DSC170 HateCrime Projects

May 22, 2020

1 Hate Crime in San Diego

Team Members:

- Pratyush Juneja: A14884072
- Aurelio Barrios:A15290207

2 Why Hate Crime? What do we hope to Accomplish?

The reason we chose this topic was initially because of an article we found detailing rates of Hate Crimes in San Diego since 2015. This definitely does not deserve to be a 21st Century Issue. Where is this hate crime occuring? What is causing this hate crime? Questions like these are just the tip of the iceberg of what we were hoping to accomplish through this project. This coupled with the recent rise in hatecrime towards asians due to the recent Covid19 outbreak made us want to do a deeper dive into this project

The goal of our project was to try and understand hate crimes spatially across San Diego. More specifically we wanted to see how the diversity of a specific area affected the hate crime rate in that area. Our initial thought was that areas with more diversity would mean that the hate crime rate in that area would be smaller. This belief arose from the idea that since people would be accustomed to being around people of different ethnic backgrounds they would therefore be more accepting of each other. Another observation we wanted to make was how average income levels of specific areas were related to the hate crime rate in these areas. Income is a huge factor in peoples day to day lives so we knew that income was an important attribute to consider. We recognize that San Diego is a place that is culturally diverse, that is why answering the question posed is extremely valuable to the San Diego community. Understanding the rise in hate crime will help us find solutions to this problem making San Diego a better and safer place. This is also why investigating this could prove to be valuable to the San Diego police force. Understanding this issue could lead to a police force that is better equipped to deal with and prevent these types of crimes and hopefully eventually eradicate the issue altogether. From a business case perspective, this tool could prove to be very valuable to organizations that aim to promote cultural diversity in the community. For example the Raza Resource Center here at UCSD aims to establish a more inclusive campus. If we apply our tool to organizations like this one but that are bigger and that aim to improve communities there would be a significant improvement in the communities. This tool can provide danger areas where there is a high hate crime rate compared to other areas which these organizations can target in order to improve inclusion. These organizations can provide events or meetings for these areas in order to diminish hate crime rates. Of course this tool can also be used for a number of other things. For example travelers can use this tool to determine where hate crime rates are high in order to avoid these areas especially if these travelers are foreigners. But this simply avoids the issue rather than try and help improve the issue, so the preffered use of this tool would be to help eradicate the issue.

We could try and figure out based on our information about what steps are required to go forward and identifying problem areas would be the first option in this.

3 Background and Literature

1. "FBI: 76 Hate Crimes Reported in San Diego in 2018", San Diego Tribune, https://www.sandiegouniontribune.com/news/public-safety/story/2019-11-12/fbi-76-hate-%20crimes-reported-in-san-diego-county-in-2018 This article is what inspired us to think of this issue. Whilst reading this article it became clear to us that hate crimes are still an issue in present day when they shouldn't be. The main mention that caught our attention was the mention that a majority of these hate crimes were motivated by racial and ethnic bias (46). This to us was completely unacceptable and therfore inspired us to investigate further.
2. "SDPD Has Investigated More Than 200 Alleged Hate Crimes Since 2015", NBC San Diego, https://www.nbcsandiego.com/news/local/san-diego-police-have-investigatedmore-than-200-alleged-hate-crimes-since-2015/168398/ This source proved to be very helpful in moving us in the direction of our question. Two key points that really got us thinking were the mention that crimes were on the rise and the mention that a majority of the hate crimes committed occurred in San Diego city. This article was written in 2018 and they mention that the number of hate crimes in 2017 were 11 and the number of registered hate crimes at the time the article was written was already at 14. These two points were eye opening and therefore helped us formulate our question.
3. https://www.nbcsandiego.com/news/local/hate-crime-charges-filed-against-man-who-punched-afghani-man-on-trolley/2279666/ This article is just an example of us that explains why this project is so important. This is exactly what we are trying to avoid. Hate-Crimes need to be a thing of the past. This article served to be more of a motivator that made us realise what we were doing was on the right path. Helping the city of San Diego identify potential problem areas is us doing our part to ensure these zones don't become a problem.
4. https://www.sandag.org/uploads/publicationid/publicationid_2105_21794.pdf Last but not the least is this publication that details the crimes motivated by race andhence can be classified as hate crimes.

- While we did not have previous research to guide our path, after reading these articeles, we decided to come up with the questions we have mentioned in the Why Hatecrime section? This helped us understnd how to deal with our data in a better aspect.

4 Imports and description of libraries used

[1]:
```
import pandas as pd # To deal with dataframe objects
import geopandas as gpd # Geopandas to work with latitude and longitude data
from arcgis.geocoding import get_geocoders, batch_geocode, geocode #In order to␣
↪convert string address to a coordinate
```

[2]:
```
import arcgis # to use ArcGis features such as feature Layers etc
from arcgis.geoenrichment import standard_geography_query,enrich #to enrich␣
↪zipcodes with demographic data for analysis
from arcgis.gis import GIS
from arcgis import geometry #to classify points as seen below
from arcgis.features import GeoAccessor, GeoSeriesAccessor #We imported these␣
↪packages to try and use these features throughout

gis = GIS() # for anonymous access
# or
gis = GIS(username='pjuneja_ucsd5')
arcgis.__version__
```

```
Enter password: ········
```

[2]:
```
'1.6.0'
```

[3]:
```
#more libraries used below and detailed there
```

5 Data Sources

- One of our major issues throughout our project which we will keep coming back to refer to as well, is that we only had one csv file with 160 records to go off. It did not have any spatial data. We immediately realised that this is because it takes a lot of bravery to talk about a hate crime experience. This was also Hate Crime data that was reported in 2015. This is something we have had to put aside to come up with a suitable analysis.
- Another one of our data sources include crime from the last 6 months. Our main assumption is that "Hate" crimes and regular crimes are two separate categories. Hate crimes are more of a mentality issue vs normal crime which doesnt look at any of these factors. However without making assumptions further we will see what our data says

The biggest issue with our project, other than the length of the dataset was that, there was no ready"spatial data." We had a super unlcean dataset with a few components of an address to go off of. Therefore this data involved a lot of cleaning steps from the get go. We had to figure out how to use this data.

- More recent cases of hate crime data would have been better but we ciulnd not find any sources for San Diego. A way of taking this project forward in my opinion and something I will be looking forward to pursuing on my own time, is if I find a larger hatecrime dataset, to train a model to learn on hatecrimes data and use this on the bigger dataset to identify which crimes could possibly be classified as hatecrimes as many hatecrimes do get reported as regular crimes to extract a better dataset for analysis

6 Data Cleaning

[4]:
```
hate_crime = pd.read_csv('hate_crimes_datasd.csv')
```

Steps involved in cleaning out data - After looking at the data below, we realised if we had to clean the data as mentioned above. Based on our metadata file, we realised block is code for areas. For Example, block 500 stands for Gaslamp. So one idea would be to replace the numerical value with string values. - Since we also wanted to get an idea of what time hate crimes usually occur, we thought it best to clean the time column, by first converting everyting to 24 hour and then creating three categories of 8 huors each(we did'nt make more cause of less number of reports) to try and identify a period where hatecrimes occur. - We also realised majority of our analysis just lied in the first half of our table for our initial analysis so we decided to segregate the table - We also had two nan which we dropped eventually - We then combined the address related fields to begin geocoding

[5]:
```
hate_crime.head(10)
```

[5]:
```
   case_number        date  year  month         time            date_time  \
0     16000456  2016-01-04  2016      1   2:00:00 AM  2016-01-04 02:00:00
1     16001278  2016-01-10  2016      1   1:30:00 AM  2016-01-10 01:30:00
2     16004522  2016-01-31  2016      1     02:30:00  2016-01-31 02:30:00
3     16005962  2016-02-09  2016      2   4:30:00 PM  2016-02-09 16:30:00
4     16005900  2016-02-10  2016      2     00:45:00  2016-02-10 00:45:00
5     16006866  2016-02-16  2016      2   4:30:00 PM  2016-02-16 16:30:00
6     16007286  2016-02-18  2016      2     20:43:00  2016-02-18 20:43:00
7     16008581  2016-02-26  2016      2  11:00:00 PM  2016-02-26 23:00:00
8     16008340  2016-02-26  2016      2   5:25:00 PM  2016-02-26 17:25:00
9     16008751  2016-02-29  2016      2     04:00:00  2016-02-29 04:00:00

     crime_code                crime   block      street  … suspect_race_2  \
0       243(D)M   Assault, No Weapon   500.0           G  …              O
1         245A1    Assault, w/Weapon  3400.0        30th  …            Unk
2     594(B)(4)            Vandalism  1400.0    Imperial  …            NaN
3  422.22(a)(4)   Threat, Phone Call  4100.0         Ute  …            NaN
4       417A1:M               Threat   100.0  University  …            NaN
5         422.6            Vandalism  5800.0  University  …            NaN
6         422.6               Threat  6200.0       Capri  …            NaN
7         422.6            Vandalism  5400.0     Gilbert  …            NaN
8        422.6A   Assault, No Weapon   500.0      Euclid  …            NaN
9       417A1:M               Threat   300.0        Park  …            NaN

  suspect_sex_0 suspect_sex_1 suspect_sex_2 victim_race_0 victim_race_1  \
0             M             M             M             O           NaN
```

```
1            M          M          M          B          NaN
2          Unk        NaN        NaN          H            W
3            M        NaN        NaN          B          NaN
4            M        NaN        NaN          B            A
5          Unk        NaN        NaN          B          NaN
6            F        NaN        NaN          W          NaN
7          Unk        NaN        NaN          I          NaN
8            M        NaN        NaN          H          NaN
9            M        NaN        NaN          B          NaN

  victim_race_2  victim_sex_0 victim_sex_1 victim_sex_2
0           NaN            M          NaN          NaN
1           NaN            M          NaN          NaN
2           NaN            F            M          NaN
3           NaN            M          NaN          NaN
4           NaN            M            F          NaN
5           NaN            M          NaN          NaN
6           NaN            M          NaN          NaN
7           NaN            M          NaN          NaN
8           NaN            F          NaN          NaN
9           NaN            M          NaN          NaN

[10 rows x 32 columns]
```

6.1 Basic EDA on data

[6]:
```
hate_crime.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 160 entries, 0 to 159
Data columns (total 32 columns):
case_number          160 non-null int64
date                 160 non-null object
year                 160 non-null int64
month                160 non-null int64
time                 160 non-null object
date_time            160 non-null object
crime_code           160 non-null object
crime                160 non-null object
block                159 non-null float64
street               160 non-null object
type                 152 non-null object
beat                 160 non-null int64
command              160 non-null object
weapon               160 non-null object
motivation           160 non-null object
```

```
number_of_suspects    160 non-null object
suspect               160 non-null object
victim_count          160 non-null int64
victim_other          39 non-null object
injury                160 non-null object
suspect_race_0        158 non-null object
suspect_race_1        25 non-null object
suspect_race_2        13 non-null object
suspect_sex_0         157 non-null object
suspect_sex_1         25 non-null object
suspect_sex_2         13 non-null object
victim_race_0         135 non-null object
victim_race_1         24 non-null object
victim_race_2         3 non-null object
victim_sex_0          136 non-null object
victim_sex_1          24 non-null object
victim_sex_2          3 non-null object
dtypes: float64(1), int64(5), object(26)
memory usage: 40.1+ KB
```

[7]:
```
hate_crime['number_of_suspects']
```

[7]:
```
0        3
1        3
2      Unk
3        1
4        1
      …
155      2
156    Unk
157      1
158    Unk
159    Unk
Name: number_of_suspects, Length: 160, dtype: object
```

6.1.1 Converting beats to string values

[8]:
```
#SDPD beat codes
dic= hate_crime['beat'].to_dict()
```

[9]:
```
#hate_crime[hate_crime['suspect']=='Unknown']#.value_c
```

This is a dataset of all codes to locations

[10]:
```
sd_pd = pd.read_csv('pd_beat_neighborhoods_datasd.csv')
```

[ ]:

```

```

[11]:
```
beats = sd_pd.set_index('Beat').to_dict()['Neighborhood']
```

[12]:
```
hate_crime['beat'] = hate_crime['beat'].map(beats)
```

[13]:
```
hate_crime['crime_code']
```

[13]:
```
0          243(D)M
1            245A1
2        594(B)(4)
3     422.22(a)(4)
4          417A1:M
          …
155            242
156            594
157            242
158            594
159           422A
Name: crime_code, Length: 160, dtype: object
```

[14]:
```
#Last 6 months crime data
crime_data = pd.read_csv('ARJISPublicCrime030420.csv')#,sep = "␣
↪",error_bad_lines=False)
```

[15]:
```
crime_data.head(10)
```

[15]:
```
                  CM_LEGEND       agency  \
0             THEFT/LARCENY    OCEANSIDE
1             THEFT/LARCENY  CHULA VISTA
2                       DUI    SAN DIEGO
3       MOTOR VEHICLE THEFT  CHULA VISTA
4  DRUGS/ALCOHOL VIOLATIONS    ESCONDIDO
5                   WEAPONS    ESCONDIDO
6                       DUI    ESCONDIDO
7                  BURGLARY     EL CAJON
8                       DUI      SHERIFF
9                     FRAUD    SAN DIEGO

                             Charge_Description_Orig         activityDate  \
0                 PETTY THEFT(All Other Larceny) (M)  12/29/2019 18:11:00
1                          PETTY THEFT(Shoplift) (M)    1/2/2020 20:15:00
2                           DUI ALC/0.08 PERCENT (M)     1/1/2020 2:36:00
3  TAKE VEHICLE W/O OWNER'S CONSENT/VEHICLE THEFT…     1/6/2020 17:00:00
4                   POSSESS CONTROLLED SUBSTANCE (M)  12/28/2019 20:00:00
5    MANUFACTURE/SALE/POSSESS/ETC METAL KNUCKLES (F)  12/28/2019 20:00:00
6                                   DUI ALCOHOL (M)   12/29/2019 0:21:00
```

```
7                         BURGLARY (RESIDENTIAL) (F)    1/2/2020 15:00:00
8                                    DUI ALCOHOL (M)   10/13/2019 2:50:00
9  THEFT BY USE OF ACCESS CARD INFORMATION [OVER …  12/21/2019 14:45:00

                          BLOCK_ADDRESS  ZipCode    community
0          1800  BLOCK COLLEGE BOULEVARD  92056.0    OCEANSIDE
1              600  BLOCK PALOMAR STREET  91911.0  CHULA VISTA
2                4000  BLOCK RUEDA DRIVE  92124.0    SAN DIEGO
3             1200  BLOCK ATWATER STREET  91913.0  CHULA VISTA
4         500 W  BLOCK WASHINGTON AVENUE  92025.0    ESCONDIDO
5         500 W  BLOCK WASHINGTON AVENUE  92025.0    ESCONDIDO
6  LAKE WOHLFORD ROAD & E VALLEY PARKWAY  92027.0    ESCONDIDO
7             1300  BLOCK NARANCA AVENUE  92021.0     EL CAJON
8          6700  BLOCK SAN MIGUEL AVENUE  91945.0  LEMON GROVE
9              9800  BLOCK HIBERT STREET  92131.0    SAN DIEGO
```

[16]:
```
#for hatecrimes relating to threats, not super accurate atm
threat = crime_data[crime_data['Charge_Description_Orig'].str.
 ↪contains('THREAT')]
#crime_data[crime_data['CM_LEGEND'].str.contains('abuse')]
```

[17]:
```
#Create two Dataframes for ease of access depending on above data keeping case␣
 ↪ID in both as a primary key
```

[18]:
```
#
hate_info=␣
 ↪hate_crime[['case_number','time','crime_code','crime','beat','block','street','type','weapo
hate_info = pd.concat([hate_info[col].astype(str).str.lower() for col in␣
 ↪hate_info.columns], axis=1)
hate_info.head()
```

[18]:
```
  case_number        time    crime_code               crime          beat  \
0    16000456  2:00:00 am       243(d)m  assault, no weapon       gaslamp
1    16001278  1:30:00 am         245a1   assault, w/weapon    north park
2    16004522    02:30:00     594(b)(4)            vandalism  east village
3    16005962  4:30:00 pm  422.22(a)(4)  threat, phone call      bay park
4    16005900    00:45:00       417a1:m              threat     hillcrest

    block      street type              weapon
0   500.0           g   st  hands, fists, feet
1  3400.0        30th   st               stick
2  1400.0    imperial   av              marker
3  4100.0         ute   dr               phone
4   100.0  university   av               knife
```

[19]:
```
hate_info['time']=pd.to_datetime(hate_info['time']).dt.strftime('%H:%M:%S')
```

6.2 Function to clean Time

[20]:
```python
#Cleaning time for better analysis
def conv_time(time):
    if time >'00:00:00' and time<='08:00:00':
        return 'Early Morning/Late Night'
    elif time>'08:00:00'and time <= '16:00:00':
        return 'Day'
    #elif time>'16:00:00'and time<='24:00:00':
    else:
        return 'Evening and Night'
    #elif time>'16:00:00'and time<='20:00:00':
        #return 'Evening'
#     elif time>'20:00:00'and time<='23:59:59':

hate_info['time'] = hate_info['time'].apply(conv_time)
```

[21]:
```python
# mention why we used 8 hour windows
hate_info.head()
```

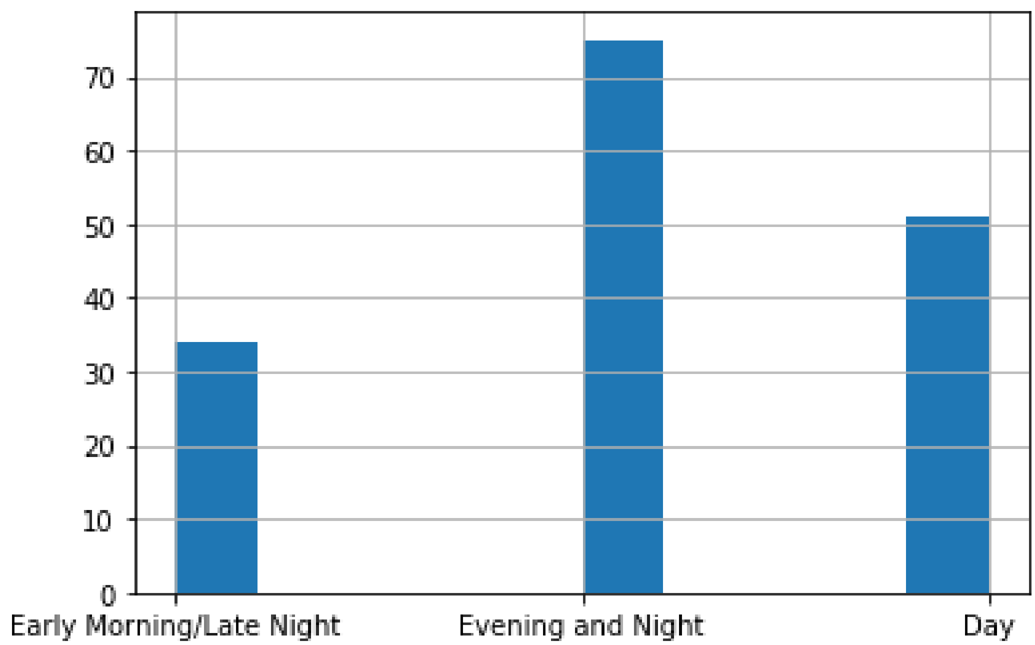
[21]:
```
  case_number                      time    crime_code              crime  \
0    16000456  Early Morning/Late Night       243(d)m  assault, no weapon
1    16001278  Early Morning/Late Night        245a1   assault, w/weapon
2    16004522  Early Morning/Late Night     594(b)(4)           vandalism
3    16005962         Evening and Night  422.22(a)(4)  threat, phone call
4    16005900  Early Morning/Late Night      417a1:m              threat

           beat   block      street type              weapon
0       gaslamp   500.0           g   st  hands, fists, feet
1    north park  3400.0        30th   st               stick
2  east village  1400.0    imperial   av              marker
3      bay park  4100.0         ute   dr               phone
4     hillcrest   100.0  university   av               knife
```

7 Statistics and Initial Visualization

Since we now had a pandas dataframe which we had cleaned, we decided to do some preliminary analysis and figure out certain stats that may help us later on in the investigation. Below is a graph of the number of crimes occured in which time period. This is followed by an analysis of trying to figure out what sorts of attacks were usually done through hate crime. More figures and stats are included after where we do a suspect analysis.

[22]:
```python
hate_info['time'].hist()
```

[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc3696198d0>
```



We now want to see if the time of day actually affects the kind of hate crime commited. Below are some statistics that apart from telling us, we need to clean assault, are kind of something we would have expected. Late night crime includes a lot of vandalism and assaults. This might be because people inherently feel they can get away with more dangerous crimes at night.

[23]:
```
a= hate_info[hate_info['time']=='Early Morning/Late Night']
a['crime'].value_counts()
```

[23]:
```
vandalism              13
threat                  6
assault, w/weapon       4
assault, no weapon      3
assault                 2
other                   2
threat, phone call      2
battery                 2
Name: crime, dtype: int64
```

[24]:
```
a= hate_info[hate_info['time']=='Day']
a['crime'].value_counts()
```

[24]:
```
vandalism              18
assault                 9
threat                  6
other                   5
assault, no weapon      4
```

```
assault, w/weapon      4
robbery                2
threat, phone call     1
bomb threat            1
burglary               1
Name: crime, dtype: int64
```

[25]:
```
a= hate_info[hate_info['time']=='Morning']
a['crime'].value_counts()
```

[25]:
```
Series([], Name: crime, dtype: int64)
```

[26]:
```
# cleaning crime
```

[27]:
```
hate_info['crime'] = hate_info['crime'].str.lower()
#contains('Assault')]
```

7.0.1 Cleaning assault column because assault is assault as a whole

[28]:
```
def conv_assault(ass):
    if 'assault' in ass:
        return 'assault'
    else:
        return ass
hate_info['crime'] = hate_info['crime'].apply(conv_assault)
```

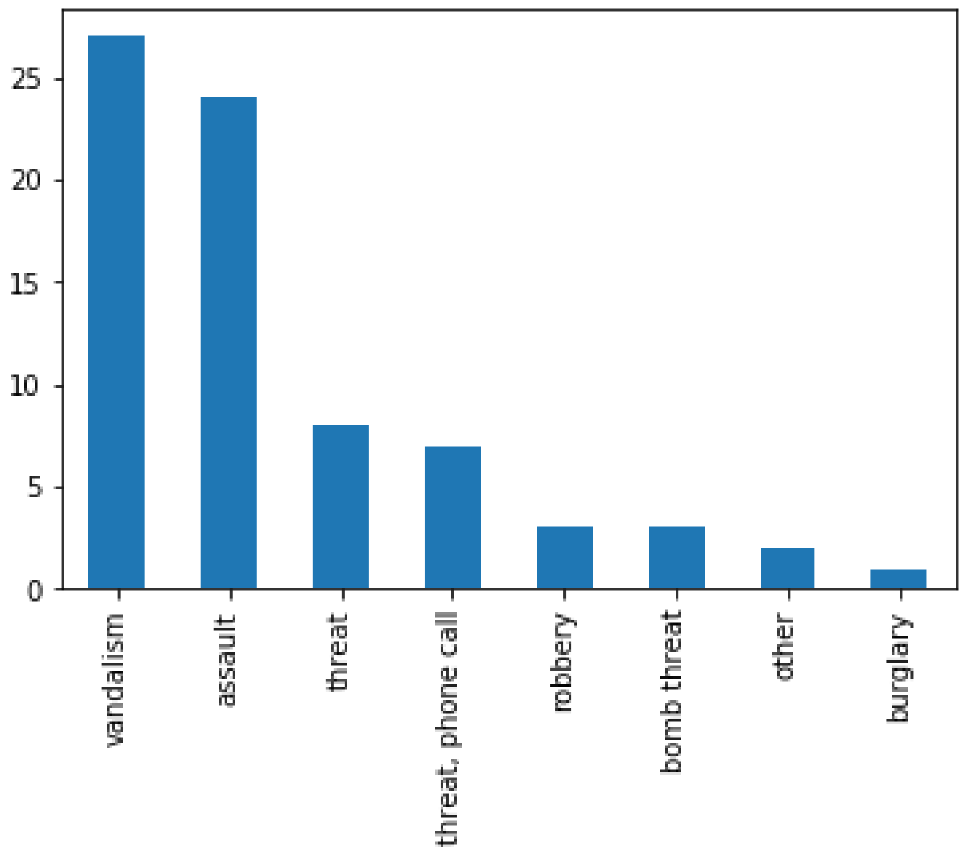
8 Visualizing Type of Crimes by day

[29]:
```
#Interesting observation
even= hate_info[hate_info['time']=='Evening and Night']
even['crime'].value_counts()
```

[29]:
```
vandalism             27
assault               24
threat                 8
threat, phone call     7
robbery                3
bomb threat            3
other                  2
burglary               1
Name: crime, dtype: int64
```

[30]:
```
even['crime'].value_counts().plot.bar()
```

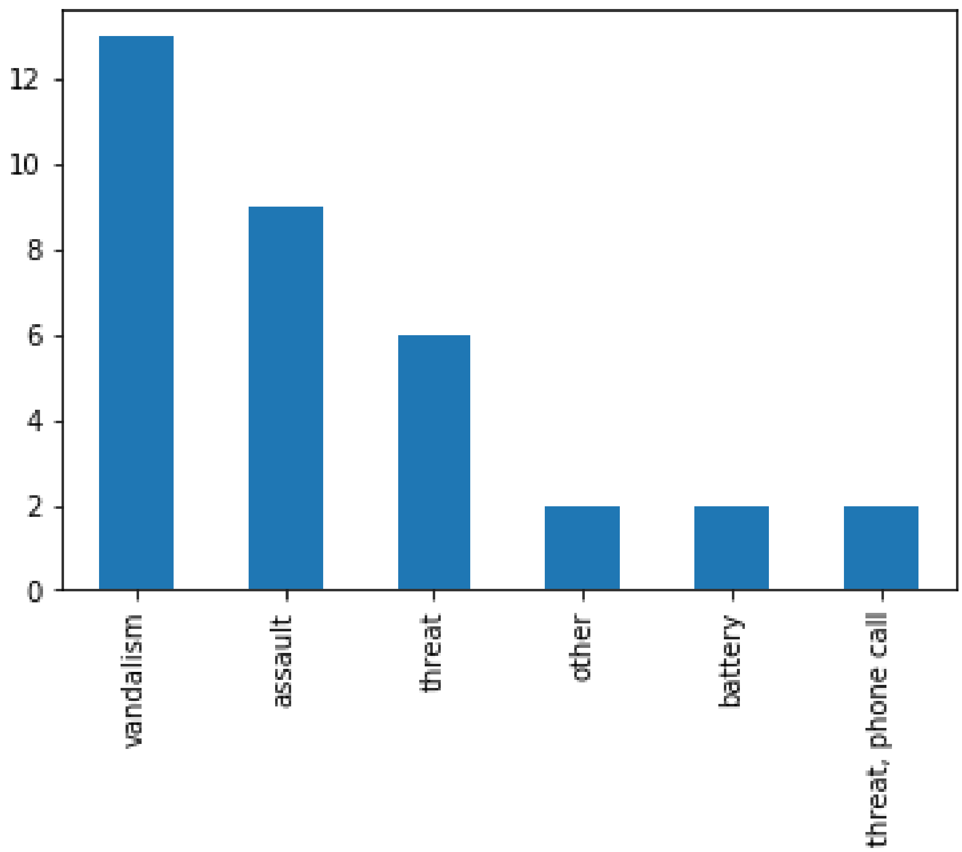
[30]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc3694b5710>



[ ]:
```

```

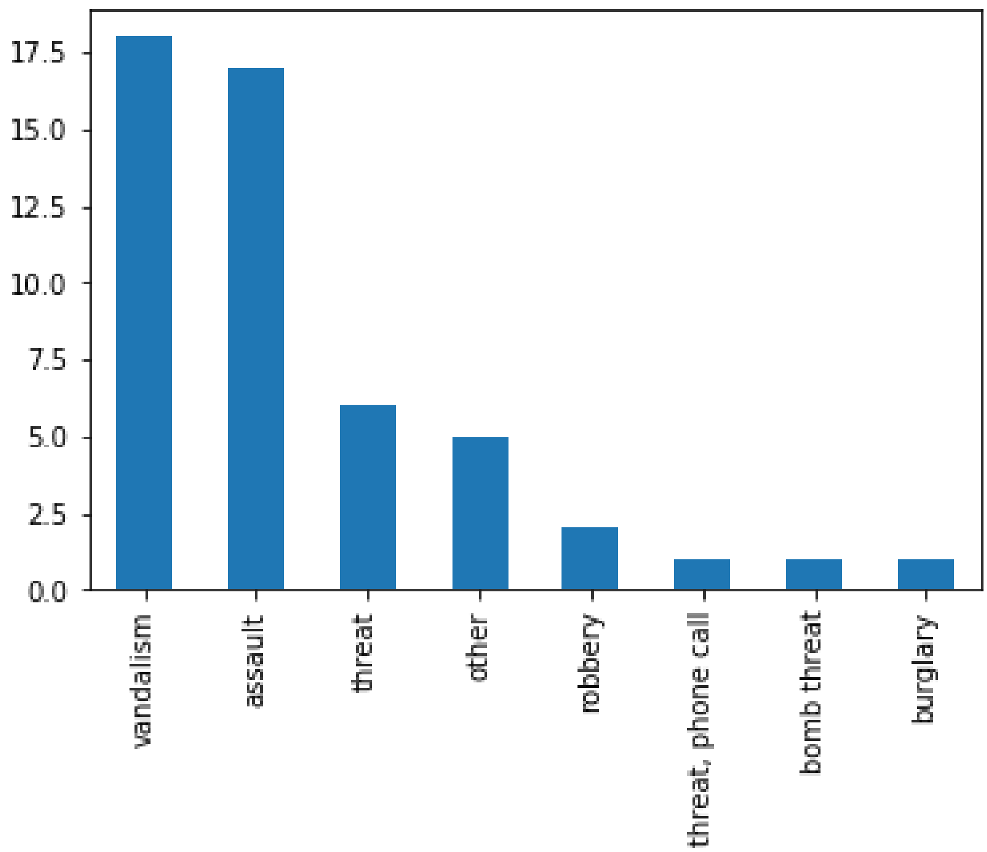
[31]:
```
early= hate_info[hate_info['time']=='Early Morning/Late Night']
early['weapon'].value_counts()
early['crime'].value_counts().plot.bar()
```

[31]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc369454d68>



[32]:
```
day= hate_info[hate_info['time']=='Day']
day['weapon'].value_counts()
day['crime'].value_counts().plot.bar()
```

[32]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc3693c4048>



[33]:
```
day['crime'].value_counts()
```

[33]:
```
vandalism             18
assault               17
threat                 6
other                  5
robbery                2
threat, phone call     1
bomb threat            1
burglary               1
Name: crime, dtype: int64
```

9 Analysis

After getting an idea of how the data behaves and figuring out majority of how the data works, we still do not where these events occured on a map of San Diego. As we know and have studied in class, one very important technique that takes into account an address and returns a coordinate is known as Geocoding. This converts our address to a coordinate we can plot and see on a map.

9.0.1 Blueprint for analysis

The steps involved in this project are 1. Cleaning address to use for GeoCode 2. Coverting Lat and longitudes to Points on a map for visualization purposes 3. However just knowing the coordinates doesn't help in analysis as much 4. After Geocoding use Join Features with a Zip Code Feature layer to obtain a Zipcode value for each hate crime incident 5. Since a project must include shortcomings , we tried using census tracts data to find the average income of these neighbourhoods, but due to non intersecting boundaries between them and zip codes, we realised the best option to use was GeoEnrichment 6. GeoEnriching zipcode with the census and demographic information to tell us about the areas these crimes occured in 7. Finding out where these areas mostly occur is an important step to suggesting improvements that could be made for this 8. Based on our hypothesis, we will look at base race,(most dominant race of the region) average income and number of hatecrimes occuring 9. Our initial assesment is that areas with a higher diversity(ie lower base race) would have a lower level of hatecrime and areas with higher income would too have a lower level of hate crime 10. Created a density map to find high correlation areas and created scatter plots to test our hypothesis 11. We then correlated our analysis with online reports of these crimes and zipcodes to see if we were in the right direction 12. Looking at last 6 month crime data and seeing if its related to hatecrime

[34]:
```
hate_info.head(14)
```

[34]:
```
    case_number                     time    crime_code              crime  \
0      16000456  Early Morning/Late Night      243(d)m            assault
1      16001278  Early Morning/Late Night        245a1            assault
2      16004522  Early Morning/Late Night     594(b)(4)          vandalism
3      16005962           Evening and Night  422.22(a)(4)  threat, phone call
4      16005900  Early Morning/Late Night      417a1:m             threat
5      16006866           Evening and Night        422.6          vandalism
6      16007286           Evening and Night        422.6             threat
7      16008581           Evening and Night        422.6          vandalism
8      16008340           Evening and Night       422.6a            assault
9      16008751  Early Morning/Late Night      417a1:m             threat
10     16009905           Evening and Night        422.6              other
11     16013490           Evening and Night       422.6a  threat, phone call
12     16015895           Evening and Night      653m(a)  threat, phone call
13     16017576           Evening and Night       422.6b          vandalism

                 beat    block                  street type  \
0             gaslamp    500.0                       g   st
1          north park   3400.0                    30th   st
2        east village   1400.0                imperial   av
3            bay park   4100.0                     ute   dr
4           hillcrest    100.0              university   av
5          el cerrito   5800.0              university   av
6           del cerro   6200.0                   capri   dr
7        college west   5400.0                 gilbert   dr
8        lincoln park    500.0                  euclid   av
```

```
9          east village    300.0                      park   bl
10        logan heights   2200.0                  imperial   av
11        carmel valley  12800.0             via nieve #74  nan
12        pacific beach   1600.0                thomas ave   av
13  mission valley east    500.0  camino de la reina #129  nan

                 weapon
0     hands, fists, feet
1                  stick
2                 marker
3                  phone
4                  knife
5                  paint
6                  phone
7   unknown sharp object
8     hands, fists, feet
9                  knife
10                 knife
11                 phone
12                 phone
13           spray paint
```

[35]:
```
#onto geocoding
hate_info = hate_info.dropna(subset = ['block'])
len(hate_info)
```

[35]:
```
160
```

[36]:
```
#We still had to clean a lit bt of the block column here
```

[37]:
```
hate_info['block'] = hate_info['block'].apply(lambda x: x[:-2])
```

[38]:
```
hate_info.head(10)
```

[38]:
```
  case_number                    time   crime_code               crime  \
0    16000456  Early Morning/Late Night     243(d)m             assault
1    16001278  Early Morning/Late Night       245a1             assault
2    16004522  Early Morning/Late Night   594(b)(4)           vandalism
3    16005962         Evening and Night 422.22(a)(4)  threat, phone call
4    16005900  Early Morning/Late Night     417a1:m              threat
5    16006866         Evening and Night       422.6           vandalism
6    16007286         Evening and Night       422.6              threat
7    16008581         Evening and Night       422.6           vandalism
8    16008340         Evening and Night      422.6a             assault
9    16008751  Early Morning/Late Night     417a1:m              threat

           beat block      street type                weapon
```

```
0       gaslamp   500           g   st    hands, fists, feet
1    north park  3400        30th   st                 stick
2  east village  1400    imperial   av                marker
3      bay park  4100         ute   dr                 phone
4     hillcrest   100  university   av                 knife
5    el cerrito  5800  university   av                 paint
6     del cerro  6200       capri   dr                 phone
7  college west  5400     gilbert   dr  unknown sharp object
8  lincoln park   500      euclid   av    hands, fists, feet
9  east village   300        park   bl                 knife
```

[39]:
```
#adding all address fields to one column in order to geocode with a high␣
↪accuracy
hate_info['address'] = hate_info['block'] + " " + hate_info['street'] + " " +␣
↪hate_info['type']+" "+ 'San Diego, CA'
```

[40]:
```
hate_info.head()
# We can now see the address column
```

[40]:
```
  case_number                        time    crime_code               crime  \
0    16000456  Early Morning/Late Night       243(d)m             assault
1    16001278  Early Morning/Late Night        245a1             assault
2    16004522  Early Morning/Late Night     594(b)(4)           vandalism
3    16005962           Evening and Night  422.22(a)(4)  threat, phone call
4    16005900  Early Morning/Late Night       417a1:m              threat

           beat block       street type              weapon  \
0       gaslamp   500           g   st  hands, fists, feet
1    north park  3400        30th   st               stick
2  east village  1400    imperial   av              marker
3      bay park  4100         ute   dr               phone
4     hillcrest   100  university   av               knife

                         address
0          500 g st San Diego, CA
1       3400 30th st San Diego, CA
2    1400 imperial av San Diego, CA
3        4100 ute dr San Diego, CA
4  100 university av San Diego, CA
```

10 GeoCoding

[41]:
```
addresses = list(hate_info['address'])
```

[42]:
```
from arcgis.geocoding import get_geocoders, batch_geocode, geocode
```

[43]:
```
geocoder = get_geocoders(gis)[0]
```

[44]:
```
results = batch_geocode(addresses)
```

[45]:
```
map1 = gis.map("San Diego County, US")
map1
```

```
MapView(layout=Layout(height='400px', width='100%'))
```

```
<IPython.core.display.HTML object>
```

[46]:
```
for address in results:
    map1.draw(address['location'])
```

[47]:
```
hate_info.head()
```

[47]:
```
   case_number                        time     crime_code               crime  \
0     16000456  Early Morning/Late Night       243(d)m             assault
1     16001278  Early Morning/Late Night         245a1             assault
2     16004522  Early Morning/Late Night      594(b)(4)           vandalism
3     16005962           Evening and Night  422.22(a)(4)  threat, phone call
4     16005900  Early Morning/Late Night        417a1:m              threat

            beat block       street type              weapon  \
0        gaslamp   500            g   st  hands, fists, feet
1     north park  3400         30th   st               stick
2   east village  1400     imperial   av              marker
3       bay park  4100          ute   dr               phone
4      hillcrest   100   university   av               knife

                               address
0           500 g st San Diego, CA
1        3400 30th st San Diego, CA
2    1400 imperial av San Diego, CA
3         4100 ute dr San Diego, CA
4   100 university av San Diego, CA
```

[48]:
```
#creating lists for lat and long and adding them to a pd.series to add to the␣
↪dataframe as seen below
latitudes = []
longitudes = []
for address in results:
    geocoded = geocode(address)
    longitude = geocoded[0]['attributes']['X']
    latitude = geocoded[0]['attributes']['Y']
```

```python
    latitudes = latitudes + [latitude]
    longitudes = longitudes + [longitude]

hate_info['latitude'] = pd.Series(latitudes)
hate_info['longitude'] = pd.Series(longitudes)
```

[49]:
```python
hate_info.head(10)
```

[49]:
```
   case_number                       time    crime_code                 crime  \
0     16000456  Early Morning/Late Night       243(d)m               assault
1     16001278  Early Morning/Late Night        245a1               assault
2     16004522  Early Morning/Late Night     594(b)(4)             vandalism
3     16005962           Evening and Night  422.22(a)(4)  threat, phone call
4     16005900  Early Morning/Late Night       417a1:m                threat
5     16006866           Evening and Night        422.6             vandalism
6     16007286           Evening and Night        422.6                threat
7     16008581           Evening and Night        422.6             vandalism
8     16008340           Evening and Night       422.6a               assault
9     16008751  Early Morning/Late Night       417a1:m                threat

           beat block      street type                weapon  \
0       gaslamp   500           g   st    hands, fists, feet
1    north park  3400        30th   st                 stick
2  east village  1400    imperial   av                marker
3      bay park  4100         ute   dr                 phone
4     hillcrest   100  university   av                 knife
5    el cerrito  5800  university   av                 paint
6     del cerro  6200       capri   dr                 phone
7  college west  5400     gilbert   dr  unknown sharp object
8  lincoln park   500      euclid   av    hands, fists, feet
9  east village   300        park   bl                 knife

                          address   latitude   longitude
0          500 g st San Diego, CA  32.712638 -117.160073
1       3400 30th st San Diego, CA  32.741139 -117.130148
2   1400 imperial av San Diego, CA  32.706347 -117.151812
3        4100 ute dr San Diego, CA  32.807511 -117.203142
4   100 university av San Diego, CA  32.748341 -117.163831
5  5800 university av San Diego, CA  32.749225 -117.072844
6      6200 capri dr San Diego, CA  32.782135 -117.065392
7    5400 gilbert dr San Diego, CA  32.760023 -117.078470
8      500 euclid av San Diego, CA  32.710521 -117.085090
9        300 park bl San Diego, CA  32.708696 -117.153825
```

[50]:
```python
latitude
```

[50]: 32.71890748603413

11 We now have Lat and Long added to the dataframe. But since they are pandas objects they cannot be used. We import the point constructor from shapely to convert these to a usable co-ordiante

[51]:
```python
from shapely.geometry import Point
df = hate_info.copy()
df.head()
```

[51]:
```
  case_number                        time    crime_code                crime  \
0    16000456  Early Morning/Late Night        243(d)m              assault
1    16001278  Early Morning/Late Night          245a1              assault
2    16004522  Early Morning/Late Night       594(b)(4)           vandalism
3    16005962           Evening and Night  422.22(a)(4)  threat, phone call
4    16005900  Early Morning/Late Night        417a1:m               threat

           beat block       street type                weapon  \
0       gaslamp   500            g   st  hands, fists, feet
1    north park  3400         30th   st                 stick
2  east village  1400     imperial   av                marker
3      bay park  4100          ute   dr                 phone
4     hillcrest   100   university   av                 knife

                              address   latitude   longitude
0           500 g st San Diego, CA  32.712638 -117.160073
1        3400 30th st San Diego, CA  32.741139 -117.130148
2     1400 imperial av San Diego, CA  32.706347 -117.151812
3         4100 ute dr San Diego, CA  32.807511 -117.203142
4   100 university av San Diego, CA  32.748341 -117.163831
```

[52]:
```python
df['Coordinates'] = list(zip(df.longitude, df.latitude))
df['Coordinates'] = df['Coordinates'].apply(Point)
df.head()
```

[52]:
```
  case_number                        time    crime_code                crime  \
0    16000456  Early Morning/Late Night        243(d)m              assault
1    16001278  Early Morning/Late Night          245a1              assault
2    16004522  Early Morning/Late Night       594(b)(4)           vandalism
3    16005962           Evening and Night  422.22(a)(4)  threat, phone call
4    16005900  Early Morning/Late Night        417a1:m               threat

           beat block       street type                weapon  \
0       gaslamp   500            g   st  hands, fists, feet
1    north park  3400         30th   st                 stick
2  east village  1400     imperial   av                marker
3      bay park  4100          ute   dr                 phone
```

```
4     hillcrest   100  university   av              knife

                              address   latitude   longitude  \
0           500 g st San Diego, CA  32.712638 -117.160073
1       3400 30th st San Diego, CA  32.741139 -117.130148
2     1400 imperial av San Diego, CA  32.706347 -117.151812
3        4100 ute dr San Diego, CA  32.807511 -117.203142
4  100 university av San Diego, CA  32.748341 -117.163831

                                     Coordinates
0  POINT (-117.1600734036998 32.71263809630025)
1  POINT (-117.1301477956986 32.74113929569858)
2  POINT (-117.1518118106902 32.70634749036676)
3  POINT (-117.2031424172799 32.80751149470067)
4  POINT (-117.1638308764208 32.74834112357916)
```

[53]:
```python
hate_info_geo = gpd.GeoDataFrame(df, geometry='Coordinates')
# hate_info_geo.plot(figsize=(10,10), legend = True)
# #new_locations.plot(ax=new_map, legend = True,marker='o', color='black',␣
↪markersize=100)
```

12 Based on what we just did above, we now have a geodataframe detailing exactly where each hatecrime was reported. This was a big step for us because we had no such information prior to this

[54]:
```python
hate_info_geo
```

[54]:
```
    case_number                   time    crime_code               crime  \
0      16000456  Early Morning/Late Night      243(d)m             assault
1      16001278  Early Morning/Late Night       245a1             assault
2      16004522  Early Morning/Late Night    594(b)(4)           vandalism
3      16005962         Evening and Night  422.22(a)(4)  threat, phone call
4      16005900  Early Morning/Late Night      417a1:m              threat
..          …                        …          …                  …
155    19048808  Early Morning/Late Night          242             battery
156    19049530  Early Morning/Late Night          594           vandalism
157    19052150                       Day          242             assault
158    19055750                       Day          594           vandalism
159    19058413         Evening and Night         422a              threat

                 beat  block          street type              weapon  \
0             gaslamp    500               g   st  hands, fists, feet
1          north park   3400            30th   st               stick
```

```
2         east village   1400         imperial   av              marker
3              bay park   4100              ute   dr               phone
4             hillcrest    100       university   av               knife
..                  …    …                 …   …               …
155  point loma heights   2400          seaside    st  hands, fists, feet
156  point loma heights   4100  west point loma   bl           black ink
157            encanto   6600         broadway  nan  hands, fists, feet
158  rancho penasquitos  13000     salmon river   rd              marker
159        core-columbia   1300              4th   av               phone

                             address   latitude   longitude  \
0                 500 g st San Diego, CA  32.712638 -117.160073
1             3400 30th st San Diego, CA  32.741139 -117.130148
2         1400 imperial av San Diego, CA  32.706347 -117.151812
3              4100 ute dr San Diego, CA  32.807511 -117.203142
4       100 university av San Diego, CA  32.748341 -117.163831
..                                   …        …          …
155        2400 seaside  st San Diego, CA  32.750025 -117.237285
156  4100 west point loma bl San Diego, CA  32.753804 -117.223555
157       6600 broadway nan San Diego, CA  32.715941 -117.055921
158    13000 salmon river rd San Diego, CA  32.953790 -117.120781
159            1300 4th  av San Diego, CA  32.718907 -117.161159

                 Coordinates
0    POINT (-117.16007 32.71264)
1    POINT (-117.13015 32.74114)
2    POINT (-117.15181 32.70635)
3    POINT (-117.20314 32.80751)
4    POINT (-117.16383 32.74834)
..                          …
155  POINT (-117.23728 32.75002)
156  POINT (-117.22356 32.75380)
157  POINT (-117.05592 32.71594)
158  POINT (-117.12078 32.95379)
159  POINT (-117.16116 32.71891)

[160 rows x 13 columns]
```

[55]:
```python
#Converting the pd dataframe to sdf for more spatial analysis
hate_info_sdf = pd.DataFrame.spatial.from_xy(hate_info,x_column = 'longitude',
 ↪y_column = 'latitude')
```

[56]:
```python
hate_info_sdf.head()
```

```
/opt/conda/lib/python3.7/site-packages/IPython/lib/pretty.py:399: FutureWarning:
'ExtensionArray._formatting_values' is deprecated. Specify
'ExtensionArray._formatter' instead.
```

```
  return _repr_pprint(obj, self, cycle)
/opt/conda/lib/python3.7/site-packages/pandas/io/formats/html.py:606:
FutureWarning: 'ExtensionArray._formatting_values' is deprecated. Specify
'ExtensionArray._formatter' instead.
  super().render()
```

[56]:
```
   case_number                     time    crime_code               crime  \
0     16000456  Early Morning/Late Night       243(d)m             assault
1     16001278  Early Morning/Late Night         245a1             assault
2     16004522  Early Morning/Late Night     594(b)(4)           vandalism
3     16005962         Evening and Night  422.22(a)(4)  threat, phone call
4     16005900  Early Morning/Late Night       417a1:m              threat

           beat block      street type             weapon  \
0       gaslamp   500           g   st  hands, fists, feet
1    north park  3400        30th   st               stick
2  east village  1400    imperial   av              marker
3      bay park  4100         ute   dr               phone
4     hillcrest   100  university   av               knife

                             address   latitude   longitude  \
0            500 g st San Diego, CA  32.712638 -117.160073
1         3400 30th st San Diego, CA  32.741139 -117.130148
2     1400 imperial av San Diego, CA  32.706347 -117.151812
3          4100 ute dr San Diego, CA  32.807511 -117.203142
4   100 university av San Diego, CA  32.748341 -117.163831

                                               SHAPE
0  {"x": -117.16007340369976, "y": 32.71263809630…
1  {"x": -117.13014779569856, "y": 32.74113929569…
2  {"x": -117.15181181069018, "y": 32.70634749036…
3  {"x": -117.20314241727988, "y": 32.80751149470…
4  {"x": -117.16383087642083, "y": 32.74834112357…
```

[57]:
```python
# hate_info_fl = hate_info_sdf.spatial.to_featurelayer(title='San Diego Hate␣
↪Crime',tags = 'hate crime')
# hate_info_fl.share(org=True)
```

[58]:
```python
hate_info_fl = gis.content.get('276d89103e164c84be12f07acdf36899')
```

[59]:
```python
hate_info_fl = hate_info_fl.layers[0]
```

[60]:
```python
hate_info_fl
```

[60]:
```
<FeatureLayer url:"https://services1.arcgis.com/eGSDp8lpKe5izqVc/arcgis/rest/ser
vices/a630f1/FeatureServer/0">
```

We used the above SDF to convert this to a feature layer that can be plotted

[61]:
```
map2 = gis.map('San Diego County, CA')
map2.add_layer(hate_info_fl)
```

Below is a map of all the coordinates of where the hatecrime occured. This information just tells us about where they occured but little information on the type of area it occured in. We don't know the zipcode or anything about the area to make assumptions of where this occured. So to do this, we need to create a feature layer of zipcodes, combine the two maps using within and obtain a list of all zipcodes where the accidents occured to understand the area of each hatecrime

[62]:
```
map2
```

```
MapView(layout=Layout(height='400px', width='100%'))
```

```
<IPython.core.display.HTML object>
```

Finding san diego zipcode layer based on gis results

[63]:
```
result = gis.content.search('San Diego Zip Codes' , item_type="Feature Layer",␣
↪outside_org=True, max_items=30)
result
from IPython.display import display
for item in result:
    display(item)
```

```
<Item title:"Julian_Minerals_JF" type:Feature Layer Collection owner:schil021>

<Item title:"Dog_and_Cat_Households_in_SD_Zips" type:Feature Layer Collection owner:Rizbee>

<Item title:"Drank Cola" type:Feature Layer Collection owner:kyle4920@esri.com_manucomm>

<Item title:"Enriched San Diego County Zip Codes Ness" type:Feature Layer Collection owner:meno

<Item title:"Enriched San Diego County Zip Codes kelvin lee" type:Feature Layer Collection owne

<Item title:"Find_Locations_in_SD_Mineral_Resources_2sk" type:Feature Layer Collection owner:k

<Item title:"Enriched San Diego County Zip Codes yuki" type:Feature Layer Collection owner:hata

<Item title:"San Diego County Zip Codes" type:Feature Layer Collection owner:samantha.wriker@e

<Item title:"SanDiegoRentalsPerZIP" type:Feature Layer Collection owner:CVaillancourt_EsriMedia
```

```
<Item title:"Enriched San Diego County Zip Codes Thehara Ambrose" type:Feature Layer Collectio

<Item title:"Zips_version2" type:Feature Layer Collection owner:readthemap_NU_Helath>

<Item title:"Aggregation_of_Get_it_done_report_2019_to_San_Diego_County_Zip_Codes_San_Diego_Co

<Item title:"enriched San Diego County Zip Codes Claudia Reardon" type:Feature Layer Collectio

<Item title:"Minerals_in_Julian__JD" type:Feature Layer Collection owner:duran057GIS>

<Item title:"Enriched San Diego County Zip Codes Ben" type:Feature Layer Collection owner:irvi

<Item title:"San Diego County Income and Uninsured SN" type:Feature Layer Collection owner:nes

<Item title:"Minerals_in_Julian___TW" type:Feature Layer Collection owner:snyde051>

<Item title:"Minerals_in_Julian_JJ" type:Feature Layer Collection owner:josep022>

<Item title:"SAN DIEGO COUNTY ZIP CODES-INCOME AND UNINSURED JH" type:Feature Layer Collection

<Item title:"ZIP Code Points to San Diego County Chronic Alcohol ED Discharge Data 2010 throug

<Item title:"Total Population vs. Minority Population OL" type:Feature Layer Collection owner:

<Item title:"San Diego ZIP Code Tech Data" type:Feature Layer Collection owner:Matt.Kaufman@am

<Item title:"Enriched San Diego County Zip Codes Benjamin Santia" type:Feature Layer Collectio

<Item title:"Zip_Codes" type:Feature Layer Collection owner:lsmith132>

<Item title:"San_Diego_County___Male_5__19___Female_5___19" type:Feature Layer Collection owne

<Item title:"Enriched San Diego County Zip Codes Moses" type:Feature Layer Collection owner:wo

<Item title:"Enriched San Diego County Zip Codes-GREGORIA" type:Feature Layer Collection owner

<Item title:"Dissolve_San_Diego_County_Zip_Codes" type:Feature Layer Collection owner:kireyes_
```

```
<Item title:"Zip Code" type:Feature Layer Collection owner:PowayGIS>
```

```
<Item title:"ZIP_CODES_SummarizeWithin" type:Feature Layer Collection owner:SuzannLeininger>
```

[64]:
```
zip_codes = gis.content.get('15e4d8d850674b0a8293e4d91978ae95').layers[0]
```

13 Adding zipcodes to the abive gis map

[65]:
```
map2.add_layer(zip_codes)
map2
```

```
MapView(layout=Layout(height='400px', width='100%'))
```

```
<IPython.core.display.HTML object>
```

[66]:
```
# col_zip_code = arcgis.
 ↪join_features(hate_info_fl,zip_codes,spatial_relationship =␣
 ↪'within',output_name = "finalprooup")
# col_zip_code.share(org = True)
```

[67]:
```
col_zip_code = gis.content.get('bcf72961387f456f8c56f60f360c4802')
```

[68]:
```
map3 = gis.map('San Diego County, CA')
map3.add_layer(hate_info_fl)
map3.add_layer(col_zip_code)
```

[69]:
```
map3
```

```
MapView(layout=Layout(height='400px', width='100%'))
```

```
<IPython.core.display.HTML object>
```

Col Zip Code is all hate crimes with a zip code attached. We now know where the crimes occured geographically and the area they occured in as well.

[238]:
```
col_zip_code = col_zip_code.layers[0].query().sdf
```

[236]:
```
hate_crime_by_zipcode = pd.DataFrame(col_zip_code.groupby('ZIP').sum().
 ↪Join_Count)
```

[237]:
```
hate_crime_by_zipcode.sort_values(by='Join_Count',ascending = False).head(10)
```

[237]:

```
       Join_Count
ZIP
92101          20
92103          12
92037          11
92102          11
92110          10
92115          10
92104           9
92116           8
92111           7
92105           7
```

[239]:

```
col_zip_code.head()
```

[239]:

```
   OBJECTID  Join_Count case_numbe                      time    crime_code  \
0         1           1   16000456  Early Morning/Late Night       243(d)m
1         2           1   16001278  Early Morning/Late Night         245a1
2         3           1   16004522  Early Morning/Late Night     594(b)(4)
3         4           1   16005962         Evening and Night  422.22(a)(4)
4         5           1   16005900  Early Morning/Late Night       417a1:m

               crime          beat block      street type  …  \
0            assault       gaslamp   500           g   st  …
1            assault    north park  3400        30th   st  …
2          vandalism  east village  1400    imperial   av  …
3  threat, phone call      bay park  4100         ute   dr  …
4             threat     hillcrest   100  university   av  …

                        address   latitude   longitude    ZIP  COMMUNITY  \
0          500 g st San Diego, CA  32.712638 -117.160073  92101  San Diego
1       3400 30th st San Diego, CA  32.741139 -117.130148  92104  San Diego
2    1400 imperial av San Diego, CA  32.706347 -117.151812  92101  San Diego
3         4100 ute dr San Diego, CA  32.807511 -117.203142  92117  San Diego
4  100 university av San Diego, CA  32.748341 -117.163831  92103  San Diego

    SHAPE_STAr    SHAPE_STLe   Shape__Area  Shape__Length  \
0  2.548928e+08  98792.532847  3.354889e+07   35841.688407
1  9.247708e+07  44329.073250  1.217787e+07   16090.264426
2  2.548928e+08  98792.532847  3.354889e+07   35841.688407
3  2.419047e+08  78195.217413  3.191454e+07   28394.157492
4  1.012375e+08  57977.990851  1.333285e+07   21037.430975

                                             SHAPE
0  {"x": -13042199.7126, "y": 3857223.2990000024,…
1  {"x": -13038868.4092, "y": 3860994.7250000015,…
2  {"x": -13041280.0363, "y": 3856391.0559, "spat…
```

```
3  {"x": -13046994.133299999, "y": 3869782.116499…
4  {"x": -13042617.992600001, "y": 3861947.898900…

[5 rows x 21 columns]
```

We now turned to the crime data collected over the last 6 months to see if any of the zipcodes matched. We found that the top zipcode 92101 was common for both and made this focal point fo our presentation and analysis

[74]:
```python
crime_data_zip = pd.DataFrame(crime_data.groupby('ZipCode').count())
#crime_data_zip
```

[75]:
```python
crime_data_zip.sort_values(by = 'CM_LEGEND',ascending =  False).head(10)
```

[75]:
```
         CM_LEGEND  agency  Charge_Description_Orig  activityDate  \
ZipCode
92101.0       5225    5225                     5225          5225
92109.0       2602    2602                     2602          2602
92054.0       2066    2066                     2066          2066
92020.0       1620    1620                     1620          1620
92113.0       1575    1575                     1575          1575
92110.0       1509    1509                     1509          1509
92025.0       1497    1497                     1497          1497
91910.0       1491    1491                     1491          1491
91911.0       1453    1453                     1453          1453
91950.0       1389    1389                     1389          1389

         BLOCK_ADDRESS  community
ZipCode
92101.0           5225       5094
92109.0           2602       2505
92054.0           2066       1982
92020.0           1620       1574
92113.0           1575       1546
92110.0           1509       1473
92025.0           1497       1496
91910.0           1491       1417
91911.0           1453       1400
91950.0           1389       1301
```

[76]:
```python
crime_data
```

[76]:
```
                  CM_LEGEND       agency  \
0             THEFT/LARCENY    OCEANSIDE
1             THEFT/LARCENY  CHULA VISTA
2                       DUI    SAN DIEGO
3       MOTOR VEHICLE THEFT  CHULA VISTA
```

```
4      DRUGS/ALCOHOL VIOLATIONS    ESCONDIDO
…                           …          …
61598    VEHICLE BREAK-IN/THEFT    SAN DIEGO
61599    VEHICLE BREAK-IN/THEFT    SAN DIEGO
61600    VEHICLE BREAK-IN/THEFT    SAN DIEGO
61601    VEHICLE BREAK-IN/THEFT    SAN DIEGO
61602             THEFT/LARCENY    SAN DIEGO

                                Charge_Description_Orig         activityDate  \
0                   PETTY THEFT(All Other Larceny) (M)  12/29/2019 18:11:00
1                            PETTY THEFT(Shoplift) (M)    1/2/2020 20:15:00
2                             DUI ALC/0.08 PERCENT (M)     1/1/2020 2:36:00
3      TAKE VEHICLE W/O OWNER'S CONSENT/VEHICLE THEFT…    1/6/2020 17:00:00
4                     POSSESS CONTROLLED SUBSTANCE (M)  12/28/2019 20:00:00
…                                                  …                   …
61598                     PETTY THEFT(Mot Veh Parts) (M)   9/23/2019 15:30:00
61599                             BURGLARY (VEHICLE) (F)   9/10/2019 11:00:00
61600                             BURGLARY (VEHICLE) (F)    9/23/2019 1:30:00
61601                             BURGLARY (VEHICLE) (F)   9/20/2019 13:00:00
61602                                        PETTY THEFT  11/21/2019 14:20:00

                           BLOCK_ADDRESS  ZipCode    community
0         1800  BLOCK COLLEGE BOULEVARD  92056.0    OCEANSIDE
1            600  BLOCK PALOMAR STREET  91911.0  CHULA VISTA
2              4000  BLOCK RUEDA DRIVE  92124.0    SAN DIEGO
3           1200  BLOCK ATWATER STREET  91913.0  CHULA VISTA
4        500 W  BLOCK WASHINGTON AVENUE  92025.0    ESCONDIDO
…                                    …        …          …
61598  1200  BLOCK CAMINO DEL RIO NORTH  92108.0    SAN DIEGO
61599       0  BLOCK FIESTA ISLAND ROAD  92109.0    SAN DIEGO
61600             0  BLOCK TWAIN AVENUE  92120.0          NaN
61601              8200  BLOCK LA VEREDA  92037.0    SAN DIEGO
61602         700  BLOCK NAUTILUS STREET  92037.0    SAN DIEGO

[61603 rows x 7 columns]
```

[77]:
```python
#Looking at victims and suspects in zipcode 92101
high_crime_zip = col_zip_code[col_zip_code['ZIP']==92101]
high_crime_zip.head()
```

[77]:
```
     OBJECTID  Join_Count case_numbe               time  crime_code  \
19         20           1   16024587                Day       245a1
39         40           1   16046634  Evening and Night       422.6
67         68           1   17015573                Day  459f & 594
90         91           1   18003081  Evening and Night      594(a)
124       125           1   18035021                Day       245a1
```

```
          crime             beat block      street type  …  \
19      assault     teralta east  4100   fairmount   av  …
39      assault  cherokee point  3900      landis   st  …
67     burglary      el cerrito  5400         lea   st  …
90    vandalism      fox canyon  3800      winona  ave  …
124     assault  colina del sol  5000  university   av  …

                          address   latitude   longitude    ZIP  \
19    4100 fairmount av San Diego, CA  32.751621 -117.100905  92105
39       3900 landis st San Diego, CA  32.745998 -117.110579  92105
67          5400 lea st San Diego, CA  32.745911 -117.079460  92105
90      3800 winona ave San Diego, CA  32.747130 -117.088068  92105
124  5000 university av San Diego, CA  32.749398 -117.086897  92105

      COMMUNITY    SHAPE_STAr   SHAPE_STLe   Shape__Area  Shape__Length  \
19   San Diego  1.533141e+08  65485.48112  2.018697e+07   23765.440339
39   San Diego  1.533141e+08  65485.48112  2.018697e+07   23765.440339
67   San Diego  1.533141e+08  65485.48112  2.018697e+07   23765.440339
90   San Diego  1.533141e+08  65485.48112  2.018697e+07   23765.440339
124  San Diego  1.533141e+08  65485.48112  2.018697e+07   23765.440339

                                                  SHAPE
19   {"x": -13035613.1699, "y": 3862382.0216000006,…
39   {"x": -13036690.070799999, "y": 3861637.774599…
67   {"x": -13033225.9117, "y": 3861626.2754999995,…
90   {"x": -13034184.1334, "y": 3861787.5434999987,…
124  {"x": -13034053.757199999, "y": 3862087.758900…

[5 rows x 21 columns]
```

Getting a list of case ID's to see which ones occured in respective zipcodes to see kind of crime occuring there

[78]:
```
case_id = list(high_crime_zip['case_numbe'])
```

[79]:
```
case_id
```

[79]:
```
['16024587',
 '16046634',
 '17015573',
 '18003081',
 '18035021',
 '18037683',
 '19024491']
```

[80]:
```
case_df = hate_crime.where(hate_crime['case_number'].isin(case_id))
case_df = case_df.dropna(how='all')
```

[81]:
```python
case_df[['motivation', 'number_of_suspects', 'suspect', 'victim_count',
       'victim_other', 'injury', 'suspect_race_0', 'suspect_race_1',
       'suspect_race_2', 'suspect_sex_0', 'suspect_sex_1', 'suspect_sex_2',
       'victim_race_0', 'victim_race_1', 'victim_race_2', 'victim_sex_0',
       'victim_sex_1', 'victim_sex_2']]
```

[81]:
```
              motivation number_of_suspects  suspect  victim_count  \
19                  Race                  3  Unknown           1.0
39                  Race                  1  Unknown           2.0
68                  Race                Unk  Unknown           1.0
91              Religion                  6    Known           2.0
125                 Race                  1    Known           2.0
126   Sexual Orientation                  2    Known           1.0
147                 Race                  1    Known           2.0

    victim_other    injury suspect_race_0 suspect_race_1 suspect_race_2  \
19           NaN  Hospital              A              A              A
39           NaN   Treated              B            NaN            NaN
68      Business      None            Unk            NaN            NaN
91           NaN      None              B              B              B
125          NaN   Treated              B            NaN            NaN
126          NaN      None              W              W            NaN
147          NaN      None              B            NaN            NaN

    suspect_sex_0 suspect_sex_1 suspect_sex_2 victim_race_0 victim_race_1  \
19              M             M             M             B           NaN
39              F           NaN           NaN             H             H
68            Unk           NaN           NaN             B           NaN
91              M             M             F             O             O
125             F           NaN           NaN             H             H
126             F             M           NaN             W           NaN
147             M           NaN           NaN             H             H

    victim_race_2 victim_sex_0 victim_sex_1 victim_sex_2
19            NaN            M          NaN          NaN
39            NaN            F            M          NaN
68            NaN            F          NaN          NaN
91            NaN            F            F          NaN
125           NaN            F            M          NaN
126           NaN            M          NaN          NaN
147           NaN            F            F          NaN
```

[82]:

14 Finding Average Income based on Census Tract data

14.0.1 Side Note: After completing our analysis we figured out that census tracts was not the best measure as we could enrich zipcodes and tracts and zipcode boundaries do not intersect. This led to us using geo-enrichment on the zipcodes as you will see below

[83]:
```
census_tracts = gpd.read_file('Census/CENSUS_TRACTS_2010.shp')
demographics = pd.read_excel('socio-demographic.xlsx')
```

[84]:
```
census_tracts.head()
```

[84]:
```
   TRACT    SHAPE_AREA     SHAPE_LEN  \
0   15.0  8.693887e+06  12443.272111
1   16.0  7.407379e+06  11329.616060
2   17.0  6.714940e+06  10791.678584
3   18.0  8.036708e+06  13929.689427
4   19.0  1.796665e+07  21026.710682

                                            geometry
0  POLYGON ((6293438.095 1853304.830, 6293503.297…
1  POLYGON ((6292472.285 1855719.495, 6292503.644…
2  POLYGON ((6292613.597 1857793.035, 6292669.857…
3  POLYGON ((6297364.831 1858582.740, 6297395.566…
4  POLYGON ((6296136.003 1863452.471, 6296242.697…
```

[85]:
```
demographics.head()
```

[85]:
```
   TRACT  YEAR  ORDINAL        INCOME GROUP  HOUSEHOLDS
0    1.0  2011        1   Less than $15,000          64
1    1.0  2011        2  $15,000 to $29,999           2
2    1.0  2011        3  $30,000 to $44,999          12
3    1.0  2011        4  $45,000 to $59,999         174
4    1.0  2011        5  $60,000 to $74,999          97
```

[86]:
```
merged = census_tracts.merge(demographics, on='TRACT')
merged.head()
```

[86]:
```
   TRACT    SHAPE_AREA     SHAPE_LEN  \
0   15.0  8.693887e+06  12443.272111
1   15.0  8.693887e+06  12443.272111
2   15.0  8.693887e+06  12443.272111
3   15.0  8.693887e+06  12443.272111
4   15.0  8.693887e+06  12443.272111

                                            geometry  YEAR  ORDINAL  \
0  POLYGON ((6293438.095 1853304.830, 6293503.297…  2011        1
```

```
1  POLYGON ((6293438.095 1853304.830, 6293503.297…  2011        2
2  POLYGON ((6293438.095 1853304.830, 6293503.297…  2011        3
3  POLYGON ((6293438.095 1853304.830, 6293503.297…  2011        4
4  POLYGON ((6293438.095 1853304.830, 6293503.297…  2011        5

         INCOME GROUP  HOUSEHOLDS
0   Less than $15,000         268
1  $15,000 to $29,999         219
2  $30,000 to $44,999         276
3  $45,000 to $59,999         264
4  $60,000 to $74,999         191
```

[87]:
```
d = {'Less than $15,000': 7500,
    '$15,000 to $29,999': 22500,
    '$30,000 to $44,999': 37500,
    '$45,000 to $59,999': 52500,
    '$60,000 to $74,999': 67500,
    '$75,000 to $99,999': 82500,
    '$100,000 to $124,999': 115000,
    '$125,000 to $149,999': 132500,
    '$150,000 to $199,999': 175000,
    '$200,000 or more': 200000 }
```

[88]:
```
avg_income_tract = merged
avg_income_tract['INCOME'] = avg_income_tract['INCOME GROUP'].map(d)
```

[89]:
```
merged.head()
```

[89]:
```
   TRACT    SHAPE_AREA     SHAPE_LEN  \
0   15.0  8.693887e+06  12443.272111
1   15.0  8.693887e+06  12443.272111
2   15.0  8.693887e+06  12443.272111
3   15.0  8.693887e+06  12443.272111
4   15.0  8.693887e+06  12443.272111

                                           geometry  YEAR  ORDINAL  \
0  POLYGON ((6293438.095 1853304.830, 6293503.297…  2011        1
1  POLYGON ((6293438.095 1853304.830, 6293503.297…  2011        2
2  POLYGON ((6293438.095 1853304.830, 6293503.297…  2011        3
3  POLYGON ((6293438.095 1853304.830, 6293503.297…  2011        4
4  POLYGON ((6293438.095 1853304.830, 6293503.297…  2011        5

         INCOME GROUP  HOUSEHOLDS  INCOME
0   Less than $15,000         268    7500
1  $15,000 to $29,999         219   22500
2  $30,000 to $44,999         276   37500
3  $45,000 to $59,999         264   52500
```

```
4  $60,000 to $74,999         191   67500
```

[90]:
```
avg_income_tract['totals']=␣
↪avg_income_tract['HOUSEHOLDS']*avg_income_tract['INCOME']
```

[91]:
```
average_income = sum(avg_income_tract['totals']) /␣
↪sum(avg_income_tract['HOUSEHOLDS'])
```

[92]:
```
avg_income_tract= avg_income_tract[['HOUSEHOLDS', 'INCOME', 'totals',␣
↪'TRACT','geometry']]
```

Calculating the average income per family in each tract

[93]:
```
avg_income_tract.head()
```

[93]:
```
   HOUSEHOLDS  INCOME    totals  TRACT  \
0         268    7500   2010000   15.0
1         219   22500   4927500   15.0
2         276   37500  10350000   15.0
3         264   52500  13860000   15.0
4         191   67500  12892500   15.0

                                            geometry
0  POLYGON ((6293438.095 1853304.830, 6293503.297…
1  POLYGON ((6293438.095 1853304.830, 6293503.297…
2  POLYGON ((6293438.095 1853304.830, 6293503.297…
3  POLYGON ((6293438.095 1853304.830, 6293503.297…
4  POLYGON ((6293438.095 1853304.830, 6293503.297…
```

[94]:
```
avg_income_tract.columns
```

[94]:
```
Index(['HOUSEHOLDS', 'INCOME', 'totals', 'TRACT', 'geometry'], dtype='object')
```

[95]:
```
avg_income_tract = avg_income_tract.groupby(['TRACT'], as_index = False).sum()
```

[96]:
```
avg_income_tract['average'] = avg_income_tract['totals'] /␣
↪avg_income_tract['HOUSEHOLDS']
```

[97]:
```
avg_income_tract[avg_income_tract['TRACT']==33.04]
```

[97]:
```
    TRACT  HOUSEHOLDS   INCOME     totals       average
75  33.04        6868  6247500  267615000  38965.492137
```

Creating a dictionary of census tract and geometry to try and find intersections as will be observed later

[98]:
```
# #merge avg_income_tract and merged on TRACT and get geometry ad dtop
# a = avg_income_tract.merge(merged,on = 'TRACT',how='right')
```

```python
# a.head()
a = merged.copy()
a = a[['TRACT','geometry']]
geom = a.set_index('TRACT').to_dict()['geometry']
geom
```

```
[98]: {15.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd03f128>,
 16.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd03f048>,
 17.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd03ffd0>,
 18.0: <shapely.geometry.polygon.Polygon at 0x7fc3694c4438>,
 19.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd10e240>,
 20.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd10e1d0>,
 20.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd184f60>,
 21.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd184438>,
 22.01: <shapely.geometry.polygon.Polygon at 0x7fc369551c18>,
 22.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd10b278>,
 23.01: <shapely.geometry.polygon.Polygon at 0x7fc373012ef0>,
 23.02: <shapely.geometry.polygon.Polygon at 0x7fc3730128d0>,
 24.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd039f28>,
 24.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd039160>,
 25.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd0d5160>,
 25.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd0d5048>,
 26.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd0ed2b0>,
 26.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd031f28>,
 27.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd031ba8>,
 27.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd031898>,
 1.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd18dcf8>,
 2.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd1825f8>,
 2.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd20afd0>,
 3.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd20ad30>,
 4.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd20a898>,
 5.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd20a438>,
 6.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd20a9e8>,
 7.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd20ada0>,
 8.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd20ae80>,
 9.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd20a6a0>,
 10.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd20a198>,
 11.0: <shapely.geometry.polygon.Polygon at 0x7fc2bc6b5a90>,
 12.0: <shapely.geometry.polygon.Polygon at 0x7fc2bc6b5710>,
 13.0: <shapely.geometry.polygon.Polygon at 0x7fc2bc6b5ac8>,
 14.0: <shapely.geometry.polygon.Polygon at 0x7fc2bc6b5080>,
 32.01: <shapely.geometry.polygon.Polygon at 0x7fc2bc6b5400>,
 32.02: <shapely.geometry.polygon.Polygon at 0x7fc2bc6b54a8>,
 32.04: <shapely.geometry.polygon.Polygon at 0x7fc2bc6b5470>,
 32.07: <shapely.geometry.polygon.Polygon at 0x7fc2bc6b59b0>,
 32.08: <shapely.geometry.polygon.Polygon at 0x7fc369491860>,
 32.09: <shapely.geometry.polygon.Polygon at 0x7fc2bc6a6be0>,
```

```
32.11: <shapely.geometry.polygon.Polygon at 0x7fc2bc6a6da0>,
32.12: <shapely.geometry.polygon.Polygon at 0x7fc2bc6a6e48>,
32.13: <shapely.geometry.polygon.Polygon at 0x7fc2bc6a6c50>,
32.14: <shapely.geometry.polygon.Polygon at 0x7fc2bc6a6b00>,
33.01: <shapely.geometry.polygon.Polygon at 0x7fc2bc6a6a90>,
33.03: <shapely.geometry.polygon.Polygon at 0x7fc369335390>,
33.04: <shapely.geometry.polygon.Polygon at 0x7fc3693353c8>,
33.05: <shapely.geometry.polygon.Polygon at 0x7fc369335a20>,
34.01: <shapely.geometry.polygon.Polygon at 0x7fc369335780>,
27.05: <shapely.geometry.polygon.Polygon at 0x7fc3693a0cc0>,
27.07: <shapely.geometry.polygon.Polygon at 0x7fc3693c4940>,
27.08: <shapely.geometry.polygon.Polygon at 0x7fc2bcee7908>,
27.09: <shapely.geometry.polygon.Polygon at 0x7fc369445d68>,
27.1: <shapely.geometry.polygon.Polygon at 0x7fc3692bfd68>,
27.11: <shapely.geometry.polygon.Polygon at 0x7fc3692bf9e8>,
27.12: <shapely.geometry.polygon.Polygon at 0x7fc369330080>,
28.01: <shapely.geometry.polygon.Polygon at 0x7fc36942eb70>,
28.03: <shapely.geometry.polygon.Polygon at 0x7fc3693809b0>,
28.04: <shapely.geometry.polygon.Polygon at 0x7fc3694b54a8>,
29.02: <shapely.geometry.polygon.Polygon at 0x7fc3694f38d0>,
29.03: <shapely.geometry.polygon.Polygon at 0x7fc37326a7b8>,
29.04: <shapely.geometry.polygon.Polygon at 0x7fc369541710>,
29.05: <shapely.geometry.polygon.Polygon at 0x7fc368f796a0>,
30.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd20f908>,
30.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd20f3c8>,
30.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd20fc18>,
31.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd20f358>,
31.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd20f240>,
31.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd20f668>,
31.07: <shapely.geometry.polygon.Polygon at 0x7fc2bd20fd30>,
31.08: <shapely.geometry.polygon.Polygon at 0x7fc2bd20f390>,
31.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd20f898>,
31.11: <shapely.geometry.polygon.Polygon at 0x7fc2bd20f860>,
31.12: <shapely.geometry.polygon.Polygon at 0x7fc2bd20f7f0>,
31.13: <shapely.geometry.polygon.Polygon at 0x7fc2bd20f940>,
31.14: <shapely.geometry.polygon.Polygon at 0x7fc2bd20fa58>,
31.15: <shapely.geometry.polygon.Polygon at 0x7fc2bd20f438>,
66.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd20fa20>,
68.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd20fda0>,
68.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd20fac8>,
69.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd20fb00>,
70.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd20ff60>,
71.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcee7eb8>,
72.0: <shapely.geometry.polygon.Polygon at 0x7fc373295710>,
73.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcee1940>,
73.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcf0cf28>,
74.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcefdeb8>,
```

75.01: <shapely.geometry.polygon.Polygon at 0x7fc2bceeb978>,
75.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd20feb8>,
76.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd20fe80>,
77.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd20fe48>,
77.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd20ffd0>,
78.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc080>,
79.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc0f0>,
34.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc160>,
34.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc1d0>,
35.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc240>,
35.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc2b0>,
36.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc320>,
36.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc390>,
36.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc400>,
38.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc470>,
39.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc4e0>,
39.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc550>,
40.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc5c0>,
41.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc630>,
42.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc6a0>,
43.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc710>,
44.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc780>,
45.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc7f0>,
46.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc860>,
47.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc8d0>,
48.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc940>,
49.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffc9b0>,
50.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffca20>,
51.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffca90>,
52.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffcb00>,
53.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffcb70>,
54.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffcbe0>,
55.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffcc50>,
56.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffccc0>,
57.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffcda0>,
58.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffce10>,
59.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffce80>,
60.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffcef0>,
61.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffcf60>,
62.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffcfd0>,
63.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5080>,
65.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcff50f0>,
83.29: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5160>,
83.3: <shapely.geometry.polygon.Polygon at 0x7fc2bcff51d0>,
83.31: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5240>,
83.33: <shapely.geometry.polygon.Polygon at 0x7fc2bcff52b0>,
83.35: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5320>,

```
83.36: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5390>,
83.37: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5400>,
83.39: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5470>,
83.4: <shapely.geometry.polygon.Polygon at 0x7fc2bcff54e0>,
83.41: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5550>,
83.43: <shapely.geometry.polygon.Polygon at 0x7fc2bcff55c0>,
83.44: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5630>,
79.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcff56a0>,
79.07: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5710>,
79.08: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5780>,
79.1: <shapely.geometry.polygon.Polygon at 0x7fc2bcff57f0>,
80.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5860>,
80.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcff58d0>,
80.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5940>,
81.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcff59b0>,
81.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5a20>,
82.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcffccf8>,
83.01: <shapely.geometry.multipolygon.MultiPolygon at 0x7fc2bd03f390>,
83.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5ba8>,
83.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5c18>,
83.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5c88>,
83.07: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5cf8>,
83.1: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5d68>,
83.11: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5dd8>,
83.12: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5e48>,
83.13: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5eb8>,
83.24: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5f28>,
83.27: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5f98>,
83.28: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9048>,
85.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd90b8>,
85.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9128>,
85.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9198>,
85.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9208>,
85.07: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9278>,
85.09: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd92e8>,
85.1: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9358>,
85.11: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd93c8>,
85.12: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9438>,
85.13: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd94a8>,
83.45: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9518>,
83.46: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9588>,
83.47: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd95f8>,
83.48: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9668>,
83.49: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd96d8>,
83.5: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9748>,
83.51: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd97b8>,
83.52: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9828>,
```

```
83.53: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9898>,
83.54: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9908>,
83.55: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9978>,
83.56: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd99e8>,
83.57: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9a58>,
83.58: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9ac8>,
83.59: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9b38>,
83.6: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9ba8>,
83.61: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9c18>,
83.62: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9c88>,
83.63: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9cf8>,
83.64: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9d68>,
83.65: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9dd8>,
83.66: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9e48>,
85.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9eb8>,
85.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9f28>,
95.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfd9f98>,
95.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd002048>,
95.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd0020b8>,
95.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd002128>,
95.07: <shapely.geometry.polygon.Polygon at 0x7fc2bd002198>,
95.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd002208>,
95.1: <shapely.geometry.polygon.Polygon at 0x7fc2bd002278>,
95.11: <shapely.geometry.polygon.Polygon at 0x7fc2bd0022e8>,
96.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd002358>,
96.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd0023c8>,
96.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd002438>,
97.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd0024a8>,
97.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd002518>,
97.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd002588>,
97.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd0025f8>,
98.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd002668>,
86.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd0026d8>,
87.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd002748>,
87.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd0027b8>,
88.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd002828>,
89.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd002898>,
89.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd002908>,
90.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd002978>,
91.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd0029e8>,
91.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd002a58>,
91.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd002ac8>,
91.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd002b38>,
91.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd002ba8>,
91.07: <shapely.geometry.polygon.Polygon at 0x7fc2bd002c18>,
92.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd002c88>,
92.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd002cf8>,
```

```
93.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd002d68>,
93.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd002dd8>,
93.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd002e48>,
93.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd002eb8>,
94.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd002f28>,
109.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd002f98>,
110.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8048>,
111.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe80b8>,
113.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8128>,
116.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8198>,
116.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8208>,
117.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8278>,
118.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe82e8>,
118.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8358>,
119.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe83c8>,
120.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8438>,
120.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe84a8>,
121.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8518>,
98.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8588>,
98.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe85f8>,
98.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8668>,
99.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcff5ac8>,
99.02: <shapely.geometry.multipolygon.MultiPolygon at 0x7fc2bd03f2e8>,
100.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe87f0>,
100.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8860>,
100.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe88d0>,
100.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8940>,
100.09: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe89b0>,
100.1: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8a20>,
100.11: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8a90>,
100.12: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8b00>,
100.13: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8b70>,
100.14: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8be0>,
100.15: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8c50>,
101.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8cc0>,
101.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8d30>,
101.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8da0>,
101.07: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8e10>,
101.09: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8e80>,
101.1: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8ef0>,
101.11: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8f60>,
101.12: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8fd0>,
102.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9080>,
103.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe90f0>,
104.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9160>,
104.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe91d0>,
105.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9240>,
```

```
105.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe92b0>,
106.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9320>,
108.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9390>,
133.1: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9400>,
133.11: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9470>,
133.12: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe94e0>,
133.13: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9550>,
133.14: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe95c0>,
134.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9630>,
134.09: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe96a0>,
134.1: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9710>,
134.11: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9780>,
121.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe97f0>,
122.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9860>,
123.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe98d0>,
123.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9940>,
123.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe99b0>,
124.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9a20>,
124.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9a90>,
125.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9b00>,
125.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9b70>,
126.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9be0>,
127.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9c50>,
128.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9cc0>,
129.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9d30>,
130.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9da0>,
131.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9e10>,
131.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9e80>,
131.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9ef0>,
132.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9f60>,
132.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe9fd0>,
132.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe080>,
132.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe0f0>,
133.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe160>,
133.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe1d0>,
133.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe240>,
133.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe2b0>,
133.07: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe320>,
133.08: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe390>,
133.09: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe400>,
139.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe470>,
139.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe4e0>,
139.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe550>,
139.07: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe5c0>,
139.08: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe630>,
139.09: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe6a0>,
140.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe710>,
```

140.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe780>,
141.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe7f0>,
141.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe860>,
134.12: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe8d0>,
134.14: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe940>,
134.15: <shapely.geometry.polygon.Polygon at 0x7fc2bcffe9b0>,
134.16: <shapely.geometry.polygon.Polygon at 0x7fc2bcffea20>,
134.17: <shapely.geometry.polygon.Polygon at 0x7fc2bcffea90>,
134.18: <shapely.geometry.polygon.Polygon at 0x7fc2bcffeb00>,
134.19: <shapely.geometry.polygon.Polygon at 0x7fc2bcffeb70>,
134.2: <shapely.geometry.polygon.Polygon at 0x7fc2bcffebe0>,
134.21: <shapely.geometry.polygon.Polygon at 0x7fc2bcffec50>,
135.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcffecc0>,
135.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcffed30>,
135.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcffeda0>,
135.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcffee10>,
136.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcffee80>,
136.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcffeef0>,
136.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcffef60>,
136.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcffefd0>,
137.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde080>,
137.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde0f0>,
138.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde160>,
138.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde1d0>,
154.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde240>,
156.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde2b0>,
157.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde320>,
157.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde390>,
157.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde400>,
158.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde470>,
158.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde4e0>,
159.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde550>,
159.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde5c0>,
160.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde630>,
161.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde6a0>,
162.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde710>,
162.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde780>,
163.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde7f0>,
163.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde860>,
164.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde8d0>,
164.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde940>,
142.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfde9b0>,
143.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdea20>,
144.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdea90>,
145.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdeb00>,
146.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdeb70>,
146.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdebe0>,

```
147.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdec50>,
148.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdecc0>,
148.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcfded30>,
148.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdeda0>,
148.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdee10>,
149.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdee80>,
149.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdeef0>,
150.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdef60>,
151.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcfdefd0>,
152.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8080>,
153.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a80f0>,
153.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8160>,
154.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a81d0>,
154.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8240>,
154.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a82b0>,
155.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8320>,
155.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8390>,
156.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8400>,
168.07: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8470>,
168.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a84e0>,
168.1: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8550>,
168.11: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a85c0>,
169.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8630>,
169.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a86a0>,
170.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8710>,
170.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8780>,
170.1: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a87f0>,
165.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8860>,
165.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a88d0>,
165.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8940>,
166.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a89b0>,
166.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8a20>,
166.07: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8a90>,
166.08: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8b00>,
166.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8b70>,
166.1: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8be0>,
166.12: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8c50>,
166.13: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8cc0>,
166.14: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8d30>,
166.15: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8da0>,
166.16: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8e10>,
166.17: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8e80>,
167.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8ef0>,
167.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8f60>,
168.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a8fd0>,
168.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2080>,
168.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d20f0>,
```

```
170.45: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2160>,
170.46: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d21d0>,
170.47: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2240>,
170.48: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d22b0>,
170.49: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2320>,
170.5: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2390>,
170.51: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2400>,
170.52: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2470>,
170.14: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d24e0>,
170.15: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2550>,
170.18: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d25c0>,
170.19: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2630>,
170.2: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d26a0>,
170.21: <shapely.geometry.polygon.Polygon at 0x7fc2bcfe8710>,
170.22: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2748>,
170.29: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d27b8>,
170.3: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2828>,
170.31: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2898>,
170.32: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2908>,
170.33: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2978>,
170.34: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d29e8>,
170.35: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2a58>,
170.36: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2ac8>,
170.37: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2b38>,
170.39: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2ba8>,
170.4: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2c18>,
170.41: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2c88>,
170.42: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2cf8>,
170.43: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2d68>,
170.44: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2dd8>,
176.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2e48>,
176.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2eb8>,
176.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2f28>,
177.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2d2f98>,
177.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb048>,
178.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb0b8>,
178.08: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb128>,
170.53: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb198>,
170.54: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb208>,
170.55: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb278>,
170.56: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb2e8>,
171.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb358>,
171.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb3c8>,
171.07: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb438>,
171.08: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb4a8>,
171.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb518>,
171.1: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb588>,
```

172.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb5f8>,
173.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb668>,
173.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb6d8>,
173.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb748>,
173.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb7b8>,
174.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb828>,
174.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb898>,
174.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb908>,
175.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb978>,
175.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cb9e8>,
186.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cba58>,
186.08: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbac8>,
186.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbb38>,
186.1: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbba8>,
186.11: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbc18>,
187.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbc88>,
188.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbcf8>,
188.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbd68>,
188.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbdd8>,
186.12: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbe48>,
186.13: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbeb8>,
186.14: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbf28>,
189.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2cbf98>,
189.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38048>,
189.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcf380b8>,
189.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38128>,
190.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38198>,
190.02: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38208>,
191.01: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38278>,
191.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcf382e8>,
191.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38358>,
191.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcf383c8>,
191.07: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38438>,
192.03: <shapely.geometry.polygon.Polygon at 0x7fc2bcf384a8>,
192.05: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38518>,
192.06: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38588>,
192.07: <shapely.geometry.polygon.Polygon at 0x7fc2bcf385f8>,
192.08: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38668>,
178.09: <shapely.geometry.polygon.Polygon at 0x7fc2bcf386d8>,
178.1: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38748>,
178.11: <shapely.geometry.polygon.Polygon at 0x7fc2bcf387b8>,
178.13: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38828>,
179.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38898>,
180.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38908>,
181.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38978>,
182.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcf389e8>,
183.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38a58>,

```
184.0: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38ac8>,
185.04: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38b38>,
185.07: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38ba8>,
185.09: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38c18>,
185.1: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38c88>,
185.11: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38cf8>,
185.12: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38d68>,
185.13: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38dd8>,
185.14: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38e48>,
185.15: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38eb8>,
185.16: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38f28>,
185.17: <shapely.geometry.polygon.Polygon at 0x7fc2bcf38f98>,
185.18: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2048>,
185.19: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a20b8>,
186.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2128>,
199.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2198>,
199.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2208>,
199.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2278>,
200.13: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a22e8>,
200.14: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2358>,
200.15: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a23c8>,
193.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2438>,
193.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a24a8>,
193.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2518>,
194.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2588>,
194.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a25f8>,
194.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2668>,
194.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a26d8>,
195.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2748>,
195.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a27b8>,
195.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2828>,
198.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2898>,
196.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2908>,
196.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2978>,
197.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a29e8>,
197.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2a58>,
198.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2ac8>,
198.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2b38>,
198.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2ba8>,
198.08: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2c18>,
198.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2c88>,
199.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2cf8>,
202.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2d68>,
202.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2dd8>,
202.07: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2e48>,
202.08: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2eb8>,
202.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2f28>,
```

202.1: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a2f98>,
202.11: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6048>,
202.13: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a60b8>,
202.14: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6128>,
203.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6198>,
203.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6208>,
203.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6278>,
203.07: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a62e8>,
203.08: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6358>,
203.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a63c8>,
204.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6438>,
204.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a64a8>,
204.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6518>,
200.16: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6588>,
200.17: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a65f8>,
200.18: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6668>,
200.19: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a66d8>,
200.2: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6748>,
200.21: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a67b8>,
200.22: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6828>,
200.23: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6898>,
200.24: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6908>,
200.25: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6978>,
200.26: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a69e8>,
200.27: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6a58>,
200.28: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6ac8>,
200.29: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6b38>,
201.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6ba8>,
201.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6c18>,
201.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6c88>,
201.07: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6cf8>,
201.08: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6d68>,
201.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6dd8>,
204.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6e48>,
205.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6eb8>,
206.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6f28>,
206.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd2a6f98>,
207.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e048>,
207.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e0b8>,
207.07: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e128>,
207.08: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e198>,
207.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e208>,
207.1: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e278>,
208.01: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e2e8>,
208.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e358>,
208.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e3c8>,
208.07: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e438>,

```
208.09: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e4a8>,
208.1: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e518>,
208.11: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e588>,
209.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e5f8>,
209.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e668>,
210.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e6d8>,
211.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e748>,
212.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e7b8>,
212.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e828>,
212.05: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e898>,
212.06: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e908>,
213.02: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e978>,
213.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd29e9e8>,
213.04: <shapely.geometry.polygon.Polygon at 0x7fc2bd29ea58>,
215.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd29eac8>,
214.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd29eb38>,
216.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd29eba8>,
218.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd29ec18>,
219.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd29ec88>,
220.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd29ecf8>,
221.0: <shapely.geometry.polygon.Polygon at 0x7fc2bd29ed68>,
209.03: <shapely.geometry.polygon.Polygon at 0x7fc2bd29edd8>}
```

[99]:
```
avg_income_tract['geometry']=avg_income_tract['TRACT'].map(geom)
```

[100]:
```
avg_income_tract= avg_income_tract[['TRACT','average','geometry']]
avg_income_tract.head()
```

[100]:

	TRACT	average	geometry
0	1.00	122181.441189	POLYGON ((6273121 1857292, 6273168.000249997 1…
1	2.01	91811.231680	POLYGON ((6278383.999750003 1856813, 6278375 1…
2	2.02	88669.357402	POLYGON ((6276705.420750007 1854067.768000007,…
3	3.00	73456.245067	POLYGON ((6279822.865999997 1853534.077999994,…
4	4.00	63765.879614	POLYGON ((6281407.650000006 1857087.834999993,…

[101]:
```
#avg_income_tract['avg income'] = average_income
```

[102]:
```
avg_income_tract.head(10)
```

[102]:

	TRACT	average	geometry
0	1.00	122181.441189	POLYGON ((6273121 1857292, 6273168.000249997 1…
1	2.01	91811.231680	POLYGON ((6278383.999750003 1856813, 6278375 1…
2	2.02	88669.357402	POLYGON ((6276705.420750007 1854067.768000007,…
3	3.00	73456.245067	POLYGON ((6279822.865999997 1853534.077999994,…
4	4.00	63765.879614	POLYGON ((6281407.650000006 1857087.834999993,…
5	5.00	72611.013986	POLYGON ((6287139.963 1859651.998750001, 62871…
6	6.00	68049.249638	POLYGON ((6282762.033999994 1856530.594999999,…

```
7   7.00   70227.037556  POLYGON ((6285023.987000003 1853290.105000004,…
8   8.00   56432.246029  POLYGON ((6286120.351750001 1853275.568000004,…
9   9.00   44801.843870  POLYGON ((6286578.569000006 1855755.341000006,…
```

[103]:
```
avg_income_tract = gpd.GeoDataFrame(avg_income_tract,geometry='geometry')
```

[104]:
```
avg_income_tract.crs = {'init':'epsg:2230'}
```

[105]:
```
result = gis.content.search('Census Tracts, San Diego ' , item_type="Feature␣
 ↪Layer", outside_org=True, max_items=30)
result
from IPython.display import display
for item in result:
    display(item)
```

```
<Item title:"Tracts Within 1 Mile Wilshire Blvd (UC 2015)" type:Feature Layer Collection owner

<Item title:"Census Tracts 2010" type:Feature Layer Collection owner:ewendt>

<Item title:"GEOG320_RichardsonH_RoadsHighways" type:Feature Layer Collection owner:richa119>

<Item title:"Barrio_Logan_MAD_WFL1" type:Feature Layer Collection owner:tysterns>

<Item title:"San Diego Census Tracts 2" type:Feature Layer Collection owner:itzai001>

<Item title:"schools_near_parks_with_80_crime" type:Feature Layer Collection owner:issey.10176

<Item title:"CENSUS TRACTS San Diego 2010" type:Feature Layer Collection owner:izaslavsky_ucsd

<Item title:"San_Diego_CA" type:Feature Layer Collection owner:IDM800002143_ohiostate>

<Item title:"Total Crime Index by Census Tract  2016 San Diego County" type:Feature Layer Coll

<Item title:"City of Escondido Roads" type:Feature Layer Collection owner:pjhickman>

<Item title:"San Diego Census 4" type:Feature Layer Collection owner:itzai001>

<Item title:"nhgis0001_shapefile_tl2010_us_county_2010" type:Feature Layer Collection owner:lm

<Item title:"Census Tracts - Wilshire Blvd 1-mile Buffer" type:Feature Layer Collection owner:
```

```
<Item title:"2012_AutoThefts_by_Census_Tract_San_Diego" type:Feature Layer Collection owner:Je

<Item title:"San_Diego_Internet_accessmarch3" type:Feature Layer Collection owner:mgg027_UCSDO

<Item title:"2010 Household Size" type:Feature Layer Collection owner:riple005>

<Item title:"Poverty Level 2010" type:Feature Layer Collection owner:riple005>

<Item title:"Intersect_of_CENSUS_TRACTS_2010_and_San_Diego_Council_District_4" type:Feature La

<Item title:"Minority Population 2010" type:Feature Layer Collection owner:riple005>

<Item title:"San Diego Public Health Data" type:Feature Layer Collection owner:j6yoon_ucsd>

<Item title:"Vehicle Ownership" type:Feature Layer Collection owner:riple005>

<Item title:"San Diego Census Tracts" type:Feature Layer Collection owner:itzai001>

<Item title:"San Diego Census Tract 3" type:Feature Layer Collection owner:itzai001>

<Item title:"San_Diego_Heart_Attack_and_Heart_Disease_2017" type:Feature Layer Collection owne

<Item title:"FoodInsecurity" type:Feature Layer Collection owner:whcheung>

<Item title:"CENSUS_TRACTS_2010" type:Feature Layer Collection owner:YMCACRS>

<Item title:"Intersect_of_San_Diego_Council_District_4_and_CENSUS_TRACTS_2010" type:Feature La

<Item title:"Intersect_of_CENSUS_TRACTS_2010_and_San_Diego_Council_District_8" type:Feature La

<Item title:"homeless_numbers" type:Feature Layer Collection owner:utgraphics>

<Item title:"San Diego County Census  Tracts" type:Feature Layer Collection owner:Rizbee>
```

[106]:
```
crime_by_tract = gis.content.get('684e982fac8c42a09c47118e6c422bec')
```

[240]:
```
#Using GIS features we were exploring to see different crimes by zipcode.
```

[107]:
```
map5 = gis.map('San Diego County, CA')
map5.add_layer(crime_by_tract)
map5.legend = True
map5
```

```
MapView(layout=Layout(height='400px', width='100%'), legend=True)
```

```
<IPython.core.display.HTML object>
```

[108]:
```
#avg_income_tract.to_file('Average_Income_Updated.shp')
```

[109]:
```
avg_income_tract.crs
```

[109]:
```
{'init': 'epsg:2230'}
```

[110]:
```
avg_income_tract_fl = gis.content.get('597a1b82e98b4dcaba3c6c7c3934b67e').
↪layers[0]
```

[111]:
```
a = gis.content.get('597a1b82e98b4dcaba3c6c7c3934b67e')
a
```

[111]:
```
<Item title:"Proj_Avg_Income" type:Feature Layer Collection
owner:aubarrio_ucsd5>
```

[112]:
```
# avg_income_tract_sdf = avg_income_tract_fl.query().sdf
```

[113]:
```
map6 = gis.map('San Diego County, CA')
#map6.add_layer(avg_income_zip_code)
#map6
```

[114]:
```
#avg_income_tract_sdf
```

[115]:
```
#avg_income_tract
```

[116]:
```
#type(avg_income_tract_sdf)
```

[117]:
```
#avg_income_tract_sdf.to_csv("avg_income")
```

[118]:
```
# avg_income_zip_code = arcgis.join_features(zip_codes,avg_income_tract_fl,␣
↪spatial_relationship = 'within',output_name = "projectPJAB_AverageUpdated")
# avg_income_zip_code.share(org = True)
```

Below we tried to merge zipcode and census tract only to fond out they only intersected at one boundary thereby making what we did above not useful in context of the project

[119]:
```
avg_income_zip_code = gis.content.get('619cf53fd8784a958ab5f30196f6f781')
```

[120]:
```
avg_income_zip_code
```

[120]:
```
<Item title:"projectPJAB_AverageUpdated" type:Feature Layer Collection
owner:pjuneja_ucsd5>
```

[121]:
```
b = avg_income_zip_code.layers[0].query().sdf
```

[122]:
```
b
```

[122]:
```
   OBJECTID  Join_Count    ZIP        COMMUNITY    SHAPE_STAr    SHAPE_STLe  \
0         1           1  92056        Oceanside  4.948242e+00     52.440204
1         2           1  92058  Camp Pendleton  2.201062e+06   6038.244965
2         3           1  92058  Camp Pendleton  4.808182e+06   9232.450459
3         4           1  92058  Camp Pendleton  6.818365e+06  10959.779040
4         5           1  92058  Camp Pendleton  1.174368e+07  14854.188451
5         6           1  92155         Coronado  8.588175e+06  16251.173722

    Shape__Area  Shape__Length   TRACT       average  Shape__Area_1  \
0  6.601562e-01      19.100803  185.14  86743.634548   5.872626e+06
1  2.935304e+05    2205.572596  187.00  48908.269031   7.827174e+08
2  6.418915e+05    3375.201143  187.00  48908.269031   7.827174e+08
3  9.098164e+05    4002.988363  187.00  48908.269031   7.827174e+08
4  1.564434e+06    5422.823089  187.00  48908.269031   7.827174e+08
5  1.129188e+06    5894.197663  216.00  79190.594785   1.975226e+07

   Shape__Length_1                                              SHAPE
0     10370.861595  {"rings": [[[-13056857.5549, 3925275.1411], [-…
1    139452.109614  {"rings": [[[-13058334.5401, 3935498.2375], [-…
2    139452.109614  {"rings": [[[-13060014.2171, 3940825.775], [-1…
3    139452.109614  {"rings": [[[-13058573.5418, 3939058.3987], [-…
4    139452.109614  {"rings": [[[-13069064.7073, 3928086.4517], [-…
5     20636.152152  {"rings": [[[-13043191.828, 3852170.1938], [-1…
```

[123]:
```
b[b['ZIP']==92056]
```

[123]:
```
   OBJECTID  Join_Count    ZIP  COMMUNITY  SHAPE_STAr  SHAPE_STLe  \
0         1           1  92056  Oceanside    4.948242   52.440204

   Shape__Area  Shape__Length   TRACT       average  Shape__Area_1  \
0     0.660156      19.100803  185.14  86743.634548   5.872626e+06

   Shape__Length_1                                              SHAPE
0     10370.861595  {"rings": [[[-13056857.5549, 3925275.1411], [-…
```

[124]:
```
# zip_income = list(b['ZIP'])
```

[125]:
```
#cdf = crime_data.where(b['ZIP'].isin(zip_income))
#case_df = case_df.dropna(how='all')
```

[126]:
```
crime_data[crime_data['ZipCode']==92056]
```

[126]:
```
                     CM_LEGEND     agency  \
0                THEFT/LARCENY  OCEANSIDE
70     DRUGS/ALCOHOL VIOLATIONS  OCEANSIDE
79               THEFT/LARCENY  OCEANSIDE
87          MOTOR VEHICLE THEFT  OCEANSIDE
89                     ASSAULT  OCEANSIDE
…                          …          …
60895            THEFT/LARCENY  OCEANSIDE
60906  DRUGS/ALCOHOL VIOLATIONS  OCEANSIDE
61057                  ROBBERY  OCEANSIDE
61253  DRUGS/ALCOHOL VIOLATIONS  OCEANSIDE
61359            THEFT/LARCENY  OCEANSIDE

                           Charge_Description_Orig         activityDate  \
0             PETTY THEFT(All Other Larceny) (M)  12/29/2019 18:11:00
70          POSS CONTROLLED SUBS PARAPHERNALIA (M)    1/2/2020 23:00:00
79             PETTY THEFT(All Other Larceny) (M)  12/11/2019 20:21:00
87     TAKE VEHICLE W/O OWNER'S CONSENT/VEHICLE THEFT…   1/29/2020 16:00:00
89            ASSAULT W/DEADLY WEAPON:NOT F/ARM (F)    2/1/2020 17:30:00
…                                              …                    …
60895          PETTY THEFT(All Other Larceny) (M)    10/1/2019 4:50:00
60906       POSS CONTROLLED SUBS PARAPHERNALIA (M)   9/21/2019 11:17:00
61057                                 ROBBERY (F)   10/7/2019 12:45:00
61253         USE/UNDER INFL OF CONTROLLED SUBS (M)    9/27/2019 0:57:00
61359                       PETTY THEFT(Shoplift) (M)  10/16/2019 11:20:00

                        BLOCK_ADDRESS  ZipCode  community
0         1800  BLOCK COLLEGE BOULEVARD  92056.0  OCEANSIDE
70     COLLEGE BOULEVARD & PLAZA DRIVE  92056.0  OCEANSIDE
79              3400  BLOCK MARRON ROAD  92056.0  OCEANSIDE
87           3300  BLOCK EL CORAZON DRIVE  92056.0  OCEANSIDE
89     4100  BLOCK OCEANSIDE BOULEVARD  92056.0  OCEANSIDE
…                                  …        …        …
60895     1900  BLOCK COLLEGE BOULEVARD  92056.0  OCEANSIDE
60906     1900  BLOCK COLLEGE BOULEVARD  92056.0  OCEANSIDE
61057     3500  BLOCK COLLEGE BOULEVARD  92056.0  OCEANSIDE
61253          3600  BLOCK SPYGLASS WAY  92056.0  OCEANSIDE
61359     1900  BLOCK COLLEGE BOULEVARD  92056.0  OCEANSIDE

[1014 rows x 7 columns]
```

[127]:
```
hate_info_sdf.head()
```

[127]:
```
   case_number                          time    crime_code              crime  \
0     16000456  Early Morning/Late Night       243(d)m            assault
1     16001278  Early Morning/Late Night         245a1            assault
2     16004522  Early Morning/Late Night     594(b)(4)          vandalism
3     16005962         Evening and Night  422.22(a)(4)  threat, phone call
4     16005900  Early Morning/Late Night       417a1:m             threat

            beat block      street type               weapon  \
0        gaslamp   500           g   st  hands, fists, feet
1     north park  3400        30th   st               stick
2  east village  1400    imperial   av              marker
3       bay park  4100         ute   dr               phone
4      hillcrest   100  university   av               knife

                          address   latitude   longitude  \
0           500 g st San Diego, CA  32.712638 -117.160073
1        3400 30th st San Diego, CA  32.741139 -117.130148
2     1400 imperial av San Diego, CA  32.706347 -117.151812
3          4100 ute dr San Diego, CA  32.807511 -117.203142
4   100 university av San Diego, CA  32.748341 -117.163831

                                               SHAPE
0  {"x": -117.16007340369976, "y": 32.71263809630…
1  {"x": -117.13014779569856, "y": 32.74113929569…
2  {"x": -117.15181181069018, "y": 32.70634749036…
3  {"x": -117.20314241727988, "y": 32.80751149470…
4  {"x": -117.16383087642083, "y": 32.74834112357…
```

[128]:
```
hate_crime_by_zipcode.head()
```

[128]:
```
       Join_Count
ZIP
91910           1
91950           1
92037          11
92101          20
92102          11
```

15 GEOENRICHMENT

[129]:
```
from arcgis.geoenrichment import standard_geography_query,enrich
```

After our work above we realised we had to change the approach of finding out more about our zipcodes and this is where zipcodes came into account

[130]:
```python
avg_income_tract.spatial.set_geometry='geometry'
```

[131]:
```python
def geo(zipcode):
    return standard_geography_query(source_country = 'US',layers = ['US.ZIP5'],
                                    geoquery=zipcode, return_geometry = True
    ).iloc[[0]]
```

[132]:
```python
a
```

[132]:
```
<Item title:"Proj_Avg_Income" type:Feature Layer Collection
owner:aubarrio_ucsd5>
```

[133]:
```python
zcode = col_zip_code.ZIP.unique().astype(int).astype(str).tolist()
```

[134]:
```python
zip_geo = list(map(geo,zcode))
```

[135]:
```python
zip_data = pd.concat(zip_geo)
```

[136]:
```python
col_zip_code.head()
```

[136]:
```
   OBJECTID  Join_Count case_numbe                     time    crime_code  \
0         1           1   16000456  Early Morning/Late Night       243(d)m
1         2           1   16001278  Early Morning/Late Night         245a1
2         3           1   16004522  Early Morning/Late Night     594(b)(4)
3         4           1   16005962         Evening and Night  422.22(a)(4)
4         5           1   16005900  Early Morning/Late Night       417a1:m

                crime           beat block      street type  …  \
0             assault        gaslamp   500           g   st  …
1             assault     north park  3400        30th   st  …
2           vandalism  east village  1400    imperial   av  …
3  threat, phone call       bay park  4100         ute   dr  …
4              threat      hillcrest   100  university   av  …

                          address   latitude   longitude    ZIP  COMMUNITY  \
0          500 g st San Diego, CA  32.712638 -117.160073  92101  San Diego
1       3400 30th st San Diego, CA  32.741139 -117.130148  92104  San Diego
2     1400 imperial av San Diego, CA  32.706347 -117.151812  92101  San Diego
3          4100 ute dr San Diego, CA  32.807511 -117.203142  92117  San Diego
4  100 university av San Diego, CA  32.748341 -117.163831  92103  San Diego

     SHAPE_STAr    SHAPE_STLe  Shape__Area  Shape__Length  \
0  2.548928e+08  98792.532847  3.354889e+07   35841.688407
1  9.247708e+07  44329.073250  1.217787e+07   16090.264426
2  2.548928e+08  98792.532847  3.354889e+07   35841.688407
3  2.419047e+08  78195.217413  3.191454e+07   28394.157492
4  1.012375e+08  57977.990851  1.333285e+07   21037.430975
```

```
                                             SHAPE
0  {"x": -13042199.7126, "y": 3857223.2990000024,…
1  {"x": -13038868.4092, "y": 3860994.7250000015,…
2  {"x": -13041280.0363, "y": 3856391.0559, "spat…
3  {"x": -13046994.133299999, "y": 3869782.116499…
4  {"x": -13042617.992600001, "y": 3861947.898900…

[5 rows x 21 columns]
```

This is how we enriched each row to get the value we needed

[137]:
```
bus = pd.concat(map(
    lambda studyrow: enrich(studyrow[1].to_frame().T,␣
 ↪["Demographic_and_Income_Profile_rep"]),
    zip_data.iterrows()
))
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: FutureWarning:
Sorting because non-concatenation axis is not aligned. A future version
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

  This is separate from the ipykernel package so we can avoid doing imports
until
```

[138]:
```
bus.columns
```

[138]:
```
Index(['AreaID', 'AreaName', 'CountryAbbr', 'DataLayerID', 'DatasetID',
       'FAMRATE', 'FAMRATE_S', 'HHRATE', 'HHRATE_S', 'HINC0_FY', 'HINC100_FY',
       'HINC150_FY', 'HINC15_FY', 'HINC200_FY', 'HINC25_FY', 'HINC35_FY',
       'HINC50_FY', 'HINC75_FY', 'HINCBASECY', 'HINCBASEFY', 'HasData', 'ID',
       'INCRATE', 'INCRATE_S', 'MajorSubdivisionAbbr', 'MajorSubdivisionName',
       'MajorSubdivisionType', 'OBJECTID_0', 'OWNRATE', 'OWNRATE_S',
       'ObjectId', 'POPRATE', 'POPRATE_S', 'RACEBASE10', 'RACEBASECY',
       'RACEBASEFY', 'SHAPE', 'Score', 'aggregationMethod',
       'apportionmentConfidence', 'populationToPolygonSizeRating',
       'sourceCountry'],
      dtype='object')
```

After reading online about each of these columns, we decided to take these ones into account which were more inline with our hypothesis

[139]:
```
other_details = bus[['AreaID','HasData', 'ID',
       'INCRATE', 'INCRATE_S', 'MajorSubdivisionAbbr', 'MajorSubdivisionName',
```

```
    'MajorSubdivisionType', 'OBJECTID_0', 'OWNRATE', 'OWNRATE_S',
    'ObjectId', 'POPRATE', 'POPRATE_S', 'RACEBASE10', 'RACEBASECY',
    'RACEBASEFY','populationToPolygonSizeRating']]
```

[140]:
```
bus.head(10)
```

[140]:
```
   AreaID   AreaName CountryAbbr DataLayerID      DatasetID  FAMRATE FAMRATE_S  \
0  92101  San Diego          US     US.ZIP5  USA_ESRI_2019      NaN       NaN
0  92104  San Diego          US     US.ZIP5  USA_ESRI_2019     0.62     0.62%
0  92117  San Diego          US     US.ZIP5  USA_ESRI_2019     0.62     0.62%
0  92103  San Diego          US     US.ZIP5  USA_ESRI_2019     0.62     0.62%
0  92115  San Diego          US     US.ZIP5  USA_ESRI_2019     0.62     0.62%
0  92120  San Diego          US     US.ZIP5  USA_ESRI_2019     0.62     0.62%
0  92102  San Diego          US     US.ZIP5  USA_ESRI_2019     0.62     0.62%
0  92130  San Diego          US     US.ZIP5  USA_ESRI_2019     0.62     0.62%
0  92109  San Diego          US     US.ZIP5  USA_ESRI_2019     0.62     0.62%
0  92108  San Diego          US     US.ZIP5  USA_ESRI_2019     0.62     0.62%

   HHRATE HHRATE_S  HINC0_FY  …  POPRATE_S  RACEBASE10  RACEBASECY  \
0     NaN      NaN      3978  …        NaN       36944       45988
0    0.62    0.62%      1639  …      0.67%       45855       47813
0    0.62    0.62%       887  …      0.67%       49820       51488
0    0.62    0.62%      1145  …      0.67%       30814       32870
0    0.62    0.62%      2630  …      0.67%       58790       62229
0    0.62    0.62%       416  …      0.67%       27952       30345
0    0.62    0.62%      1176  …      0.67%       43550       45428
0    0.62    0.62%       789  …      0.67%       49157       60674
0    0.62    0.62%      1315  …      0.67%       45689       47244
0    0.62    0.62%       756  …      0.67%       19629       24239

   RACEBASEFY                                              SHAPE  Score  \
0       53477  {"rings": [[[-117.18204335248471, 32.742493541…    100
0       49461  {"rings": [[[-117.14317000016992, 32.757229999…    100
0       52641  {"rings": [[[-117.20509000006223, 32.847050000…    100
0       34165  {"rings": [[[-117.19488523460865, 32.760073268…    100
0       64059  {"rings": [[[-117.0936399999627, 32.7805200001…    100
0       31463  {"rings": [[[-117.06467999954882, 32.854370000…    100
0       46586  {"rings": [[[-117.13199999952575, 32.727559999…    100
0       65737  {"rings": [[[-117.21119443926122, 32.984090220…    100
0       48500  {"rings": [[[-117.22909000073776, 32.823400000…    100
0       26915  {"rings": [[[-117.1343399012861, 32.7896918002…    100

                 aggregationMethod  apportionmentConfidence  \
0  BlockApportionment:US.BlockGroups                    2.576
0  BlockApportionment:US.BlockGroups                    2.576
0  BlockApportionment:US.BlockGroups                    2.576
0  BlockApportionment:US.BlockGroups                    2.576
```

```
0  BlockApportionment:US.BlockGroups                        2.576
0  BlockApportionment:US.BlockGroups                        2.576
0  BlockApportionment:US.BlockGroups                        2.576
0  BlockApportionment:US.BlockGroups                        2.576
0  BlockApportionment:US.BlockGroups                        2.576
0  BlockApportionment:US.BlockGroups                        2.576

   populationToPolygonSizeRating  sourceCountry
0                          2.191             US
0                          2.191             US
0                          2.191             US
0                          2.191             US
0                          2.191             US
0                          2.191             US
0                          2.191             US
0                          2.191             US
0                          2.191             US
0                          2.191             US

[10 rows x 42 columns]
```

16 Over here we took the the income columns from the enriched dataset to try and find out average income for each areaId which is the zipcode in this case

[141]:
```
demog = bus[['AreaID','HINC0_FY', 'HINC15_FY','HINC25_FY', 'HINC35_FY',␣
↪'HINC50_FY', 'HINC75_FY', 'HINC100_FY',
       'HINC150_FY', 'HINC200_FY']]
```

[142]:
```
demog.head()
```

[142]:
```
  AreaID  HINC0_FY  HINC15_FY  HINC25_FY  HINC35_FY  HINC50_FY  HINC75_FY  \
0  92101      3978       2321       1811       2075       3863       3364
0  92104      1639       1483       1741       2956       4443       3071
0  92117       887       1022       1070       1856       3584       2950
0  92103      1145        951        918       1392       3051       2610
0  92115      2630       1939       2318       2567       3963       2886

   HINC100_FY  HINC150_FY  HINC200_FY
0        5291        3525        4464
0        3933        2130        2017
0        4637        2577        2495
0        4220        2225        3061
0        3776        1374        1633
```

[143]:
```
demog = demog.rename(columns = {'AreaID':'ZIPCODE'})
```

[144]:
```
len(demog)
```

[144]: 30

[145]:
```
demog = demog.rename(columns = {'HINC0_FY':'0-15k','HINC15_FY':
↪'15-25k','HINC25_FY':'25-35k',
                                'HINC35_FY':'35-50k','HINC50_FY':
↪'50-75k','HINC75_FY':'75-100k',
                                'HINC100_FY':'100-150k','HINC150_FY':
↪'150-200k','HINC200_FY':'200k+'})
```

[146]:
```
demog
```

[146]:
```
   ZIPCODE  0-15k  15-25k  25-35k  35-50k  50-75k  75-100k  100-150k  150-200k  \
0    92101   3978    2321    1811    2075    3863     3364      5291      3525
0    92104   1639    1483    1741    2956    4443     3071      3933      2130
0    92117    887    1022    1070    1856    3584     2950      4637      2577
0    92103   1145     951     918    1392    3051     2610      4220      2225
0    92115   2630    1939    2318    2567    3963     2886      3776      1374
0    92120    416     435     617     809    1781     2058      3340      1709
0    92102   1176    1597    1552    2092    3077     1681      2292       755
0    92130    789     285     467     502    1285     1424      4037      4030
0    92109   1315     829    1217    1875    4163     4036      5231      2894
0    92108    756     439     683     978    2583     1810      3773      1854
0    92139    415     565     893    1177    2021     2023      2201      1093
0    91910   2209    1845    1960    3345    4554     4069      5485      2649
0    92110    824     753     561     943    2035     1783      2638      1120
0    92105   2459    2418    2573    3164    4016     2526      2733       760
0    92111   1147    1008    1148    1763    2851     2404      3732      2241
0    92113   1999    1883    1708    2054    2326     1307      1528       411
0    92126    568     673     914    1665    3301     3743      7426      4210
0    92124    254     239     399     698    1519     1688      2426      1467
0    92123    474     300     553    1009    2010     1699      3382      1726
0    92119    392     401     516     733    1707     1337      2334      1430
0    92106    289     344     284     538    1011      905      1461      1082
0    92037    994     594     688     975    1843     1785      3275      2462
0    92116    901     755    1154    2202    3130     2819      3182      1584
0    92129    483     481     367     938    1588     1971      4042      3952
0    92154   1216    1148    1547    2180    4077     3891      5135      2351
0    92128    532     543     842    1256    2440     2497      5125      3905
0    92107    545     626     619    1185    2236     1981      3327      1469
0    91950   1907    1857    1791    2385    3250     2103      2412      1340
0    92114   1044    1067    1243    2006    3186     2923      3907      1441
0    92122   1590     773     836    1601    2727     2930      5069      2876
```

```
   200k+
0   4464
0   2017
0   2495
0   3061
0   1633
0   2210
0    790
0  10053
0   3038
0   1499
0    384
0   1736
0   1149
0    701
0   1599
0    372
0   3217
0   1648
0   1236
0   1067
0   2412
0   6467
0   1663
0   5573
0   1211
0   3783
0   1821
0    498
0    909
0   3063
```

[147]:
```
d= demog.copy()
```

[148]:
```
demog['total_households'] = demog.sum(axis=1)
```

[149]:
```
demog.columns
```

[149]:
```
Index(['ZIPCODE', '0-15k', '15-25k', '25-35k', '35-50k', '50-75k', '75-100k',
       '100-150k', '150-200k', '200k+', 'total_households'],
      dtype='object')
```

[150]:
```
d['avg_val_15k'] = 7500*d['0-15k']
d['avg_val_25k'] = 20000*d['15-25k']
d['avg_val_35k'] = 30000*d['25-35k']
d['avg_val_50k'] = 42500*d['35-50k']
d['avg_val_75k'] = 62500*d['50-75k']
```

```
d['avg_val_100k'] = 87500*d['75-100k']
d['avg_val_150k'] = 125000*d['100-150k']
d['avg_val_200k'] = 175000*d['150-200k']
d['avg_val_200k+'] = 200000*d['200k+']
```

[151]:
```
d = d.drop(columns = ['0-15k', '15-25k', '25-35k', '35-50k', '50-75k',␣
↪'75-100k',
       '100-150k', '150-200k', '200k+'])
```

[152]:
```
#d['total'] = demog.sum(axis=1)
#d = d.drop(columns = ['total'])
```

[153]:
```
d.head()
```

[153]:
```
   ZIPCODE  avg_val_15k  avg_val_25k  avg_val_35k  avg_val_50k  avg_val_75k  \
0    92101     29835000     46420000     54330000     88187500    241437500
0    92104     12292500     29660000     52230000    125630000    277687500
0    92117      6652500     20440000     32100000     78880000    224000000
0    92103      8587500     19020000     27540000     59160000    190687500
0    92115     19725000     38780000     69540000    109097500    247687500

   avg_val_100k  avg_val_150k  avg_val_200k  avg_val_200k+
0     294350000     661375000     616875000      892800000
0     268712500     491625000     372750000      403400000
0     258125000     579625000     450975000      499000000
0     228375000     527500000     389375000      612200000
0     252525000     472000000     240450000      326600000
```

[154]:
```
d['total_sum_weight'] = d.sum(axis=1)
```

[155]:
```
#d
```

[156]:
```
d['total_house']=demog['total_households']
```

[157]:
```
d
```

[157]:
```
   ZIPCODE  avg_val_15k  avg_val_25k  avg_val_35k  avg_val_50k  avg_val_75k  \
0    92101     29835000     46420000     54330000     88187500    241437500
0    92104     12292500     29660000     52230000    125630000    277687500
0    92117      6652500     20440000     32100000     78880000    224000000
0    92103      8587500     19020000     27540000     59160000    190687500
0    92115     19725000     38780000     69540000    109097500    247687500
0    92120      3120000      8700000     18510000     34382500    111312500
0    92102      8820000     31940000     46560000     88910000    192312500
0    92130      5917500      5700000     14010000     21335000     80312500
0    92109      9862500     16580000     36510000     79687500    260187500
0    92108      5670000      8780000     20490000     41565000    161437500
```

0	92139	3112500	11300000	26790000	50022500	126312500
0	91910	16567500	36900000	58800000	142162500	284625000
0	92110	6180000	15060000	16830000	40077500	127187500
0	92105	18442500	48360000	77190000	134470000	251000000
0	92111	8602500	20160000	34440000	74927500	178187500
0	92113	14992500	37660000	51240000	87295000	145375000
0	92126	4260000	13460000	27420000	70762500	206312500
0	92124	1905000	4780000	11970000	29665000	94937500
0	92123	3555000	6000000	16590000	42882500	125625000
0	92119	2940000	8020000	15480000	31152500	106687500
0	92106	2167500	6880000	8520000	22865000	63187500
0	92037	7455000	11880000	20640000	41437500	115187500
0	92116	6757500	15100000	34620000	93585000	195625000
0	92129	3622500	9620000	11010000	39865000	99250000
0	92154	9120000	22960000	46410000	92650000	254812500
0	92128	3990000	10860000	25260000	53380000	152500000
0	92107	4087500	12520000	18570000	50362500	139750000
0	91950	14302500	37140000	53730000	101362500	203125000
0	92114	7830000	21340000	37290000	85255000	199125000
0	92122	11925000	15460000	25080000	68042500	170437500

	avg_val_100k	avg_val_150k	avg_val_200k	avg_val_200k+	total_sum_weight \
0	294350000	661375000	616875000	892800000	2925610000
0	268712500	491625000	372750000	403400000	2033987500
0	258125000	579625000	450975000	499000000	2149797500
0	228375000	527500000	389375000	612200000	2062445000
0	252525000	472000000	240450000	326600000	1776405000
0	180075000	417500000	299075000	442000000	1514675000
0	147087500	286500000	132125000	158000000	1092255000
0	124600000	504625000	705250000	2010600000	3472350000
0	353150000	653875000	506450000	607600000	2523902500
0	158375000	471625000	324450000	299800000	1492192500
0	177012500	275125000	191275000	76800000	937750000
0	356037500	685625000	463575000	347200000	2391492500
0	156012500	329750000	196000000	229800000	1116897500
0	221025000	341625000	133000000	140200000	1365312500
0	210350000	466500000	392175000	319800000	1705142500
0	114362500	191000000	71925000	74400000	788250000
0	327512500	928250000	736750000	643400000	2958127500
0	147700000	303250000	256725000	329600000	1180532500
0	148662500	422750000	302050000	247200000	1315315000
0	116987500	291750000	250250000	213400000	1036667500
0	79187500	182625000	189350000	482400000	1037182500
0	156187500	409375000	430850000	1293400000	2486412500
0	246662500	397750000	277200000	332600000	1599900000
0	172462500	505250000	691600000	1114600000	2647280000
0	340462500	641875000	411425000	242200000	2061915000

```
0   218487500   640625000   683375000   756600000    2545077500
0   173337500   415875000   257075000   364200000    1435777500
0   184012500   301500000   234500000    99600000    1229272500
0   255762500   488375000   252175000   181800000    1528952500
0   256375000   633625000   503300000   612600000    2296845000

   total_house
0        30692
0        23413
0        21078
0        19573
0        23086
0        13375
0        15012
0        22872
0        24598
0        14375
0        10772
0        27852
0        11806
0        21350
0        17893
0        13588
0        25717
0        10338
0        12389
0         9917
0         8326
0        19083
0        17390
0        19395
0        22756
0        20923
0        13809
0        17543
0        17726
0        21465
```

[158]:
```
d['average']=d['total_sum_weight']/d['total_house']
```

[159]:
```
d = d.sort_values('average',ascending= 'False')
```

[160]:
```
d=d[['ZIPCODE','average']]
```

16.0.1 Below is table of zipcode and average income in each zipcode

[161]:
```
d
```

[161]:

	ZIPCODE	average
0	92113	58010.744775
0	92105	63949.063232
0	91950	70071.966026
0	92102	72758.792966
0	92115	76947.284068
0	91910	85864.300589
0	92114	86254.795216
0	92104	86874.279247
0	92139	87054.400297
0	92154	90609.729302
0	92116	92001.150086
0	92110	94604.226664
0	92111	95296.624378
0	92101	95321.582171
0	92117	101992.480311
0	92109	102606.004553
0	92108	103804.695652
0	92107	103974.038670
0	92119	104534.385399
0	92103	105371.940939
0	92123	106167.971588
0	92122	107004.192872
0	92120	113246.728972
0	92124	114193.509383
0	92126	115026.150017
0	92128	121640.180662
0	92106	124571.522940
0	92037	130294.633967
0	92129	136492.910544
0	92130	151816.631689

[162]:
```
col_zip_code.columns
```

[162]:
```
Index(['OBJECTID', 'Join_Count', 'case_numbe', 'time', 'crime_code', 'crime',
       'beat', 'block', 'street', 'type', 'weapon', 'address', 'latitude',
       'longitude', 'ZIP', 'COMMUNITY', 'SHAPE_STAr', 'SHAPE_STLe',
       'Shape__Area', 'Shape__Length', 'SHAPE'],
      dtype='object')
```

[163]:
```
hate_info.head()
```

[163]:
```
  case_number                         time    crime_code                 crime  \
0    16000456  Early Morning/Late Night       243(d)m               assault
1    16001278  Early Morning/Late Night         245a1               assault
2    16004522  Early Morning/Late Night     594(b)(4)             vandalism
3    16005962           Evening and Night  422.22(a)(4)  threat, phone call
4    16005900  Early Morning/Late Night       417a1:m                threat

           beat block       street type              weapon  \
0       gaslamp   500            g   st  hands, fists, feet
1    north park  3400         30th   st               stick
2  east village  1400     imperial   av              marker
3      bay park  4100          ute   dr               phone
4     hillcrest   100   university   av               knife

                          address   latitude   longitude  \
0           500 g st San Diego, CA  32.712638 -117.160073
1        3400 30th st San Diego, CA  32.741139 -117.130148
2    1400 imperial av San Diego, CA  32.706347 -117.151812
3          4100 ute dr San Diego, CA  32.807511 -117.203142
4  100 university av San Diego, CA  32.748341 -117.163831

                                          SHAPE
0  {"x": -117.16007340369976, "y": 32.71263809630…
1  {"x": -117.13014779569856, "y": 32.74113929569…
2  {"x": -117.15181181069018, "y": 32.70634749036…
3  {"x": -117.20314241727988, "y": 32.80751149470…
4  {"x": -117.16383087642083, "y": 32.74834112357…
```

[164]:
```python
d= d.rename(columns = {'ZIPCODE':'ZIP'})
```

[165]:
```python
#d = d.drop(columns = ['crime'])
```

Our focus now is to combine our hypothesis involving income, base race and number of hatecrimes per zipcode into one table and this is an example of how we went about it

[ ]:
```python

```

[166]:
```python
#d['ZIP'].astype(int)
```

[167]:
```python
hate_crime_by_zipcode = hate_crime_by_zipcode.reset_index()
```

[168]:
```python
#a = d.merge(hate_crime_by_zipcode,on = 'ZIP')
```

[169]:
```python
hate_crime_by_zipcode.head()
```

[169]:
```
     ZIP  Join_Count
0  91910           1
```

```
1  91950           1
2  92037          11
3  92101          20
4  92102          11
```

[170]:
```python
d['crime'] = hate_crime_by_zipcode['Join_Count']
```

[171]:
```python
d = d.drop(columns = ['crime'])
```

[172]:
```python
hate_crime_by_zipcode
```

[172]:

	ZIP	Join_Count
0	91910	1
1	91950	1
2	92037	11
3	92101	20
4	92102	11
5	92103	12
6	92104	9
7	92105	7
8	92106	1
9	92107	4
10	92108	2
11	92109	5
12	92110	10
13	92111	7
14	92113	4
15	92114	2
16	92115	10
17	92116	8
18	92117	3
19	92119	1
20	92120	2
21	92122	2
22	92123	2
23	92124	2
24	92126	6
25	92128	1
26	92129	3
27	92130	7
28	92139	4
29	92154	1

[173]:
```python
# a = pd.concat(list(hate_crime_by_zipcode['Join_Count']),d)
```

[174]:
```python
#d['ZIP'].astype(int)
d.astype({'ZIP': 'int64'}).dtypes
```

[174]:
```
ZIP          int64
average    float64
dtype: object
```

[175]:
```python
d.head()
```

[175]:

	ZIP	average
0	92113	58010.744775
0	92105	63949.063232
0	91950	70071.966026
0	92102	72758.792966
0	92115	76947.284068

[176]:
```python
hate_dic =hate_crime_by_zipcode.set_index('ZIP').to_dict()['Join_Count']
```

[177]:
```python
d['crime'] =d.ZIP.map(hate_dic)
```

[178]:
```python
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30 entries, 0 to 0
Data columns (total 3 columns):
ZIP        30 non-null object
average    30 non-null float64
crime      0 non-null float64
dtypes: float64(2), object(1)
memory usage: 960.0+ bytes
```

[179]:
```python
crime_per_zip = d.copy()
```

[180]:
```python
crime_per_zip = crime_per_zip.drop(columns = ['crime'])
```

[181]:
```python
income_dict = crime_per_zip.set_index('ZIP').to_dict()['average']
```

[182]:
```python
crime_per_zip = crime_per_zip.astype('int64', copy=False)
```

[ ]:
```python

```

[183]:
```python
crime_per_zip['number_crimes'] = crime_per_zip['ZIP'].map(hate_dic)
```

[184]:
```python
hate_crime_by_zipcode.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
ZIP           30 non-null int64
Join_Count    30 non-null int64
```

```
dtypes: int64(2)
memory usage: 608.0 bytes
```

[185]:
```
crime_per_zip.sort_values(by = ['number_crimes'],ascending= False).head()
```

[185]:
```
     ZIP  average  number_crimes
0  92101    95321             20
0  92103   105371             12
0  92037   130294             11
0  92102    72758             11
0  92115    76947             10
```

[186]:
```
hate_info.head()
```

[186]:
```
  case_number                      time    crime_code              crime  \
0    16000456  Early Morning/Late Night       243(d)m            assault
1    16001278  Early Morning/Late Night         245a1            assault
2    16004522  Early Morning/Late Night     594(b)(4)          vandalism
3    16005962         Evening and Night  422.22(a)(4)  threat, phone call
4    16005900  Early Morning/Late Night       417a1:m             threat

           beat block      street type             weapon  \
0       gaslamp   500           g   st  hands, fists, feet
1    north park  3400        30th   st               stick
2  east village  1400    imperial   av              marker
3      bay park  4100         ute   dr               phone
4     hillcrest   100  university   av               knife

                               address   latitude   longitude  \
0            500 g st San Diego, CA  32.712638 -117.160073
1         3400 30th st San Diego, CA  32.741139 -117.130148
2      1400 imperial av San Diego, CA  32.706347 -117.151812
3           4100 ute dr San Diego, CA  32.807511 -117.203142
4    100 university av San Diego, CA  32.748341 -117.163831

                                           SHAPE
0  {"x": -117.16007340369976, "y": 32.71263809630…
1  {"x": -117.13014779569856, "y": 32.74113929569…
2  {"x": -117.15181181069018, "y": 32.70634749036…
3  {"x": -117.20314241727988, "y": 32.80751149470…
4  {"x": -117.16383087642083, "y": 32.74834112357…
```

[187]:
```
other_details.columns
```

[187]:
```
Index(['AreaID', 'HasData', 'ID', 'INCRATE', 'INCRATE_S',
       'MajorSubdivisionAbbr', 'MajorSubdivisionName', 'MajorSubdivisionType',
       'OBJECTID_0', 'OWNRATE', 'OWNRATE_S', 'ObjectId', 'POPRATE',
```

```
         'POPRATE_S', 'RACEBASE10', 'RACEBASECY', 'RACEBASEFY',
         'populationToPolygonSizeRating'],
        dtype='object')
```

[188]:
```python
#other_details = other_details.drop(columns = ['HasData',␣
↪'ID','MajorSubdivisionName','MajorSubdivisionType'
                                                         #␣
↪,'MajorSubdivisionAbbr','ObjectId','OBJECTID_0'])
other_details = other_details[['AreaID','RACEBASECY']]
```

[ ]:

[189]:
```python
od = other_details.astype('int64', copy=False)
```

[190]:
```python
race_dict = od.set_index('AreaID').to_dict()['RACEBASECY']
```

[191]:
```python
other_details.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30 entries, 0 to 0
Data columns (total 2 columns):
AreaID         30 non-null object
RACEBASECY     30 non-null int64
dtypes: int64(1), object(1)
memory usage: 720.0+ bytes
```

[192]:
```python
#other_details = other_details.astype('int64', copy=False)
```

[193]:
```python
crime_per_zip['race']= crime_per_zip['ZIP'].map(race_dict)
```

[194]:
```python
crime_per_zip.head()
```

[194]:

	ZIP	average	number_crimes	race
0	92113	58010	4	55813
0	92105	63949	7	71446
0	91950	70071	1	59261
0	92102	72758	11	45428
0	92115	76947	10	62229

[195]:
```python
crime_per_zip.sort_values(by = ['number_crimes'],ascending= False).head()
```

[195]:

	ZIP	average	number_crimes	race
0	92101	95321	20	45988
0	92103	105371	12	32870
0	92037	130294	11	41404
0	92102	72758	11	45428
0	92115	76947	10	62229

[196]:
```python
final = crime_per_zip.merge(col_zip_code, left_on='ZIP', right_on='ZIP')
```

[197]:
```python
final.columns
```

[197]:
```
Index(['ZIP', 'average', 'number_crimes', 'race', 'OBJECTID', 'Join_Count',
       'case_numbe', 'time', 'crime_code', 'crime', 'beat', 'block', 'street',
       'type', 'weapon', 'address', 'latitude', 'longitude', 'COMMUNITY',
       'SHAPE_STAr', 'SHAPE_STLe', 'Shape__Area', 'Shape__Length', 'SHAPE'],
      dtype='object')
```

[198]:
```python
final_drop = final.drop(columns = ['OBJECTID','COMMUNITY','beat', 'block',␣
 ↪'street',
       'type','crime_code','SHAPE'])
```

[ ]:
```python

```

[199]:
```python
final_drop_sdf = pd.DataFrame.spatial.from_xy(final_drop,x_column =␣
 ↪'longitude', y_column = 'latitude')
```

[200]:
```python
final_drop_sdf
```

[200]:
```
       ZIP  average  number_crimes   race  Join_Count case_numbe  \
0    92113    58010              4  55813           1   16028219
1    92113    58010              4  55813           1   18005199
2    92113    58010              4  55813           1   19006514
3    92113    58010              4  55813           1   19019302
4    92105    63949              7  71446           1   16024587
..     …        …              …    …           …          …
154  92130   151816              7  60674           1   17010649
155  92130   151816              7  60674           1   17019690
156  92130   151816              7  60674           1   17036619
157  92130   151816              7  60674           1   19005259
158  92130   151816              7  60674           1   19008983

                         time      crime              weapon  \
0                         Day    robbery               knife
1           Evening and Night    assault               hands
2                         Day    assault               stick
3           Evening and Night  vandalism         spray paint
4                         Day    assault                 rod
..                        …          …                   …
154                       Day    assault  hands, fists, feet
155         Evening and Night  vandalism              marker
156  Early Morning/Late Night  vandalism              marker
157         Evening and Night  vandalism                 pen
158                       Day    assault  hands, fists, feet
```

```
                                     address   latitude   longitude  \
0              3500 main st San Diego, CA  32.687809 -117.118594
1             3700 birch st San Diego, CA  32.689123 -117.114540
2             2000 logan av San Diego, CA  32.700325 -117.142498
3              2200 main st San Diego, CA  32.695622 -117.141571
4         4100 fairmount av San Diego, CA  32.751621 -117.100905
..                                    …          …           …
154    6600 carmel valley rd San Diego, CA  32.968256 -117.178362
155         4700 fairport way San Diego, CA  32.925910 -117.219040
156        12900 cristallo pl San Diego, CA  32.954789 -117.222098
157        13000 jadestone wy San Diego, CA  32.962624 -117.233624
158  3700 del mar heights rd San Diego, CA  32.956466 -117.225881

       SHAPE_STAr     SHAPE_STLe   Shape__Area  Shape__Length  \
0    1.399861e+08   65765.279016  1.841453e+07   23834.868005
1    1.399861e+08   65765.279016  1.841453e+07   23834.868005
2    1.399861e+08   65765.279016  1.841453e+07   23834.868005
3    1.399861e+08   65765.279016  1.841453e+07   23834.868005
4    1.533141e+08   65485.481120  2.018697e+07   23765.440339
..            …             …             …              …
154  5.181484e+08  107213.084764  6.854966e+07   39003.295186
155  5.181484e+08  107213.084764  6.854966e+07   39003.295186
156  5.181484e+08  107213.084764  6.854966e+07   39003.295186
157  5.181484e+08  107213.084764  6.854966e+07   39003.295186
158  5.181484e+08  107213.084764  6.854966e+07   39003.295186

                                                 SHAPE
0    {"x": -117.11859403, "y": 32.68780928, "spatia…
1    {"x": -117.11453952, "y": 32.68912298, "spatia…
2    {"x": -117.14249803, "y": 32.70032541, "spatia…
3    {"x": -117.14157103, "y": 32.69562156, "spatia…
4    {"x": -117.10090548, "y": 32.751621, "spatialR…
..                                                 …
154  {"x": -117.17836165, "y": 32.96825648, "spatia…
155  {"x": -117.21904022, "y": 32.92591022, "spatia…
156  {"x": -117.22209789, "y": 32.95478867, "spatia…
157  {"x": -117.23362423, "y": 32.96262385, "spatia…
158  {"x": -117.22588062, "y": 32.95646568, "spatia…

[159 rows x 17 columns]
```

[201]:
```
# final_drop.spatial.set_geometry='SHAPE'
```

[202]:
```
final_drop.head(10)
```

[202]:
```
     ZIP  average  number_crimes   race  Join_Count case_numbe  \
0  92113    58010              4  55813           1   16028219
```

```
1  92113    58010              4  55813           1   18005199
2  92113    58010              4  55813           1   19006514
3  92113    58010              4  55813           1   19019302
4  92105    63949              7  71446           1   16024587
5  92105    63949              7  71446           1   16046634
6  92105    63949              7  71446           1   17015573
7  92105    63949              7  71446           1   18003081
8  92105    63949              7  71446           1   18035021
9  92105    63949              7  71446           1   18037683

                time      crime              weapon  \
0                Day    robbery               knife
1  Evening and Night    assault               hands
2                Day    assault               stick
3  Evening and Night  vandalism         spray paint
4                Day    assault                 rod
5  Evening and Night    assault        pepper spray
6                Day   burglary         spray paint
7  Evening and Night  vandalism                rock
8                Day    assault  hands, fists, feet
9  Evening and Night     threat              verbal

                              address   latitude   longitude    SHAPE_STAr  \
0        3500 main st San Diego, CA  32.687809 -117.118594  1.399861e+08
1       3700 birch st San Diego, CA  32.689123 -117.114540  1.399861e+08
2       2000 logan av San Diego, CA  32.700325 -117.142498  1.399861e+08
3        2200 main st San Diego, CA  32.695622 -117.141571  1.399861e+08
4   4100 fairmount av San Diego, CA  32.751621 -117.100905  1.533141e+08
5      3900 landis st San Diego, CA  32.745998 -117.110579  1.533141e+08
6         5400 lea st San Diego, CA  32.745911 -117.079460  1.533141e+08
7    3800 winona ave San Diego, CA  32.747130 -117.088068  1.533141e+08
8  5000 university av San Diego, CA  32.749398 -117.086897  1.533141e+08
9        3800 43rd st San Diego, CA  32.747985 -117.102362  1.533141e+08

     SHAPE_STLe   Shape__Area  Shape__Length  \
0  65765.279016  1.841453e+07   23834.868005
1  65765.279016  1.841453e+07   23834.868005
2  65765.279016  1.841453e+07   23834.868005
3  65765.279016  1.841453e+07   23834.868005
4  65485.481120  2.018697e+07   23765.440339
5  65485.481120  2.018697e+07   23765.440339
6  65485.481120  2.018697e+07   23765.440339
7  65485.481120  2.018697e+07   23765.440339
8  65485.481120  2.018697e+07   23765.440339
9  65485.481120  2.018697e+07   23765.440339

                                                 SHAPE
```

```
0  {"x": -117.11859403, "y": 32.68780928, "spatia…
1  {"x": -117.11453952, "y": 32.68912298, "spatia…
2  {"x": -117.14249803, "y": 32.70032541, "spatia…
3  {"x": -117.14157103, "y": 32.69562156, "spatia…
4  {"x": -117.10090548, "y": 32.751621, "spatialR…
5  {"x": -117.11057945, "y": 32.745998, "spatialR…
6  {"x": -117.07946038, "y": 32.74591112, "spatia…
7  {"x": -117.08806823, "y": 32.74712958, "spatia…
8  {"x": -117.08689704, "y": 32.7493978, "spatial…
9  {"x": -117.1023615, "y": 32.747985, "spatialRe…
```

[203]:
```
#zipcode_extent = cd.set_index('ZIP').to_dict()['SHAPE']
```

[204]:
```
# final_drop_fl = final_drop.spatial.to_featurelayer(title='SAUpdate',tags =␣
↪'hatecrime')
# final_drop_fl.share(org=True)
```

[ ]:
```

```

[205]:
```
fl_drop = gis.content.get('5253ac81624f4fe7a6d48b1e5e1c6a06')
```

[206]:
```
fl_drop = fl_drop.layers[0]
```

[207]:
```
map7 = gis.map("San Diego County, US")
map7.add_layer(fl_drop)
```

[208]:
```
map7
```

```
MapView(layout=Layout(height='400px', width='100%'))
```

```
<IPython.core.display.HTML object>
```

[209]:
```
from arcgis.features.analysis import calculate_density
```

[210]:
```
final.head()
```

[210]:
```
     ZIP  average  number_crimes   race  OBJECTID  Join_Count case_numbe  \
0  92113    58010              4  55813        24           1   16028219
1  92113    58010              4  55813        94           1   18005199
2  92113    58010              4  55813       135           1   19006514
3  92113    58010              4  55813       144           1   19019302
4  92105    63949              7  71446        20           1   16024587

                time crime_code    crime  …      weapon  \
0                Day        211  robbery  …       knife
1  Evening and Night      242:m  assault  …       hands
```

```
2                Day       245a1     assault  …        stick
3  Evening and Night  594(b)(1)  vandalism  …  spray paint
4                Day       245a1     assault  …          rod

                         address   latitude   longitude  COMMUNITY  \
0       3500 main st San Diego, CA  32.687809 -117.118594  San Diego
1      3700 birch st San Diego, CA  32.689123 -117.114540  San Diego
2      2000 logan av San Diego, CA  32.700325 -117.142498  San Diego
3       2200 main st San Diego, CA  32.695622 -117.141571  San Diego
4  4100 fairmount av San Diego, CA  32.751621 -117.100905  San Diego

     SHAPE_STAr    SHAPE_STLe   Shape__Area Shape__Length  \
0  1.399861e+08  65765.279016  1.841453e+07  23834.868005
1  1.399861e+08  65765.279016  1.841453e+07  23834.868005
2  1.399861e+08  65765.279016  1.841453e+07  23834.868005
3  1.399861e+08  65765.279016  1.841453e+07  23834.868005
4  1.533141e+08  65485.481120  2.018697e+07  23765.440339

                                               SHAPE
0  {'x': -13037582.2496, 'y': 3853938.8038000017,…
1  {'x': -13037130.9043, 'y': 3854112.5650999993,…
2  {'x': -13040243.2312, 'y': 3855594.392999999, …
3  {'x': -13040140.0375, 'y': 3854972.157499999, …
4  {'x': -13035613.1699, 'y': 3862382.0216000006,…

[5 rows x 24 columns]
```

[211]:
```
hate_info_sdf.head(10)
```

[211]:
```
   case_number                     time    crime_code               crime  \
0     16000456  Early Morning/Late Night       243(d)m             assault
1     16001278  Early Morning/Late Night         245a1             assault
2     16004522  Early Morning/Late Night      594(b)(4)           vandalism
3     16005962         Evening and Night  422.22(a)(4)  threat, phone call
4     16005900  Early Morning/Late Night       417a1:m              threat
5     16006866         Evening and Night         422.6           vandalism
6     16007286         Evening and Night         422.6              threat
7     16008581         Evening and Night         422.6           vandalism
8     16008340         Evening and Night        422.6a             assault
9     16008751  Early Morning/Late Night       417a1:m              threat

           beat block      street type              weapon  \
0       gaslamp   500           g   st    hands, fists, feet
1    north park  3400        30th   st               stick
2  east village  1400    imperial   av              marker
3      bay park  4100         ute   dr               phone
4     hillcrest   100  university   av               knife
```

```
5    el cerrito  5800  university   av                 paint
6     del cerro  6200       capri   dr                 phone
7  college west  5400     gilbert   dr  unknown sharp object
8  lincoln park   500      euclid   av     hands, fists, feet
9  east village   300        park   bl                 knife

                         address   latitude   longitude  \
0            500 g st San Diego, CA  32.712638 -117.160073
1         3400 30th st San Diego, CA  32.741139 -117.130148
2     1400 imperial av San Diego, CA  32.706347 -117.151812
3          4100 ute dr San Diego, CA  32.807511 -117.203142
4    100 university av San Diego, CA  32.748341 -117.163831
5   5800 university av San Diego, CA  32.749225 -117.072844
6        6200 capri dr San Diego, CA  32.782135 -117.065392
7      5400 gilbert dr San Diego, CA  32.760023 -117.078470
8        500 euclid av San Diego, CA  32.710521 -117.085090
9          300 park bl San Diego, CA  32.708696 -117.153825

                                             SHAPE
0  {"x": -117.16007340369976, "y": 32.71263809630…
1  {"x": -117.13014779569856, "y": 32.74113929569…
2  {"x": -117.15181181069018, "y": 32.70634749036…
3  {"x": -117.20314241727988, "y": 32.80751149470…
4  {"x": -117.16383087642083, "y": 32.74834112357…
5  {"x": -117.07284360260134, "y": 32.74922546522…
6  {"x": -117.06539223750941, "y": 32.78213501005…
7  {"x": -117.0784703674847, "y": 32.760023132515…
8  {"x": -117.08509006621132, "y": 32.71052051777…
9  {"x": -117.1538254202943, "y": 32.708695648113…
```

[212]:
```
crime_by_tract
```

[212]:
```
<Item title:"Total Crime Index by Census Tract  2016 San Diego County"
type:Feature Layer Collection owner:Rizbee>
```

[213]:
```
len(crime_data)
```

[213]:
```
61603
```

[214]:
```
crime_data[crime_data['ZipCode']==18072]
```

[214]:
```
Empty DataFrame
Columns: [CM_LEGEND, agency, Charge_Description_Orig, activityDate,
BLOCK_ADDRESS, ZipCode, community]
Index: []
```

[215]:
```
crime_data_sd = crime_data[crime_data['ZipCode'] >=90000.0]
```

[216]:
```
len(crime_data_sd)
```

[216]: 61188

[217]:
```
zip_sd = crime_data_sd[['ZipCode']]
```

[218]:
```
zip_sd_crime = zip_sd.groupby('ZipCode').count()
```

[219]:
```
zip_count = crime_data_sd.groupby('ZipCode').ZipCode.agg('count').
↪to_frame("count").reset_index()
```

[220]:
```
zip_count.head()
```

[220]:
```
   ZipCode  count
0  90038.0      1
1  90068.0      1
2  90069.0      1
3  91066.0      1
4  91355.0      1
```

[221]:
```
final_drop.head()
```

[221]:
```
     ZIP  average  number_crimes   race  Join_Count case_numbe  \
0  92113    58010              4  55813           1   16028219
1  92113    58010              4  55813           1   18005199
2  92113    58010              4  55813           1   19006514
3  92113    58010              4  55813           1   19019302
4  92105    63949              7  71446           1   16024587

                time      crime       weapon                         address  \
0                Day    robbery        knife      3500 main st San Diego, CA
1  Evening and Night    assault        hands     3700 birch st San Diego, CA
2                Day    assault        stick     2000 logan av San Diego, CA
3  Evening and Night  vandalism  spray paint      2200 main st San Diego, CA
4                Day    assault          rod  4100 fairmount av San Diego, CA

    latitude   longitude    SHAPE_STAr    SHAPE_STLe   Shape__Area  \
0  32.687809 -117.118594  1.399861e+08  65765.279016  1.841453e+07
1  32.689123 -117.114540  1.399861e+08  65765.279016  1.841453e+07
2  32.700325 -117.142498  1.399861e+08  65765.279016  1.841453e+07
3  32.695622 -117.141571  1.399861e+08  65765.279016  1.841453e+07
4  32.751621 -117.100905  1.533141e+08  65485.481120  2.018697e+07

   Shape__Length                                              SHAPE
0   23834.868005  {"x": -117.11859403, "y": 32.68780928, "spatia…
1   23834.868005  {"x": -117.11453952, "y": 32.68912298, "spatia…
2   23834.868005  {"x": -117.14249803, "y": 32.70032541, "spatia…
```

```
3   23834.868005  {"x": -117.14157103, "y": 32.69562156, "spatia…
4   23765.440339  {"x": -117.10090548, "y": 32.751621, "spatialR…
```

[222]:
```python
zip_sdf_lst = list(final_drop['ZIP'])
```

[223]:
```python
case= zip_count.where(zip_count['ZipCode'].isin(zip_sdf_lst)).reset_index()
```

[224]:
```python
case = case.set_index('index')

case = case.rename(columns = {'count':'All_Crime_Count'})
```

[225]:
```python
case= case.dropna(how='all')
```

[226]:
```python
case = case.reset_index().drop(columns = ['index'])
```

[227]:
```python
all_crime_merge = case.merge(final_drop, left_on='ZipCode', right_on='ZIP')
```

[228]:
```python
all_crime_merge = all_crime_merge.drop(columns = ['ZIP'])
```

17 Final Table

17.0.1 Below is our final table after a lot of analysis and spatial work. We now have each zipcode with how many crimes in total that ocurred there, coupled with the number of hate crimes that occur there. We also have the average income in the average column as well as the population of base race for each of the zipcodes from the hatecrime table.

[229]:
```python
all_crime_merge.sort_values(by='All_Crime_Count',ascending = False).head(10)
```

[229]:
```
    ZipCode  All_Crime_Count  average  number_crimes   race  Join_Count  \
19  92101.0           5225.0    95321             20  45988           1
22  92101.0           5225.0    95321             20  45988           1
20  92101.0           5225.0    95321             20  45988           1
23  92101.0           5225.0    95321             20  45988           1
18  92101.0           5225.0    95321             20  45988           1
17  92101.0           5225.0    95321             20  45988           1
16  92101.0           5225.0    95321             20  45988           1
15  92101.0           5225.0    95321             20  45988           1
14  92101.0           5225.0    95321             20  45988           1
13  92101.0           5225.0    95321             20  45988           1

   case_numbe                     time               crime  \
19   17016020  Early Morning/Late Night             assault
22   17025240         Evening and Night             assault
20   17019750         Evening and Night  threat, phone call
```

```
23   17043828  Early Morning/Late Night             assault
18   17006219                       Day             assault
17   17005240         Evening and Night             assault
16   16052116                       Day             assault
15   16008751  Early Morning/Late Night              threat
14   16004522  Early Morning/Late Night           vandalism
13   16000456  Early Morning/Late Night             assault

              weapon                             address   latitude  \
19              cane              1100 a st San Diego, CA  32.718932
22           vehicle          1400 4th  ave San Diego, CA  32.719943
20             phone               700 a st San Diego, CA  32.718929
23              lock            900 park bl San Diego, CA  32.714760
18              cane            200 park bl San Diego, CA  32.707529
17  hands, fists, feet            1200 k  st San Diego, CA  32.708427
16  hands, fists, feet  2200 morley field dr San Diego, CA  32.739898
15             knife            300 park bl San Diego, CA  32.708696
14            marker      1400 imperial av San Diego, CA  32.706347
13  hands, fists, feet              500 g st San Diego, CA  32.712638

     longitude    SHAPE_STAr    SHAPE_STLe   Shape__Area  Shape__Length  \
19 -117.154663  2.548928e+08  98792.532847  3.354889e+07   35841.688407
22 -117.161191  2.548928e+08  98792.532847  3.354889e+07   35841.688407
20 -117.158273  2.548928e+08  98792.532847  3.354889e+07   35841.688407
23 -117.153818  2.548928e+08  98792.532847  3.354889e+07   35841.688407
18 -117.154899  2.548928e+08  98792.532847  3.354889e+07   35841.688407
17 -117.153291  2.548928e+08  98792.532847  3.354889e+07   35841.688407
16 -117.142708  2.548928e+08  98792.532847  3.354889e+07   35841.688407
15 -117.153825  2.548928e+08  98792.532847  3.354889e+07   35841.688407
14 -117.151812  2.548928e+08  98792.532847  3.354889e+07   35841.688407
13 -117.160073  2.548928e+08  98792.532847  3.354889e+07   35841.688407

                                                SHAPE
19  {'x': -117.15466253, 'y': 32.7189316, 'spatial…
22  {'x': -117.16119055, 'y': 32.7199425, 'spatial…
20  {'x': -117.1582734, 'y': 32.7189291, 'spatialR…
23  {'x': -117.15381829, 'y': 32.7147598, 'spatial…
18  {'x': -117.15489919, 'y': 32.70752876, 'spatia…
17  {'x': -117.15329081, 'y': 32.70842713, 'spatia…
16  {'x': -117.14270809, 'y': 32.73989832, 'spatia…
15  {'x': -117.15382542, 'y': 32.70869565, 'spatia…
14  {'x': -117.15181181, 'y': 32.70634749, 'spatia…
13  {'x': -117.1600734, 'y': 32.7126381, 'spatialR…
```

[230]:
```
all_crime_sdf = pd.DataFrame.spatial.from_xy(all_crime_merge,x_column =␣
↪'longitude', y_column = 'latitude')
```

[231]:
```
# all_crime_fl = all_crime_sdf.spatial.to_featurelayer(title='SAAllCrime',tags␣
↪= 'hatecrime')
# all_crime_fl.share(org=True)
```

[232]:
```
all_crime_fl = gis.content.get('ffac1ebceca24795ad7ac0bcec05c130')
```

[241]:
```
# cal_density = calculate_density(input_layer =␣
↪all_crime_fl,bounding_polygon_layer = zip_codes,classification_type =␣
↪'GeometricInterval'
#                                ,output_name = 'Hate_crimePJ_zip')
# cal_density
```

[241]:
```
<Item title:"Hate_crimePJ_zip" type:Feature Layer Collection
owner:pjuneja_ucsd5>
```

[242]:
```
# cal_density.share(org = True)
```

[242]:
```
{'results': [{'itemId': 'df6e6e71ad184634a255695a2761cb04',
   'success': True,
   'notSharedWith': []}]}
```

[243]:
```
cal_density = gis.content.get('df6e6e71ad184634a255695a2761cb04')
```

[244]:
```
cal_density = cal_density.layers[0]
```

[245]:
```
map8 = gis.map('San Diego County, CA')
map8.add_layer(cal_density)
map8
```

```
MapView(layout=Layout(height='400px', width='100%'))
```

```
<IPython.core.display.HTML object>
```

18 SUMMARY OF PRODUCTS AND RESULTS

Based on the density chart above, we see one really high ontensity area which is high in normal crime as well as hate crime. This is hatecrime area 92101. Based on our hypothesis we decided to do a lot more analysis and got a bunch of maps that we went on to include in our presentation but for our project in order to take this further, we need to atake into account whatever we learnt from our spatial data adn put it to the test by comparing our model values throughout and making scatter plots as shown below.

Below Is our analysis for what we believe to be related to our hypothesis. All feature layers have been Uploaded to ArcGis Online. These are the scatter plots that we believe to be relevant to our project.

19 Why GIS?

The difference between our project and something that could be done just with pandas is, we needed gis to give us a lot of the information we couldn't have just stumbled across online. We managed to get a lot of maps from online which led to us starting from just a plain csv without any knowledge of the data. GIS took a normal hate crime project to the next level by allowing us to tell where the crime occured, the area of which it occured and information about that area which helped us conduct a lot of analysis. What follows below is a spatial analysis and a a few stats about what followed.

[284]:
```
ab = gis.content.get('fa79371b9a7c497f8af5d2057bc7a5b8')
ab
```

[284]:
```
<Item title:"Heat Map of all crimes and Hate Crimes" type:Web Map
owner:pjuneja_UCSDOnline3>
```

This is a heat map of all top hate crimes and crimes and we seea bright red spot. This is the zipcode 92101 and e included this in our presentation. This area has an average income of about 100k withn80% of its population being all white people. This fits into our hypothesis of observing hate crimes.

[283]:
```
from arcgis.mapping import WebMap
```

[285]:
```
sd_hate = WebMap(ab)
```

[286]:
```
sd_hate
```

```
MapView(hide_mode_switch=True, layout=Layout(height='400px', width='100%'))
```

```
<IPython.core.display.HTML object>
```

[281]:
```

```

[287]:
```
hate_crime_by_race = gis.content.get('ca95251d1d2e44c3b74ec73f589887af')
```

[288]:
```
hate_crime_by_race = WebMap(hate_crime_by_race)
```

In this map the size of the circle represents base population by race. Now I know in a prior discussion we mentioned that race is a number a not a percentage. However when we look at the below map we need to think of the size of a circle as a measure of diversity. THe smaller the circle the more diverse the area. I think we can telll from the mpa below that areas that are more diverse seem to be lighter, that is have lesser hate crime tahn areas with areas with a low diveristy that is the bugger circles whihc are less diverse and have a higher base case.

[292]:
```
hate_crime_by_race.legend = True
hate_crime_by_race
```

```
MapView(hide_mode_switch=True, layout=Layout(height='400px', width='100%'))
```

```
&lt;IPython.core.display.HTML object&gt;
```

[294]:
```
hate_crime_by_income = gis.content.get('863b45cf51cc450daea88af618919b7e')
```

[295]:
```
hate_crime_by_income = WebMap(hate_crime_by_income)
```

The map below represents the average income per family in that area with larger the circle larger the income. The shade represents number of hate crimes in the area. To understand this map, areas with higher income seem to have a mixed number of hate crimes

[296]:
```
hate_crime_by_income
```

```
MapView(hide_mode_switch=True, layout=Layout(height='400px', width='100%'))
```

```
<IPython.core.display.HTML object>
```

[246]:
```
all_crime_92101 = all_crime_sdf[all_crime_sdf['ZipCode']==92101.0]
```

[261]:
```
all_crime_92101.head()
```

[261]:
```
    ZipCode  All_Crime_Count  average  number_crimes   race  Join_Count  \
13  92101.0           5225.0    95321             20  45988           1
14  92101.0           5225.0    95321             20  45988           1
15  92101.0           5225.0    95321             20  45988           1
16  92101.0           5225.0    95321             20  45988           1
17  92101.0           5225.0    95321             20  45988           1

   case_numbe                    time      crime              weapon  \
13   16000456  Early Morning/Late Night    assault  hands, fists, feet
14   16004522  Early Morning/Late Night  vandalism              marker
15   16008751  Early Morning/Late Night     threat               knife
16   16052116                       Day    assault  hands, fists, feet
17   17005240         Evening and Night    assault  hands, fists, feet

                              address   latitude   longitude    SHAPE_STAr  \
13              500 g st San Diego, CA  32.712638 -117.160073  2.548928e+08
14       1400 imperial av San Diego, CA  32.706347 -117.151812  2.548928e+08
15            300 park bl San Diego, CA  32.708696 -117.153825  2.548928e+08
16  2200 morley field dr San Diego, CA  32.739898 -117.142708  2.548928e+08
17             1200 k  st San Diego, CA  32.708427 -117.153291  2.548928e+08

     SHAPE_STLe   Shape__Area  Shape__Length  \
13  98792.532847  3.354889e+07   35841.688407
```

```
14  98792.532847  3.354889e+07   35841.688407
15  98792.532847  3.354889e+07   35841.688407
16  98792.532847  3.354889e+07   35841.688407
17  98792.532847  3.354889e+07   35841.688407

                                                    SHAPE
13  {"x": -117.1600734, "y": 32.7126381, "spatialR…
14  {"x": -117.15181181, "y": 32.70634749, "spatia…
15  {"x": -117.15382542, "y": 32.70869565, "spatia…
16  {"x": -117.14270809, "y": 32.73989832, "spatia…
17  {"x": -117.15329081, "y": 32.70842713, "spatia…
```

[262]:
```
#all_crime_92101_fl = all_crime_92101.spatial.
↪to_featurelayer(title='921AllCrime1',tags = 'hatecrime')
#all_crime_92101_fl.share(org=True)
```

[249]:
```
all_crime_merge.columns
```

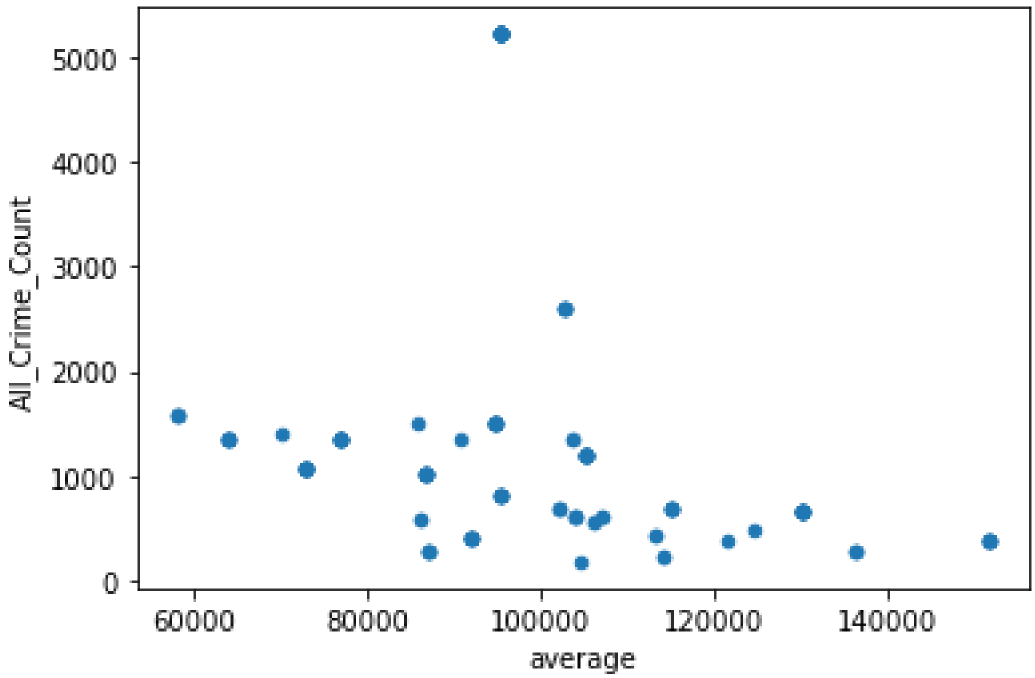
[249]:
```
Index(['ZipCode', 'All_Crime_Count', 'average', 'number_crimes', 'race',
       'Join_Count', 'case_numbe', 'time', 'crime', 'weapon', 'address',
       'latitude', 'longitude', 'SHAPE_STAr', 'SHAPE_STLe', 'Shape__Area',
       'Shape__Length', 'SHAPE'],
      dtype='object')
```

[250]:
```
#plot of avg income/all crimes
#avg/hatecrimes

#RaceBase vs AllCrime
#Raceabase vs HateCrime
```

From our first graph of All crimes vs average income we can see that areas which have a higher income have a lower crime rate which is something that makes sense because better and higher income areas would have a lesser crime rate.

[251]:
```
all_crime_merge.plot.scatter(x='average', y='All_Crime_Count')
```

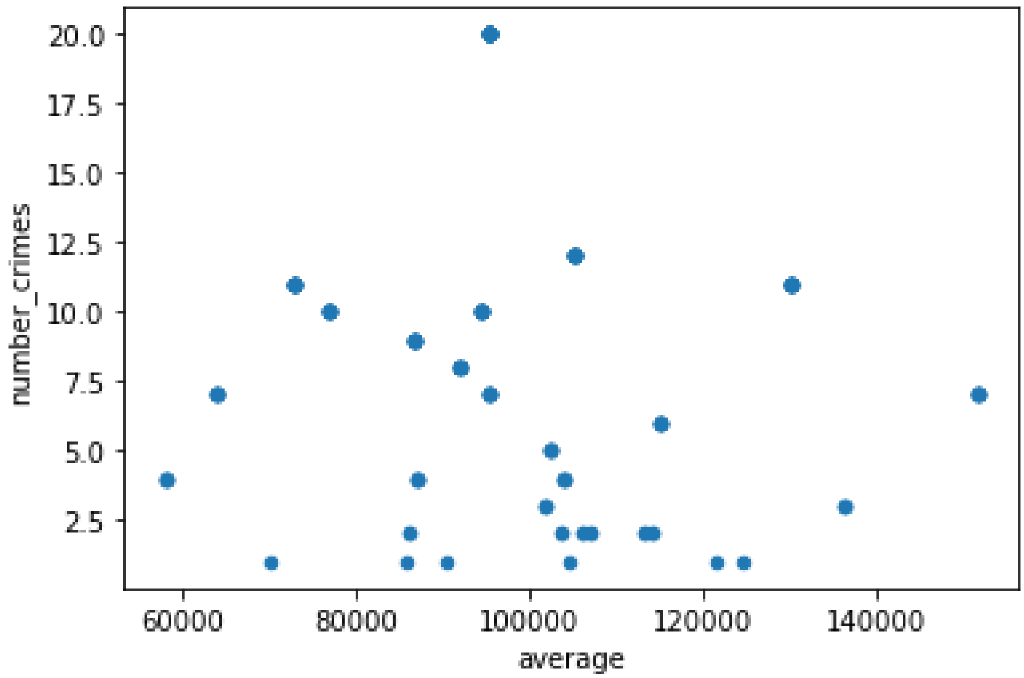
[251]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc2bd270eb8>
```



Now if we look at the number of hate crimes vs average income we can see this there is no specific trend for one possibly because of the amount of data we had. But whats interesting is that we can see how a place with an average income of about 100k has a lot of hate crime. This is primarily in our opinion, due to a higher base race of population present htere giving a sense of entitilment which may affect this.

[252]:
```
all_crime_merge.plot.scatter(x='average', y='number_crimes')
```

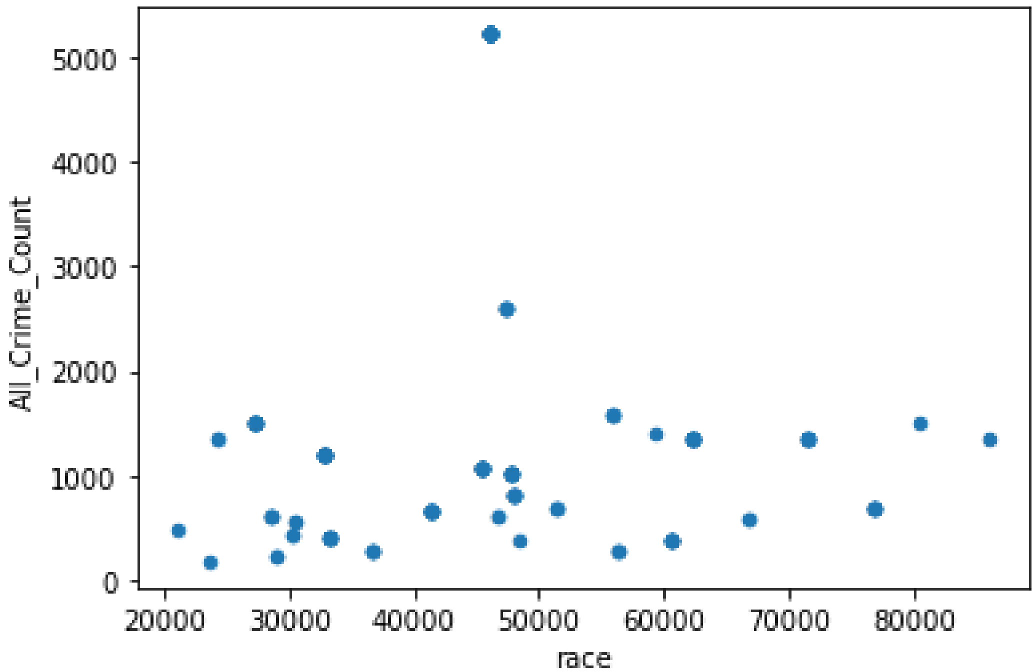
[252]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc2bd237dd8>



This cell below refers to crime depending on base rase of the population in that area. What we need to to understand and this will be mentioned in the considerations as well, base race is a number and not a percentage which could have affected our values. We don'thave a clear trend but can see a slight increase which is enough to point out to us that as there are more numbers of a base race living in an area, minorities seeem to be targetted more and hence have a higher number of hatecrimes

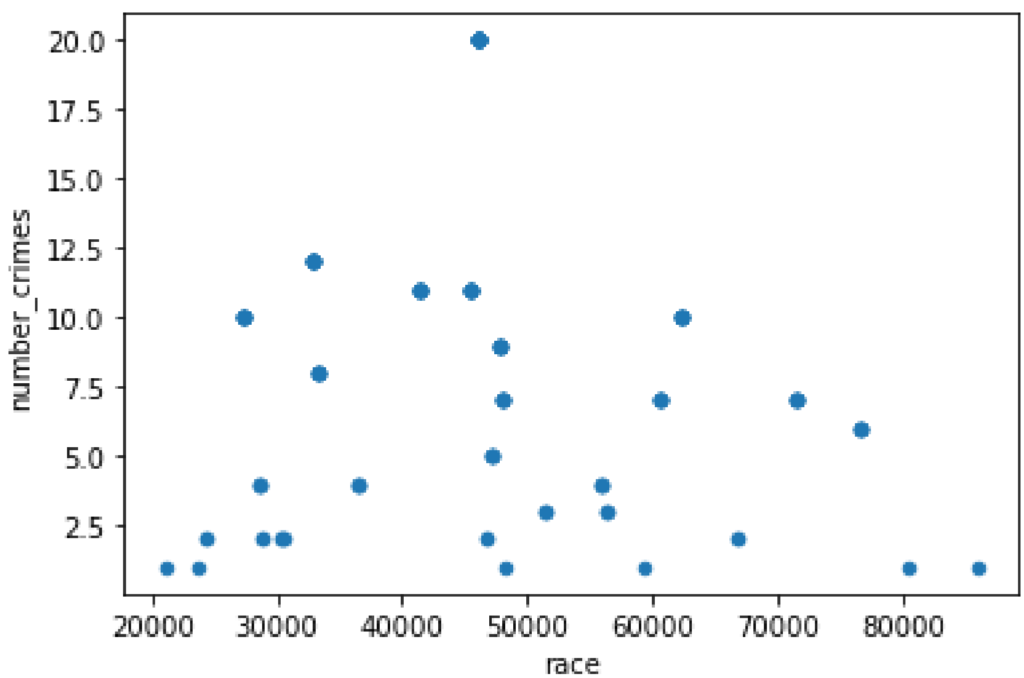
[253]:
```
all_crime_merge.plot.scatter(x='race', y='All_Crime_Count')
```

[253]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc369454a58>
```




```
[254]: all_crime_merge.plot.scatter(x='race', y='number_crimes')
```

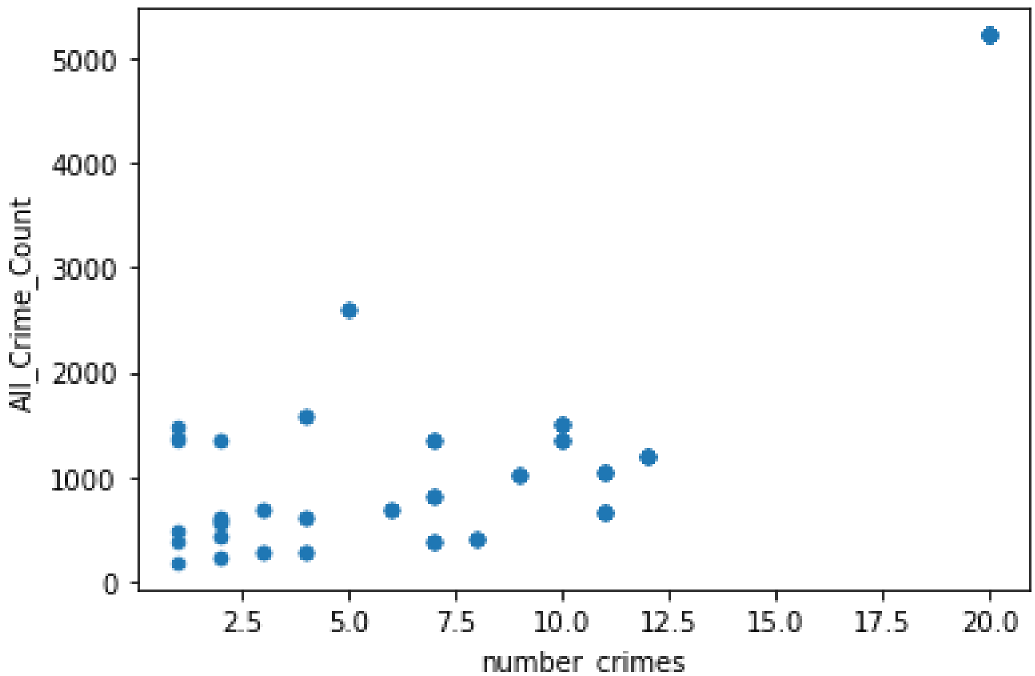
[254]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc36938ca58>



This scatter plot shows us that areas with a higher number of hatecrimes also seem to have a higher number of crimes.

[255]:
```
all_crime_merge.plot.scatter(x='number_crimes', y='All_Crime_Count')
```

[255]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc3692ae6a0>
```



20 Discussion

DISCUSSION about CAVEATS The main caveats of this project was the amount of data we had. We had a 160 reports. The major thing worth pointing out that peple don't actually report hate crimes. there are only a handful of reports. This was tough as it did not help us as there wasn't suffecient data. Another issue that we had to keep dealing with was after geoenrichment the base race population values that we got turned out to be a number and not a percentage ad hence did not leave enough room to understand our data in context of what was wrong.

While working on this dataset, a common question my partner and I realised needed to be answered is who are hate crimes committed against? and who commits the? The simple answer if we ask anyone it's always the race in the minority. Tradiotnally white people can also be victims of hate crimes and to test this out and to add to our discussion we did this below.

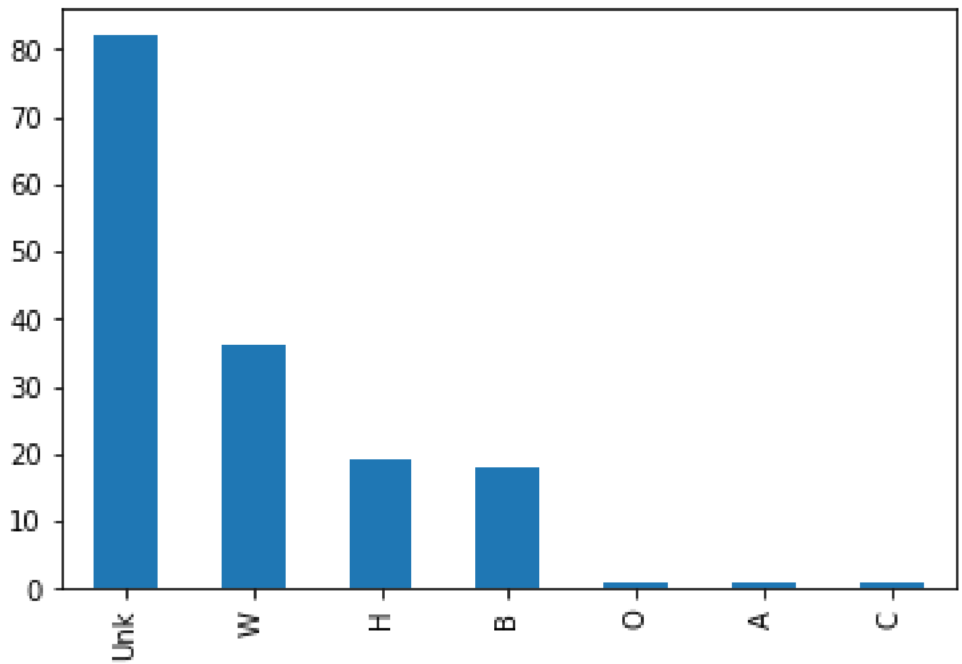
[257]:
```
hate_info2= hate_crime[['case_number','number_of_suspects', 'suspect',␣
↪'victim_count',
```

```
        'victim_other', 'injury', 'suspect_race_0', 'suspect_race_1',
        'suspect_race_2', 'suspect_sex_0', 'suspect_sex_1', 'suspect_sex_2',
        'victim_race_0', 'victim_race_1', 'victim_race_2', 'victim_sex_0',
        'victim_sex_1', 'victim_sex_2']]
```

[297]:
```
#hate_info2
```

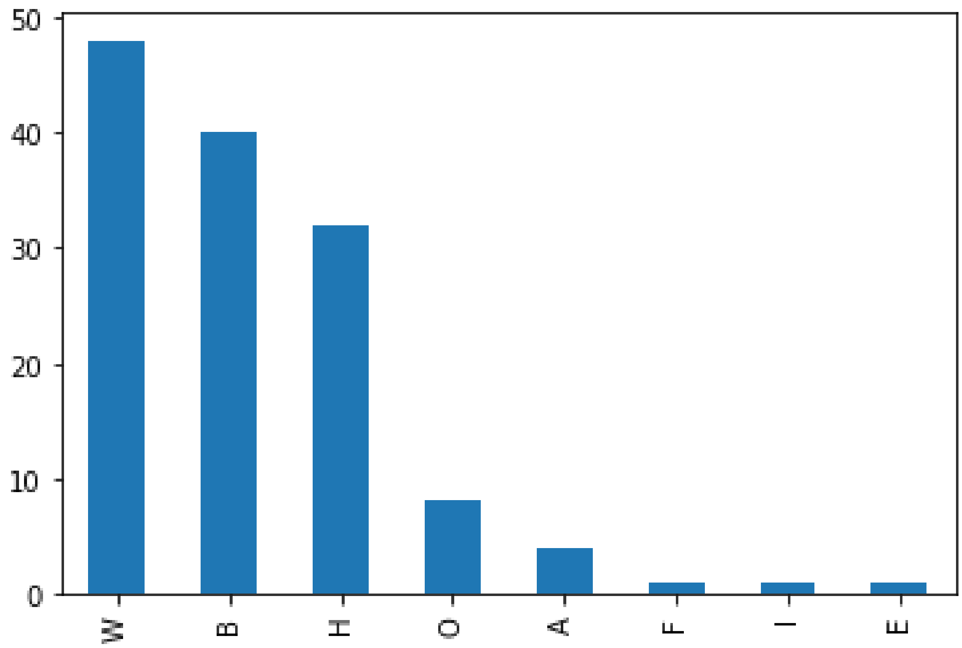
[259]:
```
hate_info2['suspect_race_0'].value_counts().plot.bar()
```

[259]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc362a15080>



[260]:
```
hate_info2['victim_race_0'].value_counts().plot.bar()
```

[260]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc2b6d82860>



As we can white people have the highest number in both which is very interesting. Throughout our project and through our analysis I think it was safe to assume thateven though our hypothesis made sense with the overarching theme of the project,which was predicting where hate crime would be the highest and I think we have had succesful findings that can be used by tthe city planner etc to ensure these areas are safer for all members od the community.

Discussion in relation to readings Our initial literature showed us the state of hate crime in San Diego and not much about the topic itself. Based on our understanding we decided to investigate base rce, income and the total number of crimes in relation to what we believed would be be suitable for analysis. The readings did not have anything concrete in terms of analysis for us to do hence this projct was more of an exploratory measure into a problem we believed needed tackling here in San Diego. For example a couple of the zipcodes we discovered such as 92101, was very high in cime and hate crime even though that area seemed like a well to do area with a high mean income. These areas could have learning centers etc built so people can respect the area and the cultures of the poeple that live there and ensure these crimes do not occur. An increase in patrolling routes in these areas would also make these areas a lot safer.

21 Conclusion and Future work

While our hypothesis seems to have been proved in its own way correct, there is a lot of scope for this project. These could act as recommendations for the city of San Diego which involve learning centers in areas with high rates of hate crime and crime. Another solution mentioned above would be increasing police stations and adding more patrolling routes in this area to help reduce these crime rates, and ensure a more safe environment for all the communities that are there

Future Scope of this project involves creating geospatial statistics for analysis. I would have really liked doing network analysis to to create a mesh of all hate crime occurences and identify good patrolling routes, locations for learning centers and creation of police precincts. Also once we get a hugher number of reports, we can create a model to predict which crimesare hatecrimes on a larger crime database as these usually dont get reported seperately. This will increase the number of records we have and create more avenues for analysis

In conclusion our hypothesis seemed to have worked and above is how we documented our project accordingly.

```
[ ]:
```