
0 -75.144757 ...           NaN          NaN
1 -75.092534 ...           NaN          NaN
2 -75.146154 ...           NaN          NaN
3 -75.145586 ...           NaN          NaN
4 -75.146035 ...           NaN          NaN

sequestration_enforcement_tax payment_agreement_tax principal_due_tax \
0                   NaN          NaN          NaN
1                   NaN          NaN          NaN
2                   NaN          NaN          NaN
3                   NaN          NaN          NaN
4                   NaN          NaN          NaN

total_due_tax total_assessment_tax lat_tax lng_tax geometry_tax
0       NaN          NaN          NaN          NaN          NaN
1       NaN          NaN          NaN          NaN          NaN
2       NaN          NaN          NaN          NaN          NaN
3       NaN          NaN          NaN          NaN          NaN
4       NaN          NaN          NaN          NaN          NaN

```

[5 rows x 41 columns]

[]: tax_assess_comb.columns

```

[ ]: Index(['parcel_number_assess', 'category_code_assess',
       'category_code_description_assess', 'census_tract_assess',
       'depth_assess', 'location_assess', 'market_value_assess',
       'year_built_assess', 'zip_code_assess', 'lat_assess', 'lng_assess',
       'geometry_assess', 'vacant_assess', 'market_value_2015_assess',
       'market_value_2016_assess', 'market_value_2017_assess',
       'market_value_2018_assess', 'market_value_2019_assess',
       'market_value_2020_assess', 'market_value_2021_assess',
       'market_value_2022_assess', 'opa_number_tax', 'street_address_tax',
       'zip_code_tax', 'num_years_owed_tax', 'most_recent_year_owed_tax',
       'oldest_year_owed_tax', 'year_of_last_assessment_tax',
       'general_building_description_tax', 'building_category_tax',
       'is_actionable_tax', 'bankruptcy_tax', 'sheriff_sale_tax',
       'sequestration_enforcement_tax', 'payment_agreement_tax',
       'principal_due_tax', 'total_due_tax', 'total_assessment_tax', 'lat_tax',
       'lng_tax', 'geometry_tax'],
      dtype='object')

```

```
[ ]: tax_assess_comb.head().T
```

```
[ ]:          0 \
parcel_number_assess      57127281
category_code_assess       6
category_code_description_assess Vacant Land
census_tract_assess      142.0
depth_assess              439.0
location_assess           968 N LAWRENCE ST
market_value_assess        0.0
year_built_assess          0.0
zip_code_assess            19123.0
lat_assess                 -75.144757
lng_assess                  39.967847
geometry_assess            POINT (-75.144756917366 39.96784709341962)
vacant_assess                1
market_value_2015_assess      0.0
market_value_2016_assess      0.0
market_value_2017_assess      0.0
market_value_2018_assess      0.0
market_value_2019_assess      0.0
market_value_2020_assess      0.0
market_value_2021_assess      0.0
market_value_2022_assess      0.0
opa_number_tax                NaN
street_address_tax             NaN
zip_code_tax                  NaN
num_years_owed_tax             NaN
most_recent_year_owed_tax      NaN
oldest_year_owed_tax            NaN
year_of_last_assessment_tax     NaN
general_building_description_tax NaN
building_category_tax            NaN
is_actionable_tax                NaN
bankruptcy_tax                  NaN
sheriff_sale_tax                NaN
sequestration_enforcement_tax     NaN
payment_agreement_tax             NaN
principal_due_tax                NaN
total_due_tax                  NaN
total_assessment_tax             NaN
lat_tax                         NaN
lng_tax                         NaN
geometry_tax                     NaN
```

1

\

parcel_number_assess	451106305
category_code_assess	6
category_code_description_assess	Vacant Land
census_tract_assess	379.0
depth_assess	256.0
location_assess	2550 CASTOR AVE
market_value_assess	0.0
year_built_assess	0.0
zip_code_assess	19134.0
lat_assess	-75.092534
lng_assess	39.991459
geometry_assess	POINT (-75.09253430091161 39.99145897715201)
vacant_assess	1
market_value_2015_assess	0.0
market_value_2016_assess	0.0
market_value_2017_assess	0.0
market_value_2018_assess	0.0
market_value_2019_assess	0.0
market_value_2020_assess	0.0
market_value_2021_assess	0.0
market_value_2022_assess	0.0
opa_number_tax	NaN
street_address_tax	NaN
zip_code_tax	NaN
num_years_owed_tax	NaN
most_recent_year_owed_tax	NaN
oldest_year_owed_tax	NaN
year_of_last_assessment_tax	NaN
general_building_description_tax	NaN
building_category_tax	NaN
is_actionable_tax	NaN
bankruptcy_tax	NaN
sheriff_sale_tax	NaN
sequestration_enforcement_tax	NaN
payment_agreement_tax	NaN
principal_due_tax	NaN
total_due_tax	NaN
total_assessment_tax	NaN
lat_tax	NaN
lng_tax	NaN
geometry_tax	NaN
	2
\	
parcel_number_assess	57166705
category_code_assess	6
category_code_description_assess	Vacant Land

census_tract_assess	142.0
depth_assess	43.53
location_assess	513 POPLAR ST
market_value_assess	0.0
year_built_assess	0.0
zip_code_assess	19123.0
lat_assess	-75.146154
lng_assess	39.967067
geometry_assess	POINT (-75.14615444716026 39.96706685767916)
vacant_assess	1
market_value_2015_assess	0.0
market_value_2016_assess	0.0
market_value_2017_assess	0.0
market_value_2018_assess	0.0
market_value_2019_assess	0.0
market_value_2020_assess	0.0
market_value_2021_assess	0.0
market_value_2022_assess	0.0
opa_number_tax	NaN
street_address_tax	NaN
zip_code_tax	NaN
num_years_owed_tax	NaN
most_recent_year_owed_tax	NaN
oldest_year_owed_tax	NaN
year_of_last_assessment_tax	NaN
general_building_description_tax	NaN
building_category_tax	NaN
is_actionable_tax	NaN
bankruptcy_tax	NaN
sheriff_sale_tax	NaN
sequestration_enforcement_tax	NaN
payment_agreement_tax	NaN
principal_due_tax	NaN
total_due_tax	NaN
total_assessment_tax	NaN
lat_tax	NaN
lng_tax	NaN
geometry_tax	NaN
	3
\	
parcel_number_assess	56143925
category_code_assess	6
category_code_description_assess	Vacant Land
census_tract_assess	367.0
depth_assess	68.83
location_assess	428 FAIRMOUNT AVE

market_value_assess	0.0
year_built_assess	0.0
zip_code_assess	19123.0
lat_assess	-75.145586
lng_assess	39.962679
geometry_assess	POINT (-75.14558649490823 39.96267918822164)
vacant_assess	1
market_value_2015_assess	0.0
market_value_2016_assess	0.0
market_value_2017_assess	0.0
market_value_2018_assess	0.0
market_value_2019_assess	0.0
market_value_2020_assess	0.0
market_value_2021_assess	0.0
market_value_2022_assess	0.0
opa_number_tax	NaN
street_address_tax	NaN
zip_code_tax	NaN
num_years_owed_tax	NaN
most_recent_year_owed_tax	NaN
oldest_year_owed_tax	NaN
year_of_last_assessment_tax	NaN
general_building_description_tax	NaN
building_category_tax	NaN
is_actionable_tax	NaN
bankruptcy_tax	NaN
sheriff_sale_tax	NaN
sequestration_enforcement_tax	NaN
payment_agreement_tax	NaN
principal_due_tax	NaN
total_due_tax	NaN
total_assessment_tax	NaN
lat_tax	NaN
lng_tax	NaN
geometry_tax	NaN
	4
parcel_number_assess	56143950
category_code_assess	6
category_code_description_assess	Vacant Land
census_tract_assess	367.0
depth_assess	68.83
location_assess	442 FAIRMOUNT AVE
market_value_assess	0.0
year_built_assess	0.0
zip_code_assess	19123.0
lat_assess	-75.146035

lng_assess	39.962772
geometry_assess	POINT (-75.1460354483123 39.96277166528522)
vacant_assess	1
market_value_2015_assess	0.0
market_value_2016_assess	0.0
market_value_2017_assess	0.0
market_value_2018_assess	0.0
market_value_2019_assess	0.0
market_value_2020_assess	0.0
market_value_2021_assess	0.0
market_value_2022_assess	0.0
opa_number_tax	NaN
street_address_tax	NaN
zip_code_tax	NaN
num_years_owed_tax	NaN
most_recent_year_owed_tax	NaN
oldest_year_owed_tax	NaN
year_of_last_assessment_tax	NaN
general_building_description_tax	NaN
building_category_tax	NaN
is_actionable_tax	NaN
bankruptcy_tax	NaN
sheriff_sale_tax	NaN
sequestration_enforcement_tax	NaN
payment_agreement_tax	NaN
principal_due_tax	NaN
total_due_tax	NaN
total_assessment_tax	NaN
lat_tax	NaN
lng_tax	NaN
geometry_tax	NaN

[]: tax_assess_comb.shape

[]: (581336, 41)

[]: tax_assess_comb['parcel_number_assess'].nunique()

[]: 581336

[]: tax_assess_comb['opa_number_tax'].nunique()

[]: 70637

[]: tax_assess_comb.isna().sum() # the assess file is bigger than the tax. We should
→ replace the null values on the tax column with sth else

```
[ ]: parcel_number_assess          0
category_code_assess              0
category_code_description_assess  77
census_tract_assess              3
depth_assess                     594
location_assess                  0
market_value_assess               231
year_built_assess                2511
zip_code_assess                  1
lat_assess                       0
lng_assess                       0
geometry_assess                  0
vacant_assess                    0
market_value_2015_assess         3605
market_value_2016_assess         3604
market_value_2017_assess         3604
market_value_2018_assess         3597
market_value_2019_assess         2339
market_value_2020_assess         937
market_value_2021_assess         387
market_value_2022_assess         269
opa_number_tax                   510699
street_address_tax               510699
zip_code_tax                      510699
num_years_owed_tax               510699
most_recent_year_owed_tax        510699
oldest_year_owed_tax             510699
year_of_last_assessment_tax     510785
general_building_description_tax 510786
building_category_tax            510786
is_actionable_tax                510699
bankruptcy_tax                   510699
sheriff_sale_tax                 510699
sequestration_enforcement_tax   510699
payment_agreement_tax            510699
principal_due_tax                510699
total_due_tax                     510699
total_assessment_tax             510785
lat_tax                           510699
lng_tax                           510699
geometry_tax                      510699
dtype: int64
```

```
[ ]: tax_assess_comb.loc[tax_assess_comb['opa_number_tax']>0].head().T #checking if ↳ all the joining was done correctly
#category_code_assess, category_code_description_assess, ↳
#general_building_description_tax also have vacant lot included
```

```
#created vacant_assess column, which can be used for modeling
```

[]:	6	\
parcel_number_assess	183009065	
category_code_assess	6	
category_code_description_assess	Vacant Land	
census_tract_assess	157.0	
depth_assess	0.0	
location_assess	105 GUEST ST	
market_value_assess	0.0	
year_built_assess	0.0	
zip_code_assess	19122.0	
lat_assess	-75.135164	
lng_assess	39.974838	
geometry_assess	POINT (-75.135164409753 39.974838097182754)	
vacant_assess	1	
market_value_2015_assess	0.0	
market_value_2016_assess	0.0	
market_value_2017_assess	0.0	
market_value_2018_assess	0.0	
market_value_2019_assess	0.0	
market_value_2020_assess	0.0	
market_value_2021_assess	0.0	
market_value_2022_assess	0.0	
opa_number_tax	183009065.0	
street_address_tax	105 GUEST ST	
zip_code_tax	19122.0	
num_years_owed_tax	1.0	
most_recent_year_owed_tax	2016.0	
oldest_year_owed_tax	2016.0	
year_of_last_assessment_tax	2021.0	
general_building_description_tax	parkingLot	
building_category_tax	commercial	
is_actionable_tax	True	
bankruptcy_tax	False	
sheriff_sale_tax	N	
sequestration_enforcement_tax	False	
payment_agreement_tax	False	
principal_due_tax	267.23	
total_due_tax	542.18	
total_assessment_tax	0.0	
lat_tax	-75.135164	
lng_tax	39.974838	
geometry_tax	POINT (-75.135164409753 39.974838097182754)	

9

\

parcel_number_assess	57168041
category_code_assess	6
category_code_description_assess	Vacant Land
census_tract_assess	142.0
depth_assess	0.0
location_assess	74-82 LAUREL ST
market_value_assess	0.0
year_built_assess	0.0
zip_code_assess	19123.0
lat_assess	-75.136798
lng_assess	39.96477
geometry_assess	POINT (-75.13679847443524 39.96477018508294)
vacant_assess	1
market_value_2015_assess	0.0
market_value_2016_assess	0.0
market_value_2017_assess	0.0
market_value_2018_assess	0.0
market_value_2019_assess	0.0
market_value_2020_assess	0.0
market_value_2021_assess	0.0
market_value_2022_assess	0.0
opa_number_tax	57168041.0
street_address_tax	74-82 LAUREL ST APT 10
zip_code_tax	19123.0
num_years_owed_tax	1.0
most_recent_year_owed_tax	2017.0
oldest_year_owed_tax	2017.0
year_of_last_assessment_tax	2021.0
general_building_description_tax	vacantLand
building_category_tax	residential
is_actionable_tax	True
bankruptcy_tax	False
sheriff_sale_tax	Y
sequestration_enforcement_tax	False
payment_agreement_tax	False
principal_due_tax	7334.11
total_due_tax	11393.98
total_assessment_tax	0.0
lat_tax	-75.136798
lng_tax	39.96477
geometry_tax	POINT (-75.13679847443524 39.96477018508294)

\

parcel_number_assess	23082116
category_code_assess	6
category_code_description_assess	Vacant Land

census_tract_assess	16.0
depth_assess	180.13
location_assess	525R FITZWATER ST
market_value_assess	0.0
year_built_assess	0.0
zip_code_assess	19147.0
lat_assess	-75.152358
lng_assess	39.9403
geometry_assess	POINT (-75.15235769133129 39.94029984813049)
vacant_assess	1
market_value_2015_assess	0.0
market_value_2016_assess	0.0
market_value_2017_assess	0.0
market_value_2018_assess	0.0
market_value_2019_assess	0.0
market_value_2020_assess	0.0
market_value_2021_assess	0.0
market_value_2022_assess	0.0
opa_number_tax	23082116.0
street_address_tax	525R FITZWATER ST # CA
zip_code_tax	19147.0
num_years_owed_tax	1.0
most_recent_year_owed_tax	2013.0
oldest_year_owed_tax	2013.0
year_of_last_assessment_tax	2021.0
general_building_description_tax	vacantLand
building_category_tax	residential
is_actionable_tax	True
bankruptcy_tax	False
sheriff_sale_tax	Y
sequestration_enforcement_tax	False
payment_agreement_tax	False
principal_due_tax	0.0
total_due_tax	74.0
total_assessment_tax	0.0
lat_tax	-75.152358
lng_tax	39.9403
geometry_tax	POINT (-75.15235769133129 39.94029984813049)

parcel_number_assess	79 \
category_code_assess	884790010
category_code_description_assess	4
census_tract_assess	Commercial
depth_assess	809.0
location_assess	257.7
market_value_assess	3100 S 61ST ST
	120600.0

year_built_assess	1950.0
zip_code_assess	19153.0
lat_assess	-75.211156
lng_assess	39.920482
geometry_assess	POINT (-75.21115609225518 39.9204816469858)
vacant_assess	0
market_value_2015_assess	39300.0
market_value_2016_assess	39300.0
market_value_2017_assess	39300.0
market_value_2018_assess	112100.0
market_value_2019_assess	119900.0
market_value_2020_assess	120600.0
market_value_2021_assess	120600.0
market_value_2022_assess	120600.0
opa_number_tax	884790010.0
street_address_tax	3100 S 61ST ST
zip_code_tax	19153.0
num_years_owed_tax	25.0
most_recent_year_owed_tax	2021.0
oldest_year_owed_tax	1997.0
year_of_last_assessment_tax	2021.0
general_building_description_tax	miscCommercial
building_category_tax	commercial
is_actionable_tax	True
bankruptcy_tax	False
sheriff_sale_tax	N
sequestration_enforcement_tax	False
payment_agreement_tax	False
principal_due_tax	20310.46
total_due_tax	46658.12
total_assessment_tax	120600.0
lat_tax	-75.211156
lng_tax	39.920482
geometry_tax	POINT (-75.21115609225518 39.9204816469858)
	88
parcel_number_assess	885500860
category_code_assess	6
category_code_description_assess	Vacant Land
census_tract_assess	199.0
depth_assess	24.6
location_assess	1131R W WESTMORELAND ST
market_value_assess	3200.0
year_built_assess	0.0
zip_code_assess	19140.0
lat_assess	-75.148048
lng_assess	40.002911

geometry_assess	POINT (-75.14804801037884 40.00291103862602)
vacant_assess	1
market_value_2015_assess	1100.0
market_value_2016_assess	1100.0
market_value_2017_assess	1100.0
market_value_2018_assess	3200.0
market_value_2019_assess	3200.0
market_value_2020_assess	3200.0
market_value_2021_assess	3200.0
market_value_2022_assess	3200.0
opa_number_tax	885500860.0
street_address_tax	1131R W WESTMORELAND ST
zip_code_tax	19140.0
num_years_owed_tax	6.0
most_recent_year_owed_tax	2021.0
oldest_year_owed_tax	2016.0
year_of_last_assessment_tax	2021.0
general_building_description_tax	vacantLand
building_category_tax	commercial
is_actionable_tax	True
bankruptcy_tax	False
sheriff_sale_tax	N
sequestration_enforcement_tax	False
payment_agreement_tax	False
principal_due_tax	232.72
total_due_tax	337.67
total_assessment_tax	3200.0
lat_tax	-75.148048
lng_tax	40.002911
geometry_tax	POINT (-75.14804801037884 40.00291103862602)

```
[ ]: violation_final = pd.read_csv("violation_final.csv", index_col=0) #property
    ↳ code violation file
violation_final = violation_final.add_suffix('_vio') # adding vio in the end of
    ↳ columns to understand where they are coming from
violation_final['opa_account_num_vio'] = violation_final['opa_account_num_vio'].\
    ↳ astype(int)
violation_final.head()
```

```
[ ]:   opa_account_num_vio  casenumber_diff_vio  casecreateddate_year_diff_vio \
0          881126002           34                  6
1          884202000            8                  3
2          881074500           64                  6
3          884114060            9                  2
4          881041000           11                  6
```

violationcode_diff_vio num_vacant_code_vio

```
0          94          0
1          73          1
2          65          0
3          61          0
4          60          0
```

```
[ ]: violation_final.dtypes
```

```
[ ]: opa_account_num_vio           int64
casenumber_diff_vio            int64
casecreateddate_year_diff_vio  int64
violationcode_diff_vio        int64
num_vacant_code_vio          int64
dtype: object
```

```
[ ]: tax_assess_vio_comb = tax_assess_comb.merge(violation_final, left_on = 'parcel_number_assess', right_on = 'opa_account_num_vio', how= 'left')
tax_assess_vio_comb.head()
```

```
[ ]:   parcel_number_assess  category_code_assess  \
0          57127281              6
1          451106305             6
2          57166705              6
3          56143925              6
4          56143950              6

      category_code_description_assess  census_tract_assess  depth_assess  \
0                  Vacant Land          142.0       439.00
1                  Vacant Land          379.0       256.00
2                  Vacant Land          142.0       43.53
3                  Vacant Land          367.0       68.83
4                  Vacant Land          367.0       68.83

      location_assess  market_value_assess  year_built_assess  zip_code_assess  \
0  968 N LAWRENCE ST            0.0          0.0       19123.0
1  2550 CASTOR AVE            0.0          0.0       19134.0
2    513 POPLAR ST            0.0          0.0       19123.0
3  428 FAIRMOUNT AVE           0.0          0.0       19123.0
4  442 FAIRMOUNT AVE           0.0          0.0       19123.0

      lat_assess  ...  total_due_tax total_assessment_tax  lat_tax  lng_tax  \
0 -75.144757  ...        NaN            NaN        NaN        NaN
1 -75.092534  ...        NaN            NaN        NaN        NaN
2 -75.146154  ...        NaN            NaN        NaN        NaN
3 -75.145586  ...        NaN            NaN        NaN        NaN
4 -75.146035  ...        NaN            NaN        NaN        NaN
```

```

geometry_tax  opa_account_num_vio  casenumber_diff_vio  \
0           NaN                  NaN                  NaN
1           NaN                  NaN                  NaN
2           NaN                  NaN                  NaN
3           NaN                  NaN                  NaN
4           NaN                  NaN                  NaN

casetreateddate_year_diff_vio  violationcode_diff_vio  num_vacant_code_vio
0                           NaN                  NaN                  NaN
1                           NaN                  NaN                  NaN
2                           NaN                  NaN                  NaN
3                           NaN                  NaN                  NaN
4                           NaN                  NaN                  NaN

[5 rows x 46 columns]

```

[]: tax_assess_vio_comb.shape

[]: (581336, 46)

[]: #tax_assess_vio_comb.to_csv("tax_assess_vio_comb.csv", index = False)

0.2.20 Including No. of Crimes with 50m

[]: tax_assess_vio_comb = pd.read_csv("tax_assess_vio_comb.csv")

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3457: DtypeWarning: Columns (7) have
mixed types. Specify dtype option on import or set low_memory=False.
exec(code_obj, self.user_global_ns, self.user_ns)
```

[]: tax_assess_vio_comb.columns

```

[ ]: Index(['parcel_number_assess', 'category_code_assess',
       'category_code_description_assess', 'census_tract_assess',
       'depth_assess', 'location_assess', 'market_value_assess',
       'year_built_assess', 'zip_code_assess', 'lat_assess', 'lng_assess',
       'geometry_assess', 'vacant_assess', 'market_value_2015_assess',
       'market_value_2016_assess', 'market_value_2017_assess',
       'market_value_2018_assess', 'market_value_2019_assess',
       'market_value_2020_assess', 'market_value_2021_assess',
       'market_value_2022_assess', 'opa_number_tax', 'street_address_tax',
       'zip_code_tax', 'num_years_owed_tax', 'most_recent_year_owed_tax',
       'oldest_year_owed_tax', 'year_of_last_assessment_tax',
       'general_building_description_tax', 'building_category_tax',
       'is_actionable_tax', 'bankruptcy_tax', 'sheriff_sale_tax',
       'sequestration_enforcement_tax', 'payment_agreement_tax'],
      dtype='object')

```

```
'principal_due_tax', 'total_due_tax', 'total_assessment_tax', 'lat_tax',
'lng_tax', 'geometry_tax', 'opa_account_num_vio', 'casenumber_diff_vio',
'casecreateddate_year_diff_vio', 'violationcode_diff_vio',
'num_vacant_code_vio'],
dtype='object')
```

```
[ ]: #combining latitude and longitude data to make geometry column
crs = {'init': 'epsg:4326'}
geometry = [Point(xy) for xy in zip(tax_assess_vio_comb["lat_assess"], ↴
                                     tax_assess_vio_comb["lng_assess"])]
geometry[:3]

tax_assess_vio_comb = gpd.GeoDataFrame(tax_assess_vio_comb,
                                         crs = crs,
                                         geometry = geometry)

tax_assess_vio_comb.head()
```

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/pyproj/crs/crs.py:131: FutureWarning: '+init=<authority>:<code>' syntax
is deprecated. '<authority>:<code>' is the preferred initialization method. When
making the change, be mindful of axis order changes:
https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-
proj-6
```

```
in_crs_string = _prepare_from_proj_string(in_crs_string)
```

```
[ ]: parcel_number_assess  category_code_assess  \
0          57127281              6
1          451106305             6
2          57166705              6
3          56143925              6
4          56143950              6

category_code_description_assess  census_tract_assess  depth_assess  \
0           Vacant Land          142.0        439.00
1           Vacant Land          379.0        256.00
2           Vacant Land          142.0        43.53
3           Vacant Land          367.0        68.83
4           Vacant Land          367.0        68.83

location_assess  market_value_assess  year_built_assess  zip_code_assess  \
0  968 N LAWRENCE ST            0.0            0.0        19123.0
1  2550 CASTOR AVE            0.0            0.0        19134.0
2   513 POPLAR ST             0.0            0.0        19123.0
3  428 FAIRMOUNT AVE           0.0            0.0        19123.0
4  442 FAIRMOUNT AVE           0.0            0.0        19123.0
```

```

lat_assess ... total_assessment_tax lat_tax lng_tax geometry_tax \
0 -75.144757 ... NaN NaN NaN NaN
1 -75.092534 ... NaN NaN NaN NaN
2 -75.146154 ... NaN NaN NaN NaN
3 -75.145586 ... NaN NaN NaN NaN
4 -75.146035 ... NaN NaN NaN NaN

opa_account_num_vio casenumber_diff_vio casecreateddate_year_diff_vio \
0 NaN NaN NaN
1 NaN NaN NaN
2 NaN NaN NaN
3 NaN NaN NaN
4 NaN NaN NaN

violationcode_diff_vio num_vacant_code_vio geometry
0 NaN NaN POINT (-75.14476 39.96785)
1 NaN NaN POINT (-75.09253 39.99146)
2 NaN NaN POINT (-75.14615 39.96707)
3 NaN NaN POINT (-75.14559 39.96268)
4 NaN NaN POINT (-75.14604 39.96277)

[5 rows x 47 columns]

```

[]: tax_assess_vio_comb.crs

[]: <Geographic 2D CRS: +init=epsg:4326 +type=crs>
Name: WGS 84
Axis Info [ellipsoidal]:
- lon[east]: Longitude (degree)
- lat[north]: Latitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich

[]: crime.crs

[]: <Geographic 2D CRS: +init=epsg:4326 +type=crs>
Name: WGS 84
Axis Info [ellipsoidal]:
- lon[east]: Longitude (degree)
- lat[north]: Latitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)

```
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

```
[ ]: #converting crime and combined dataset into something that is easier to work with
      crime = crime.to_crs(epsg = 3311)
      tax_assess_vio_comb = tax_assess_vio_comb.to_crs(epsg = 3311)
```

```
[ ]: crime.crs
```

```
[ ]: <Derived Projected CRS: EPSG:3311>
Name: NAD83(HARN) / California Albers
Axis Info [cartesian]:
- X[east]: Easting (metre)
- Y[north]: Northing (metre)
Area of Use:
- name: United States (USA) - California.
- bounds: (-124.45, 32.53, -114.12, 42.01)
Coordinate Operation:
- name: California Albers
- method: Albers Equal Area
Datum: NAD83 (High Accuracy Reference Network)
- Ellipsoid: GRS 1980
- Prime Meridian: Greenwich
```

```
[ ]: tax_assess_vio_comb.crs
```

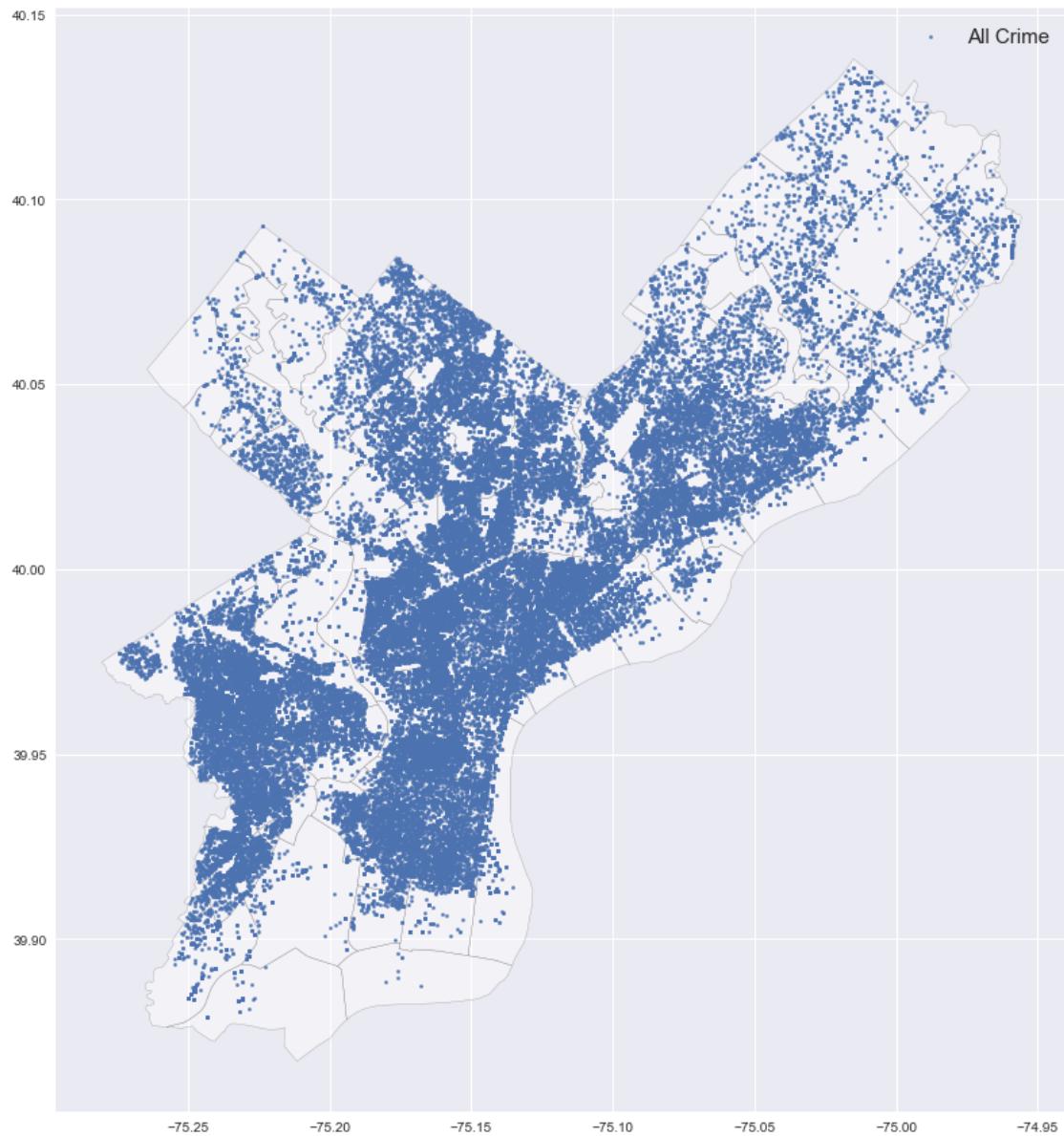
```
[ ]: <Derived Projected CRS: EPSG:3311>
Name: NAD83(HARN) / California Albers
Axis Info [cartesian]:
- X[east]: Easting (metre)
- Y[north]: Northing (metre)
Area of Use:
- name: United States (USA) - California.
- bounds: (-124.45, 32.53, -114.12, 42.01)
Coordinate Operation:
- name: California Albers
- method: Albers Equal Area
Datum: NAD83 (High Accuracy Reference Network)
- Ellipsoid: GRS 1980
- Prime Meridian: Greenwich
```

```
[ ]: # creating 50m radius around each property, 1 min walk. Typical house is at least within 6m radius
      tax_assess_vio_comb['buffer_50m'] = tax_assess_vio_comb.buffer(50)
```

```
[ ]: #crime in the last 6 months
crime_6m = crime.loc[(crime['dispatch_date'] >= '2021-07-01')
                      & (crime['dispatch_date'] <= '2021-12-31')]

[ ]: #crime in the last 3.5 years
crime_3y6m = crime.loc[(crime['dispatch_date'] >= '2018-07-01')
                           & (crime['dispatch_date'] <= '2021-12-31')]

[ ]: #plotting where crimes are occurring in the last 6 months
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white",
                                      edgecolor='black')
crime_6m.to_crs(epsg = 4326).plot(ax = ax, legend=True, markersize = 5, label =
                                     "All Crime")
#crime[crime_df['text_general_code'] == 'Thefts'].plot(ax = ax, markersize =
                                                       20, color = "red", marker = "^", label = "Thefts")
plt.legend(prop = {'size' : 15})
plt.show()
```



```
[ ]: crime_6m.shape
```

```
[ ]: (71804, 18)
```

```
[ ]: crime_3y6m.shape
```

```
[ ]: (509879, 18)
```

```
[ ]: crime.shape
```

```
[ ]: (1076153, 18)
```

```
[ ]: print(crime_6m['dispatch_date'].max())
print(crime_6m['dispatch_date'].min())
```

```
2021-12-31 00:00:00
2021-07-01 00:00:00
```

```
[ ]: print(crime_3y6m['dispatch_date'].max())
print(crime_3y6m['dispatch_date'].min())
```

```
2021-12-31 00:00:00
2018-07-01 00:00:00
```

```
[ ]: print(crime['dispatch_date'].max())
print(crime['dispatch_date'].min())
```

```
2021-12-31 00:00:00
2015-01-01 00:00:00
```

```
[ ]: crime['objectid'].nunique() # object if is unique number of it is fine to count
      ↪by ID
```

```
[ ]: 1076153
```

```
[ ]: # Spatial join, appending attributes from right table to left one
joined = gpd.sjoin(
    # Right table - crime
    crime_6m,
    # Left table - the combined dataset with the geometry reset from the
      ↪original
    # points to the 50m buffer and selecting only `parcel_number_assess` and
    # `buffer_50m` column
    tax_assess_vio_comb.set_geometry('buffer_50m')[['parcel_number_assess', ↪
      ↪'buffer_50m']],
    # Operation (spatial predicate) to use for the spatial join (`within`)
    op="within"
)
```

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3377: FutureWarning: The `op`
parameter is deprecated and will be removed in a future release. Please use the
`predicate` parameter instead.
    if (await self.run_code(code, result, async_=asy)):
```

```
[ ]: joined_3y = gpd.sjoin(
    crime_3y6m,
    tax_assess_vio_comb.set_geometry('buffer_50m')[['parcel_number_assess', ↪
      ↪'buffer_50m']],
```

```

# Operation (spatial predicate) to use for the spatial join (`within`)
op="within"
)

```

```

/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3377: FutureWarning: The `op` 
parameter is deprecated and will be removed in a future release. Please use the 
`predicate` parameter instead.

```

```
    if (await self.run_code(code, result, async_=asy)):
```

```

[ ]: joined_all = gpd.sjoin(
    crime,
    tax_assess_vio_comb.set_geometry('buffer_50m')[['parcel_number_assess', □
    ↳'buffer_50m']],
    # Operation (spatial predicate) to use for the spatial join (`within`)
    op="within"
)

```

```

/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3377: FutureWarning: The `op` 
parameter is deprecated and will be removed in a future release. Please use the 
`predicate` parameter instead.

```

```
    if (await self.run_code(code, result, async_=asy)):
```

```
[ ]: joined_3y.head()
```

```

[ ]:                                     the_geom  \
2      0101000020E61000002FD31F2F1ECE52C07129BE0C0CF5...
304993 0101000020E61000002FD31F2F1ECE52C07129BE0C0CF5...
304994 0101000020E61000002FD31F2F1ECE52C07129BE0C0CF5...
304995 0101000020E61000002FD31F2F1ECE52C07129BE0C0CF5...
908654 0101000020E61000004780892E1ECE52C05336930C0CF5...

                                     the_geom_webmercator  objectid  dc_dist  \
2      0101000020110F0000401FFA8143F15FC1160AD2D2C283...      119      12
304993 0101000020110F0000401FFA8143F15FC1160AD2D2C283...    922315      12
304994 0101000020110F0000401FFA8143F15FC1160AD2D2C283...    922316      12
304995 0101000020110F0000401FFA8143F15FC1160AD2D2C283...    922317      12
908654 0101000020110F000029C8FA8043F15FC1397BA2D2C283...  2648893      12

      psa  dispatch_date_time dispatch_date dispatch_time hour_ \
2      1  2018-07-23 01:58:00  2018-07-23 01:58:00   1.0
304993 1  2019-09-21 13:47:00  2019-09-21 13:47:00   13.0
304994 1  2019-04-03 10:20:00  2019-04-03 10:20:00   10.0
304995 1  2019-01-22 18:36:00  2019-01-22 18:36:00   18.0
908654 1  2020-10-27 23:40:57  2020-10-27 23:40:57   23.0

```

	dc_key	location_block	ucr_general	\		
2	201812053606	6600 BLOCK ESSINGTON AVE	300			
304993	201912082788	6600 BLOCK ESSINGTON AVE	1200			
304994	201912025413	6600 BLOCK ESSINGTON AVE	1200			
304995	201912005736	6600 BLOCK ESSINGTON AVE	1400			
908654	202012078480	6600 BLOCK ESSINGTON AV	500			
	text_general_code	point_x	point_y	lat	lng	\
2	Robbery No Firearm	-75.220592	39.91443	39.91443	-75.220592	
304993	Embezzlement	-75.220592	39.91443	39.91443	-75.220592	
304994	Embezzlement	-75.220592	39.91443	39.91443	-75.220592	
304995	Vandalism/Criminal Mischief	-75.220592	39.91443	39.91443	-75.220592	
908654	Burglary Non-Residential	-75.220592	39.91443	39.91443	-75.220592	
	geometry	index_right	parcel_number_assess			
2	POINT (3685875.975 1097948.254)	63377	884792501			
304993	POINT (3685875.975 1097948.254)	63377	884792501			
304994	POINT (3685875.975 1097948.254)	63377	884792501			
304995	POINT (3685875.975 1097948.254)	63377	884792501			
908654	POINT (3685875.986 1097948.257)	63377	884792501			

```
[ ]: #crime count in the last 6 months that happened within 50m in each parcel number
# Group by parcel number
crime_count = joined.groupby(
    "parcel_number_assess"
    # Keep only parcel number column (`parcel_number_assess`)
)[
    "objectid"
    # Count POIs by Airbnb + convert Series into DataFrame
].count().to_frame(
    'crime_count_50m_6m'
)
# Print top of the table
crime_count.head()
```

```
[ ]: crime_count_50m_6m
parcel_number_assess
11000600
11000700
11000800
11000900
11001000
```

parcel_number_assess	crime_count_50m_6m
11000600	1
11000700	1
11000800	1
11000900	3
11001000	3

```
[ ]: #crime count in the last 3.5 years that happened within 50m in each parcel
#number
crime_count_3y = joined_3y.groupby(
    "parcel_number_assess"
```

```
)[
    "objectid"
].count().to_frame(
    'crime_count_50m_3y6m'
)
crime_count_3y.head()
```

```
[ ]: crime_count_50m_3y6m
parcel_number_assess
11000600 9
11000700 9
11000800 9
11000900 11
11001000 11
```

```
[ ]: #crime count that happened within 50m in each parcel number
crime_count_all = joined_all.groupby(
    "parcel_number_assess"
)[
    "objectid"
].count().to_frame(
    'crime_count_50m_all'
)
crime_count_all.head()
```

```
[ ]: crime_count_50m_all
parcel_number_assess
11000600 23
11000700 23
11000800 23
11000900 25
11001000 26
```

```
[ ]: crime_count['crime_count'].max()
```

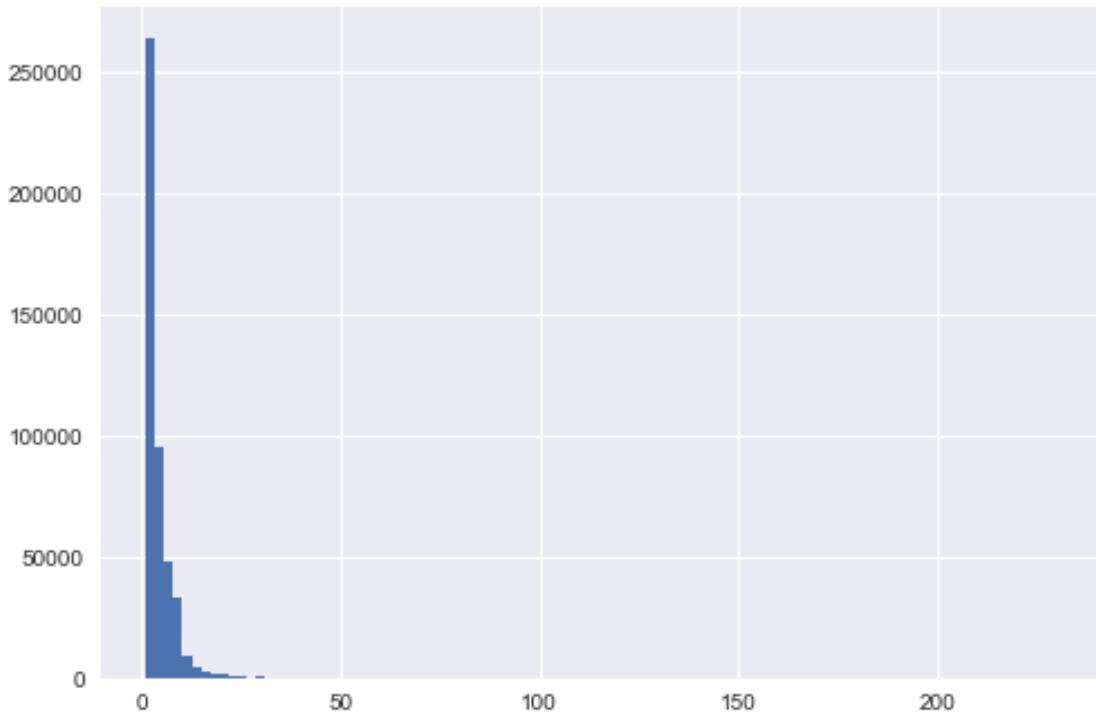
```
[ ]: 230
```

```
[ ]: crime_count['crime_count'].min()
```

```
[ ]: 1
```

```
[ ]: crime_count['crime_count'].hist(bins = 100)
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: #combining parcel number data with crime data from the last 6 month and
    ↪replacing the null values with 0
tax_assess_vio_comb_crimecount = tax_assess_vio_comb.merge(
    crime_count, left_on='parcel_number_assess', right_index=True
).fillna({"crime_count_50m_6m": 0})
```

```
[ ]: #combining parcel number data with crime data from the last 3.5 years and
    ↪replacing the null values with 0
tax_assess_vio_comb_crimecount = tax_assess_vio_comb_crimecount.merge(
    crime_count_3y, left_on='parcel_number_assess', right_index=True
).fillna({"crime_count_50m_3y6m": 0})
```

```
[ ]: #combining parcel number data with all crime data and replacing the null values
    ↪with 0
tax_assess_vio_comb_crimecount = tax_assess_vio_comb_crimecount.merge(
    crime_count_all, left_on='parcel_number_assess', right_index=True
).fillna({"crime_count_50m_all": 0})
```

```
[ ]: tax_assess_vio_comb_crimecount.sort_values(['crime_count_50m_6m'], ascending =
    ↪False).head()
```

```
[ ]:     parcel_number_assess  category_code_assess  \
207372          882070900          4
```

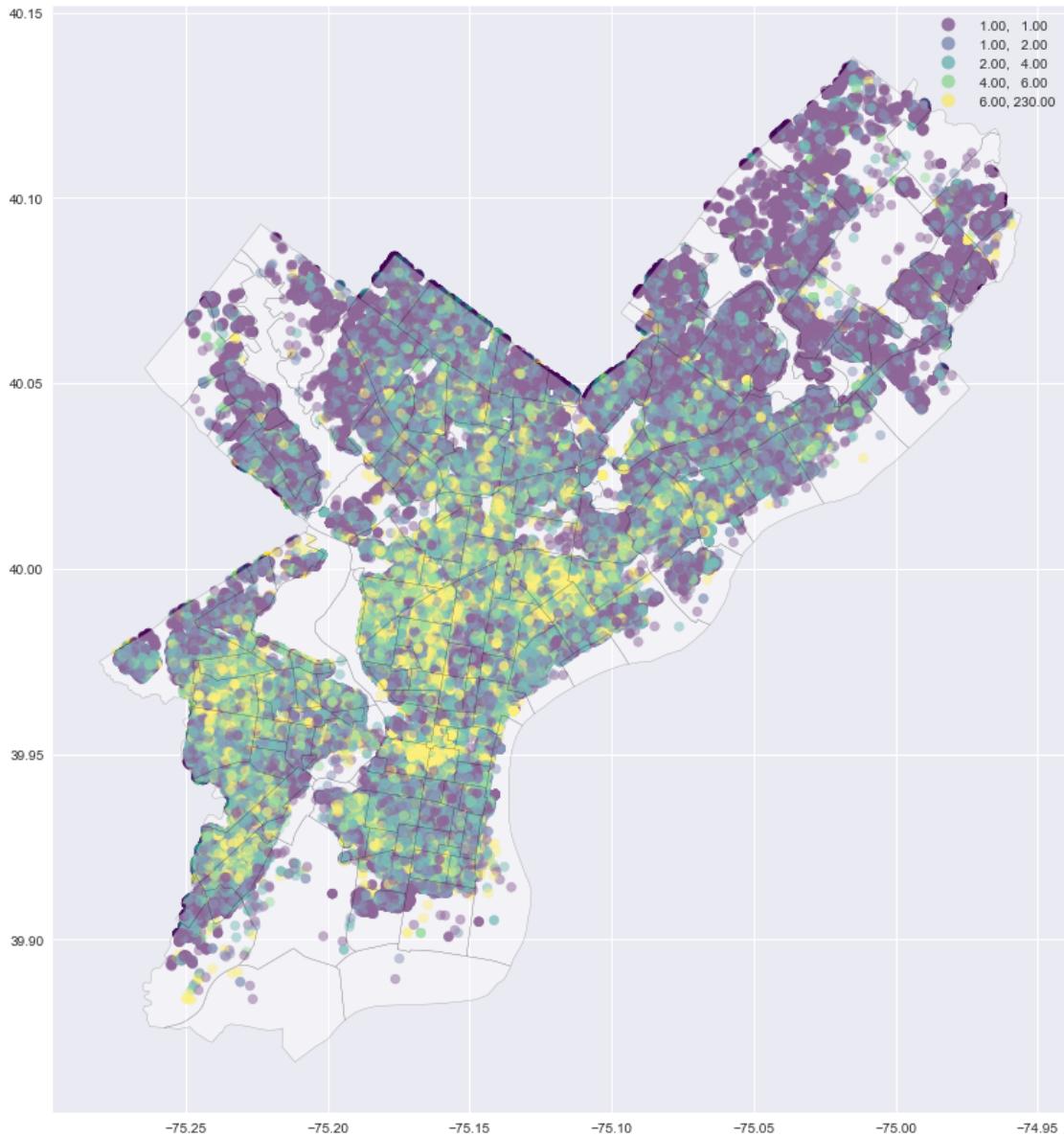
196611	786243200	4			
539859	561309400	2			
482521	561309500	2			
503614	561309600	2			
207372	category_code_description_assess	census_tract_assess	depth_assess	\	
	Commercial	379.0	516.58		
196611	Commercial	188.0	32.89		
539859	Multi Family	334.0	90.00		
482521	Multi Family	334.0	90.00		
503614	Multi Family	334.0	90.00		
207372	location_assess	market_value_assess	year_built_assess	\	
	2701 CASTOR AVE	15055400.0	2005		
196611	3201-05 KENSINGTON AVE	85000.0	1930		
539859	2301 BLEIGH AVE	190500.0	1925.0		
482521	2303 BLEIGH AVE	205000.0	1925.0		
503614	2305 BLEIGH AVE	201600.0	1925.0		
207372	zip_code_assess	lat_assess	...	casenumber_diff_vio	\
	19134.0	-75.089351	...	2.0	
196611	19134.0	-75.113042	...	NaN	
539859	19152.0	-75.057740	...	3.0	
482521	19152.0	-75.057661	...	2.0	
503614	19152.0	-75.057579	...	NaN	
207372	casecreateddate_year_diff_vio	violationcode_diff_vio		\	
	2.0	5.0			
196611	NaN	NaN			
539859	1.0	4.0			
482521	1.0	9.0			
503614	NaN	NaN			
207372	num_vacant_code_vio		geometry	\	
	0.0	POINT (3691990.609 1110604.880)			
196611	NaN	POINT (3689900.389 1110249.500)			
539859	0.0	POINT (3691391.938 1117683.189)			
482521	0.0	POINT (3691400.449 1117681.514)			
503614	NaN	POINT (3691409.041 1117680.103)			
207372	POLYGON ((3692040.609 1110604.880, 3692040.368...		buffer_50m	\	
196611	POLYGON ((3689950.389 1110249.500, 3689950.148...				
539859	POLYGON ((3691441.938 1117683.189, 3691441.698...				
482521	POLYGON ((3691450.449 1117681.514, 3691450.208...				
503614	POLYGON ((3691459.041 1117680.103, 3691458.800...				

	buffer_500m	crime_count_50m_6m	\
207372	POLYGON ((3692490.609 1110604.880, 3692488.201...		230
196611	POLYGON ((3690400.389 1110249.500, 3690397.981...		151
539859	POLYGON ((3691891.938 1117683.189, 3691889.531...		110
482521	POLYGON ((3691900.449 1117681.514, 3691898.041...		109
503614	POLYGON ((3691909.041 1117680.103, 3691906.633...		109
	crime_count_50m_3y6m	crime_count_50m_all	
207372	524	857	
196611	1507	2997	
539859	534	926	
482521	533	925	
503614	533	927	

[5 rows x 52 columns]

```
[ ]: #plotting where crimes are occurring with 50m of parcels
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
tax_assess_vio_comb_crimecount.to_crs(epsg = 4326).
    plot(column="crime_count_50m",
          scheme="quantiles",
          alpha=0.5,
          legend=True,
          ax=ax)
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white",
    edgecolor='black')
```

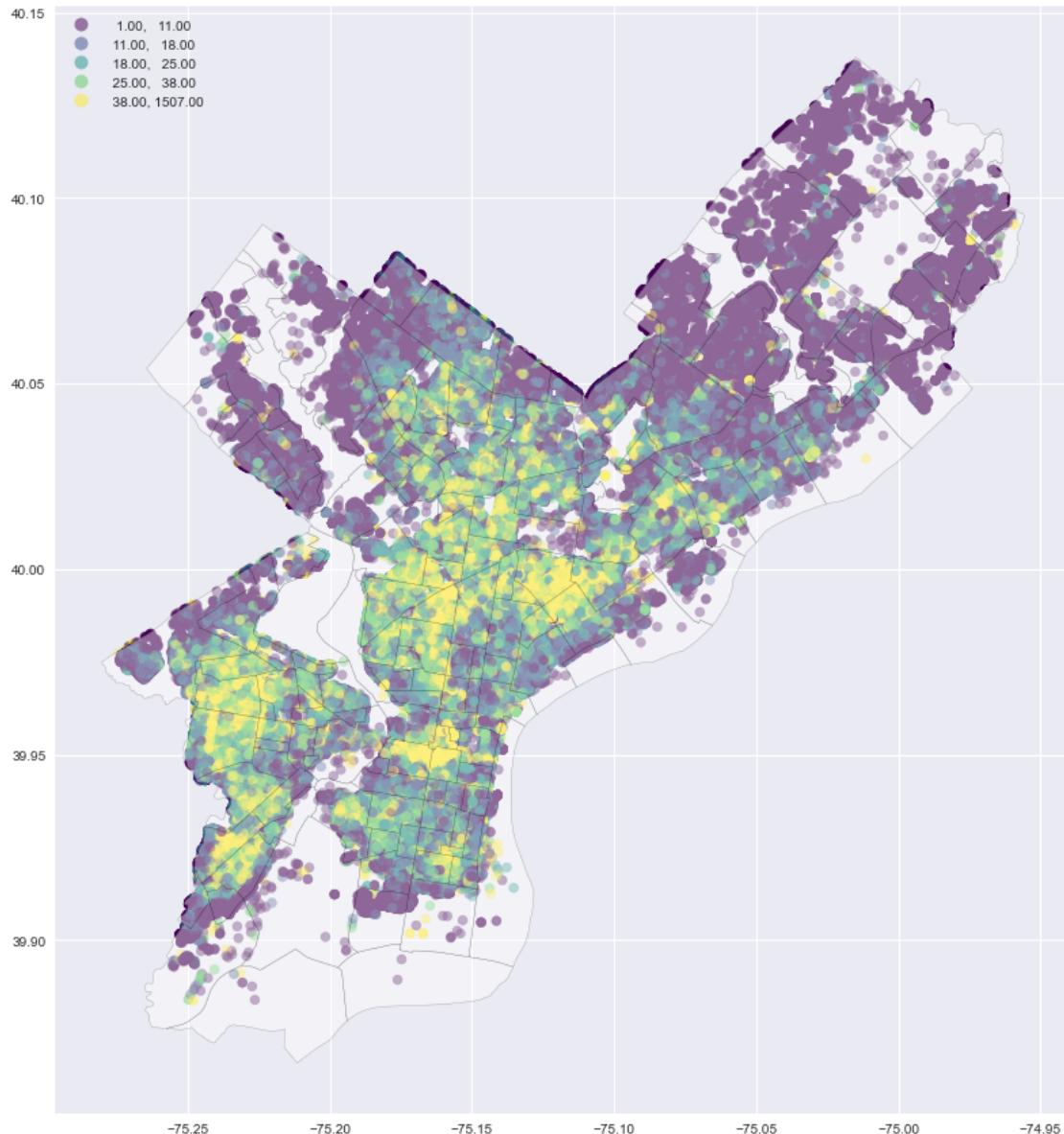
[]: <AxesSubplot:>



```
[ ]: #plotting where crimes are occurring with 50m of parcels in the last 3 year 6months
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
tax_assess_vio_comb_crimecount.to_crs(epsg = 4326).
    plot(column="crime_count_50m_3y6m",
          scheme="quantiles",
          alpha=0.5,
          legend=True,
          ax=ax)
```

```
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white",
                                         edgecolor='black')
```

[]: <AxesSubplot:>

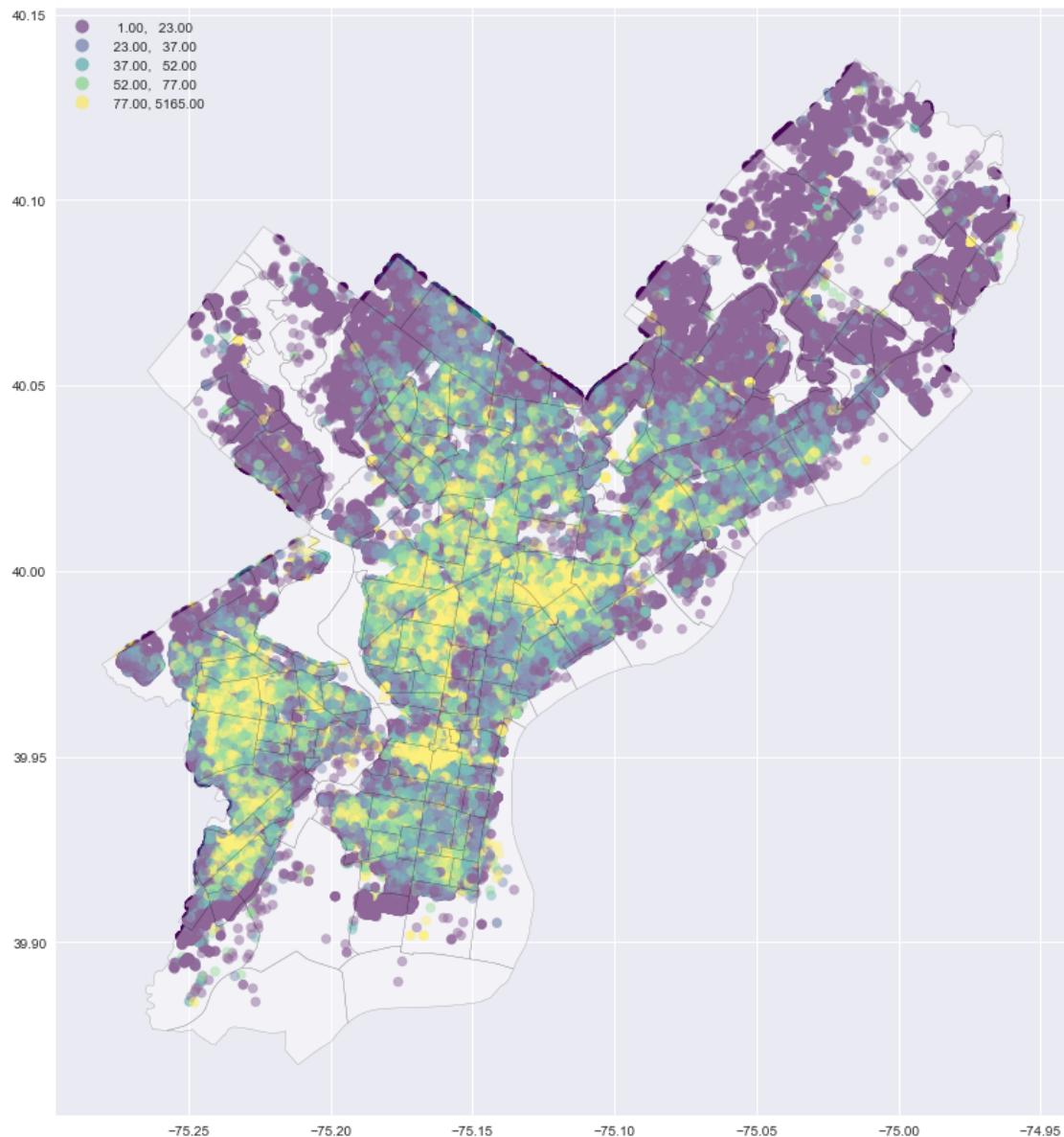


[]: *#plotting where crimes are occurring with 50m of parcels since 2015*

```
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
tax_assess_vio_comb_crimecount.to_crs(epsg = 4326).
    plot(column="crime_count_50m_all",
         scheme="quantiles",
```

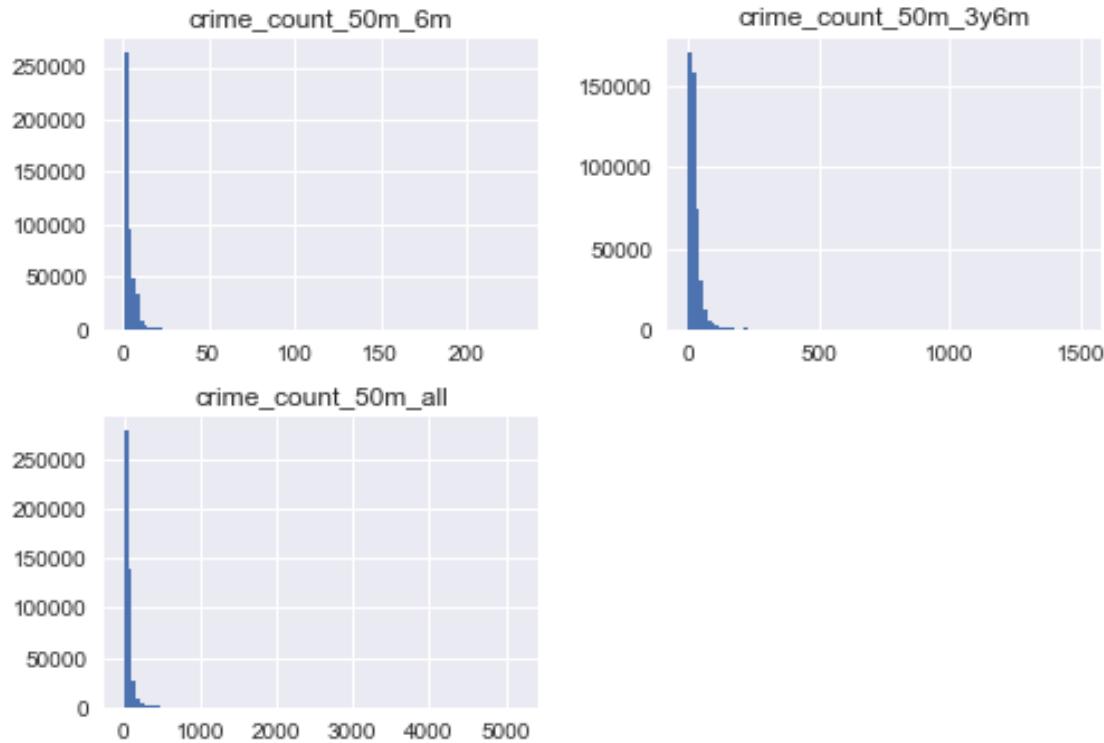
```
alpha=0.5,  
legend=True,  
ax=ax)  
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white",  
edgecolor='black')
```

[]: <AxesSubplot:>



```
[ ]: tax_assess_vio_comb_crimecount[['crime_count_50m_6m','crime_count_50m_3y6m','crime_count_50m_a  
↪hist(bins = 100)
```

```
[ ]: array([ [
```



```
[ ]: tax_assess_vio_comb_crimecount[['crime_count_50m_6m','crime_count_50m_3y6m','crime_count_50m_all']]  
→describe()
```

```
[ ]:      crime_count_50m_6m  crime_count_50m_3y6m  crime_count_50m_all
count      463855.000000      463855.000000      463855.000000
mean       4.138640        28.105182       58.142523
std        4.687586        33.080203       72.946461
min        1.000000        1.000000       1.000000
25%        2.000000        13.000000       27.000000
50%        3.000000        21.000000       44.000000
75%        5.000000        34.000000       70.000000
max       230.000000      1507.000000      5165.000000
```

```
[ ]: #tax_assess_vio_comb_crimecount.to_csv("all_crimecount.csv")
```

0.2.21 Including 311 calls counts within 50m

```
[ ]: #tax_assess_vio_comb_crimecount = pd.readcsv('tax_assess_vio_comb_crimecount.  
→csv')  
  
[ ]: tax_assess_vio_comb_crimecount.columns  
  
[ ]: Index(['parcel_number_assess', 'category_code_assess',  
           'category_code_description_assess', 'census_tract_assess',  
           'depth_assess', 'location_assess', 'market_value_assess',  
           'year_built_assess', 'zip_code_assess', 'lat_assess', 'lng_assess',  
           'geometry_assess', 'vacant_assess', 'market_value_2015_assess',  
           'market_value_2016_assess', 'market_value_2017_assess',  
           'market_value_2018_assess', 'market_value_2019_assess',  
           'market_value_2020_assess', 'market_value_2021_assess',  
           'market_value_2022_assess', 'opa_number_tax', 'street_address_tax',  
           'zip_code_tax', 'num_years_owed_tax', 'most_recent_year_owed_tax',  
           'oldest_year_owed_tax', 'year_of_last_assessment_tax',  
           'general_building_description_tax', 'building_category_tax',  
           'is_actionable_tax', 'bankruptcy_tax', 'sheriff_sale_tax',  
           'sequestration_enforcement_tax', 'payment_agreement_tax',  
           'principal_due_tax', 'total_due_tax', 'total_assessment_tax', 'lat_tax',  
           'lng_tax', 'geometry_tax', 'opa_account_num_vio', 'casenumber_diff_vio',  
           'caserecreateddate_year_diff_vio', 'violationcode_diff_vio',  
           'num_vacant_code_vio', 'geometry', 'buffer_50m', 'buffer_500m',  
           'crime_count_50m_6m', 'crime_count_50m_3y6m', 'crime_count_50m_all'],  
           dtype='object')  
  
[ ]: #combining latitude and longitude data to make geometry column  
crs = {'init': 'epsg:4326'}  
geometry = [Point(xy) for xy in  
→zip(tax_assess_vio_comb_crimecount["lat_assess"],  
→tax_assess_vio_comb_crimecount["lng_assess"])]  
geometry[:3]  
  
tax_assess_vio_comb_crimecount = gpd.  
→GeoDataFrame(tax_assess_vio_comb_crimecount,  
               crs = crs,  
               geometry = geometry)  
  
tax_assess_vio_comb_crimecount.head()
```

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-  
packages/pyproj/crs/crs.py:131: FutureWarning: '+init=<authority>:<code>' syntax  
is deprecated. '<authority>:<code>' is the preferred initialization method. When  
making the change, be mindful of axis order changes:  
https://pyproj4.github.io/pyproj/stable/gotchas.html#axis-order-changes-in-  
proj-6
```

```

in_crs_string = _prepare_from_proj_string(in_crs_string)

[ ]:   parcel_number_assess  category_code_assess  \
1          451106305           6
2          57166705            6
3          56143925            6
4          56143950            6
6          183009065           6

category_code_description_assess  census_tract_assess  depth_assess  \
1          Vacant Land        379.0       256.00
2          Vacant Land        142.0       43.53
3          Vacant Land        367.0       68.83
4          Vacant Land        367.0       68.83
6          Vacant Land        157.0       0.00

location_assess  market_value_assess year_built_assess  zip_code_assess  \
1  2550 CASTOR AVE         0.0          0.0      19134.0
2  513 POPLAR ST          0.0          0.0      19123.0
3  428 FAIRMOUNT AVE      0.0          0.0      19123.0
4  442 FAIRMOUNT AVE      0.0          0.0      19123.0
6  105 GUEST ST            0.0          0.0      19122.0

lat_assess  ...  casenumber_diff_vio  casecreateddate_year_diff_vio  \
1 -75.092534  ...           NaN           NaN
2 -75.146154  ...           NaN           NaN
3 -75.145586  ...           NaN           NaN
4 -75.146035  ...           NaN           NaN
6 -75.135164  ...           NaN           NaN

violationcode_diff_vio  num_vacant_code_vio           geometry  \
1           NaN           NaN  POINT (-75.09253 39.99146)
2           NaN           NaN  POINT (-75.14615 39.96707)
3           NaN           NaN  POINT (-75.14559 39.96268)
4           NaN           NaN  POINT (-75.14604 39.96277)
6           NaN           NaN  POINT (-75.13516 39.97484)

buffer_50m  \
1  POLYGON ((3691764.218 1110548.399, 3691763.977...
2  POLYGON ((3688925.025 1106049.599, 3688924.785...
3  POLYGON ((3689190.293 1105637.813, 3689190.053...
4  POLYGON ((3689151.484 1105629.488, 3689151.243...
6  POLYGON ((3689366.932 1107245.687, 3689366.691...

buffer_500m  crime_count_50m_6m  \
1  POLYGON ((3692214.218 1110548.399, 3692211.810...      1
2  POLYGON ((3689375.025 1106049.599, 3689372.618...      2

```

```
3 POLYGON ((3689640.293 1105637.813, 3689637.886... 1
4 POLYGON ((3689601.484 1105629.488, 3689599.077... 4
6 POLYGON ((3689816.932 1107245.687, 3689814.525... 6
```

```
crime_count_50m_3y6m crime_count_50m_all
1 5 8
2 25 50
3 20 40
4 21 43
6 19 32
```

[5 rows x 52 columns]

```
[ ]: tax_assess_vio_comb_crimecount.crs
```

```
[ ]: <Geographic 2D CRS: +init=epsg:4326 +type=crs>
Name: WGS 84
Axis Info [ellipsoidal]:
- lon[east]: Longitude (degree)
- lat[north]: Latitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

```
[ ]: philly_311_full.crs
```

```
[ ]: <Geographic 2D CRS: +init=epsg:4326 +type=crs>
Name: WGS 84
Axis Info [ellipsoidal]:
- lon[east]: Longitude (degree)
- lat[north]: Latitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

```
[ ]: #converting crime and combined dataset into something that is easier to work
    ↵with
philly_311_full = philly_311_full.to_crs(epsg = 3311)
tax_assess_vio_comb_crimecount = tax_assess_vio_comb_crimecount.to_crs(epsg = u
    ↵3311)
```

```
[ ]: philly_311_full.crs
```

```
[ ]: <Derived Projected CRS: EPSG:3311>
Name: NAD83(HARN) / California Albers
Axis Info [cartesian]:
- X[east]: Easting (metre)
- Y[north]: Northing (metre)
Area of Use:
- name: United States (USA) - California.
- bounds: (-124.45, 32.53, -114.12, 42.01)
Coordinate Operation:
- name: California Albers
- method: Albers Equal Area
Datum: NAD83 (High Accuracy Reference Network)
- Ellipsoid: GRS 1980
- Prime Meridian: Greenwich
```

```
[ ]: tax_assess_vio_comb_crimecount.crs
```

```
[ ]: <Derived Projected CRS: EPSG:3311>
Name: NAD83(HARN) / California Albers
Axis Info [cartesian]:
- X[east]: Easting (metre)
- Y[north]: Northing (metre)
Area of Use:
- name: United States (USA) - California.
- bounds: (-124.45, 32.53, -114.12, 42.01)
Coordinate Operation:
- name: California Albers
- method: Albers Equal Area
Datum: NAD83 (High Accuracy Reference Network)
- Ellipsoid: GRS 1980
- Prime Meridian: Greenwich
```

```
[ ]: philly_311_full.columns
```

```
[ ]: Index(['service_request_id', 'service_name', 'lat', 'lon', 'geometry',
       'requested_yeарmonth'],
       dtype='object')
```

```
[ ]: #as 311 data is not available for 2021, we cannot do 311 calls for the last 6 months
```

```
#311 calls in the last 3.5 years
philly_311_full_3y6m = philly_311_full.
    ↪loc[(philly_311_full['requested_yeарmonth'] >= '2018-07-01')
        & (philly_311_full['requested_yeарmonth'] <= '2021-12-31')]
```

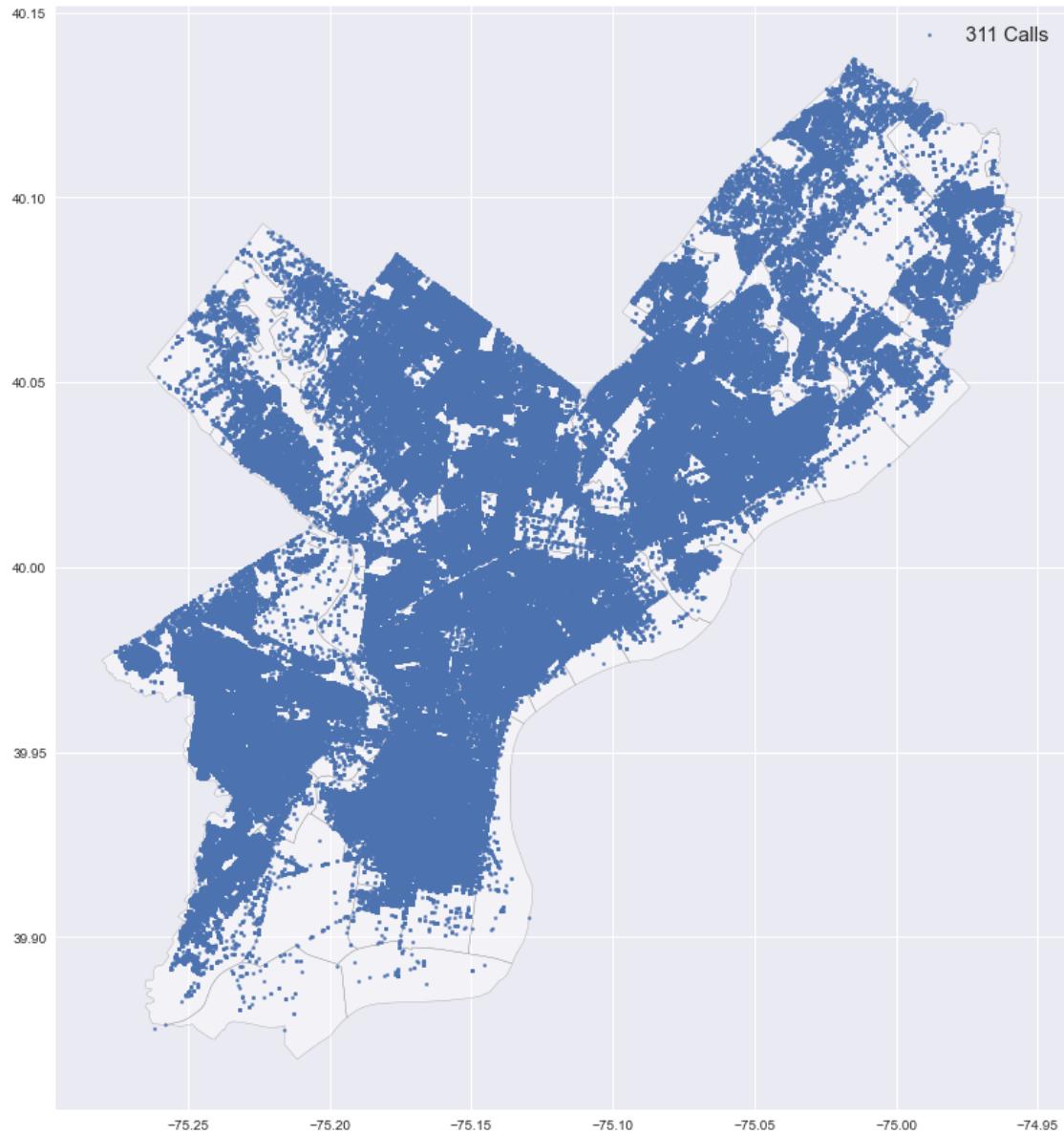
```
[ ]: philly_311_full_3y6m.head()
```

```
[ ]:      service_request_id    service_name      lat      lon  \
527835            12097785  Graffiti Removal  39.936281 -75.158160
527836            12071690  Street Defect   39.945154 -75.151640
527837            12071793  Street Defect   39.946252 -75.146114
527838            12071719  Street Defect   39.946106 -75.155503
527862            12072188  Illegal Dumping  40.050916 -75.087138

                           geometry requested_yearmonth
527835  POINT (3689520.395 1102537.955)          2018-07-01
527836  POINT (3689567.123 1103669.203)          2018-07-01
527837  POINT (3689931.704 1103992.875)          2018-07-01
527838  POINT (3689225.172 1103612.967)          2018-07-01
527862  POINT (3689111.047 1116635.993)          2018-07-01
```

```
[ ]: #plotting where 311 calls occurred in the last 3.5 year
```

```
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white", ↴
    edgecolor='black')
philly_311_full_3y6m.to_crs(epsg = 4326).plot(ax = ax,legend=True, markersize = ↴
    5, label = "311 Calls")
plt.legend(prop = {'size' : 15})
plt.show()
```



```
[ ]: philly_311_full_3y6m.shape
```

```
[ ]: (623301, 6)
```

```
[ ]: philly_311_full.shape
```

```
[ ]: (1273024, 6)
```

```
[ ]: #01 means that it is a full month
print(philly_311_full_3y6m['requested_yeарmonth'].max())
print(philly_311_full_3y6m['requested_yeарmonth'].min())
```

```
2020-12-01 00:00:00  
2018-07-01 00:00:00
```

```
[ ]: #01 means that it is a full month  
print(philly_311_full['requested_yearmonth'].max())  
print(philly_311_full['requested_yearmonth'].min())
```

```
2020-12-01 00:00:00  
2015-01-01 00:00:00
```

```
[ ]: joined_3y_311 = gpd.sjoin(  
    philly_311_full_3y6m,  
    tax_assess_vio_comb_crimecount.  
    set_geometry('buffer_50m')[['parcel_number_assess', 'buffer_50m']],  
    # Operation (spatial predicate) to use for the spatial join (`within`)  
    op="within"  
)
```

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-  
packages/IPython/core/interactiveshell.py:3377: FutureWarning: The `op`  
parameter is deprecated and will be removed in a future release. Please use the  
`predicate` parameter instead.
```

```
if (await self.run_code(code, result, async_=asy)):
```

```
[ ]: joined_all_311 = gpd.sjoin(  
    philly_311_full,  
    tax_assess_vio_comb_crimecount.  
    set_geometry('buffer_50m')[['parcel_number_assess', 'buffer_50m']],  
    # Operation (spatial predicate) to use for the spatial join (`within`)  
    op="within"  
)
```

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-  
packages/IPython/core/interactiveshell.py:3377: FutureWarning: The `op`  
parameter is deprecated and will be removed in a future release. Please use the  
`predicate` parameter instead.
```

```
if (await self.run_code(code, result, async_=asy)):
```

```
[ ]: joined_3y_311.head()
```

```
 [ ]:      service_request_id      service_name          lat          lon  \  
 527835            12097785  Graffiti Removal  39.936281 -75.158160  
 651415            12110126  Graffiti Removal  39.936797 -75.158303  
 651416            12114590  Graffiti Removal  39.936637 -75.158135  
 661796            12074456  Street Defect   39.936229 -75.158640  
 662530            12078323  Street Defect   39.936354 -75.158586  
                                         geometry requested_yearmonth index_right \
```

```
527835 POINT (3689520.395 1102537.955) 2018-07-01 5365
651415 POINT (3689483.338 1102583.415) 2018-07-01 5365
651416 POINT (3689504.234 1102574.144) 2018-07-01 5365
661796 POINT (3689486.486 1102514.101) 2018-07-01 5365
662530 POINT (3689484.270 1102528.617) 2018-07-01 5365
```

```
parcel_number_assess
527835 871007100
651415 871007100
651416 871007100
661796 871007100
662530 871007100
```

```
[ ]: joined_all_311.head()
```

```
service_request_id service_name lat \
0 8967043 Graffiti Removal 39.947292
48 8967042 Graffiti Removal 39.947292
12642 9084166 Rubbish/Recyclable Material Collection 39.947338
55226 9014937 Vacant House or Commercial 39.947183
63063 9530817 Graffiti Removal 39.947419
```

```
lon geometry requested_yearmonth \
0 -75.222567 POINT (3684064.881 1101122.461) 2015-01-01
48 -75.222567 POINT (3684064.881 1101122.461) 2015-01-01
12642 -75.222604 POINT (3684059.741 1101125.635) 2015-03-01
55226 -75.222437 POINT (3684080.301 1101116.834) 2015-02-01
63063 -75.222702 POINT (3684048.199 1101129.848) 2015-11-01
```

```
index_right parcel_number_assess
0 7872 871602360
48 7872 871602360
12642 7872 871602360
55226 7872 871602360
63063 7872 871602360
```

```
[ ]: #number of 311 calls within 50m of each parcel number in the last 3.5 year
call_count_3y = joined_3y_311.groupby(
    "parcel_number_assess"
)[
    "service_request_id"
].count().to_frame(
    'call_count_50m_3y6m'
)
call_count_3y.head()
```

```
[ ]: call_count_50m_3y6m  
parcel_number_assess  
11000600 43  
11000700 45  
11000800 49  
11000900 51  
11001000 56
```

```
[ ]: #number of 311 calls within 50m of each parcel number  
call_count_all = joined_all_311.groupby(  
    "parcel_number_assess")  
)[  
    "service_request_id"  
].count().to_frame()  
    'call_count_50m_all'  
)  
call_count_all.head()
```

```
[ ]: call_count_50m_all  
parcel_number_assess  
11000600 103  
11000700 110  
11000800 118  
11000900 118  
11001000 121
```

```
[ ]: call_count_3y['call_count_50m_3y6m'].describe()
```

```
[ ]: count    460901.000000  
mean      40.628506  
std       28.781408  
min       1.000000  
25%      21.000000  
50%      35.000000  
75%      54.000000  
max      515.000000  
Name: call_count_50m_3y6m, dtype: float64
```

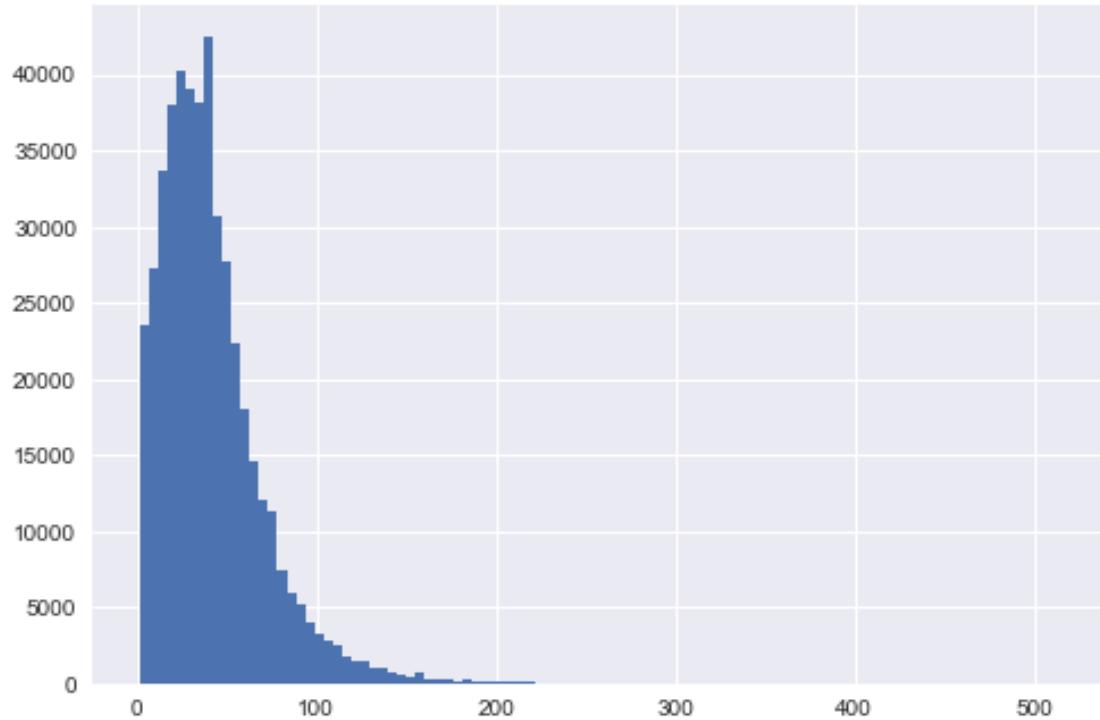
```
[ ]: call_count_all['call_count_50m_all'].describe()
```

```
[ ]: count    461907.000000  
mean      83.398730  
std       56.244291  
min       1.000000  
25%      43.000000  
50%      74.000000  
75%      111.000000
```

```
max      1198.000000  
Name: call_count_50m_all, dtype: float64
```

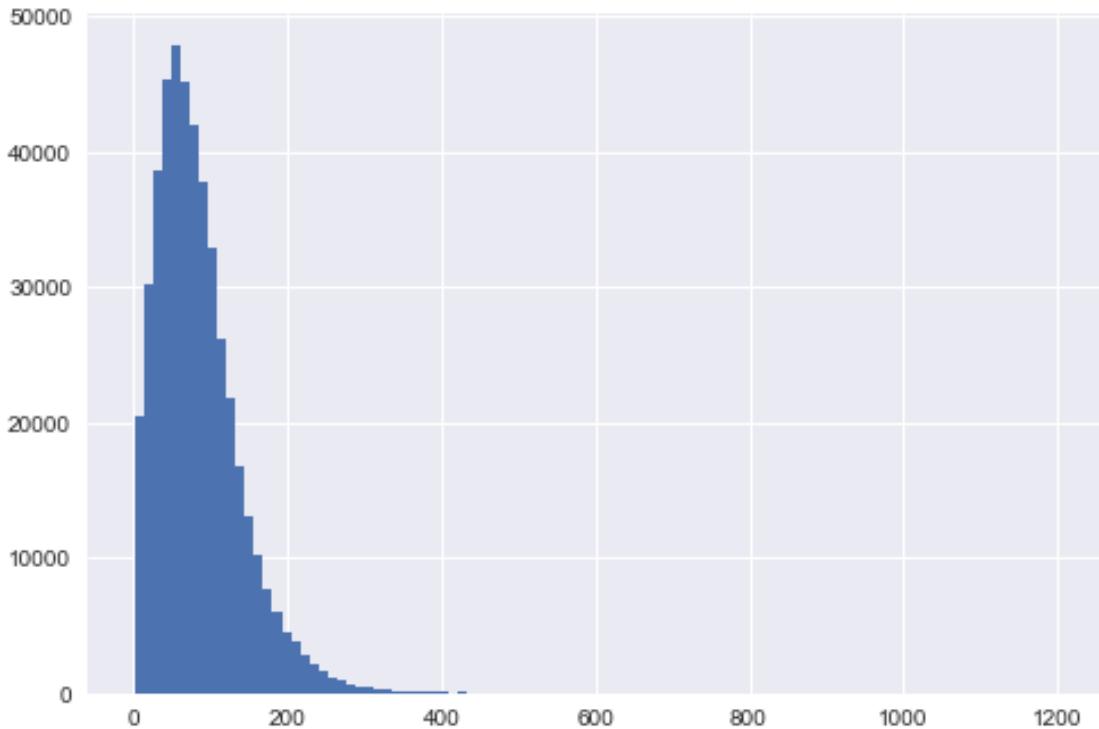
```
[ ]: call_count_3y['call_count_50m_3y6m'].hist(bins = 100)
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: call_count_all['call_count_50m_all'].hist(bins = 100)
```

```
[ ]: <AxesSubplot:>
```



```
[ ]: #combining 311 calls in the last 3.5 years with parcel number dataset.
    ↪replacing null values with zero
tax_assess_vio_comb_crimecount_call = tax_assess_vio_comb_crimecount.merge(
    call_count_3y, left_on='parcel_number_assess', right_index=True
).fillna({"call_count_50m_3y6m": 0})
```

```
[ ]: #combining 311 calls with parcel number dataset. replacing null values with zero
tax_assess_vio_comb_crimecount_call = tax_assess_vio_comb_crimecount_call.merge(
    call_count_all, left_on='parcel_number_assess', right_index=True
).fillna({"call_count_all": 0})
```

```
[ ]: tax_assess_vio_comb_crimecount_call.sort_values(['call_count_50m_3y6m'],
    ↪ascending = False).head()
```

```
[ ]:      parcel_number_assess  category_code_assess \
325514              481311400                      1
327446              481311500                      1
235014              882970195                      4
338443              481305800                      1
326998              481305700                      1

                                category_code_description_assess  census_tract_assess  depth_assess \
325514                           Single Family                  37.0          46.0
```

327446	Single Family	37.0	46.0
235014	Commercial	37.0	70.0
338443	Single Family	37.0	47.5
326998	Single Family	37.0	47.5
	location_assess	market_value_assess	year_built_assess
325514	2038 S OPAL ST	182900.0	1920
327446	2040 S OPAL ST	86900.0	1920
235014	2029 S 20TH ST	94900.0	1920
338443	2041 S OPAL ST	87300.0	1920
326998	2039 S OPAL ST	87300.0	1920
	zip_code_assess	lat_assess	...
325514	19145.0	-75.178721	...
327446	19145.0	-75.178728	...
235014	19145.0	-75.178928	...
338443	19145.0	-75.178464	...
326998	19145.0	-75.178454	...

	num_vacant_code_vio		geometry
325514	0.0	POINT (3688479.203 1100716.217)	
327446	7.0	POINT (3688480.631 1100712.148)	
235014	0.0	POINT (3688458.805 1100717.299)	
338443	0.0	POINT (3688502.889 1100718.176)	
326998	0.0	POINT (3688501.653 1100722.353)	

	buffer_50m		
325514	POLYGON ((3688529.203 1100716.217, 3688528.962...		
327446	POLYGON ((3688530.631 1100712.148, 3688530.391...		
235014	POLYGON ((3688508.805 1100717.299, 3688508.564...		
338443	POLYGON ((3688552.889 1100718.176, 3688552.649...		
326998	POLYGON ((3688551.653 1100722.353, 3688551.413...		

	buffer_500m	crime_count_50m_6m	
325514	POLYGON ((3688979.203 1100716.217, 3688976.795...	6	
327446	POLYGON ((3688980.631 1100712.148, 3688978.224...	5	
235014	POLYGON ((3688958.805 1100717.299, 3688956.397...	6	
338443	POLYGON ((3689002.889 1100718.176, 3689000.482...	4	
326998	POLYGON ((3689001.653 1100722.353, 3688999.246...	4	

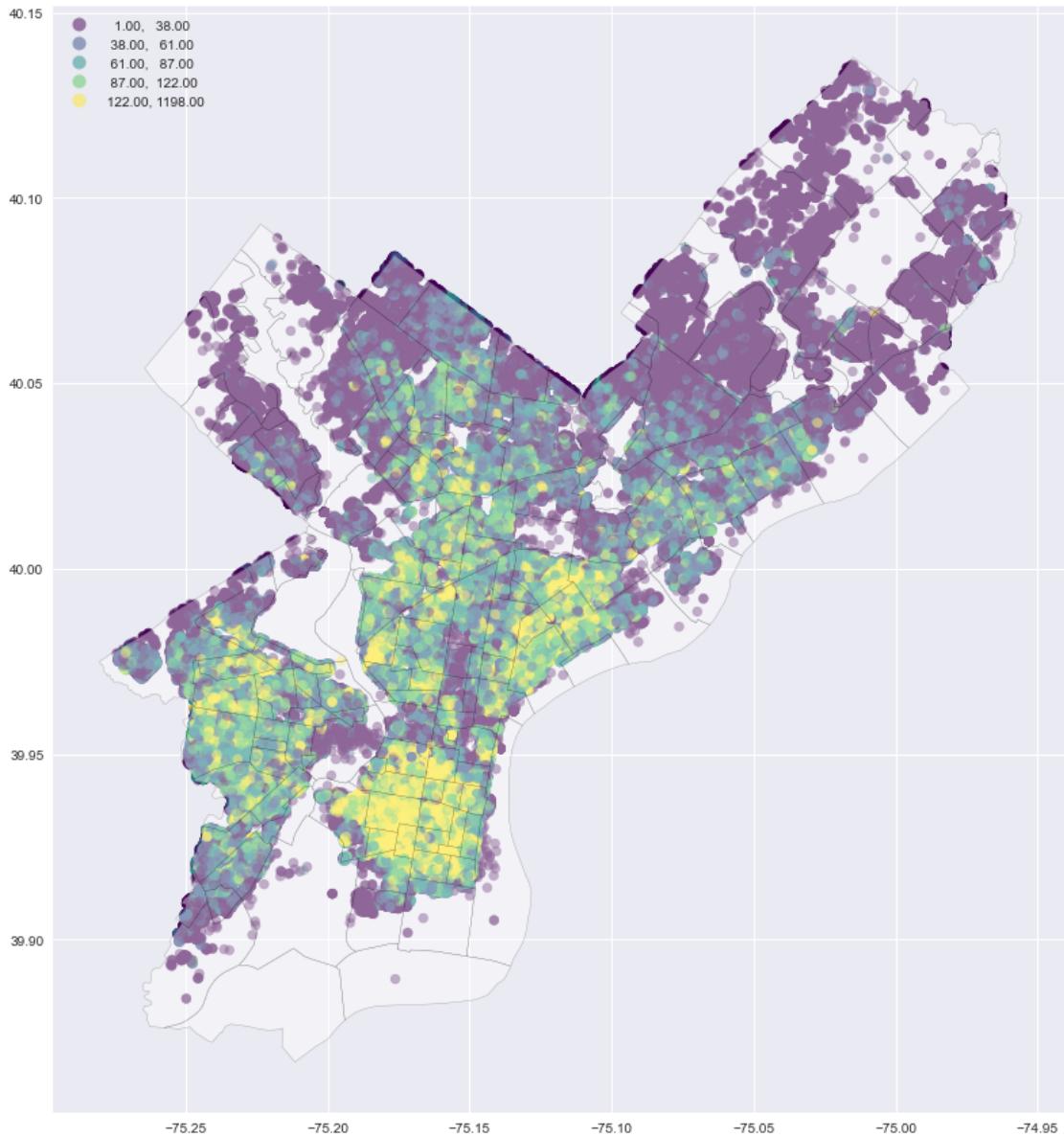
	crime_count_50m_3y6m	crime_count_50m_all	call_count_50m_3y6m
325514	25	66	515
327446	24	64	460
235014	31	59	446
338443	24	70	445
326998	22	73	444

```
call_count_50m_all
325514           1192
327446           1178
235014            860
338443            956
326998            916
```

[5 rows x 54 columns]

```
[ ]: #plotting where 311 calls are occurring with 50m of parcels
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
tax_assess_vio_comb_crimecount_call.to_crs(epsg = 4326).
    plot(column="call_count_50m_all",
          scheme="quantiles",
          alpha=0.5,
          legend=True,
          ax=ax)
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white", □
    edgecolor='black')
```

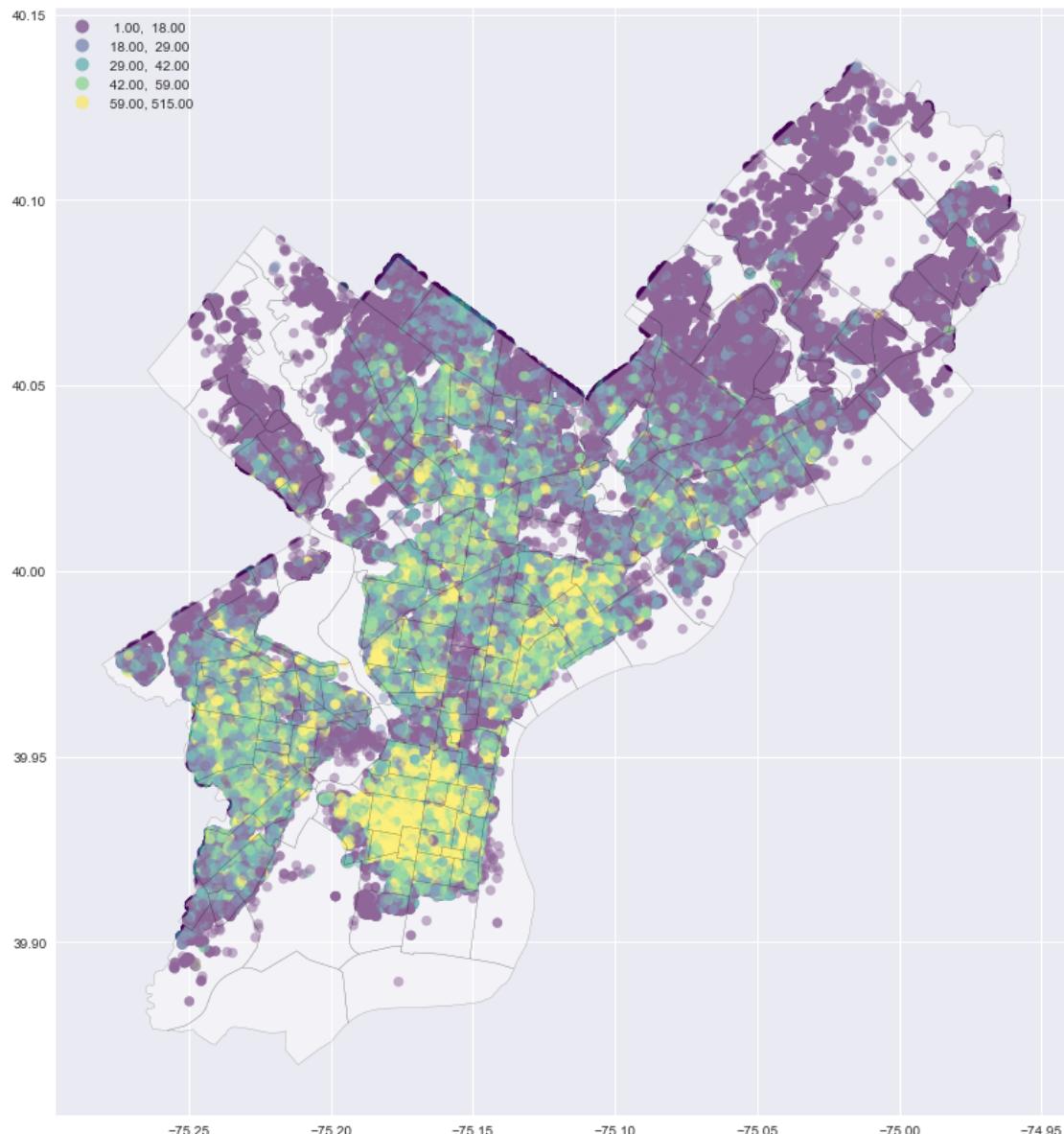
[]: <AxesSubplot:>



```
[ ]: #plotting where 311 calls are occurring with 50m of parcels in the last 3 year 6 months
fig, ax = plt.subplots(figsize =(15,15))
plt.style.use('seaborn')
tax_assess_vio_comb_crimecount_call.to_crs(epsg = 4326).
    plot(column="call_count_50m_3y6m",
          scheme="quantiles",
          alpha=0.5,
          legend=True,
          ax=ax)
```

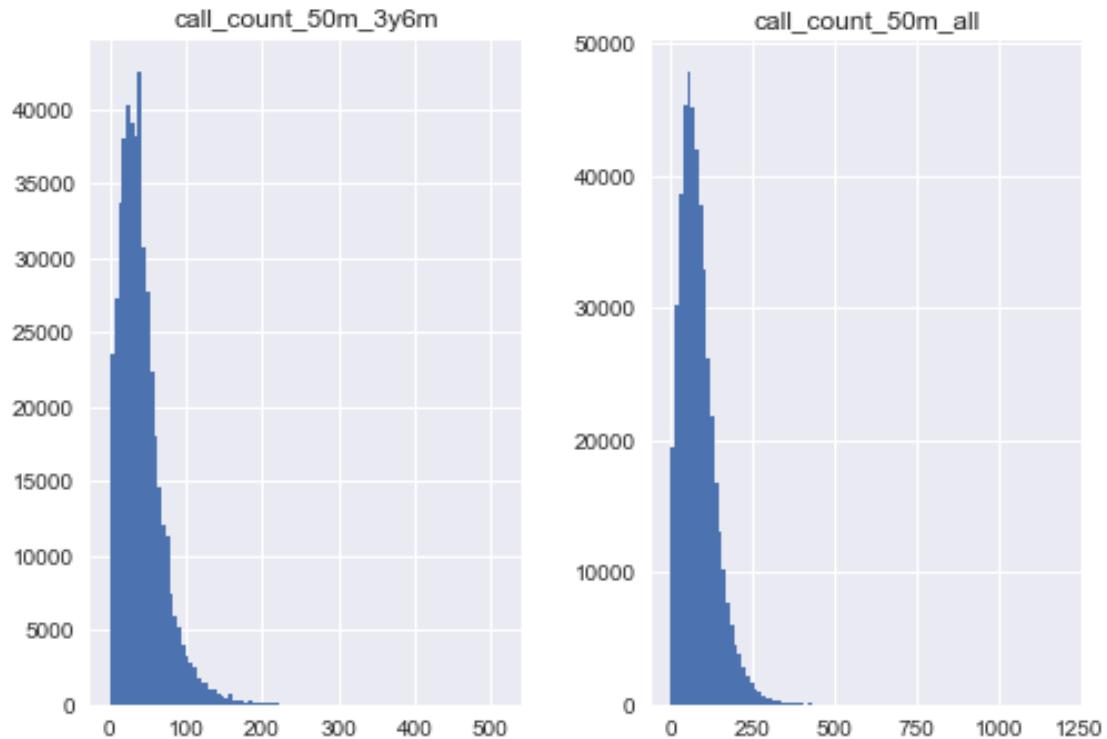
```
street_map.to_crs(epsg = 4326).plot(ax = ax, alpha = 0.4, color = "white",  
edgecolor='black')
```

[]: <AxesSubplot:>



```
[ ]: tax_assess_vio_comb_crimecount_call[['call_count_50m_3y6m','call_count_50m_all']].  
hist(bins = 100)
```

```
[ ]: array([[<AxesSubplot:title={'center':'call_count_50m_3y6m'}>,  
<AxesSubplot:title={'center':'call_count_50m_all'}>]],  
dtype=object)
```



```
[ ]: tax_assess_vio_comb_crimecount_call[['call_count_50m_3y6m','call_count_50m_all']].  
    ↪describe()
```

```
[ ]:      call_count_50m_3y6m  call_count_50m_all  
count      460901.000000      460901.000000  
mean       40.628506       83.574473  
std        28.781408       56.179405  
min        1.000000       1.000000  
25%       21.000000       44.000000  
50%       35.000000       74.000000  
75%       54.000000      111.000000  
max       515.000000      1198.000000
```

```
[ ]: #tax_assess_vio_comb_crimecount_call.to_csv("all_crimecount_call.csv")
```

0.2.22 Combining all data with ACS blockgroup level data

```
[ ]: acs = pd.read_csv("acs.csv", index_col=0)  
acs.head() # acs dataset
```

	TRACTCE10	BLKGRPCE10	GEOID10	Shape__Are	Shape__Len	\
0	10800	1	421010108001	1.742508e+06	8200.327170	
1	10800	2	421010108002	1.117026e+06	4364.980144	

```

2      10900          2  421010109002  4.706347e+05   3048.109084
3      11000          2  421010110002  1.172871e+06   5169.004282
4      11000          1  421010110001  1.531076e+06   10476.574129

                                         geometry  other_vacant \
0  POLYGON ((-75.1985147835869 39.9694490669934, ...
1  POLYGON ((-75.1978337828035 39.965705066843, ...
2  POLYGON ((-75.1876567798379 39.9644950666312, ...
3  POLYGON ((-75.2098357868676 39.9735130677027, ...
4  POLYGON ((-75.1985487837945 39.9733040682377, ...

total_housing  occupied  vacant  perc_vacant  perc_other_vacant  total_pop \
0            243       202      41    0.168724           0.168724     702
1            360       239     121    0.336111           0.227778     605
2            236       221      15    0.063559           0.029661     457
3            478       348     130    0.271967           0.152720     709
4            240       187      53    0.220833           0.083333     631

                                         area_m  pop_density_km
0  POLYGON ((3684773.96362486 1104249.064914737, ...
1  POLYGON ((3685015.051804622 1103905.1943166377...
2  POLYGON ((3685850.0367564433 1104181.114204087...
3  POLYGON ((3683707.7432979234 1104211.136855546...
4  POLYGON ((3684576.458607227 1104629.0618173946...

[ ]: acs = acs.add_suffix('_acs') # adding acs on the end
acs.head()

[ ]: TRACTCE10_acs  BLKGRPCE10_acs  GEOID10_acs  Shape__Are_acs \
0      10800          1  421010108001  1.742508e+06
1      10800          2  421010108002  1.117026e+06
2      10900          2  421010109002  4.706347e+05
3      11000          2  421010110002  1.172871e+06
4      11000          1  421010110001  1.531076e+06

Shape__Len_acs                                         geometry_acs \
0      8200.327170  POLYGON ((-75.1985147835869 39.9694490669934, ...
1      4364.980144  POLYGON ((-75.1978337828035 39.965705066843, ...
2      3048.109084  POLYGON ((-75.1876567798379 39.9644950666312, ...
3      5169.004282  POLYGON ((-75.2098357868676 39.9735130677027, ...
4      10476.574129  POLYGON ((-75.1985487837945 39.9733040682377, ...

other_vacant_acs  total_housing_acs  occupied_acs  vacant_acs \
0             41           243        202         41
1             82           360        239        121
2              7           236        221         15
3             73           478        348        130

```

```

4          20          240          187          53

    perc_vacant_acs  perc_other_vacant_acs  total_pop_acs  \
0          0.168724              0.168724          702
1          0.336111              0.227778          605
2          0.063559              0.029661          457
3          0.271967              0.152720          709
4          0.220833              0.083333          631

                                         area_m_acs  pop_density_km_acs
0  POLYGON ((3684773.96362486 1104249.064914737, ...
1  POLYGON ((3685015.051804622 1103905.1943166377...
2  POLYGON ((3685850.0367564433 1104181.114204087...
3  POLYGON ((3683707.7432979234 1104211.136855546...
4  POLYGON ((3684576.458607227 1104629.0618173946...

[ ]: tax_assess_vio_comb_crimecount_call.head()

[ ]: parcel_number_assess  category_code_assess  \
1          451106305          6
2          57166705          6
3          56143925          6
4          56143950          6
6          183009065          6

category_code_description_assess  census_tract_assess  depth_assess  \
1          Vacant Land          379.0          256.00
2          Vacant Land          142.0          43.53
3          Vacant Land          367.0          68.83
4          Vacant Land          367.0          68.83
6          Vacant Land          157.0          0.00

location_assess  market_value_assess  year_built_assess  zip_code_assess  \
1          2550 CASTOR AVE          0.0          0.0          19134.0
2          513 POPLAR ST          0.0          0.0          19123.0
3          428 FAIRMOUNT AVE          0.0          0.0          19123.0
4          442 FAIRMOUNT AVE          0.0          0.0          19123.0
6          105 GUEST ST          0.0          0.0          19122.0

lat_assess  ...  violationcode_diff_vio num_vacant_code_vio  \
1 -75.092534  ...          NaN          NaN
2 -75.146154  ...          NaN          NaN
3 -75.145586  ...          NaN          NaN
4 -75.146035  ...          NaN          NaN
6 -75.135164  ...          NaN          NaN

geometry  \

```

```

1 POINT (3691714.218 1110548.399)
2 POINT (3688875.025 1106049.599)
3 POINT (3689140.293 1105637.813)
4 POINT (3689101.484 1105629.488)
5 POINT (3689316.932 1107245.687)

                                buffer_50m \
1 POLYGON ((3691764.218 1110548.399, 3691763.977...
2 POLYGON ((3688925.025 1106049.599, 3688924.785...
3 POLYGON ((3689190.293 1105637.813, 3689190.053...
4 POLYGON ((3689151.484 1105629.488, 3689151.243...
5 POLYGON ((3689366.932 1107245.687, 3689366.691...

                                buffer_500m  crime_count_50m_6m \
1 POLYGON ((3692214.218 1110548.399, 3692211.810...      1
2 POLYGON ((3689375.025 1106049.599, 3689372.618...      2
3 POLYGON ((3689640.293 1105637.813, 3689637.886...      1
4 POLYGON ((3689601.484 1105629.488, 3689599.077...      4
5 POLYGON ((3689816.932 1107245.687, 3689814.525...      6

    crime_count_50m_3y6m  crime_count_50m_all  call_count_50m_3y6m \
1                  5                 8             3
2                 25                50            44
3                 20                40            31
4                 21                43            40
5                 19                32            74

    call_count_50m_all
1                  6
2                 97
3                 79
4                 96
5                156

```

[5 rows x 54 columns]

[]: tax_assess_vio_comb_crimecount_call.columns

[]: Index(['parcel_number_assess', 'category_code_assess',
 'category_code_description_assess', 'census_tract_assess',
 'depth_assess', 'location_assess', 'market_value_assess',
 'year_built_assess', 'zip_code_assess', 'lat_assess', 'lng_assess',
 'geometry_assess', 'vacant_assess', 'market_value_2015_assess',
 'market_value_2016_assess', 'market_value_2017_assess',
 'market_value_2018_assess', 'market_value_2019_assess',
 'market_value_2020_assess', 'market_value_2021_assess',
 'market_value_2022_assess', 'opa_number_tax', 'street_address_tax',

```
'zip_code_tax', 'num_years_owed_tax', 'most_recent_year_owed_tax',
'oldest_year_owed_tax', 'year_of_last_assessment_tax',
'general_building_description_tax', 'building_category_tax',
'is_actionable_tax', 'bankruptcy_tax', 'sheriff_sale_tax',
'sequestration_enforcement_tax', 'payment_agreement_tax',
'principal_due_tax', 'total_due_tax', 'total_assessment_tax', 'lat_tax',
'lng_tax', 'geometry_tax', 'opa_account_num_vio', 'casenumber_diff_vio',
'caserecreateddate_year_diff_vio', 'violationcode_diff_vio',
'num_vacant_code_vio', 'geometry', 'buffer_50m', 'buffer_500m',
'crime_count_50m_6m', 'crime_count_50m_3y6m', 'crime_count_50m_all',
'call_count_50m_3y6m', 'call_count_50m_all'],
dtype='object')
```

```
[ ]: tax_assess_vio_comb_crimecount_call['census_tract_assess'].nunique()
```

```
[ ]: 324
```

```
[ ]: tax_assess_vio_comb_crimecount_call['census_tract_assess'].tail()
```

```
[ ]: 581331    136.0
581332    192.0
581333    188.0
581334    203.0
581335    367.0
Name: census_tract_assess, dtype: float64
```

```
[ ]: tax_assess_vio_comb_crimecount_call['census_tract_assess'].astype(int).unique()
#census tract is written different so we cannot join based on this
```

```
[ ]: array([379, 142, 367, 157, 16, 143, 158, 28, 1, 10, 144, 373, 136,
       14, 366, 125, 161, 27, 24, 15, 7, 12, 25, 294, 353, 3,
       376, 387, 253, 199, 140, 358, 329, 87, 132, 19, 214, 160, 112,
       61, 74, 95, 263, 302, 33, 6, 378, 86, 4, 348, 107, 807,
       331, 344, 362, 342, 183, 236, 351, 361, 384, 42, 70, 73, 85,
       90, 164, 172, 203, 337, 36, 38, 336, 101, 104, 152, 249, 210,
       245, 257, 9, 809, 195, 178, 180, 188, 190, 191, 198, 201, 202,
       72, 105, 240, 271, 275, 281, 293, 300, 238, 243, 254, 268, 279,
       283, 284, 286, 103, 110, 113, 117, 383, 39, 67, 206, 273, 280,
       145, 147, 149, 169, 173, 179, 184, 192, 207, 213, 55, 60, 62,
       64, 290, 291, 316, 318, 323, 325, 341, 69, 352, 349, 390, 365,
       174, 345, 319, 65, 137, 338, 357, 277, 386, 134, 382, 372, 119,
       285, 79, 363, 2, 231, 11, 17, 270, 246, 82, 321, 176, 88,
       310, 377, 282, 8, 21, 389, 269, 177, 98, 217, 133, 115, 385,
       18, 204, 215, 41, 317, 299, 356, 380, 218, 131, 388, 151, 100,
       81, 165, 301, 163, 5, 360, 63, 141, 84, 153, 40, 287, 306,
       237, 276, 320, 326, 267, 106, 307, 289, 264, 175, 77, 359, 241,
       20, 381, 247, 265, 96, 252, 29, 121, 211, 66, 13, 340, 108,
```

```
114, 93, 83, 209, 167, 288, 205, 54, 23, 71, 335, 78, 197,  
266, 138, 30, 139, 102, 135, 166, 162, 339, 292, 315, 94, 37,  
146, 32, 212, 200, 168, 111, 256, 274, 244, 242, 22, 80, 156,  
312, 148, 248, 308, 92, 91, 171, 305, 278, 118, 31, 375, 334,  
347, 109, 170, 261, 313, 333, 346, 56, 255, 330, 298, 311, 314,  
239, 120, 355, 262, 122, 369, 332, 272, 208, 235, 309, 891, 258,  
803, 259, 219, 216, 364, 800, 260, 220, 802, 801, 806, 50])
```

```
[ ]: acs['TRACTCE10'].unique()
```

```
[ ]: array([ 10800, 10900, 11000, 11100, 11200, 11300, 11400, 26600,  
26700, 26800, 26900, 27000, 27100, 27200, 27300, 36202,  
36201, 36203, 29200, 34803, 5000, 3800, 35302, 35800,  
8302, 16300, 17400, 16400, 17300, 16800, 17100, 20600,  
20200, 17500, 19900, 20000, 20300, 17602, 17601, 19200,  
11500, 11800, 11900, 12000, 12100, 12204, 27402, 27401,  
27500, 27600, 27700, 1002, 100, 36600, 36900, 1001,  
1600, 200, 36700, 402, 980500, 18800, 17900, 18400,  
29800, 32100, 29900, 19100, 19000, 30100, 19800, 28800,  
28700, 12500, 13100, 13200, 13300, 13402, 13401, 13500,  
13602, 13700, 13800, 13900, 14000, 14200, 27800, 27901,  
27902, 28000, 28100, 28200, 28300, 28400, 28600, 300,  
700, 980000, 804, 803, 1202, 980900, 2500, 980700,  
980100, 21800, 3001, 20500, 21100, 23600, 23700, 24100,  
24000, 24200, 24400, 24500, 26400, 25200, 25400, 25500,  
25600, 14300, 14400, 14600, 14700, 14900, 15102, 15101,  
15200, 15300, 37700, 15700, 15800, 28902, 28901, 29000,  
33702, 980200, 37200, 37300, 4201, 980600, 19502, 5600,  
5500, 8101, 8601, 980800, 25800, 26000, 26100, 26200,  
600, 902, 901, 1102, 1101, 1201, 37800, 16000,  
16100, 16200, 16500, 16600, 16702, 29100, 29400, 30000,  
9801, 10000, 10300, 10400, 24800, 37500, 12201, 12203,  
37600, 31600, 36301, 28500, 29300, 31800, 30600, 30800,  
30900, 33500, 33400, 31000, 1300, 1400, 1500, 1700,  
1800, 1900, 2100, 2200, 2300, 2400, 2701, 2702,  
16701, 3901, 16902, 16901, 17000, 17201, 17202, 30200,  
39000, 30502, 30501, 36303, 37900, 18300, 31200, 33200,  
31900, 32500, 32300, 32600, 33000, 33300, 34400, 35602,  
34802, 35900, 2802, 2801, 2900, 3002, 3100, 3200,  
3300, 35301, 17702, 17701, 17800, 30700, 31102, 20800,  
34200, 20900, 21000, 21600, 21900, 22000, 23100, 801,  
3600, 38300, 6000, 7200, 8602, 3701, 3902, 3702,  
36502, 4101, 18002, 18001, 35702, 38200, 31101, 31300,  
31401, 31402, 31501, 31502, 35500, 36000, 980300, 36100,  
8702, 9802, 38900, 13601, 19501, 4002, 4001, 4102,  
4202, 35701, 19700, 31700, 32000, 989100, 35100, 36501,  
35601, 38100, 38000, 20101, 5400, 20102, 20400, 20700,
```

```
32900, 34501, 34502, 34801, 34000, 21400, 21500, 38400,
23500, 38500, 38700, 38600, 38800, 6100, 6200, 6300,
6400, 6500, 6600, 6700, 21200, 21300, 33102, 33101,
33600, 980400, 9100, 500, 401, 6900, 7000, 7101,
7102, 7300, 21700, 23800, 33701, 33800, 33900, 34100,
2000, 7400, 7700, 7800, 7900, 8000, 8102, 23900,
24300, 26301, 24600, 34600, 34701, 8200, 8400, 8500,
9300, 9000, 9200, 9400, 10200, 8301, 24700, 24900,
25300, 34900, 35200, 9500, 9600, 10100, 10500, 10600,
10700, 11700, 8701, 8802, 8801, 25700, 25900, 14100,
26302, 26500, 36400, 34702, 36302, 14800, 14500, 15600])
```

```
[ ]: acs['TRACTCE10'].nunique()
```

```
[ ]: 384
```

```
[ ]: acs['TRACTCE10'].head()
```

```
[ ]: 0    10800
1    10800
2    10900
3    11000
4    11000
Name: TRACTCE10, dtype: int64
```

```
[ ]: acs.tail()
```

```
[ ]:   TRACTCE10  BLKGRPCE10      GEOID10      Shape__Are  Shape__Len \
1331     16200          2  421010162002  2.405057e+06  6772.642577
1332     15600          2  421010156002  1.909996e+06  6128.309258
1333     15700          3  421010157003  3.285646e+06  9381.401996
1334     17900          5  421010179005  2.968631e+06  9625.938356
1335     17800          6  421010178006  1.489867e+06  6889.853029
```

```
                                geometry  other_vacant \
1331  POLYGON ((-75.14350 39.98628, -75.14284 39.986...
1332  POLYGON ((-75.14102 39.97963, -75.14135 39.978...
1333  POLYGON ((-75.13818 39.97925, -75.13808 39.979...
1334  POLYGON ((-75.11561 39.98356, -75.11604 39.982...
1335  POLYGON ((-75.11675 39.98866, -75.11733 39.988...
```

```
      total_housing  occupied  vacant  perc_vacant  perc_other_vacant \
1331            386       303      83      0.215026        0.093264
1332            347       283      64      0.184438        0.086455
1333            617       462     155      0.251216        0.102107
1334            571       571       0      0.000000        0.000000
1335            179       118      61      0.340782        0.340782
```

```
total_pop                                area_m \
1331      888  POLYGON ((3688103.934 1108053.020, 3688158.341...
1332      785  POLYGON ((3688629.437 1107491.127, 3688680.164...
1333     1160  POLYGON ((3688863.822 1107564.822, 3688872.510...
1334     2126  POLYGON ((3690361.526 1108869.140, 3690357.571...
1335      382  POLYGON ((3690016.095 1109328.872, 3689987.602...

pop_density_km
1331    3974.196182
1332    4423.840293
1333    3800.142471
1334    7708.477368
1335    2759.796332
```

```
[ ]: acs.shape
```

```
[ ]: (1336, 15)
```

```
[ ]: acs.crs
```

```
[ ]: <Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

```
[ ]: acs = acs.to_crs(epsg = 3311)
acs.crs
```

```
[ ]: <Derived Projected CRS: EPSG:3311>
Name: NAD83(HARN) / California Albers
Axis Info [cartesian]:
- X[east]: Easting (metre)
- Y[north]: Northing (metre)
Area of Use:
- name: United States (USA) - California.
- bounds: (-124.45, 32.53, -114.12, 42.01)
Coordinate Operation:
- name: California Albers
- method: Albers Equal Area
```

```
Datum: NAD83 (High Accuracy Reference Network)
- Ellipsoid: GRS 1980
- Prime Meridian: Greenwich
```

```
[ ]: tax_assess_vio_comb_crimecount_call.crs
```

```
[ ]: <Derived Projected CRS: EPSG:3311>
Name: NAD83(HARN) / California Albers
Axis Info [cartesian]:
- X[east]: Easting (metre)
- Y[north]: Northing (metre)
Area of Use:
- name: United States (USA) - California.
- bounds: (-124.45, 32.53, -114.12, 42.01)
Coordinate Operation:
- name: California Albers
- method: Albers Equal Area
Datum: NAD83 (High Accuracy Reference Network)
- Ellipsoid: GRS 1980
- Prime Meridian: Greenwich
```

```
[ ]: acs = acs.rename(columns = {'geometry_acs':'geometry'}) # naming to geometry as
      ↪it was anot joining when _acs was added
acs.head()
```

```
[ ]:   TRACTCE10_acs  BLKGRPCE10_acs  GEOID10_acs  Shape__Are_acs  \
0          10800                  1  421010108001  1.742508e+06
1          10800                  2  421010108002  1.117026e+06
2          10900                  2  421010109002  4.706347e+05
3          11000                  2  421010110002  1.172871e+06
4          11000                  1  421010110001  1.531076e+06

      Shape__Len_acs                                     geometry  \
0    8200.327170  POLYGON ((-75.19851 39.96945, -75.19744 39.969...
1    4364.980144  POLYGON ((-75.19783 39.96571, -75.20006 39.965...
2    3048.109084  POLYGON ((-75.18766 39.96450, -75.18755 39.963...
3    5169.004282  POLYGON ((-75.20984 39.97351, -75.21221 39.973...
4   10476.574129  POLYGON ((-75.19855 39.97330, -75.19854 39.973...

      other_vacant_acs  total_housing_acs  occupied_acs  vacant_acs  \
0              41           243         202          41
1              82           360         239          121
2               7           236         221           15
3              73           478         348          130
4              20           240         187           53

      perc_vacant_acs  perc_other_vacant_acs  total_pop_acs  \

```

```

0      0.168724      0.168724      702
1      0.336111      0.227778      605
2      0.063559      0.029661      457
3      0.271967      0.152720      709
4      0.220833      0.083333      631

                           area_m_acs  pop_density_km_acs
0  POLYGON ((3684773.964 1104249.065, 3684870.739...  4336.368995
1  POLYGON ((3685015.052 1103905.194, 3684857.753...  5829.834621
2  POLYGON ((3685850.037 1104181.114, 3685886.064...  10451.930347
3  POLYGON ((3683707.743 1104211.137, 3683540.308...  6506.669922
4  POLYGON ((3684576.459 1104629.062, 3684590.061...  4436.045393

```

[]: #joining parcel number dataset with acs dataset
acs_city = gpd.sjoin(tax_assess_vio_comb_crimecount_call, acs, op='within')

```
/Users/priankaball/opt/anaconda3/envs/geo_env/lib/python3.10/site-
packages/IPython/core/interactiveshell.py:3377: FutureWarning: The `op`  

parameter is deprecated and will be removed in a future release. Please use the  

`predicate` parameter instead.
```

```
    if (await self.run_code(code, result, async_=asy)):
```

[]: acs_city.shape # same shape as the other file

[]: (460901, 69)

[]: tax_assess_vio_comb_crimecount_call.shape

[]: (460901, 54)

[]: acs_city.isna().sum()

```

[ ]: parcel_number_assess          0
category_code_assess              0
category_code_description_assess  64
census_tract_assess              0
depth_assess                     507
...
perc_vacant_acs                 157
perc_other_vacant_acs           157
total_pop_acs                   0
area_m_acs                       0
pop_density_km_acs               0
Length: 69, dtype: int64

```

[]: acs_city.columns

```
[ ]: Index(['parcel_number_assess', 'category_code_assess',
   'category_code_description_assess', 'census_tract_assess',
   'depth_assess', 'location_assess', 'market_value_assess',
   'year_built_assess', 'zip_code_assess', 'lat_assess', 'lng_assess',
   'geometry_assess', 'vacant_assess', 'market_value_2015_assess',
   'market_value_2016_assess', 'market_value_2017_assess',
   'market_value_2018_assess', 'market_value_2019_assess',
   'market_value_2020_assess', 'market_value_2021_assess',
   'market_value_2022_assess', 'opa_number_tax', 'street_address_tax',
   'zip_code_tax', 'num_years_owed_tax', 'most_recent_year_owed_tax',
   'oldest_year_owed_tax', 'year_of_last_assessment_tax',
   'general_building_description_tax', 'building_category_tax',
   'is_actionable_tax', 'bankruptcy_tax', 'sheriff_sale_tax',
   'sequestration_enforcement_tax', 'payment_agreement_tax',
   'principal_due_tax', 'total_due_tax', 'total_assessment_tax', 'lat_tax',
   'lng_tax', 'geometry_tax', 'opa_account_num_vio', 'casenumber_diff_vio',
   'casecreateddate_year_diff_vio', 'violationcode_diff_vio',
   'num_vacant_code_vio', 'geometry', 'buffer_50m', 'buffer_500m',
   'crime_count_50m_6m', 'crime_count_50m_3y6m', 'crime_count_50m_all',
   'call_count_50m_3y6m', 'call_count_50m_all', 'index_right',
   'TRACTCE10_acs', 'BLKGRPCE10_acs', 'GEOID10_acs', 'Shape__Are_acs',
   'Shape__Len_acs', 'other_vacant_acs', 'total_housing_acs',
   'occupied_acs', 'vacant_acs', 'perc_vacant_acs',
   'perc_other_vacant_acs', 'total_pop_acs', 'area_m_acs',
   'pop_density_km_acs'],
  dtype='object')
```

```
[ ]: acs_city[acs_city[['perc_vacant_acs','perc_other_vacant_acs']].isna().
   ↪any(axis=1)]
```

	parcel_number_assess	category_code_assess	category_code_description_assess	census_tract_assess	depth_assess
199	884136200	5	Industrial	807.0	387.00
484	885672160	6	Vacant Land	807.0	692.00
1860	882280400	4	Commercial	807.0	623.00
5481	882055190	4			
29875	882898100	4			
...			
536744	786441501	4			
218769	875106340	6			
530833	788020600	4			
537425	788020010	4			
221776	788008310	4			

5481	Commercial	807.0	1065.00
29875	Commercial	807.0	260.25
...
536744	Commercial	803.0	0.00
218769	Vacant Land	806.0	76.50
530833	Commercial	806.0	500.00
537425	Commercial	806.0	1025.60
221776	Commercial	50.0	1.00

	location_assess	market_value_assess	year_built_assess	\
199	15 OREGON AVE	7474400.0	1960.0	
484	3506 S CHRIS COLUMBUS BLV	662800.0	0.0	
1860	2908 S CHRIS COLUMBUS BLV	1420600.0	2006.0	
5481	1851 S CHRIS COLUMBUS BLV	17526100.0	1990.0	
29875	1510 S CHRIS COLUMBUS BLV	1310700.0	1900.0	
...	
536744	2650-60 RED LION RD	3605000.0	1978.0	
218769	3850 S 11TH ST	2264500.0	0	
530833	3601 S BROAD ST	253215000.0	2006.0	
537425	1 CITIZENS BANK WAY	405637000.0	2004.0	
221776	5000L S BROAD ST	532500.0	1903	

	zip_code_assess	lat_assess	...	Shape_Len_acs	other_vacant_acs	\
199	19148.0	-75.143797	...	48816.591063	0	
484	19148.0	-75.141400	...	48816.591063	0	
1860	19148.0	-75.141400	...	48816.591063	0	
5481	19148.0	-75.140591	...	48816.591063	0	
29875	19147.0	-75.145437	...	48816.591063	0	
...	
536744	19154.0	-75.013990	...	36643.990690	0	
218769	19148.0	-75.171806	...	21280.073657	0	
530833	19148.0	-75.171806	...	21280.073657	0	
537425	19148.0	-75.166310	...	21280.073657	0	
221776	19112.0	-75.176162	...	48030.059127	0	

	total_housing_acs	occupied_acs	vacant_acs	perc_vacant_acs	\
199	0	0	0	NaN	
484	0	0	0	NaN	
1860	0	0	0	NaN	
5481	0	0	0	NaN	
29875	0	0	0	NaN	
...	
536744	0	0	0	NaN	
218769	0	0	0	NaN	
530833	0	0	0	NaN	
537425	0	0	0	NaN	
221776	0	0	0	NaN	

```

    perc_other_vacant_acs  total_pop_acs  \
199                 NaN          0
484                 NaN          0
1860                NaN          0
5481                NaN          0
29875               NaN          0
...
536744               ...        ...
218769               NaN          0
530833               NaN          0
537425               NaN          0
221776               NaN          0

                                         area_m_acs  pop_density_km_acs
199      POLYGON ((3691173.723 1102523.854, 3691486.833...          0.0
484      POLYGON ((3691173.723 1102523.854, 3691486.833...          0.0
1860     POLYGON ((3691173.723 1102523.854, 3691486.833...          0.0
5481     POLYGON ((3691173.723 1102523.854, 3691486.833...          0.0
29875    POLYGON ((3691173.723 1102523.854, 3691486.833...          0.0
...
536744    ...          ...
218769    POLYGON ((3690456.477 1099727.098, 3690552.027...          0.0
530833    POLYGON ((3690456.477 1099727.098, 3690552.027...          0.0
537425    POLYGON ((3690456.477 1099727.098, 3690552.027...          0.0
221776    POLYGON ((3688587.469 1097647.692, 3688590.806...          0.0

```

[157 rows x 69 columns]

```
[ ]: #replacing the null values in per_vacant and perc_other_vacant with 0.
# This happened as the vacant lot and other vacant lot was 0 for some of the
# blockgroups
acs_city[['perc_vacant_acs', 'perc_other_vacant_acs']] =_
    acs_city[['perc_vacant_acs', 'perc_other_vacant_acs']].fillna(0)
```

```
[ ]: #acs_city.to_csv("final_acs_city.csv")
```

```
[ ]: acs_city[['market_value_assess', 'market_value_2021_assess']].isna().sum()
```

```
[ ]: market_value_assess      177
market_value_2021_assess    321
dtype: int64
```

```
[ ]: acs_city[['census_tract_assess', 'TRACTCE10_acs']]
```

```
[ ]:      census_tract_assess  TRACTCE10_acs
1                  379.0       37900
```

```
4230           379.0      37900
36251          379.0      37900
40957          379.0      37900
71601          379.0      37900
...
556166         362.0      36201
556169         362.0      36201
556171         362.0      36201
556593         362.0      36201
570565         362.0      36201
```

[460901 rows x 2 columns]

```
[ ]: acs_city['year_of_last_assessment_tax'].value_counts()
```

```
[ ]: 2021.0    61641
Name: year_of_last_assessment_tax, dtype: int64
```

```
[ ]: acs_city['building_category_tax'].value_counts()
```

```
[ ]: residential    56555
commercial       5085
Name: building_category_tax, dtype: int64
```

```
[ ]: acs_city['general_building_description_tax'].value_counts()
```

```
[ ]: house                  43518
vacantLand              9734
apartmentSmall          3100
mixedUsage                2232
condo                     663
industrial                 394
retail                      371
nonProfit                   287
apartmentLarge            284
miscCommercial             266
garage                      216
parkingLot                  182
miscResidential             174
theater_stadium_other amuse   72
Restaurant_Bar               61
officeBuilding                 49
hotel                        20
parking_garage                  13
bank                           3
utility                         1
Name: general_building_description_tax, dtype: int64
```

```
[ ]: acs_city['vacant_assess'].value_counts()

[ ]: 0    424202
     1    36699
Name: vacant_assess, dtype: int64

[ ]: acs_city.iloc[:,60:70].head(10)

[ ]:      other_vacant_acs  total_housing_acs  occupied_acs  vacant_acs  \
1                  48            547          499          48
4230                48            547          499          48
36251                48            547          499          48
40957                48            547          499          48
71601                48            547          499          48
76568                48            547          499          48
126925               48            547          499          48
143871               48            547          499          48
185976               48            547          499          48
187366               48            547          499          48

      perc_vacant_acs  perc_other_vacant_acs  total_pop_acs  \
1           0.087751            0.087751        1044
4230         0.087751            0.087751        1044
36251         0.087751            0.087751        1044
40957         0.087751            0.087751        1044
71601         0.087751            0.087751        1044
76568         0.087751            0.087751        1044
126925         0.087751            0.087751        1044
143871         0.087751            0.087751        1044
185976         0.087751            0.087751        1044
187366         0.087751            0.087751        1044

      area_m_acs  pop_density_km_acs
1      POLYGON ((3692491.495 1111202.304, 3692467.807... 1335.759173
4230    POLYGON ((3692491.495 1111202.304, 3692467.807... 1335.759173
36251    POLYGON ((3692491.495 1111202.304, 3692467.807... 1335.759173
40957    POLYGON ((3692491.495 1111202.304, 3692467.807... 1335.759173
71601    POLYGON ((3692491.495 1111202.304, 3692467.807... 1335.759173
76568    POLYGON ((3692491.495 1111202.304, 3692467.807... 1335.759173
126925   POLYGON ((3692491.495 1111202.304, 3692467.807... 1335.759173
143871   POLYGON ((3692491.495 1111202.304, 3692467.807... 1335.759173
185976   POLYGON ((3692491.495 1111202.304, 3692467.807... 1335.759173
187366   POLYGON ((3692491.495 1111202.304, 3692467.807... 1335.759173

[ ]: acs_city_final = acs_city.drop(['category_code_assess', 'category_code_description_assess', 'census_tract_assess',
```

```

    ↵'location_assess', 'market_value_2022_assess', 'market_value_assess', 'opa_number_tax',
        'street_address_tax', 'zip_code_tax', ↵
    ↵'general_building_description_tax', 'most_recent_year_owed_tax',
        'oldest_year_owed_tax', 'year_of_last_assessment_tax', ↵
    ↵'lat_tax', 'lng_tax', 'geometry_tax',
        'opa_account_num_vio', 'buffer_500m', 'index_right'], axis = 1)
    # removing columns that might be not necessary or where vacant ↵
    ↵lot is mentioned

```

```
[ ]: acs_city_final = acs_city_final.reset_index(drop = True)
acs_city_final.head()
```

	parcel_number_assess	depth_assess	year_built_assess	zip_code_assess	geometry_assess	vacant_assess	market_value_2015_assess	market_value_2016_assess
0	451106305	256.00	0.0	19134.0	POINT (-75.09253430091161 39.99145897715201)	1	0.0	0.0
1	882021470	64.00	1929.0	19137.0	POINT (-75.08780743936106 39.99142427758184)	0	346400.0	346400.0
2	882059200	200.75	1980	19137.0	POINT (-75.09359890860864 39.9946039149071)	0	1326900.0	1326900.0
3	882076300	279.81	2008	19134.0	POINT (-75.0952607737023 39.99364667194213)	0	1735600.0	1735600.0
4	453297000	248.00	1920	19137.0	POINT (-75.08459994726104 39.99212306645117)	0	172100.0	172100.0
...
460896	662369900	95.00	1973.0	19154.0	POINT (-74.98845008414838 40.08349387778363)	0	186000.0	186000.0
460897	662371300	95.00	1973.0	19154.0	POINT (-74.98745459827565 40.083163199092816)	0	346400.0	346400.0
460898	662371100	95.00	1973.0	19154.0	POINT (-74.98758897475999 40.083196702045214)	0	1326900.0	1326900.0
460899	662420600	90.00	1966.0	19154.0	POINT (-74.98518964927634 40.08319350520666)	0	1735600.0	1735600.0
460900	662404400	92.00	1974.0	19154.0	POINT (-74.98966778788741 40.08410597385051)	0	0.0	0.0
...
460896	-74.988450	40.083494	POINT (-74.98845008414838 40.08349387778363)	0	0.0	0	0.0	0.0
460897	-74.987455	40.083163	POINT (-74.98745459827565 40.083163199092816)	0	346400.0	0	346400.0	346400.0
460898	-74.987589	40.083197	POINT (-74.98758897475999 40.083196702045214)	0	1326900.0	0	1326900.0	1326900.0
460899	-74.985190	40.083194	POINT (-74.98518964927634 40.08319350520666)	0	1735600.0	0	1735600.0	1735600.0
460900	-74.989668	40.084106	POINT (-74.98966778788741 40.08410597385051)	0	0.0	0	0.0	0.0

460897	0	186000.0	186000.0
460898	0	186000.0	186000.0
460899	0	181000.0	181000.0
460900	0	192500.0	192500.0
...
0	...	12268.026027	48
1	...	12268.026027	48
2	...	12268.026027	48
3	...	12268.026027	48
4	...	12268.026027	48
...
460896	...	8707.673782	23
460897	...	8707.673782	23
460898	...	8707.673782	23
460899	...	8707.673782	23
460900	...	8707.673782	23
...
0	occupied_acs	499	48
1	occupied_acs	499	48
2	occupied_acs	499	48
3	occupied_acs	499	48
4	occupied_acs	499	48
...
460896	...	296	23
460897	...	296	23
460898	...	296	23
460899	...	296	23
460900	...	296	23
...
0	total_pop_acs	1044	POLYGON ((3692491.495 1111202.304, 3692467.807...
1	total_pop_acs	1044	POLYGON ((3692491.495 1111202.304, 3692467.807...
2	total_pop_acs	1044	POLYGON ((3692491.495 1111202.304, 3692467.807...
3	total_pop_acs	1044	POLYGON ((3692491.495 1111202.304, 3692467.807...
4	total_pop_acs	1044	POLYGON ((3692491.495 1111202.304, 3692467.807...
...
460896	...	784	POLYGON ((3695446.376 1123816.005, 3695348.544...
460897	...	784	POLYGON ((3695446.376 1123816.005, 3695348.544...
460898	...	784	POLYGON ((3695446.376 1123816.005, 3695348.544...
460899	...	784	POLYGON ((3695446.376 1123816.005, 3695348.544...
460900	...	784	POLYGON ((3695446.376 1123816.005, 3695348.544...
...
0	pop_density_km_acs	1335.759173	
1	pop_density_km_acs	1335.759173	

```
2          1335.759173
3          1335.759173
4          1335.759173
...
       ...
460896      1928.816766
460897      1928.816766
460898      1928.816766
460899      1928.816766
460900      1928.816766
```

```
[460901 rows x 50 columns]
```

```
[ ]: acs_city_final.columns
```

```
[ ]: Index(['parcel_number_assess', 'depth_assess', 'year_built_assess',
       'zip_code_assess', 'lat_assess', 'lng_assess', 'geometry_assess',
       'vacant_assess', 'market_value_2015_assess', 'market_value_2016_assess',
       'market_value_2017_assess', 'market_value_2018_assess',
       'market_value_2019_assess', 'market_value_2020_assess',
       'market_value_2021_assess', 'num_years_owed_tax',
       'building_category_tax', 'is_actionable_tax', 'bankruptcy_tax',
       'sheriff_sale_tax', 'sequestration_enforcement_tax',
       'payment_agreement_tax', 'principal_due_tax', 'total_due_tax',
       'total_assessment_tax', 'casenumber_diff_vio',
       'casetreateddate_year_diff_vio', 'violationcode_diff_vio',
       'num_vacant_code_vio', 'geometry', 'buffer_50m', 'crime_count_50m_6m',
       'crime_count_50m_3y6m', 'crime_count_50m_all', 'call_count_50m_3y6m',
       'call_count_50m_all', 'TRACTCE10_acs', 'BLKGRPCE10_acs', 'GEOID10_acs',
       'Shape__Are_acs', 'Shape__Len_acs', 'other_vacant_acs',
       'total_housing_acs', 'occupied_acs', 'vacant_acs', 'perc_vacant_acs',
       'perc_other_vacant_acs', 'total_pop_acs', 'area_m_acs',
       'pop_density_km_acs'],
      dtype='object')
```

```
[ ]: #acs_city_final.to_csv('acs_city_final.csv') # data to be used in ML model
```