Midterm (CLEANUP)

December 7, 2020

MIDTERM (Cleanup) + Function

Alejandro Gonzalez + Mike Van Gorder UP206

Table of Contents

1 Research Question

Housing density bills keep failing at the state level, yet the housing pressure continues to mount. Do policies to increase density in Los Angeles affect single family neighborhoods differently?

By exploring accessory dwelling unit data, can we analyze change at the local level?

</div>

2 Data Sources

Utilizing American Community Service Data from Social Explorer – Comparing 2013 and 2018

Filtering ADU Data from LA Data Portal–Department of Building and Safety – exploring Zip Codes 90008 and 90039

2.1 Data Exploration

In this section we will use Social Explorer to pull data from the two neighborhoods: Atwater Village and Leimert Park.

[1]:
```
## Will start with importing data
import pandas as pd
```

[2]:
```python
# We will begin by loading a data file for Atwater Village that details the␣
↪structures built in the area.
# note the relative filepath! where is this file located?
df = pd.read_csv('data/Atwater2017_HHIncomeCategories.csv')
```

[3]:
```python
df.Geo_FIPS.head()
```

[3]:
```
0    60371863011
1    60371863012
2    60371863013
3    60371863021
4    60371864011
Name: Geo_FIPS, dtype: int64
```

[4]:
```python
df = pd.read_csv(
    'data/Atwater2017_HHIncomeCategories.csv',
    dtype=
    {
        'Geo_FIPS':str,
        'Geo_STATE':str,
        'Geo_COUNTY': str
    }
)
```

[5]:
```python
##Now we are importing a data set corresponding to Leimert Park.
df2 = pd.read_csv('data/Leimert2017_HHIncomeCategories.csv')
```

[6]:
```python
df2.Geo_FIPS.head()
```

[6]:
```
0    60372190203
1    60372340001
2    60372340002
3    60372340003
4    60372340004
Name: Geo_FIPS, dtype: int64
```

[7]:
```python
df2 = pd.read_csv(
    'data/Leimert2017_HHIncomeCategories.csv',
    dtype=
    {
        'Geo_FIPS':str,
        'Geo_STATE':str,
        'Geo_COUNTY': str
    }
)
```

2.2 Data Analysis

[8]:
```
df.columns[df.isna().all()].tolist()
```

[8]:
```
['Geo_US',
 'Geo_REGION',
 'Geo_DIVISION',
 'Geo_STATECE',
 'Geo_COUSUB',
 'Geo_PLACE',
 'Geo_PLACESE',
 'Geo_CONCIT',
 'Geo_AIANHH',
 'Geo_AIANHHFP',
 'Geo_AIHHTLI',
 'Geo_AITSCE',
 'Geo_AITS',
 'Geo_ANRC',
 'Geo_CBSA',
 'Geo_CSA',
 'Geo_METDIV',
 'Geo_MACC',
 'Geo_MEMI',
 'Geo_NECTA',
 'Geo_CNECTA',
 'Geo_NECTADIV',
 'Geo_UA',
 'Geo_UACP',
 'Geo_CDCURR',
 'Geo_SLDU',
 'Geo_SLDL',
 'Geo_VTD',
 'Geo_ZCTA3',
 'Geo_ZCTA5',
 'Geo_SUBMCD',
 'Geo_SDELM',
 'Geo_SDSEC',
 'Geo_SDUNI',
 'Geo_UR',
 'Geo_PCI',
 'Geo_TAZ',
 'Geo_UGA',
 'Geo_BTTR',
 'Geo_BTBG',
 'Geo_PUMA5',
 'Geo_PUMA1']
```

[9]:
```python
df = df.dropna(axis=1,how="all")
```

[10]:
```python
df.head()
```

[10]:
```
      Geo_FIPS            Geo_GEOID  \
0  060371863011  15000US060371863011
1  060371863012  15000US060371863012
2  060371863013  15000US060371863013
3  060371863021  15000US060371863021
4  060371864011  15000US060371864011

                                            Geo_NAME  \
0  Block Group 1, Census Tract 1863.01, Los Angel…
1  Block Group 2, Census Tract 1863.01, Los Angel…
2  Block Group 3, Census Tract 1863.01, Los Angel…
3  Block Group 1, Census Tract 1863.02, Los Angel…
4  Block Group 1, Census Tract 1864.01, Los Angel…

                                           Geo_QName Geo_STUSAB  Geo_SUMLEV  \
0  Block Group 1, Census Tract 1863.01, Los Angel…         ca         150
1  Block Group 2, Census Tract 1863.01, Los Angel…         ca         150
2  Block Group 3, Census Tract 1863.01, Los Angel…         ca         150
3  Block Group 1, Census Tract 1863.02, Los Angel…         ca         150
4  Block Group 1, Census Tract 1864.01, Los Angel…         ca         150

  Geo_GEOCOMP Geo_FILEID  Geo_LOGRECNO Geo_STATE Geo_COUNTY  Geo_TRACT  \
0           0      ACSSF         15232        06        037     186301
1           0      ACSSF         15233        06        037     186301
2           0      ACSSF         15234        06        037     186301
3           0      ACSSF         15235        06        037     186302
4           0      ACSSF         15238        06        037     186401

  Geo_BLKGRP  SE_B14001_001  SE_B14001_002  SE_B14001_003 SE_B14001_004  \
0          1            330             46             67            54
1          2            327             95            148            31
2          3            268             73            129            44
3          1            584            129            115           109
4          1            675            199            167           172

  SE_B14001_005  SE_B14001_006
0            50            113
1             9             44
2            10             12
3            71            160
4            69             68
```

[11]:
```python
# list of additional columns to drop
columns_to_drop =␣
↪['Geo_GEOID','Geo_STUSAB','Geo_SUMLEV','Geo_GEOCOMP','Geo_FILEID','Geo_LOGRECNO']
```

[12]:
```python
# next, drop them!
df = df.drop(columns_to_drop,axis=1)
df.head()
```

[12]:
```
       Geo_FIPS                                           Geo_NAME  \
0  060371863011  Block Group 1, Census Tract 1863.01, Los Angel…
1  060371863012  Block Group 2, Census Tract 1863.01, Los Angel…
2  060371863013  Block Group 3, Census Tract 1863.01, Los Angel…
3  060371863021  Block Group 1, Census Tract 1863.02, Los Angel…
4  060371864011  Block Group 1, Census Tract 1864.01, Los Angel…

                                          Geo_QName Geo_STATE Geo_COUNTY  \
0  Block Group 1, Census Tract 1863.01, Los Angel…        06        037
1  Block Group 2, Census Tract 1863.01, Los Angel…        06        037
2  Block Group 3, Census Tract 1863.01, Los Angel…        06        037
3  Block Group 1, Census Tract 1863.02, Los Angel…        06        037
4  Block Group 1, Census Tract 1864.01, Los Angel…        06        037

  Geo_TRACT  Geo_BLKGRP  SE_B14001_001  SE_B14001_002  SE_B14001_003  \
0    186301           1            330             46             67
1    186301           2            327             95            148
2    186301           3            268             73            129
3    186302           1            584            129            115
4    186401           1            675            199            167

  SE_B14001_004  SE_B14001_005  SE_B14001_006
0            54             50            113
1            31              9             44
2            44             10             12
3           109             71            160
4           172             69             68
```

[13]:
```python
columns = list(df) # this is the same as df.columns.to_list()
columns
```

[13]:
```
['Geo_FIPS',
 'Geo_NAME',
 'Geo_QName',
 'Geo_STATE',
 'Geo_COUNTY',
 'Geo_TRACT',
 'Geo_BLKGRP',
 'SE_B14001_001',
```

```
 'SE_B14001_002',
 'SE_B14001_003',
 'SE_B14001_004',
 'SE_B14001_005',
 'SE_B14001_006']
```

[14]:
```
df.columns = ['FIPS',
'Geo_NAME',
 'Geo_QName',
 'Geo_STATE',
 'Geo_COUNTY',
 'Geo_TRACT',
 'Geo_BLKGRP',
 'Total Households in Atwater Village',
 'Household Income Less than $25,000',
 'Household Income $25,000 to $49,999',
 'Household Income $50,000 to $74,999',
 'Household Income $75,000 to $99,999',
 'Household Income $100,000 or More']
```

[15]:
```
df.head()
```

[15]:
```
           FIPS                                           Geo_NAME  \
0  060371863011  Block Group 1, Census Tract 1863.01, Los Angel…
1  060371863012  Block Group 2, Census Tract 1863.01, Los Angel…
2  060371863013  Block Group 3, Census Tract 1863.01, Los Angel…
3  060371863021  Block Group 1, Census Tract 1863.02, Los Angel…
4  060371864011  Block Group 1, Census Tract 1864.01, Los Angel…

                                         Geo_QName Geo_STATE Geo_COUNTY  \
0  Block Group 1, Census Tract 1863.01, Los Angel…        06        037
1  Block Group 2, Census Tract 1863.01, Los Angel…        06        037
2  Block Group 3, Census Tract 1863.01, Los Angel…        06        037
3  Block Group 1, Census Tract 1863.02, Los Angel…        06        037
4  Block Group 1, Census Tract 1864.01, Los Angel…        06        037

  Geo_TRACT  Geo_BLKGRP  Total Households in Atwater Village  \
0    186301           1                                  330
1    186301           2                                  327
2    186301           3                                  268
3    186302           1                                  584
4    186401           1                                  675

   Household Income Less than $25,000  Household Income $25,000 to $49,999  \
0                                  46                                   67
1                                  95                                  148
2                                  73                                  129
```

```
3                                  129                                   115
4                                  199                                   167

   Household Income $50,000 to $74,999  Household Income $75,000 to $99,999  \
0                                   54                                   50
1                                   31                                    9
2                                   44                                   10
3                                  109                                   71
4                                  172                                   69

   Household Income $100,000 or More
0                                113
1                                 44
2                                 12
3                                160
4                                 68
```

[16]:

```
df2.columns[df2.isna().all()].tolist()
```

[16]:

```
['Geo_US',
 'Geo_REGION',
 'Geo_DIVISION',
 'Geo_STATECE',
 'Geo_COUSUB',
 'Geo_PLACE',
 'Geo_PLACESE',
 'Geo_CONCIT',
 'Geo_AIANHH',
 'Geo_AIANHHFP',
 'Geo_AIHHTLI',
 'Geo_AITSCE',
 'Geo_AITS',
 'Geo_ANRC',
 'Geo_CBSA',
 'Geo_CSA',
 'Geo_METDIV',
 'Geo_MACC',
 'Geo_MEMI',
 'Geo_NECTA',
 'Geo_CNECTA',
 'Geo_NECTADIV',
 'Geo_UA',
 'Geo_UACP',
 'Geo_CDCURR',
 'Geo_SLDU',
 'Geo_SLDL',
 'Geo_VTD',
```

```
 'Geo_ZCTA3',
 'Geo_ZCTA5',
 'Geo_SUBMCD',
 'Geo_SDELM',
 'Geo_SDSEC',
 'Geo_SDUNI',
 'Geo_UR',
 'Geo_PCI',
 'Geo_TAZ',
 'Geo_UGA',
 'Geo_BTTR',
 'Geo_BTBG',
 'Geo_PUMA5',
 'Geo_PUMA1']
```

[17]:
```
df2 = df2.dropna(axis=1,how="all")
```

[18]:
```
columns_to_drop =␣
 ↪['Geo_GEOID','Geo_STUSAB','Geo_SUMLEV','Geo_GEOCOMP','Geo_FILEID','Geo_LOGRECNO']
```

[19]:
```
df2 = df2.drop(columns_to_drop,axis=1)
df2.head()
```

[19]:
```
       Geo_FIPS                                           Geo_NAME  \
0  060372190203  Block Group 3, Census Tract 2190.20, Los Angel…
1  060372340001  Block Group 1, Census Tract 2340, Los Angeles …
2  060372340002  Block Group 2, Census Tract 2340, Los Angeles …
3  060372340003  Block Group 3, Census Tract 2340, Los Angeles …
4  060372340004  Block Group 4, Census Tract 2340, Los Angeles …

                                         Geo_QName Geo_STATE Geo_COUNTY  \
0  Block Group 3, Census Tract 2190.20, Los Angel…        06        037
1  Block Group 1, Census Tract 2340, Los Angeles …        06        037
2  Block Group 2, Census Tract 2340, Los Angeles …        06        037
3  Block Group 3, Census Tract 2340, Los Angeles …        06        037
4  Block Group 4, Census Tract 2340, Los Angeles …        06        037

  Geo_TRACT  Geo_BLKGRP  SE_B14001_001  SE_B14001_002  SE_B14001_003  \
0    219020           3            370            167             46
1    234000           1            634            147            140
2    234000           2            267             44             58
3    234000           3            247             14             47
4    234000           4            617            219             71

   SE_B14001_004  SE_B14001_005  SE_B14001_006
0            115             33              9
1            146             56            145
```

```
2             22              14              129
3             28              25              133
4            117             112               98
```

[20]:
```python
columns = list(df2) # this is the same as df.columns.to_list()
columns
```

[20]:
```
['Geo_FIPS',
 'Geo_NAME',
 'Geo_QName',
 'Geo_STATE',
 'Geo_COUNTY',
 'Geo_TRACT',
 'Geo_BLKGRP',
 'SE_B14001_001',
 'SE_B14001_002',
 'SE_B14001_003',
 'SE_B14001_004',
 'SE_B14001_005',
 'SE_B14001_006']
```

[21]:
```python
df2.columns = ['FIPS',
'Geo_NAME',
 'Geo_QName',
 'Geo_STATE',
 'Geo_COUNTY',
 'Geo_TRACT',
 'Geo_BLKGRP',
 'Total Households in Leimert Park',
 'Household Income Less than $25,000',
 'Household Income $25,000 to $49,999',
 'Household Income $50,000 to $74,999',
 'Household Income $75,000 to $99,999',
 'Household Income $100,000 or More']
```

[22]:
```python
df2.head()
```

[22]:
```
           FIPS                                           Geo_NAME  \
0  060372190203  Block Group 3, Census Tract 2190.20, Los Angel…
1  060372340001  Block Group 1, Census Tract 2340, Los Angeles …
2  060372340002  Block Group 2, Census Tract 2340, Los Angeles …
3  060372340003  Block Group 3, Census Tract 2340, Los Angeles …
4  060372340004  Block Group 4, Census Tract 2340, Los Angeles …

                                        Geo_QName Geo_STATE Geo_COUNTY  \
0  Block Group 3, Census Tract 2190.20, Los Angel…        06        037
1  Block Group 1, Census Tract 2340, Los Angeles …        06        037
```

```
2  Block Group 2, Census Tract 2340, Los Angeles …        06        037
3  Block Group 3, Census Tract 2340, Los Angeles …        06        037
4  Block Group 4, Census Tract 2340, Los Angeles …        06        037

   Geo_TRACT  Geo_BLKGRP  Total Households in Leimert Park  \
0     219020           3                               370
1     234000           1                               634
2     234000           2                               267
3     234000           3                               247
4     234000           4                               617

   Household Income Less than $25,000  Household Income $25,000 to $49,999  \
0                                 167                                   46
1                                 147                                  140
2                                  44                                   58
3                                  14                                   47
4                                 219                                   71

   Household Income $50,000 to $74,999  Household Income $75,000 to $99,999  \
0                                  115                                   33
1                                  146                                   56
2                                   22                                   14
3                                   28                                   25
4                                  117                                  112

   Household Income $100,000 or More
0                                  9
1                                145
2                                129
3                                133
4                                 98
```

[23]:
```
# access a single column like df['col_name'] to use to make a chart
df['Total Households in Atwater Village'].head()
```

[23]:
```
0    330
1    327
2    268
3    584
4    675
Name: Total Households in Atwater Village, dtype: int64
```

[24]:
```
# create a new column and normalize
# also repeat this for 'p_more_100k'
df['p_less_25k'] = df['Household Income Less than $25,000']/df['Total␣
 ↪Households in Atwater Village']*100
```

[25]:
```
# same process for'p_more_100k'
df['p_more_100k'] = df['Household Income $100,000 or More']/df['Total␣
↪Households in Atwater Village']*100
```

[26]:
```
# create a column to define the neighborhood
df['neighborhood'] = 'Atwater'
```

[27]:
```
df.head()
```

[27]:
```
           FIPS                                           Geo_NAME  \
0  060371863011  Block Group 1, Census Tract 1863.01, Los Angel…
1  060371863012  Block Group 2, Census Tract 1863.01, Los Angel…
2  060371863013  Block Group 3, Census Tract 1863.01, Los Angel…
3  060371863021  Block Group 1, Census Tract 1863.02, Los Angel…
4  060371864011  Block Group 1, Census Tract 1864.01, Los Angel…

                                         Geo_QName Geo_STATE Geo_COUNTY  \
0  Block Group 1, Census Tract 1863.01, Los Angel…        06        037
1  Block Group 2, Census Tract 1863.01, Los Angel…        06        037
2  Block Group 3, Census Tract 1863.01, Los Angel…        06        037
3  Block Group 1, Census Tract 1863.02, Los Angel…        06        037
4  Block Group 1, Census Tract 1864.01, Los Angel…        06        037

  Geo_TRACT  Geo_BLKGRP  Total Households in Atwater Village  \
0    186301           1                                  330
1    186301           2                                  327
2    186301           3                                  268
3    186302           1                                  584
4    186401           1                                  675

   Household Income Less than $25,000  Household Income $25,000 to $49,999  \
0                                  46                                   67
1                                  95                                  148
2                                  73                                  129
3                                 129                                  115
4                                 199                                  167

   Household Income $50,000 to $74,999  Household Income $75,000 to $99,999  \
0                                   54                                   50
1                                   31                                    9
2                                   44                                   10
3                                  109                                   71
4                                  172                                   69

   Household Income $100,000 or More  p_less_25k  p_more_100k neighborhood
0                                113   13.939394    34.242424      Atwater
1                                 44   29.051988    13.455657      Atwater
```

```
2                              12  27.238806   4.477612    Atwater
3                             160  22.089041  27.397260    Atwater
4                              68  29.481481  10.074074    Atwater
```

[28]:
```
# change df to df_merge
# this is only for the bar charts, not for maps
summary_df = df.groupby(['neighborhood']).mean()['p_less_25k'].reset_index()
```

[29]:
```
summary_df.head()
```

[29]:
```
  neighborhood  p_less_25k
0      Atwater   23.325771
```

[30]:
```
df2['Total Households in Leimert Park'].head()
```

[30]:
```
0    370
1    634
2    267
3    247
4    617
Name: Total Households in Leimert Park, dtype: int64
```

[31]:
```
df2['p_less_25k'] = df2['Household Income Less than $25,000']/df2['Total␣
 ↪Households in Leimert Park']*100
```

[32]:
```
df2['p_more_100k'] = df2['Household Income $100,000 or More']/df2['Total␣
 ↪Households in Leimert Park']*100
```

[33]:
```
df2['neighborhood'] = 'Leimert'
```

[34]:
```
summary_df2 = df2.groupby(['neighborhood']).mean()['p_less_25k'].reset_index()
```

[35]:
```
summary_df2.head()
```

[35]:
```
  neighborhood  p_less_25k
0      Leimert   30.765927
```

[36]:
```
df_merged1 = summary_df.append(summary_df2)
```

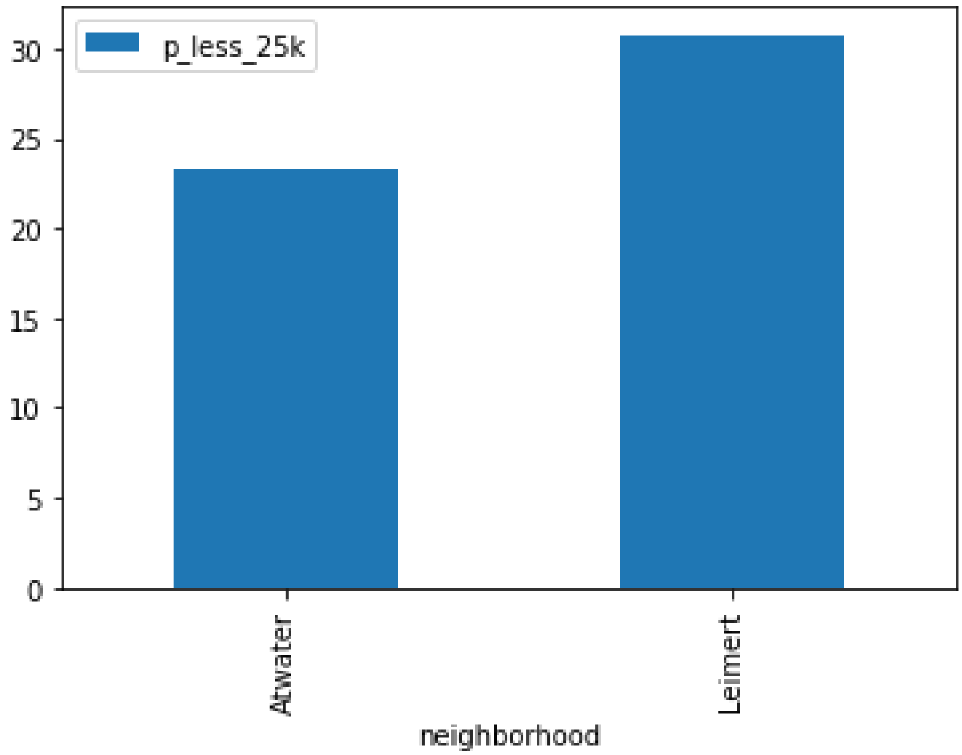
[37]:
```
df_merged1.head()
```

[37]:
```
  neighborhood  p_less_25k
0      Atwater   23.325771
0      Leimert   30.765927
```

[38]:
```
## We can see the 2017 Percentage of Households Side-By-Side to view what area␣
 ↪had more households making 25K or less
```

[39]:
```
df_merged1.plot.bar(x = 'neighborhood', y='p_less_25k')
```

[39]: <matplotlib.axes._subplots.AxesSubplot at 0x7f18159a2dc0>



[40]:
```
summaryat_df = df.groupby(['neighborhood']).mean()['p_more_100k'].reset_index()
```

[41]:
```
summaryat_df.head()
```

[41]:

	neighborhood	p_more_100k
0	Atwater	26.453039

[42]:
```
summaryle_df = df2.groupby(['neighborhood']).mean()['p_more_100k'].reset_index()
```

[43]:
```
summaryle_df.head()
```

[43]:

	neighborhood	p_more_100k
0	Leimert	22.915889

[44]:
```
df_merged2 = summaryat_df.append(summaryle_df)
```

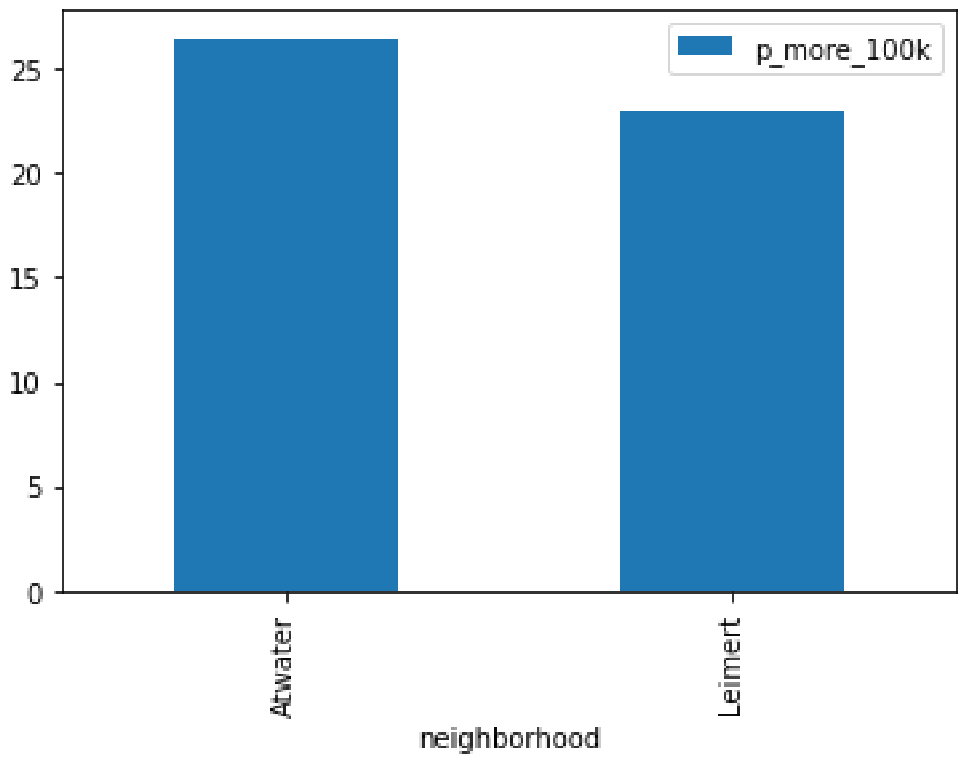
[45]:
```
df_merged2.head()
```

[45]:
```
  neighborhood  p_more_100k
0      Atwater    26.453039
0      Leimert    22.915889
```

[46]:
```
## We can see the 2017 Percentage of Households Side-By-Side to view what area
↪had more households making 100K or more
```

[47]:
```
df_merged2.plot.bar(x = 'neighborhood', y='p_more_100k')
```

[47]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1813551070>



[48]:
```
import plotly.express as px
import pandas as pd
```

[49]:
```
px.bar(df, x="Geo_NAME", y=["Household Income Less than $25,000",
 "Household Income $25,000 to $49,999",
 "Household Income $50,000 to $74,999",
 "Household Income $75,000 to $99,999",
 "Household Income $100,000 or More"], title="Income Breakdown by Block Group
↪in Atwater Village (2017)",
       labels={'Geo_NAME':'Census Tract Block Group','value':'Total Households
↪in Atwater Village', 'variable': 'Household Income'})
```

[50]:
```
px.bar(df2, x="Geo_NAME", y=["Household Income Less than $25,000",
 "Household Income $25,000 to $49,999",
 "Household Income $50,000 to $74,999",
 "Household Income $75,000 to $99,999",
 "Household Income $100,000 or More"], title="Income Breakdown by Block Group␣
↪in Leimert Park (2017)",
       labels={'Geo_NAME':'Census Tract Block Group','value':'Total Households␣
↪in Leimert Park', 'variable': 'Household Income'})
```

[51]:
```
## The bar graphs above were another way to understand Houehold Income before␣
↪isolating the highest and lowest salaries
```

2.3 Data Troubles

Given our neighborhood level approach to Atwater Village and Leimert Park – we decided to use Census Blocks instead of census tract data. This meeant that we were unable to use the census tract geojson maps from the LA Times. HOWEVER, ***Data from the US CENSUS*** *2019 TIGER/Line® Shapefiles: Block Groups provides FIPS code at the census block level.*

[62]:
```
import geopandas as gpd
block_groups= gpd.read_file('data/tl_2019_06_bg.shp')
```

[63]:
```
# subset to only LA block groups
block_groups_LA = block_groups[block_groups.COUNTYFP == '037']
```

[64]:
```
block_groups_LA.shape
```

[64]:
```
(6425, 13)
```

[65]:
```
df.head()
```

[65]:
```
          FIPS                                          Geo_NAME  \
0  060371863011  Block Group 1, Census Tract 1863.01, Los Angel…
1  060371863012  Block Group 2, Census Tract 1863.01, Los Angel…
2  060371863013  Block Group 3, Census Tract 1863.01, Los Angel…
3  060371863021  Block Group 1, Census Tract 1863.02, Los Angel…
4  060371864011  Block Group 1, Census Tract 1864.01, Los Angel…

                                         Geo_QName Geo_STATE Geo_COUNTY  \
0  Block Group 1, Census Tract 1863.01, Los Angel…        06        037
1  Block Group 2, Census Tract 1863.01, Los Angel…        06        037
2  Block Group 3, Census Tract 1863.01, Los Angel…        06        037
3  Block Group 1, Census Tract 1863.02, Los Angel…        06        037
4  Block Group 1, Census Tract 1864.01, Los Angel…        06        037

  Geo_TRACT  Geo_BLKGRP  Total Households in Atwater Village  \
0    186301           1                                  330
```

```
1     186301          2                            327
2     186301          3                            268
3     186302          1                            584
4     186401          1                            675

   Household Income Less than $25,000  Household Income $25,000 to $49,999  \
0                                  46                                   67
1                                  95                                  148
2                                  73                                  129
3                                 129                                  115
4                                 199                                  167

   Household Income $50,000 to $74,999  Household Income $75,000 to $99,999  \
0                                   54                                   50
1                                   31                                    9
2                                   44                                   10
3                                  109                                   71
4                                  172                                   69

   Household Income $100,000 or More  p_less_25k  p_more_100k neighborhood
0                                113   13.939394    34.242424      Atwater
1                                 44   29.051988    13.455657      Atwater
2                                 12   27.238806     4.477612      Atwater
3                                160   22.089041    27.397260      Atwater
4                                 68   29.481481    10.074074      Atwater
```

[67]:
```
block_groups_LA.plot
```

[67]:
```
<bound method GeoDataFrame.plot of       STATEFP COUNTYFP TRACTCE BLKGRPCE
GEOID      NAMELSAD  MTFCC  \
30          06      037   187200        2  060371872002  Block Group 2  G5030
31          06      037   187300        1  060371873001  Block Group 1  G5030
32          06      037   187300        2  060371873002  Block Group 2  G5030
36          06      037   543702        4  060375437024  Block Group 4  G5030
37          06      037   543702        5  060375437025  Block Group 5  G5030
…        …      …      …      …          …             …     …
23205       06      037   651101        1  060376511011  Block Group 1  G5030
23208       06      037   651101        2  060376511012  Block Group 2  G5030
23209       06      037   651201        1  060376512011  Block Group 1  G5030
23210       06      037   651201        2  060376512012  Block Group 2  G5030
23211       06      037   651201        4  060376512014  Block Group 4  G5030

      FUNCSTAT    ALAND  AWATER     INTPTLAT      INTPTLON  \
30           S   543859    8150  +34.1069009  -118.2493596
31           S   708405       0  +34.1041310  -118.2566334
32           S   460439       0  +34.0977378  -118.2568632
36           S   272187       0  +33.8108754  -118.2732027
```

```
37          S   211461       0  +33.8164435  -118.2734112
…       …      …     …           …            …
23205       S   871360       0  +33.8183814  -118.3326884
23208       S  4701799    2404  +33.8078567  -118.3379141
23209       S   275367       0  +33.8215124  -118.3613471
23210       S   486260       0  +33.8173894  -118.3692016
23211       S   329954       0  +33.8234527  -118.3722731

                                                geometry
30     POLYGON ((-118.25704 34.10773, -118.25664 34.1…
31     POLYGON ((-118.26544 34.11216, -118.26540 34.1…
32     POLYGON ((-118.25924 34.09416, -118.25924 34.0…
36     POLYGON ((-118.27553 33.80936, -118.27553 33.8…
37     POLYGON ((-118.27549 33.81246, -118.27549 33.8…
…                                                  …
23205  POLYGON ((-118.34007 33.82307, -118.34005 33.8…
23208  POLYGON ((-118.35103 33.80753, -118.35102 33.8…
23209  POLYGON ((-118.37264 33.81508, -118.37244 33.8…
23210  POLYGON ((-118.37444 33.81961, -118.37428 33.8…
23211  POLYGON ((-118.37450 33.82273, -118.37444 33.8…

[6425 rows x 13 columns]>
```

[68]:
```python
columns = list(block_groups)
columns
```

[68]:
```
['STATEFP',
 'COUNTYFP',
 'TRACTCE',
 'BLKGRPCE',
 'GEOID',
 'NAMELSAD',
 'MTFCC',
 'FUNCSTAT',
 'ALAND',
 'AWATER',
 'INTPTLAT',
 'INTPTLON',
 'geometry']
```

[69]:
```python
block_groups.head()
```

[69]:
```
  STATEFP COUNTYFP TRACTCE BLKGRPCE         GEOID       NAMELSAD  MTFCC  \
0      06      053  011101        3  060530111013  Block Group 3  G5030
1      06      053  011102        2  060530111022  Block Group 2  G5030
2      06      097  151308        4  060971513084  Block Group 4  G5030
3      06      051  000102        2  060510001022  Block Group 2  G5030
```

```
4      06      097  151309        2  060971513092  Block Group 2  G5030

  FUNCSTAT      ALAND  AWATER     INTPTLAT      INTPTLON  \
0        S    1399858     456  +36.4159450  -121.3164457
1        S    1451663       0  +36.4352108  -121.3300395
2        S     271694       0  +38.3545101  -122.6980543
3        S  433209402  199611  +38.4252824  -119.4757433
4        S    2306909       0  +38.3714772  -122.6782698

                                          geometry
0  POLYGON ((-121.32744 36.42351, -121.32726 36.4…
1  POLYGON ((-121.33743 36.43170, -121.33714 36.4…
2  POLYGON ((-122.70180 38.35481, -122.70180 38.3…
3  POLYGON ((-119.63717 38.32793, -119.63716 38.3…
4  POLYGON ((-122.69010 38.37132, -122.69009 38.3…
```

[70]:
```
blocks = block_groups_LA[['GEOID','geometry']]
blocks.head()
```

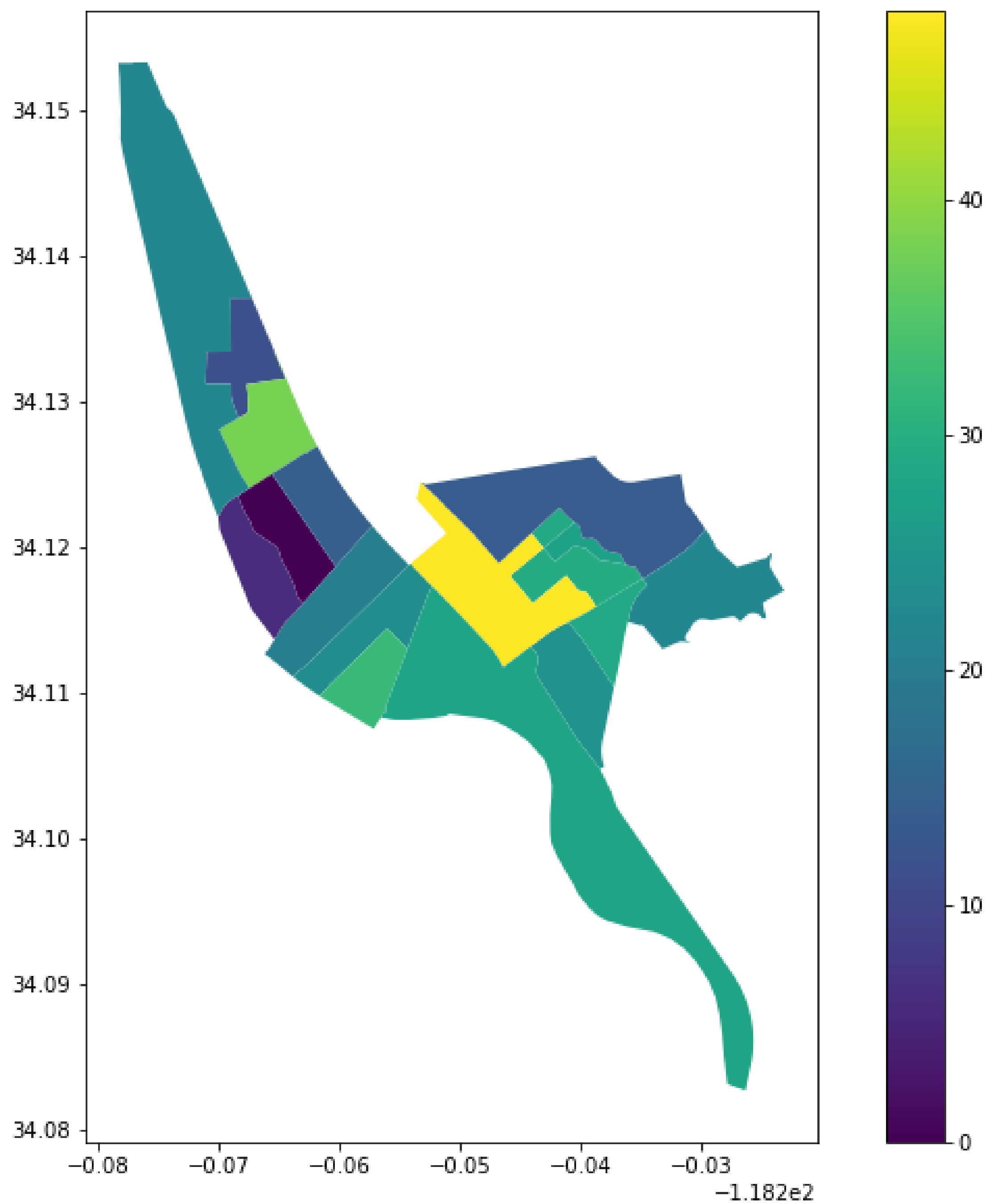
[70]:
```
           GEOID                                           geometry
30  060371872002  POLYGON ((-118.25704 34.10773, -118.25664 34.1…
31  060371873001  POLYGON ((-118.26544 34.11216, -118.26540 34.1…
32  060371873002  POLYGON ((-118.25924 34.09416, -118.25924 34.0…
36  060375437024  POLYGON ((-118.27553 33.80936, -118.27553 33.8…
37  060375437025  POLYGON ((-118.27549 33.81246, -118.27549 33.8…
```

[123]:
```
blocks.columns = ['FIPS','geometry']
```

[124]:
```
# create a new dataframe based on the join (ATWATER)
blocks_units=blocks.merge(df,on="FIPS")
```

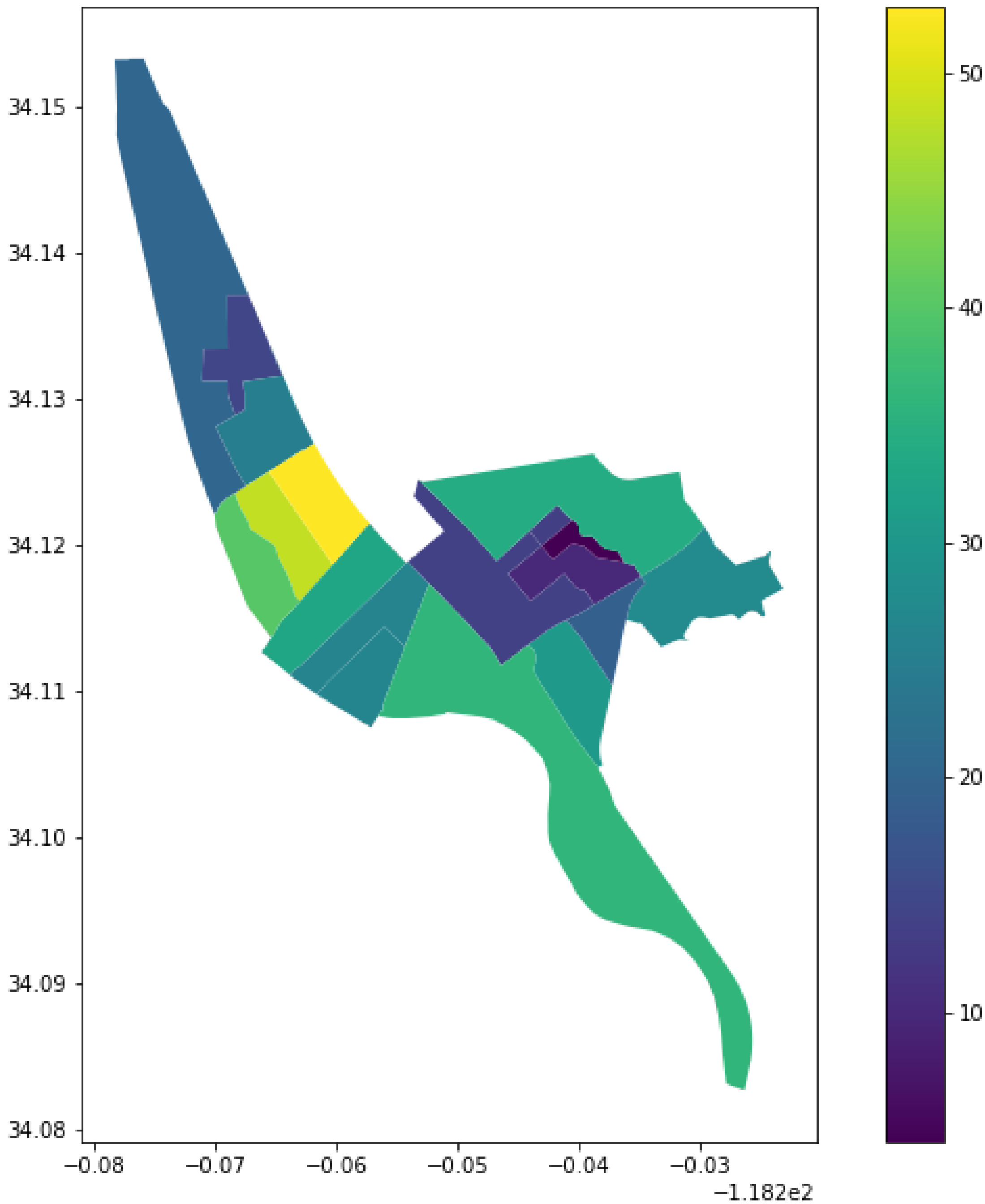
[125]:
```
# subsetting by neighborhood
blocks_units[blocks_units.neighborhood=='Atwater'].plot(figsize=(12,10),
                 column='p_less_25k',
                 legend=True,)
```

[125]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f17f3e0ca90>
```



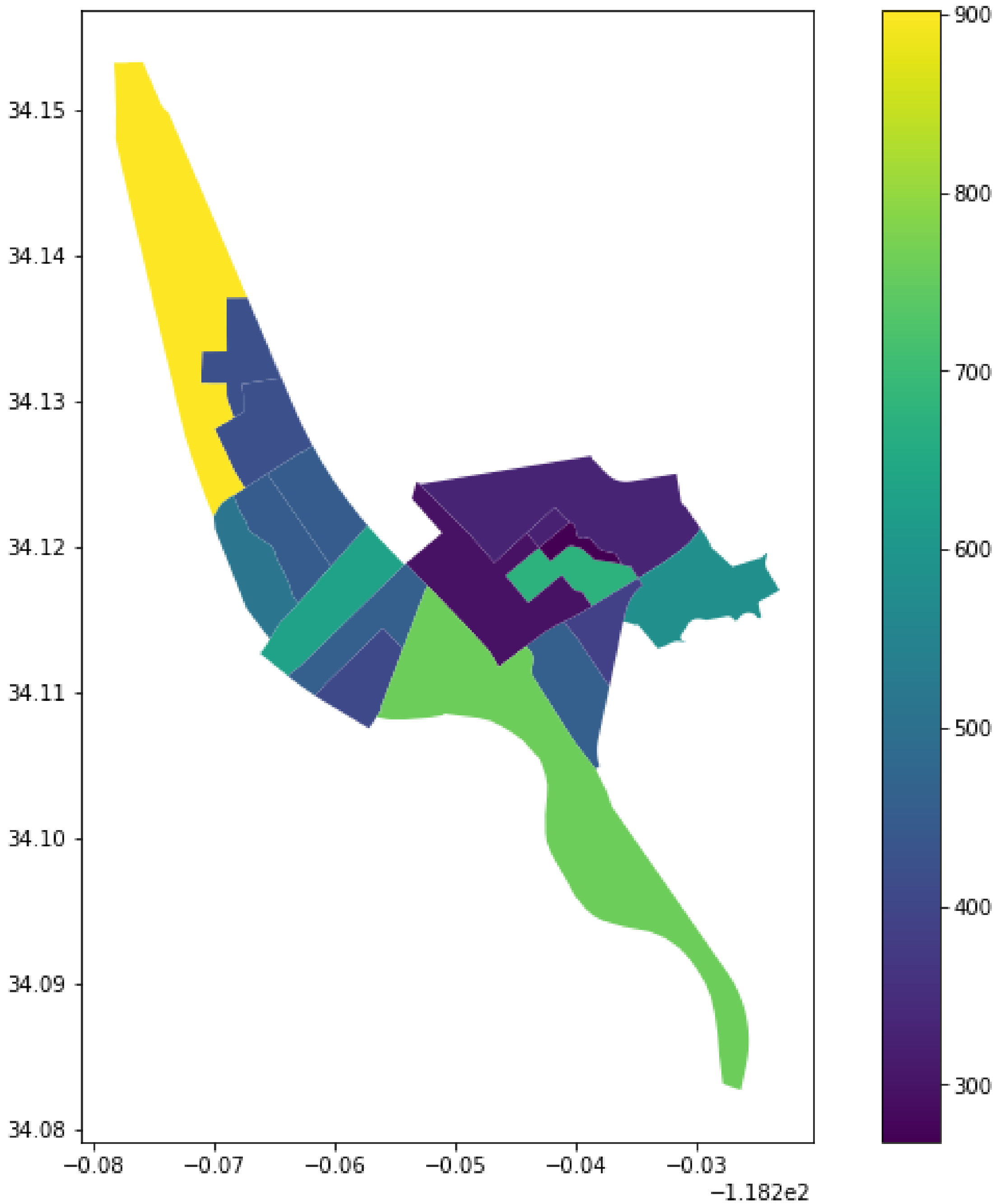
[114]:
```
blocks_units[blocks_units.neighborhood=='Atwater'].plot(figsize=(12,10),
                        column='p_more_100k',
                        legend=True,)
```

[114]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f4215880>



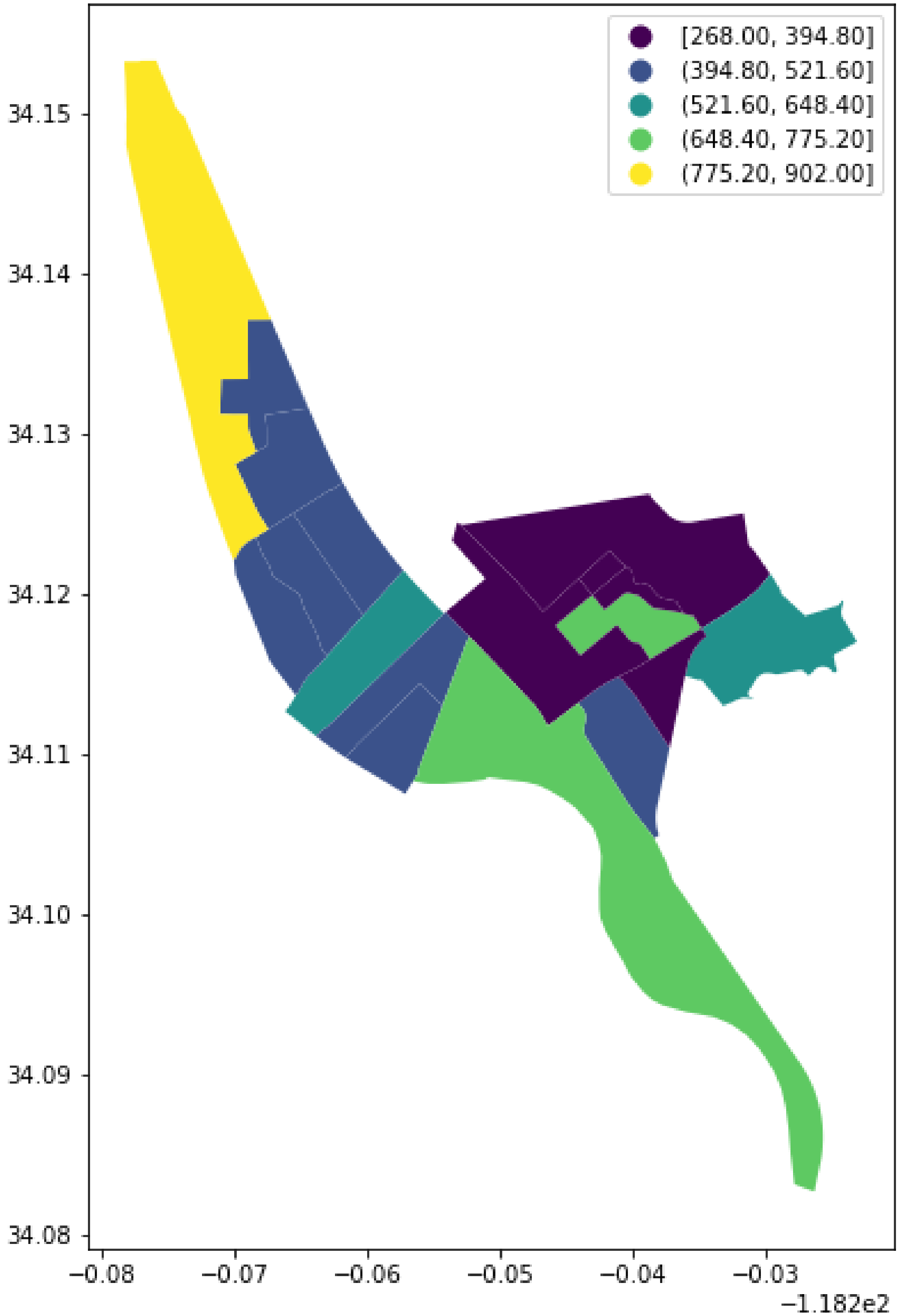

```
[115]: blocks_units.plot(figsize=(12,10),
                        column='Total Households in Atwater Village',
                        legend=True)
```

```
[115]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f410a2e0>
```



[116]:
```
blocks_units1.plot(figsize=(12,10),
                   column='Total Households in Atwater Village',
                   legend=True,
                   scheme='equal_interval')
```

[116]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f17f41ec790>`

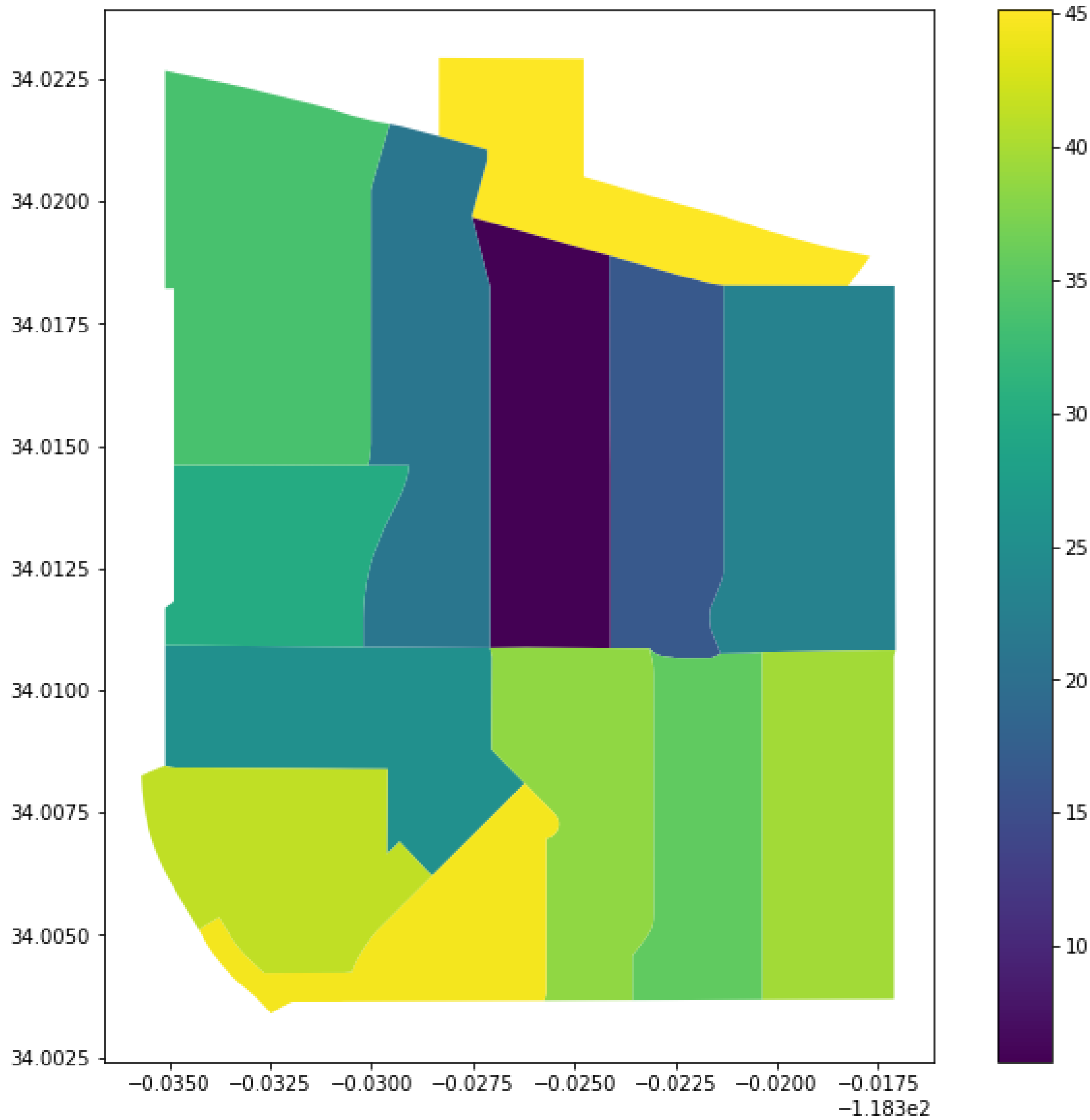


[268.00, 394.80]
(394.80, 521.60]
(521.60, 648.40]
(648.40, 775.20]
(775.20, 902.00]
34.15
34.14
34.13
34.12
34.11
34.10
34.09
34.08
−0.08
−0.07
−0.06
−0.05
−0.04
−0.03
−1.182e2

[121]:
```python
# create a new dataframe based on the join (LEIMERT)
blocks_units=blocks.merge(df2,on="FIPS")
```

[122]:
```python
blocks_units[blocks_units.neighborhood=='Leimert'].plot(figsize=(12,10),
                  column='p_less_25k',
                  legend=True,
                 )
```

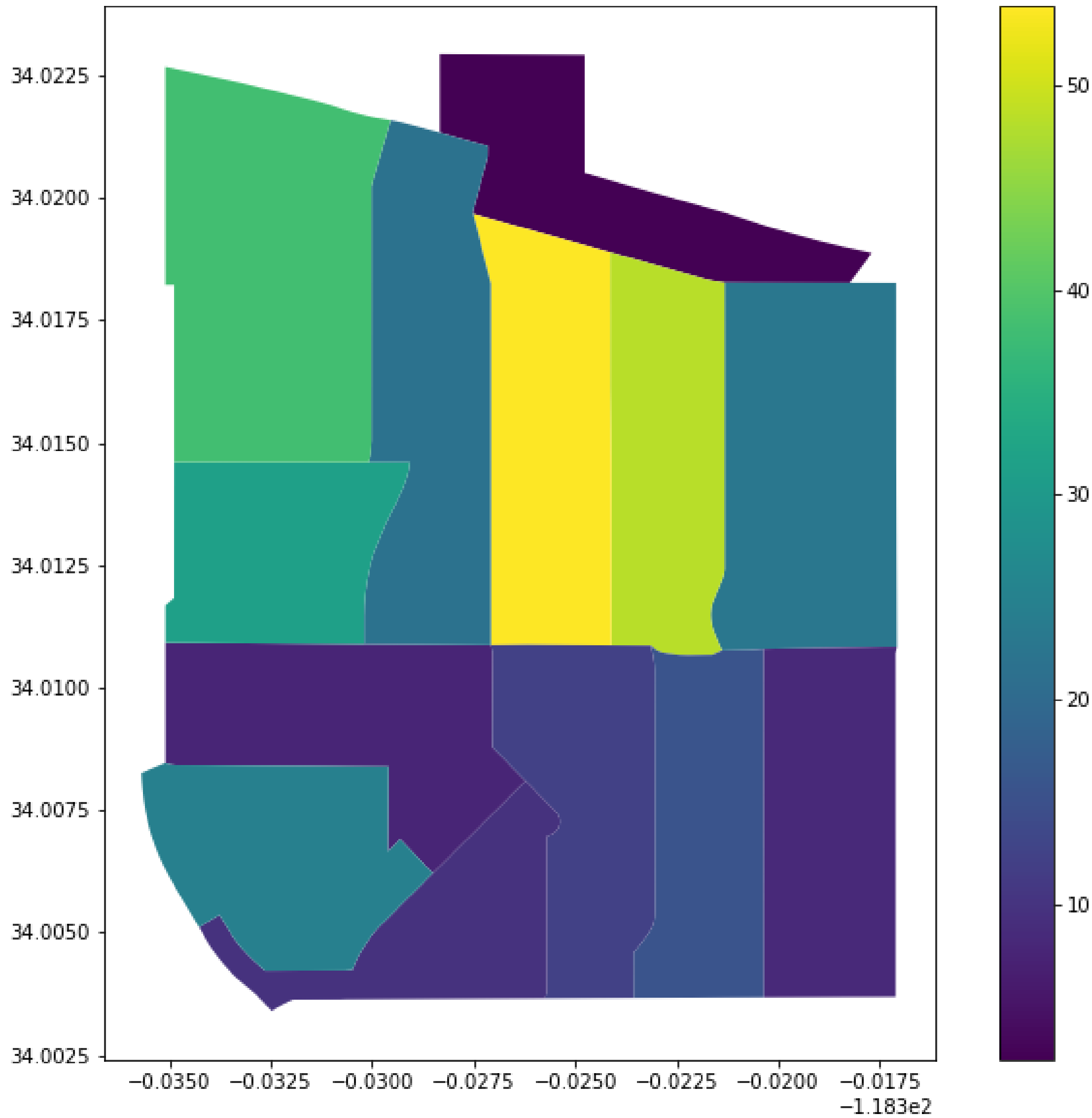
[122]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f3e93040>



[80]:
```python
blocks_units[blocks_units.neighborhood=='Leimert'].plot(figsize=(12,10),
                  column='p_more_100k',
```

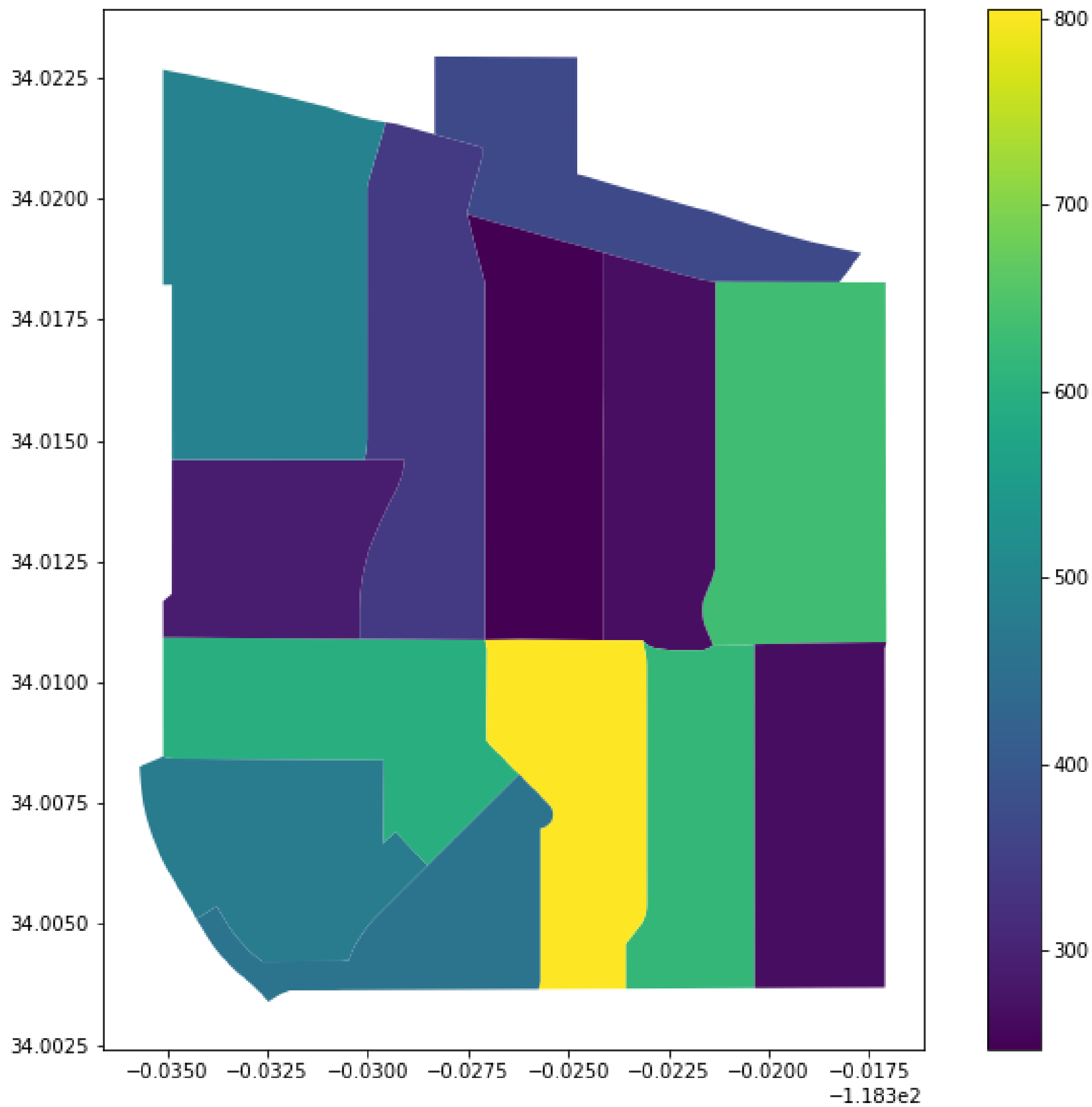
```
                 legend=True,)
```

[80]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17fb6f9880>




```
[81]: blocks_units.plot(figsize=(12,10),
                 column='Total Households in Leimert Park',
                 legend=True)
```

[81]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17fb6d6ca0>

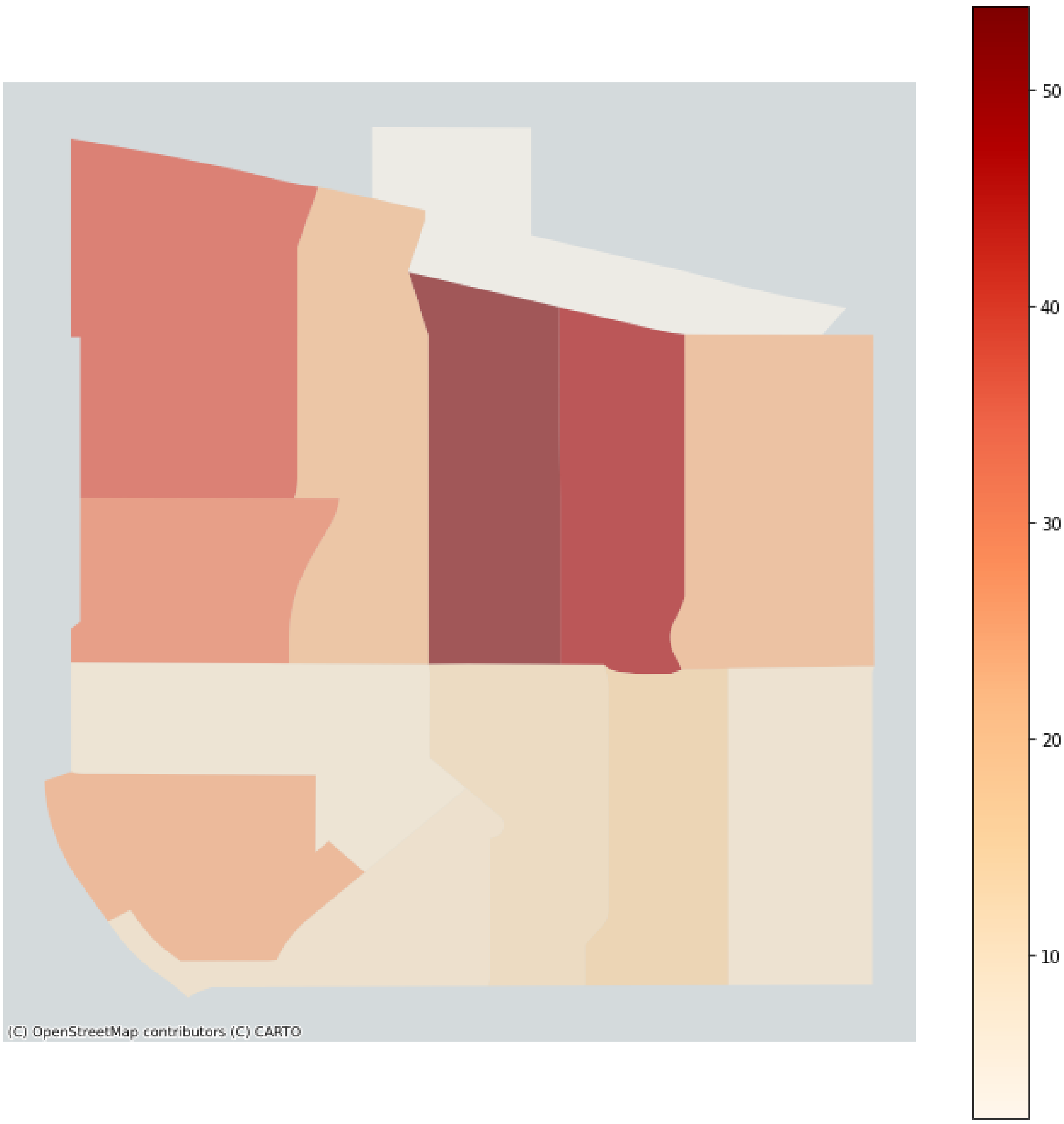



```
[ ]: ## Now we can see the Open Street Maps to understand how the unit type is␣
      ↪distributed
```

```
[100]: import contextily as ctx

blocks_units = blocks_units.to_crs(epsg=3857)

ax = blocks_units1.plot(figsize=(12,12),
                 column='p_more_100k',
                 legend=True,
                 alpha=0.6,
                 cmap='OrRd',)
```

```
# no axis
ax.axis('off')

# add a basemap
ctx.add_basemap(ax,source=ctx.providers.CartoDB.Positron)
```

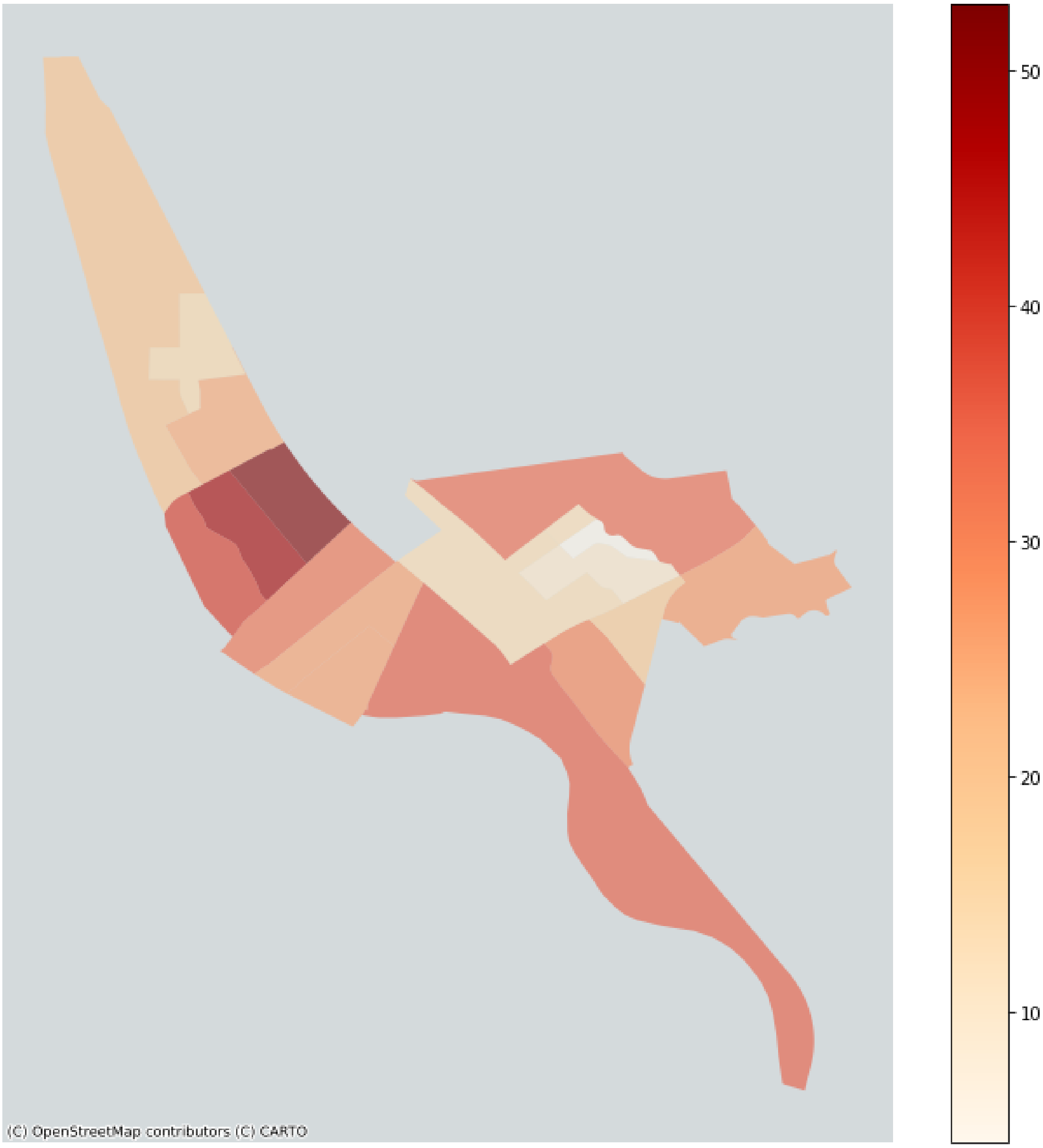
```
/opt/conda/lib/python3.8/site-packages/contextily/tile.py:632: UserWarning:

The inferred zoom level of 32 is not valid for the current tile provider (valid
zooms: 0 - 19).
```



[126]:

```
blocks_units= blocks_units.to_crs(epsg=3857)

ax = blocks_units1.plot(figsize=(12,12),
                 column='p_more_100k',
                 legend=True,
                 alpha=0.6,
                 cmap='OrRd',)



# no axis
ax.axis('off')

# add a basemap
ctx.add_basemap(ax,source=ctx.providers.CartoDB.Positron)
```

```
/opt/conda/lib/python3.8/site-packages/contextily/tile.py:632: UserWarning:

The inferred zoom level of 31 is not valid for the current tile provider (valid
zooms: 0 - 19).
```



2.4 More Charts! (Race + Median Value)

[127]:
```
df3 = pd.read_csv('data/Atwater2017_Race.csv')
```

[128]:
```
df3 = pd.read_csv(
    'data/Atwater2017_Race.csv',
    dtype=
    {
        'Geo_FIPS':str,
        'Geo_STATE':str,
```

```
        'Geo_COUNTY': str
    }
)
```

[129]:
```
df3.columns[df3.isna().all()].tolist()
```

[129]:
```
['Geo_US',
 'Geo_REGION',
 'Geo_DIVISION',
 'Geo_STATECE',
 'Geo_COUSUB',
 'Geo_PLACE',
 'Geo_PLACESE',
 'Geo_CONCIT',
 'Geo_AIANHH',
 'Geo_AIANHHFP',
 'Geo_AIHHTLI',
 'Geo_AITSCE',
 'Geo_AITS',
 'Geo_ANRC',
 'Geo_CBSA',
 'Geo_CSA',
 'Geo_METDIV',
 'Geo_MACC',
 'Geo_MEMI',
 'Geo_NECTA',
 'Geo_CNECTA',
 'Geo_NECTADIV',
 'Geo_UA',
 'Geo_UACP',
 'Geo_CDCURR',
 'Geo_SLDU',
 'Geo_SLDL',
 'Geo_VTD',
 'Geo_ZCTA3',
 'Geo_ZCTA5',
 'Geo_SUBMCD',
 'Geo_SDELM',
 'Geo_SDSEC',
 'Geo_SDUNI',
 'Geo_UR',
 'Geo_PCI',
 'Geo_TAZ',
 'Geo_UGA',
 'Geo_BTTR',
 'Geo_BTBG',
 'Geo_PUMA5',
```

```
 'Geo_PUMA1']
```

[130]:
```
df3 = df3.dropna(axis=1,how="all")
```

[131]:
```
columns_to_drop =␣
 ↪['Geo_GEOID','Geo_STUSAB','Geo_SUMLEV','Geo_GEOCOMP','Geo_FILEID','Geo_LOGRECNO','Geo_STATE
 ↪'Geo_COUNTY','SE_A04001_002','SE_A04001_011','SE_A04001_012','SE_A04001_013','SE_A04001_014
```

[132]:
```
df3 = df3.drop(columns_to_drop,axis=1)
df3.head()
```

[132]:
```
      Geo_FIPS                                         Geo_NAME  \
0  060371863011  Block Group 1, Census Tract 1863.01, Los Angel…
1  060371863012  Block Group 2, Census Tract 1863.01, Los Angel…
2  060371863013  Block Group 3, Census Tract 1863.01, Los Angel…
3  060371863021  Block Group 1, Census Tract 1863.02, Los Angel…
4  060371864011  Block Group 1, Census Tract 1864.01, Los Angel…

                                         Geo_QName  Geo_TRACT  Geo_BLKGRP  \
0  Block Group 1, Census Tract 1863.01, Los Angel…     186301           1
1  Block Group 2, Census Tract 1863.01, Los Angel…     186301           2
2  Block Group 3, Census Tract 1863.01, Los Angel…     186301           3
3  Block Group 1, Census Tract 1863.02, Los Angel…     186302           1
4  Block Group 1, Census Tract 1864.01, Los Angel…     186401           1

   SE_A04001_001  SE_A04001_003  SE_A04001_004  SE_A04001_005  SE_A04001_006  \
0            881            322             53              0             78
1           1349            103             23              0             89
2            889             59             25              0            196
3           1573            566             26              0            409
4           2676             78              0              0            312

   SE_A04001_007  SE_A04001_008  SE_A04001_009  SE_A04001_010
0              0              0             44            384
1              0              0              6           1128
2              0              0              4            605
3              0              0             84            488
4              0              0              7           2279
```

[133]:
```
columns3 = list(df3) # this is the same as df.columns.to_list()
columns3
```

[133]:
```
['Geo_FIPS',
 'Geo_NAME',
 'Geo_QName',
 'Geo_TRACT',
 'Geo_BLKGRP',
```

```
 'SE_A04001_001',
 'SE_A04001_003',
 'SE_A04001_004',
 'SE_A04001_005',
 'SE_A04001_006',
 'SE_A04001_007',
 'SE_A04001_008',
 'SE_A04001_009',
 'SE_A04001_010']
```

[134]:
```
##RENAMED TO INCLUDE PROPER TITLES

df3.columns = ['FIPS',
 'Geo_NAME',
 'Geo_QName',
 'Geo_TRACT',
 'Geo_BLKGRP',
 'Total Population',
 'White',
 'Black',
 'AmIndian',
'Native Hawaiian and Other Pacific Islander Alone',
 'Asian',
 'Some Other Race Alone',
 'Two or More Races',
 'Hispanic or Latino']
```

[135]:
```
import plotly.express as px
import pandas as pd
```

[136]:
```
px.bar(df3, x="Geo_NAME", y=["White","Black","AmIndian","Native Hawaiian and␣
↪Other Pacific Islander Alone","Asian","Some Other Race Alone","Two or More␣
↪Races","Hispanic or Latino"], title="Population by Race in Atwater Village␣
↪(2017)",
       labels={'Geo_NAME':'Census Tract Block Group','value':'Population',␣
↪'variable': 'Race'})
```

[137]:
```
df4 = pd.read_csv('data/Leimert2017_Race.csv')
```

[138]:
```
df4 = pd.read_csv(
    'data/Leimert2017_Race.csv',
    dtype=
    {
        'Geo_FIPS':str,
        'Geo_STATE':str,
        'Geo_COUNTY': str
    }
```

```
)
```

[139]:
```
df4.columns[df4.isna().all()].tolist()
```

[139]:
```
['Geo_US',
 'Geo_REGION',
 'Geo_DIVISION',
 'Geo_STATECE',
 'Geo_COUSUB',
 'Geo_PLACE',
 'Geo_PLACESE',
 'Geo_CONCIT',
 'Geo_AIANHH',
 'Geo_AIANHHFP',
 'Geo_AIHHTLI',
 'Geo_AITSCE',
 'Geo_AITS',
 'Geo_ANRC',
 'Geo_CBSA',
 'Geo_CSA',
 'Geo_METDIV',
 'Geo_MACC',
 'Geo_MEMI',
 'Geo_NECTA',
 'Geo_CNECTA',
 'Geo_NECTADIV',
 'Geo_UA',
 'Geo_UACP',
 'Geo_CDCURR',
 'Geo_SLDU',
 'Geo_SLDL',
 'Geo_VTD',
 'Geo_ZCTA3',
 'Geo_ZCTA5',
 'Geo_SUBMCD',
 'Geo_SDELM',
 'Geo_SDSEC',
 'Geo_SDUNI',
 'Geo_UR',
 'Geo_PCI',
 'Geo_TAZ',
 'Geo_UGA',
 'Geo_BTTR',
 'Geo_BTBG',
 'Geo_PUMA5',
 'Geo_PUMA1']
```

[140]:
```python
df4 = df4.dropna(axis=1,how="all")
```

[141]:
```python
columns_to_drop =␣
↪['Geo_GEOID','Geo_STUSAB','Geo_SUMLEV','Geo_GEOCOMP','Geo_FILEID','Geo_LOGRECNO','Geo_STATE
↪'Geo_COUNTY','SE_A04001_002','SE_A04001_011','SE_A04001_012','SE_A04001_013','SE_A04001_014
```

[142]:
```python
df4 = df4.drop(columns_to_drop,axis=1)
df4.head()
```

[142]:
```
       Geo_FIPS                                           Geo_NAME  \
0  060372190203  Block Group 3, Census Tract 2190.20, Los Angel…
1  060372340001  Block Group 1, Census Tract 2340, Los Angeles …
2  060372340002  Block Group 2, Census Tract 2340, Los Angeles …
3  060372340003  Block Group 3, Census Tract 2340, Los Angeles …
4  060372340004  Block Group 4, Census Tract 2340, Los Angeles …

                                          Geo_QName  Geo_TRACT  Geo_BLKGRP  \
0  Block Group 3, Census Tract 2190.20, Los Angel…     219020           3
1  Block Group 1, Census Tract 2340, Los Angeles …     234000           1
2  Block Group 2, Census Tract 2340, Los Angeles …     234000           2
3  Block Group 3, Census Tract 2340, Los Angeles …     234000           3
4  Block Group 4, Census Tract 2340, Los Angeles …     234000           4

   SE_A04001_001  SE_A04001_003  SE_A04001_004  SE_A04001_005  SE_A04001_006  \
0           1074             10            480              0             33
1           1849              7            849              0              0
2            559             59            428              0              9
3            697             41            606              0             23
4           1267            128            852              0              6

   SE_A04001_007  SE_A04001_008  SE_A04001_009  SE_A04001_010
0              0              0              0            551
1              0              0             45            948
2              0              0             11             52
3              0              0             19              8
4              0              0             45            236
```

[143]:
```python
columns4 = list(df4) # this is the same as df.columns.to_list()
columns4
```

[143]:
```
['Geo_FIPS',
 'Geo_NAME',
 'Geo_QName',
 'Geo_TRACT',
 'Geo_BLKGRP',
 'SE_A04001_001',
 'SE_A04001_003',
```

```
 'SE_A04001_004',
 'SE_A04001_005',
 'SE_A04001_006',
 'SE_A04001_007',
 'SE_A04001_008',
 'SE_A04001_009',
 'SE_A04001_010']
```

[146]:
```
df4.columns = ['FIPS',
 'Geo_NAME',
 'Geo_QName',
 'Geo_TRACT',
 'Geo_BLKGRP',
 'Total Population',
 'White',
 'Black',
 'AmIndian',
'Native Hawaiian and Other Pacific Islander Alone',
 'Asian',
 'Some Other Race Alone',
 'Two or More Races',
 'Hispanic or Latino']
```

[147]:
```
df4['Total Population'].head()
```

[147]:
```
0    1074
1    1849
2     559
3     697
4    1267
Name: Total Population, dtype: int64
```

[149]:
```
px.bar(df4, x="Geo_NAME", y=["White","Black","AmIndian","Native Hawaiian and␣
 ↪Other Pacific Islander Alone","Asian","Some Other Race Alone","Two or More␣
 ↪Races","Hispanic or Latino"], title="Population by Race in Leimert Park␣
 ↪(2017)",
       labels={'Geo_NAME':'Census Tract Block Group','value':'Population',␣
 ↪'variable': 'Race'})
```

[152]:
```
# create a new column and normalize
# also repeat this for 'p_White'
df3['p_Latino'] = df3['Hispanic or Latino']/df3['Total Population']*100
```

[153]:
```
df3.head()
```

[153]:
```
           FIPS                                          Geo_NAME  \
0  060371863011  Block Group 1, Census Tract 1863.01, Los Angel…
```

```
1  060371863012  Block Group 2, Census Tract 1863.01, Los Angel…
2  060371863013  Block Group 3, Census Tract 1863.01, Los Angel…
3  060371863021  Block Group 1, Census Tract 1863.02, Los Angel…
4  060371864011  Block Group 1, Census Tract 1864.01, Los Angel…

                                          Geo_QName  Geo_TRACT  Geo_BLKGRP  \
0  Block Group 1, Census Tract 1863.01, Los Angel…     186301           1
1  Block Group 2, Census Tract 1863.01, Los Angel…     186301           2
2  Block Group 3, Census Tract 1863.01, Los Angel…     186301           3
3  Block Group 1, Census Tract 1863.02, Los Angel…     186302           1
4  Block Group 1, Census Tract 1864.01, Los Angel…     186401           1

   Total Population  White  Black  AmIndian  \
0               881    322     53         0
1              1349    103     23         0
2               889     59     25         0
3              1573    566     26         0
4              2676     78      0         0

   Native Hawaiian and Other Pacific Islander Alone  Asian  \
0                                                78      0
1                                                89      0
2                                               196      0
3                                               409      0
4                                               312      0

   Some Other Race Alone  Two or More Races  Hispanic or Latino   p_Latino
0                      0                 44                 384  43.586833
1                      0                  6                1128  83.617494
2                      0                  4                 605  68.053993
3                      0                 84                 488  31.023522
4                      0                  7                2279  85.164425
```

[154]:

```
# create a new column for White Percentage
# also repeat this for 'p_White'
df3['p_White'] = df3['White']/df3['Total Population']*100
```

[155]:

```
df3.head()
```

[155]:

```
            FIPS                                          Geo_NAME  \
0  060371863011  Block Group 1, Census Tract 1863.01, Los Angel…
1  060371863012  Block Group 2, Census Tract 1863.01, Los Angel…
2  060371863013  Block Group 3, Census Tract 1863.01, Los Angel…
3  060371863021  Block Group 1, Census Tract 1863.02, Los Angel…
4  060371864011  Block Group 1, Census Tract 1864.01, Los Angel…

                                          Geo_QName  Geo_TRACT  Geo_BLKGRP  \
```

```
0  Block Group 1, Census Tract 1863.01, Los Angel…     186301           1
1  Block Group 2, Census Tract 1863.01, Los Angel…     186301           2
2  Block Group 3, Census Tract 1863.01, Los Angel…     186301           3
3  Block Group 1, Census Tract 1863.02, Los Angel…     186302           1
4  Block Group 1, Census Tract 1864.01, Los Angel…     186401           1

   Total Population  White  Black  AmIndian  \
0               881    322     53         0
1              1349    103     23         0
2               889     59     25         0
3              1573    566     26         0
4              2676     78      0         0

   Native Hawaiian and Other Pacific Islander Alone  Asian  \
0                                                78      0
1                                                89      0
2                                               196      0
3                                               409      0
4                                               312      0

   Some Other Race Alone  Two or More Races  Hispanic or Latino   p_Latino  \
0                      0                 44                 384  43.586833
1                      0                  6                1128  83.617494
2                      0                  4                 605  68.053993
3                      0                 84                 488  31.023522
4                      0                  7                2279  85.164425

     p_White
0  36.549376
1   7.635285
2   6.636670
3  35.982200
4   2.914798
```

[156]:
```
# create a column to define the neighborhood
df3['neighborhood'] = 'Atwater'
```

[157]:
```
summary_df3 = df3.groupby(['neighborhood']).mean()['p_Latino'].reset_index()
```

[159]:
```
summary_df3.head()
```

[159]:
```
  neighborhood   p_Latino
0      Atwater  52.722182
```

[168]:
```
summary_df3W = df3.groupby(['neighborhood']).mean()['p_White'].reset_index()
```

[169]:
```
summary_df3W.head()
```

[169]:

	neighborhood	p_White
0	Atwater	23.655247

[162]:
```
df_mergedAtLW = summary_df3.append(summary_df3W)
```

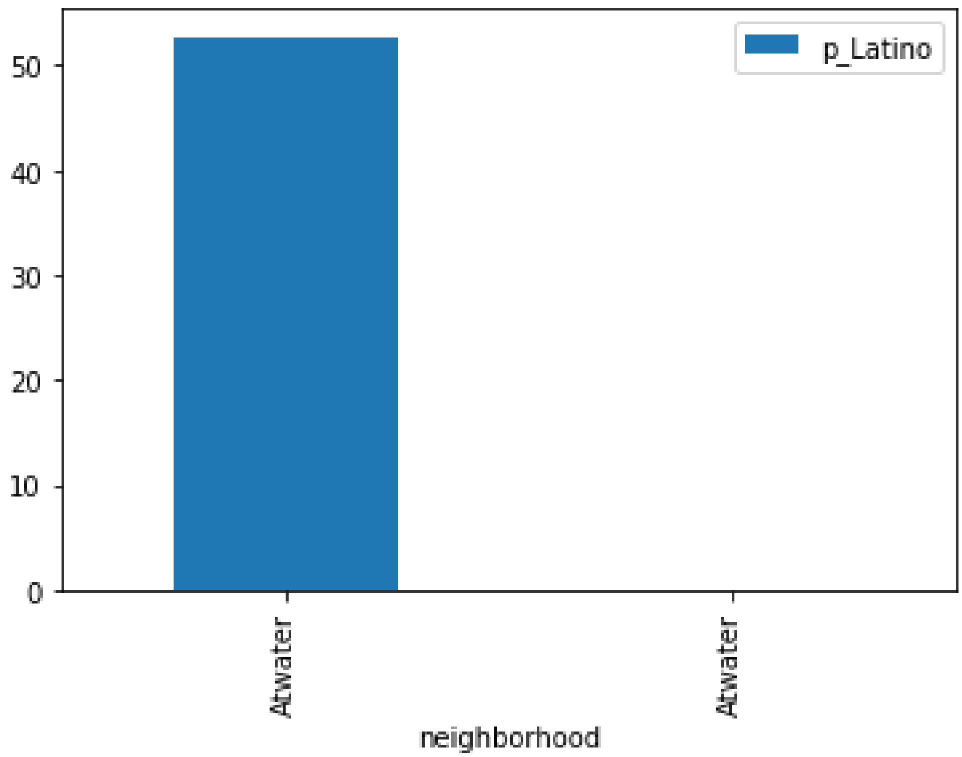
[163]:
```
df_mergedAtLW.head()
```

[163]:

	neighborhood	p_Latino	p_White
0	Atwater	52.722182	NaN
0	Atwater	NaN	23.655247

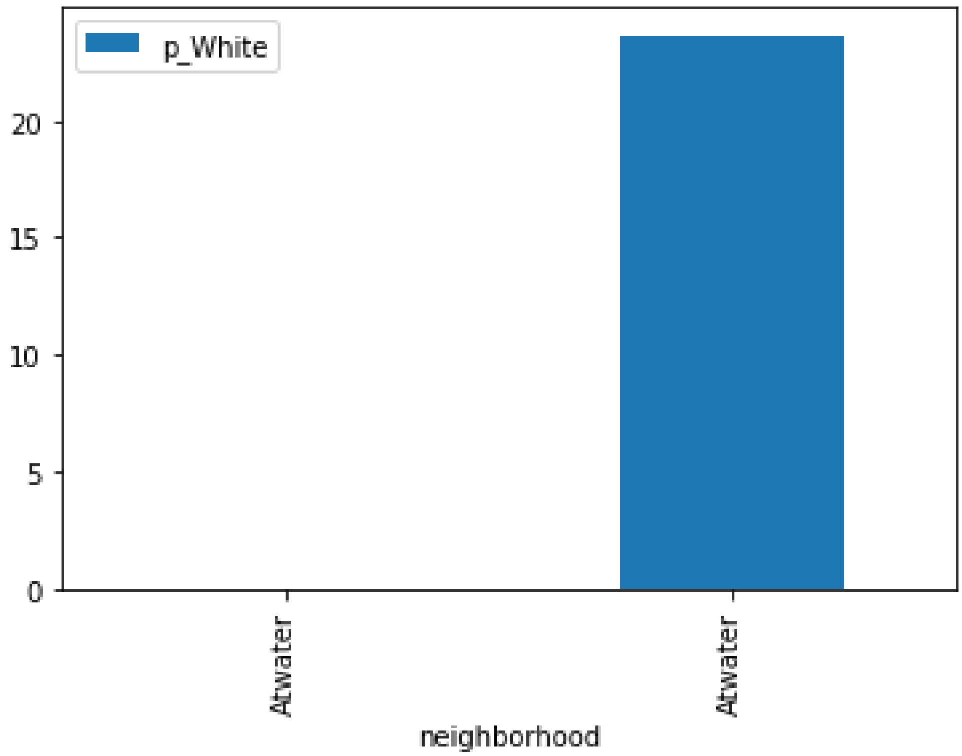
[166]:
```
df_mergedAtLW.plot.bar(x = 'neighborhood', y='p_Latino')
```

[166]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f3bb0c40>



[170]:
```
df_mergedAtLW.plot.bar(x = 'neighborhood', y='p_White')
```

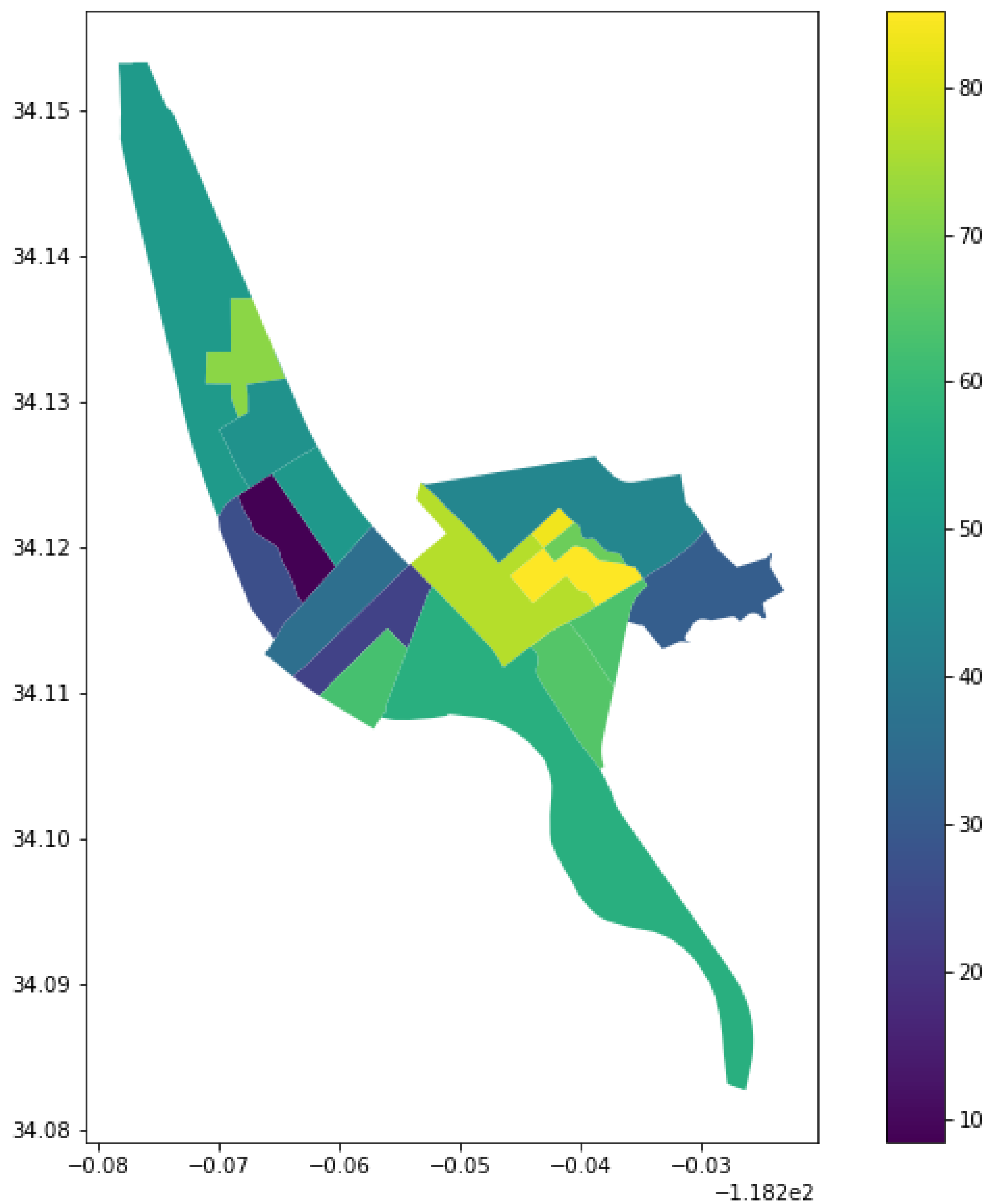
[170]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f3bd63d0>



[175]:
```
# create a new dataframe based on the join (ATWATER)
blocks_units_race_latino=blocks.merge(df3,on="FIPS")
```

[176]:
```
blocks_units_race_latino[blocks_units.neighborhood=='Atwater'].
 ↪plot(figsize=(12,10),
                 column='p_Latino',
                 legend=True,)
```

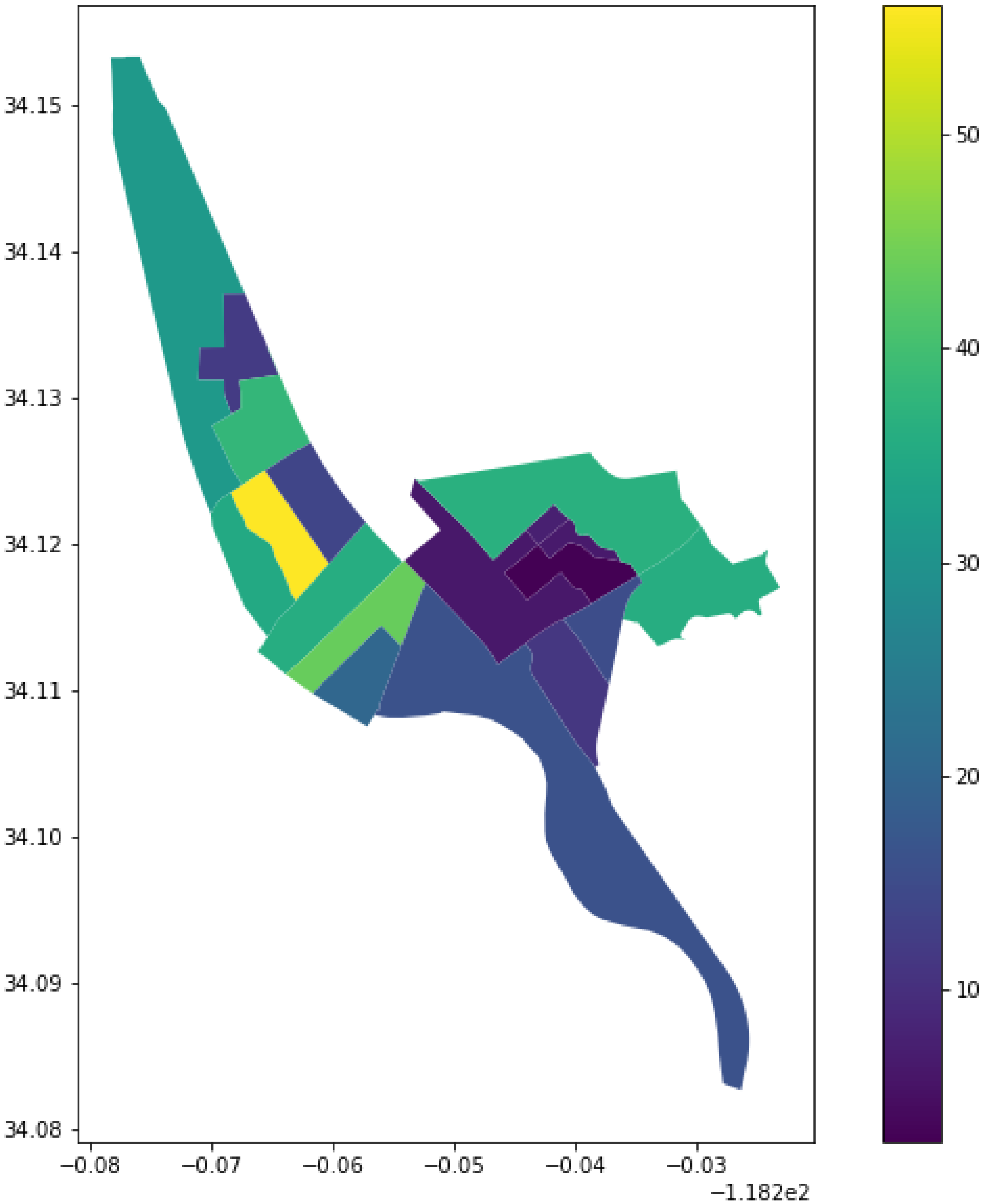
[176]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f3a694f0>



[177]:
```
blocks_units_race_white=blocks.merge(df3,on="FIPS")
```

[178]:
```
blocks_units_race_white[blocks_units.neighborhood=='Atwater'].
↪plot(figsize=(12,10),
                 column='p_White',
                 legend=True,)
```

[178]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f3a56550>



[187]:
```
# create a new column and normalize
# also repeat this for 'p_White'
df4['p_Black'] = df4['Black']/df4['Total Population']*100
```

[188]:
```
# create a new column for White Percentage
# also repeat this for 'p_White'
df4['p_White'] = df4['White']/df4['Total Population']*100
```

[189]:
```
# create a column to define the neighborhood
df4['neighborhood'] = 'Leimert'
```

[190]:
```
summary_df4B = df4.groupby(['neighborhood']).mean()['p_Black'].reset_index()
```

[191]:
```
summary_df4W = df4.groupby(['neighborhood']).mean()['p_White'].reset_index()
```

[192]:
```
df_mergedAtBW = summary_df4B.append(summary_df4W)
```

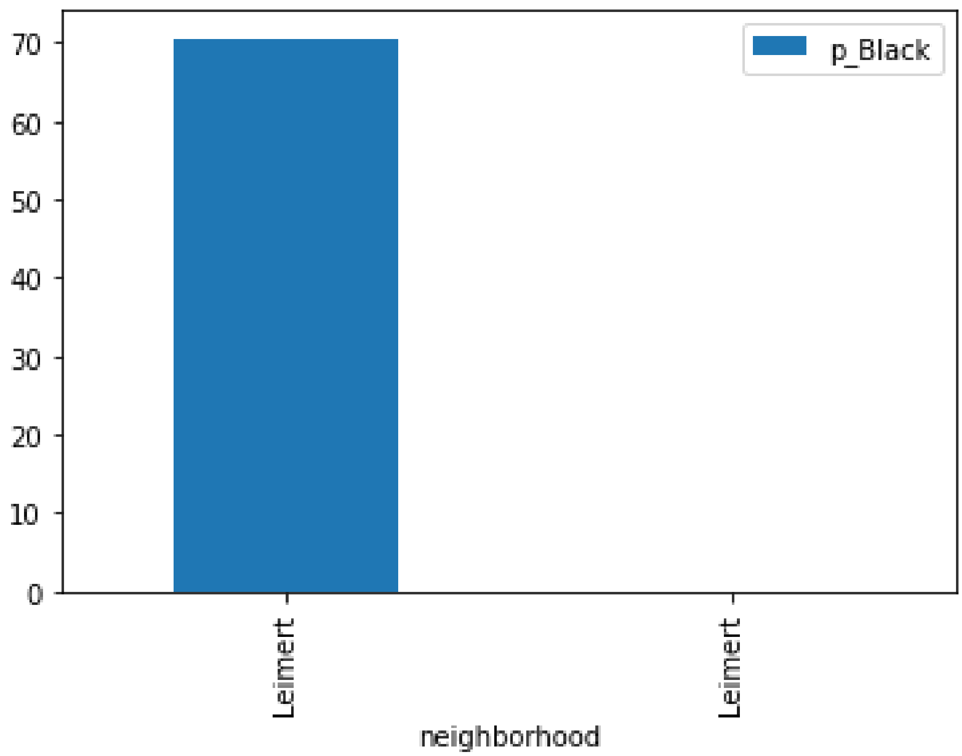
[193]:
```
df_mergedAtBW.head()
```

[193]:

	neighborhood	p_Black	p_White
0	Leimert	70.634012	NaN
0	Leimert	NaN	4.847755

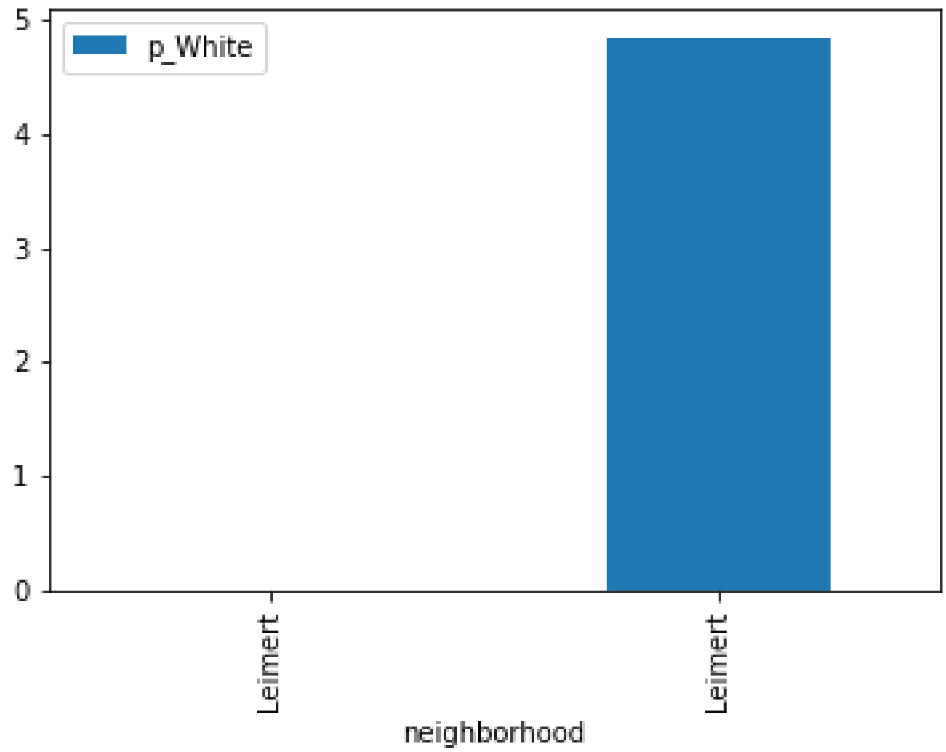
[195]:
```
df_mergedAtBW.plot.bar(x = 'neighborhood', y='p_Black')
```

[195]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f39b69d0>



[196]:
```
df_mergedAtBW.plot.bar(x = 'neighborhood', y='p_White')
```

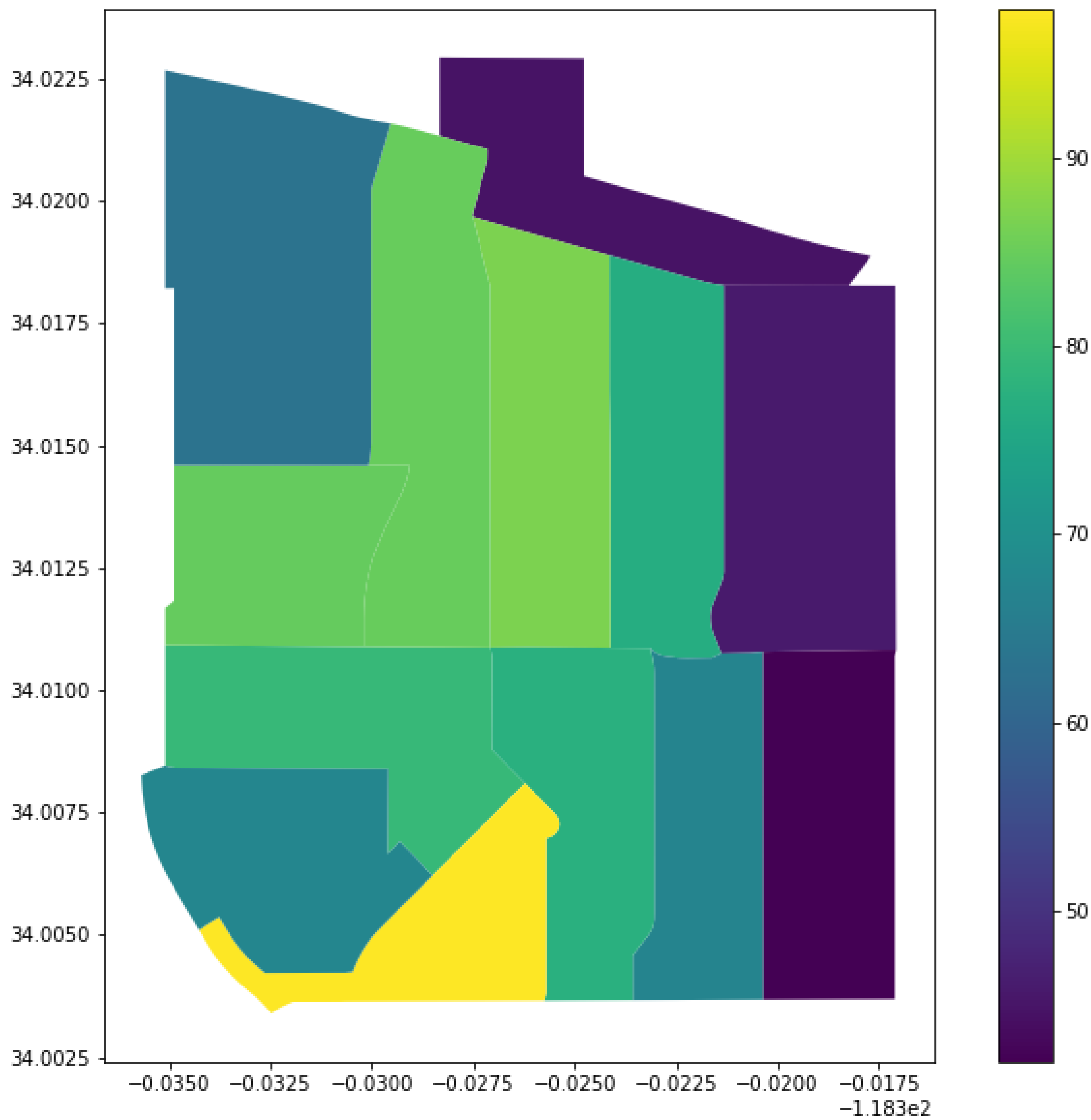
[196]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f39885e0>



[197]:
```
# create a new dataframe based on the join (LEIMERT)
blocks_units_race_black=blocks.merge(df4,on="FIPS")
```

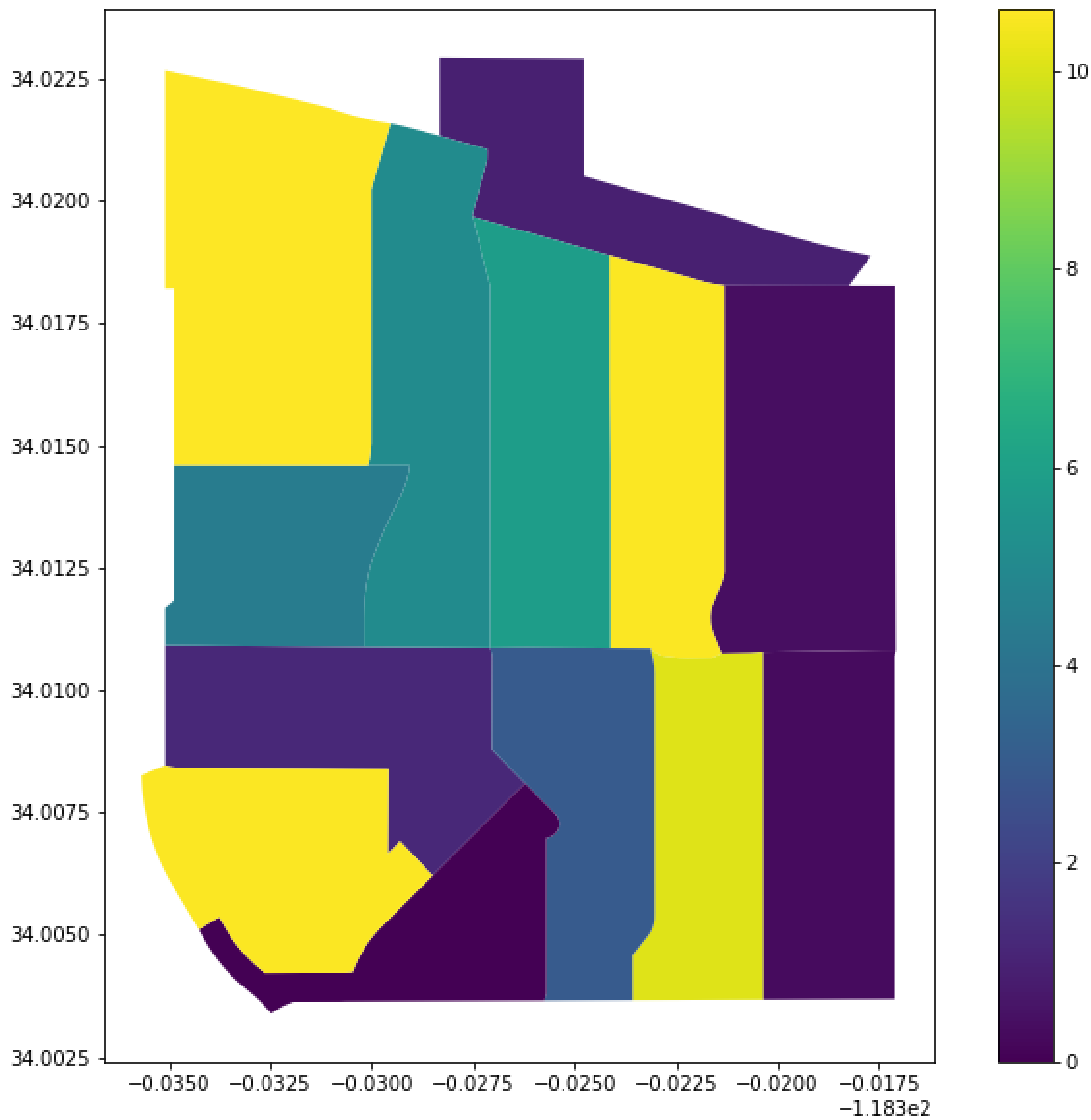
[204]:
```
# by mapping out the Leimert Park neighborhood by Black households, it is clear␣
↪that all areas have Black households
blocks_units_race_black[blocks_units_race_black.neighborhood=='Leimert'].
↪plot(figsize=(12,10),
                 column='p_Black',
                 legend=True,)
```

[204]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f3b45ac0>



[206]:
```
# By mapping out the Leimert Park neighborhood by Race, it is clear that there␣
↪are not many White households in this area
blocks_units_race_black[blocks_units_race_black.neighborhood=='Leimert'].
↪plot(figsize=(12,10),
                 column='p_White',
                 legend=True,)
```

[206]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f37f28b0>



[207]:
```
#Now we can look at the home values to further understand if any correlations␣
↪exist between Race and Home Value
df5 = pd.read_csv('data/Atwater2017_HomeValue.csv')
```

[208]:
```
df5 = pd.read_csv(
    'data/Atwater2017_HomeValue.csv',
    dtype=
    {
        'Geo_FIPS':str,
        'Geo_STATE':str,
        'Geo_COUNTY': str
    }
```

```
)
```

[209]:
```
df5.columns[df5.isna().all()].tolist()
```

[209]:
```
['Geo_US',
 'Geo_REGION',
 'Geo_DIVISION',
 'Geo_STATECE',
 'Geo_COUSUB',
 'Geo_PLACE',
 'Geo_PLACESE',
 'Geo_CONCIT',
 'Geo_AIANHH',
 'Geo_AIANHHFP',
 'Geo_AIHHTLI',
 'Geo_AITSCE',
 'Geo_AITS',
 'Geo_ANRC',
 'Geo_CBSA',
 'Geo_CSA',
 'Geo_METDIV',
 'Geo_MACC',
 'Geo_MEMI',
 'Geo_NECTA',
 'Geo_CNECTA',
 'Geo_NECTADIV',
 'Geo_UA',
 'Geo_UACP',
 'Geo_CDCURR',
 'Geo_SLDU',
 'Geo_SLDL',
 'Geo_VTD',
 'Geo_ZCTA3',
 'Geo_ZCTA5',
 'Geo_SUBMCD',
 'Geo_SDELM',
 'Geo_SDSEC',
 'Geo_SDUNI',
 'Geo_UR',
 'Geo_PCI',
 'Geo_TAZ',
 'Geo_UGA',
 'Geo_BTTR',
 'Geo_BTBG',
 'Geo_PUMA5',
 'Geo_PUMA1']
```

[210]:
```
df5 = df5.dropna(axis=1,how="all")
```

[211]:
```
columns_to_drop =␣
↪['Geo_GEOID','Geo_STUSAB','Geo_SUMLEV','Geo_GEOCOMP','Geo_FILEID','Geo_LOGRECNO','Geo_STATE
↪'Geo_COUNTY']
```

[212]:
```
df5 = df5.drop(columns_to_drop,axis=1)
df5.head()
```

[212]:
```
       Geo_FIPS                                          Geo_NAME  \
0  060371863011  Block Group 1, Census Tract 1863.01, Los Angel…
1  060371863012  Block Group 2, Census Tract 1863.01, Los Angel…
2  060371863013  Block Group 3, Census Tract 1863.01, Los Angel…
3  060371863021  Block Group 1, Census Tract 1863.02, Los Angel…
4  060371864011  Block Group 1, Census Tract 1864.01, Los Angel…

                                          Geo_QName  Geo_TRACT  Geo_BLKGRP  \
0  Block Group 1, Census Tract 1863.01, Los Angel…     186301           1
1  Block Group 2, Census Tract 1863.01, Los Angel…     186301           2
2  Block Group 3, Census Tract 1863.01, Los Angel…     186301           3
3  Block Group 1, Census Tract 1863.02, Los Angel…     186302           1
4  Block Group 1, Census Tract 1864.01, Los Angel…     186401           1

   SE_A10036_001
0       444300.0
1            NaN
2       587500.0
3       547500.0
4       632100.0
```

[213]:
```
columns5 = list(df5) # this is the same as df.columns.to_list()
columns5
```

[213]:
```
['Geo_FIPS',
 'Geo_NAME',
 'Geo_QName',
 'Geo_TRACT',
 'Geo_BLKGRP',
 'SE_A10036_001']
```

[214]:
```
df5.columns = ['FIPS',
 'Geo_NAME',
 'Geo_QName',
 'Geo_TRACT',
 'Geo_BLKGRP',
 'Median Home Value']
```

[215]:
```
df5['Median Home Value'].head()
```

[215]:
```
0    444300.0
1         NaN
2    587500.0
3    547500.0
4    632100.0
Name: Median Home Value, dtype: float64
```

[216]:
```
px.bar(df5, x="Geo_NAME", y=["Median Home Value"], title="Median Home Value in␣
↪Atwater Village (2017)",
       labels={'Geo_NAME':'Census Tract Block Group','value':'Value',␣
↪'variable': 'Value'})
```

[217]:
```
df6 = pd.read_csv('data/Leimert2017_HomeValue.csv')
```

[218]:
```
df6 = pd.read_csv(
    'data/Leimert2017_HomeValue.csv',
    dtype=
    {
        'Geo_FIPS':str,
        'Geo_STATE':str,
        'Geo_COUNTY': str
    }
)
```

[219]:
```
df6.columns[df6.isna().all()].tolist()
```

[219]:
```
['Geo_US',
 'Geo_REGION',
 'Geo_DIVISION',
 'Geo_STATECE',
 'Geo_COUSUB',
 'Geo_PLACE',
 'Geo_PLACESE',
 'Geo_CONCIT',
 'Geo_AIANHH',
 'Geo_AIANHHFP',
 'Geo_AIHHTLI',
 'Geo_AITSCE',
 'Geo_AITS',
 'Geo_ANRC',
 'Geo_CBSA',
 'Geo_CSA',
 'Geo_METDIV',
 'Geo_MACC',
 'Geo_MEMI',
```

```
 'Geo_NECTA',
 'Geo_CNECTA',
 'Geo_NECTADIV',
 'Geo_UA',
 'Geo_UACP',
 'Geo_CDCURR',
 'Geo_SLDU',
 'Geo_SLDL',
 'Geo_VTD',
 'Geo_ZCTA3',
 'Geo_ZCTA5',
 'Geo_SUBMCD',
 'Geo_SDELM',
 'Geo_SDSEC',
 'Geo_SDUNI',
 'Geo_UR',
 'Geo_PCI',
 'Geo_TAZ',
 'Geo_UGA',
 'Geo_BTTR',
 'Geo_BTBG',
 'Geo_PUMA5',
 'Geo_PUMA1']
```

[220]:
```
df6 = df6.dropna(axis=1,how="all")
```

[221]:
```
columns_to_drop =␣
 ↪['Geo_GEOID','Geo_STUSAB','Geo_SUMLEV','Geo_GEOCOMP','Geo_FILEID','Geo_LOGRECNO','Geo_STATE
 ↪'Geo_COUNTY']
```

[222]:
```
df6 = df6.drop(columns_to_drop,axis=1)
df6.head()
```

[222]:
```
       Geo_FIPS                                           Geo_NAME  \
0  060372190203  Block Group 3, Census Tract 2190.20, Los Angel…
1  060372340001  Block Group 1, Census Tract 2340, Los Angeles …
2  060372340002  Block Group 2, Census Tract 2340, Los Angeles …
3  060372340003  Block Group 3, Census Tract 2340, Los Angeles …
4  060372340004  Block Group 4, Census Tract 2340, Los Angeles …

                                          Geo_QName  Geo_TRACT  Geo_BLKGRP  \
0  Block Group 3, Census Tract 2190.20, Los Angel…     219020           3
1  Block Group 1, Census Tract 2340, Los Angeles …     234000           1
2  Block Group 2, Census Tract 2340, Los Angeles …     234000           2
3  Block Group 3, Census Tract 2340, Los Angeles …     234000           3
4  Block Group 4, Census Tract 2340, Los Angeles …     234000           4
```

```
   SE_A10036_001
0         302900
1         406900
2         447500
3         614900
4         454900
```

[223]:
```
columns6 = list(df6) # this is the same as df.columns.to_list()
columns6
```

[223]:
```
['Geo_FIPS',
 'Geo_NAME',
 'Geo_QName',
 'Geo_TRACT',
 'Geo_BLKGRP',
 'SE_A10036_001']
```

[224]:
```
df6.columns = ['FIPS',
 'Geo_NAME',
 'Geo_QName',
 'Geo_TRACT',
 'Geo_BLKGRP',
 'Median Home Value']
```

[225]:
```
df6['Median Home Value'].head()
```

[225]:
```
0    302900
1    406900
2    447500
3    614900
4    454900
Name: Median Home Value, dtype: int64
```

[226]:
```
px.bar(df6, x="Geo_NAME", y=["Median Home Value"], title="Median Home Value in␣
 ↪Leimert Park (2017)",
       labels={'Geo_NAME':'Census Tract Block Group','value':'Value',␣
 ↪'variable': 'Value'})
```

[227]:
```
# create a column to define the neighborhood
df5['neighborhood'] = 'Atwater'
```

[228]:
```
df6['neighborhood'] = 'Leimert'
```

[229]:
```
summary_df5 = df5.groupby(['neighborhood']).mean()['Median Home Value'].
 ↪reset_index()
```

[230]:
```
summary_df6 = df6.groupby(['neighborhood']).mean()['Median Home Value'].
↪reset_index()
```

[231]:
```
df_mergedValue = summary_df5.append(summary_df6)
```

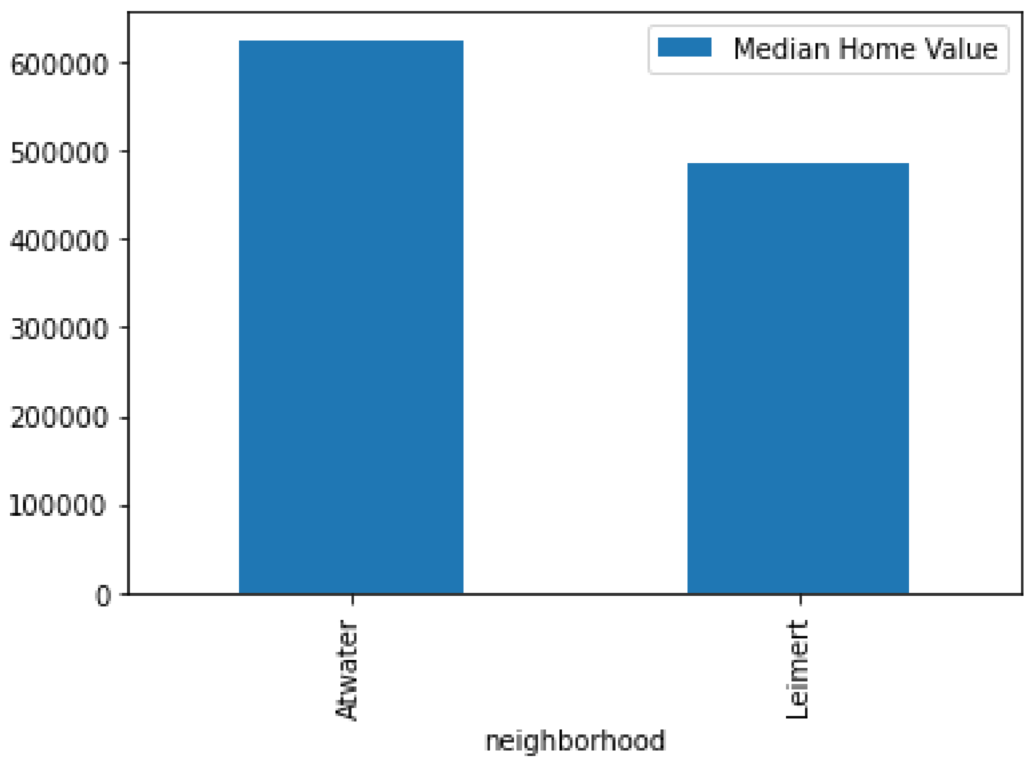
[232]:
```
df_mergedValue.head()
```

[232]:

	neighborhood	Median Home Value
0	Atwater	624881.25
0	Leimert	484000.00

[233]:
```
df_mergedValue.plot.bar(x = 'neighborhood', y='Median Home Value')
```

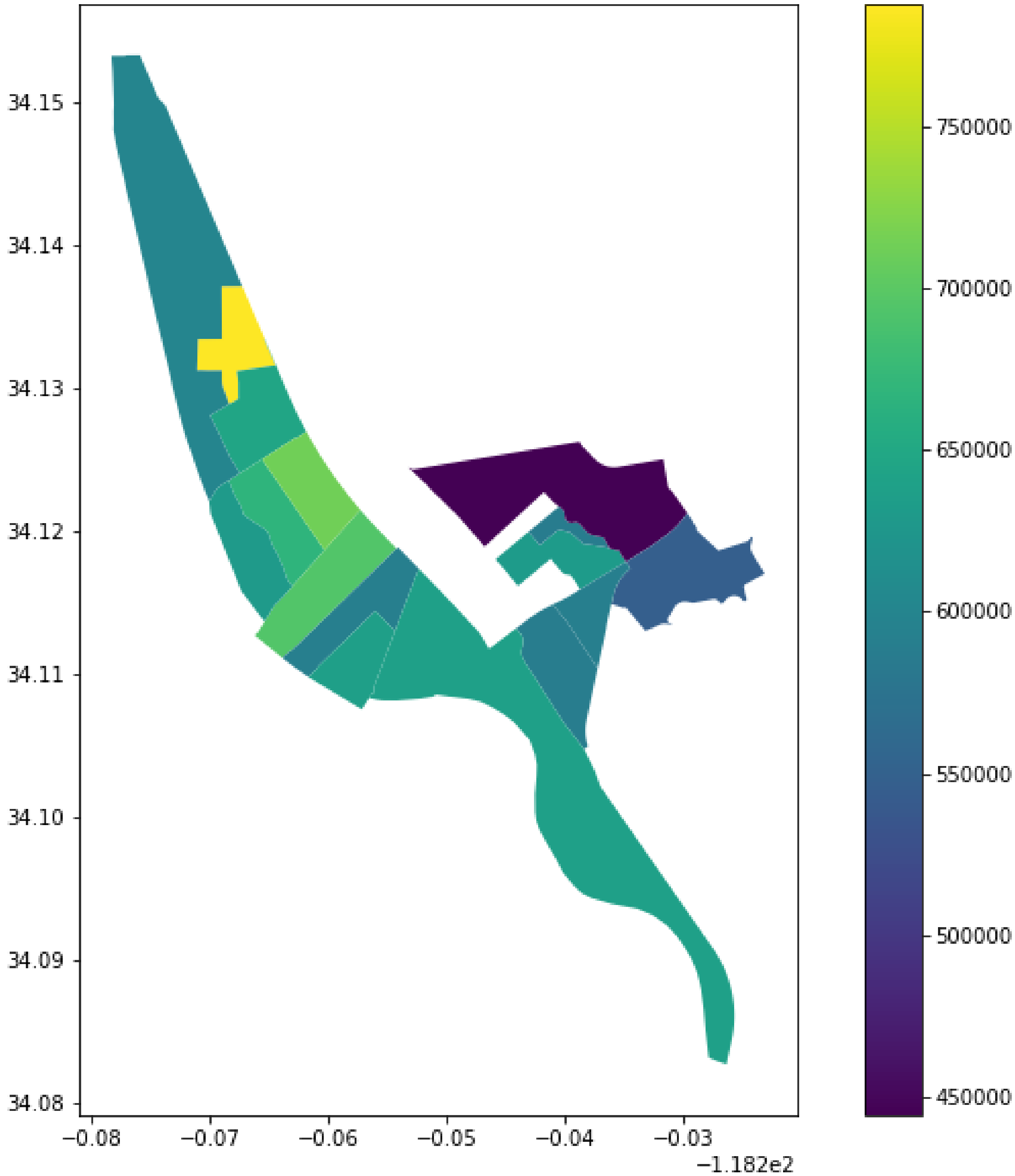
[233]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f379f910>



[234]:
```
# create a new dataframe based on the join (ATWATER)
blocks_units_value=blocks.merge(df5,on="FIPS")
```

[235]:
```
blocks_units_value[blocks_units.neighborhood=='Atwater'].plot(figsize=(12,10),
                 column='Median Home Value',
                 legend=True,)
```

[235]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f3740460>

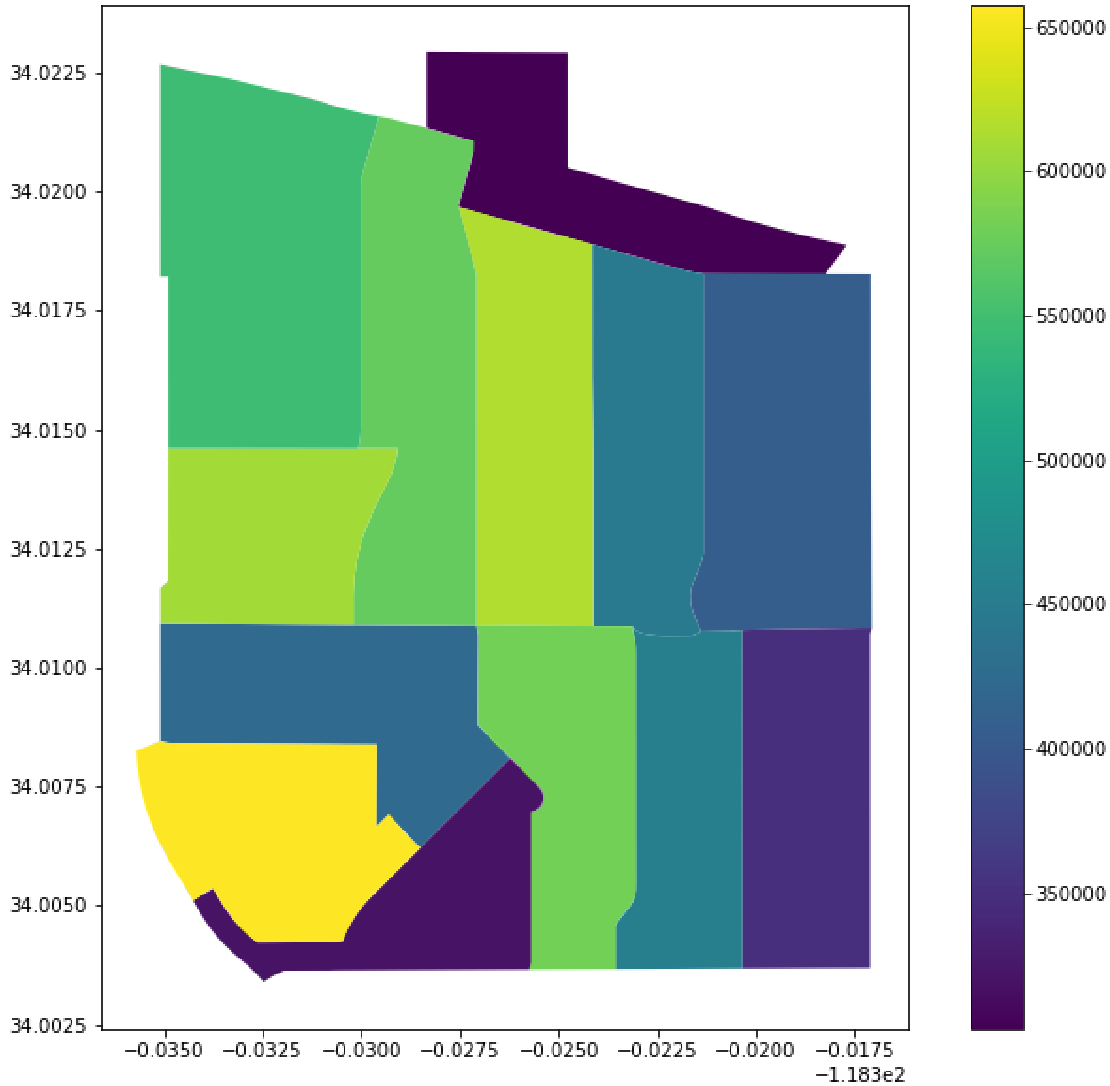


[236]:
```
# create a new dataframe based on the join (LEIMERT)
blocks_units_valueL=blocks.merge(df6,on="FIPS")
```

[238]:
```
blocks_units_valueL[blocks_units_valueL.neighborhood=='Leimert'].
 ↪plot(figsize=(12,10),
                     column='Median Home Value',
```

```
                    legend=True,)
```

[238]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f36ac9a0>



2.5 More Maps!

To place the neighborhoods of Atwater Village and Leimert Park into a better context relative to their own neighborhoods, we will use open street maps to understand other land uses in these respective areas.

[239]:
```
# to download osm data
import osmnx as ox
```

```
# to manipulate data
import pandas as pd

# to manipulate and visualize spatial data
import geopandas as gpd

# to provide basemaps
import contextily as ctx
```

[240]:
```
address1 = 'Leimert Park, Los Angeles, CA'
```

[241]:
```
%%time
# %%time is a magic command to see how long it takes this cell to run
# jeff has written - -based on an address, FIND ME buildings 1000m from address

# get the data from OSM that are tagged as 'building' for a 1000m X 1000m␣
↪square area
osm = ox.geometries_from_address(address1,tags={'building':True},dist=1000)
```

```
CPU times: user 1.63 s, sys: 139 ms, total: 1.77 s
Wall time: 15.4 s
```

[242]:
```
osm.shape
```

[242]: (7627, 50)

[243]:
```
type(osm)
```

[243]: geopandas.geodataframe.GeoDataFrame

[244]:
```
columns_to_keep = ['geometry','building']
osm = osm[columns_to_keep]
osm.sample(10)
```

[244]:

	geometry	building
6429	POLYGON ((-118.32307 34.00494, -118.32305 34.0…	house
3791	POLYGON ((-118.33681 34.00705, -118.33687 34.0…	house
6614	POLYGON ((-118.32089 34.00537, -118.32089 34.0…	house
160	POLYGON ((-118.33129 34.00670, -118.33129 34.0…	residential
4357	POLYGON ((-118.34189 33.99966, -118.34184 33.9…	house
4307	POLYGON ((-118.33320 33.99957, -118.33313 33.9…	house
1705	POLYGON ((-118.32793 34.01372, -118.32793 34.0…	house
5845	POLYGON ((-118.32501 34.00512, -118.32501 34.0…	house
1281	POLYGON ((-118.32975 34.00355, -118.32975 34.0…	commercial
2918	POLYGON ((-118.33130 34.01374, -118.33129 34.0…	house

[245]:
```
osm_building_counts = osm.building.value_counts()
osm_building_counts
```

[245]:
```
house             5637
residential        976
apartments         560
commercial         149
yes                141
retail             130
school              16
farm_auxiliary       3
warehouse            3
shed                 3
roof                 2
church               2
industrial           2
hotel                1
office               1
hospital             1
Name: building, dtype: int64
```

[246]:
```
df_osm_building_types = pd.DataFrame(osm_building_counts)
df_osm_building_types
```

[246]:

	building
house	5637
residential	976
apartments	560
commercial	149
yes	141
retail	130
school	16
farm_auxiliary	3
warehouse	3
shed	3
roof	2
church	2
industrial	2
hotel	1
office	1
hospital	1

[248]:
```
df_osm_building_types = df_osm_building_types.reset_index()
df_osm_building_types
```

[248]:

	index	building
0	house	5637

```
1      residential      976
2       apartments      560
3       commercial      149
4              yes      141
5           retail      130
6           school       16
7    farm_auxiliary        3
8        warehouse        3
9             shed        3
10            roof        2
11          church        2
12      industrial        2
13           hotel        1
14          office        1
15        hospital        1
```

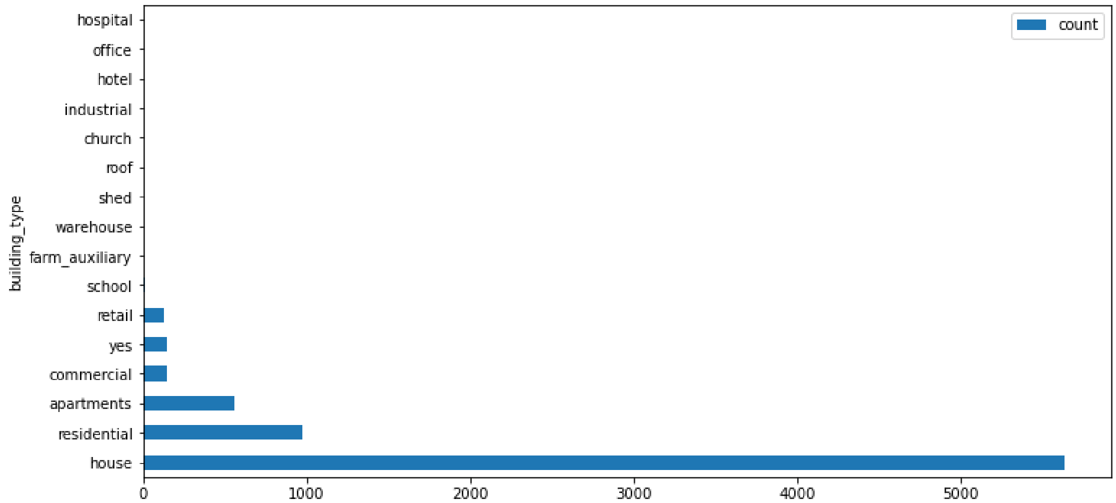
[249]:
```
df_osm_building_types.columns = ['building_type','count']
df_osm_building_types
```

[249]:

	building_type	count
0	house	5637
1	residential	976
2	apartments	560
3	commercial	149
4	yes	141
5	retail	130
6	school	16
7	farm_auxiliary	3
8	warehouse	3
9	shed	3
10	roof	2
11	church	2
12	industrial	2
13	hotel	1
14	office	1
15	hospital	1

[251]:
```
df_osm_building_types.plot.barh(figsize=(12,6),
                                x='building_type')
```

[251]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17ef527670>



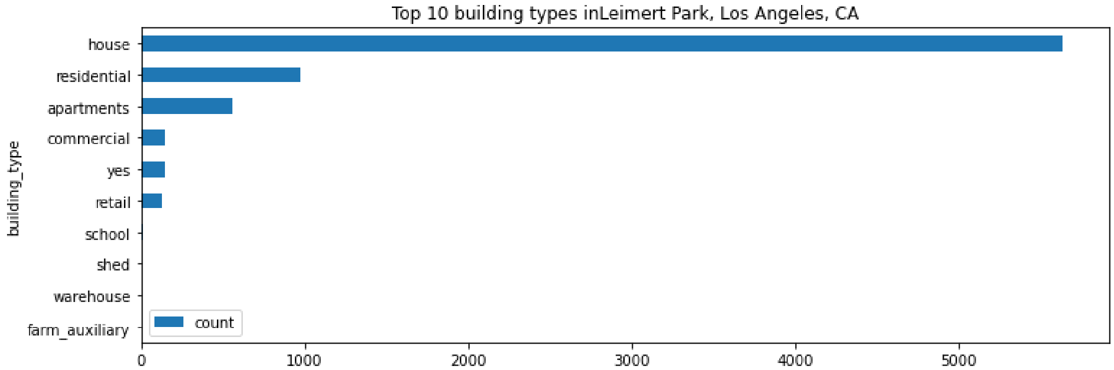
[252]:
```
df_osm_building_types = df_osm_building_types.sort_values(by='count',␣
↪ascending=True)
df_osm_building_types
```

[252]:

	building_type	count
13	hotel	1
14	office	1
15	hospital	1
10	roof	2
11	church	2
12	industrial	2
7	farm_auxiliary	3
8	warehouse	3
9	shed	3
6	school	16
5	retail	130
4	yes	141
3	commercial	149
2	apartments	560
1	residential	976
0	house	5637

[253]:
```
df_osm_building_types[-10:].plot.barh(figsize=(12,4),
                                      x='building_type',
                                      y='count',
                                      title="Top 10 building types in"+address1)
```

[253]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17ef406be0>



[254]:
```python
type(osm)
address1 = 'Leimert Park, CA'
```

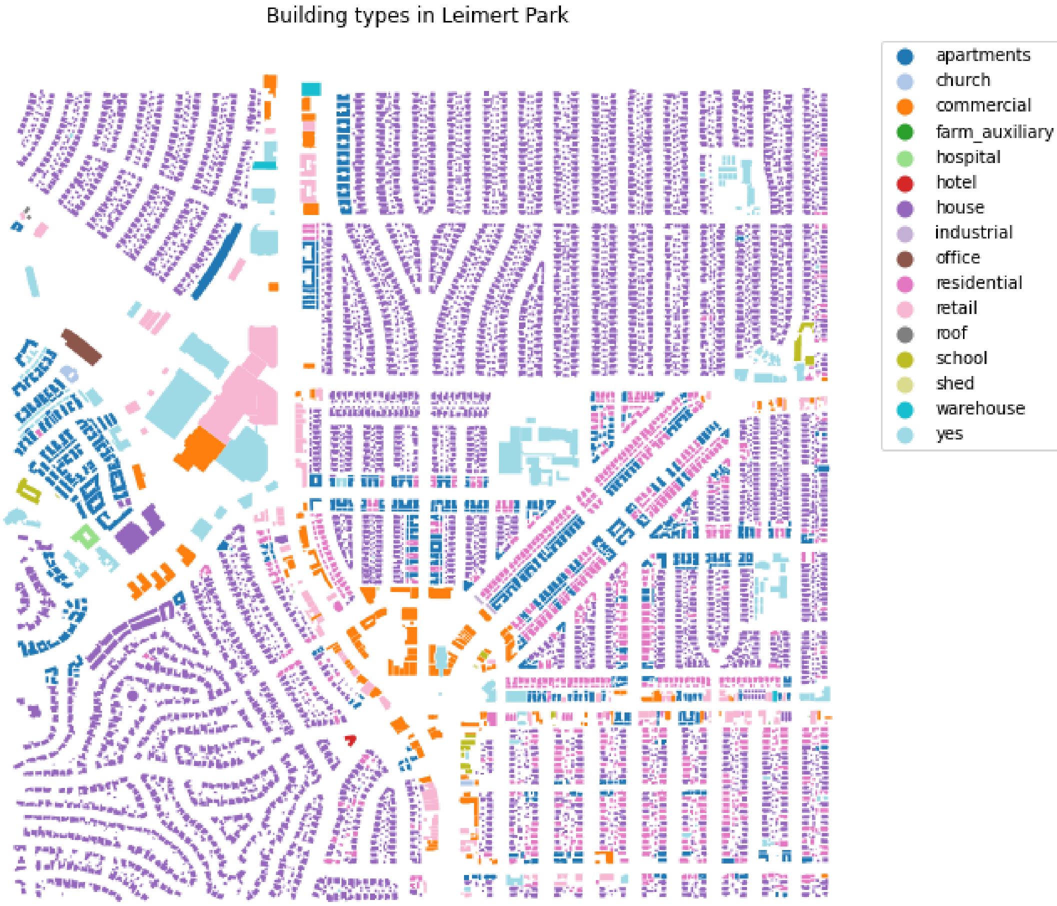
[256]:
```python
# plot entire dataset
ax1 = osm.plot(figsize=(10,10))
```


```
[258]: ax1 = osm.plot(figsize=(10,10),
                  column='building',
                  cmap='tab20',
                  legend=True)
```

[259]:

```
# create the map plot
ax1 = osm.plot(figsize=(10,10),
          column='building',
          cmap='tab20',
          legend=True,
          legend_kwds={'loc':'upper left','bbox_to_anchor':(1,1)})

# additional attributes to the map plot

# add a title
ax1.set_title('Building types in Leimert Park')

# get rid of the axis
ax1.axis('off')
```

[259]: (-118.34398192999998, -118.31929946999999, 33.998351435000004, 34.018788665)



[260]:
```
ax = osm.plot(figsize=(10,10),
         column='building',
         cmap='tab20',
         legend=True)
```

[261]:
```
address2 = 'Atwater Village, Los Angeles, CA'
```

[262]:
```
osm2 = ox.geometries_from_address(address2,tags={'building':True},dist=1000)
```

[263]:
```
osm2.shape
```

[263]: (5940, 44)

[264]:
```
type(osm2)
```

[264]: geopandas.geodataframe.GeoDataFrame

[265]:
```
osm2.sample(10)
```

[265]:
```
           unique_id      osmid element_type amenity  \
5491  way/429029621  429029621          way     NaN
1752  way/427752258  427752258          way     NaN
303   way/427563158  427563158          way     NaN
1886  way/427752481  427752481          way     NaN
4203  way/428415850  428415850          way     NaN
1075  way/427565123  427565123          way     NaN
3570  way/428412507  428412507          way     NaN
1744  way/427752227  427752227          way     NaN
5806  way/429029999  429029999          way     NaN
3444  way/428412308  428412308          way     NaN

                                               geometry  \
5491  POLYGON ((-118.25279 34.11104, -118.25283 34.1…
1752  POLYGON ((-118.26488 34.11694, -118.26489 34.1…
303   POLYGON ((-118.25491 34.11799, -118.25494 34.1…
1886  POLYGON ((-118.26591 34.12015, -118.26591 34.1…
4203  POLYGON ((-118.26024 34.11983, -118.26020 34.1…
1075  POLYGON ((-118.25525 34.11223, -118.25524 34.1…
3570  POLYGON ((-118.26716 34.12357, -118.26720 34.1…
1744  POLYGON ((-118.26525 34.11868, -118.26522 34.1…
5806  POLYGON ((-118.25300 34.11306, -118.25292 34.1…
3444  POLYGON ((-118.26625 34.12229, -118.26625 34.1…

                                                  nodes addr:city  \
5491  [4281569798, 4281569749, 4281569823, 428156988…       NaN
1752  [4269374192, 4269374083, 4269374079, 426937407…       NaN
303   [4267657682, 4267657677, 4267657671, 426765765…       NaN
1886  [4269378559, 4269378556, 4269378392, 426937839…       NaN
4203  [4275173426, 4275173396, 4275173393, 427517337…       NaN
1075  [4267669663, 4267669651, 4267669660, 426766963…       NaN
3570  [4275145211, 4275145268, 4275145327, 427514527…       NaN
1744  [4269376949, 4269376916, 4269376909, 426937690…       NaN
5806  [4281574830, 4281574707, 4281574629, 428157463…       NaN
3444  [4275143773, 4275143780, 4275143812, 427514380…       NaN

     addr:housenumber addr:postcode addr:state  … shop smoking layer phone  \
5491              NaN           NaN        NaN  …  NaN     NaN   NaN   NaN
1752              NaN           NaN        NaN  …  NaN     NaN   NaN   NaN
303               NaN           NaN        NaN  …  NaN     NaN   NaN   NaN
1886              NaN           NaN        NaN  …  NaN     NaN   NaN   NaN
4203              NaN           NaN        NaN  …  NaN     NaN   NaN   NaN
1075              NaN           NaN        NaN  …  NaN     NaN   NaN   NaN
3570              NaN           NaN        NaN  …  NaN     NaN   NaN   NaN
1744              NaN           NaN        NaN  …  NaN     NaN   NaN   NaN
5806              NaN           NaN        NaN  …  NaN     NaN   NaN   NaN
3444              NaN           NaN        NaN  …  NaN     NaN   NaN   NaN
```

```
     building:use tourism healthcare source:start_date ways type
5491          NaN     NaN        NaN               NaN  NaN  NaN
1752          NaN     NaN        NaN               NaN  NaN  NaN
303           NaN     NaN        NaN               NaN  NaN  NaN
1886          NaN     NaN        NaN               NaN  NaN  NaN
4203          NaN     NaN        NaN               NaN  NaN  NaN
1075          NaN     NaN        NaN               NaN  NaN  NaN
3570          NaN     NaN        NaN               NaN  NaN  NaN
1744          NaN     NaN        NaN               NaN  NaN  NaN
5806          NaN     NaN        NaN               NaN  NaN  NaN
3444          NaN     NaN        NaN               NaN  NaN  NaN

[10 rows x 44 columns]
```

[266]:
```
list(osm2)
```

[266]:
```
['unique_id',
 'osmid',
 'element_type',
 'amenity',
 'geometry',
 'nodes',
 'addr:city',
 'addr:housenumber',
 'addr:postcode',
 'addr:state',
 'addr:street',
 'building',
 'cuisine',
 'ele',
 'height',
 'lacounty:ain',
 'lacounty:bld_id',
 'name',
 'start_date',
 'office',
 'building:units',
 'operator',
 'source',
 'brand',
 'brand:wikidata',
 'brand:wikipedia',
 'takeaway',
 'website',
 'building:levels',
 'description',
```

```
 'opening_hours',
 'payment:cash',
 'payment:credit_cards',
 'second_hand',
 'shop',
 'smoking',
 'layer',
 'phone',
 'building:use',
 'tourism',
 'healthcare',
 'source:start_date',
 'ways',
 'type']
```

[267]:
```
columns_to_keep2 = ['geometry','building']
osm2 = osm2[columns_to_keep2]
osm.sample(10)
```

[267]:

	geometry	building
1924	POLYGON ((-118.33219 34.01501, -118.33219 34.0...	house
4658	POLYGON ((-118.32669 34.01205, -118.32669 34.0...	house
4068	POLYGON ((-118.33860 34.00294, -118.33862 34.0...	house
6174	POLYGON ((-118.32349 34.00590, -118.32355 34.0...	house
6188	POLYGON ((-118.32349 34.00873, -118.32354 34.0...	apartments
3897	POLYGON ((-118.33827 34.00306, -118.33826 34.0...	house
2180	POLYGON ((-118.34001 34.01424, -118.34009 34.0...	house
4374	POLYGON ((-118.33148 33.99976, -118.33142 33.9...	retail
4404	POLYGON ((-118.32655 33.99954, -118.32659 33.9...	house
4590	POLYGON ((-118.32593 34.01287, -118.32572 34.0...	house

[268]:
```
osm_building_counts2 = osm2.building.value_counts()
osm_building_counts2
```

[268]:
```
house          3399
residential    1303
apartments      571
yes             214
industrial      166
commercial      113
retail          102
warehouse        57
factory           4
kindergarten      3
hotel             3
greenhouse        2
roof              1
```

```
school              1
train_station       1
Name: building, dtype: int64
```

[269]:
```
# series is a one dimensial (can not change vlues of, only one row of values)
↪want to convert to a data frame
type(osm_building_counts2)
```

[269]:
```
pandas.core.series.Series
```

[270]:
```
df_osm_building_types2 = pd.DataFrame(osm_building_counts2)
df_osm_building_types2
```

[270]:

	building
house	3399
residential	1303
apartments	571
yes	214
industrial	166
commercial	113
retail	102
warehouse	57
factory	4
kindergarten	3
hotel	3
greenhouse	2
roof	1
school	1
train_station	1

[271]:
```
df_osm_building_types2 = df_osm_building_types2.reset_index()
df_osm_building_types2
```

[271]:

	index	building
0	house	3399
1	residential	1303
2	apartments	571
3	yes	214
4	industrial	166
5	commercial	113
6	retail	102
7	warehouse	57
8	factory	4
9	kindergarten	3
10	hotel	3
11	greenhouse	2
12	roof	1

```
13         school         1
14  train_station         1
```

[272]:
```
df_osm_building_types2.columns = ['building_type','count']
df_osm_building_types2
```

[272]:
```
    building_type  count
0           house   3399
1     residential   1303
2      apartments    571
3             yes    214
4      industrial    166
5      commercial    113
6          retail    102
7       warehouse     57
8         factory      4
9    kindergarten      3
10          hotel      3
11     greenhouse      2
12           roof      1
13         school      1
14  train_station      1
```

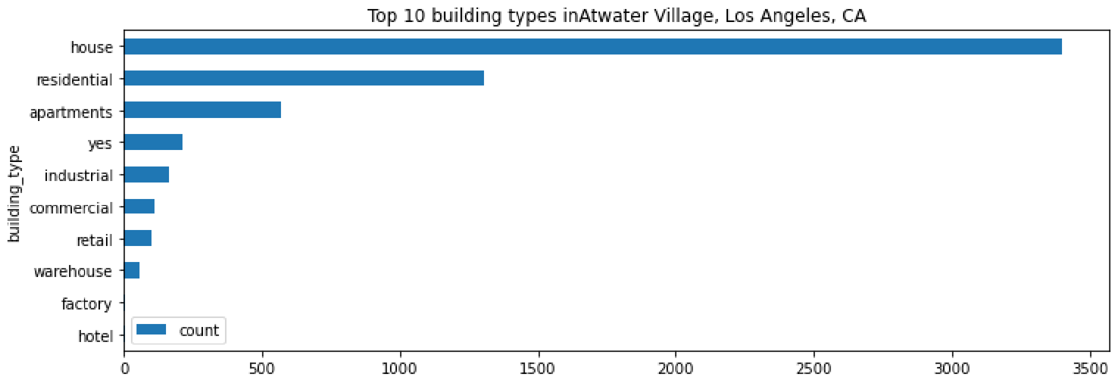
[275]:
```
df_osm_building_types2 = df_osm_building_types2.sort_values(by='count',␣
↪ascending=True)
df_osm_building_types2
```

[275]:
```
    building_type  count
12           roof      1
13         school      1
14  train_station      1
11     greenhouse      2
9    kindergarten      3
10          hotel      3
8         factory      4
7       warehouse     57
6          retail    102
5      commercial    113
4      industrial    166
3             yes    214
2      apartments    571
1     residential   1303
0           house   3399
```

[277]:
```
df_osm_building_types2[-10:].plot.barh(figsize=(12,4),
                                       x='building_type',
                                       y='count',
```

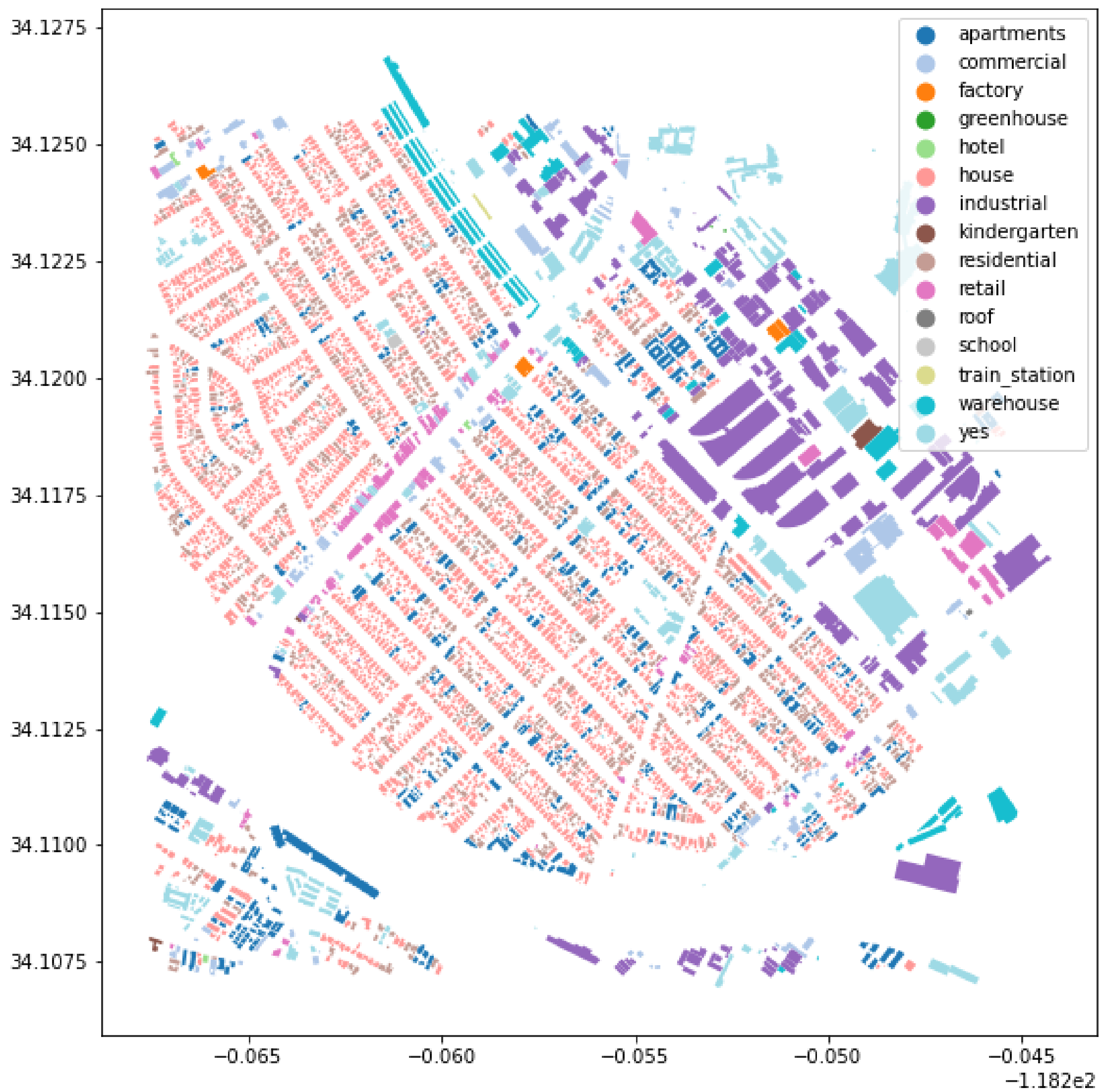
```
title="Top 10 building types in"+address2)
```

[277]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f0536a00>



[278]:
```
ax2 = osm2.plot(figsize=(10,10))
```

[279]:
```
ax2 = osm2.plot(figsize=(10,10),
          column='building',
          cmap='tab20',
          legend=True)
```

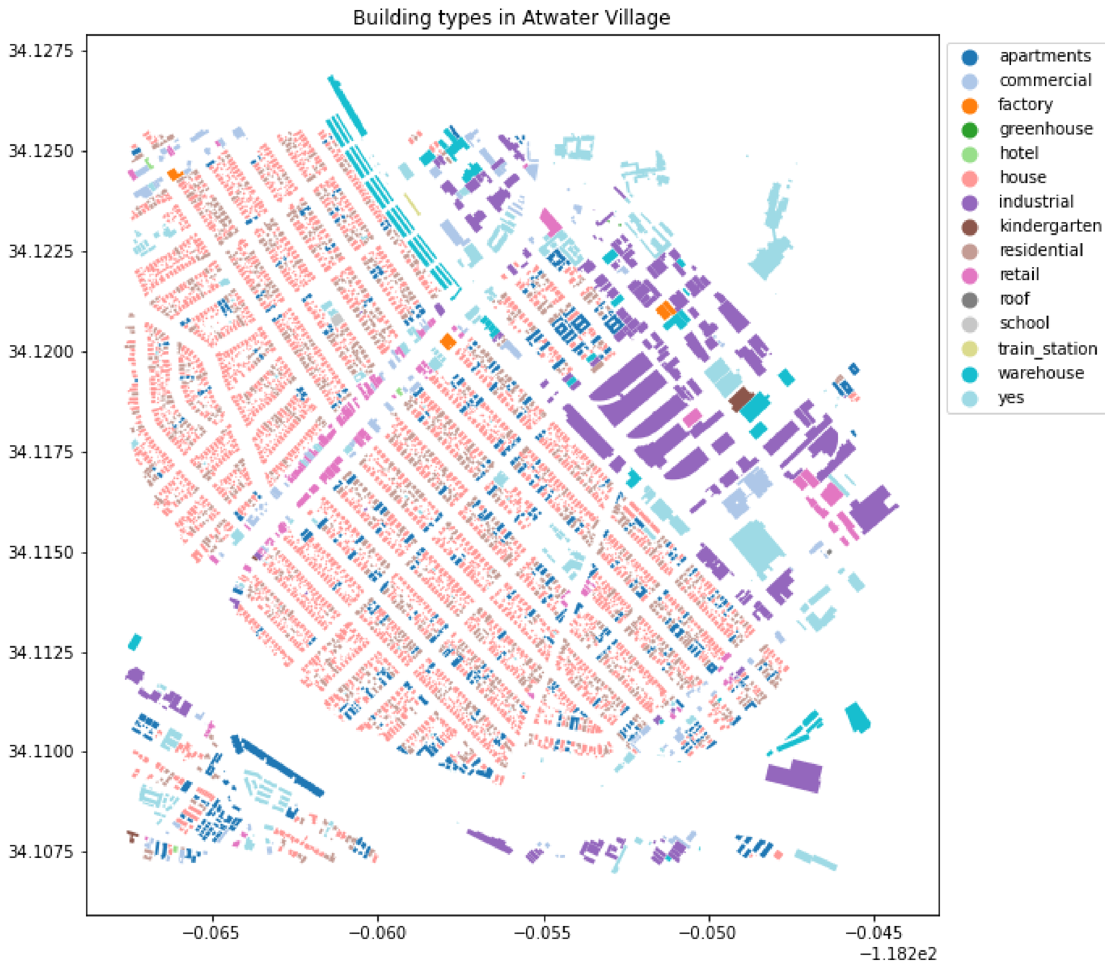


[281]:

```python
# create the map plot for Atwater
ax2 = osm2.plot(figsize=(10,10),
         column='building',
         cmap='tab20',
         legend=True,
         legend_kwds={'loc':'upper left','bbox_to_anchor':(1,1)})

# additional attributes to the map plot

# add a title
ax2.set_title('Building types in Atwater Village')

# get rid of the axis
```

```
ax.axis('off')
```

[281]: (-118.34398192999998, -118.31929946999999, 33.998351435000004, 34.018788665)



2.6 ADU Permit Data

[282]:
```
##First we want to review our data from the LA Data Portal for ADU Construction
```

[283]:
```
import pandas as pd
import plotly.express as px
from sodapy import Socrata
import geopandas as gpd
```

[284]:
```
adu = gpd.read_file('https://data.lacity.org/resource/hyem-e7yr.geojson')

adu.head()
```

[284]:
```
  assessor_parcel zip_code location_1_address  \
0             034    91367               None
1             046    91316               None
2             024    90025               None
3             014    90034               None
4             027    91436

                                   work_description  \
0  NEW FIRE SPRINKLER SYSTEM FOR ADU  PER NFPA 13…
1  NFPA13D FOR ADU.  EXISTING 1'' DOMESTIC WATER …
2  NFPA 13D SYSTEM . 1" DOMESTIC METER SEVRVES TH…
3  NEW FIRE SPRINKLER SYSTEM FOR PER NFPA 13D FOR…
4  New fire sprinkler system for ADU per NFPA-13D…

  :@computed_region_2dna_qi2s applicant_address_3  \
0                        None          ARLETA, CA
1                        None      SUN VALLEY, CA
2                        None                None
3                        None  WOODLAND HILLS, CA
4                          62        WEST HILLS,CA

  floor_area_l_a_zoning_code_definition address_fraction_end project_number  \
0                                  None                 None           None
1                                  None                 None           None
2                                  None                 None           None
3                                  None                 None           None
4                                  None                 None           None

  suffix_direction  … event_code reference_old_permit  \
0             None  …       None                 None
1             None  …       None                 None
2             None  …       None                 None
3             None  …       None                 None
4             None  …       None                 None

  applicant_relationship :@computed_region_k96s_3jcv contractor_state  \
0             Contractor                        None               CA
1             Contractor                        None               CA
2             Contractor                        None               CA
3             Contractor                        None               CA
4   Agent for Contractor                         327               CA

  license_expiration_date :@computed_region_qz3q_ghft applicant_address_2  \
0     2021-06-30T00:00:00                        None                None
1     2021-12-31T00:00:00                        None              UNIT G
2     2021-10-31T00:00:00                        None                None
3     2021-01-31T00:00:00                        None                None
```

```
4     2021-10-31T00:00:00                         19737                None

          permit_sub_type                    geometry
0  1 or 2 Family Dwelling                        None
1  1 or 2 Family Dwelling                        None
2  1 or 2 Family Dwelling                        None
3  1 or 2 Family Dwelling                        None
4  1 or 2 Family Dwelling  POINT (-118.49822 34.14598)

[5 rows x 65 columns]
```

[285]:
```
adu = adu[['issue_date','geometry']]

# print it with .sample, which gives you random rows
adu.head()
```

[285]:
```
            issue_date                     geometry
0  2020-12-03T00:00:00                         None
1  2020-10-30T00:00:00                         None
2  2020-10-27T00:00:00                         None
3  2020-09-30T00:00:00                         None
4  2020-09-18T00:00:00  POINT (-118.49822 34.14598)
```

[286]:
```
##To further understand the ADU data set we can look at the shape and its␣
 ↪columns
list(adu)
```

[286]:
```
['issue_date', 'geometry']
```

[287]:
```
#Now that we have our ADU Data we can use Geopandas to allow us to convert␣
 ↪different types of data into a spatial format.
adu.crs
```

[287]:
```
<Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

[288]:
```
#We can a latitude and longtitude column so that we can map it
adu['x'] = adu.geometry.x
```

```
adu['y'] = adu.geometry.y
```

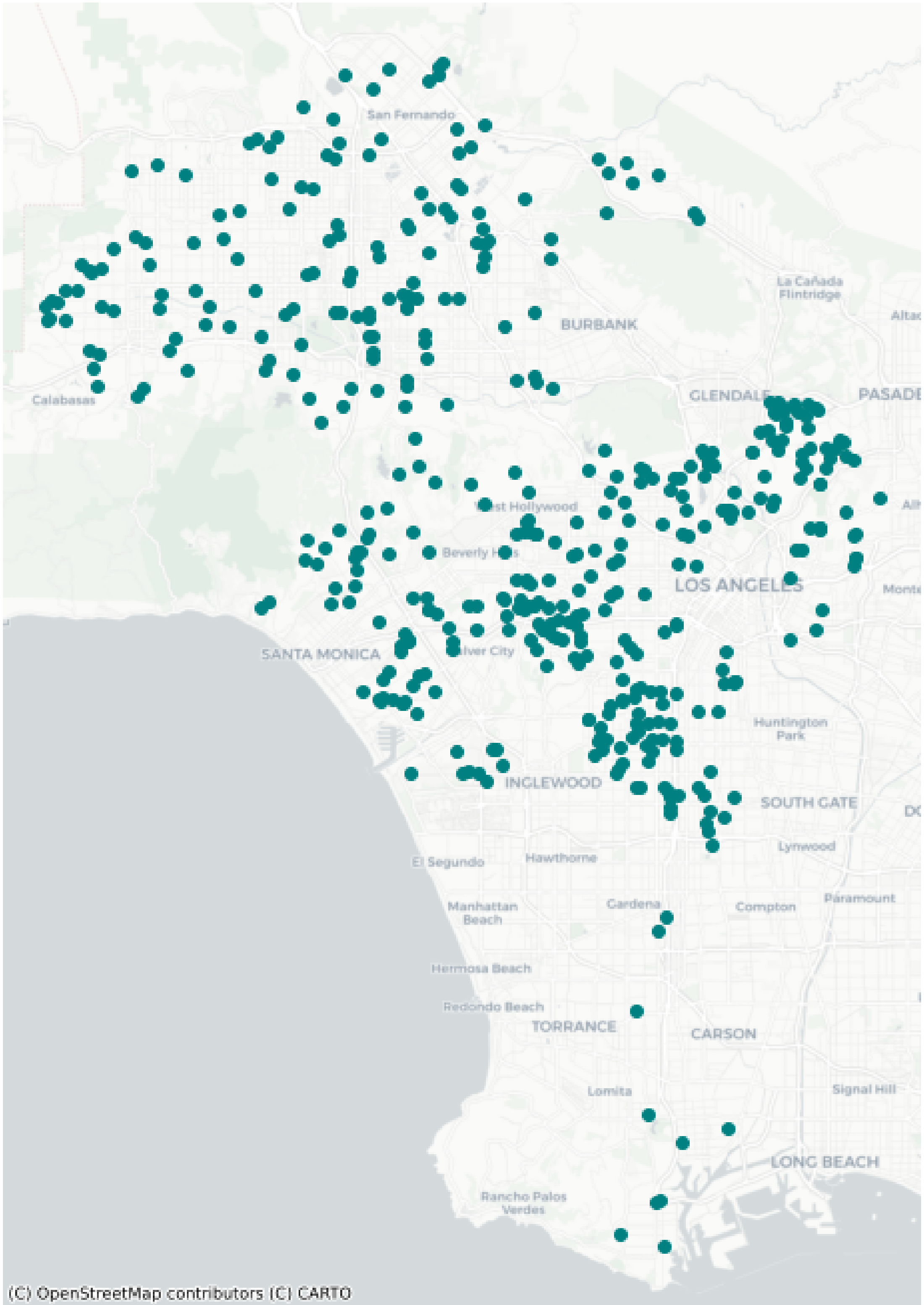
[289]:
```
adu.head()
```

[289]:
```
            issue_date                      geometry          x         y
0  2020-12-03T00:00:00                          None        NaN       NaN
1  2020-10-30T00:00:00                          None        NaN       NaN
2  2020-10-27T00:00:00                          None        NaN       NaN
3  2020-09-30T00:00:00                          None        NaN       NaN
4  2020-09-18T00:00:00  POINT (-118.49822 34.14598) -118.49822  34.14598
```

[290]:
```
adu = adu.dropna()
```

[291]:
```
# get the layers into a web mercator projection
# reproject to web mercator
adu = adu.to_crs('EPSG:3857')
```

[292]:
```
# map it!
ax = adu.plot(figsize=(12,12),color='teal')

# no axis
ax.axis('off')

# add a basemap
ctx.add_basemap(ax,source=ctx.providers.CartoDB.Positron)
```



San Fernando
BURBANK
La Cañada Flintridge
GLENDALE
Calabasas
West Hollywood
Beverly Hills
LOS ANGELES
SANTA MONICA
Culver City
Huntington Park
INGLEWOOD
SOUTH GATE
Lynwood
El Segundo
Hawthorne
Manhattan Beach
Gardena
Compton
Paramount
Hermosa Beach
Redondo Beach
TORRANCE
CARSON
Lomita
Signal Hill
LONG BEACH
Rancho Palos Verdes
(C) OpenStreetMap contributors (C) CARTO

[293]:
```
#Now we map these ADU points along a base map! But we have to define this base␣
↪map first so we get neighborhood boundaries from the LA Times
neighborhoods = gpd.read_file('http://s3-us-west-2.amazonaws.com/boundaries.
↪latimes.com/archive/1.0/boundary-set/la-county-neighborhoods-v5.geojson')
```

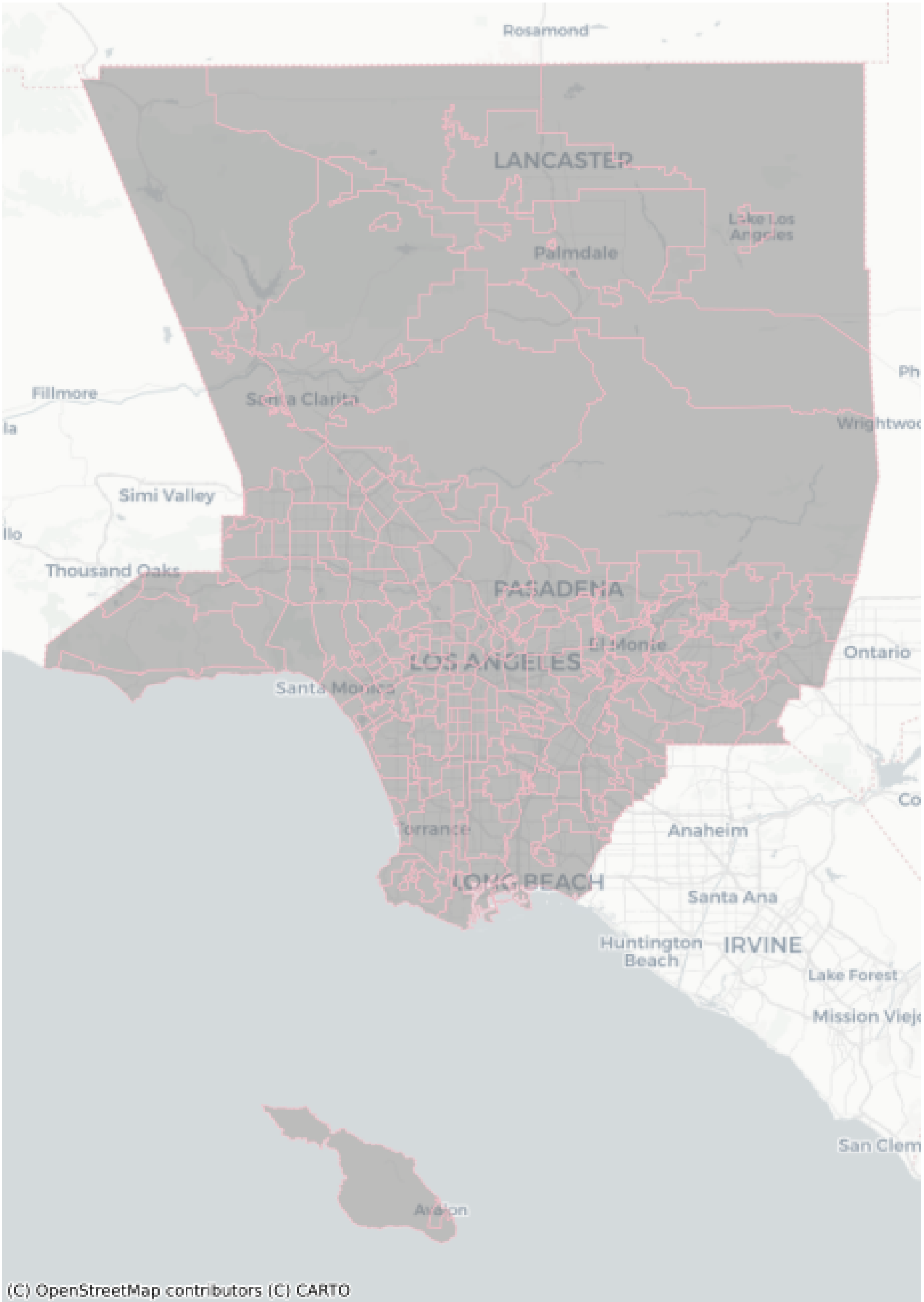
[294]:
```
# trim the data to the bare minimum columns
neighborhoods = neighborhoods[['name','geometry']]
neighborhoods.head()
```

[294]:

	name	geometry
0	Acton	MULTIPOLYGON (((-118.20262 34.53899, -118.1894…
1	Adams-Normandie	MULTIPOLYGON (((-118.30901 34.03741, -118.3004…
2	Agoura Hills	MULTIPOLYGON (((-118.76193 34.16820, -118.7263…
3	Agua Dulce	MULTIPOLYGON (((-118.25468 34.55830, -118.2555…
4	Alhambra	MULTIPOLYGON (((-118.12175 34.10504, -118.1168…

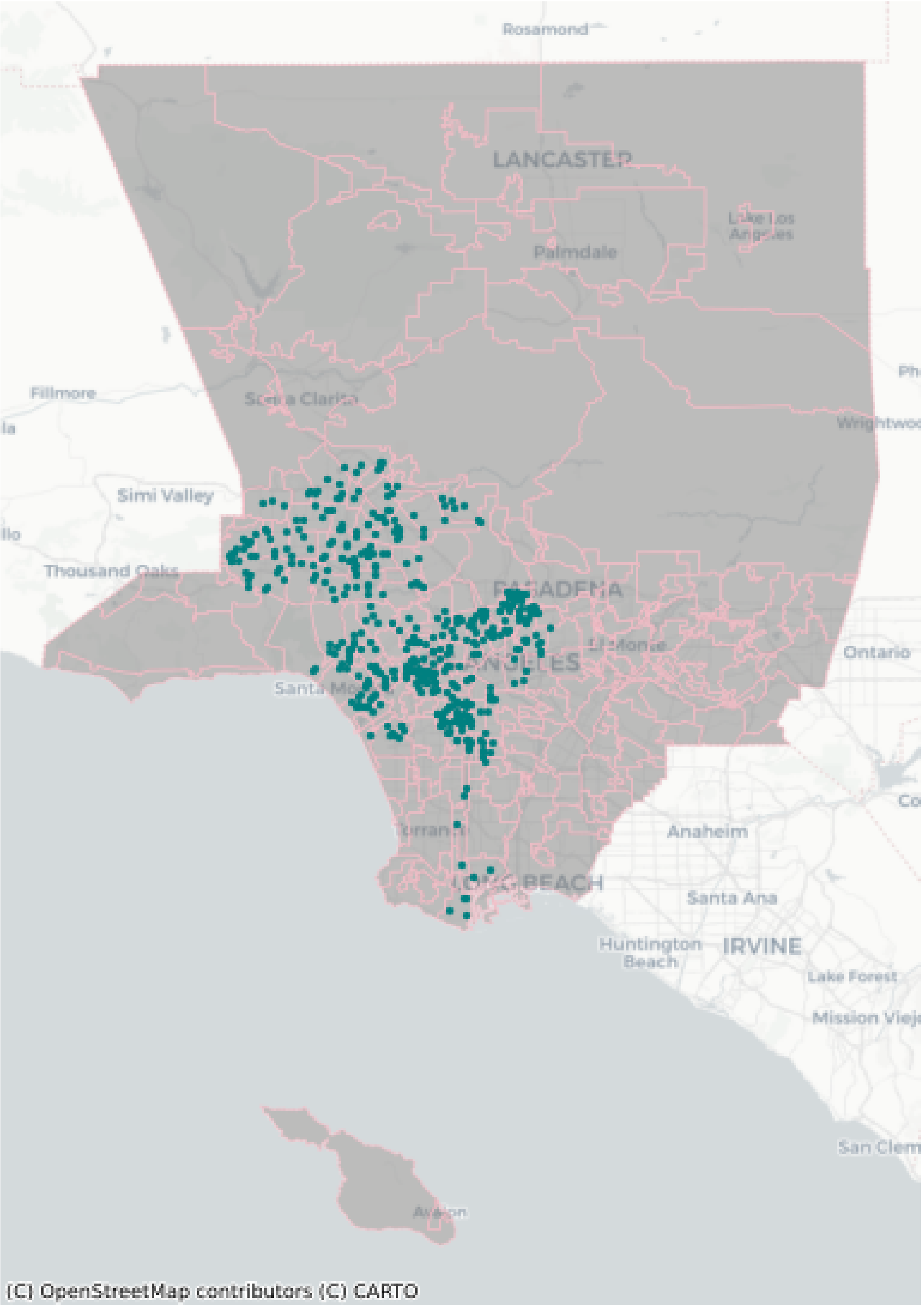
[295]:
```
# get the layers into a web mercator projection
# reproject to web mercator to use contextily library
neighborhoods = neighborhoods.to_crs(epsg=3857)
```

[296]:
```
# plot it!
ax=neighborhoods.plot(figsize=(12,12),
                      color='gray',
                      edgecolor='pink',
                      alpha=0.5)

# no axis
ax.axis('off')

# add a basemap
ctx.add_basemap(ax,source=ctx.providers.CartoDB.Positron)
```


Rosamond
LANCASTER
Lake Los Angeles
Palmdale
Fillmore
Santa Clarita
Wrightwood
Simi Valley
Thousand Oaks
PASADENA
El Monte
LOS ANGELES
Ontario
Santa Monica
Anaheim
Torrance
LONG BEACH
Santa Ana
Huntington Beach
IRVINE
Lake Forest
Avalon
(C) OpenStreetMap contributors (C) CARTO

2.7 Two Layer Map

[297]:
```
# first define which layers will be your "base"
base = neighborhoods.plot(figsize=(12,10),
                          color='gray',
                          edgecolor='pink',
                          alpha=0.5)

# define the layer that will go on top, and add the base layer to the `ax`␣
↪argument
ax = adu.plot(ax=base, color='teal', markersize=5)

# no axis
ax.axis('off')

# add a basemap
ctx.add_basemap(ax,source=ctx.providers.CartoDB.Positron)
```



[298]:

```
# get the bounding box coordinates for the adu data
adu.geometry.total_bounds
```

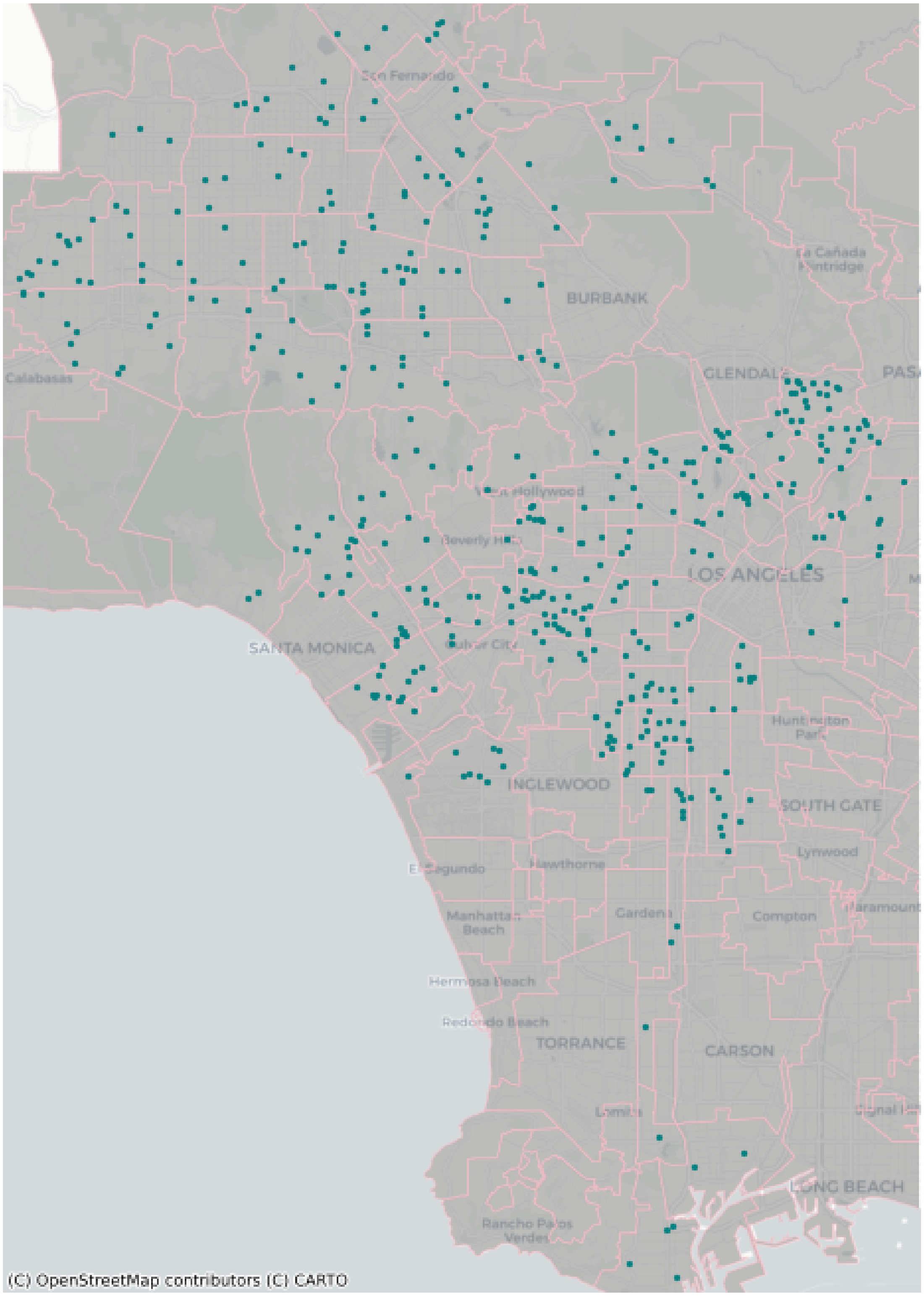
[298]: array([-13208653.14189765,   3992182.02224806, -13153433.10848965,
          4070591.39670432])

[299]:
```python
# shortcut to put them into their own variables
minx, miny, maxx, maxy = adu.geometry.total_bounds
print(minx)
print(maxx)
print(miny)
print(maxy)
```

```
-13208653.141897654
-13153433.108489651
3992182.022248056
4070591.396704316
```

[300]:
```python
# use the bounding box coordinates to set the x and y limits
base = neighborhoods.plot(figsize=(12,12),
                          color='gray',
                          edgecolor='pink',
                          alpha=0.5)

ax = adu.plot(ax=base,
              color='teal',
              markersize=5
             )

ax.set_xlim(minx - 1000, maxx + 1000) # added/substracted value is to give some␣
 ↪margin around total bounds
ax.set_ylim(miny - 1000, maxy + 1000)

# no axis
ax.axis('off')

# add a basemap
ctx.add_basemap(ax,source=ctx.providers.CartoDB.Positron)

ax
```

[300]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f062f760>


San Fernando
BURBANK
La Cañada
Flintridge
Calabasas
GLENDALE
West Hollywood
Beverly Hills
LOS ANGELES
SANTA MONICA
Culver City
Huntington
Park
INGLEWOOD
SOUTH GATE
Lynwood
El Segundo
Hawthorne
Manhattan
Beach
Gardena
Compton
Paramount
Hermosa Beach
Redondo Beach
TORRANCE
CARSON
Lomita
Signal Hill
LONG BEACH
Rancho Palos
Verdes
(C) OpenStreetMap contributors (C) CARTO

2.8 Function

Our research would like to look at ADU production at the neighborhood level in two distinct neighborhoods – Atwater Village and Leimert Park. By having a function that zooms in to these specific neighborhoods our data is more legible.

We will:

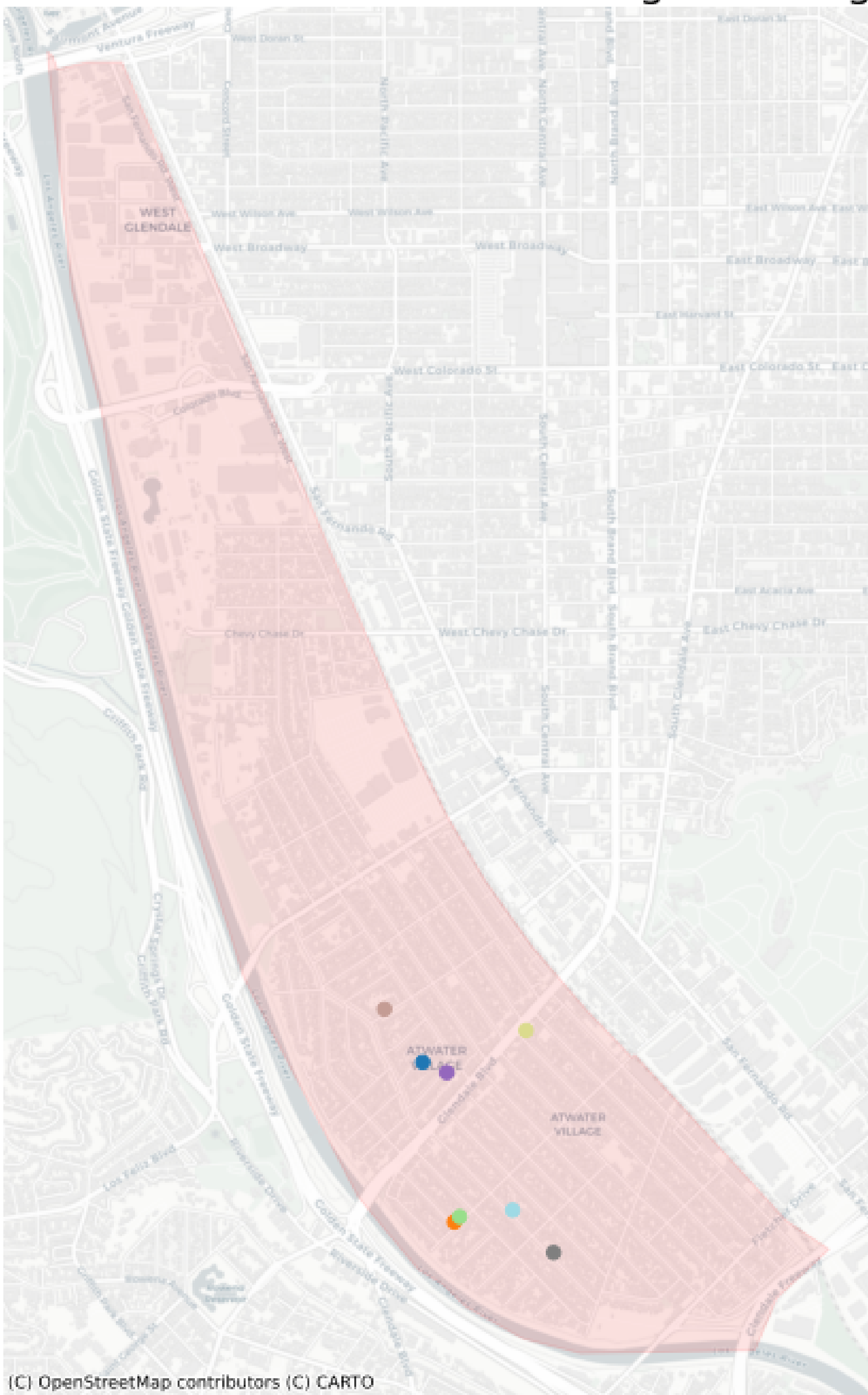
- zoom to the Atwater Village and Leimert Park neighborhoods
- show an outline of the neighborhood
- show ADU data
- add a legend for ADU type

[302]:
```
# subset the neighborhoods geodataframe for a single neighborhood
neighborhood = neighborhoods[neighborhoods.name=='Atwater Village']

# use the bounding box coordinates to set the x and y limits
minx, miny, maxx, maxy = neighborhood.geometry.total_bounds

# do a spatial join to get crime in neighborhood
adus_in_neighborhood = gpd.sjoin(adu,neighborhood,how='inner')

# define the base layer to be the neighborhood polygon
base = neighborhood.plot(figsize=(12,12),
                         color='red',
                         edgecolor='red',
                         alpha=0.1)

# add the crime data, making sure to add the neighborhood polygon
ax = adus_in_neighborhood.plot(ax=base,
#                                  column='zone',
                                markersize=40,
                                legend=True,
                                cmap='tab20',
                                legend_kwds={
                                   'loc': 'upper right',
                                   'bbox_to_anchor':(1.3,1)
                                }                  # this puts the legend to␣
 ↪the side
                            )

# set the map extent to the extent of the neighborhood bounds
ax.set_xlim(minx - 200, maxx + 200) # added/substracted value is to give some␣
 ↪margin around total bounds
ax.set_ylim(miny - 200, maxy + 200)

# turn off the axis
ax.axis('off')
```

```
# add a title
ax.set_title('ADUs Constructed in '+neighborhood.name.values[0]+' Los␣
↪Angeles',fontsize=20)

# add a basemap
ctx.add_basemap(ax,source=ctx.providers.CartoDB.Positron)
ax
```

[302]: <matplotlib.axes._subplots.AxesSubplot at 0x7f17f131beb0>

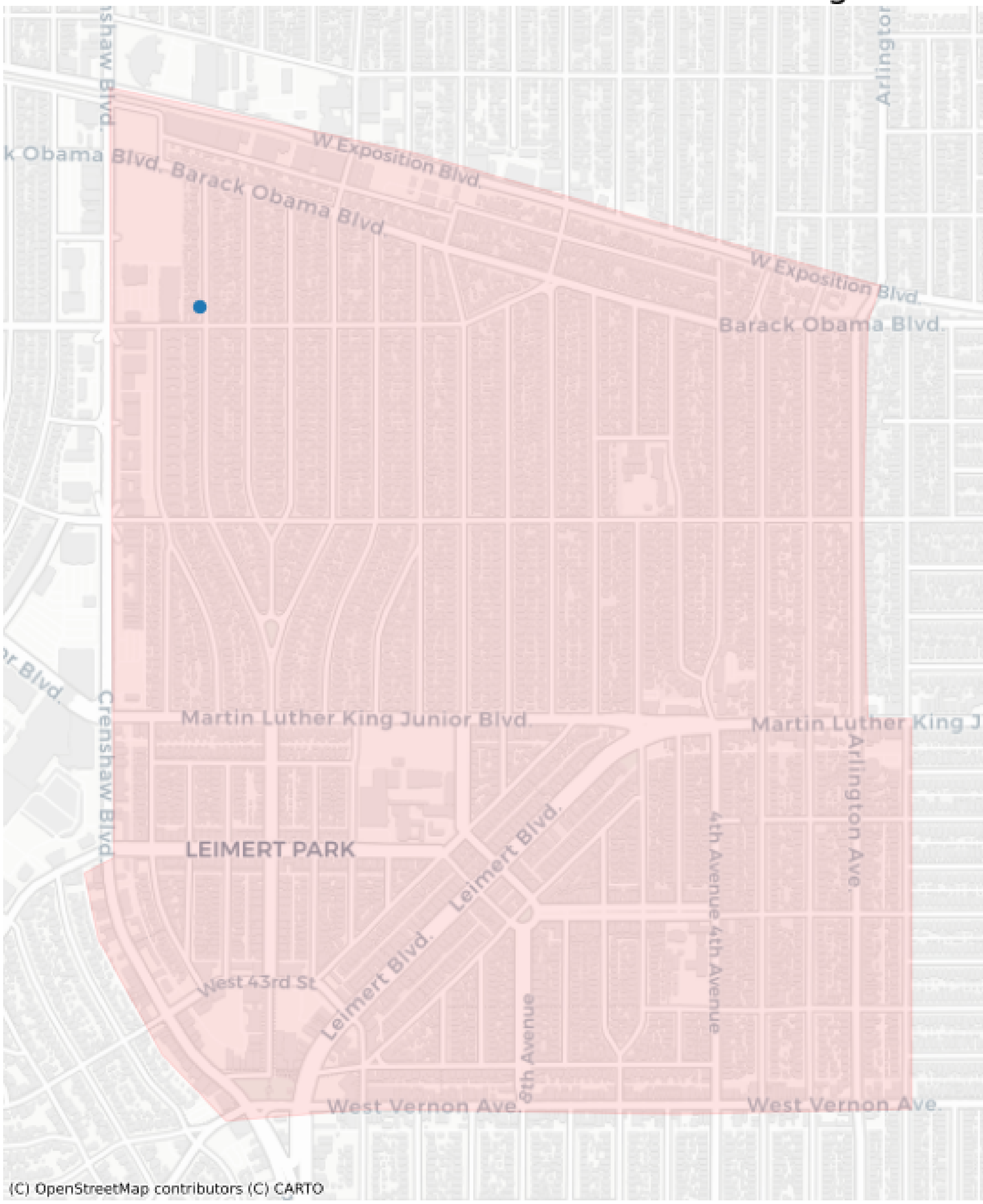
ADUs Constructed in Atwater Village Los Angeles



[303]:
```
# subset the neighborhoods geodataframe for a single neighborhood
neighborhood = neighborhoods[neighborhoods.name=='Leimert Park']

# use the bounding box coordinates to set the x and y limits
minx, miny, maxx, maxy = neighborhood.geometry.total_bounds

# do a spatial join to get crime in neighborhood
adus_in_neighborhood = gpd.sjoin(adu,neighborhood,how='inner')

# define the base layer to be the neighborhood polygon
base = neighborhood.plot(figsize=(12,12),
                         color='red',
                         edgecolor='red',
                         alpha=0.1)

# add the crime data, making sure to add the neighborhood polygon
ax = adus_in_neighborhood.plot(ax=base,
#                                  column='applicant_address_3',
                                markersize=40,
                                legend=True,
                                cmap='tab20',
                                legend_kwds={
                                   'loc': 'upper right',
                                   'bbox_to_anchor':(1.3,1)
                                }                  # this puts the legend to␣
 ↪the side
                            )

# set the map extent to the extent of the neighborhood bounds
ax.set_xlim(minx - 200, maxx + 200) # added/substracted value is to give some␣
 ↪margin around total bounds
ax.set_ylim(miny - 200, maxy + 200)

# turn off the axis
ax.axis('off')

# add a title
ax.set_title('ADUs Constructed in '+neighborhood.name.values[0]+' Los␣
 ↪Angeles',fontsize=20)

# add a basemap
ctx.add_basemap(ax,source=ctx.providers.CartoDB.Positron)
ax
```

[303]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f17f12bc670>
```

ADUs Constructed in Leimert Park Los Angeles




```
[ ]:
```