f21-final

December 6, 2021

1 How Accessible is Publicly Available Braodband in Los Angeles?

Michael Criste | Fall 2021 | Midterm

1.1 Project Proposal

revisit initial proposal here and update to reflect current ideas and deliverables.

How accessible is publicly available broadband access? Federal, state, and local governemnts leverage billions of dollars and partnerships with private-industry partners to make broadband more accessible and increase adoption rates at home. These programs often fund publicly accessible broadband through public schools and libraries to aid children and their families to get online in an effort to increase educational outcomes leading to more economic opportunities. A primary example is the FCC's E-RATE universal services program, which provides funds for obtaining and implementing broadband in school and libraries across urban and rural disctricts. In addition to these access points, private businesses often provide free wifi to customers. Dispite the many access points, challenges still exist: including hours of operation, transportation, location (and weather exposure), device availability, and purchase requirements.

1.1.1 Why it Matters

As our society has moved online, we have begun the process of reproducing the current systems of inequality into the digital dimension. This is demonstrated by current broadband adoption rates and academic performance seen across income levels and racial groups. Some may suggest that the solution is as simple as giving every student a wifi-enabled laptop, but this far from a solution to a problem that exists within an interconnected ecosystem of hurdles built on current trends of oppression. In most surveys regarding broadband access, respondents often cite the high cost of fixed broadband access as the top deterrent. In addition to the monthly cost of service, there are additional fees and taxes and cost of devices (including computers, routers, etc.). In an article from the Washington Center for Equitable Growth found that 31.4 percent of household with an annual income lower than $50,000 with school-aged children lacked broadband access in their homes. Furthermore, regardless of income levels Black and Latinx children disproportionately lack access when compared to white and Asian children (Crampton, 2018). This creates a homework gap between those who have access at home to support their studies, and those who do not (Yankelevich et al., 2017). Considering that 94 percent of all school districts serving low-income students report assigning internet-based homeowrk (Shapiro, 2015), the need for broadband becomes alarming clear.

1.1.2 Spatial Scope

Census tracts within the City of Los Angeles

1.1.3 Data Sources

2019 American Community Survey from Census Reporter and LA City Open Data

1.1.4 Goals and Intentions

After first identifying communities with an outstanding percentage of residents who completely lack internet access and those with limited access, I will then map the public access points in relation to these communtiies to assess the proximity and quality of resources through distance, available services (available computers, hotspot loan program, etc.), and populations served.

1.2 Data Exploration

The first portion of this exploration will focus on the cleaning and formatting the data. *expand further*

[1]:
```
# IMPORT

#for data processing
import pandas as pd
import numpy as np
import libpysal as lps

#for spatial data
import geopandas as gpd
import esda
from esda.moran import Moran, Moran_Local

#for mapping
import contextily as ctx
import osmnx as ox
import networkx as nx

#for dataviz
import matplotlib.pyplot as plt
import plotly.express as px
import urllib.request, json
import seaborn as sns

import plotly.graph_objects as go
from plotly.subplots import make_subplots

#for data retrieval
from sodapy import socrata
```

```
#For spatial Statics
import splot
from splot.esda import moran_scatterplot, plot_moran,␣
 ↪lisa_cluster,plot_moran_simulation
import libpysal as lps
```

```
/opt/conda/lib/python3.9/site-packages/geopandas/_compat.py:106: UserWarning:
The Shapely GEOS version (3.9.1-CAPI-1.14.2) is incompatible with the GEOS
version PyGEOS was compiled with (3.10.1-CAPI-1.16.0). Conversions between both
will be slow.
  warnings.warn(
```

[2]:
```
# UPLOAD DATA

#age variable
age = gpd.read_file('data/acs2019_1yr_B28005_16000US0644000.geojson')

#educ varaible
educ = gpd.read_file('data/acs2019_1yr_B28006_16000US0644000.geojson')

#race variable
race = gpd.read_file('data/acs2019_1yr_B28009_all races-b.csv')

#household subscription status
hh = gpd.read_file('data/acs2019_5yr_B28002_14000US06037293306.geojson')

#libraries within LA City
lib = gpd.read_file('data/LA_Library_Map.csv')

#data from social explorer
se = gpd.read_file('data/R12960103_SL140.csv')

#data from City of LA
tracts = gpd.read_file('data/2010_Census_Tracts_by_Council_District.geojson')
crowd = gpd.read_file('data/Census_Tracts_with_Severely_Overcrowded_Households.
 ↪geojson')
```

[3]:
```
# api
#lacity_url1 = 'https://services5.arcgis.com/7nsPwEMP38bSkCjy/arcgis/rest/
 ↪services/Census%20Tracts%202010%20overlaid%20on%20CDs/FeatureServer/0/query?
 ↪where=1%3D1&outFields=TRACTCE10,POP,TOOLTIP_1,District&geometry=&geometryType=esriGeometryE
#lacity_url2 = 'https://maps.lacity.org/lahub/rest/services/
 ↪Housing_and_Community_Investment_Department/MapServer/1/query?
 ↪where=1%3D1&outFields=*&outSR=4326&f=json'

# call the api and bring the data in
#with urllib.request.urlopen(lacity_url1) as url1:
```

```
#    data = json.loads(url1.read().decode())

# convert the data to a dataframe
#df = pd.json_normalize(data)
#df
```

[4]:
```
# Understand the data
age.info()
educ.info()
race.info()
hh.info()
lib.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 1 entries, 0 to 0
Data columns (total 41 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   geoid             1 non-null      object
 1   name              1 non-null      object
 2   B28005001         1 non-null      float64
 3   B28005001, Error  1 non-null      float64
 4   B28005002         1 non-null      float64
 5   B28005002, Error  1 non-null      float64
 6   B28005003         1 non-null      float64
 7   B28005003, Error  1 non-null      float64
 8   B28005004         1 non-null      float64
 9   B28005004, Error  1 non-null      float64
 10  B28005005         1 non-null      float64
 11  B28005005, Error  1 non-null      float64
 12  B28005006         1 non-null      float64
 13  B28005006, Error  1 non-null      float64
 14  B28005007         1 non-null      float64
 15  B28005007, Error  1 non-null      float64
 16  B28005008         1 non-null      float64
 17  B28005008, Error  1 non-null      float64
 18  B28005009         1 non-null      float64
 19  B28005009, Error  1 non-null      float64
 20  B28005010         1 non-null      float64
 21  B28005010, Error  1 non-null      float64
 22  B28005011         1 non-null      float64
 23  B28005011, Error  1 non-null      float64
 24  B28005012         1 non-null      float64
 25  B28005012, Error  1 non-null      float64
 26  B28005013         1 non-null      float64
 27  B28005013, Error  1 non-null      float64
 28  B28005014         1 non-null      float64
```

```
 29  B28005014, Error  1 non-null      float64
 30  B28005015         1 non-null      float64
 31  B28005015, Error  1 non-null      float64
 32  B28005016         1 non-null      float64
 33  B28005016, Error  1 non-null      float64
 34  B28005017         1 non-null      float64
 35  B28005017, Error  1 non-null      float64
 36  B28005018         1 non-null      float64
 37  B28005018, Error  1 non-null      float64
 38  B28005019         1 non-null      float64
 39  B28005019, Error  1 non-null      float64
 40  geometry          1 non-null      geometry
dtypes: float64(38), geometry(1), object(2)
memory usage: 456.0+ bytes
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 1 entries, 0 to 0
Data columns (total 41 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   geoid             1 non-null      object
 1   name              1 non-null      object
 2   B28006001         1 non-null      float64
 3   B28006001, Error  1 non-null      float64
 4   B28006002         1 non-null      float64
 5   B28006002, Error  1 non-null      float64
 6   B28006003         1 non-null      float64
 7   B28006003, Error  1 non-null      float64
 8   B28006004         1 non-null      float64
 9   B28006004, Error  1 non-null      float64
 10  B28006005         1 non-null      float64
 11  B28006005, Error  1 non-null      float64
 12  B28006006         1 non-null      float64
 13  B28006006, Error  1 non-null      float64
 14  B28006007         1 non-null      float64
 15  B28006007, Error  1 non-null      float64
 16  B28006008         1 non-null      float64
 17  B28006008, Error  1 non-null      float64
 18  B28006009         1 non-null      float64
 19  B28006009, Error  1 non-null      float64
 20  B28006010         1 non-null      float64
 21  B28006010, Error  1 non-null      float64
 22  B28006011         1 non-null      float64
 23  B28006011, Error  1 non-null      float64
 24  B28006012         1 non-null      float64
 25  B28006012, Error  1 non-null      float64
 26  B28006013         1 non-null      float64
 27  B28006013, Error  1 non-null      float64
 28  B28006014         1 non-null      float64
```

```
 29  B28006014, Error  1 non-null      float64
 30  B28006015         1 non-null      float64
 31  B28006015, Error  1 non-null      float64
 32  B28006016         1 non-null      float64
 33  B28006016, Error  1 non-null      float64
 34  B28006017         1 non-null      float64
 35  B28006017, Error  1 non-null      float64
 36  B28006018         1 non-null      float64
 37  B28006018, Error  1 non-null      float64
 38  B28006019         1 non-null      float64
 39  B28006019, Error  1 non-null      float64
 40  geometry          1 non-null      geometry
dtypes: float64(38), geometry(1), object(2)
memory usage: 456.0+ bytes
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   race      20 non-null     object
 1   status    20 non-null     object
 2   count     20 non-null     object
 3   geometry  0 non-null      geometry
dtypes: geometry(1), object(3)
memory usage: 768.0+ bytes
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 1005 entries, 0 to 1004
Data columns (total 29 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   geoid             1005 non-null   object
 1   name              1005 non-null   object
 2   B28002001         1005 non-null   float64
 3   B28002001, Error  1005 non-null   float64
 4   B28002002         1005 non-null   float64
 5   B28002002, Error  1005 non-null   float64
 6   B28002003         1005 non-null   float64
 7   B28002003, Error  1005 non-null   float64
 8   B28002004         1005 non-null   float64
 9   B28002004, Error  1005 non-null   float64
 10  B28002005         1005 non-null   float64
 11  B28002005, Error  1005 non-null   float64
 12  B28002006         1005 non-null   float64
 13  B28002006, Error  1005 non-null   float64
 14  B28002007         1005 non-null   float64
 15  B28002007, Error  1005 non-null   float64
 16  B28002008         1005 non-null   float64
 17  B28002008, Error  1005 non-null   float64
```

```
 18  B28002009          1005 non-null   float64
 19  B28002009, Error   1005 non-null   float64
 20  B28002010          1005 non-null   float64
 21  B28002010, Error   1005 non-null   float64
 22  B28002011          1005 non-null   float64
 23  B28002011, Error   1005 non-null   float64
 24  B28002012          1005 non-null   float64
 25  B28002012, Error   1005 non-null   float64
 26  B28002013          1005 non-null   float64
 27  B28002013, Error   1005 non-null   float64
 28  geometry           1005 non-null   geometry
dtypes: float64(26), geometry(1), object(2)
memory usage: 227.8+ KB
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 73 entries, 0 to 72
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Branch Name       73 non-null     object
 1   Phone Number      73 non-null     object
 2   Email             73 non-null     object
 3   Council District  73 non-null     object
 4   Location          73 non-null     object
 5   geometry          0 non-null      geometry
dtypes: geometry(1), object(5)
memory usage: 3.5+ KB
```

[5]:
```
#simplify the available data and rename columns for processing. I am going to␣
 ↪skip this step for race because I compiled the individual tables from census␣
 ↪reporter in excel to save some time.

#age first
age_col2keep = ['geoid',
     'name',
     'B28005001',
     'B28005002',
     'B28005003',
     'B28005004',
     'B28005005',
     'B28005006',
     'B28005007',
     'B28005008',
     'B28005009',
     'B28005010',
     'B28005011',
     'B28005012',
     'B28005013',
```

```
     'B28005014',
     'B28005015',
     'B28005016',
     'B28005017',
     'B28005018',
     'B28005019',
     'geometry']
list(age_col2keep)
```

[5]:
```
['geoid',
 'name',
 'B28005001',
 'B28005002',
 'B28005003',
 'B28005004',
 'B28005005',
 'B28005006',
 'B28005007',
 'B28005008',
 'B28005009',
 'B28005010',
 'B28005011',
 'B28005012',
 'B28005013',
 'B28005014',
 'B28005015',
 'B28005016',
 'B28005017',
 'B28005018',
 'B28005019',
 'geometry']
```

[6]:
```
#next, education
educ_col2keep = ['geoid',
     'name',
     'B28006001',
     'B28006002',
     'B28006003',
     'B28006004',
     'B28006005',
     'B28006006',
     'B28006007',
     'B28006008',
     'B28006009',
     'B28006010',
     'B28006011',
     'B28006012',
```

```
    'B28006013',
    'B28006014',
    'B28006015',
    'B28006016',
    'B28006017',
    'B28006018',
    'B28006019',
    'geometry']
```

[7]:
```
#same step for household data
list(hh)
```

[7]:
```
['geoid',
 'name',
 'B28002001',
 'B28002001, Error',
 'B28002002',
 'B28002002, Error',
 'B28002003',
 'B28002003, Error',
 'B28002004',
 'B28002004, Error',
 'B28002005',
 'B28002005, Error',
 'B28002006',
 'B28002006, Error',
 'B28002007',
 'B28002007, Error',
 'B28002008',
 'B28002008, Error',
 'B28002009',
 'B28002009, Error',
 'B28002010',
 'B28002010, Error',
 'B28002011',
 'B28002011, Error',
 'B28002012',
 'B28002012, Error',
 'B28002013',
 'B28002013, Error',
 'geometry']
```

[8]:
```
hh_col2keep = ['geoid',
 'name',
 'B28002001',
 'B28002002',
 'B28002003',
```

```
 'B28002004',
 'B28002005',
 'B28002006',
 'B28002007',
 'B28002008',
 'B28002009',
 'B28002010',
 'B28002011',
 'B28002012',
 'B28002013',
 'geometry']
```

[9]:
```
age = age[age_col2keep]
list(age)
```

[9]:
```
['geoid',
 'name',
 'B28005001',
 'B28005002',
 'B28005003',
 'B28005004',
 'B28005005',
 'B28005006',
 'B28005007',
 'B28005008',
 'B28005009',
 'B28005010',
 'B28005011',
 'B28005012',
 'B28005013',
 'B28005014',
 'B28005015',
 'B28005016',
 'B28005017',
 'B28005018',
 'B28005019',
 'geometry']
```

[10]:
```
educ = educ[educ_col2keep]
list(age)
```

[10]:
```
['geoid',
 'name',
 'B28005001',
 'B28005002',
 'B28005003',
 'B28005004',
```

```
 'B28005005',
 'B28005006',
 'B28005007',
 'B28005008',
 'B28005009',
 'B28005010',
 'B28005011',
 'B28005012',
 'B28005013',
 'B28005014',
 'B28005015',
 'B28005016',
 'B28005017',
 'B28005018',
 'B28005019',
 'geometry']
```

[11]:
```
hh = hh[hh_col2keep]
list(hh)
```

[11]:
```
['geoid',
 'name',
 'B28002001',
 'B28002002',
 'B28002003',
 'B28002004',
 'B28002005',
 'B28002006',
 'B28002007',
 'B28002008',
 'B28002009',
 'B28002010',
 'B28002011',
 'B28002012',
 'B28002013',
 'geometry']
```

[12]:
```
age.columns = ['geoid',
 'name',
 'Total',
 'Under 18 years',
 '<18, Has a computer',
 '<18, With dial-up Internet subscription alone',
 '<18, With a broadband Internet subscription',
 '<18, Without an Internet subscription',
 '<18, No computer',
 '18 to 64 years',
```

```
 '18-64, Has a computer',
 '18-64, With dial-up Internet subscription alone',
 '18-64, With a broadband Internet subscription',
 '18-64, Without an Internet subscription',
 '18-64, No computer',
 '65 years and over',
 '65, Has a computer',
 '65, With dial-up Internet subscription alone',
 '65, With a broadband Internet subscription',
 '65, Without an Internet subscription',
 '65, No computer',
 'geometry']
age.head()
```

[12]:

```
            geoid             name      Total  Under 18 years  \
0  16000US0644000  Los Angeles, CA  3885625.0        800901.0

   <18, Has a computer  <18, With dial-up Internet subscription alone  \
0             788025.0                                          222.0

   <18, With a broadband Internet subscription  \
0                                     745718.0

   <18, Without an Internet subscription  <18, No computer  18 to 64 years  \
0                                42085.0           12876.0       2578959.0

   …  18-64, With a broadband Internet subscription  \
0  …                                      2378932.0

   18-64, Without an Internet subscription  18-64, No computer  \
0                                 145935.0             52365.0

   65 years and over  65, Has a computer  \
0          505765.0            434466.0

   65, With dial-up Internet subscription alone  \
0                                        1549.0

   65, With a broadband Internet subscription  \
0                                    388032.0

   65, Without an Internet subscription  65, No computer  \
0                                44885.0          71299.0

                                            geometry
0  MULTIPOLYGON (((-118.66818 34.18987, -118.6681…
```

[1 rows x 22 columns]

[13]:
```
educ.columns=['geoid',
 'name',
 'Total',
 'Less than high school graduate or equivalency',
 '<HS, Has a computer',
 '<HS, With dial-up Internet subscription alone',
 '<HS, With a broadband Internet subscription',
 '<HS, Without an Internet subscription',
 '<HS, No computer',
 "High school graduate (includes equivalency), some college or associate's␣
 ↪degree",
 '>HS, Has a computer',
 '>HS, With dial-up Internet subscription alone',
 '>HS, With a broadband Internet subscription',
 '>HS, Without an Internet subscription',
 '>HS, No computer',
 "Bachelor's degree or higher",
 '>Bach, Has a computer',
 '>Bach, With dial-up Internet subscription alone',
 '>Bach, With a broadband Internet subscription',
 '>Bach, Without an Internet subscription',
 '>Bach, No computer',
 'geometry']
educ.head()
```

[13]:
```
            geoid             name      Total  \
0  16000US0644000  Los Angeles, CA  2728694.0

   Less than high school graduate or equivalency  <HS, Has a computer  \
0                                       584784.0             533424.0

   <HS, With dial-up Internet subscription alone  \
0                                          531.0

   <HS, With a broadband Internet subscription  \
0                                     467300.0

   <HS, Without an Internet subscription  <HS, No computer  \
0                                65593.0           51360.0

   High school graduate (includes equivalency), some college or associate's
degree  \
0                                       1154728.0

   …  >HS, With a broadband Internet subscription  \
```

```
0  …                                    1024262.0

   >HS, Without an Internet subscription  >HS, No computer  \
0                                 75978.0           52878.0

   Bachelor's degree or higher  >Bach, Has a computer  \
0                     989182.0               973551.0

   >Bach, With dial-up Internet subscription alone  \
0                                           905.0

   >Bach, With a broadband Internet subscription  \
0                                       943721.0

   >Bach, Without an Internet subscription  >Bach, No computer  \
0                                   28925.0             15631.0

                                                  geometry
0  MULTIPOLYGON (((-118.66818 34.18987, -118.6681…

[1 rows x 22 columns]
```

[14]:
```
hh.columns = ['geoid',
 'name',
 'Total',
 'With an Internet Subscription',
 'Dial-up with no other type of Internet',
 'Broadband of any type',
 'Cellular data plan',
 'Cellular data plan with no other type of Internet subscription',
 'Broadband such as cable, fiber optic or DSL',
 'Broadband such as cable, fiber optic or DSL with no other type of Internet␣
 ↪subscription',
 'Satellite Internet service',
 'Satellite Internet service with no other type of Internet subscription',
 'Other service with no other type of Internet subscription',
 'Internet access without a subscription',
 'No Internet access',
 'geometry']
hh.head()
```

[14]:
```
                geoid                                     name   Total  \
0  14000US06037101110  Census Tract 1011.10, Los Angeles, CA  1575.0
1  14000US06037101122  Census Tract 1011.22, Los Angeles, CA  1312.0
2  14000US06037101210  Census Tract 1012.10, Los Angeles, CA  2297.0
3  14000US06037101220  Census Tract 1012.20, Los Angeles, CA  1357.0
4  14000US06037101300     Census Tract 1013, Los Angeles, CA  1445.0
```

```
   With an Internet Subscription  Dial-up with no other type of Internet  \
0                         1223.0                                     0.0
1                         1095.0                                    12.0
2                         1687.0                                     0.0
3                         1017.0                                     0.0
4                         1185.0                                    18.0

   Broadband of any type  Cellular data plan  \
0                 1223.0              973.0
1                 1083.0              959.0
2                 1687.0             1453.0
3                 1017.0              868.0
4                 1167.0              955.0

   Cellular data plan with no other type of Internet subscription  \
0                                                97.0
1                                               113.0
2                                               113.0
3                                               142.0
4                                                70.0

   Broadband such as cable, fiber optic or DSL  \
0                                       1045.0
1                                        960.0
2                                       1388.0
3                                        848.0
4                                       1080.0

   Broadband such as cable, fiber optic or DSL with no other type of Internet
subscription  \
0                                            218.0
1                                            124.0
2                                            182.0
3                                            131.0
4                                            203.0

   Satellite Internet service  \
0                       108.0
1                        97.0
2                       196.0
3                        44.0
4                       100.0

   Satellite Internet service with no other type of Internet subscription  \
0                                                 21.0
1                                                  0.0
```

```
2                                                 33.0
3                                                  0.0
4                                                  0.0

   Other service with no other type of Internet subscription  \
0                                                  0.0
1                                                  0.0
2                                                  0.0
3                                                  0.0
4                                                  0.0

   Internet access without a subscription  No Internet access  \
0                                    95.0                257.0
1                                    90.0                127.0
2                                    72.0                538.0
3                                    59.0                281.0
4                                    99.0                161.0

                                            geometry
0  MULTIPOLYGON (((-118.30229 34.25870, -118.3009…
1  MULTIPOLYGON (((-118.30334 34.27371, -118.3033…
2  MULTIPOLYGON (((-118.29945 34.25598, -118.2979…
3  MULTIPOLYGON (((-118.28593 34.25227, -118.2859…
4  MULTIPOLYGON (((-118.27822 34.25068, -118.2782…
```

[15]:
```
# normalize the data
age['Percent with a computer under 18 years']=age['<18, Has a computer']/
 ↪age['Under 18 years']*100
age['Percent with a computer and dial-up internet subscription under 18␣
 ↪years']=age['<18, With dial-up Internet subscription alone']/age['Under 18␣
 ↪years']*100
age['Percent with a computer and broadband internet subscription under 18␣
 ↪years']=age['<18, With a broadband Internet subscription']/age['Under 18␣
 ↪years']*100
age['Percent with a computer and no internet subscription under 18␣
 ↪years']=age['<18, Without an Internet subscription']/age['Under 18␣
 ↪years']*100
age['Percent with no computer under 18 years']=age['<18, No computer']/
 ↪age['Under 18 years']*100
age['Percent with a computer between 18 and 64 years']=age['<18, Has a␣
 ↪computer']/age['18 to 64 years']*100
age['Percent with a computer and dial-up internet subscription between 18 and␣
 ↪64 years']=age['18-64, With dial-up Internet subscription alone']/age['18 to␣
 ↪64 years']*100
```

```
age['Percent with a computer and broadband internet subscription between 18 and␣
 ↪64 years']=age['18-64, With a broadband Internet subscription']/age['18 to␣
 ↪64 years']*100
age['Percent with a computer and no internet subscription between 18 and 64␣
 ↪years']=age['18-64, Without an Internet subscription']/age['18 to 64␣
 ↪years']*100
age['Percent with a no computer between 18 and 64']=age['18-64, No computer']/
 ↪age['18 to 64 years']*100
age['Percent with a computer over 65 years']=age['65, Has a computer']/age['65␣
 ↪years and over']*100
age['Percent with a computer and dial-up internet subscription over 65␣
 ↪years']=age['65, With dial-up Internet subscription alone']/age['65 years␣
 ↪and over']*100
age['Percent with a computer and broadband internet subscription over 65␣
 ↪years']=age['65, With a broadband Internet subscription']/age['65 years and␣
 ↪over']*100
age['Percent with a computer and no internet subscription over 65␣
 ↪years']=age['65, Without an Internet subscription']/age['65 years and␣
 ↪over']*100
age['Percent with a no computer over 65']=age['65, No computer']/age['65 years␣
 ↪and over']*100
age.head()
```

[15]:

```
            geoid             name      Total  Under 18 years  \
0  16000US0644000  Los Angeles, CA  3885625.0        800901.0

   <18, Has a computer  <18, With dial-up Internet subscription alone  \
0             788025.0                                          222.0

   <18, With a broadband Internet subscription  \
0                                     745718.0

   <18, Without an Internet subscription  <18, No computer  18 to 64 years  \
0                                42085.0           12876.0       2578959.0

   …  Percent with a computer between 18 and 64 years  \
0  …                                       30.555934

   Percent with a computer and dial-up internet subscription between 18 and 64
years  \
0                                       0.066965

   Percent with a computer and broadband internet subscription between 18 and 64
years  \
0                                      92.243886
```

```
   Percent with a computer and no internet subscription between 18 and 64 years
\
0                                                5.658679

   Percent with a no computer between 18 and 64  \
0                                       2.03047

   Percent with a computer over 65 years  \
0                                85.902741

   Percent with a computer and dial-up internet subscription over 65 years  \
0                                                0.306269

   Percent with a computer and broadband internet subscription over 65 years  \
0                                               76.721798

   Percent with a computer and no internet subscription over 65 years  \
0                                                8.874675

   Percent with a no computer over 65
0                              14.097259

[1 rows x 37 columns]
```

[16]:
```
educ['Percent with less than HS and has a computer']=educ['<HS, Has a␣
 ↪computer']/educ['Less than high school graduate or equivalency']*100
educ['Percent with less than HS, has a computer with dial-up']=educ['<HS, With␣
 ↪dial-up Internet subscription alone']/educ['Less than high school graduate␣
 ↪or equivalency']*100
educ['Percent with less than HS, has a computer with broadband']=educ['<HS,␣
 ↪With a broadband Internet subscription']/educ['Less than high school␣
 ↪graduate or equivalency']*100
educ['Percent with less than HS, has a computer with no internet']=educ['<HS,␣
 ↪Without an Internet subscription']/educ['Less than high school graduate or␣
 ↪equivalency']*100
educ['Percent with less than HS, has no computer']=educ['<HS, No computer']/
 ↪educ['Less than high school graduate or equivalency']*100
educ['Percent with more than HS and has a computer']=educ['>HS, Has a␣
 ↪computer']/educ["High school graduate (includes equivalency), some college␣
 ↪or associate's degree"]*100
educ['Percent with more than HS, has a computer with dial-up']=educ['>HS, With␣
 ↪dial-up Internet subscription alone']/educ["High school graduate (includes␣
 ↪equivalency), some college or associate's degree"]*100
educ['Percent with more than HS, has a computer with broadband']=educ['>HS,␣
 ↪With a broadband Internet subscription']/educ["High school graduate␣
 ↪(includes equivalency), some college or associate's degree"]*100
```

```
educ['Percent with more than HS, has a computer with no internet']=educ['>HS,␣
 ↪Without an Internet subscription']/educ["High school graduate (includes␣
 ↪equivalency), some college or associate's degree"]*100
educ['Percent with more than HS, has no computer']=educ['>HS, No computer']/
 ↪educ["High school graduate (includes equivalency), some college or␣
 ↪associate's degree"]*100
educ['Percent with more than Bach and has a computer']=educ['>Bach, Has a␣
 ↪computer']/educ["Bachelor's degree or higher"]*100
educ['Percent with more than Bach, has a computer with dial-up']=educ['>Bach,␣
 ↪With dial-up Internet subscription alone']/educ["Bachelor's degree or␣
 ↪higher"]*100
educ['Percent with more than Bach, has a computer with broadband']=educ['>Bach,␣
 ↪With a broadband Internet subscription']/educ["Bachelor's degree or␣
 ↪higher"]*100
educ['Percent with more than Bach, has a computer with no␣
 ↪internet']=educ['>Bach, Without an Internet subscription']/educ["Bachelor's␣
 ↪degree or higher"]*100
educ['Percent with more than Bach, has no computer']=educ['>Bach, No computer']/
 ↪educ["Bachelor's degree or higher"]*100
educ.head()
```

[16]:

```
            geoid             name      Total  \
0  16000US0644000  Los Angeles, CA  2728694.0

   Less than high school graduate or equivalency  <HS, Has a computer  \
0                                       584784.0             533424.0

   <HS, With dial-up Internet subscription alone  \
0                                          531.0

   <HS, With a broadband Internet subscription  \
0                                     467300.0

   <HS, Without an Internet subscription  <HS, No computer  \
0                                65593.0           51360.0

   High school graduate (includes equivalency), some college or associate's
degree  \
0                                          1154728.0

   …  Percent with more than HS and has a computer  \
0  …                                      95.42074

   Percent with more than HS, has a computer with dial-up  \
0                                           0.139427
```

```
   Percent with more than HS, has a computer with broadband  \
0                                          88.701582

   Percent with more than HS, has a computer with no internet  \
0                                           6.579731

   Percent with more than HS, has no computer  \
0                                     4.57926

   Percent with more than Bach and has a computer  \
0                                      98.419805

   Percent with more than Bach, has a computer with dial-up  \
0                                           0.09149

   Percent with more than Bach, has a computer with broadband  \
0                                          95.404182

   Percent with more than Bach, has a computer with no internet  \
0                                           2.924133

   Percent with more than Bach, has no computer
0                                      1.580195

[1 rows x 37 columns]
```

[17]:
```
#to simplify the normalization process for household subscription status I will␣
 ↪only convert those that I need for mapping
hh['Percent with an internet subscription']=hh['With an Internet Subscription']/
 ↪hh['Total']*100
hh['Percent with a cellular data plan with no other type of␣
 ↪Internet']=hh['Cellular data plan with no other type of Internet␣
 ↪subscription']/hh['Total']*100
hh['Percent with no internet access']=hh['No Internet access']/hh['Total']*100
hh.head()
```

[17]:
```
                geoid                                    name   Total  \
0  14000US06037101110  Census Tract 1011.10, Los Angeles, CA  1575.0
1  14000US06037101122  Census Tract 1011.22, Los Angeles, CA  1312.0
2  14000US06037101210  Census Tract 1012.10, Los Angeles, CA  2297.0
3  14000US06037101220  Census Tract 1012.20, Los Angeles, CA  1357.0
4  14000US06037101300     Census Tract 1013, Los Angeles, CA  1445.0

   With an Internet Subscription  Dial-up with no other type of Internet  \
0                         1223.0                                     0.0
1                         1095.0                                    12.0
2                         1687.0                                     0.0
```

```
3                        1017.0                                      0.0
4                        1185.0                                     18.0

   Broadband of any type  Cellular data plan  \
0                 1223.0              973.0
1                 1083.0              959.0
2                 1687.0             1453.0
3                 1017.0              868.0
4                 1167.0              955.0

   Cellular data plan with no other type of Internet subscription  \
0                                                97.0
1                                               113.0
2                                               113.0
3                                               142.0
4                                                70.0

   Broadband such as cable, fiber optic or DSL  \
0                                       1045.0
1                                        960.0
2                                       1388.0
3                                        848.0
4                                       1080.0

   Broadband such as cable, fiber optic or DSL with no other type of Internet
subscription  \
0                                               218.0
1                                               124.0
2                                               182.0
3                                               131.0
4                                               203.0

   Satellite Internet service  \
0                       108.0
1                        97.0
2                       196.0
3                        44.0
4                       100.0

   Satellite Internet service with no other type of Internet subscription  \
0                                               21.0
1                                                0.0
2                                               33.0
3                                                0.0
4                                                0.0

   Other service with no other type of Internet subscription  \
```

```
0                                                 0.0
1                                                 0.0
2                                                 0.0
3                                                 0.0
4                                                 0.0

   Internet access without a subscription  No Internet access  \
0                                    95.0              257.0
1                                    90.0              127.0
2                                    72.0              538.0
3                                    59.0              281.0
4                                    99.0              161.0

                                             geometry  \
0  MULTIPOLYGON (((-118.30229 34.25870, -118.3009…
1  MULTIPOLYGON (((-118.30334 34.27371, -118.3033…
2  MULTIPOLYGON (((-118.29945 34.25598, -118.2979…
3  MULTIPOLYGON (((-118.28593 34.25227, -118.2859…
4  MULTIPOLYGON (((-118.27822 34.25068, -118.2782…

   Percent with an internet subscription  \
0                              77.650794
1                              83.460366
2                              73.443622
3                              74.944731
4                              82.006920

   Percent with a cellular data plan with no other type of Internet  \
0                                          6.158730
1                                          8.612805
2                                          4.919460
3                                         10.464259
4                                          4.844291

   Percent with no internet access
0                        16.317460
1                         9.679878
2                        23.421855
3                        20.707443
4                        11.141869
```

[18]:
```
#now that that's over I can trim the data even further to prepare to visualize
age_simp=['Percent with a computer and dial-up internet subscription under 18␣
 ↪years',
      'Percent with a computer and broadband internet subscription under 18␣
 ↪years',
      'Percent with a computer and no internet subscription under 18 years',
```

```
       'Percent with no computer under 18 years',
       'Percent with a computer and dial-up internet subscription between 18␣
 ↪and 64 years',
       'Percent with a computer and broadband internet subscription between 18␣
 ↪and 64 years',
       'Percent with a computer and no internet subscription between 18 and 64␣
 ↪years',
       'Percent with a no computer between 18 and 64',
       'Percent with a computer and dial-up internet subscription over 65␣
 ↪years',
       'Percent with a computer and broadband internet subscription over 65␣
 ↪years',
       'Percent with a computer and no internet subscription over 65 years',
       'Percent with a no computer over 65']
age = age[age_simp]

educ_simp=['Percent with less than HS, has a computer with dial-up',
       'Percent with less than HS, has a computer with broadband',
       'Percent with less than HS, has a computer with no internet',
       'Percent with less than HS, has no computer',
       'Percent with more than HS, has a computer with dial-up',
       'Percent with more than HS, has a computer with broadband',
       'Percent with more than HS, has a computer with no internet',
       'Percent with more than HS, has no computer',
       'Percent with more than Bach, has a computer with dial-up',
       'Percent with more than Bach, has a computer with broadband',
       'Percent with more than Bach, has a computer with no internet',
       'Percent with more than Bach, has no computer']
educ = educ[educ_simp]

race_simp = ['race',
             'status',
             'count',]
race = race[race_simp]

#I'm also going to shorten the labels because in the next step I am going added␣
 ↪the age ranges as it's own variable
age.columns = ['Percent with a computer and dial-up',
       'Percent with a computer and broadband',
       'Percent with a computer and no internet',
       'Percent with no computer',
       'Percent with a computer and dial-up',
       'Percent with a computer and broadband',
       'Percent with a computer and no internet',
       'Percent with no computer',
       'Percent with a computer and dial-up',
```

```
       'Percent with a computer and broadband',
       'Percent with a computer and no internet',
       'Percent with no computer']

educ.columns=['Percent with a computer and dial-up',
       'Percent with a computer and broadband',
       'Percent with a computer and no internet',
       'Percent with no computer',
       'Percent with a computer and dial-up',
       'Percent with a computer and broadband',
       'Percent with a computer and no internet',
       'Percent with no computer',
       'Percent with a computer and dial-up',
       'Percent with a computer and broadband',
       'Percent with a computer and no internet',
       'Percent with no computer']
```

[19]:
```
age=age.T.reset_index().rename(columns={'index':'status',0:'percent'})
educ=educ.T.reset_index().rename(columns={'index':'status',0:'percent'})
```

[20]:
```
age
educ
```

[20]:

	status	percent
0	Percent with a computer and dial-up	0.090803
1	Percent with a computer and broadband	79.909847
2	Percent with a computer and no internet	11.216620
3	Percent with no computer	8.782730
4	Percent with a computer and dial-up	0.139427
5	Percent with a computer and broadband	88.701582
6	Percent with a computer and no internet	6.579731
7	Percent with no computer	4.579260
8	Percent with a computer and dial-up	0.091490
9	Percent with a computer and broadband	95.404182
10	Percent with a computer and no internet	2.924133
11	Percent with no computer	1.580195

[21]:
```
group_col = ['Under 18',
       'Under 18',
       'Under 18',
       'Between 18 and 64 years',
       'Between 18 and 64 years',
       'Between 18 and 64 years',
       'Between 18 and 64 years',
       'Over 65 years',
       'Over 65 years',
```

```
       'Over 65 years',
       'Over 65 years',]
age.loc[:,'Age Group'] = group_col
age
```

[21]:

	status	percent	Age Group
0	Percent with a computer and dial-up	0.027719	Under 18
1	Percent with a computer and broadband	93.109885	Under 18
2	Percent with a computer and no internet	5.254707	Under 18
3	Percent with no computer	1.607689	Under 18
4	Percent with a computer and dial-up	0.066965	Between 18 and 64 years
5	Percent with a computer and broadband	92.243886	Between 18 and 64 years
6	Percent with a computer and no internet	5.658679	Between 18 and 64 years
7	Percent with no computer	2.030470	Between 18 and 64 years
8	Percent with a computer and dial-up	0.306269	Over 65 years
9	Percent with a computer and broadband	76.721798	Over 65 years
10	Percent with a computer and no internet	8.874675	Over 65 years
11	Percent with no computer	14.097259	Over 65 years

[22]:
```
educ_lvl = ['Less than high school graduate or equivalency',
            'Less than high school graduate or equivalency',
            'Less than high school graduate or equivalency',
            'Less than high school graduate or equivalency',
            "High school graduate, but less than Bachelor's",
            "High school graduate, but less than Bachelor's",
            "High school graduate, but less than Bachelor's",
            "High school graduate, but less than Bachelor's",
            "Bachelor's degree or higher",
            "Bachelor's degree or higher",
            "Bachelor's degree or higher",
            "Bachelor's degree or higher",]
educ.loc[:,'Educational Attainment'] = educ_lvl
educ
```

[22]:
```
                                    status    percent  \
0       Percent with a computer and dial-up   0.090803
1     Percent with a computer and broadband  79.909847
2   Percent with a computer and no internet  11.216620
3                  Percent with no computer   8.782730
4       Percent with a computer and dial-up   0.139427
5     Percent with a computer and broadband  88.701582
6   Percent with a computer and no internet   6.579731
7                  Percent with no computer   4.579260
8       Percent with a computer and dial-up   0.091490
9     Percent with a computer and broadband  95.404182
10  Percent with a computer and no internet   2.924133
11                 Percent with no computer   1.580195

                            Educational Attainment
0    Less than high school graduate or equivalency
1    Less than high school graduate or equivalency
2    Less than high school graduate or equivalency
3    Less than high school graduate or equivalency
4   High school graduate, but less than Bachelor's
5   High school graduate, but less than Bachelor's
6   High school graduate, but less than Bachelor's
7   High school graduate, but less than Bachelor's
8                      Bachelor's degree or higher
9                      Bachelor's degree or higher
10                     Bachelor's degree or higher
11                     Bachelor's degree or higher
```

[23]:
```
#Now I'll trim the household data for mapping
hh_trim = hh.copy()
hh_trim = ['geoid',
           'name',
           'Total',
           'Percent with an internet subscription',
           'Percent with a cellular data plan with no other type of Internet',
           'Percent with no internet access',
           'geometry']
hh = hh[hh_trim]
hh.tail()
```

[23]:
```
                  geoid                                    name      Total  \
1000  14000US06037980026  Census Tract 9800.26, Los Angeles, CA        4.0
1001  14000US06037980028  Census Tract 9800.28, Los Angeles, CA        0.0
1002  14000US06037980031  Census Tract 9800.31, Los Angeles, CA       24.0
1003  14000US06037990200     Census Tract 9902, Los Angeles, CA        0.0
1004     16000US0644000                         Los Angeles, CA  1383869.0
```

```
      Percent with an internet subscription  \
1000                             100.000000
1001                                    NaN
1002                             100.000000
1003                                    NaN
1004                              83.290326

      Percent with a cellular data plan with no other type of Internet  \
1000                                              0.000000
1001                                                   NaN
1002                                              0.000000
1003                                                   NaN
1004                                              9.884823

      Percent with no internet access  \
1000                         0.000000
1001                              NaN
1002                         0.000000
1003                              NaN
1004                        13.397077

                                                 geometry
1000  MULTIPOLYGON (((-118.35173 34.28034, -118.3517…
1001  MULTIPOLYGON (((-118.45246 33.94315, -118.4464…
1002  MULTIPOLYGON (((-118.29105 33.75378, -118.2905…
1003  MULTIPOLYGON (((-118.63598 34.03255, -118.6325…
1004  MULTIPOLYGON (((-118.66818 34.18987, -118.6681…
```

[24]:
```
#also, I need to remove the totals for the City...which is the last row in the␣
 ↪data set
hh=hh.drop([1004])
hh.tail()
```

[24]:
```
                  geoid                                    name  Total  \
999   14000US06037980024  Census Tract 9800.24, Los Angeles, CA   75.0
1000  14000US06037980026  Census Tract 9800.26, Los Angeles, CA    4.0
1001  14000US06037980028  Census Tract 9800.28, Los Angeles, CA    0.0
1002  14000US06037980031  Census Tract 9800.31, Los Angeles, CA   24.0
1003  14000US06037990200     Census Tract 9902, Los Angeles, CA    0.0

      Percent with an internet subscription  \
999                              94.666667
1000                            100.000000
1001                                   NaN
1002                            100.000000
1003                                   NaN
```

```
      Percent with a cellular data plan with no other type of Internet  \
999                                                             0.0
1000                                                            0.0
1001                                                            NaN
1002                                                            0.0
1003                                                            NaN

      Percent with no internet access  \
999                          5.333333
1000                         0.000000
1001                              NaN
1002                         0.000000
1003                              NaN

                                                 geometry
999   MULTIPOLYGON (((-118.51849 34.18389, -118.5184…
1000  MULTIPOLYGON (((-118.35173 34.28034, -118.3517…
1001  MULTIPOLYGON (((-118.45246 33.94315, -118.4464…
1002  MULTIPOLYGON (((-118.29105 33.75378, -118.2905…
1003  MULTIPOLYGON (((-118.63598 34.03255, -118.6325…
```

[25]:
```
#Remove non-visible rows
hh.dropna(inplace=True)
```

[26]:
```
# Clean up the library data by seperating the geogrpahic information out of the␣
 ↪location column
lib.head()
```

[26]:
```
                              Branch Name    Phone Number            Email  \
0                          Vermont Square  (323) 290-7405  vmtsqr@lapl.org
1  Vernon - Leon H. Washington Jr. Memorial   (323) 234-9106  vrnonl@lapl.org
2                        Washington Irving  (323) 734-6303  wirvng@lapl.org
3               West Los Angeles Regional   (310) 575-8323  westla@lapl.org
4                     West Valley Regional  (818) 345-9806  wvally@lapl.org

  Council District                                        Location geometry
0                9  1201 W. 48th. Street\nLos Angeles, CA 90037\n(…     None
1                9  4504 S. Central Avenue\nLos Angeles, CA 90011\…     None
2               10  4117 W. Washington Boulevard\nLos Angeles, CA …     None
3               11  11360 Santa Monica Boulevard\nLos Angeles, CA …     None
4                3  19036 Vanowen Street\nReseda, CA 91335\n(34.19…     None
```

[27]:
```
#check to see the datatypes
lib.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 73 entries, 0 to 72
```

```
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Branch Name       73 non-null     object
 1   Phone Number      73 non-null     object
 2   Email             73 non-null     object
 3   Council District  73 non-null     object
 4   Location          73 non-null     object
 5   geometry          0 non-null      geometry
dtypes: geometry(1), object(5)
memory usage: 3.5+ KB
```

[28]:
```
#begin the splitting process
lib[['Address','CityZip','LatLon']]=lib.Location.str.split("\n",expand=True)
lib.head()
```

[28]:

	Branch Name	Phone Number	Email	Council District	Location	geometry	Address	CityZip	LatLon
0	Vermont Square	(323) 290-7405	vmtsqr@lapl.org	9	1201 W. 48th. Street\nLos Angeles, CA 90037\n(…	None	1201 W. 48th. Street	Los Angeles, CA 90037	(33.999699, -118.295799)
1	Vernon - Leon H. Washington Jr. Memorial	(323) 234-9106	vrnonl@lapl.org	9	4504 S. Central Avenue\nLos Angeles, CA 90011\…	None	4504 S. Central Avenue	Los Angeles, CA 90011	(34.002701, -118.256104)
2	Washington Irving	(323) 734-6303	wirvng@lapl.org	10	4117 W. Washington Boulevard\nLos Angeles, CA …	None	4117 W. Washington Boulevard	Los Angeles, CA 90018	(34.039951, -118.329147)
3	West Los Angeles Regional	(310) 575-8323	westla@lapl.org	11	11360 Santa Monica Boulevard\nLos Angeles, CA …	None	11360 Santa Monica Boulevard	Los Angeles, CA 90025	(34.0457, -118.450302)
4	West Valley Regional	(818) 345-9806	wvally@lapl.org	3	19036 Vanowen Street\nReseda, CA 91335\n(34.19…	None	19036 Vanowen Street	Reseda, CA 91335	(34.193604, -118.546898)

[29]:
```
lib[['lat','lon']]=lib.LatLon.str.split(",",expand=True)
lib.head()
```

[29]:
```
                              Branch Name    Phone Number            Email  \
0                          Vermont Square  (323) 290-7405  vmtsqr@lapl.org
1  Vernon - Leon H. Washington Jr. Memorial  (323) 234-9106  vrnonl@lapl.org
2                       Washington Irving  (323) 734-6303  wirvng@lapl.org
3              West Los Angeles Regional   (310) 575-8323  westla@lapl.org
4                    West Valley Regional  (818) 345-9806  wvally@lapl.org

  Council District                                          Location  \
0                9  1201 W. 48th. Street\nLos Angeles, CA 90037\n(…
1                9  4504 S. Central Avenue\nLos Angeles, CA 90011\…
2               10  4117 W. Washington Boulevard\nLos Angeles, CA …
3               11  11360 Santa Monica Boulevard\nLos Angeles, CA …
4                3  19036 Vanowen Street\nReseda, CA 91335\n(34.19…

  geometry                       Address               CityZip  \
0     None          1201 W. 48th. Street  Los Angeles, CA 90037
1     None        4504 S. Central Avenue  Los Angeles, CA 90011
2     None  4117 W. Washington Boulevard  Los Angeles, CA 90018
3     None  11360 Santa Monica Boulevard  Los Angeles, CA 90025
4     None          19036 Vanowen Street       Reseda, CA 91335

                     LatLon         lat            lon
0  (33.999699, -118.295799)  (33.999699   -118.295799)
1  (34.002701, -118.256104)  (34.002701   -118.256104)
2  (34.039951, -118.329147)  (34.039951   -118.329147)
3    (34.0457, -118.450302)    (34.0457   -118.450302)
4  (34.193604, -118.546898)  (34.193604   -118.546898)
```

[30]:
```
#clean uo before further processing
lib_col2keep = ['Branch Name',
                'Phone Number',
                'Email',
                'Address',
                'CityZip',
                'lat',
                'lon']
lib = lib[lib_col2keep]
lib
```

[30]:
```
                              Branch Name    Phone Number  \
0                          Vermont Square  (323) 290-7405
1   Vernon - Leon H. Washington Jr. Memorial  (323) 234-9106
2                       Washington Irving  (323) 734-6303
3              West Los Angeles Regional   (310) 575-8323
4                    West Valley Regional  (818) 345-9806
..                                    …             …
68                      Sunland - Tujunga  (818) 352-4481
```

```
69                                  Sylmar  (818) 367-6102
70                              Valley Plaza  (818) 765-9251
71                                 Van Nuys   (818) 756-8453
72              Venice - Abbot Kinney Memorial  (310) 821-1769

              Email                         Address                     CityZip  \
0   vmtsqr@lapl.org           1201 W. 48th. Street      Los Angeles, CA 90037
1   vrnonl@lapl.org         4504 S. Central Avenue      Los Angeles, CA 90011
2   wirvng@lapl.org  4117 W. Washington Boulevard      Los Angeles, CA 90018
3   westla@lapl.org  11360 Santa Monica Boulevard      Los Angeles, CA 90025
4   wvally@lapl.org           19036 Vanowen Street           Reseda, CA 91335
..              …                             …                        …
68  snlndt@lapl.org        7771 Foothill Boulevard          Tujunga, CA 91042
69  sylmar@lapl.org              14561 Polk Street           Sylmar, CA 91342
70  valplz@lapl.org           12311 Vanowen Street  North Hollywood, CA 91605
71  vnnuys@lapl.org                6250 Sylmar Ave         Van Nuys, CA 91401
72  venice@lapl.org        501 S. Venice Boulevard           Venice, CA 90291

           lat            lon
0   (33.999699   -118.295799)
1   (34.002701   -118.256104)
2   (34.039951   -118.329147)
3     (34.0457   -118.450302)
4   (34.193604   -118.546898)
..         …             …
68  (34.258099   -118.301498)
69   (34.30735    -118.45015)
70  (34.194302     -118.4011)
71  (34.183899   -118.446297)
72  (33.987598   -118.465103)

[73 rows x 7 columns]
```

[31]:
```
#now I must remove the parenthesis in the lat and lon columns to plot them
lib['lat'] = lib['lat'].str[1:]
lib['lon'] = lib['lon'].str[:-1]
lib
```

[31]:
```
                                 Branch Name    Phone Number  \
0                              Vermont Square  (323) 290-7405
1   Vernon - Leon H. Washington Jr. Memorial   (323) 234-9106
2                          Washington Irving  (323) 734-6303
3                 West Los Angeles Regional   (310) 575-8323
4                       West Valley Regional  (818) 345-9806
..                                       …             …
68                         Sunland - Tujunga  (818) 352-4481
69                                  Sylmar  (818) 367-6102
```

```
70                              Valley Plaza  (818) 765-9251
71                                 Van Nuys   (818) 756-8453
72              Venice - Abbot Kinney Memorial  (310) 821-1769

              Email                        Address                     CityZip  \
0   vmtsqr@lapl.org           1201 W. 48th. Street      Los Angeles, CA 90037
1   vrnonl@lapl.org          4504 S. Central Avenue     Los Angeles, CA 90011
2   wirvng@lapl.org  4117 W. Washington Boulevard      Los Angeles, CA 90018
3   westla@lapl.org  11360 Santa Monica Boulevard      Los Angeles, CA 90025
4   wvally@lapl.org           19036 Vanowen Street           Reseda, CA 91335
..              …                              …                        …
68  snlndt@lapl.org        7771 Foothill Boulevard          Tujunga, CA 91042
69  sylmar@lapl.org              14561 Polk Street           Sylmar, CA 91342
70  valplz@lapl.org           12311 Vanowen Street  North Hollywood, CA 91605
71  vnnuys@lapl.org                6250 Sylmar Ave         Van Nuys, CA 91401
72  venice@lapl.org        501 S. Venice Boulevard           Venice, CA 90291

          lat           lon
0   33.999699   -118.295799
1   34.002701   -118.256104
2   34.039951   -118.329147
3     34.0457   -118.450302
4   34.193604   -118.546898
..        …             …
68  34.258099   -118.301498
69   34.30735    -118.45015
70  34.194302     -118.4011
71  34.183899   -118.446297
72  33.987598   -118.465103

[73 rows x 7 columns]
```

[32]:
```
lib.head()
```

[32]:
```
                              Branch Name    Phone Number            Email  \
0                          Vermont Square  (323) 290-7405  vmtsqr@lapl.org
1  Vernon - Leon H. Washington Jr. Memorial   (323) 234-9106  vrnonl@lapl.org
2                        Washington Irving  (323) 734-6303  wirvng@lapl.org
3                West Los Angeles Regional   (310) 575-8323  westla@lapl.org
4                     West Valley Regional  (818) 345-9806  wvally@lapl.org

                       Address                CityZip        lat  \
0          1201 W. 48th. Street  Los Angeles, CA 90037  33.999699
1         4504 S. Central Avenue  Los Angeles, CA 90011  34.002701
2  4117 W. Washington Boulevard  Los Angeles, CA 90018  34.039951
3  11360 Santa Monica Boulevard  Los Angeles, CA 90025    34.0457
4          19036 Vanowen Street       Reseda, CA 91335  34.193604
```

```
            lon
0   -118.295799
1   -118.256104
2   -118.329147
3   -118.450302
4   -118.546898
```

[33]:
```
lib.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73 entries, 0 to 72
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Branch Name   73 non-null     object
 1   Phone Number  73 non-null     object
 2   Email         73 non-null     object
 3   Address       73 non-null     object
 4   CityZip       73 non-null     object
 5   lat           73 non-null     object
 6   lon           73 non-null     object
dtypes: object(7)
memory usage: 4.1+ KB
```

[34]:
```
#I am going to reorder the list alphbetically by Branch Name
lib.sort_values('Branch Name', ascending = True)
lib
```

[34]:
```
                                Branch Name    Phone Number  \
0                             Vermont Square  (323) 290-7405
1   Vernon - Leon H. Washington Jr. Memorial  (323) 234-9106
2                          Washington Irving  (323) 734-6303
3                 West Los Angeles Regional   (310) 575-8323
4                       West Valley Regional  (818) 345-9806
..                                       …             …
68                        Sunland - Tujunga  (818) 352-4481
69                                  Sylmar  (818) 367-6102
70                             Valley Plaza  (818) 765-9251
71                                Van Nuys   (818) 756-8453
72            Venice - Abbot Kinney Memorial  (310) 821-1769

              Email                       Address                    CityZip  \
0   vmtsqr@lapl.org          1201 W. 48th. Street      Los Angeles, CA 90037
1   vrnonl@lapl.org         4504 S. Central Avenue     Los Angeles, CA 90011
2   wirvng@lapl.org  4117 W. Washington Boulevard     Los Angeles, CA 90018
3   westla@lapl.org  11360 Santa Monica Boulevard     Los Angeles, CA 90025
```

```
4   wvally@lapl.org          19036 Vanowen Street           Reseda, CA 91335
..              …                             …                          …
68  snlndt@lapl.org       7771 Foothill Boulevard          Tujunga, CA 91042
69  sylmar@lapl.org             14561 Polk Street           Sylmar, CA 91342
70  valplz@lapl.org          12311 Vanowen Street  North Hollywood, CA 91605
71  vnnuys@lapl.org               6250 Sylmar Ave         Van Nuys, CA 91401
72  venice@lapl.org       501 S. Venice Boulevard           Venice, CA 90291

          lat           lon
0   33.999699   -118.295799
1   34.002701   -118.256104
2   34.039951   -118.329147
3     34.0457   -118.450302
4   34.193604   -118.546898
..        …           …
68  34.258099   -118.301498
69   34.30735    -118.45015
70  34.194302     -118.4011
71  34.183899   -118.446297
72  33.987598   -118.465103

[73 rows x 7 columns]
```

[35]:
```
#Glad that is done, but it is not over yet...Only 18 of the 73 libraries offer␣
↪wifi hotspot rentals, so I will need to add a column and signify which␣
↪libraries do
hotspot = ['no',
           'no',
           'yes',
           'no',
           'yes',
           'yes',
           'no',
           'no',
           'no',
           'yes',
           'no',
           'no',
           'no',
           'yes',
           'no',
           'no',
           'no',
           'no',
           'no',
           'no',
           'no',
```

```
        'no',
        'no',
        'no',
        'yes',
        'no',
        'yes',
        'yes',
        'no',
        'no',
        'no',
        'yes',
        'no',
        'no',
        'no',
        'yes',
        'no',
        'no',
        'no',
        'no',
        'no',
        'no',
        'no',
        'yes',
        'no',
        'no',
        'no',
        'yes',
        'no',
        'no',
        'yes',
        'no',
        'no',
        'no',
        'no',
        'yes',
        'no',
        'yes',
        'no',
        'no',
        'no',
        'no',
        'no',
        'no',
        'yes',
        'no',
        'no',
        'yes',
```

```
           'no',
           'yes',
           'no',
           'no',
           'no']
```

[36]:
```
lib.loc[:,'Hotspot Loan'] = hotspot
lib
```

[36]:

	Branch Name	Phone Number	Email	Address	CityZip	lat	lon	Hotspot Loan
0	Vermont Square	(323) 290-7405	vmtsqr@lapl.org	1201 W. 48th. Street	Los Angeles, CA 90037	33.999699	-118.295799	no
1	Vernon - Leon H. Washington Jr. Memorial	(323) 234-9106	vrnonl@lapl.org	4504 S. Central Avenue	Los Angeles, CA 90011	34.002701	-118.256104	no
2	Washington Irving	(323) 734-6303	wirvng@lapl.org	4117 W. Washington Boulevard	Los Angeles, CA 90018	34.039951	-118.329147	yes
3	West Los Angeles Regional	(310) 575-8323	westla@lapl.org	11360 Santa Monica Boulevard	Los Angeles, CA 90025	34.0457	-118.450302	no
4	West Valley Regional	(818) 345-9806	wvally@lapl.org	19036 Vanowen Street	Reseda, CA 91335	34.193604	-118.546898	yes
..	…	…	…	…	…	…	…	…
68	Sunland - Tujunga	(818) 352-4481	snlndt@lapl.org	7771 Foothill Boulevard	Tujunga, CA 91042	34.258099	-118.301498	no
69	Sylmar	(818) 367-6102	sylmar@lapl.org	14561 Polk Street	Sylmar, CA 91342	34.30735	-118.45015	yes
70	Valley Plaza	(818) 765-9251	valplz@lapl.org	12311 Vanowen Street	North Hollywood, CA 91605	34.194302	-118.4011	no
71	Van Nuys	(818) 756-8453	vnnuys@lapl.org	6250 Sylmar Ave	Van Nuys, CA 91401	34.183899	-118.446297	no
72	Venice - Abbot Kinney Memorial	(310) 821-1769	venice@lapl.org	501 S. Venice Boulevard	Venice, CA 90291	33.987598	-118.465103	no

```
[73 rows x 8 columns]
```

[37]:
```
# trim city census tract data to prepare for spatial join
tracts.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 1310 entries, 0 to 1309
Data columns (total 34 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   OBJECTID       1310 non-null   int64
 1   NAME           1310 non-null   object
 2   Effective      1310 non-null   int64
 3   Revised        1310 non-null   int64
 4   District       1310 non-null   int64
 5   District_Name  1310 non-null   object
 6   TOOLTIP        1310 non-null   object
 7   NLA_URL        1310 non-null   object
 8   TRACTCE10      1310 non-null   object
 9   POP            1310 non-null   int64
 10  LATPOP_D       1310 non-null   int64
 11  WHIPOP_D       1310 non-null   int64
 12  BLAPOP_D       1310 non-null   int64
 13  AMIPOP_D       1310 non-null   int64
 14  ASIPOP_D       1310 non-null   int64
 15  HPIPOP_D       1310 non-null   int64
 16  OTHPOP_D       1310 non-null   int64
 17  MMRPOP_D       1310 non-null   int64
 18  VAP            1310 non-null   int64
 19  LATVAP_D       1310 non-null   int64
 20  WHIVAP_D       1310 non-null   int64
 21  BLAVAP_D       1310 non-null   int64
 22  AMIVAP_D       1310 non-null   int64
 23  ASIVAP_D       1310 non-null   int64
 24  HPIVAP_D       1310 non-null   int64
 25  OTHVAP_D       1310 non-null   int64
 26  MMRVAP_D       1310 non-null   int64
 27  H0010001       1310 non-null   int64
 28  H0010002       1310 non-null   int64
 29  H0010003       1310 non-null   int64
 30  TOOLTIP_1      1310 non-null   object
 31  NLA_URL_1      1310 non-null   object
 32  AnalysisArea   1310 non-null   float64
 33  geometry       1310 non-null   geometry
dtypes: float64(1), geometry(1), int64(25), object(7)
memory usage: 348.1+ KB
```

[38]:
```
tracts.sample(10)
```

[38]:

	OBJECTID	NAME	Effective	Revised	District	\
653	654	Mitch O'Farrell	1372636800000	1369785600000	13	
985	986	Herb J. Wesson Jr.	1341187200000	1358380800000	10	
1090	1091	Curren D. Price Jr.	1372636800000	1369785600000	9	
1030	1031	Herb J. Wesson Jr.	1341187200000	1358380800000	10	
519	520	David Ryu	1341187200000	1358380800000	4	
254	255	Bob Blumenfield	1372636800000	1369785600000	3	
621	622	Mitch O'Farrell	1372636800000	1369785600000	13	
522	523	David Ryu	1341187200000	1358380800000	4	
454	455	Paul Koretz	1341187200000	1358380800000	5	
716	717	Jose Huizar	1341187200000	1358380800000	14	

	District_Name	\
653	13 - Mitch O'Farrell	
985	10 - Herb J. Wesson Jr.	
1090	9 - Curren D. Price Jr.	
1030	10 - Herb J. Wesson Jr.	
519	4 - David Ryu	
254	3 - Bob Blumenfield	
621	13 - Mitch O'Farrell	
522	4 - David Ryu	
454	5 - Paul Koretz	
716	14 - Jose Huizar	

	TOOLTIP	\
653	Councilmember: Mitch O'Farrell\nDistrict: 13	
985	Councilmember: Herb J. Wesson Jr.\nDistrict: 10	
1090	Councilmember: Curren D. Price Jr.\nDistrict: 9	
1030	Councilmember: Herb J. Wesson Jr.\nDistrict: 10	
519	Councilmember: David Ryu\nDistrict: 4	
254	Councilmember: Bob Blumenfield\nDistrict: 3	
621	Councilmember: Mitch O'Farrell\nDistrict: 13	
522	Councilmember: David Ryu\nDistrict: 4	
454	Councilmember: Paul Koretz\nDistrict: 5	
716	Councilmember: Jose Huizar\nDistrict: 14	

	NLA_URL	TRACTCE10	POP	…	HPIVAP_D	\
653	http://cd13.lacity.org/?nla_win=p,75,75	192620	3868	…	2	
985	http://cd10.lacity.org/?nla_win=p,75,75	211703	4750	…	3	
1090	http://cd9.lacity.org/?nla_win=p,75,75	227010	4662	…	1	
1030	http://cd10.lacity.org/?nla_win=p,75,75	218220	4358	…	0	
519	http://cd4.lacity.org/?nla_win=p,75,75	143200	3743	…	7	
254	http://cd3.lacity.org/?nla_win=p,75,75	132900	3461	…	3	
621	http://cd13.lacity.org/?nla_win=p,75,75	190201	2870	…	7	
522	http://cd4.lacity.org/?nla_win=p,75,75	143603	2222	…	1	

```
454    http://cd5.lacity.org/?nla_win=p,75,75    265602  3042  …         7
716   http://cd14.lacity.org/?nla_win=p,75,75    186100  4715  …         9

      OTHVAP_D  MMRVAP_D  H0010001  H0010002  H0010003  \
653         11        14      1338      1253        85
985         24        20      2113      1954       159
1090         1         2      1059      1000        59
1030        17        44      1563      1436       127
519         10        26      2141      1986       155
254          5         7      1329      1275        54
621         10        21      1669      1488       181
522         10        15      1310      1244        66
454          8         8      1852      1575       277
716          9        20      1857      1756       101

                                  TOOLTIP_1  \
653   Tract: 1926.20\nTotal Population: 3868
985   Tract: 2117.03\nTotal Population: 4750
1090  Tract: 2270.10\nTotal Population: 4662
1030  Tract: 2182.20\nTotal Population: 4358
519   Tract: 1432.00\nTotal Population: 3743
254   Tract: 1329.00\nTotal Population: 3461
621   Tract: 1902.01\nTotal Population: 2870
522   Tract: 1436.03\nTotal Population: 2222
454   Tract: 2656.02\nTotal Population: 3042
716   Tract: 1861.00\nTotal Population: 4715

                                          NLA_URL_1  AnalysisArea  \
653   navigatela/reports/census_tracts_2010.cfm?PK=1…  9.403874e-02
985   navigatela/reports/census_tracts_2010.cfm?PK=2…  8.928390e-03
1090  navigatela/reports/census_tracts_2010.cfm?PK=2…  2.236293e-01
1030  navigatela/reports/census_tracts_2010.cfm?PK=2…  2.597610e-01
519   navigatela/reports/census_tracts_2010.cfm?PK=1…  1.608068e-01
254   navigatela/reports/census_tracts_2010.cfm?PK=1…  2.645149e-01
621   navigatela/reports/census_tracts_2010.cfm?PK=1…  8.755077e-02
522   navigatela/reports/census_tracts_2010.cfm?PK=1…  2.100000e-07
454   navigatela/reports/census_tracts_2010.cfm?PK=2…  1.296929e-01
716   navigatela/reports/census_tracts_2010.cfm?PK=1…  5.199968e-01

                                             geometry
653   POLYGON ((-118.29170 34.07633, -118.29297 34.0…
985   POLYGON ((-118.30911 34.06533, -118.30973 34.0…
1090  POLYGON ((-118.23967 34.01801, -118.23968 34.0…
1030  POLYGON ((-118.33431 34.04667, -118.33423 34.0…
519   POLYGON ((-118.37041 34.15287, -118.37041 34.1…
254   POLYGON ((-118.52731 34.18559, -118.52731 34.1…
621   POLYGON ((-118.32767 34.10544, -118.32764 34.1…
```

```
522   POLYGON ((-118.36169 34.14345, -118.36176 34.1…
454   POLYGON ((-118.42898 34.06089, -118.42996 34.0…
716   POLYGON ((-118.22067 34.12695, -118.22162 34.1…

[10 rows x 34 columns]
```

[39]:
```
list(tracts)
```

[39]:
```
['OBJECTID',
 'NAME',
 'Effective',
 'Revised',
 'District',
 'District_Name',
 'TOOLTIP',
 'NLA_URL',
 'TRACTCE10',
 'POP',
 'LATPOP_D',
 'WHIPOP_D',
 'BLAPOP_D',
 'AMIPOP_D',
 'ASIPOP_D',
 'HPIPOP_D',
 'OTHPOP_D',
 'MMRPOP_D',
 'VAP',
 'LATVAP_D',
 'WHIVAP_D',
 'BLAVAP_D',
 'AMIVAP_D',
 'ASIVAP_D',
 'HPIVAP_D',
 'OTHVAP_D',
 'MMRVAP_D',
 'H0010001',
 'H0010002',
 'H0010003',
 'TOOLTIP_1',
 'NLA_URL_1',
 'AnalysisArea',
 'geometry']
```

[40]:
```
#remove irrelevant columns
tract_col2keep = ['District',
                  'TRACTCE10',
                  'POP',
```

```
                  'TOOLTIP_1',
                  'geometry']
```

[41]:
```
#redifine
tracts = tracts[tract_col2keep]
```

[42]:
```
tracts.dtypes
```

[42]:
```
District        int64
TRACTCE10      object
POP             int64
TOOLTIP_1      object
geometry     geometry
dtype: object
```

[43]:
```
#rename TRACTCE10 to match larger dataframe from census reporter
tracts.columns = ['District',
                  'Census Tract',
                  'POP',
                  'TOOLTIP_1',
                  'geometry']
tracts
```

[43]:

	District	Census Tract	POP	TOOLTIP_1	\
0	7	101110	4731	Tract: 1011.10\nTotal Population: 4731	
1	7	101122	3664	Tract: 1011.22\nTotal Population: 3664	
2	7	101210	5990	Tract: 1012.10\nTotal Population: 5990	
3	7	101220	3363	Tract: 1012.20\nTotal Population: 3363	
4	7	101300	4199	Tract: 1013.00\nTotal Population: 4199	
…	…	…	…	…	
1305	11	701100	44	Tract: 7011.00\nTotal Population: 44	
1306	8	240800	4377	Tract: 2408.00\nTotal Population: 4377	
1307	15	240800	4377	Tract: 2408.00\nTotal Population: 4377	
1308	8	240900	5620	Tract: 2409.00\nTotal Population: 5620	
1309	15	240900	5620	Tract: 2409.00\nTotal Population: 5620	

	geometry
0	POLYGON ((-118.29793 34.26323, -118.29697 34.2…
1	POLYGON ((-118.27743 34.25991, -118.27747 34.2…
2	POLYGON ((-118.28594 34.25590, -118.28594 34.2…
3	POLYGON ((-118.27818 34.25577, -118.27824 34.2…
4	POLYGON ((-118.26528 34.25238, -118.26531 34.2…
…	…
1305	POLYGON ((-118.45963 34.06544, -118.45999 34.0…
1306	MULTIPOLYGON (((-118.25521 33.93652, -118.2552…
1307	MULTIPOLYGON (((-118.25521 33.93652, -118.2552…
1308	POLYGON ((-118.25521 33.93652, -118.25520 33.9…

```
1309  POLYGON ((-118.25521 33.93652, -118.25520 33.9…

[1310 rows x 5 columns]
```

[44]:
```
#begin exploration of the large dataset from census explorer
se.shape
```

[44]:
```
(2347, 163)
```

[45]:
```
se.sample(10)
```

[45]:
```
             FIPS Geographic Identifier             Area Name  \
1040  06037310701    14000US06037310701  Census Tract 3107.01
450   06037192420    14000US06037192420  Census Tract 1924.20
1016  06037301900    14000US06037301900     Census Tract 3019
48    06037106510    14000US06037106510  Census Tract 1065.10
1626  06037534900    14000US06037534900     Census Tract 5349
879   06037269800    14000US06037269800     Census Tract 2698
1674  06037541400    14000US06037541400     Census Tract 5414
1268  06037430002    14000US06037430002  Census Tract 4300.02
64    06037108104    14000US06037108104  Census Tract 1081.04
1178  06037405002    14000US06037405002  Census Tract 4050.02

                                      Qualifying Name  \
1040  Census Tract 3107.01, Los Angeles County, Cali…
450   Census Tract 1924.20, Los Angeles County, Cali…
1016  Census Tract 3019, Los Angeles County, California
48    Census Tract 1065.10, Los Angeles County, Cali…
1626  Census Tract 5349, Los Angeles County, California
879   Census Tract 2698, Los Angeles County, California
1674  Census Tract 5414, Los Angeles County, California
1268  Census Tract 4300.02, Los Angeles County, Cali…
64    Census Tract 1081.04, Los Angeles County, Cali…
1178  Census Tract 4050.02, Los Angeles County, Cali…

     State Postal Abbreviation Summary Level Geographic Component  \
1040                        ca           140                    00
450                         ca           140                    00
1016                        ca           140                    00
48                          ca           140                    00
1626                        ca           140                    00
879                         ca           140                    00
1674                        ca           140                    00
1268                        ca           140                    00
64                          ca           140                    00
1178                        ca           140                    00
```

```
     File identification Logical Record Number US  …  \
1040               ACSSF                0004910     …
450                ACSSF                0004320     …
1016               ACSSF                0004886     …
48                 ACSSF                0003918     …
1626               ACSSF                0005496     …
879                ACSSF                0004749     …
1674               ACSSF                0005544     …
1268               ACSSF                0005138     …
64                 ACSSF                0003934     …
1178               ACSSF                0005048     …

     Foreign-Born Population: Foreign-Born Population: 2010 or Later  \
1040                      952                                    131
450                      1752                                    211
1016                     5363                                   1175
48                       1639                                     87
1626                     2888                                    175
879                       972                                    182
1674                     2951                                    286
1268                     1906                                    210
64                        554                                     58
1178                     1294                                     75

     Foreign-Born Population: 2000 to 2009  \
1040                                   379
450                                    461
1016                                  1498
48                                     229
1626                                   652
879                                    189
1674                                   537
1268                                   225
64                                      87
1178                                   218

     Foreign-Born Population: 1990 to 1999  \
1040                                   182
450                                    512
1016                                   789
48                                     425
1626                                   730
879                                    281
1674                                   835
1268                                   492
64                                     170
1178                                   293
```

	Foreign-Born Population: Before 1990	\
1040	260	
450	568	
1016	1901	
48	898	
1626	1331	
879	320	
1674	1293	
1268	979	
64	239	
1178	708	

	% Foreign-Born Population: 2010 or Later	\
1040	13.76	
450	12.04	
1016	21.91	
48	5.31	
1626	6.06	
879	18.72	
1674	9.69	
1268	11.02	
64	10.47	
1178	5.8	

	% Foreign-Born Population: 2000 to 2009	\
1040	39.81	
450	26.31	
1016	27.93	
48	13.97	
1626	22.58	
879	19.44	
1674	18.2	
1268	11.8	
64	15.7	
1178	16.85	

	% Foreign-Born Population: 1990 to 1999	\
1040	19.12	
450	29.22	
1016	14.71	
48	25.93	
1626	25.28	
879	28.91	
1674	28.3	
1268	25.81	
64	30.69	

```
1178                                    22.64

     % Foreign-Born Population: Before 1990 geometry
1040                                    27.31     None
450                                     32.42     None
1016                                    35.45     None
48                                      54.79     None
1626                                    46.09     None
879                                     32.92     None
1674                                    43.82     None
1268                                    51.36     None
64                                      43.14     None
1178                                    54.71     None

[10 rows x 163 columns]
```

[46]:
```
list(se)
```

[46]:
```
['FIPS',
 'Geographic Identifier',
 'Area Name',
 'Qualifying Name',
 'State Postal Abbreviation',
 'Summary Level',
 'Geographic Component',
 'File identification',
 'Logical Record Number',
 'US',
 'Census Region',
 'Census Division',
 'State (Census Code)',
 'State (FIPS Code)',
 'County of current residence',
 'County Subdivision (FIPS)',
 'Place (FIPS Code)',
 'Place (State FIPS + Place FIPS)',
 'Census Tract',
 'Block Group',
 'Consolidated City',
 'American Indian Area/Alaska Native Area/Hawaiian Home Land (Census)',
 'American Indian Area/Alaska Native Area/Hawaiian Home Land (FIPS)',
 'American Indian Trust Land/Hawaiian Home Land Indicator',
 'American Indian Tribal Subdivision (Census)',
 'American Indian Tribal Subdivision (FIPS)',
 'Alaska Native Regional Corporation (FIPS)',
 'Metropolitan and Micropolitan Statistical Area',
 'Combined Statistical Area',
```

```
'Metropolitan Division',
'Metropolitan Area Central City',
'Metropolitan/Micropolitan Indicator Flag',
'New England City and Town Area',
'New England City and Town Area Division',
'Urban Area',
'Urban Area Central Place',
'Current Congressional District ***',
'State Legislative District Upper',
'State Legislative District Lower',
'Voting District',
'ZIP Code Tabulation Area (3-digit)',
'5-digit ZIP Code Tabulation Area',
'Subminor Civil Division (FIPS)',
'State-School District (Elementary)',
'State-School District (Secondary)',
'State-School District (Unified)',
'Urban/Rural',
'Principal City Indicator',
'Traffic Analysis Zone',
'Urban Growth Area',
'Tribal Tract',
'Tribal Block Group',
'Public Use Microdata Area - 5% File',
'Public Use Microdata Area - 1% File',
'Total Population',
'Population Density (Per Sq. Mile)',
'Area (Land)',
'Total Population:',
'Total Population: Under 5 Years',
'Total Population: 5 to 9 Years',
'Total Population: 10 to 14 Years',
'Total Population: 15 to 17 Years',
'Total Population: 18 to 24 Years',
'Total Population: 25 to 34 Years',
'Total Population: 35 to 44 Years',
'Total Population: 45 to 54 Years',
'Total Population: 55 to 64 Years',
'Total Population: 65 to 74 Years',
'Total Population: 75 to 84 Years',
'Total Population: 85 Years and Over',
'% Total Population: Under 5 Years',
'% Total Population: 5 to 9 Years',
'% Total Population: 10 to 14 Years',
'% Total Population: 15 to 17 Years',
'% Total Population: 18 to 24 Years',
'% Total Population: 25 to 34 Years',
```

```
 '% Total Population: 35 to 44 Years',
 '% Total Population: 45 to 54 Years',
 '% Total Population: 55 to 64 Years',
 '% Total Population: 65 to 74 Years',
 '% Total Population: 75 to 84 Years',
 '% Total Population: 85 Years and Over',
 'Population 25 Years and Over:',
 'Population 25 Years and Over: Less than High School',
 'Population 25 Years and Over: High School Graduate (Includes Equivalency)',
 'Population 25 Years and Over: Some College',
 "Population 25 Years and Over: Bachelor's Degree",
 "Population 25 Years and Over: Master's Degree",
 'Population 25 Years and Over: Professional School Degree',
 'Population 25 Years and Over: Doctorate Degree',
 '% Population 25 Years and Over: Less than High School',
 '% Population 25 Years and Over: High School Graduate (Includes Equivalency)',
 '% Population 25 Years and Over: Some College',
 "% Population 25 Years and Over: Bachelor's Degree",
 "% Population 25 Years and Over: Master's Degree",
 '% Population 25 Years and Over: Professional School Degree',
 '% Population 25 Years and Over: Doctorate Degree',
 'Population 3 Years and Over:',
 'Population 3 Years and Over: Enrolled in School',
 'Population 3 Years and Over: Not Enrolled in School',
 '% Population 3 Years and Over: Enrolled in School',
 '% Population 3 Years and Over: Not Enrolled in School',
 'Civilian Population 16 to 19 Years:',
 'Civilian Population 16 to 19 Years: Not High School Graduate, Not Enrolled
(Dropped Out)',
 'Civilian Population 16 to 19 Years: High School Graduate, or Enrolled (In
School)',
 '% Civilian Population 16 to 19 Years: Not High School Graduate, Not Enrolled
(Dropped Out)',
 '% Civilian Population 16 to 19 Years: High School Graduate, or Enrolled (In
School)',
 'Civilian Population in Labor Force 16 Years and Over:',
 'Civilian Population in Labor Force 16 Years and Over: Employed',
 'Civilian Population in Labor Force 16 Years and Over: Unemployed',
 '% Civilian Population in Labor Force 16 Years and Over: Employed',
 '% Civilian Population in Labor Force 16 Years and Over: Unemployed',
 'Households:',
 'Households: Less than $10,000',
 'Households: $10,000 to $14,999',
 'Households: $15,000 to $19,999',
 'Households: $20,000 to $24,999',
 'Households: $25,000 to $29,999',
 'Households: $30,000 to $34,999',
```

```
 'Households: $35,000 to $39,999',
 'Households: $40,000 to $44,999',
 'Households: $45,000 to $49,999',
 'Households: $50,000 to $59,999',
 'Households: $60,000 to $74,999',
 'Households: $75,000 to $99,999',
 'Households: $100,000 to $124,999',
 'Households: $125,000 to $149,999',
 'Households: $150,000 to $199,999',
 'Households: $200,000 or More',
 '% Households: Less than $10,000',
 '% Households: $10,000 to $14,999',
 '% Households: $15,000 to $19,999',
 '% Households: $20,000 to $24,999',
 '% Households: $25,000 to $29,999',
 '% Households: $30,000 to $34,999',
 '% Households: $35,000 to $39,999',
 '% Households: $40,000 to $44,999',
 '% Households: $45,000 to $49,999',
 '% Households: $50,000 to $59,999',
 '% Households: $60,000 to $74,999',
 '% Households: $75,000 to $99,999',
 '% Households: $100,000 to $124,999',
 '% Households: $125,000 to $149,999',
 '% Households: $150,000 to $199,999',
 '% Households: $200,000 or More',
 'Total Population: Native Born',
 'Total Population: Foreign Born',
 'Total Population: Foreign Born: Naturalized Citizen',
 'Total Population: Foreign Born: Not a Citizen',
 '% Total Population: Native Born',
 '% Total Population: Foreign Born',
 '% Total Population: Foreign Born: Naturalized Citizen',
 '% Total Population: Foreign Born: Not a Citizen',
 'Foreign-Born Population:',
 'Foreign-Born Population: 2010 or Later',
 'Foreign-Born Population: 2000 to 2009',
 'Foreign-Born Population: 1990 to 1999',
 'Foreign-Born Population: Before 1990',
 '% Foreign-Born Population: 2010 or Later',
 '% Foreign-Born Population: 2000 to 2009',
 '% Foreign-Born Population: 1990 to 1999',
 '% Foreign-Born Population: Before 1990',
 'geometry']
```

[47]:
```
se_col2keep = ['FIPS',
 'Census Tract',
```

```
'Total Population',
'Population Density (Per Sq. Mile)',
'Area (Land)',
'Total Population:',
'Total Population: Under 5 Years',
'Total Population: 5 to 9 Years',
'Total Population: 10 to 14 Years',
'Total Population: 15 to 17 Years',
'Total Population: 18 to 24 Years',
'Total Population: 25 to 34 Years',
'Total Population: 35 to 44 Years',
'Total Population: 45 to 54 Years',
'Total Population: 55 to 64 Years',
'Total Population: 65 to 74 Years',
'Total Population: 75 to 84 Years',
'Total Population: 85 Years and Over',
'% Population 25 Years and Over: Less than High School',
'% Population 25 Years and Over: High School Graduate (Includes Equivalency)',
'% Population 25 Years and Over: Some College',
"% Population 25 Years and Over: Bachelor's Degree",
"% Population 25 Years and Over: Master's Degree",
'% Population 25 Years and Over: Professional School Degree',
'% Population 25 Years and Over: Doctorate Degree',
'% Population 3 Years and Over: Enrolled in School',
'% Population 3 Years and Over: Not Enrolled in School',
'% Civilian Population 16 to 19 Years: Not High School Graduate, Not Enrolled␣
↪(Dropped Out)',
'% Civilian Population 16 to 19 Years: High School Graduate, or Enrolled (In␣
↪School)',
'% Civilian Population in Labor Force 16 Years and Over: Employed',
'% Civilian Population in Labor Force 16 Years and Over: Unemployed',
'Households:',
'Households: Less than $10,000',
'Households: $10,000 to $14,999',
'Households: $15,000 to $19,999',
'Households: $20,000 to $24,999',
'Households: $25,000 to $29,999',
'Households: $30,000 to $34,999',
'Households: $35,000 to $39,999',
'Households: $40,000 to $44,999',
'Households: $45,000 to $49,999',
'Households: $50,000 to $59,999',
'Households: $60,000 to $74,999',
'Households: $75,000 to $99,999',
'Households: $100,000 to $124,999',
'Households: $125,000 to $149,999',
'Households: $150,000 to $199,999',
```

```
 'Households: $200,000 or More',
 '% Households: Less than $10,000',
 '% Households: $10,000 to $14,999',
 '% Households: $15,000 to $19,999',
 '% Households: $20,000 to $24,999',
 '% Households: $25,000 to $29,999',
 '% Households: $30,000 to $34,999',
 '% Households: $35,000 to $39,999',
 '% Households: $40,000 to $44,999',
 '% Households: $45,000 to $49,999',
 '% Households: $50,000 to $59,999',
 '% Households: $60,000 to $74,999',
 '% Households: $75,000 to $99,999',
 '% Households: $100,000 to $124,999',
 '% Households: $125,000 to $149,999',
 '% Households: $150,000 to $199,999',
 '% Households: $200,000 or More',
 '% Total Population: Native Born',
 '% Total Population: Foreign Born',
 '% Total Population: Foreign Born: Naturalized Citizen',
 '% Total Population: Foreign Born: Not a Citizen',
 '% Foreign-Born Population: 2010 or Later',
 '% Foreign-Born Population: 2000 to 2009',
 '% Foreign-Born Population: 1990 to 1999',
 '% Foreign-Born Population: Before 1990']
```

[48]:
```
se = se[se_col2keep]
se
```

[48]:

	FIPS	Census Tract	Total Population
0	Geo_FIPS	Geo_TRACT	SE_A00002_001
1	06037101110	101110	4283
2	06037101122	101122	3405
3	06037101210	101210	6347
4	06037101220	101220	3702
…	…	…	…
2342	06037980031	980031	1175
2343	06037980033	980033	16
2344	06037990100	990100	0
2345	06037990200	990200	0
2346	06037990300	990300	0

	Population Density (Per Sq. Mile)	Area (Land)	Total Population:
0	SE_A00002_002	SE_A00002_003	SE_A06001_001
1	9710.18	0.441083510811633	4283
2	3335.385	1.02087152527348	3405
3	25263.42	0.251232824244746	6347

```
4                              13719.17  0.269841404670601              3702
…                                 …                 …              …
2342                           240.7254   4.88107975789849              1175
2343                           3.271574   4.89061184839466                16
2344                                                    0                 0
2345                                                    0                 0
2346                                                    0                 0

     Total Population: Under 5 Years Total Population: 5 to 9 Years  \
0                       SE_A01001_002                  SE_A01001_003
1                                 283                            259
2                                  78                            105
3                                 422                            477
4                                 138                            162
…                                 …                              …
2342                               20                              0
2343                                0                              0
2344                                0                              0
2345                                0                              0
2346                                0                              0

     Total Population: 10 to 14 Years Total Population: 15 to 17 Years  …  \
0                        SE_A01001_004                  SE_A01001_005  …
1                                 200                            132  …
2                                 234                             12  …
3                                 383                            198  …
4                                 309                            142  …
…                                 …                              …  …
2342                                5                              0  …
2343                                0                              0  …
2344                                0                              0  …
2345                                0                              0  …
2346                                0                              0  …

     % Households: $150,000 to $199,999 % Households: $200,000 or More  \
0                      PCT_SE_A14001_016              PCT_SE_A14001_017
1                                   7.94                           4.89
2                                   8.77                          13.72
3                                   2.31                           2.53
4                                   5.08                           3.02
…                                   …                              …
2342                                   0                          20.83
2343                                   0                              0
2344
2345
2346
```

```
     % Total Population: Native Born % Total Population: Foreign Born  \
0                  PCT_SE_A06001_002                PCT_SE_A06001_003
1                              66.75                            33.25
2                              76.21                            23.79
3                               50.1                             49.9
4                              52.76                            47.24
…                              …                                …
2342                           87.57                            12.43
2343                             100                                0
2344
2345
2346

     % Total Population: Foreign Born: Naturalized Citizen  \
0                                    PCT_SE_A06001_004
1                                                20.64
2                                                21.91
3                                                30.63
4                                                 33.5
…                                                 …
2342                                              9.53
2343                                                 0
2344
2345
2346

     % Total Population: Foreign Born: Not a Citizen  \
0                                  PCT_SE_A06001_005
1                                              12.61
2                                               1.88
3                                              19.27
4                                              13.75
…                                              …
2342                                            2.89
2343                                               0
2344
2345
2346

     % Foreign-Born Population: 2010 or Later  \
0                          PCT_SE_A10058_002
1                                      15.24
2                                       5.56
3                                       8.56
4                                      10.52
…                                      …
2342                                      0
```

```
2343
2344
2345
2346

     % Foreign-Born Population: 2000 to 2009  \
0                          PCT_SE_A10058_003
1                                      25.49
2                                        5.8
3                                       35.4
4                                      27.73
…                                          …
2342                                   16.44
2343
2344
2345
2346

     % Foreign-Born Population: 1990 to 1999  \
0                          PCT_SE_A10058_004
1                                      23.38
2                                      40.49
3                                      24.31
4                                      26.01
…                                          …
2342                                   32.88
2343
2344
2345
2346

     % Foreign-Born Population: Before 1990
0                          PCT_SE_A10058_005
1                                      35.88
2                                      48.15
3                                      31.73
4                                      35.73
…                                          …
2342                                   50.68
2343
2344
2345
2346

[2347 rows x 72 columns]
```

[49]:
```
#drop the first row
se.drop([0])
```

[49]:

	FIPS	Census Tract	Total Population	\
1	06037101110	101110	4283	
2	06037101122	101122	3405	
3	06037101210	101210	6347	
4	06037101220	101220	3702	
5	06037101300	101300	3884	
…	…	…	…	
2342	06037980031	980031	1175	
2343	06037980033	980033	16	
2344	06037990100	990100	0	
2345	06037990200	990200	0	
2346	06037990300	990300	0	

	Population Density (Per Sq. Mile)	Area (Land)	Total Population:	\
1	9710.18	0.441083510811633	4283	
2	3335.385	1.02087152527348	3405	
3	25263.42	0.251232824244746	6347	
4	13719.17	0.269841404670601	3702	
5	3897.202	0.996612339516631	3884	
…	…	…	…	
2342	240.7254	4.88107975789849	1175	
2343	3.271574	4.89061184839466	16	
2344		0	0	
2345		0	0	
2346		0	0	

	Total Population: Under 5 Years	Total Population: 5 to 9 Years	\
1	283	259	
2	78	105	
3	422	477	
4	138	162	
5	89	214	
…	…	…	
2342	20	0	
2343	0	0	
2344	0	0	
2345	0	0	
2346	0	0	

	Total Population: 10 to 14 Years	Total Population: 15 to 17 Years	…	\
1	200	132	…	
2	234	12	…	
3	383	198	…	
4	309	142	…	

```
5                                   93                                    43  …
…                                   …                                     …   …
2342                                 5                                     0  …
2343                                 0                                     0  …
2344                                 0                                     0  …
2345                                 0                                     0  …
2346                                 0                                     0  …

     % Households: $150,000 to $199,999 % Households: $200,000 or More  \
1                                  7.94                              4.89
2                                  8.77                             13.72
3                                  2.31                              2.53
4                                  5.08                              3.02
5                                    11                             12.32
…                                   …                                 …
2342                                 0                             20.83
2343                                 0                                 0
2344
2345
2346

     % Total Population: Native Born % Total Population: Foreign Born  \
1                              66.75                             33.25
2                              76.21                             23.79
3                               50.1                              49.9
4                              52.76                             47.24
5                              57.31                             42.69
…                              …                                 …
2342                           87.57                             12.43
2343                             100                                 0
2344
2345
2346

     % Total Population: Foreign Born: Naturalized Citizen  \
1                                                    20.64
2                                                    21.91
3                                                    30.63
4                                                     33.5
5                                                    32.98
…                                                    …
2342                                                  9.53
2343                                                     0
2344
2345
2346
```

```
     % Total Population: Foreign Born: Not a Citizen  \
1                                                12.61
2                                                 1.88
3                                                19.27
4                                                13.75
5                                                 9.71
…                                                    …
2342                                              2.89
2343                                                 0
2344
2345
2346

     % Foreign-Born Population: 2010 or Later  \
1                                       15.24
2                                        5.56
3                                        8.56
4                                       10.52
5                                       11.58
…                                           …
2342                                        0
2343
2344
2345
2346

     % Foreign-Born Population: 2000 to 2009  \
1                                      25.49
2                                        5.8
3                                       35.4
4                                      27.73
5                                      17.07
…                                          …
2342                                   16.44
2343
2344
2345
2346

     % Foreign-Born Population: 1990 to 1999  \
1                                      23.38
2                                      40.49
3                                      24.31
4                                      26.01
5                                         19
…                                          …
2342                                   32.88
```

```
2343
2344
2345
2346
```

```
     % Foreign-Born Population: Before 1990
1                                      35.88
2                                      48.15
3                                      31.73
4                                      35.73
5                                      52.35
…                                          …
2342                                   50.68
2343
2344
2345
2346

[2346 rows x 72 columns]
```

[50]:
```
#I want to join the two data frames by the Census Tract columns
census = pd.merge(se, tracts, on='Census Tract', how='outer')
census
```

[50]:
```
              FIPS Census Tract Total Population  \
0        Geo_FIPS    Geo_TRACT    SE_A00002_001
1     06037101110       101110             4283
2     06037101122       101122             3405
3     06037101210       101210             6347
4     06037101220       101220             3702
…           …           …                  …
2642  06037980033       980033               16
2643  06037990100       990100                0
2644  06037990200       990200                0
2645  06037990300       990300                0
2646          NaN       930401              NaN

     Population Density (Per Sq. Mile)        Area (Land) Total Population:  \
0                          SE_A00002_002      SE_A00002_003     SE_A06001_001
1                                9710.18  0.441083510811633              4283
2                               3335.385   1.02087152527348              3405
3                               25263.42  0.251232824244746              6347
4                               13719.17  0.269841404670601              3702
…                                    …                  …                 …
2642                            3.271574   4.89061184839466                16
2643                                                     0                 0
2644                                                     0                 0
```

```
2645                                                 0                 0
2646                                 NaN               NaN               NaN

     Total Population: Under 5 Years Total Population: 5 to 9 Years  \
0                       SE_A01001_002                 SE_A01001_003
1                                 283                           259
2                                  78                           105
3                                 422                           477
4                                 138                           162
…                                 …                             …
2642                                0                             0
2643                                0                             0
2644                                0                             0
2645                                0                             0
2646                              NaN                           NaN

     Total Population: 10 to 14 Years Total Population: 15 to 17 Years  …  \
0                        SE_A01001_004                 SE_A01001_005  …
1                                  200                           132  …
2                                  234                            12  …
3                                  383                           198  …
4                                  309                           142  …
…                                  …                             …  …
2642                                0                             0  …
2643                                0                             0  …
2644                                0                             0  …
2645                                0                             0  …
2646                              NaN                           NaN  …

     % Total Population: Foreign Born: Naturalized Citizen  \
0                                    PCT_SE_A06001_004
1                                                20.64
2                                                21.91
3                                                30.63
4                                                 33.5
…                                                 …
2642                                                 0
2643
2644
2645
2646                                               NaN

     % Total Population: Foreign Born: Not a Citizen  \
0                                 PCT_SE_A06001_005
1                                             12.61
2                                              1.88
3                                             19.27
```

```
4                                                 13.75
…                                                 …
2642                                                  0
2643
2644
2645
2646                                                NaN

     % Foreign-Born Population: 2010 or Later  \
0                           PCT_SE_A10058_002
1                                       15.24
2                                        5.56
3                                        8.56
4                                       10.52
…                                       …
2642
2643
2644
2645
2646                                      NaN

     % Foreign-Born Population: 2000 to 2009  \
0                           PCT_SE_A10058_003
1                                       25.49
2                                         5.8
3                                        35.4
4                                       27.73
…                                       …
2642
2643
2644
2645
2646                                      NaN

     % Foreign-Born Population: 1990 to 1999  \
0                           PCT_SE_A10058_004
1                                       23.38
2                                       40.49
3                                       24.31
4                                       26.01
…                                       …
2642
2643
2644
2645
2646                                      NaN
```

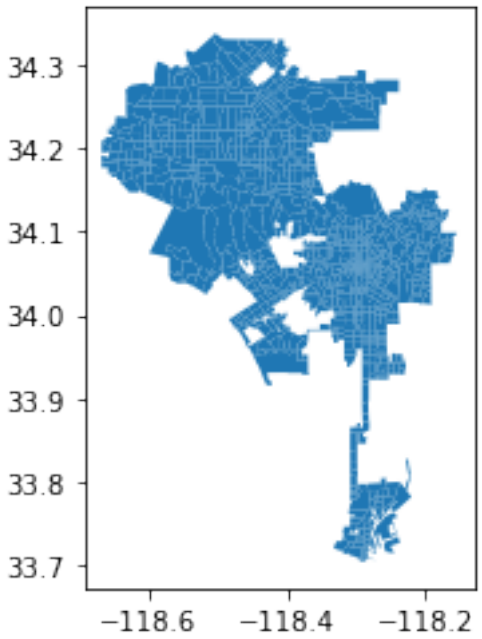
```
     % Foreign-Born Population: Before 1990 District     POP  \
0                          PCT_SE_A10058_005      NaN     NaN
1                                      35.88      7.0  4731.0
2                                      48.15      7.0  3664.0
3                                      31.73      7.0  5990.0
4                                      35.73      7.0  3363.0
…                                      …       …     …
2642                                                  NaN     NaN
2643                                                  NaN     NaN
2644                                                  NaN     NaN
2645                                                  NaN     NaN
2646                                        NaN       3.0  2471.0

                                    TOOLTIP_1  \
0                                         NaN
1     Tract: 1011.10\nTotal Population: 4731
2     Tract: 1011.22\nTotal Population: 3664
3     Tract: 1012.10\nTotal Population: 5990
4     Tract: 1012.20\nTotal Population: 3363
…                                        …
2642                                      NaN
2643                                      NaN
2644                                      NaN
2645                                      NaN
2646  Tract: 9304.01\nTotal Population: 2471

                                                 geometry
0                                                    None
1     POLYGON ((-118.29793 34.26323, -118.29697 34.2…
2     POLYGON ((-118.27743 34.25991, -118.27747 34.2…
3     POLYGON ((-118.28594 34.25590, -118.28594 34.2…
4     POLYGON ((-118.27818 34.25577, -118.27824 34.2…
…                                                   …
2642                                                 None
2643                                                 None
2644                                                 None
2645                                                 None
2646  POLYGON ((-118.62331 34.16446, -118.62420 34.1…

[2647 rows x 76 columns]
```

[51]:
```
#the data from Social Explorer was for the entire county, now that the tables␣
 ↪are joined I can delete the rows of data not within the City

#convert df to gdf
census = gpd.GeoDataFrame(census, crs="EPSG:4326")
type(census)
```

[51]:
```
geopandas.geodataframe.GeoDataFrame
```

[52]:
```
#check to see if transition was successful
census.plot()
```

[52]:
```
<AxesSubplot:>
```



[53]:
```
#Remove non-visible rows
census.dropna(inplace=True)
```

[54]:
```
#check to see if transition was successful
census
```

[54]:

	FIPS	Census Tract	Total Population	\
1	06037101110	101110	4283	
2	06037101122	101122	3405	
3	06037101210	101210	6347	
4	06037101220	101220	3702	
5	06037101300	101300	3884	
…	…	…	…	
2635	06037980024	980024	223	
2636	06037980024	980024	223	
2638	06037980026	980026	26	
2639	06037980028	980028	0	
2641	06037980031	980031	1175	

```
     Population Density (Per Sq. Mile)        Area (Land) Total Population:  \
1                              9710.18  0.441083510811633              4283
2                             3335.385   1.02087152527348              3405
3                             25263.42  0.251232824244746              6347
4                             13719.17  0.269841404670601              3702
5                             3897.202  0.996612339516631              3884
…                                   …                  …                 …
2635                          72.20638   3.08836990750536               223
2636                          72.20638   3.08836990750536               223
2638                          4.856592   5.35354874231078                26
2639                                 0   6.67946106314006                 0
2641                          240.7254   4.88107975789849              1175

     Total Population: Under 5 Years Total Population: 5 to 9 Years  \
1                                 283                           259
2                                  78                           105
3                                 422                           477
4                                 138                           162
5                                  89                           214
…                                   …                             …
2635                               16                            25
2636                               16                            25
2638                                0                             0
2639                                0                             0
2641                               20                             0

     Total Population: 10 to 14 Years Total Population: 15 to 17 Years  …  \
1                                 200                            132  …
2                                 234                             12  …
3                                 383                            198  …
4                                 309                            142  …
5                                  93                             43  …
…                                   …                             …  …
2635                                9                              0  …
2636                                9                              0  …
2638                                0                             15  …
2639                                0                              0  …
2641                                5                              0  …

     % Total Population: Foreign Born: Naturalized Citizen  \
1                                              20.64
2                                              21.91
3                                              30.63
4                                               33.5
5                                              32.98
…                                                  …
2635                                            8.52
```

```
2636                                                 8.52
2638                                                    0
2639
2641                                                 9.53

     % Total Population: Foreign Born: Not a Citizen  \
1                                               12.61
2                                                1.88
3                                               19.27
4                                               13.75
5                                                9.71
…                                                …
2635                                             1.79
2636                                             1.79
2638                                                0
2639
2641                                             2.89

     % Foreign-Born Population: 2010 or Later  \
1                                       15.24
2                                        5.56
3                                        8.56
4                                       10.52
5                                       11.58
…                                       …
2635                                        0
2636                                        0
2638
2639
2641                                        0

     % Foreign-Born Population: 2000 to 2009  \
1                                      25.49
2                                        5.8
3                                       35.4
4                                      27.73
5                                      17.07
…                                      …
2635                                   34.78
2636                                   34.78
2638
2639
2641                                   16.44

     % Foreign-Born Population: 1990 to 1999  \
1                                      23.38
2                                      40.49
```

```
3                                    24.31
4                                    26.01
5                                       19
…                                    …
2635                                 43.48
2636                                 43.48
2638
2639
2641                                 32.88

     % Foreign-Born Population: Before 1990 District     POP  \
1                                    35.88      7.0  4731.0
2                                    48.15      7.0  3664.0
3                                    31.73      7.0  5990.0
4                                    35.73      7.0  3363.0
5                                    52.35      7.0  4199.0
…                                    …      …     …
2635                                 21.74      5.0   186.0
2636                                 21.74      4.0   186.0
2638                                            7.0    20.0
2639                                           11.0     4.0
2641                                 50.68     15.0  1262.0

                                  TOOLTIP_1  \
1     Tract: 1011.10\nTotal Population: 4731
2     Tract: 1011.22\nTotal Population: 3664
3     Tract: 1012.10\nTotal Population: 5990
4     Tract: 1012.20\nTotal Population: 3363
5     Tract: 1013.00\nTotal Population: 4199
…                                    …
2635   Tract: 9800.24\nTotal Population: 186
2636   Tract: 9800.24\nTotal Population: 186
2638    Tract: 9800.26\nTotal Population: 20
2639     Tract: 9800.28\nTotal Population: 4
2641  Tract: 9800.31\nTotal Population: 1262

                                                 geometry
1     POLYGON ((-118.29793 34.26323, -118.29697 34.2…
2     POLYGON ((-118.27743 34.25991, -118.27747 34.2…
3     POLYGON ((-118.28594 34.25590, -118.28594 34.2…
4     POLYGON ((-118.27818 34.25577, -118.27824 34.2…
5     POLYGON ((-118.26528 34.25238, -118.26531 34.2…
…                                                   …
2635  MULTIPOLYGON (((-118.46942 34.16092, -118.4693…
2636  MULTIPOLYGON (((-118.46917 34.17875, -118.4696…
2638  POLYGON ((-118.26528 34.25238, -118.26539 34.2…
2639  POLYGON ((-118.38574 33.95531, -118.38586 33.9…
```

```
2641  MULTIPOLYGON (((-118.24568 33.76382, -118.2454…

[1309 rows x 76 columns]
```

[55]:
```
list(census)
```

[55]:
```
['FIPS',
 'Census Tract',
 'Total Population',
 'Population Density (Per Sq. Mile)',
 'Area (Land)',
 'Total Population:',
 'Total Population: Under 5 Years',
 'Total Population: 5 to 9 Years',
 'Total Population: 10 to 14 Years',
 'Total Population: 15 to 17 Years',
 'Total Population: 18 to 24 Years',
 'Total Population: 25 to 34 Years',
 'Total Population: 35 to 44 Years',
 'Total Population: 45 to 54 Years',
 'Total Population: 55 to 64 Years',
 'Total Population: 65 to 74 Years',
 'Total Population: 75 to 84 Years',
 'Total Population: 85 Years and Over',
 '% Population 25 Years and Over: Less than High School',
 '% Population 25 Years and Over: High School Graduate (Includes Equivalency)',
 '% Population 25 Years and Over: Some College',
 "% Population 25 Years and Over: Bachelor's Degree",
 "% Population 25 Years and Over: Master's Degree",
 '% Population 25 Years and Over: Professional School Degree',
 '% Population 25 Years and Over: Doctorate Degree',
 '% Population 3 Years and Over: Enrolled in School',
 '% Population 3 Years and Over: Not Enrolled in School',
 '% Civilian Population 16 to 19 Years: Not High School Graduate, Not Enrolled
(Dropped Out)',
 '% Civilian Population 16 to 19 Years: High School Graduate, or Enrolled (In
School)',
 '% Civilian Population in Labor Force 16 Years and Over: Employed',
 '% Civilian Population in Labor Force 16 Years and Over: Unemployed',
 'Households:',
 'Households: Less than $10,000',
 'Households: $10,000 to $14,999',
 'Households: $15,000 to $19,999',
 'Households: $20,000 to $24,999',
 'Households: $25,000 to $29,999',
 'Households: $30,000 to $34,999',
 'Households: $35,000 to $39,999',
```

```
 'Households: $40,000 to $44,999',
 'Households: $45,000 to $49,999',
 'Households: $50,000 to $59,999',
 'Households: $60,000 to $74,999',
 'Households: $75,000 to $99,999',
 'Households: $100,000 to $124,999',
 'Households: $125,000 to $149,999',
 'Households: $150,000 to $199,999',
 'Households: $200,000 or More',
 '% Households: Less than $10,000',
 '% Households: $10,000 to $14,999',
 '% Households: $15,000 to $19,999',
 '% Households: $20,000 to $24,999',
 '% Households: $25,000 to $29,999',
 '% Households: $30,000 to $34,999',
 '% Households: $35,000 to $39,999',
 '% Households: $40,000 to $44,999',
 '% Households: $45,000 to $49,999',
 '% Households: $50,000 to $59,999',
 '% Households: $60,000 to $74,999',
 '% Households: $75,000 to $99,999',
 '% Households: $100,000 to $124,999',
 '% Households: $125,000 to $149,999',
 '% Households: $150,000 to $199,999',
 '% Households: $200,000 or More',
 '% Total Population: Native Born',
 '% Total Population: Foreign Born',
 '% Total Population: Foreign Born: Naturalized Citizen',
 '% Total Population: Foreign Born: Not a Citizen',
 '% Foreign-Born Population: 2010 or Later',
 '% Foreign-Born Population: 2000 to 2009',
 '% Foreign-Born Population: 1990 to 1999',
 '% Foreign-Born Population: Before 1990',
 'District',
 'POP',
 'TOOLTIP_1',
 'geometry']
```

[56]:
```
census['geometry_cent'] = census['geometry'].centroid
```

```
/tmp/ipykernel_70/2659745872.py:1: UserWarning: Geometry is in a geographic CRS.
Results from 'centroid' are likely incorrect. Use 'GeoSeries.to_crs()' to re-
project geometries to a projected CRS before this operation.

  census['geometry_cent'] = census['geometry'].centroid
```

[57]:
```
#create % without high school education from census
edu1 = ['FIPS',
        'Census Tract',
        'Population Density (Per Sq. Mile)',
        'Total Population',
        '% Population 25 Years and Over: Less than High School',
        'geometry',
        'geometry_cent']
```

[58]:
```
census_edu1 = census[edu1]
```

[59]:
```
#there are some empty cells in the data that the NaN didn't drop, must find␣
 ↪them and drop to continue
census_edu1['% Population 25 Years and Over: Less than High School'].
 ↪replace('', np.nan, inplace=True)
```

```
/opt/conda/lib/python3.9/site-packages/pandas/core/generic.py:6619:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
```

[60]:
```
census_edu1.dropna(inplace=True)
```

```
/opt/conda/lib/python3.9/site-packages/pandas/util/_decorators.py:311:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return func(*args, **kwargs)
```

[61]:
```
census_edu1
```

[61]:

	FIPS	Census Tract	Population Density (Per Sq. Mile)	\
1	06037101110	101110	9710.18	
2	06037101122	101122	3335.385	
3	06037101210	101210	25263.42	
4	06037101220	101220	13719.17	
5	06037101300	101300	3897.202	
…	…	…	…	
2634	06037980024	980024	72.20638	
2635	06037980024	980024	72.20638	
2636	06037980024	980024	72.20638	
2638	06037980026	980026	4.856592	

```
2641  06037980031       980031                              240.7254

     Total Population % Population 25 Years and Over: Less than High School  \
1                4283                                                  15.17
2                3405                                                   6.1
3                6347                                                  16.06
4                3702                                                  19.92
5                3884                                                  10.24
…                 …                                                     …
2634              223                                                   3.16
2635              223                                                   3.16
2636              223                                                   3.16
2638               26                                                      0
2641             1175                                                      8

                                                 geometry  \
1     POLYGON ((-118.29793 34.26323, -118.29697 34.2…
2     POLYGON ((-118.27743 34.25991, -118.27747 34.2…
3     POLYGON ((-118.28594 34.25590, -118.28594 34.2…
4     POLYGON ((-118.27818 34.25577, -118.27824 34.2…
5     POLYGON ((-118.26528 34.25238, -118.26531 34.2…
…                                                   …
2634  POLYGON ((-118.47381 34.18560, -118.47381 34.1…
2635  MULTIPOLYGON (((-118.46942 34.16092, -118.4693…
2636  MULTIPOLYGON (((-118.46917 34.17875, -118.4696…
2638  POLYGON ((-118.26528 34.25238, -118.26539 34.2…
2641  MULTIPOLYGON (((-118.24568 33.76382, -118.2454…

                    geometry_cent
1     POINT (-118.29300 34.25948)
2     POINT (-118.29016 34.26773)
3     POINT (-118.29074 34.25298)
4     POINT (-118.28164 34.25161)
5     POINT (-118.27101 34.24878)
…                             …
2634  POINT (-118.48850 34.17693)
2635  POINT (-118.49298 34.16860)
2636  POINT (-118.46949 34.17895)
2638  POINT (-118.27924 34.27519)
2641  POINT (-118.26404 33.74316)

[1301 rows x 7 columns]
```

[62]:
```
census_edu1['% Population 25 Years and Over: Less than High␣
 ↪School']=census_edu1['% Population 25 Years and Over: Less than High␣
 ↪School'].astype(float)
census_edu1.info()
```

```
/opt/conda/lib/python3.9/site-packages/geopandas/geodataframe.py:1322:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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

```
<class 'geopandas.geodataframe.GeoDataFrame'>
Int64Index: 1301 entries, 1 to 2641
Data columns (total 7 columns):
 #   Column                                                Non-Null Count
Dtype
---  ------                                                --------------
-----
 0   FIPS                                                  1301 non-null
object
 1   Census Tract                                         1301 non-null
object
 2   Population Density (Per Sq. Mile)                     1301 non-null
object
 3   Total Population                                      1301 non-null
object
 4   % Population 25 Years and Over: Less than High School  1301 non-null
float64
 5   geometry                                              1301 non-null
geometry
 6   geometry_cent                                         1301 non-null
geometry
dtypes: float64(1), geometry(2), object(4)
memory usage: 81.3+ KB
```

[63]:
```
#create variable for percent of children not enrolled in school
edu2 = ['FIPS',
        'Census Tract',
        '% Population 3 Years and Over: Not Enrolled in School',
        'geometry']
```

[64]:
```
census_edu2 = census[edu2]
```

[65]:
```
census_edu2['% Population 3 Years and Over: Not Enrolled in School'].
↪replace('', np.nan, inplace=True)
```

```
/opt/conda/lib/python3.9/site-packages/pandas/core/generic.py:6619:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
```

[66]:
```
census_edu1.dropna(inplace=True)
```

```
/opt/conda/lib/python3.9/site-packages/pandas/util/_decorators.py:311:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return func(*args, **kwargs)
```

[67]:
```
census_edu2['% Population 3 Years and Over: Not Enrolled in␣
↪School']=census_edu2['% Population 3 Years and Over: Not Enrolled in␣
↪School'].astype(float)
census_edu2.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
Int64Index: 1309 entries, 1 to 2641
Data columns (total 4 columns):
 #   Column                                                Non-Null Count
Dtype
---  ------                                                --------------
-----
 0   FIPS                                                  1309 non-null
object
 1   Census Tract                                         1309 non-null
object
 2   % Population 3 Years and Over: Not Enrolled in School  1303 non-null
float64
 3   geometry                                              1309 non-null
geometry
dtypes: float64(1), geometry(1), object(2)
memory usage: 51.1+ KB
```

[68]:
```
edu3 = ['FIPS',
        'Census Tract',
        '% Civilian Population 16 to 19 Years: Not High School Graduate, Not␣
↪Enrolled (Dropped Out)',
        'geometry']
```

[69]:
```
census_edu3 = census[edu3]
```

[70]:
```
census_edu3['% Civilian Population 16 to 19 Years: Not High School Graduate,␣
↪Not Enrolled (Dropped Out)'].replace('', np.nan, inplace=True)
```

[71]:
```
census_edu3.dropna(inplace=True)
```

[72]:
```
census_edu3['% Civilian Population 16 to 19 Years: Not High School Graduate,␣
↪Not Enrolled (Dropped Out)']=census_edu3['% Civilian Population 16 to 19␣
↪Years: Not High School Graduate, Not Enrolled (Dropped Out)'].astype(float)
census_edu3.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
Int64Index: 1273 entries, 1 to 2638
Data columns (total 4 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   FIPS
1273 non-null   object
 1   Census Tract
1273 non-null   object
 2   % Civilian Population 16 to 19 Years: Not High School Graduate, Not
Enrolled (Dropped Out)  1273 non-null   float64
 3   geometry
1273 non-null   geometry
dtypes: float64(1), geometry(1), object(2)
memory usage: 49.7+ KB
```

[73]:
```
#create Foreign Born variable
fb = ['FIPS',
      'Census Tract',
      'Population Density (Per Sq. Mile)',
      '% Total Population: Foreign Born',
      'geometry',
      'geometry_cent']
```

[74]:
```
census_fb = census[fb]
```

[75]:
```
census_fb['% Total Population: Foreign Born'].replace('', np.nan, inplace=True)
```

[76]:
```
census_fb['% Total Population: Foreign Born']=census_fb['% Total Population:␣
↪Foreign Born'].astype(float)
census_fb['Population Density (Per Sq. Mile)']=census_fb['Population Density␣
↪(Per Sq. Mile)'].astype(float)
census_fb.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
Int64Index: 1309 entries, 1 to 2641
Data columns (total 6 columns):
 #   Column                              Non-Null Count  Dtype
```

```
---  ------                                ---------------  -----
 0   FIPS                                  1309 non-null   object
 1   Census Tract                          1309 non-null   object
 2   Population Density (Per Sq. Mile)     1309 non-null   float64
 3   % Total Population: Foreign Born      1303 non-null   float64
 4   geometry                              1309 non-null   geometry
 5   geometry_cent                         1309 non-null   geometry
dtypes: float64(2), geometry(2), object(2)
memory usage: 71.6+ KB
```

[77]:
```
#create an age variable from the larger census dataframe
age2 = ['FIPS',
       'Census Tract',
       'Population Density (Per Sq. Mile)',
       'Total Population:',
       'Total Population: Under 5 Years',
       'Total Population: 5 to 9 Years',
       'Total Population: 10 to 14 Years',
       'Total Population: 15 to 17 Years',
       'Total Population: 18 to 24 Years',
       'Total Population: 25 to 34 Years',
       'Total Population: 35 to 44 Years',
       'Total Population: 45 to 54 Years',
       'Total Population: 55 to 64 Years',
       'Total Population: 65 to 74 Years',
       'Total Population: 75 to 84 Years',
       'geometry',
       'geometry_cent']
```

[78]:
```
census_age = census[age2]
```

[79]:
```
census_age.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
Int64Index: 1309 entries, 1 to 2641
Data columns (total 17 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   FIPS                                  1309 non-null   object
 1   Census Tract                          1309 non-null   object
 2   Population Density (Per Sq. Mile)     1309 non-null   object
 3   Total Population:                     1309 non-null   object
 4   Total Population: Under 5 Years       1309 non-null   object
 5   Total Population: 5 to 9 Years        1309 non-null   object
 6   Total Population: 10 to 14 Years      1309 non-null   object
 7   Total Population: 15 to 17 Years      1309 non-null   object
 8   Total Population: 18 to 24 Years      1309 non-null   object
 9   Total Population: 25 to 34 Years      1309 non-null   object
```

```
 10  Total Population: 35 to 44 Years   1309 non-null   object
 11  Total Population: 45 to 54 Years   1309 non-null   object
 12  Total Population: 55 to 64 Years   1309 non-null   object
 13  Total Population: 65 to 74 Years   1309 non-null   object
 14  Total Population: 75 to 84 Years   1309 non-null   object
 15  geometry                           1309 non-null   geometry
 16  geometry_cent                      1309 non-null   geometry
dtypes: geometry(2), object(15)
memory usage: 184.1+ KB
```

[80]:
```
#Change data type from object to integer
census_age['Total Population:'] = census_age['Total Population:'].astype(int)
census_age['Total Population: Under 5 Years'] = census_age['Total Population:␣
 ↪Under 5 Years'].astype(int)
census_age['Total Population: 5 to 9 Years'] = census_age['Total Population: 5␣
 ↪to 9 Years'].astype(int)
census_age['Total Population: 10 to 14 Years'] = census_age['Total Population:␣
 ↪10 to 14 Years'].astype(int)
census_age['Total Population: 15 to 17 Years'] = census_age['Total Population:␣
 ↪15 to 17 Years'].astype(int)
census_age['Total Population: 18 to 24 Years'] = census_age['Total Population:␣
 ↪18 to 24 Years'].astype(int)
census_age['Total Population: 25 to 34 Years'] = census_age['Total Population:␣
 ↪25 to 34 Years'].astype(int)
census_age['Total Population: 35 to 44 Years'] = census_age['Total Population:␣
 ↪35 to 44 Years'].astype(int)
census_age['Total Population: 45 to 54 Years'] = census_age['Total Population:␣
 ↪45 to 54 Years'].astype(int)
census_age['Total Population: 55 to 64 Years'] = census_age['Total Population:␣
 ↪55 to 64 Years'].astype(int)
census_age['Total Population: 65 to 74 Years'] = census_age['Total Population:␣
 ↪65 to 74 Years'].astype(int)
census_age['Total Population: 75 to 84 Years'] = census_age['Total Population:␣
 ↪75 to 84 Years'].astype(int)
```

[81]:
```
#simplify the age categories
census_age['Under 18'] = census_age['Total Population: Under 5 Years'] +␣
 ↪census_age['Total Population: 5 to 9 Years'] + census_age['Total Population:␣
 ↪10 to 14 Years'] + census_age['Total Population: 15 to 17 Years']
census_age['18 to 34'] = census_age['Total Population: 18 to 24 Years']  +␣
 ↪census_age['Total Population: 25 to 34 Years']
census_age['35 to 64'] = census_age['Total Population: 35 to 44 Years'] +␣
 ↪census_age['Total Population: 45 to 54 Years'] + census_age['Total␣
 ↪Population: 55 to 64 Years']
census_age['Over 65'] =  census_age['Total Population: 65 to 74 Years'] +␣
 ↪census_age['Total Population: 75 to 84 Years']
```

[82]:
```
census_age.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
Int64Index: 1309 entries, 1 to 2641
Data columns (total 21 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   FIPS                                1309 non-null   object
 1   Census Tract                        1309 non-null   object
 2   Population Density (Per Sq. Mile)   1309 non-null   object
 3   Total Population:                   1309 non-null   int64
 4   Total Population: Under 5 Years     1309 non-null   int64
 5   Total Population: 5 to 9 Years      1309 non-null   int64
 6   Total Population: 10 to 14 Years    1309 non-null   int64
 7   Total Population: 15 to 17 Years    1309 non-null   int64
 8   Total Population: 18 to 24 Years    1309 non-null   int64
 9   Total Population: 25 to 34 Years    1309 non-null   int64
 10  Total Population: 35 to 44 Years    1309 non-null   int64
 11  Total Population: 45 to 54 Years    1309 non-null   int64
 12  Total Population: 55 to 64 Years    1309 non-null   int64
 13  Total Population: 65 to 74 Years    1309 non-null   int64
 14  Total Population: 75 to 84 Years    1309 non-null   int64
 15  geometry                            1309 non-null   geometry
 16  geometry_cent                       1309 non-null   geometry
 17  Under 18                            1309 non-null   int64
 18  18 to 34                            1309 non-null   int64
 19  35 to 64                            1309 non-null   int64
 20  Over 65                             1309 non-null   int64
dtypes: geometry(2), int64(16), object(3)
memory usage: 225.0+ KB
```

[83]:
```
#simplify the columns
age_col2keep = ['FIPS',
                'Census Tract',
                'Population Density (Per Sq. Mile)',
                'Total Population:',
                'Under 18',
                '18 to 34',
                '35 to 64',
                'Over 65',
                'geometry',
               'geometry_cent']
```

[84]:
```
#redefine the columns
census_age = census_age[age_col2keep]
```

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

[85]:
```
           FIPS Census Tract Population Density (Per Sq. Mile)  \
1  06037101110       101110                            9710.18
2  06037101122       101122                           3335.385
3  06037101210       101210                           25263.42
4  06037101220       101220                           13719.17
5  06037101300       101300                           3897.202

   Total Population:  Under 18  18 to 34  35 to 64  Over 65  \
1               4283       874       943      1739      649
2               3405       429       824      1472      636
3               6347      1480      1499      2744      573
4               3702       751       808      1547      533
5               3884       439       642      1863      714

                                              geometry  \
1  POLYGON ((-118.29793 34.26323, -118.29697 34.2…
2  POLYGON ((-118.27743 34.25991, -118.27747 34.2…
3  POLYGON ((-118.28594 34.25590, -118.28594 34.2…
4  POLYGON ((-118.27818 34.25577, -118.27824 34.2…
5  POLYGON ((-118.26528 34.25238, -118.26531 34.2…

                 geometry_cent
1  POINT (-118.29300 34.25948)
2  POINT (-118.29016 34.26773)
3  POINT (-118.29074 34.25298)
4  POINT (-118.28164 34.25161)
5  POINT (-118.27101 34.24878)
```

[86]:
```
#normalize the data into percentages for mapping
census_age['Percent under 18']=census_age['Under 18']/census_age['Total␣
↪Population:']*100
census_age['Percent 18 to 34']=census_age['18 to 34']/census_age['Total␣
↪Population:']*100
census_age['Percent 35 to 64']=census_age['35 to 64']/census_age['Total␣
↪Population:']*100
census_age['Percent over 65']=census_age['Over 65']/census_age['Total␣
↪Population:']*100
```

[87]:
```
census_age.head()
```

[87]:
```
           FIPS Census Tract Population Density (Per Sq. Mile)
1  06037101110       101110                            9710.18
2  06037101122       101122                           3335.385
3  06037101210       101210                           25263.42
4  06037101220       101220                           13719.17
5  06037101300       101300                           3897.202
```

```
   Total Population:  Under 18  18 to 34  35 to 64  Over 65  \
1               4283       874       943      1739      649
2               3405       429       824      1472      636
3               6347      1480      1499      2744      573
4               3702       751       808      1547      533
5               3884       439       642      1863      714

                                              geometry  \
1  POLYGON ((-118.29793 34.26323, -118.29697 34.2…
2  POLYGON ((-118.27743 34.25991, -118.27747 34.2…
3  POLYGON ((-118.28594 34.25590, -118.28594 34.2…
4  POLYGON ((-118.27818 34.25577, -118.27824 34.2…
5  POLYGON ((-118.26528 34.25238, -118.26531 34.2…

                 geometry_cent  Percent under 18  Percent 18 to 34  \
1  POINT (-118.29300 34.25948)         20.406257         22.017278
2  POINT (-118.29016 34.26773)         12.599119         24.199706
3  POINT (-118.29074 34.25298)         23.318103         23.617457
4  POINT (-118.28164 34.25161)         20.286332         21.826040
5  POINT (-118.27101 34.24878)         11.302781         16.529351

   Percent 35 to 64  Percent over 65
1         40.602382        15.152930
2         43.230543        18.678414
3         43.233023         9.027887
4         41.788223        14.397623
5         47.966014        18.383110
```

[88]:
```
age_col2keep2 = ['FIPS',
                 'Census Tract',
                 'Population Density (Per Sq. Mile)',
                 'Percent under 18',
                 'Percent 18 to 34',
                 'Percent 35 to 64',
                 'Percent over 65',
                 'geometry',
                 'geometry_cent']
```

[89]:
```
census_age = census_age[age_col2keep2]
```

[90]:
```
census_age.head()
```

[90]:
```
          FIPS Census Tract Population Density (Per Sq. Mile)  \
1  06037101110       101110                           9710.18
2  06037101122       101122                          3335.385
3  06037101210       101210                          25263.42
4  06037101220       101220                          13719.17
```

```
5  06037101300       101300                          3897.202

   Percent under 18  Percent 18 to 34  Percent 35 to 64  Percent over 65  \
1         20.406257         22.017278         40.602382        15.152930
2         12.599119         24.199706         43.230543        18.678414
3         23.318103         23.617457         43.233023         9.027887
4         20.286332         21.826040         41.788223        14.397623
5         11.302781         16.529351         47.966014        18.383110

                                             geometry  \
1  POLYGON ((-118.29793 34.26323, -118.29697 34.2…
2  POLYGON ((-118.27743 34.25991, -118.27747 34.2…
3  POLYGON ((-118.28594 34.25590, -118.28594 34.2…
4  POLYGON ((-118.27818 34.25577, -118.27824 34.2…
5  POLYGON ((-118.26528 34.25238, -118.26531 34.2…

                 geometry_cent
1  POINT (-118.29300 34.25948)
2  POINT (-118.29016 34.26773)
3  POINT (-118.29074 34.25298)
4  POINT (-118.28164 34.25161)
5  POINT (-118.27101 34.24878)
```

[91]:
```
census_age.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
Int64Index: 1309 entries, 1 to 2641
Data columns (total 9 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   FIPS                              1309 non-null   object
 1   Census Tract                      1309 non-null   object
 2   Population Density (Per Sq. Mile)  1309 non-null   object
 3   Percent under 18                  1303 non-null   float64
 4   Percent 18 to 34                  1303 non-null   float64
 5   Percent 35 to 64                  1303 non-null   float64
 6   Percent over 65                   1303 non-null   float64
 7   geometry                          1309 non-null   geometry
 8   geometry_cent                     1309 non-null   geometry
dtypes: float64(4), geometry(2), object(3)
memory usage: 102.3+ KB
```

[92]:
```
#create a density data frame
density = ['FIPS',
           'Census Tract',
           'Population Density (Per Sq. Mile)',
           'geometry']
```

[93]:
```
census_dens = census[density]
```

[94]:
```
census_dens.head()
```

[94]:
```
           FIPS Census Tract Population Density (Per Sq. Mile)  \
1  06037101110       101110                            9710.18
2  06037101122       101122                           3335.385
3  06037101210       101210                           25263.42
4  06037101220       101220                           13719.17
5  06037101300       101300                           3897.202

                                                 geometry
1  POLYGON ((-118.29793 34.26323, -118.29697 34.2…
2  POLYGON ((-118.27743 34.25991, -118.27747 34.2…
3  POLYGON ((-118.28594 34.25590, -118.28594 34.2…
4  POLYGON ((-118.27818 34.25577, -118.27824 34.2…
5  POLYGON ((-118.26528 34.25238, -118.26531 34.2…
```

[95]:
```
census_dens.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
Int64Index: 1309 entries, 1 to 2641
Data columns (total 4 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   FIPS                              1309 non-null   object
 1   Census Tract                      1309 non-null   object
 2   Population Density (Per Sq. Mile)  1309 non-null   object
 3   geometry                          1309 non-null   geometry
dtypes: geometry(1), object(3)
memory usage: 51.1+ KB
```

[96]:
```
census_dens['Population Density (Per Sq. Mile)'] = census_dens['Population␣
↪Density (Per Sq. Mile)'].astype(float)
```

```
/opt/conda/lib/python3.9/site-packages/geopandas/geodataframe.py:1322:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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

[97]:
```
#create an income variable from census data
income = ['FIPS',
          'Census Tract',
```

```
        'Population Density (Per Sq. Mile)',
        'Households:',
        'Households: Less than $10,000',
        'Households: $10,000 to $14,999',
        'Households: $15,000 to $19,999',
        'Households: $20,000 to $24,999',
        'Households: $25,000 to $29,999',
        'Households: $30,000 to $34,999',
        'Households: $35,000 to $39,999',
        'Households: $40,000 to $44,999',
        'Households: $45,000 to $49,999',
        'Households: $50,000 to $59,999',
        'Households: $60,000 to $74,999',
        'Households: $75,000 to $99,999',
        'Households: $100,000 to $124,999',
        'Households: $125,000 to $149,999',
        'Households: $150,000 to $199,999',
        'Households: $200,000 or More',
        'geometry',
        'geometry_cent']
```

[98]:
```
census_inc = census[income]
```

[99]:
```
census_inc.head()
```

[99]:
```
          FIPS Census Tract Population Density (Per Sq. Mile) Households:  \
1  06037101110       101110                           9710.18        1575
2  06037101122       101122                          3335.385        1312
3  06037101210       101210                          25263.42        2297
4  06037101220       101220                          13719.17        1357
5  06037101300       101300                          3897.202        1445

  Households: Less than $10,000 Households: $10,000 to $14,999  \
1                            86                              91
2                            66                              21
3                           271                             135
4                            85                             128
5                            78                              18

  Households: $15,000 to $19,999 Households: $20,000 to $24,999  \
1                            101                              44
2                              0                              47
3                            278                              39
4                            124                              57
5                             59                              33

  Households: $25,000 to $29,999 Households: $30,000 to $34,999  …  \
```

```
1                              55                                78  …
2                              36                                25  …
3                             108                               169  …
4                              35                                94  …
5                              68                                33  …

  Households: $45,000 to $49,999 Households: $50,000 to $59,999  \
1                              99                                99
2                              52                                34
3                             178                               163
4                              83                                84
5                              32                                86

  Households: $60,000 to $74,999 Households: $75,000 to $99,999  \
1                             107                               274
2                             166                               298
3                             165                               261
4                              63                               122
5                             147                               195

  Households: $100,000 to $124,999 Households: $125,000 to $149,999  \
1                              151                                75
2                              119                               146
3                              148                               106
4                              114                                68
5                              137                               122

  Households: $150,000 to $199,999 Households: $200,000 or More  \
1                              125                                77
2                              115                               180
3                               53                                58
4                               69                                41
5                              159                               178

                                            geometry  \
1  POLYGON ((-118.29793 34.26323, -118.29697 34.2…
2  POLYGON ((-118.27743 34.25991, -118.27747 34.2…
3  POLYGON ((-118.28594 34.25590, -118.28594 34.2…
4  POLYGON ((-118.27818 34.25577, -118.27824 34.2…
5  POLYGON ((-118.26528 34.25238, -118.26531 34.2…

                 geometry_cent
1  POINT (-118.29300 34.25948)
2  POINT (-118.29016 34.26773)
3  POINT (-118.29074 34.25298)
4  POINT (-118.28164 34.25161)
5  POINT (-118.27101 34.24878)
```

[5 rows x 22 columns]

[100]:
```
#convert to integers
census_inc['Households:'] = census_inc['Households:'].astype(int)
census_inc['Households: Less than $10,000'] = census_inc['Households: Less than␣
↪$10,000'].astype(int)
census_inc['Households: $10,000 to $14,999'] = census_inc['Households: $10,000␣
↪to $14,999'].astype(int)
census_inc['Households: $15,000 to $19,999'] = census_inc['Households: $15,000␣
↪to $19,999'].astype(int)
census_inc['Households: $20,000 to $24,999'] = census_inc['Households: $20,000␣
↪to $24,999'].astype(int)
census_inc['Households: $25,000 to $29,999'] = census_inc['Households: $25,000␣
↪to $29,999'].astype(int)
census_inc['Households: $30,000 to $34,999'] = census_inc['Households: $30,000␣
↪to $34,999'].astype(int)
census_inc['Households: $35,000 to $39,999'] = census_inc['Households: $35,000␣
↪to $39,999'].astype(int)
census_inc['Households: $40,000 to $44,999'] = census_inc['Households: $40,000␣
↪to $44,999'].astype(int)
census_inc['Households: $45,000 to $49,999'] = census_inc['Households: $45,000␣
↪to $49,999'].astype(int)
census_inc['Households: $50,000 to $59,999'] = census_inc['Households: $50,000␣
↪to $59,999'].astype(int)
census_inc['Households: $60,000 to $74,999'] = census_inc['Households: $60,000␣
↪to $74,999'].astype(int)
census_inc['Households: $75,000 to $99,999'] = census_inc['Households: $75,000␣
↪to $99,999'].astype(int)
census_inc['Households: $100,000 to $124,999'] = census_inc['Households:␣
↪$100,000 to $124,999'].astype(int)
census_inc['Households: $125,000 to $149,999'] = census_inc['Households:␣
↪$125,000 to $149,999'].astype(int)
census_inc['Households: $150,000 to $199,999'] = census_inc['Households:␣
↪$150,000 to $199,999'].astype(int)
census_inc['Households: $200,000 or More'] = census_inc['Households: $200,000␣
↪or More'].astype(int)
```

```
/opt/conda/lib/python3.9/site-packages/geopandas/geodataframe.py:1322:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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

[101]:
```
#simplify the income brackets
census_inc['Less than $30k'] = census_inc['Households: Less than $10,000'] +␣
 ↪census_inc['Households: $10,000 to $14,999'] + census_inc['Households:␣
 ↪$15,000 to $19,999'] + census_inc['Households: $20,000 to $24,999'] +␣
 ↪census_inc['Households: $25,000 to $29,999']
census_inc['$30k to $75k'] = census_inc['Households: $30,000 to $34,999'] +␣
 ↪census_inc['Households: $35,000 to $39,999'] + census_inc['Households:␣
 ↪$40,000 to $44,999'] + census_inc['Households: $45,000 to $49,999'] +␣
 ↪census_inc['Households: $50,000 to $59,999'] + census_inc['Households:␣
 ↪$60,000 to $74,999']
census_inc['$75k to $150k'] = census_inc['Households: $75,000 to $99,999'] +␣
 ↪census_inc['Households: $100,000 to $124,999'] + census_inc['Households:␣
 ↪$125,000 to $149,999']
census_inc['Over $150k'] =  census_inc['Households: $150,000 to $199,999'] +␣
 ↪census_inc['Households: $200,000 or More']
```

[102]:
```
census_inc.head()
```

[102]:

	FIPS	Census Tract	Population Density (Per Sq. Mile)	Households:	\
1	06037101110	101110	9710.18	1575	
2	06037101122	101122	3335.385	1312	
3	06037101210	101210	25263.42	2297	
4	06037101220	101220	13719.17	1357	
5	06037101300	101300	3897.202	1445	

	Households: Less than $10,000	Households: $10,000 to $14,999	\
1	86	91	
2	66	21	
3	271	135	
4	85	128	
5	78	18	

	Households: $15,000 to $19,999	Households: $20,000 to $24,999	\
1	101	44	
2	0	47	
3	278	39	
4	124	57	
5	59	33	

	Households: $25,000 to $29,999	Households: $30,000 to $34,999	…	\
1	55	78	…	
2	36	25	…	
3	108	169	…	
4	35	94	…	
5	68	33	…	

	Households: $100,000 to $124,999	Households: $125,000 to $149,999	\

```
1                                151                                 75
2                                119                                146
3                                148                                106
4                                114                                 68
5                                137                                122

   Households: $150,000 to $199,999  Households: $200,000 or More  \
1                               125                             77
2                               115                            180
3                                53                             58
4                                69                             41
5                               159                            178

                                          geometry  \
1  POLYGON ((-118.29793 34.26323, -118.29697 34.2…
2  POLYGON ((-118.27743 34.25991, -118.27747 34.2…
3  POLYGON ((-118.28594 34.25590, -118.28594 34.2…
4  POLYGON ((-118.27818 34.25577, -118.27824 34.2…
5  POLYGON ((-118.26528 34.25238, -118.26531 34.2…

                 geometry_cent  Less than $30k  $30k to $75k  $75k to $150k  \
1  POINT (-118.29300 34.25948)             377           496            500
2  POINT (-118.29016 34.26773)             170           284            563
3  POINT (-118.29074 34.25298)             831           840            515
4  POINT (-118.28164 34.25161)             429           514            304
5  POINT (-118.27101 34.24878)             256           398            454

   Over $150k
1         202
2         295
3         111
4         110
5         337

[5 rows x 26 columns]
```

[103]:
```
income_col2keep = ['FIPS',
                   'Census Tract',
                   'Population Density (Per Sq. Mile)',
                   'Households:',
                   'Less than $30k',
                   '$30k to $75k',
                   '$75k to $150k',
                   'Over $150k',
                   'geometry',
                  'geometry_cent']
```

[104]:
```
census_inc = census_inc[income_col2keep]
```

[105]:
```
census_inc.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
Int64Index: 1309 entries, 1 to 2641
Data columns (total 10 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   FIPS                              1309 non-null   object
 1   Census Tract                      1309 non-null   object
 2   Population Density (Per Sq. Mile)  1309 non-null   object
 3   Households:                       1309 non-null   int64
 4   Less than $30k                    1309 non-null   int64
 5   $30k to $75k                      1309 non-null   int64
 6   $75k to $150k                     1309 non-null   int64
 7   Over $150k                        1309 non-null   int64
 8   geometry                          1309 non-null   geometry
 9   geometry_cent                     1309 non-null   geometry
dtypes: geometry(2), int64(5), object(3)
memory usage: 112.5+ KB
```

[106]:
```
#normaliza the data
census_inc['Percent less than $30k']=census_inc['Less than $30k']/
 ↪census_inc['Households:']*100
census_inc['Percent $30k to $75k']=census_inc['$30k to $75k']/
 ↪census_inc['Households:']*100
census_inc['Percent $75k to $150k']=census_inc['$75k to $150k']/
 ↪census_inc['Households:']*100
census_inc['Percent over $150k']=census_inc['Over $150k']/
 ↪census_inc['Households:']*100
```

[107]:
```
census_inc.head()
```

[107]:

	FIPS	Census Tract	Population Density (Per Sq. Mile)	Households:	Less than $30k	$30k to $75k	$75k to $150k	Over $150k
1	06037101110	101110	9710.18	1575	377	496	500	202
2	06037101122	101122	3335.385	1312	170	284	563	295
3	06037101210	101210	25263.42	2297	831	840	515	111
4	06037101220	101220	13719.17	1357	429	514	304	110
5	06037101300	101300	3897.202	1445	256	398	454	337

```
                                                  geometry  \
1  POLYGON ((-118.29793 34.26323, -118.29697 34.2…
2  POLYGON ((-118.27743 34.25991, -118.27747 34.2…
3  POLYGON ((-118.28594 34.25590, -118.28594 34.2…
4  POLYGON ((-118.27818 34.25577, -118.27824 34.2…
5  POLYGON ((-118.26528 34.25238, -118.26531 34.2…

                 geometry_cent  Percent less than $30k  Percent $30k to $75k  \
1  POINT (-118.29300 34.25948)                23.936508             31.492063
2  POINT (-118.29016 34.26773)                12.957317             21.646341
3  POINT (-118.29074 34.25298)                36.177623             36.569438
4  POINT (-118.28164 34.25161)                31.613854             37.877671
5  POINT (-118.27101 34.24878)                17.716263             27.543253

   Percent $75k to $150k  Percent over $150k
1              31.746032           12.825397
2              42.911585           22.484756
3              22.420549            4.832390
4              22.402358            8.106116
5              31.418685           23.321799
```

[108]:
```
income_col2keep2 = ['FIPS',
                    'Census Tract',
                    'Population Density (Per Sq. Mile)',
                    'Percent less than $30k',
                    'Percent $30k to $75k',
                    'Percent $75k to $150k',
                    'Percent over $150k',
                    'geometry',
                   'geometry_cent']
```

[109]:
```
census_inc = census_inc[income_col2keep2]
```

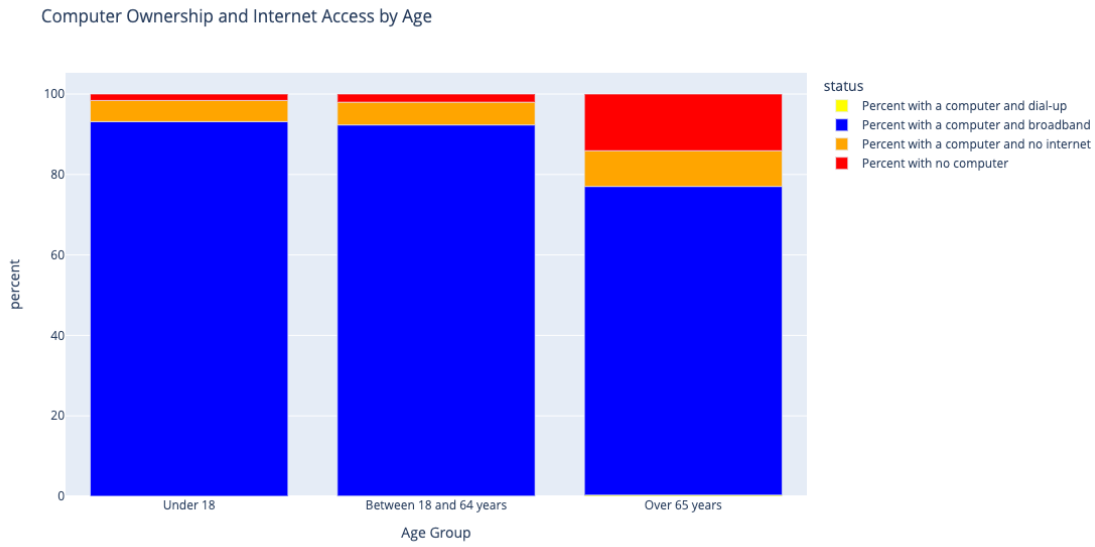
1.3 Visualizing the Data

Let's begin to visualize the data! This is hgher level data for the city so I think stacked bar graphs will be best to compare computer ownership and type of internet with the selected variable of age, educational attainment, and race.

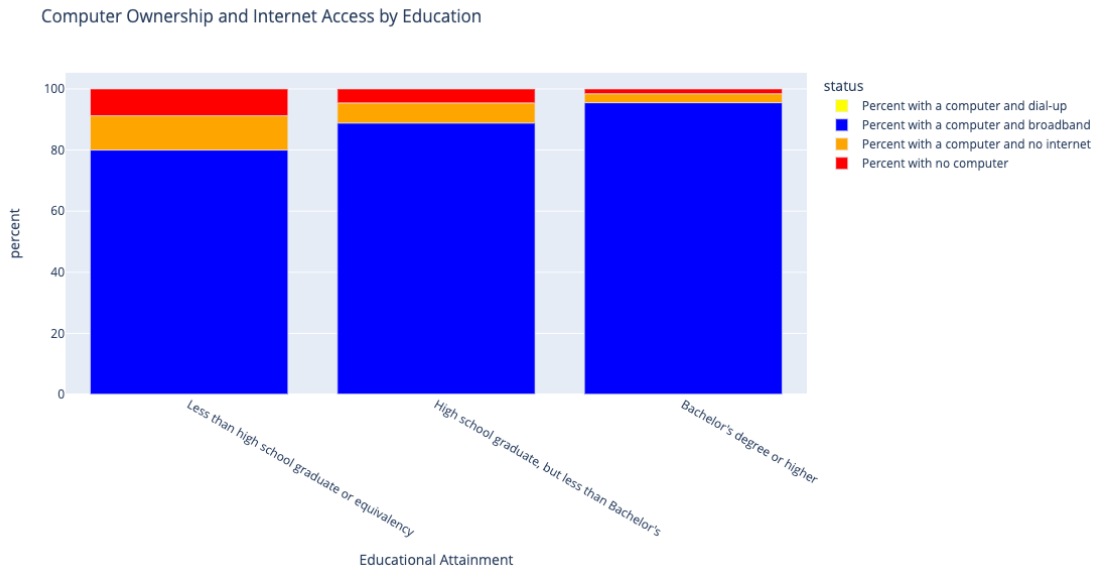
1.3.1 Creating Graphs and Charts

[110]:
```
fig = px.bar(age,
             x = 'Age Group',
             y = 'percent',
             width=800,
             height=600,
```

```
            color = 'status',
            color_discrete_sequence = ['yellow','blue','orange','red'],
            title = 'Computer Ownership and Internet Access by Age',
)
fig.show()
```



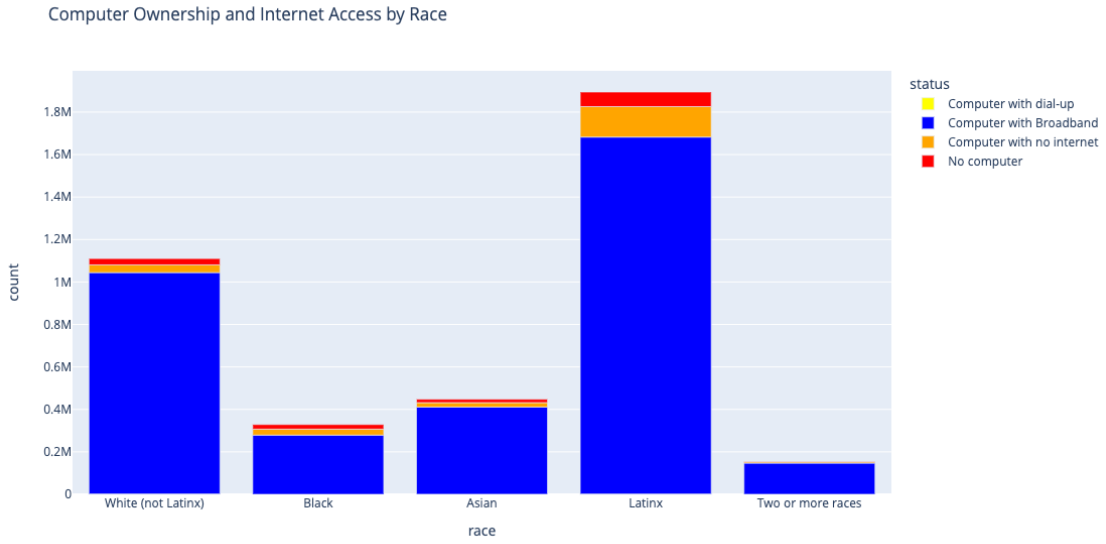
[111]:
```
fig = px.bar(educ,
            x = 'Educational Attainment',
            y = 'percent',
            width=800,
            height=600,
            color = 'status',
            color_discrete_sequence = ['yellow','blue','orange','red'],
            title = 'Computer Ownership and Internet Access by Education',
)
fig.show()
```



[112]:
```
#Currently I do not intended to normalize the race variable to demonstrate the␣
 ↪magnitude of digital exclusion between groups. To do this, I need to change␣
 ↪the data type for counts:
race['count']=race['count'].astype(int)
race.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   race    20 non-null     object
 1   status  20 non-null     object
 2   count   20 non-null     int64
dtypes: int64(1), object(2)
memory usage: 608.0+ bytes
```

[113]:
```
fig = px.bar(race,
             x = 'race',
             y = 'count',
             width = 800,
             height = 600,
             color = 'status',
             color_discrete_sequence = ['yellow','blue','orange','red'],
             title = 'Computer Ownership and Internet Access by Race',
)
fig.show()
```



1.3.2 Mapping Internet Access

[114]:
```
#Let's map some data!
#hh.plot(figsize=(20,10),
#        column = 'Percent with no internet access',
#        legend = True,
#        scheme='naturalbreaks',
#        cmap = 'YlOrRd',
#        k=7)
```

[115]:
```
#census_edu1.plot(figsize=(20,10),
#        column = '% Population 25 Years and Over: Less than High School',
#        legend = True,
#        scheme='naturalbreaks',
#        cmap = 'YlOrRd')
```

[116]:
```
#census_edu2.plot(figsize=(20,10),
#        column = '% Population 3 Years and Over: Not Enrolled in School',
#        legend = True,
#        scheme='naturalbreaks',
#        cmap = 'YlOrRd')
#this looks alarming...like no children are in school, I need to understand␣
 ↪this better before moving forward
```

[117]:
```
#census_edu3.plot(figsize=(20,10),
#        column = '% Civilian Population 16 to 19 Years: Not High School␣
 ↪Graduate, Not Enrolled (Dropped Out)',
```

```
#         legend = True,
#         scheme='naturalbreaks',
#         cmap = 'YlOrRd')
# The value of this is questionable, will hold off using in further analysis
```

[118]:
```
#census_fb.plot(figsize=(20,10),
#         column = '% Total Population: Foreign Born',
#         legend = True,
#         scheme='naturalbreaks',
#         cmap = 'YlOrRd')
```

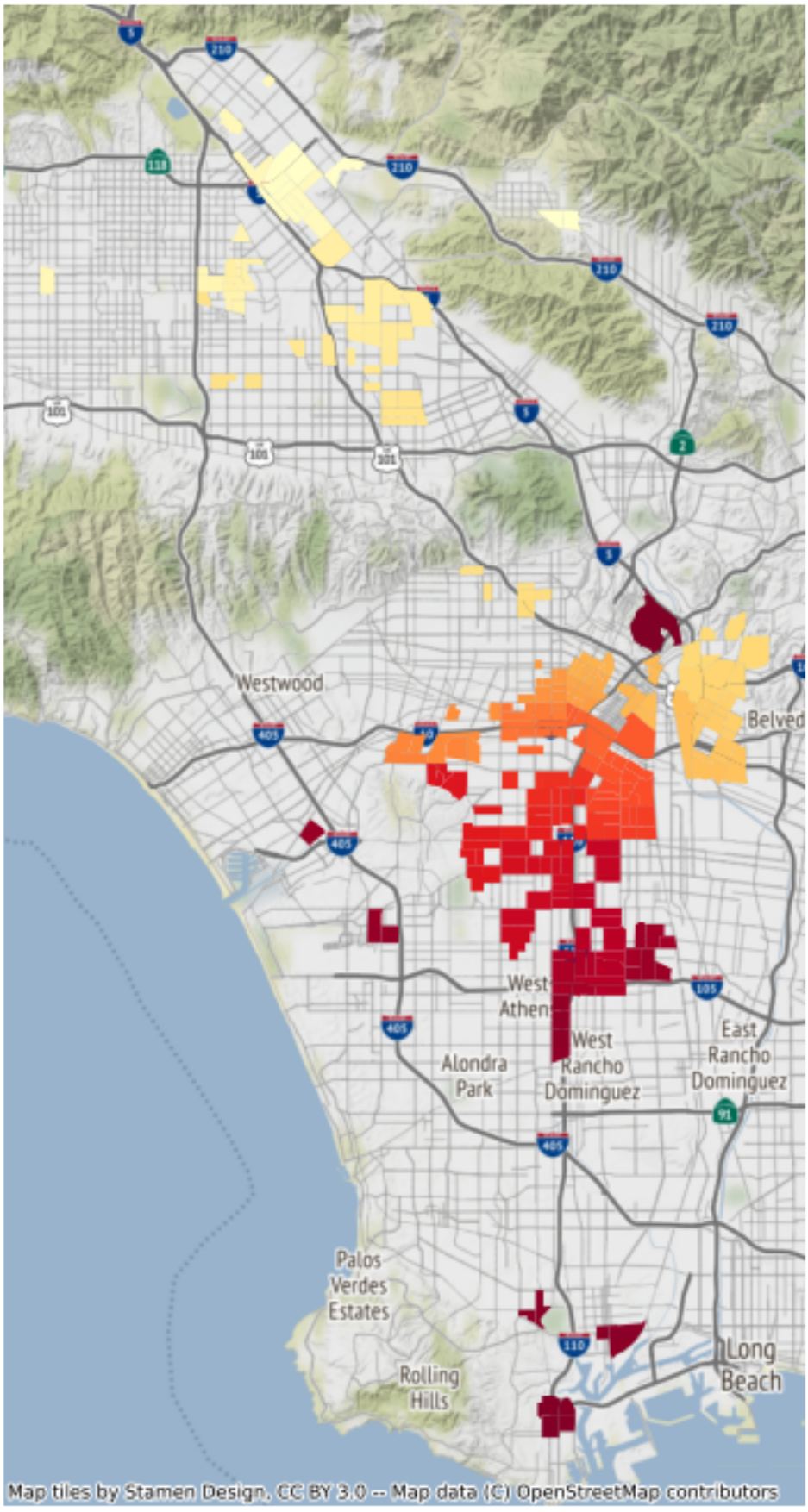
[119]:
```
#census_age.plot(figsize=(20,10),
#         column = 'Percent over 65',
#         legend = True,
#         scheme='naturalbreaks',
#         cmap = 'YlOrRd')
```

[120]:
```
#census_dens.plot(figsize=(20,10),
#         column = 'Population Density (Per Sq. Mile)',
#         legend = True,
#         scheme='naturalbreaks',
#         cmap = 'YlOrRd')
```

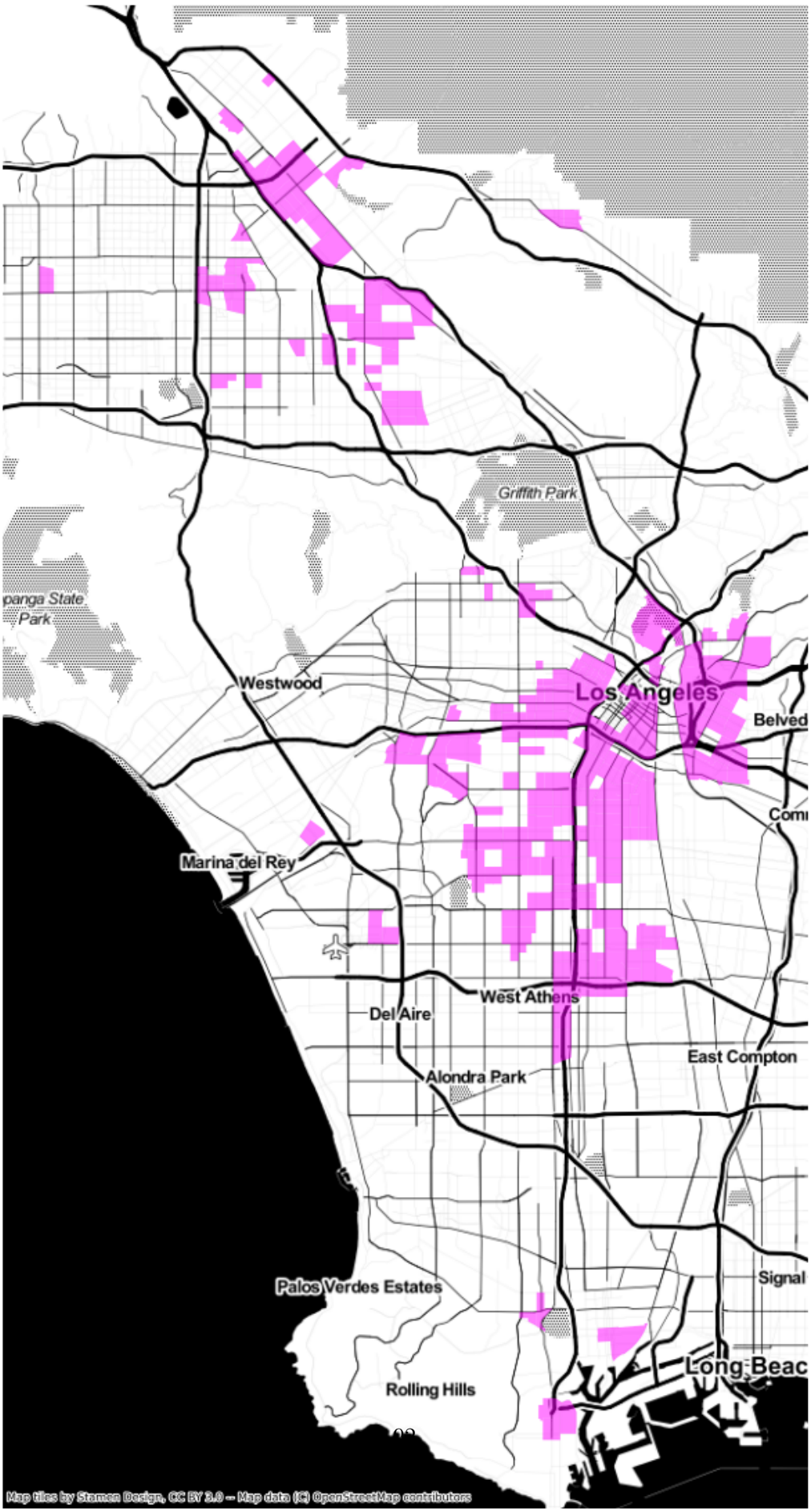
[121]:
```
#census_inc.plot(figsize=(20,10),
#         column = 'Percent less than $30k',
#         legend = True,
#         scheme='naturalbreaks',
#         cmap = 'YlOrRd')
```

[122]:
```
#let's kick it up a notch
hh_web_mercator = hh.to_crs(epsg=3857)
```

[123]:
```
fig, ax = plt.subplots(figsize = (20, 12))
hh_web_mercator[hh_web_mercator['Percent with no internet access']>20].
 ↪plot(ax=ax, scheme = 'naturalbreaks', cmap = 'YlOrRd', k=7)
ax.axis('off')
ctx.add_basemap(ax)
```


Westwood
Belved
West
Athens
Alondra
Park
West
Rancho
Dominguez
East
Rancho
Dominguez
Palos
Verdes
Estates
Rolling
Hills
Long
Beach
Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors

[124]:
```
fig, ax = plt.subplots(figsize = (20, 20))
hh_web_mercator[hh_web_mercator['Percent with no internet access']>20].
 ↪plot(ax=ax, color = '#ff00ff', alpha=0.5)
ax.axis('off')
ctx.add_basemap(ax, source=ctx.providers.Stamen.Toner)
```


Griffith Park
Westwood
Los Angeles
Belved
Marina del Rey
West Athens
Del Aire
East Compton
Alondra Park
Palos Verdes Estates
Signal
Long Beac
Rolling Hills
Map tiles by Stamen Design, CC BY 3.0 — Map data (C) OpenStreetMap contributors

1.4 Mapping locations (YK)

- convert lat/lon's to floats
- convert lib to geodataframe
- reproject to web mercator

[125]:
```
lib['lat'] = lib['lat'].astype(float)
lib['lon'] = lib['lon'].astype(float)
```

[126]:
```
lib_gdf = gpd.GeoDataFrame(
    lib, geometry=gpd.points_from_xy(lib.lon, lib.lat),crs="EPSG:4326")
```

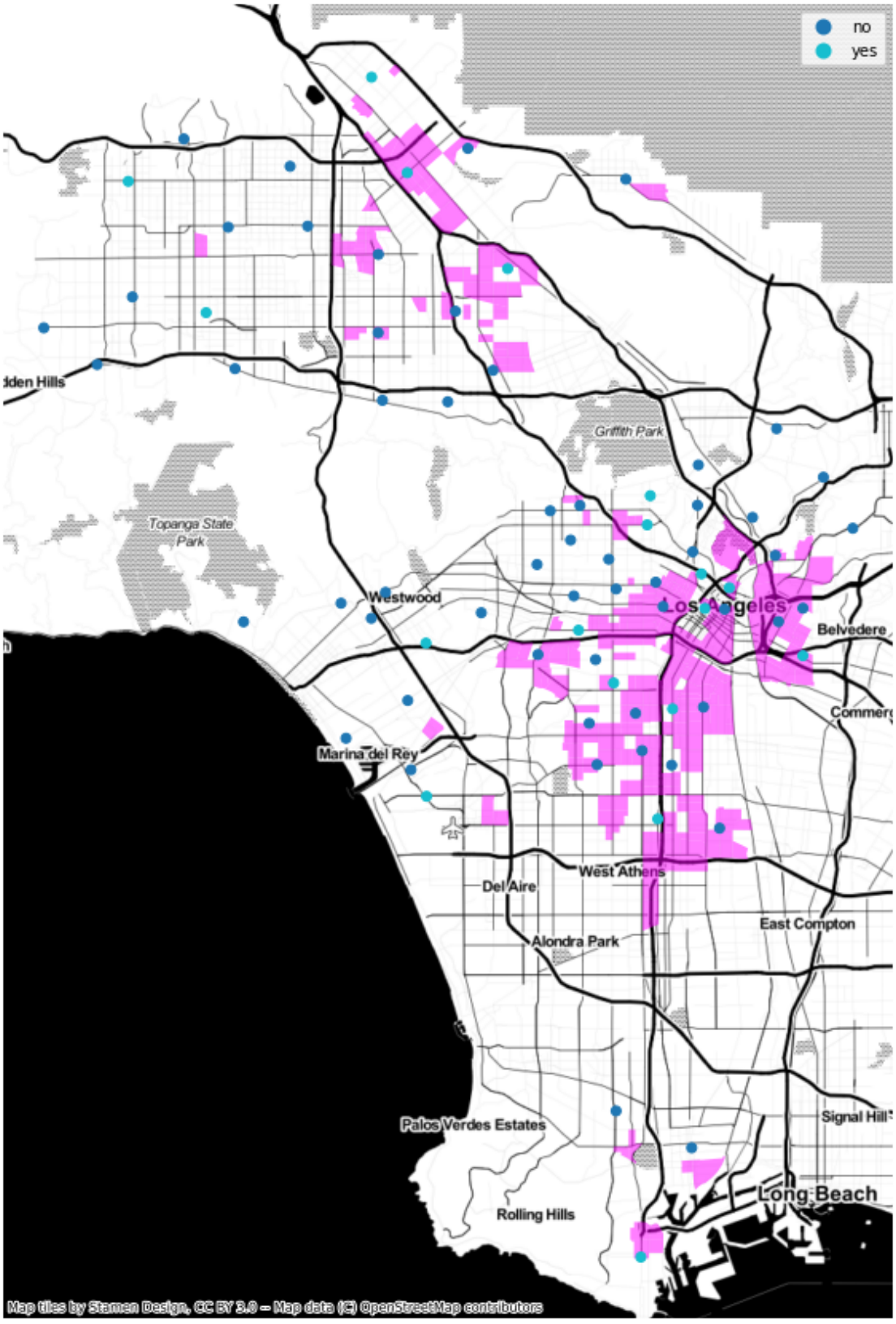
[127]:
```
lib_gdf = lib_gdf.to_crs(epsg=3857)
```

[128]:
```
list(census_fb)
```

[128]:
```
['FIPS',
 'Census Tract',
 'Population Density (Per Sq. Mile)',
 '% Total Population: Foreign Born',
 'geometry',
 'geometry_cent']
```

[129]:
```
census_fb = census_fb.to_crs(epsg=3857)
```

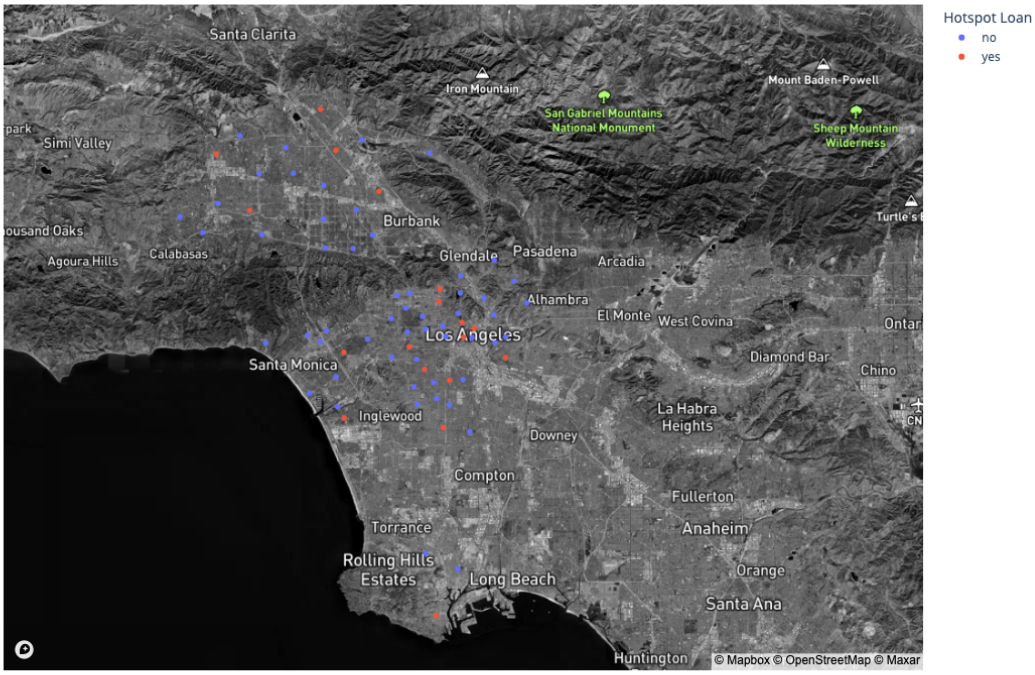
[130]:
```
fig, ax = plt.subplots(figsize = (10, 20))
hh_web_mercator[hh_web_mercator['Percent with no internet access']>20].
 ↪plot(ax=ax, color = '#ff00ff', alpha=0.5)
lib_gdf.plot(ax=ax,column='Hotspot Loan', legend= True)
ax.axis('off')
ctx.add_basemap(ax, source=ctx.providers.Stamen.Toner)

#Thanks Yoh!
```


no
yes
Griffith Park
Topanga State Park
Westwood
Los Angeles
Belvedere
Marina del Rey
Del Aire
West Athens
East Compton
Alondra Park
Palos Verdes Estates
Rolling Hills
Long Beach
Map tiles by Stamen Design, CC BY 3.0 — Map data (C) OpenStreetMap contributors

[131]:
```
#Next, I will attempt to make an interactive version of the map above.

#mapbox access token
token = 'pk.
 ↪eyJ1IjoidWNsYS1jcmlzdGVtYyIsImEiOiJja2w0czA1bXYxeWszMnFxbnc0djltd295In0.
 ↪ocHbUsFaHy65rMNwCGi1lA'
px.set_mapbox_access_token(token)
```

[132]:
```
mapboxstyle = 'mapbox://styles/ucla-cristemc/ckvvjyivv419i15qyzcl8k0w7'
```

[133]:
```
#recreate map above with library locations
libloc = px.scatter_mapbox(lib_gdf, lat="lat", lon='lon', color = "Hotspot␣
 ↪Loan",
                    mapbox_style=mapboxstyle,
                     width = 1000,
                     height = 800,
                     zoom=9,
                     center = {"lat": 34.0504, "lon": -118.2555},
                     hover_name='Branch Name',
                     hover_data=['Address',
                                 'CityZip',
                                 'Email',
                                 'Phone Number'])
libloc
```



[134]:
```
# save as html file
libloc.write_html("libloc.html")
```

[135]:
```
hh_web_mercator2 = hh_web_mercator.to_crs('epsg:4326')
```

[136]:
```
hh_col1 = hh_web_mercator2[('Percent with no internet access')]>20
```

[137]:
```
#new variable containing only tracts with 20% or higher
hh_grt20 = hh_web_mercator2.sort_values(by='Percent with no internet␣
↪access',ascending = False)
```

[138]:
```
hh_grt20[['geoid','Percent with no internet access']].head(10)
```

[138]:

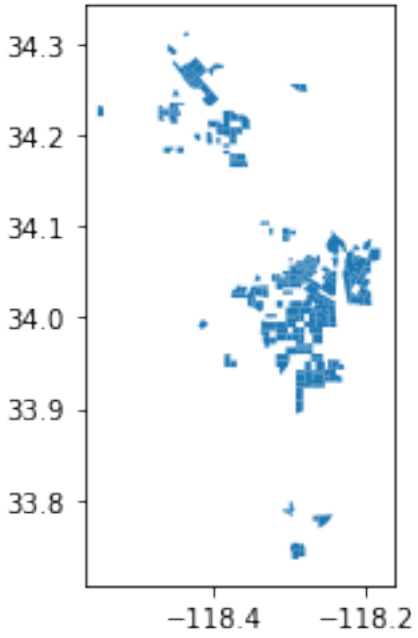
	geoid	Percent with no internet access
536	14000US06037206300	62.515413
561	14000US06037208903	51.718213
569	14000US06037209402	49.105691
534	14000US06037206050	48.322981
562	14000US06037208904	44.771495
696	14000US06037224320	43.022222
528	14000US06037205110	42.357060
535	14000US06037206200	41.462401

```
544   14000US06037207502                    39.890710
558   14000US06037208801                    39.223153
```

[139]:
```
#remove tracts with less than 20%
hh_grt20 = hh_grt20[hh_grt20['Percent with no internet access']>=20]
```

[140]:
```
hh_grt20.plot()
```

[140]:
```
<AxesSubplot:>
```

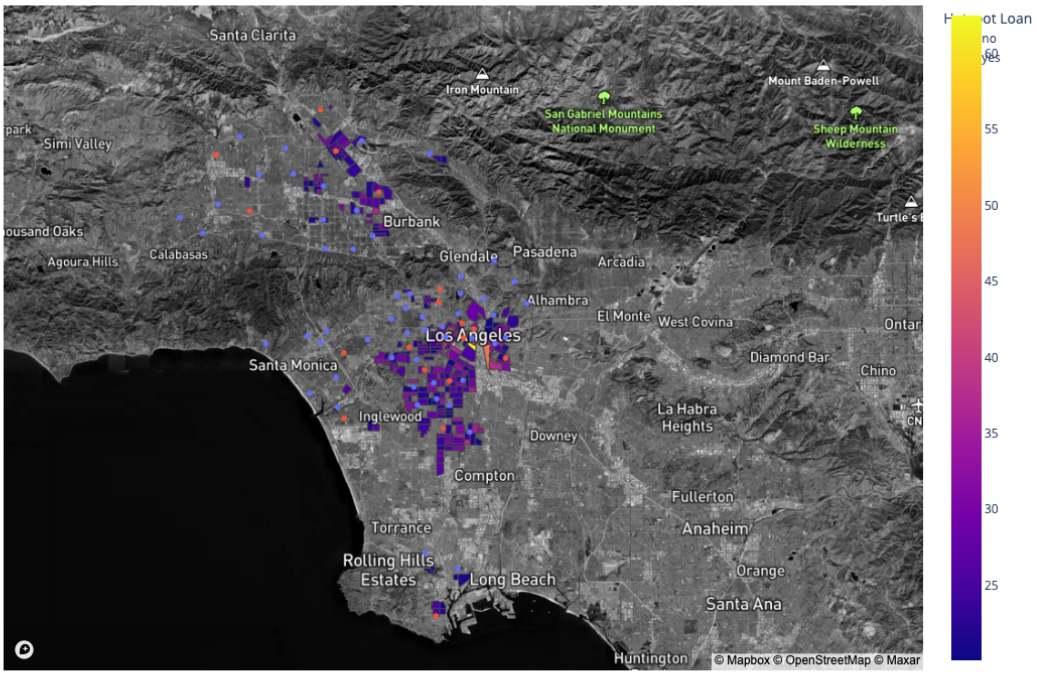


[141]:
```
hh_grt20.shape
```

[141]:
```
(265, 7)
```

[142]:
```
#Build off this map before continuing with other census data. I will use ploty␣
↪& mapbox for a more rich experience
libhh1 = px.scatter_mapbox(lib_gdf, lat="lat", lon='lon', color = "Hotspot␣
↪Loan",
                        mapbox_style=mapboxstyle,
                        width = 1000,
                        height = 800,
                        zoom=9,
                        center = {"lat": 34.0504, "lon": -118.2555},
                        hover_name='Branch Name',
                        hover_data=['Address',
```

```
                                'CityZip',
                                'Email',
                                'Phone Number'],)
                    #title='Percent of Adult Population With Less Than A High␣
 ↪School Degree')

libhh2 = px.choropleth_mapbox(hh_grt20,
                    geojson=hh_grt20.geometry,
                    locations= hh_grt20.index,
                    color = hh_grt20['Percent with no internet access'],
                    #mapbox_style=mapboxstyle,
                    width = 1000,
                    height = 800,
                    zoom=9,
                    center = {"lat": 34.0504, "lon": -118.2555},
                    #color_continuous_scale=["yellow"],
                    opacity=1)


libhh1.add_trace(
    libhh2.data[0]
)
libhh1.show()
```



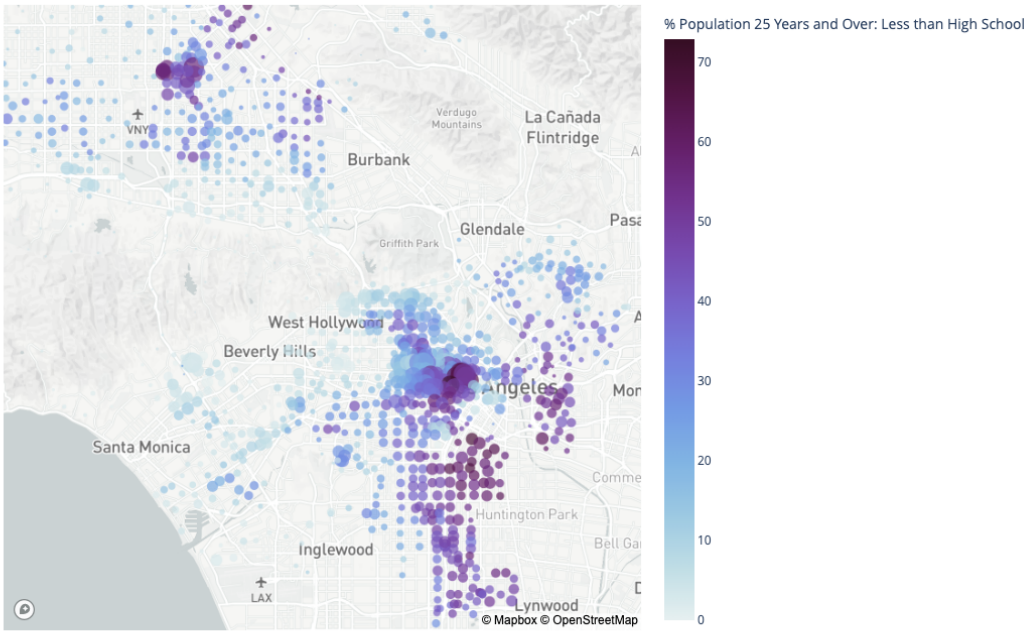
[143]:
```
# save as html file
libhh1.write_html("libhh.html")
```

[144]:
```
census_edu1['Population Density (Per Sq. Mile)'] = census_edu1['Population␣
↪Density (Per Sq. Mile)'].astype(float)
census_edu1['% Population 25 Years and Over: Less than High School'] =␣
↪census_edu1['% Population 25 Years and Over: Less than High School'].
↪astype(float)
census_edu1['Total Population'] = census_edu1['Total Population'].astype(float)
```

```
/opt/conda/lib/python3.9/site-packages/geopandas/geodataframe.py:1322:
SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[145]:
```
fig = px.scatter_mapbox(census_edu1,
                        lat=census_edu1.geometry_cent.y,
                        lon=census_edu1.geometry_cent.x,
                        color="% Population 25 Years and Over: Less than High␣
↪School",
                        size='Population Density (Per Sq. Mile)',
                        width = 1000,
                        height = 800,
                        size_max=15,
                        zoom=10,
                        title='Percent of Adult Population With Less Than A␣
↪High School Degree',
                        color_continuous_scale=px.colors.sequential.dense)
fig.show()
```

Percent of Adult Population With Less Than A High School Degree



[146]:
```
edu1_gdf = gpd.GeoDataFrame(
    census_edu1, geometry=gpd.points_from_xy(census_edu1.geometry_cent.x,␣
↪census_edu1.geometry_cent.y),crs="EPSG:4326")
```

[147]:
```
census_edu1 = census_edu1.to_crs(epsg=3857)
```

[148]:
```
#Currently in markdown to keep the file size smaller and loads more quickly
#fig = px.scatter_mapbox(lib_gdf, lat="lat", lon='lon', color = "Hotspot Loan",
#                     # mapbox_style=mapboxstyle,
#                      width = 1000,
#                      height = 800,
#                      zoom=9,
#                      center = {"lat": 34.0504, "lon": -118.2555},
#                      hover_name='Branch Name',
#                      hover_data=['Address',
#                                  'CityZip',
#                                  'Email',
#                                  'Phone Number'],
#                                 #'lat':False, Want to remove lat & lon␣
↪because it is irrelevant info for most people
#                                 #'lon':False],
```

```
#                     #color_continuous_scale=deep,
#                     title='Percent of Households Lacking Braodband Access')
#
#fig_px = px.choropleth_mapbox(hh,
#                               geojson=hh.geometry,
#                               locations= hh.index,
#                               color = hh['Percent with no internet access'],
#                               #mapbox_style=mapboxstyle,
#                               width = 1000,
#                               height = 800,
#                               zoom=9,
#                               center = {"lat": 34.0504, "lon": -118.2555},
#                               #color_continuous_scale=deep,
#                               opacity=0.7)
#fig.add_trace(
#    fig_px.data[0]
#)
#fig.show()
```

[149]:
```
hh['geometry_cent'] = hh['geometry'].centroid
```

```
/tmp/ipykernel_70/4084327279.py:1: UserWarning:

Geometry is in a geographic CRS. Results from 'centroid' are likely incorrect.
Use 'GeoSeries.to_crs()' to re-project geometries to a projected CRS before this
operation.
```

[150]:
```
hh_gdf = gpd.GeoDataFrame(
    hh, geometry=gpd.points_from_xy(hh.geometry_cent.x, hh.geometry_cent.
 ↪y),crs="EPSG:4326")
```

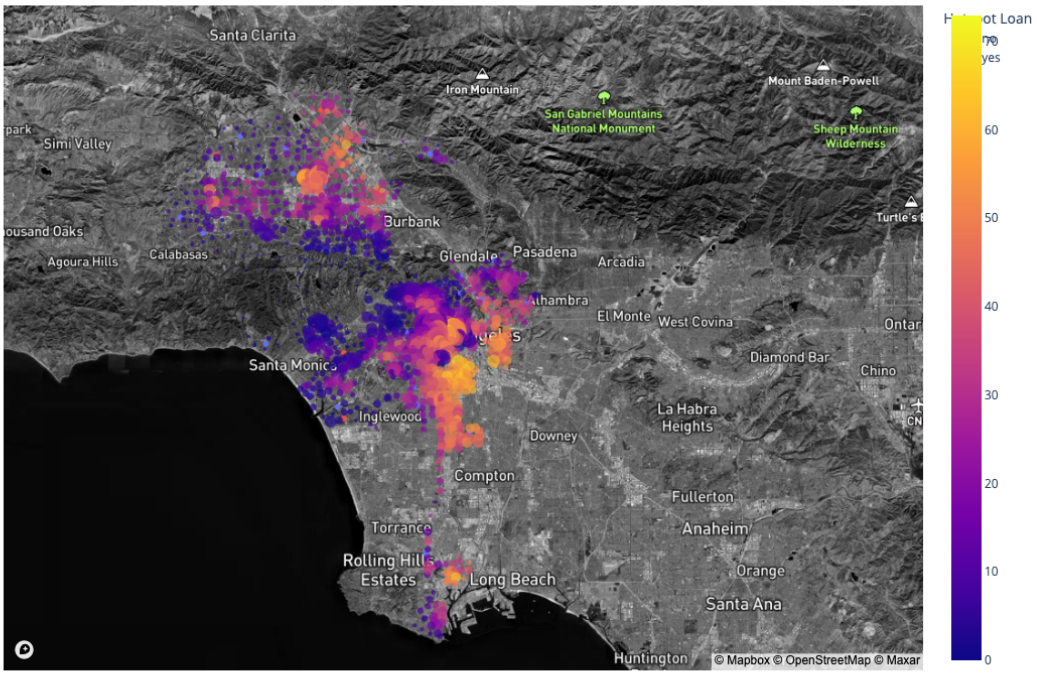
[151]:
```
hh = hh.to_crs(epsg=3857)
```

[152]:
```
#before this can be done, must add popoulation density metric through a join/
 ↪merge
#lib_hh3 = px.scatter_mapbox(lib_gdf, lat="lat", lon='lon', color = "Hotspot␣
 ↪Loan",
#                     mapbox_style=mapboxstyle,
#                      width = 1000,
#                      height = 800,
#                      zoom=9,
#                      center = {"lat": 34.0504, "lon": -118.2555},
#                      hover_name='Branch Name',
#                      hover_data=['Address',
#                                  'CityZip',
```

```
#                                    'Email',
#                                    'Phone Number'],)
#                                    #'lat':False, Want to remove lat & lon␣
 ↪because it is irrelevant info for most people
#                                    #'lon':False],
#                       #showlegend = False,
#                       #title='Percent of household completely lacking access')
#lib_hh4 = px.scatter_mapbox(hh,
#                           lat=hh.geometry_cent.y,
#                           lon=hh.geometry_cent.x,
#                           color="% Population 25 Years and Over: Less than High␣
 ↪School",
#                           size='Population Density (Per Sq. Mile)',
#                           width = 1000,
#                           height = 800,
#                           size_max=15,
#                           #zoom=10,
#                           #title='Percent of household completely lacking␣
 ↪access',
#                           color_continuous_scale=px.colors.sequential.dense)
#
#lib_hh3.add_trace(
#    lib_hh4.data[0]
#)
#lib_hh3.show()
```

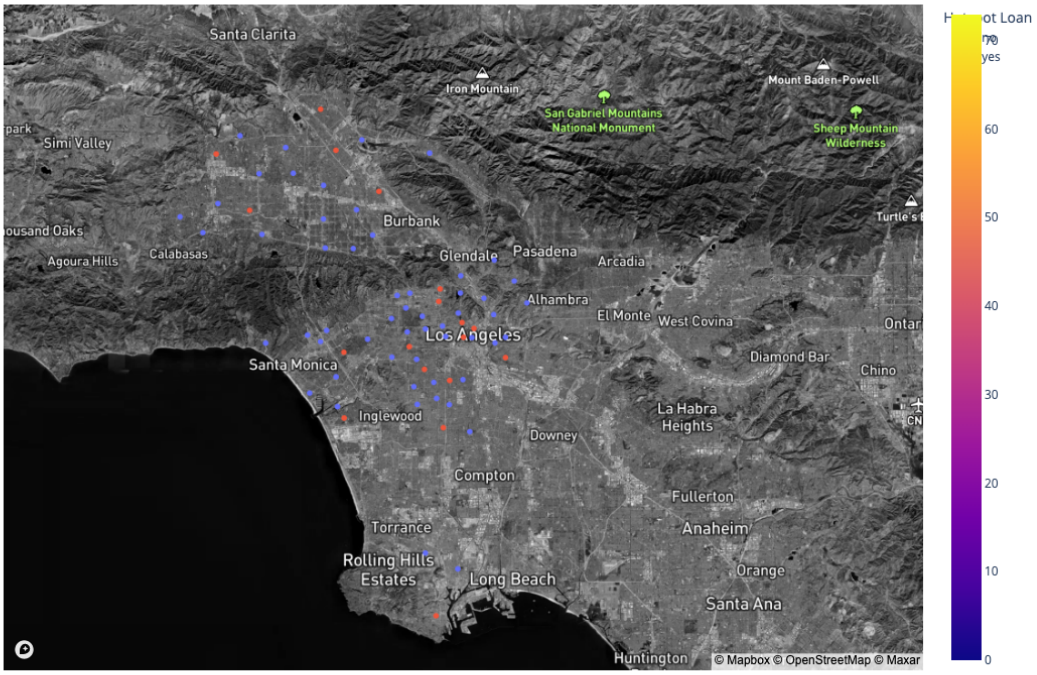
[153]:
```
#Create map above but with census_edu1 data
#Currently in markdown to keep the file size smaller and loads more quickly
lib_educ1 = px.scatter_mapbox(lib_gdf, lat="lat", lon='lon', color = "Hotspot␣
 ↪Loan",
                      mapbox_style=mapboxstyle,
                       width = 1000,
                       height = 800,
                       zoom=9,
                       center = {"lat": 34.0504, "lon": -118.2555},
                       hover_name='Branch Name',
                       hover_data=['Address',
                                   'CityZip',
                                   'Email',
                                   'Phone Number'],)
                                   #'lat':False, Want to remove lat & lon because␣
 ↪it is irrelevant info for most people
                                   #'lon':False],
                       #showlegend = False,
                       #title='Percent of Adult Population With Less Than A High␣
 ↪School Degree')
```

```
lib_educ2 = px.scatter_mapbox(census_edu1,
                        lat=census_edu1.geometry_cent.y,
                        lon=census_edu1.geometry_cent.x,
                        color="% Population 25 Years and Over: Less than High␣
 ↪School",
                        size='Population Density (Per Sq. Mile)',
                        width = 1000,
                        height = 800,
                        size_max=15,
                        #zoom=10,
                        #title='Percent of Adult Population With Less Than A␣
 ↪High School Degree',
                        color_continuous_scale=px.colors.sequential.dense)

lib_educ1.add_trace(
    lib_educ2.data[0]
)
lib_educ1.show()
```



[154]:
```
# save as html file
lib_educ1.write_html("lib_educ1-scale.html")
```

[155]:
```
lib_educ3 = px.scatter_mapbox(lib_gdf, lat="lat", lon='lon', color = "Hotspot␣
↪Loan",
                      mapbox_style=mapboxstyle,
                      width = 1000,
                      height = 800,
                      zoom=9,
                      center = {"lat": 34.0504, "lon": -118.2555},
                      hover_name='Branch Name',
                      hover_data=['Address',
                                  'CityZip',
                                  'Email',
                                  'Phone Number'],)
                                  #'lat':False, Want to remove lat & lon because␣
↪it is irrelevant info for most people
                                  #'lon':False],
                      #title='Percent of Population Over 65')

lib_educ4 = px.choropleth_mapbox(census_edu1,
                      geojson=census_edu1.geometry,
                      locations= census_edu1.index,
                      color = census_edu1["% Population 25 Years and Over: Less␣
↪than High School"],
                      #mapbox_style=mapboxstyle,
                      width = 1000,
                      height = 800,
                      zoom=9,
                      center = {"lat": 34.0504, "lon": -118.2555},
                      #color_continuous_scale=["purple","#FB1E8F"],
                      opacity=0.7,
                      )
lib_educ3.add_trace(
    lib_educ4.data[0]
)
lib_educ3.show()
```

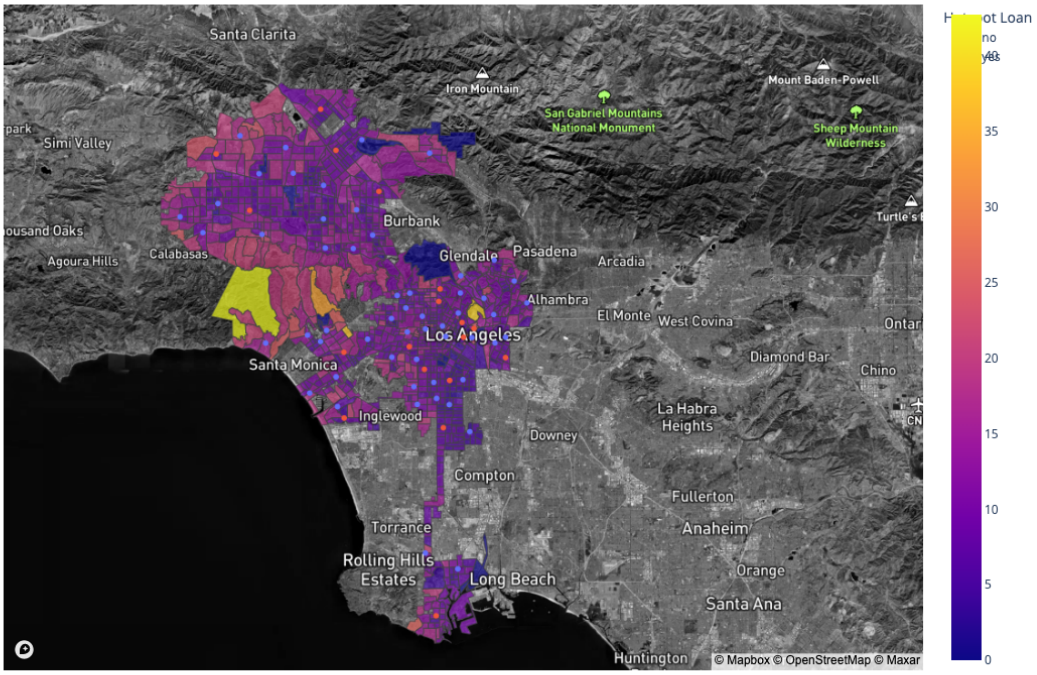


[156]:
```
#Beginning to build above map with Graph objects rather than Express...A work␣
 ↪in progress
#site_lat = lib_gdf.lat
#site_lon = lib_gdf.lon
#fig = go.Figure()
#
#fig.add_trace(go.Scatter(
#    x=site_lat,
#    y=site_lon,
#    mode = "markers",
#    marker = dict(
#        size = 12,
#        symbol= ["circle-dot"],
#        colorscale='Viridis'))
#    #text=lib_gdf['Branch Name'])
#)
#
#fig.add_trace(go.Scattermapbox(
#    lat=census_edu1.geometry_cent.y,
#    lon=census_edu1.geometry_cent.x,
#    mode="markers",
#    marker = dict(
```

```
#        color=census_edu1["% Population 25 Years and Over: Less than High␣
 ↪School"],
#        colorscale='Viridis',
#    #size='Population Density (Per Sq. Mile)',
#)
#)
#
#fig.update_layout(
#    title='Percent of Adult Population With Less Than A High School Degree',
#    autosize=True,
#    showlegend=False,
#    mapbox = dict(
#        accesstoken = token,
#        style="light",
#    center = go.layout.mapbox.Center(
#        lat=34.0504,
#        lon=-118.2555
#    ),
#        zoom=9
#)
#)
#fig.show()
```

[157]:
```
#I prefer the scatter plot over the chloropleth map, but the makers for the␣
 ↪library are difficult to see. I am going to use a more robust plotly libary␣
 ↪to alter the markers
#
#trace2 = px.scatter_mapbox(census_edu1,
#                        lat=census_edu1.geometry_cent.y,
#                        lon=census_edu1.geometry_cent.x,
#                        color="% Population 25 Years and Over: Less than High␣
 ↪School",
#                        size='Population Density (Per Sq. Mile)',
#                        width = 1000,
#                        height = 800,
#                        size_max=15,
#                        zoom=10,
#                        title='Percent of Adult Population With Less Than A␣
 ↪High School Degree',
#                        color_continuous_scale=px.colors.sequential.dense)
#trace1.add_trace(
#    trace2.data[0]
#)
#trace1.show()
```

[158]:
```
#Create map above but with census_age data
lib_age1 = px.scatter_mapbox(lib_gdf, lat="lat", lon='lon', color = "Hotspot␣
 ↪Loan",
                      mapbox_style=mapboxstyle,
                      width = 1000,
                      height = 800,
                      zoom=9,
                      center = {"lat": 34.0504, "lon": -118.2555},
                      hover_name='Branch Name',
                      hover_data=['Address',
                                  'CityZip',
                                  'Email',
                                  'Phone Number'],)
                                  #'lat':False, Want to remove lat & lon because␣
 ↪it is irrelevant info for most people
                                  #'lon':False],
                      #title='Percent of Population Over 65')

lib_age2 = px.choropleth_mapbox(census_age,
                      geojson=census_age.geometry,
                      locations= census_age.index,
                      color = census_age['Percent over 65'],
                      #mapbox_style=mapboxstyle,
                      width = 1000,
                      height = 800,
                      #zoom=10,
                      center = {"lat": 34.0504, "lon": -118.2555},
                      color_continuous_scale=["purple","#FB1E8F"],
                      opacity=0.7,
                      )
lib_age1.add_trace(
    lib_age2.data[0]
)
lib_age1.show()
```



[159]:
```
# save as html file
lib_age1.write_html("lib_age-chlor.html")
```

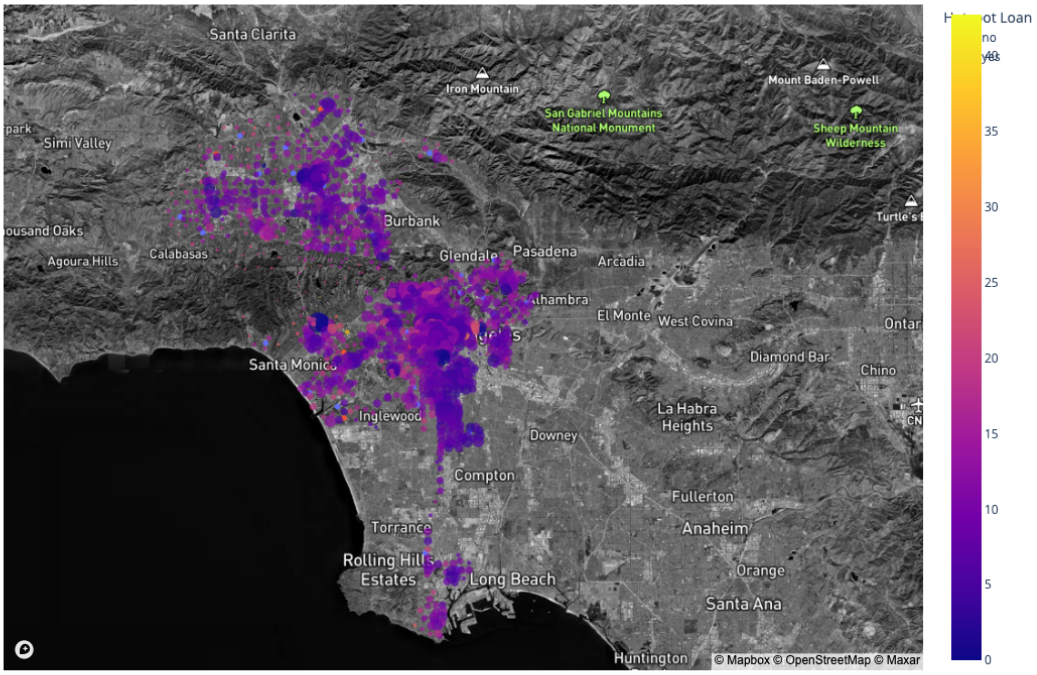
[160]:
```
age_gdf = gpd.GeoDataFrame(
    census_age, geometry=gpd.points_from_xy(census_age.geometry_cent.x,␣
↪census_age.geometry_cent.y),crs="EPSG:4326")
```

[161]:
```
census_edu1 = census_edu1.to_crs(epsg=3857)
```

[162]:
```
census_age['Population Density (Per Sq. Mile)'] = census_age['Population␣
↪Density (Per Sq. Mile)'].astype(float)
#census_age['% Population 25 Years and Over: Less than High School'] =␣
↪census_age['% Population 25 Years and Over: Less than High School'].
↪astype(float)
```

[163]:
```
lib_age3 = px.scatter_mapbox(lib_gdf, lat="lat", lon='lon', color = "Hotspot␣
↪Loan",
                    mapbox_style=mapboxstyle,
                     width = 1000,
                     height = 800,
                     zoom=9,
```

```
                      center = {"lat": 34.0504, "lon": -118.2555},
                      hover_name='Branch Name',
                      hover_data=['Address',
                                  'CityZip',
                                  'Email',
                                  'Phone Number'],)
                      #title='Percent of Adult Population Over 65')
lib_age4 = px.scatter_mapbox(census_age,
                         lat=census_age.geometry_cent.y,
                         lon=census_age.geometry_cent.x,
                         color="Percent over 65",
                         size='Population Density (Per Sq. Mile)',
                         width = 1000,
                         height = 800,
                         size_max=15,
                         #zoom=10,
                         #title='Percent of Adult Population Over 65',
                         color_continuous_scale=px.colors.sequential.dense)

lib_age3.add_trace(
    lib_age4.data[0]
)
lib_age3.show()
```

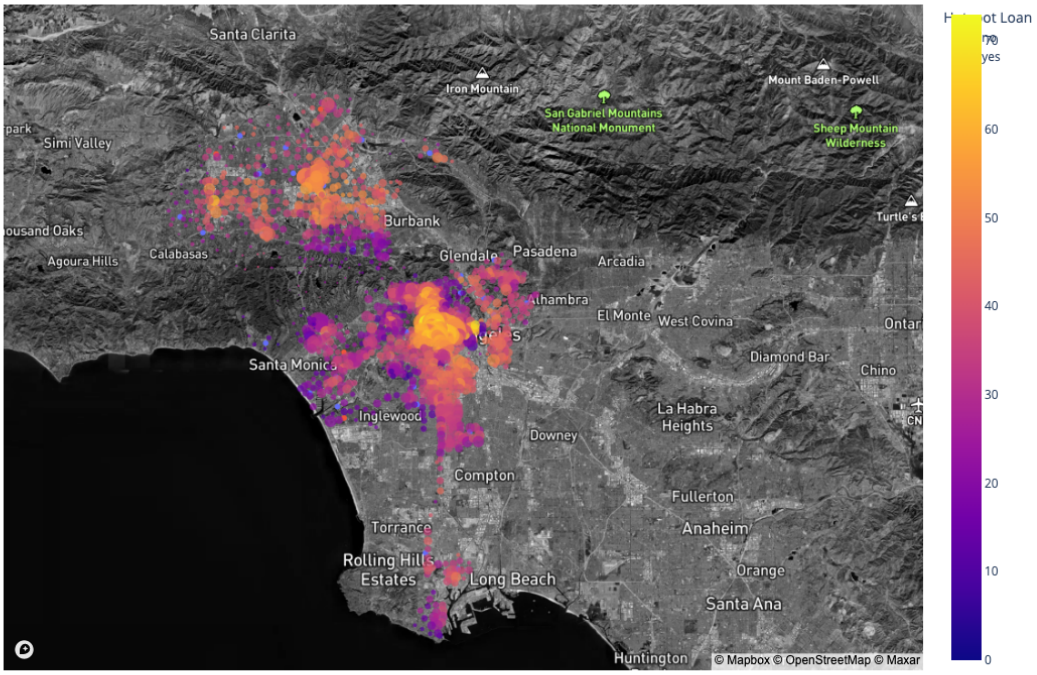


[164]:
```
# save as html file
lib_age3.write_html("lib_age-scatter.html")
```

[165]:
```
fb_gdf = gpd.GeoDataFrame(
    census_fb, geometry=gpd.points_from_xy(census_fb.geometry_cent.x, census_fb.
 ↪geometry_cent.y),crs="EPSG:4326")
```

[166]:
```
census_fb = census_fb.to_crs(epsg=3857)
```

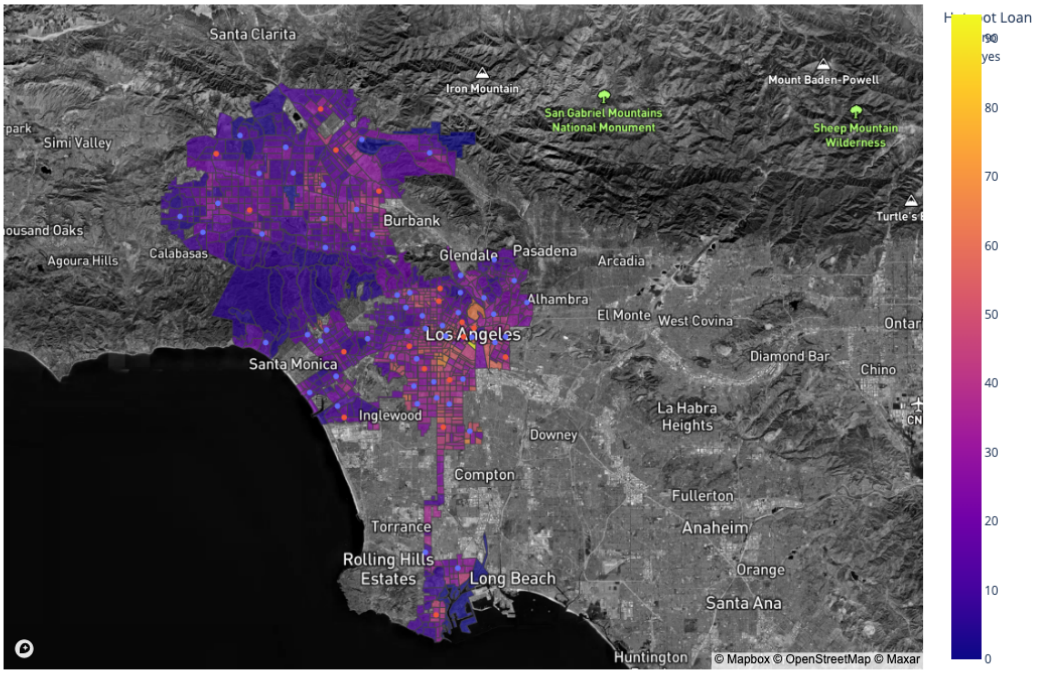
[167]:
```
fig_fb1 = px.scatter_mapbox(lib_gdf, lat="lat", lon='lon', color = "Hotspot␣
 ↪Loan",
                     mapbox_style=mapboxstyle,
                      width = 1000,
                      height = 800,
                      zoom=9,
                      center = {"lat": 34.0504, "lon": -118.2555},
                      hover_name='Branch Name',
                      hover_data=['Address',
                                  'CityZip',
                                  'Email',
                                  'Phone Number'],)
                      #title='Percent of Total Population that is Foreign Born')
fig_fb2 = px.scatter_mapbox(census_fb,
                         lat=census_fb.geometry_cent.y,
                         lon=census_fb.geometry_cent.x,
                         color='% Total Population: Foreign Born',
                         size='Population Density (Per Sq. Mile)',
                         width = 1000,
                         height = 800,
                         size_max=15,
                         #zoom=10,
                         #title='Percent of Total Population that is Foreign␣
 ↪Born',
                         color_continuous_scale=px.colors.sequential.dense)

fig_fb1.add_trace(
    fig_fb2.data[0]
)
fig_fb1.show()
```



[168]:
```
# save as html file
fig_fb1.write_html("lib_fb-scatter.html")
```

[169]:
```
#Currently in markdown to keep the file size smaller and loads more quickly

#Create map above but with census_inc data
fig_inc1 = px.scatter_mapbox(lib_gdf, lat="lat", lon='lon', color = "Hotspot␣
 ↪Loan",
                    mapbox_style=mapboxstyle,
                    width = 1000,
                    height = 800,
                    zoom=9,
                    center = {"lat": 34.0504, "lon": -118.2555},
                    hover_name='Branch Name',
                    hover_data=['Address',
                                'CityZip',
                                'Email',
                                'Phone Number'],)
                                #'lat':False, Want to remove lat & lon because␣
 ↪it is irrelevant info for most people
                                #'lon':False],
```

```
                    #title='Percent of Census Tract earning less than $30k')

fig_inc2 = px.choropleth_mapbox(census_inc,
                    geojson=census_inc.geometry,
                    locations= census_inc.index,
                    color = census_inc['Percent less than $30k'],
                    #mapbox_style=mapboxstyle,
                    width = 1000,
                    height = 800,
                    zoom=9,
                    center = {"lat": 34.0504, "lon": -118.2555},
                    color_continuous_scale=["purple","#FB1E8F"],
                    opacity=0.7,
                    )
fig_inc1.add_trace(
    fig_inc2.data[0]
)
fig_inc1.show()
```



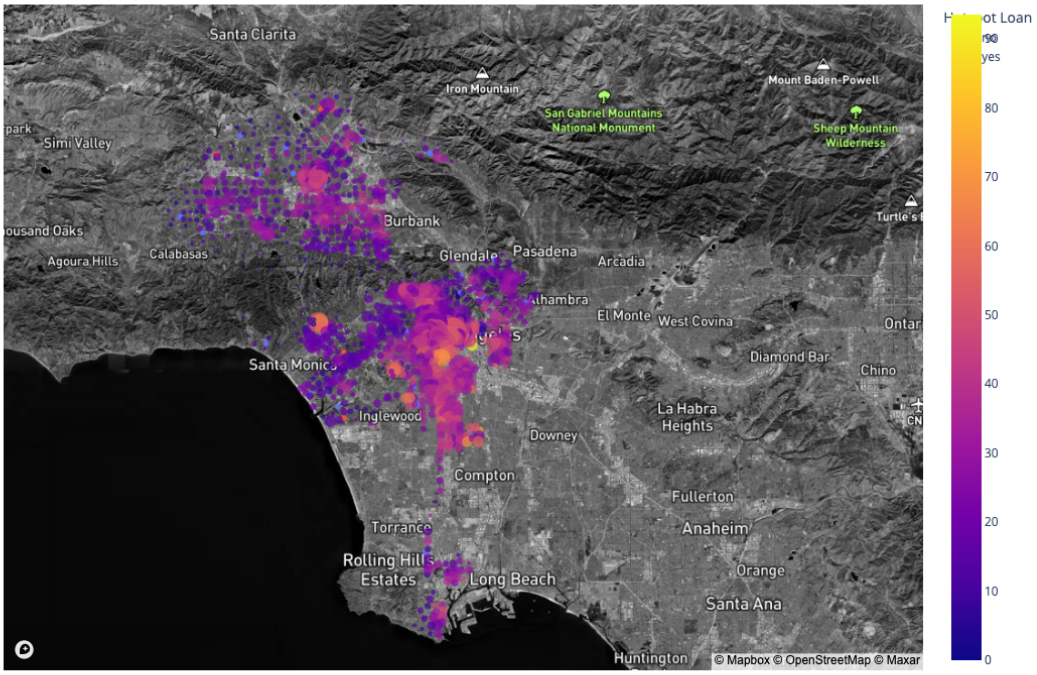
[170]:
```
# save as html file
fig_inc1.write_html("lib_inc-chlor.html")
```

[171]:
```
census_inc['Population Density (Per Sq. Mile)'] = census_inc['Population␣
↪Density (Per Sq. Mile)'].astype(float)
```

[172]:
```
inc_gdf = gpd.GeoDataFrame(
    census_inc, geometry=gpd.points_from_xy(census_inc.geometry_cent.x,␣
↪census_inc.geometry_cent.y),crs="EPSG:4326")
```

[173]:
```
census_inc = census_inc.to_crs(epsg=3857)
```

[174]:
```
fig_inc3 = px.scatter_mapbox(lib_gdf, lat="lat", lon='lon', color = "Hotspot␣
↪Loan",
                     mapbox_style=mapboxstyle,
                      width = 1000,
                      height = 800,
                      zoom=9,
                      center = {"lat": 34.0504, "lon": -118.2555},
                      hover_name='Branch Name',
                      hover_data=['Address',
                                  'CityZip',
                                  'Email',
                                  'Phone Number'],)
                      #title='Percent of Census Tract earning less than $30k')
fig_inc4 = px.scatter_mapbox(census_inc,
                         lat=census_inc.geometry_cent.y,
                         lon=census_inc.geometry_cent.x,
                         color='Percent less than $30k',
                         size='Population Density (Per Sq. Mile)',
                         width = 1000,
                         height = 800,
                         size_max=15,
                         #zoom=10,
                         #title='Percent of Census Tract earning less than $30k',
                         color_continuous_scale=px.colors.sequential.dense)

fig_inc3.add_trace(
    fig_inc4.data[0]
)
fig_inc3.show()
```



[175]:
```
# save as html file
fig_inc3.write_html("lib_inc-scatter.html")
```

1.5 Spatial Analysis

[176]:
```
#I will begin by creating a spatial autocorrelation of broadband access
```

[177]:
```
#calculate spatial weight
wq =  lps.weights.KNN.from_dataframe(hh,k=8)

#Row-standardization
wq.transform = 'r'
```

[178]:
```
hh
```

[178]:

	geoid	name	Total	\
0	14000US06037101110	Census Tract 1011.10, Los Angeles, CA	1575.0	
1	14000US06037101122	Census Tract 1011.22, Los Angeles, CA	1312.0	
2	14000US06037101210	Census Tract 1012.10, Los Angeles, CA	2297.0	
3	14000US06037101220	Census Tract 1012.20, Los Angeles, CA	1357.0	
4	14000US06037101300	Census Tract 1013, Los Angeles, CA	1445.0	

```
…                  …                                          …     …
994   14000US06037980019  Census Tract 9800.19, Los Angeles, CA    65.0
996   14000US06037980021  Census Tract 9800.21, Los Angeles, CA     8.0
999   14000US06037980024  Census Tract 9800.24, Los Angeles, CA    75.0
1000  14000US06037980026  Census Tract 9800.26, Los Angeles, CA     4.0
1002  14000US06037980031  Census Tract 9800.31, Los Angeles, CA    24.0

      Percent with an internet subscription  \
0                                  77.650794
1                                  83.460366
2                                  73.443622
3                                  74.944731
4                                  82.006920
…                                      …
994                               100.000000
996                               100.000000
999                                94.666667
1000                              100.000000
1002                              100.000000

      Percent with a cellular data plan with no other type of Internet  \
0                                                      6.158730
1                                                      8.612805
2                                                      4.919460
3                                                     10.464259
4                                                      4.844291
…                                                         …
994                                                    0.000000
996                                                    0.000000
999                                                    0.000000
1000                                                   0.000000
1002                                                   0.000000

      Percent with no internet access                          geometry  \
0                           16.317460  POINT (-13168315.067 4063696.435)
1                            9.679878  POINT (-13167998.945 4064807.386)
2                           23.421855  POINT (-13168063.933 4062820.829)
3                           20.707443  POINT (-13167051.132 4062637.118)
4                           11.141869  POINT (-13165867.264 4062255.399)
…                               …                                   …
994                          0.000000  POINT (-13196723.209 4041067.976)
996                          0.000000  POINT (-13177721.527 4064657.496)
999                          5.333333  POINT (-13190092.890 4052550.257)
1000                         0.000000  POINT (-13166780.915 4065812.558)
1002                         0.000000  POINT (-13164957.075 3993358.055)

                    geometry_cent
```

```
0     POINT (-118.29299 34.25947)
1     POINT (-118.29015 34.26772)
2     POINT (-118.29073 34.25297)
3     POINT (-118.28163 34.25161)
4     POINT (-118.27100 34.24877)
…                             …
994   POINT (-118.54818 34.09130)
996   POINT (-118.37749 34.26661)
999   POINT (-118.48862 34.17668)
1000  POINT (-118.27921 34.27518)
1002  POINT (-118.26282 33.73563)

[997 rows x 8 columns]
```

[179]:
```
#create a new column for the spatial lag for residents with no access
hh['no access_lag'] = lps.weights.lag_spatial(wq, hh['Percent with no internet␣
↪access'])
```

[180]:
```
#create a new column for the spatial lag of percent of residents with only a␣
↪cellular data plan
hh['cellular_lag'] = lps.weights.lag_spatial(wq, hh['Percent with a cellular␣
↪data plan with no other type of Internet'])
```

[181]:
```
hh.sample(10)
```

[181]:
```
                  geoid                                      name   Total  \
933  14000US06037277200     Census Tract 2772, Los Angeles, CA  1009.0
206  14000US06037125200     Census Tract 1252, Los Angeles, CA  1653.0
707  14000US06037227020  Census Tract 2270.20, Los Angeles, CA   770.0
39   14000US06037106112  Census Tract 1061.12, Los Angeles, CA  1447.0
551  14000US06037208402  Census Tract 2084.02, Los Angeles, CA   827.0
636  14000US06037216200     Census Tract 2162, Los Angeles, CA  2811.0
74   14000US06037109700     Census Tract 1097, Los Angeles, CA  1366.0
108  14000US06037115301  Census Tract 1153.01, Los Angeles, CA  1653.0
110  14000US06037115401  Census Tract 1154.01, Los Angeles, CA  1933.0
67   14000US06037109200     Census Tract 1092, Los Angeles, CA   944.0

     Percent with an internet subscription  \
933                              77.998018
206                              73.805203
707                              74.935065
39                               92.121631
551                              82.224909
636                              91.675560
74                               96.925329
108                              86.025408
110                              90.739783
```

```
67                                95.868644

     Percent with a cellular data plan with no other type of Internet  \
933                                                 10.802775
206                                                  6.291591
707                                                 21.818182
39                                                   6.634416
551                                                 12.454655
636                                                  5.905372
74                                                   5.856515
108                                                  8.771930
110                                                 13.605794
67                                                   7.627119

     Percent with no internet access                          geometry  \
933                        22.001982  POINT (-13178250.508 4022443.005)
206                        18.027828  POINT (-13178375.884 4050965.442)
707                        21.818182  POINT (-13163744.964 4030917.555)
39                          6.565308  POINT (-13181612.848 4069896.221)
551                        15.356711  POINT (-13165312.368 4037985.398)
636                         2.881537  POINT (-13174465.976 4036475.014)
74                          2.196193  POINT (-13189232.136 4064027.856)
108                        13.369631  POINT (-13196755.968 4061432.587)
110                         7.346094  POINT (-13193043.582 4059028.300)
67                          4.131356  POINT (-13188536.019 4066191.283)

                  geometry_cent  no access_lag  cellular_lag
933  POINT (-118.38224 33.95263)       9.015716      5.909473
206  POINT (-118.38336 34.16490)      12.303948      5.898446
707  POINT (-118.25193 34.01575)      27.075993     15.415546
39   POINT (-118.41244 34.30549)      13.116122     10.077192
551  POINT (-118.26601 34.06836)      22.975776     12.418256
636  POINT (-118.34824 34.05712)       5.940069      6.231934
74   POINT (-118.48089 34.26193)       8.660948      6.546227
108  POINT (-118.54848 34.24266)       7.344740      7.040273
110  POINT (-118.51513 34.22481)       7.299646      7.882797
67   POINT (-118.47463 34.27799)       9.996111      6.551303
```

[182]:
```
#calculate the difference between access and lag
hh['no access_diff'] = hh['Percent with no internet access'] - hh['no␣
↪access_lag']
```

[183]:
```
#calculate the difference between cellular access and lag
hh['cellular_diff'] = hh['Percent with a cellular data plan with no other type␣
↪of Internet'] - hh['cellular_lag']
```

[184]:
```
hh.sort_values(by='no access_diff')
```

[184]:

	geoid	name	Total	\
532	14000US06037206031	Census Tract 2060.31, Los Angeles, CA	2469.0	
531	14000US06037206020	Census Tract 2060.20, Los Angeles, CA	297.0	
546	14000US06037207900	Census Tract 2079, Los Angeles, CA	4204.0	
543	14000US06037207501	Census Tract 2075.01, Los Angeles, CA	1395.0	
523	14000US06037204700	Census Tract 2047, Los Angeles, CA	1267.0	
..	…	…	…	
45	14000US06037106407	Census Tract 1064.07, Los Angeles, CA	913.0	
538	14000US06037207102	Census Tract 2071.02, Los Angeles, CA	1329.0	
534	14000US06037206050	Census Tract 2060.50, Los Angeles, CA	805.0	
544	14000US06037207502	Census Tract 2075.02, Los Angeles, CA	2196.0	
536	14000US06037206300	Census Tract 2063, Los Angeles, CA	2433.0	

	Percent with an internet subscription	\
532	93.317132	
531	97.306397	
546	93.910561	
543	97.132616	
523	78.531965	
..	…	
45	69.112815	
538	62.452972	
534	47.950311	
544	57.695811	
536	31.113851	

	Percent with a cellular data plan with no other type of Internet	\
532	3.928716	
531	4.377104	
546	4.709800	
543	5.949821	
523	12.549329	
..	…	
45	11.391019	
538	6.922498	
534	10.310559	
544	5.737705	
536	7.768187	

	Percent with no internet access	geometry	\
532	3.888214	POINT (-13161592.692 4033055.418)	
531	2.693603	POINT (-13161496.952 4036531.682)	
546	4.234063	POINT (-13164661.654 4034245.181)	
543	1.290323	POINT (-13163860.126 4036383.355)	
523	13.575375	POINT (-13159349.905 4032962.926)	
..	…	…	
45	29.244250	POINT (-13184297.243 4070581.169)	

```
538                      36.945071  POINT (-13162362.011 4036765.352)
534                      48.322981  POINT (-13160501.011 4033034.843)
544                      39.890710  POINT (-13163873.096 4035881.492)
536                      62.515413  POINT (-13162774.317 4034323.546)

                  geometry_cent  no access_lag  cellular_lag  no access_diff  \
532  POINT (-118.23260 34.03167)      34.754754     13.009724      -30.866541
531  POINT (-118.23174 34.05755)      25.981114      8.723361      -23.287511
546  POINT (-118.26017 34.04053)      26.802993     10.369239      -22.568930
543  POINT (-118.25297 34.05644)      21.710977      8.544705      -20.420655
523  POINT (-118.21245 34.03098)      32.462533     11.162010      -18.887159
..                           …            …             …              …
45   POINT (-118.43656 34.31057)      10.213078      7.849581       19.031172
538  POINT (-118.23951 34.05929)      16.407947      7.575659       20.537124
534  POINT (-118.22279 34.03152)      27.124138     11.052794       21.198843
544  POINT (-118.25308 34.05271)      16.571564      8.063647       23.319147
536  POINT (-118.24321 34.04111)      21.605121      6.629384       40.910292

     cellular_diff
532      -9.081008
531      -4.346257
546      -5.659438
543      -2.594884
523       1.387319
..             …
45        3.541438
538      -0.653161
534      -0.742235
544      -2.325942
536       1.138803

[997 rows x 12 columns]
```

[185]:
```
hh.sort_values(by='cellular_diff')
```

[185]:
```
                   geoid                                      name   Total  \
992   14000US06037980014  Census Tract 9800.14, Los Angeles, CA     5.0
1002  14000US06037980031  Census Tract 9800.31, Los Angeles, CA    24.0
489   14000US06037199300     Census Tract 1993, Los Angeles, CA  1315.0
999   14000US06037980024  Census Tract 9800.24, Los Angeles, CA    75.0
783   14000US06037238400     Census Tract 2384, Los Angeles, CA  1906.0
…                    …                                      …     …
224   14000US06037127605  Census Tract 1276.05, Los Angeles, CA  1170.0
471   14000US06037195802  Census Tract 1958.02, Los Angeles, CA   938.0
374   14000US06037183820  Census Tract 1838.20, Los Angeles, CA  1140.0
988   14000US06037930200     Census Tract 9302, Los Angeles, CA   147.0
987   14000US06037701100     Census Tract 7011, Los Angeles, CA    48.0
```

```
      Percent with an internet subscription  \
992                                100.000000
1002                               100.000000
489                                 89.885932
999                                 94.666667
783                                 69.254984
…                                         …
224                                 84.358974
471                                 89.978678
374                                 85.614035
988                                 92.517007
987                                 91.666667

      Percent with a cellular data plan with no other type of Internet  \
992                                                 0.000000
1002                                                0.000000
489                                                 4.334601
999                                                 0.000000
783                                                 6.033578
…                                                      …
224                                                28.803419
471                                                33.049041
374                                                36.403509
988                                                34.013605
987                                                56.250000

      Percent with no internet access                          geometry  \
992                          0.000000  POINT (-13162233.577 3999642.483)
1002                         0.000000  POINT (-13164957.075 3993358.055)
489                          9.657795  POINT (-13157876.265 4041246.108)
999                          5.333333  POINT (-13190092.890 4052550.257)
783                         23.452256  POINT (-13170096.711 4022942.269)
…                              …                                  …
224                          8.461538  POINT (-13189782.396 4054379.327)
471                          9.488273  POINT (-13166612.898 4038967.260)
374                         14.385965  POINT (-13157643.812 4042769.767)
988                          7.482993  POINT (-13167795.245 4074663.822)
987                          0.000000  POINT (-13186573.611 4036536.966)

                    geometry_cent  no access_lag  cellular_lag  \
992   POINT (-118.23836 33.78256)      16.176595     23.398366
1002  POINT (-118.26282 33.73563)      17.798419     15.841763
489   POINT (-118.19921 34.09263)      13.160369     17.533400
999   POINT (-118.48862 34.17668)       9.008535     12.298649
783   POINT (-118.30899 33.95635)      20.926566     18.069675
…                           …            …            …
```

```
224   POINT (-118.48583 34.19027)      11.479385     11.181521
471   POINT (-118.27770 34.07567)      11.205183     15.352719
374   POINT (-118.19713 34.10396)      10.227516     14.834889
988   POINT (-118.28832 34.34086)      11.482896      6.138625
987   POINT (-118.45701 34.05759)       3.372899      6.013215

      no access_diff  cellular_diff
992        -16.176595     -23.398366
1002       -17.798419     -15.841763
489         -3.502574     -13.198799
999         -3.675202     -12.298649
783          2.525690     -12.036096
…                 …              …
224         -3.017847      17.621897
471         -1.716911      17.696322
374          4.158448      21.568620
988         -3.999903      27.874981
987         -3.372899      50.236785

[997 rows x 12 columns]
```

[186]:
```
#Access: tract with highest negative difference
hh_donut = hh.sort_values(by='no access_diff').head(1)
hh_donut
```

[186]:
```
                  geoid                                    name   Total  \
532  14000US06037206031  Census Tract 2060.31, Los Angeles, CA  2469.0

     Percent with an internet subscription  \
532                              93.317132

     Percent with a cellular data plan with no other type of Internet  \
532                                           3.928716

     Percent with no internet access                            geometry  \
532                         3.888214  POINT (-13161592.692 4033055.418)

                  geometry_cent  no access_lag  cellular_lag  no access_diff  \
532  POINT (-118.23260 34.03167)      34.754754     13.009724      -30.866541

     cellular_diff
532      -9.081008
```

[187]:
```
#tract with highest positive difference
hh_diamond = hh.sort_values(by='no access_diff').tail(1)
hh_diamond
```

[187]:
```
                  geoid                                  name   Total  \
536  14000US06037206300  Census Tract 2063, Los Angeles, CA  2433.0

     Percent with an internet subscription  \
536                             31.113851

     Percent with a cellular data plan with no other type of Internet  \
536                                          7.768187

     Percent with no internet access                           geometry  \
536                        62.515413  POINT (-13162774.317 4034323.546)

                   geometry_cent  no access_lag  cellular_lag  no access_diff  \
536  POINT (-118.24321 34.04111)      21.605121      6.629384       40.910292

     cellular_diff
536       1.138803
```

[188]:
```
#Cellular: tract with highest negative difference
hh_donut2 = hh.sort_values(by='cellular_diff').head(1)
hh_donut2
```

[188]:
```
                  geoid                                    name  Total  \
992  14000US06037980014  Census Tract 9800.14, Los Angeles, CA    5.0

     Percent with an internet subscription  \
992                                 100.0

     Percent with a cellular data plan with no other type of Internet  \
992                                                0.0

     Percent with no internet access                           geometry  \
992                              0.0  POINT (-13162233.577 3999642.483)

                   geometry_cent  no access_lag  cellular_lag  no access_diff  \
992  POINT (-118.23836 33.78256)      16.176595     23.398366      -16.176595

     cellular_diff
992     -23.398366
```

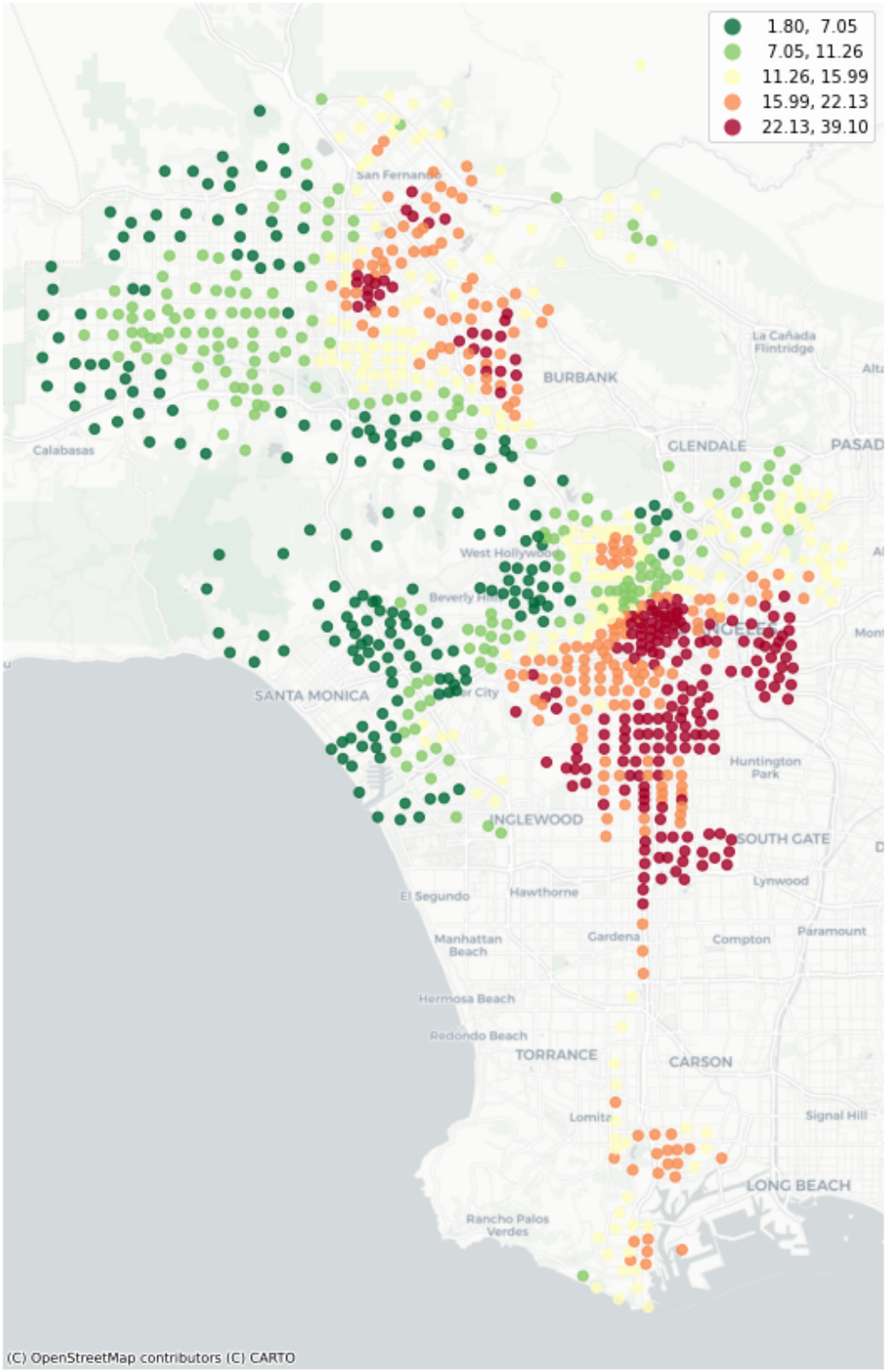
[189]:
```
#tract with highest positive difference
hh_diamond2 = hh.sort_values(by='cellular_diff').tail(1)
hh_diamond2
```

[189]:
```
                  geoid                                  name  Total  \
987  14000US06037701100  Census Tract 7011, Los Angeles, CA   48.0
```

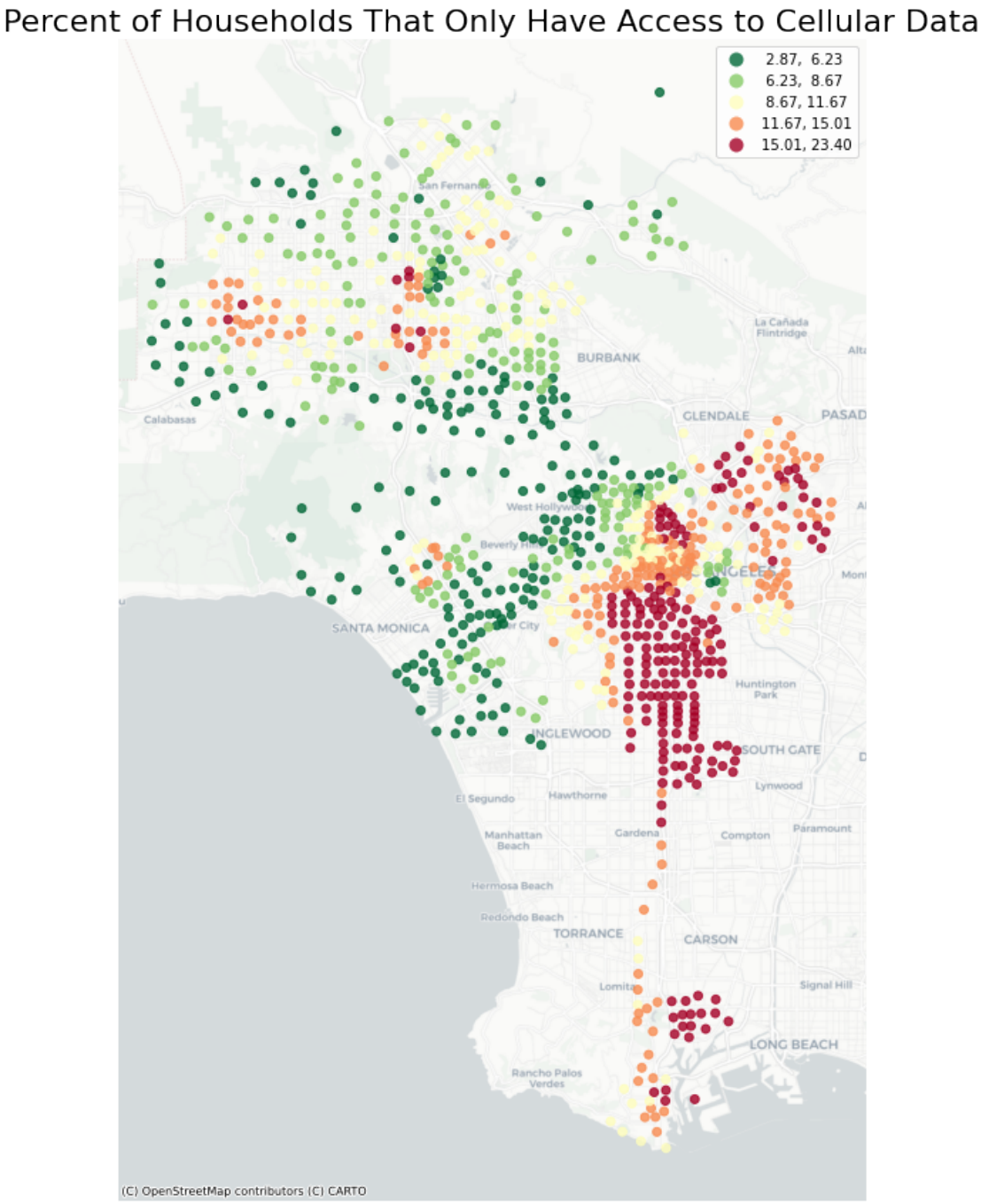
```
     Percent with an internet subscription  \
987                              91.666667

     Percent with a cellular data plan with no other type of Internet  \
987                                              56.25

     Percent with no internet access                            geometry  \
987                              0.0  POINT (-13186573.611 4036536.966)

                    geometry_cent  no access_lag  cellular_lag  no access_diff  \
987  POINT (-118.45701 34.05759)       3.372899      6.013215       -3.372899

     cellular_diff
987      50.236785
```

[190]:
```
#Access: use subplots that make it easier to create multiple layered maps
fig, ax = plt.subplots(figsize=(15, 15))

# spatial lag choropleth
hh.plot(ax=ax,
         figsize=(15,15),
         column='no access_lag',
         legend=True,
         alpha=0.8,
         cmap='RdYlGn_r',
         scheme='quantiles')

ax.axis('off')
ax.set_title('Percent of Household That Lack Broadband Access',fontsize=22)

ctx.add_basemap(ax,source=ctx.providers.CartoDB.Positron)
```

Percent of Household That Lack Broadband Access



[191]:
```
#Cellular: use subplots that make it easier to create multiple layered maps
fig, ax = plt.subplots(figsize=(15, 15))

# spatial lag choropleth
hh.plot(ax=ax,
         figsize=(15,15),
         column='cellular_lag',
         legend=True,
         alpha=0.8,
         cmap='RdYlGn_r',
         scheme='quantiles')

ax.axis('off')
ax.set_title('Percent of Households That Only Have Access to Cellular␣
 ↪Data',fontsize=22)

ctx.add_basemap(ax,source=ctx.providers.CartoDB.Positron)
```



[192]:
```
#To begin Moran's plot, I furst need string together the nanes of the hh columns
hh.columns
```

[192]:
```
Index(['geoid', 'name', 'Total', 'Percent with an internet subscription',
       'Percent with a cellular data plan with no other type of Internet',
       'Percent with no internet access', 'geometry', 'geometry_cent',
```

```
       'no access_lag', 'cellular_lag', 'no access_diff', 'cellular_diff'],
      dtype='object')
```

[194]:
```
hh.columns = ['geoid',
              'name',
              'total',
              'percent_with_subscription',
              'percent_only_cell',
              'percent_no_access',
              'geometry',
              'geometry_cent',
              'no_access_lag',
              'cellular_lag',
              'no_access_diff',
              'cellular_diff']
hh
```

[194]:

	geoid	name	total	\
0	14000US06037101110	Census Tract 1011.10, Los Angeles, CA	1575.0	
1	14000US06037101122	Census Tract 1011.22, Los Angeles, CA	1312.0	
2	14000US06037101210	Census Tract 1012.10, Los Angeles, CA	2297.0	
3	14000US06037101220	Census Tract 1012.20, Los Angeles, CA	1357.0	
4	14000US06037101300	Census Tract 1013, Los Angeles, CA	1445.0	
…	…	…	…	
994	14000US06037980019	Census Tract 9800.19, Los Angeles, CA	65.0	
996	14000US06037980021	Census Tract 9800.21, Los Angeles, CA	8.0	
999	14000US06037980024	Census Tract 9800.24, Los Angeles, CA	75.0	
1000	14000US06037980026	Census Tract 9800.26, Los Angeles, CA	4.0	
1002	14000US06037980031	Census Tract 9800.31, Los Angeles, CA	24.0	

	percent_with_subscription	percent_only_cell	percent_no_access	\
0	77.650794	6.158730	16.317460	
1	83.460366	8.612805	9.679878	
2	73.443622	4.919460	23.421855	
3	74.944731	10.464259	20.707443	
4	82.006920	4.844291	11.141869	
…	…	…	…	
994	100.000000	0.000000	0.000000	
996	100.000000	0.000000	0.000000	
999	94.666667	0.000000	5.333333	
1000	100.000000	0.000000	0.000000	
1002	100.000000	0.000000	0.000000	

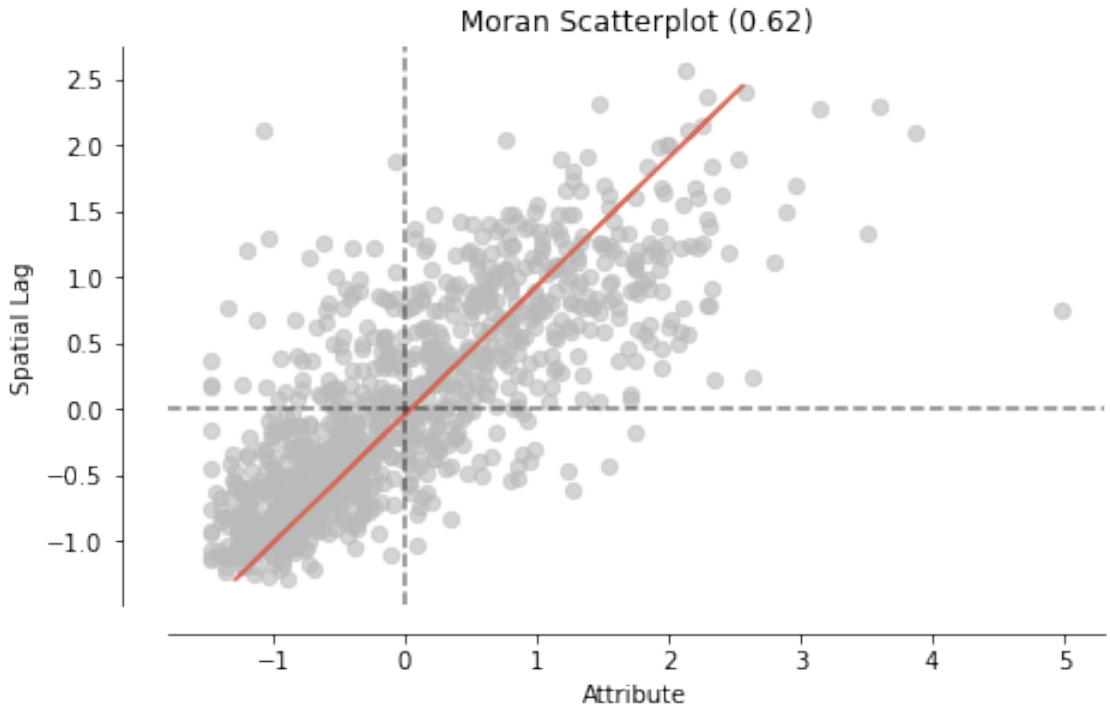
	geometry	geometry_cent	\
0	POINT (-13168315.067 4063696.435)	POINT (-118.29299 34.25947)	
1	POINT (-13167998.945 4064807.386)	POINT (-118.29015 34.26772)	
2	POINT (-13168063.933 4062820.829)	POINT (-118.29073 34.25297)	

```
3     POINT (-13167051.132 4062637.118)  POINT (-118.28163 34.25161)
4     POINT (-13165867.264 4062255.399)  POINT (-118.27100 34.24877)
…                                   …                            …
994   POINT (-13196723.209 4041067.976)  POINT (-118.54818 34.09130)
996   POINT (-13177721.527 4064657.496)  POINT (-118.37749 34.26661)
999   POINT (-13190092.890 4052550.257)  POINT (-118.48862 34.17668)
1000  POINT (-13166780.915 4065812.558)  POINT (-118.27921 34.27518)
1002  POINT (-13164957.075 3993358.055)  POINT (-118.26282 33.73563)

      no_access_lag  cellular_lag  no_access_diff  cellular_diff
0         10.588444      7.069860        5.729017      -0.911130
1         11.906816      6.206276       -2.226938       2.406529
2         10.563585      7.105668       12.858270      -2.186207
3         10.902887      6.412568        9.804556       4.051692
4         12.098583      7.115064       -0.956715      -2.270773
…               …             …               …              …
994        3.880614      5.051632       -3.880614      -5.051632
996       15.989471      7.855571      -15.989471      -7.855571
999        9.008535     12.298649       -3.675202     -12.298649
1000      12.847556      7.840809      -12.847556      -7.840809
1002      17.798419     15.841763      -17.798419     -15.841763

[997 rows x 12 columns]
```

[195]:
```
#Access: Moran's plot
y = hh.percent_no_access
moran = Moran(y, wq)
moran.I
```

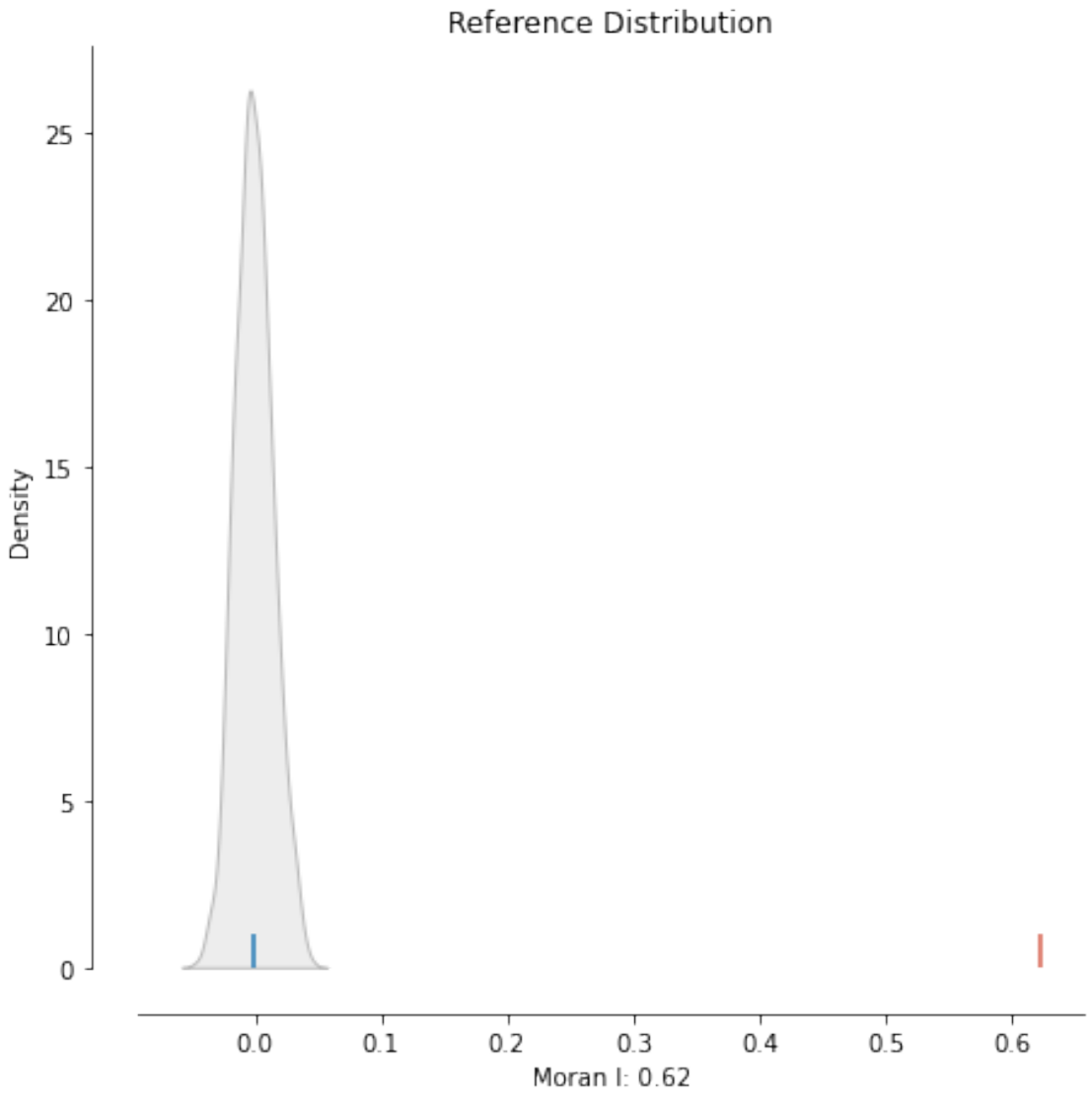
[195]:
```
0.6229996569360207
```

[196]:
```
fig, ax = moran_scatterplot(moran, aspect_equal=True)
plt.show()
```



[197]:
```
plot_moran_simulation(moran,aspect_equal=False)
```

[197]:
```
(<Figure size 504x504 with 1 Axes>,
 <AxesSubplot:title={'center':'Reference Distribution'}, xlabel='Moran I: 0.62',
ylabel='Density'>)
```

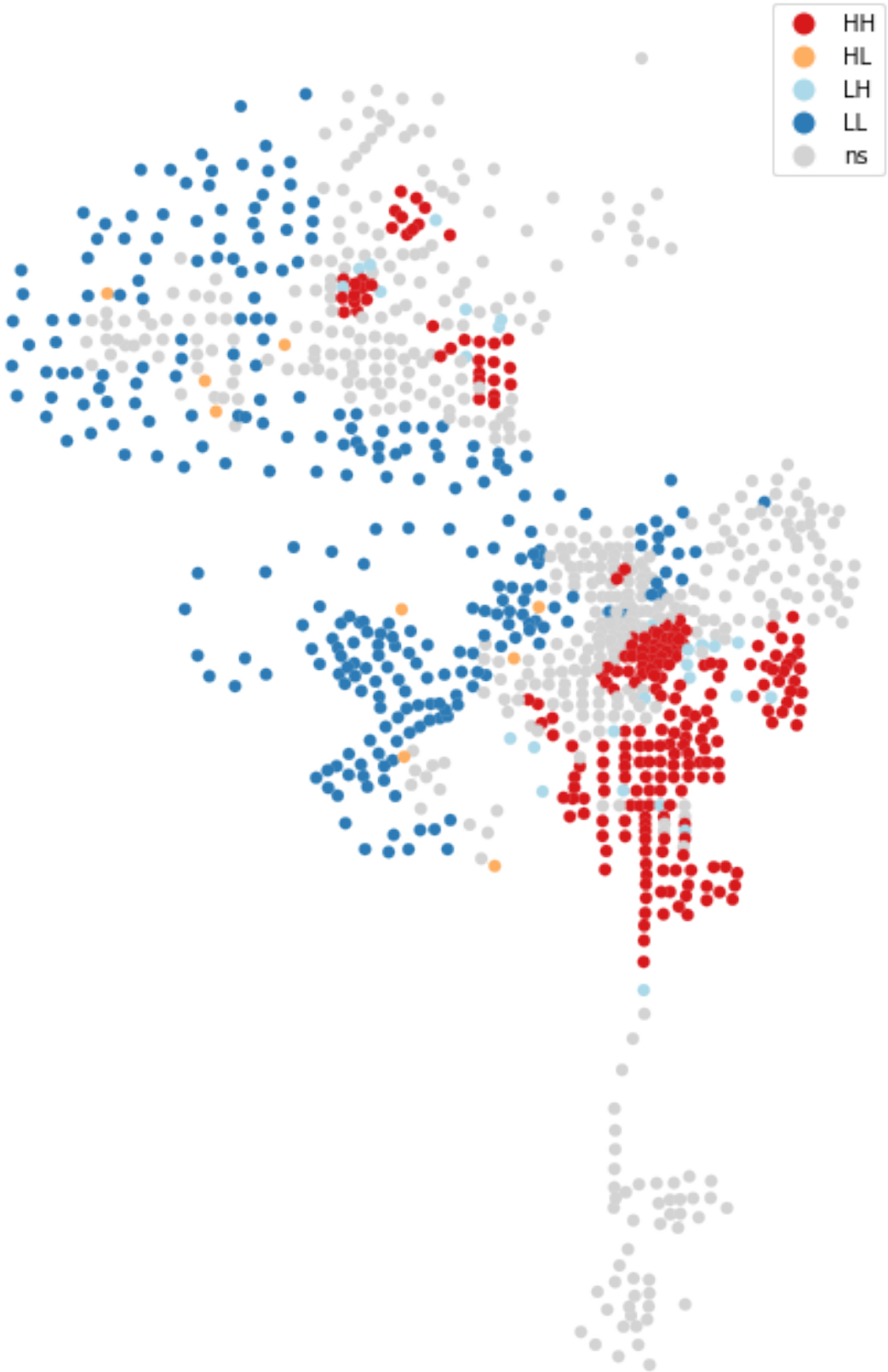


[198]:
```
# calculate local moran values
lisa = esda.moran.Moran_Local(y, wq)
```

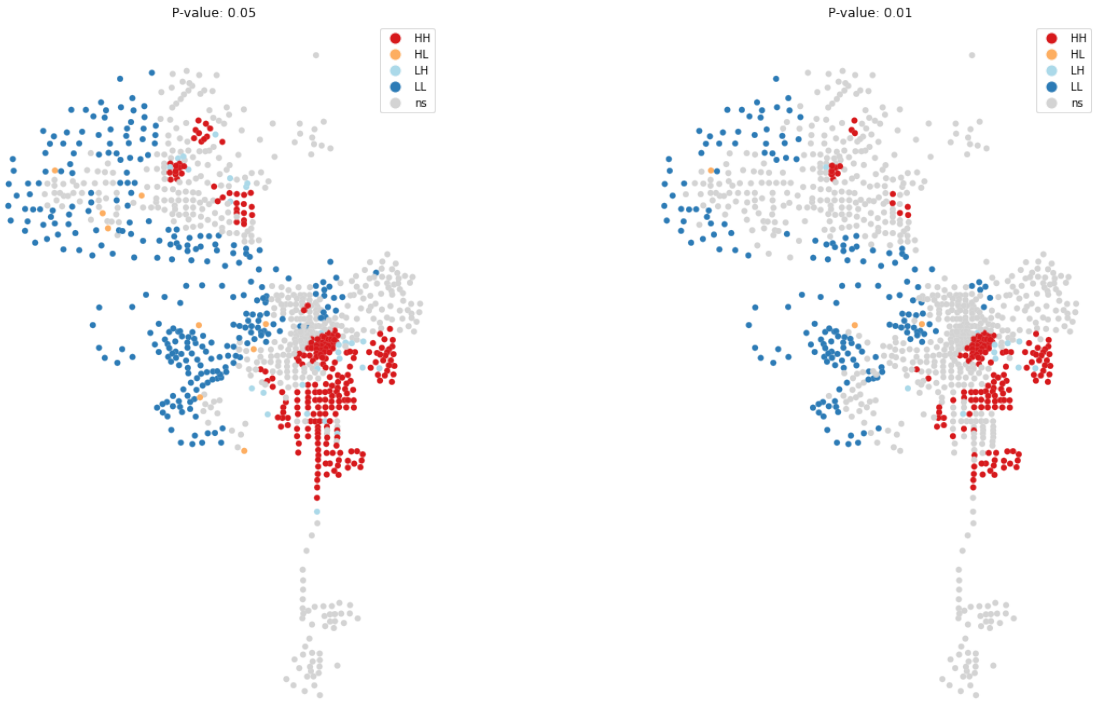
[199]:
```
# Plot
fig,ax = plt.subplots(figsize=(20,15))

moran_scatterplot(lisa, ax=ax, p=0.05)
ax.set_xlabel("Lacking Access")
ax.set_ylabel('Spatial Lag of Access')

# add some labels
plt.text(1.95, 0.5, "HH", fontsize=25)
plt.text(1.95, -1, "HL", fontsize=25)
```

```
plt.text(-2, 1, "LH", fontsize=25)
plt.text(-1, -1, "LL", fontsize=25)
plt.show()
```

[200]:
```
fig, ax = plt.subplots(figsize=(14,12))
lisa_cluster(lisa, hh, p=0.05, ax=ax)
plt.show()
```


HH
HL
LH
LL
ns

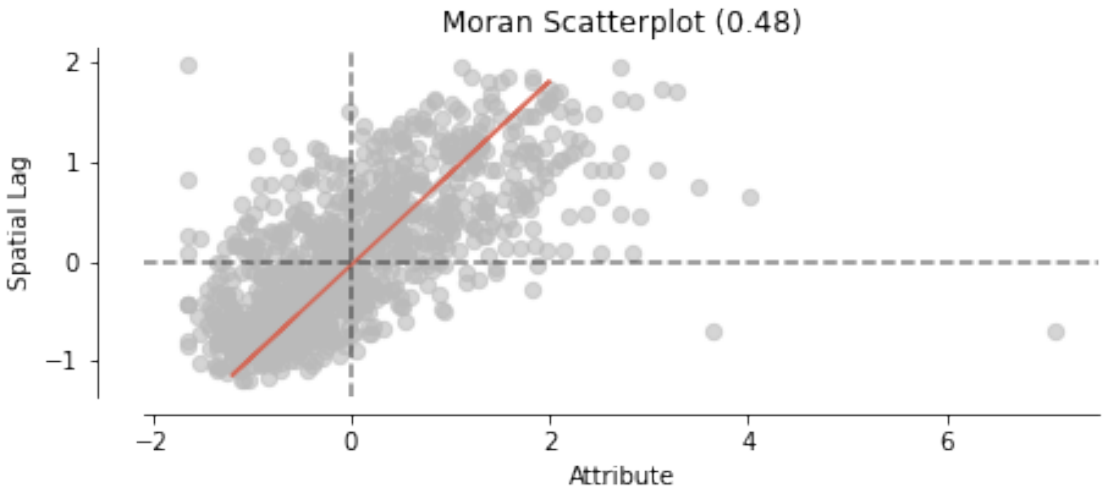
[201]:
```
# create the 1x2 subplots
fig, ax = plt.subplots(1, 2, figsize=(20, 12))

# regular count map on the left
lisa_cluster(lisa, hh, p=0.05, ax=ax[0])

ax[0].axis("off")
ax[0].set_title("P-value: 0.05")

# spatial lag map on the right
lisa_cluster(lisa, hh, p=0.01, ax=ax[1])
ax[1].axis("off")
ax[1].set_title("P-value: 0.01")

plt.show()
```



[202]:
```
#Cellular: Moran's Plot
y = hh.percent_only_cell
moran = Moran(y, wq)
moran.I
```

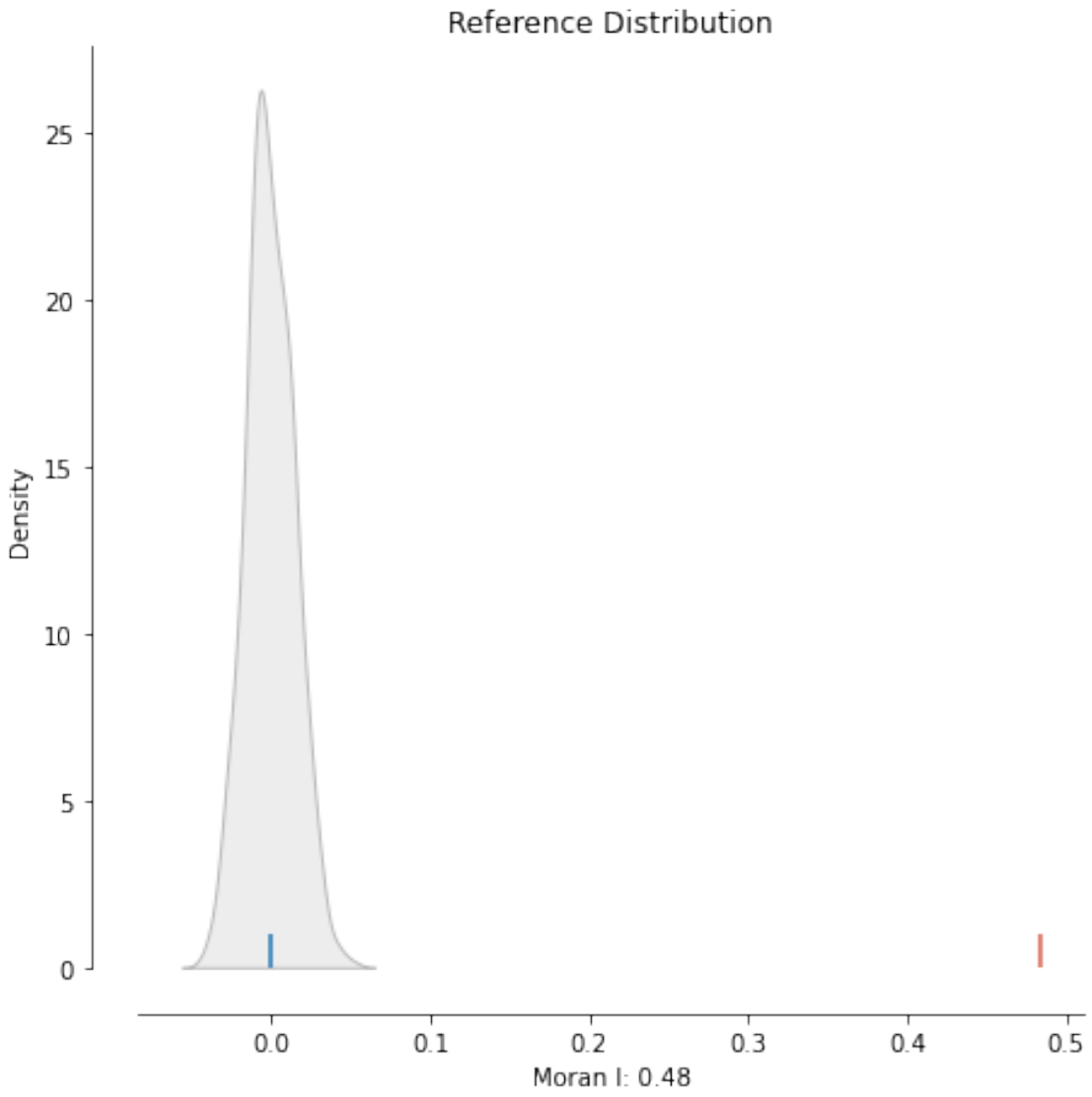
[202]: 0.4841275719503593

[203]:
```
fig, ax = moran_scatterplot(moran, aspect_equal=True)
plt.show()
```



[204]:
```
plot_moran_simulation(moran,aspect_equal=False)
```

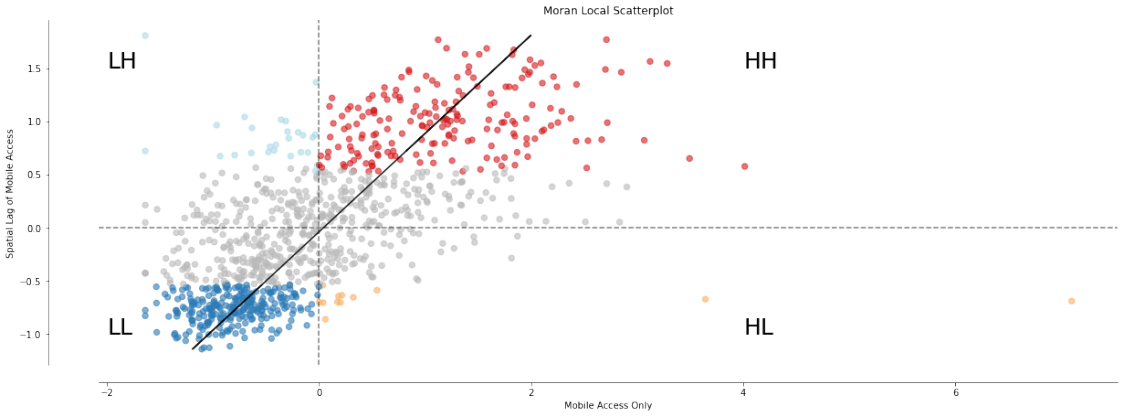
[204]:
```
(<Figure size 504x504 with 1 Axes>,
 <AxesSubplot:title={'center':'Reference Distribution'}, xlabel='Moran I: 0.48',
ylabel='Density'>)
```



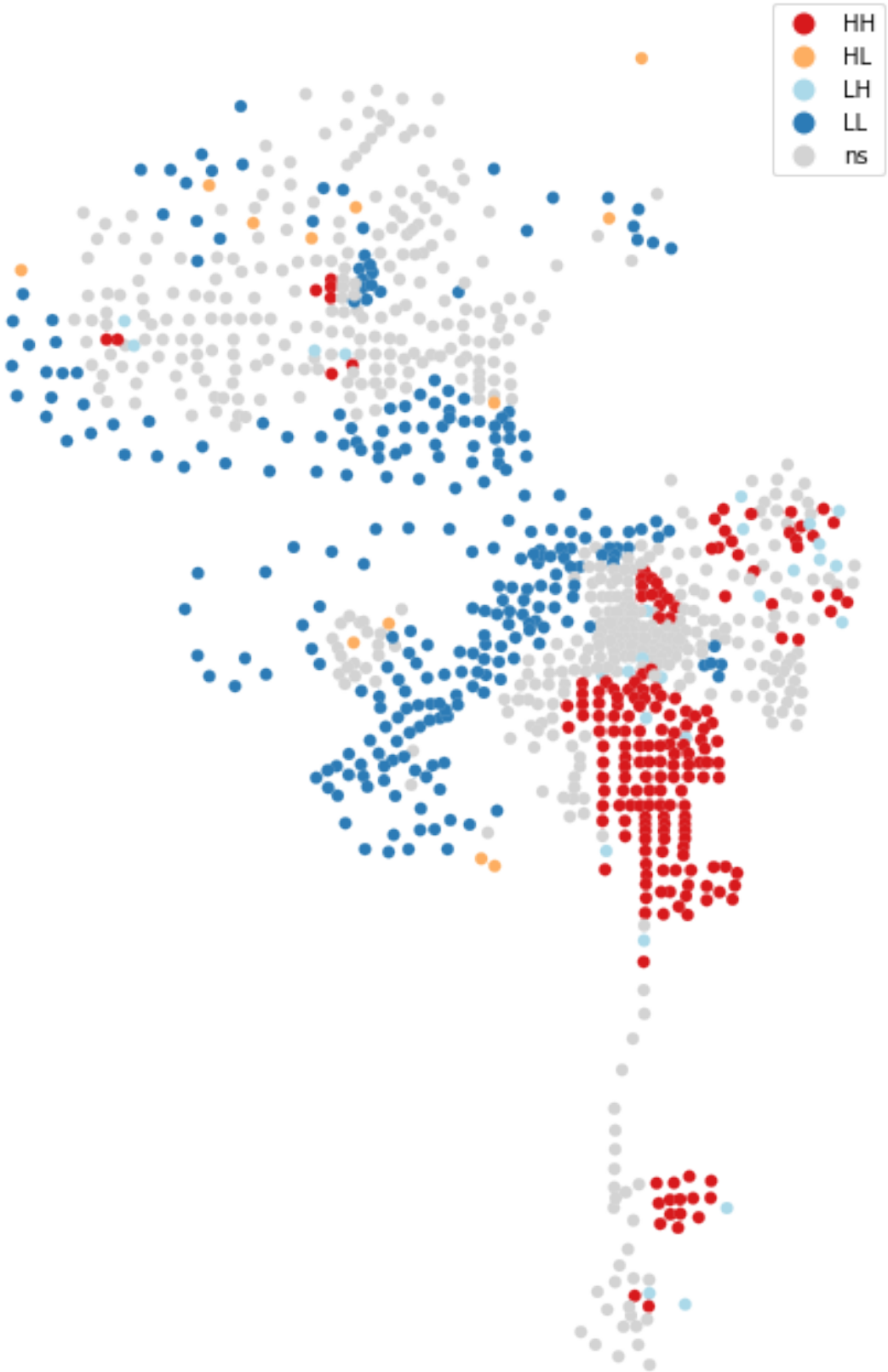
[205]:
```
# calculate local moran values
lisa = esda.moran.Moran_Local(y, wq)
```

[206]:
```
# Plot
fig,ax = plt.subplots(figsize=(20,15))

moran_scatterplot(lisa, ax=ax, p=0.05)
ax.set_xlabel("Mobile Access Only")
ax.set_ylabel('Spatial Lag of Mobile Access')

# add some labels
plt.text(4, 1.5, "HH", fontsize=25)
plt.text(4, -1, "HL", fontsize=25)
```

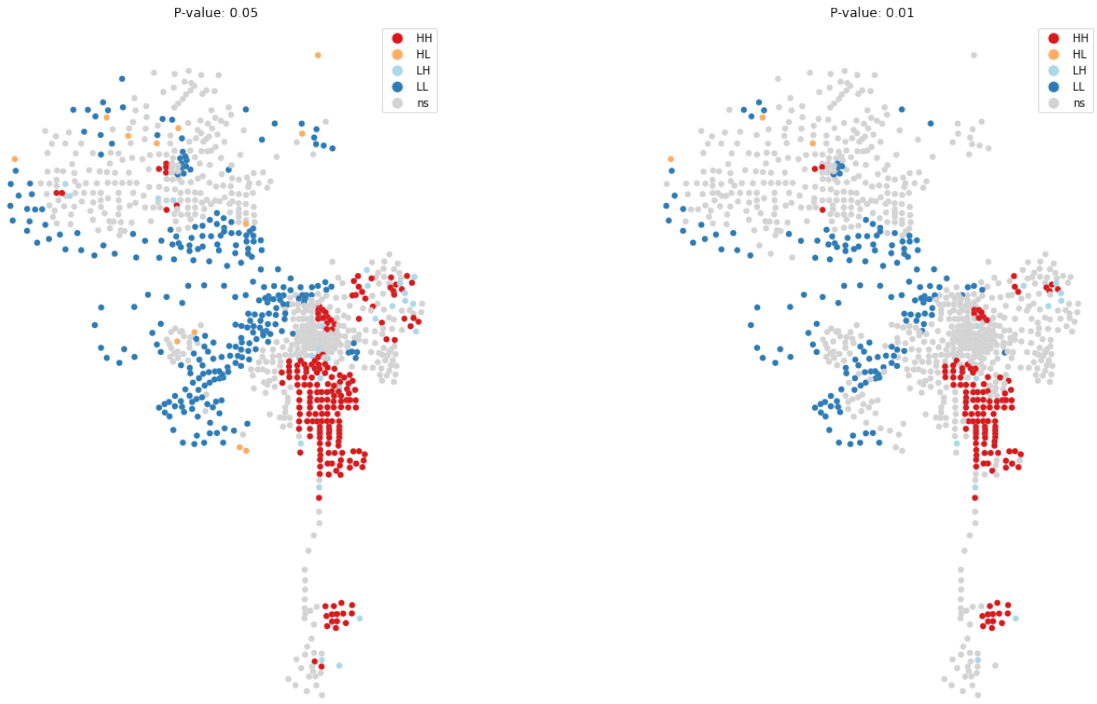
```
plt.text(-2, 1.5, "LH", fontsize=25)
plt.text(-2, -1, "LL", fontsize=25)
plt.show()
```



[207]:
```
fig, ax = plt.subplots(figsize=(14,12))
lisa_cluster(lisa, hh, p=0.05, ax=ax)
plt.show()
```


HH
HL
LH
LL
ns

[208]:
```
# create the 1x2 subplots
fig, ax = plt.subplots(1, 2, figsize=(20, 12))

# regular count map on the left
lisa_cluster(lisa, hh, p=0.05, ax=ax[0])

ax[0].axis("off")
ax[0].set_title("P-value: 0.05")

# spatial lag map on the right
lisa_cluster(lisa, hh, p=0.01, ax=ax[1])
ax[1].axis("off")
ax[1].set_title("P-value: 0.01")

plt.show()
```



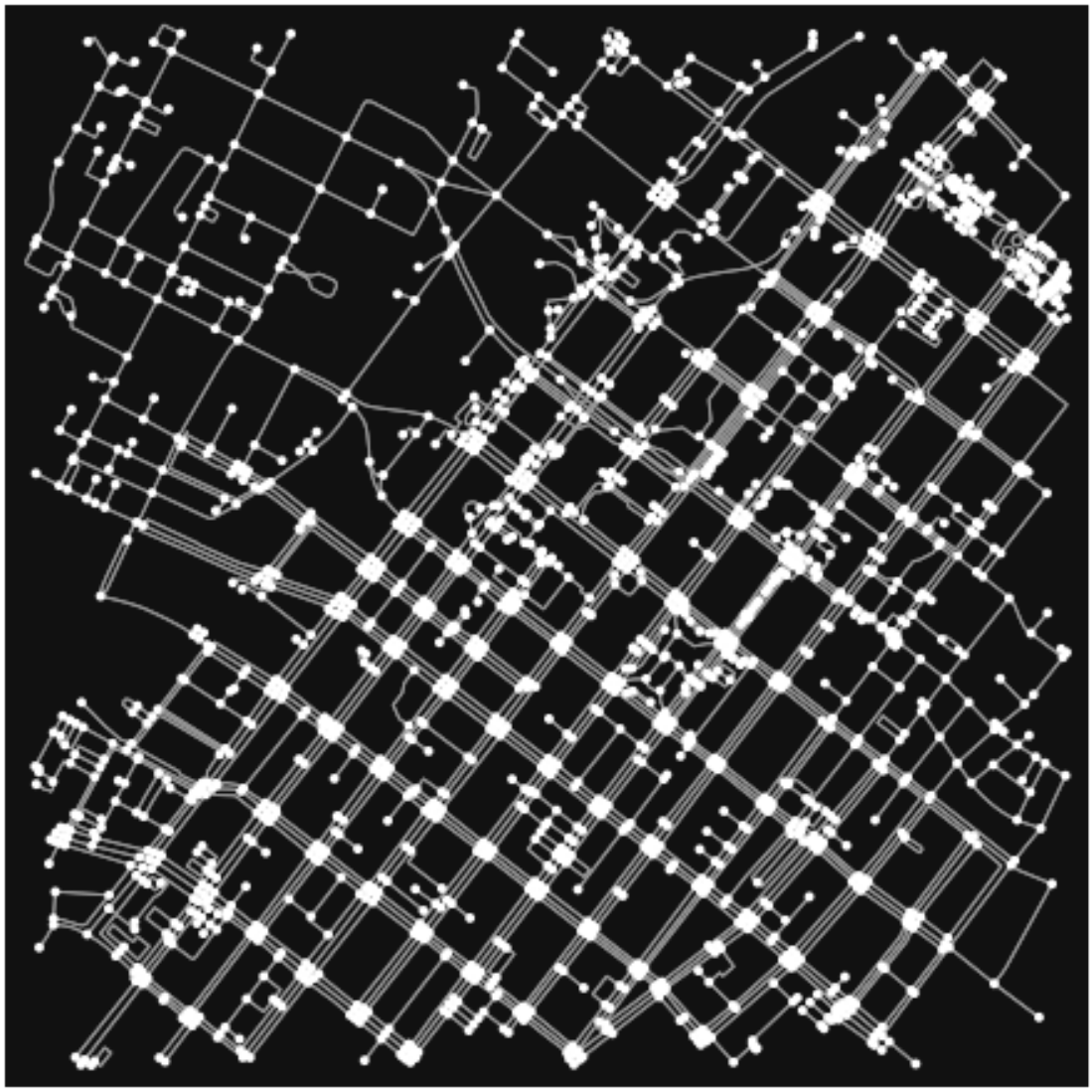
1.5.1 Open Street Maps

[209]:
```
#I will do a trial of the isochrom map from the LA Central library Branch
#first, configure that place, network type, trip times and truael speed for a␣
 ↪walking map
center_point = (34.0504,-118.2555)
network_type = 'walk'
trip_times = [5, 15, 30]
meters_per_minute = 75
```

[210]:
```
%%time
G = ox.graph_from_point(center_point, network_type=network_type)
```

```
CPU times: user 3.02 s, sys: 77.8 ms, total: 3.09 s
Wall time: 3.25 s
```

[211]:
```
fig, ax = ox.plot_graph(G)
```



[212]:
```
G = ox.project_graph(G, to_crs='epsg:3857')
```

[213]:
```
#convert nodes and edges to geodataframes
gdf_nodes, gdf_edges = ox.graph_to_gdfs(G)
```

[214]:
```
gdf_nodes.sample(10)
```

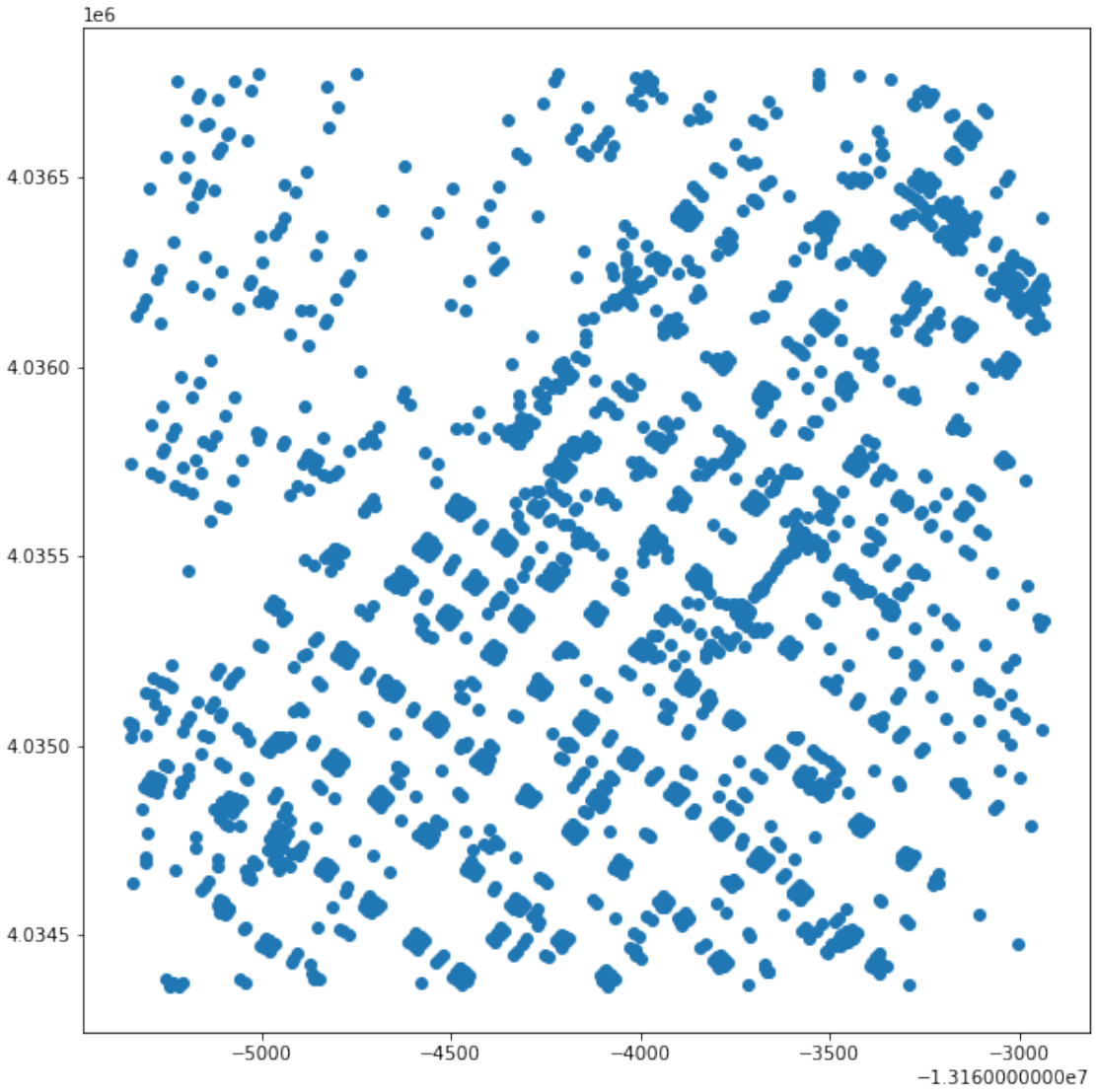
[214]:

osmid	y	x	highway	street_count	lon	\
3663669105	4.035364e+06	-1.316374e+07	NaN	4	-118.251914	
3898644373	4.036393e+06	-1.316318e+07	NaN	3	-118.246866	
72017905	4.035068e+06	-1.316376e+07	NaN	4	-118.252042	
7396420742	4.035421e+06	-1.316298e+07	NaN	1	-118.245074	

```
5336791438  4.035061e+06 -1.316375e+07  crossing            4 -118.251961
2820039363  4.034974e+06 -1.316362e+07       NaN            4 -118.250848
7618992483  4.035077e+06 -1.316473e+07       NaN            4 -118.260781
2889872749  4.035264e+06 -1.316478e+07  crossing            3 -118.261248
4190905966  4.036442e+06 -1.316370e+07       NaN            3 -118.251545
3259253740  4.036027e+06 -1.316340e+07  crossing            4 -118.248822

                  lat                          geometry
osmid
3663669105  34.048857  POINT (-13163742.796 4035364.189)
3898644373  34.056512  POINT (-13163180.900 4036392.747)
72017905    34.046655  POINT (-13163757.123 4035068.348)
7396420742  34.049281  POINT (-13162981.382 4035421.168)
5336791438  34.046602  POINT (-13163748.084 4035061.200)
2820039363  34.045956  POINT (-13163624.152 4034974.438)
7618992483  34.046721  POINT (-13164729.911 4035077.201)
2889872749  34.048114  POINT (-13164781.952 4035264.432)
4190905966  34.056877  POINT (-13163701.797 4036441.750)
3259253740  34.053791  POINT (-13163398.663 4036027.092)
```

[215]:
```
gdf_nodes.plot (figsize=(10,10))
```

[215]:
```
<AxesSubplot:>
```



[216]:
```
stats = ox.basic_stats(G, circuity_dist='euclidean')
stats
```

```
/opt/conda/lib/python3.9/site-packages/osmnx/stats.py:317: UserWarning:

The `circuity_dist` argument has been deprecated and will be removed in a future
release.
```

[216]:
```
{'n': 2076,
 'm': 6562,
 'k_avg': 6.321772639691715,
 'edge_length_total': 277308.9640000002,
```

```
 'edge_length_avg': 42.25982383419692,
 'streets_per_node_avg': 3.226878612716763,
 'streets_per_node_counts': {0: 0, 1: 277, 2: 0, 3: 790, 4: 994, 5: 14, 6: 1},
 'streets_per_node_proportions': {0: 0.0,
  1: 0.13342967244701348,
  2: 0.0,
  3: 0.3805394990366089,
  4: 0.47880539499036606,
  5: 0.00674373795761079,
  6: 0.0004816955684007707},
 'intersection_count': 1799,
 'street_length_total': 138654.48199999979,
 'street_segment_count': 3281,
 'street_length_avg': 42.25982383419682,
 'circuity_avg': 0.8638343149246476,
 'self_loop_proportion': 0.001828710758914965}
```

[217]:
```
#get the bounding box coordinates
minx, miny, maxx, maxy = gdf_nodes.geometry.total_bounds
print(minx)
print(miny)
print(maxx)
print(maxy)
```

```
-13165349.225457974
4034364.472519913
-13162935.184376476
4036774.739681407
```

[218]:
```
#calculate the centroid
centroid_x = (maxx-minx)/2 + minx
centroid_y = (maxy-miny)/2 + miny
print(centroid_x)
print(centroid_y)
```

```
-13164142.204917226
4035569.60610066
```

[219]:
```
#find the nearest node
center_node = ox.distance.nearest_nodes(G,Y=centroid_y,X=centroid_x)
print('The id for the nearest node is ' + str(center_node))
```

```
The id for the nearest node is 4632585779
```

[220]:
```
gdf_nodes.loc[[center_node]]
```

[220]:
```
                        y             x highway  street_count         lon  \
osmid
4632585779  4.035566e+06 -1.316416e+07     NaN             1 -118.255681

                   lat                            geometry
osmid
4632585779  34.050356  POINT (-13164162.181 4035565.586)
```

[221]:
```
#Begin mapping the network layers
fig, ax = plt.subplots (figsize=(20,20))

gdf_edges.plot(ax=ax,
               linewidth=0.5,
               edgecolor='gainsboro',
               zorder=10)

gdf_nodes.plot(ax=ax,
               markersize=2,
               zorder=20)

gdf_nodes.loc[[center_node]].plot(ax=ax,
                                  color = 'r',
                                  zorder=30)

ax.axis('off')

ctx.add_basemap(ax, source=ctx.providers.Stamen.Toner)
```

[222]:
```
#start calculating the travel cost to build the isochrome
gdf_edges[['osmid','name','highway','length']].sample(5)
```

[222]:

u	v	key	osmid	name	highway	length
5336777281	69070509	0	405162741	West 7th Street	primary	
25062101	122718690	0	[405162737, 234662681]	West 3rd Street	primary	
3664674404	2814375263	0	333204228	NaN	footway	
6785700102	122718687	0	13303219	Harlem Place	service	
6787478803	6787478811	0	255621801	NaN	service	

```
5336777281 69070509   0     14.208
25062101   122718690  0     66.005
3664674404 2814375263 0     11.797
6785700102 122718687  0    146.385
6787478803 6787478811 0     19.297
```

[223]:
```
gdf_edges['walk_time'] = gdf_edges['length']/meters_per_minute
```

[224]:
```
gdf_edges[['osmid','name','highway','length','walk_time']].sample(10)
```

[224]:
```
                                          osmid                 name  \
u          v          key
2871078051 2820039396 0                277498186                  NaN
6784922388 5690135237 0                283199649                  NaN
6784768487 3663669113 0                382219102      West 4th Street
7570719458 7570719456 0    [809713229, 809713230]                  NaN
122855842  3663662940 0                399311989  South Flower Street
3898603016 2817343438 0                277204564                  NaN
3663669095 2814313193 0                276874706                  NaN
3660374237 2820039394 0                277498493                  NaN
2820039384 3660374243 0                277498496                  NaN
3698058102 4467499781 0                712563330                  NaN

                             highway   length  walk_time
u          v          key
2871078051 2820039396 0      footway   39.394   0.525253
6784922388 5690135237 0      footway  182.915   2.438867
6784768487 3663669113 0      primary   54.941   0.732547
7570719458 7570719456 0      service   20.523   0.273640
122855842  3663662940 0    secondary   11.009   0.146787
3898603016 2817343438 0      footway   86.646   1.155280
3663669095 2814313193 0      footway    9.995   0.133267
3660374237 2820039394 0      footway    8.394   0.111920
2820039384 3660374243 0      footway   11.036   0.147147
3698058102 4467499781 0      footway   16.539   0.220520
```

[225]:
```
#begin to visualize the isochrone
iso_colors = ox.plot.get_colors(n=len(trip_times),
                                cmap='plasma',
                                start=0,
                                return_hex=True)
print(trip_times)
print(iso_colors)
```

```
[5, 15, 30]
['#0d0887', '#cc4778', '#f0f921']
```

[226]:
```
trip_times.sort(reverse=True)
print(trip_times)
print(iso_colors)
```

```
[30, 15, 5]
['#0d0887', '#cc4778', '#f0f921']
```

[227]:
```
time_color = list(zip(trip_times, iso_colors))
time_color
```

[227]:
```
[(30, '#0d0887'), (15, '#cc4778'), (5, '#f0f921')]
```

[228]:
```
#Let's loop!
for time,color in list(time_color):
    print('The color for '+str(time)+' minutes is ' + color)
```

```
The color for 30 minutes is #0d0887
The color for 15 minutes is #cc4778
The color for 5 minutes is #f0f921
```

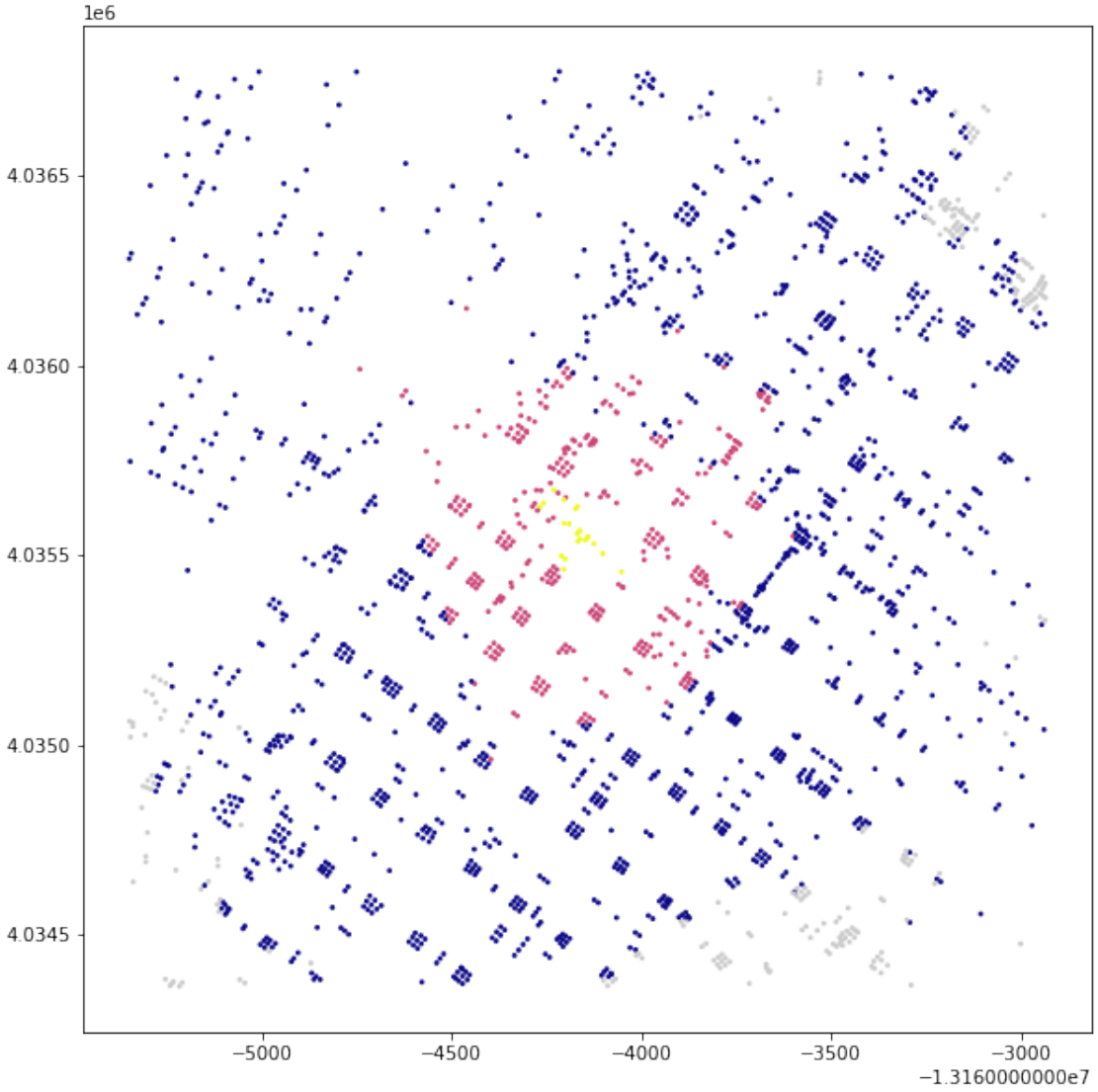
[229]:
```
#loop through each trip time and associated color
for time, color in list(time_color):

    # for each trip time, create an egograph of nodes that fall within that␣
↪distance
    subgraph = nx.ego_graph(G, center_node, radius=time, distance='time')

    print('There are ' + str(len(subgraph.nodes())) + ' nodes within ' +␣
↪str(time) + ' minutes ')

    # for each of those nodes, update the gdf_nodes dataframe and assign it␣
↪with its associated distance color
    for node in subgraph.nodes():
        gdf_nodes.loc[node,'time'] = str(time) + ' mins'
        gdf_nodes.loc[node,'color'] = color
```

```
There are 1838 nodes within 30 minutes
There are 385 nodes within 15 minutes
There are 22 nodes within 5 minutes
```

[230]:
```
gdf_nodes['color'].fillna('#cccccc', inplace=True)
```

[231]:
```
#mapping!
gdf_nodes.plot(figsize=(10,10),
               color=gdf_nodes['color'],
               markersize=3)
```

[231]: <AxesSubplot:>



[232]:
```
#Now I will being creating polygons for each time zone
gdf_5 = gdf_nodes[gdf_nodes['time']=='5 mins']
```

[233]:
```
# dissolve the nodes by time
isochrones = gdf_nodes.dissolve("time")
isochrones
```

[233]:

time	geometry	y	\
15 mins	MULTIPOINT (-13164741.822 4035990.613, -131646…	4.035526e+06	
30 mins	MULTIPOINT (-13165349.225 4036281.251, -131653…	4.036374e+06	

```
5 mins   MULTIPOINT (-13164267.356 4035626.193, -131642…  4.035463e+06

                    x          highway  street_count         lon        lat  \
time
15 mins -1.316456e+07  traffic_signals             4 -118.259253  34.050061
30 mins -1.316351e+07  traffic_signals             4 -118.249821  34.056371
5 mins  -1.316420e+07             None             3 -118.256057  34.049596

           color
time
15 mins  #cc4778
30 mins  #0d0887
5 mins   #f0f921
```

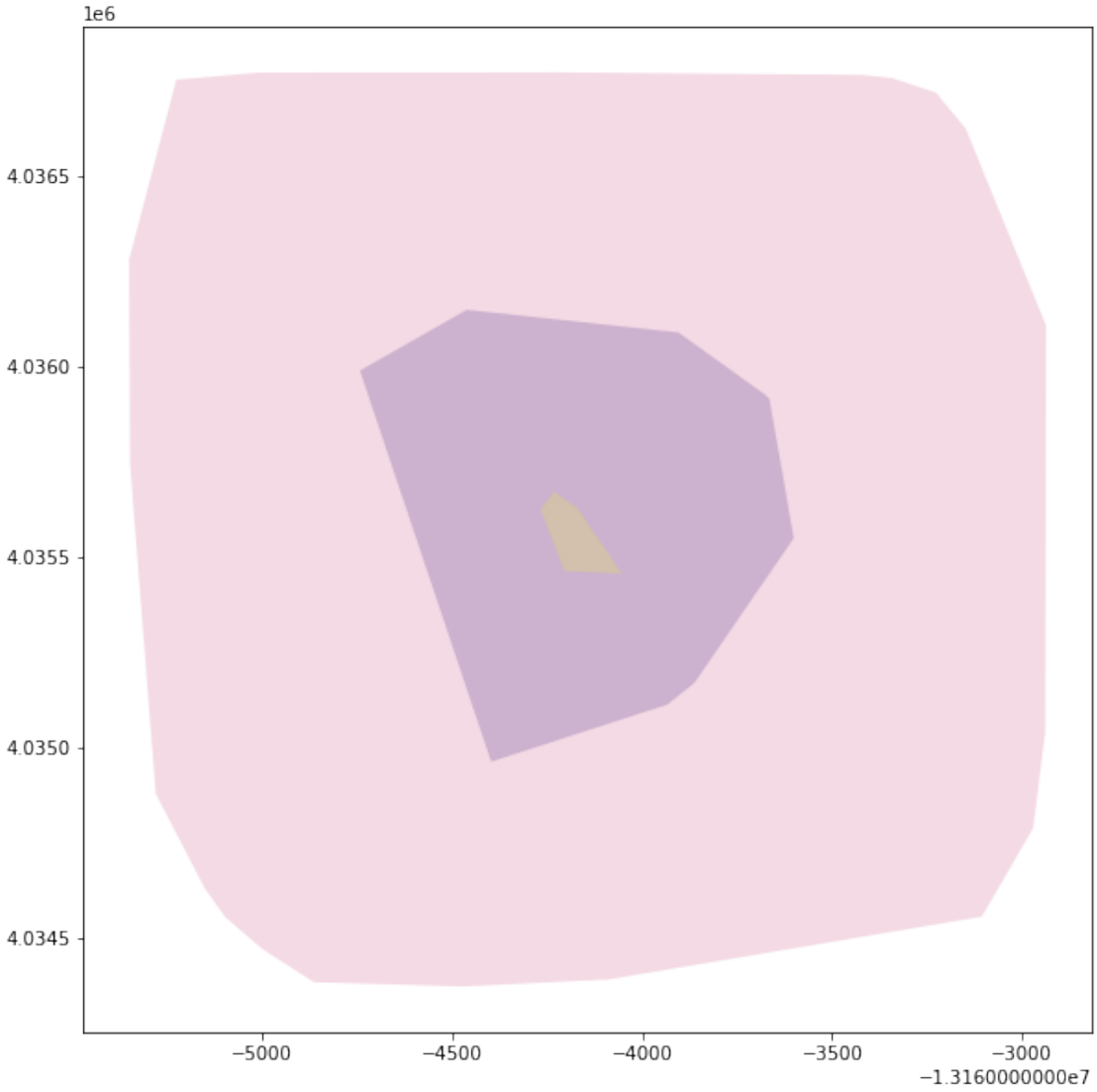
[234]:
```
isochrones = isochrones.convex_hull.reset_index()
isochrones
```

[234]:
```
      time                                                  0
0  15 mins  POLYGON ((-13164396.509 4034962.440, -13164741…
1  30 mins  POLYGON ((-13164470.970 4034371.458, -13164862…
2   5 mins  POLYGON ((-13164052.487 4035456.369, -13164204…
```

[235]:
```
isochrones.columns=['time','geometry']
```

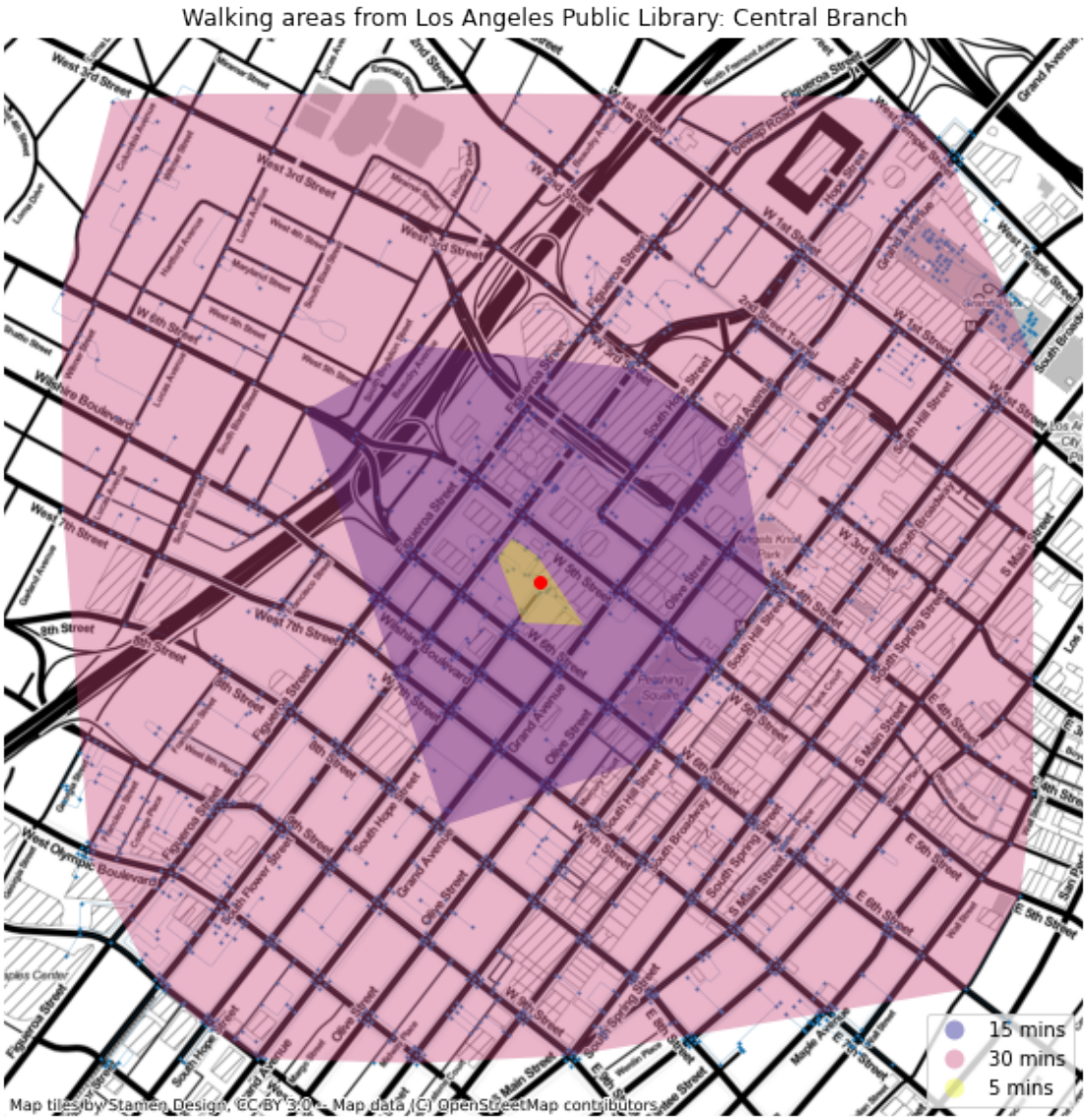
[236]:
```
isochrones.plot(figsize=(10,10),alpha=0.2,cmap='plasma')
```

[236]:
```
<AxesSubplot:>
```



[237]:
```
fig, ax = plt.subplots(figsize=(10,15))

isochrones.plot(alpha=0.4,
                ax=ax,
                column='time',
                cmap='plasma',
                legend=True,
                zorder=20)

gdf_nodes.loc[[center_node]].plot(ax=ax,color='r', zorder=30)

gdf_nodes.plot(ax=ax,
               markersize=1,
```

```
                zorder=10)

gdf_edges.plot(ax=ax,
                linewidth=0.5,
                alpha=0.2,
                zorder=10)

ax.axis('off')


ax.set_title('Walking areas from Los Angeles Public Library: Central Branch')

ctx.add_basemap(ax, source=ctx.providers.Stamen.Toner)
```



[238]:
```
#Now I will make a isochrone map for biking to the library
center_point = (34.0504,-118.2555)
network_type = 'bike'
trip_times = [15, 30]
meters_per_minute = 200
```

[239]:
```
center_node = ox.distance.nearest_nodes(G,Y=centroid_y,X=centroid_x)
print('The id for the nearest node is ' + str(center_node))
```

```
The id for the nearest node is 4632585779
```

[240]:
```
gdf_edges['bike_time'] = gdf_edges['length']/meters_per_minute
```

[241]:
```
gdf_edges[['osmid','name','highway','length','walk_time']].sample(10)
```

[241]:

u	v	key	osmid	name	highway	length	walk_time
3308560564	3308560563	0	324103016	NaN	footway	9.583	0.127773
6699521846	6352157027	0	678394585	NaN	service	13.453	0.179373
8376376547	8376376551	0	901727164	NaN	service	26.268	0.350240
2814324716	2814324717	0	276866167	NaN	footway	102.991	1.373213
3698058102	268642905	0	399312999	North Grand Avenue	primary	16.715	0.222867
7187928357	3308914962	0	161708071	Beaudry Avenue	tertiary	26.409	0.352120
6731580980	6731580973	1	716663900	NaN	service	75.567	1.007560
2814383557	3663827018	0	283202901	NaN	footway	13.345	0.177933
8656138870	8656138869	0	933939611	NaN	footway	5.292	0.070560
2814318214	3663676111	0	276862849	NaN	footway	132.435	1.765800

[242]:
```
#begin to visualize the isochrone
iso_colors = ox.plot.get_colors(n=len(trip_times),
                                cmap='cool',
                                start=0,
                                return_hex=True)
```

```
print(trip_times)
print(iso_colors)
```

```
[15, 30]
['#00ffff', '#ff00ff']
```

[243]:
```
trip_times.sort(reverse=True)
print(trip_times)
print(iso_colors)
```

```
[30, 15]
['#00ffff', '#ff00ff']
```

[244]:
```
time_color = list(zip(trip_times, iso_colors))
time_color
```

[244]:
```
[(30, '#00ffff'), (15, '#ff00ff')]
```

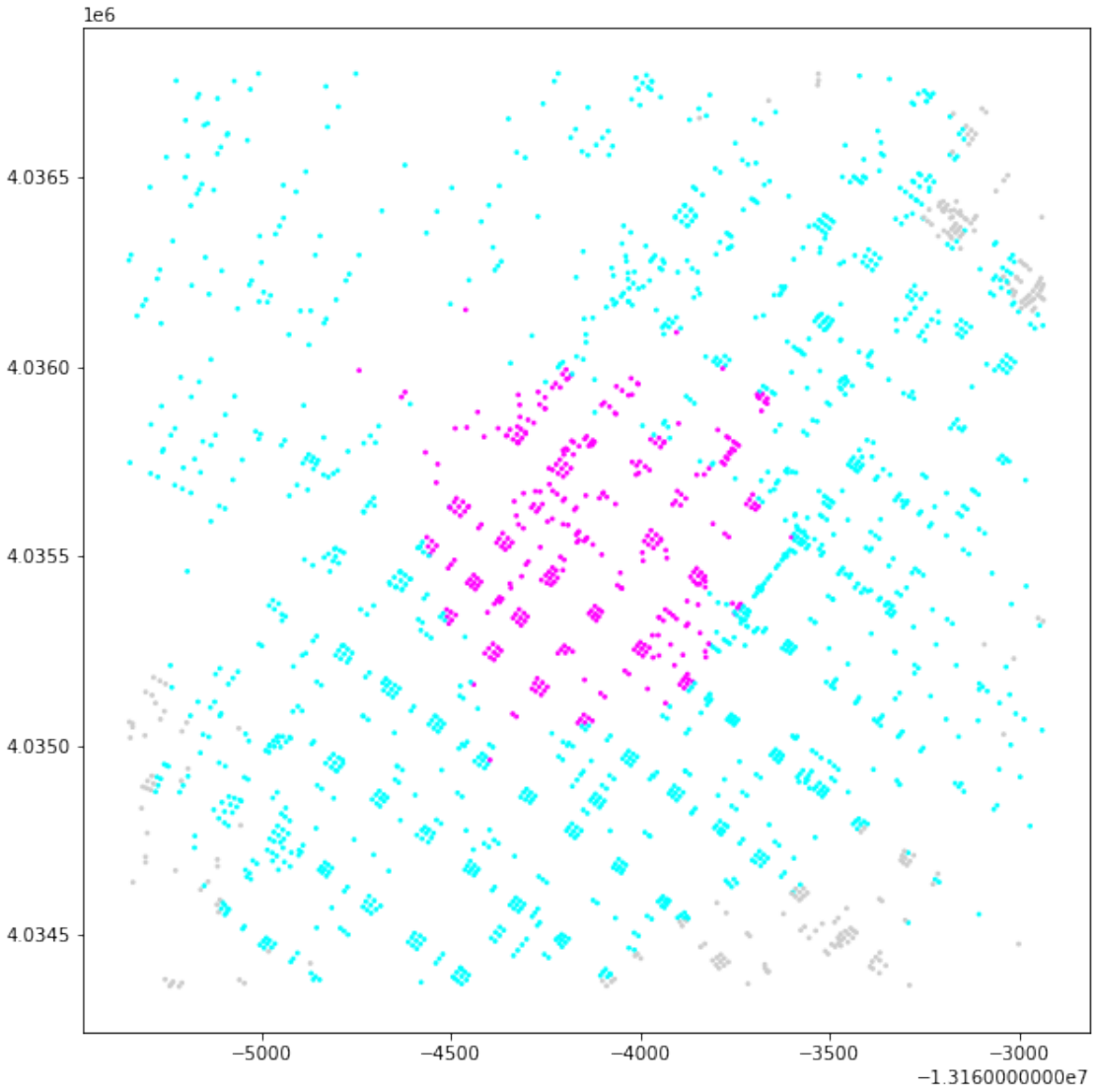
[245]:
```
#loop through each trip time and associated color
for time, color in list(time_color):

    # for each trip time, create an egograph of nodes that fall within that␣
↪distance
    subgraph = nx.ego_graph(G, center_node, radius=time, distance='time')

    print('There are ' + str(len(subgraph.nodes())) + ' nodes within ' +␣
↪str(time) + ' minutes ')

    # for each of those nodes, update the gdf_nodes dataframe and assign it␣
↪with its associated distance color
    for node in subgraph.nodes():
        gdf_nodes.loc[node,'time'] = str(time) + ' mins'
        gdf_nodes.loc[node,'color'] = color
```

```
There are 1838 nodes within 30 minutes
There are 385 nodes within 15 minutes
```

[246]:
```
gdf_nodes['color'].fillna('#cccccc', inplace=True)
```

[247]:
```
#mapping!
gdf_nodes.plot(figsize=(10,10),
               color=gdf_nodes['color'],
               markersize=3)
```

[247]:
```
<AxesSubplot:>
```



[248]:
```
#Now I will being creating polygons for each time zone
gdf_15 = gdf_nodes[gdf_nodes['time']=='15 mins']
```

[249]:
```
# dissolve the nodes by time
isochrones = gdf_nodes.dissolve("time")
isochrones
```

[249]:
```
                                                  geometry             y  \
time
15 mins  MULTIPOINT (-13164741.822 4035990.613, -131646…  4.035526e+06
30 mins  MULTIPOINT (-13165349.225 4036281.251, -131653…  4.036374e+06

                     x          highway  street_count         lon        lat  \
```

```
time
15 mins -1.316456e+07  traffic_signals             4 -118.259253  34.050061
30 mins -1.316351e+07  traffic_signals             4 -118.249821  34.056371

           color
time
15 mins  #ff00ff
30 mins  #00ffff
```

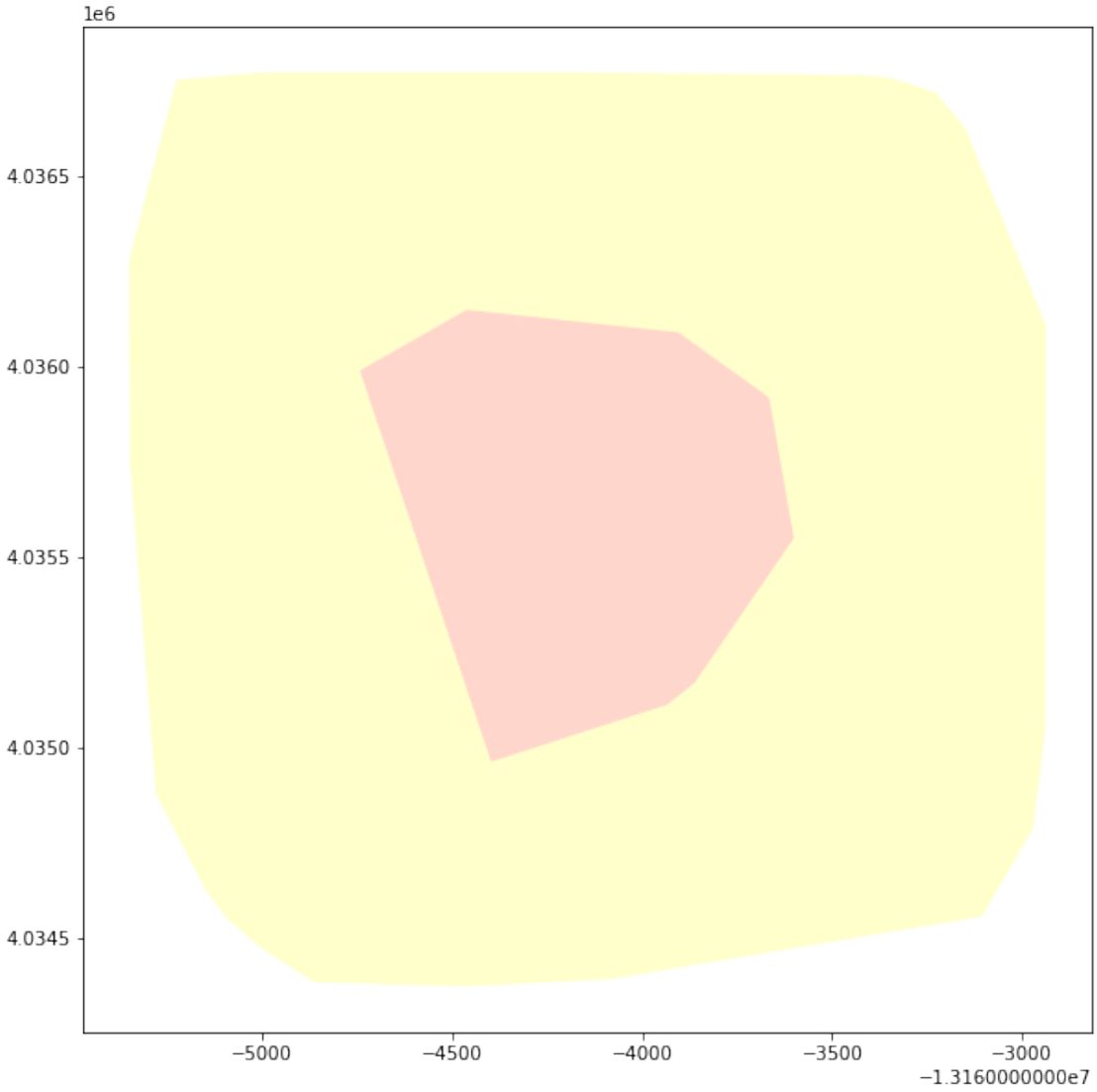
[250]:
```
isochrones = isochrones.convex_hull.reset_index()
isochrones
```

[250]:
```
      time                                                  0
0  15 mins  POLYGON ((-13164396.509 4034962.440, -13164741…
1  30 mins  POLYGON ((-13164470.970 4034371.458, -13164862…
```

[251]:
```
isochrones.columns=['time','geometry']
```

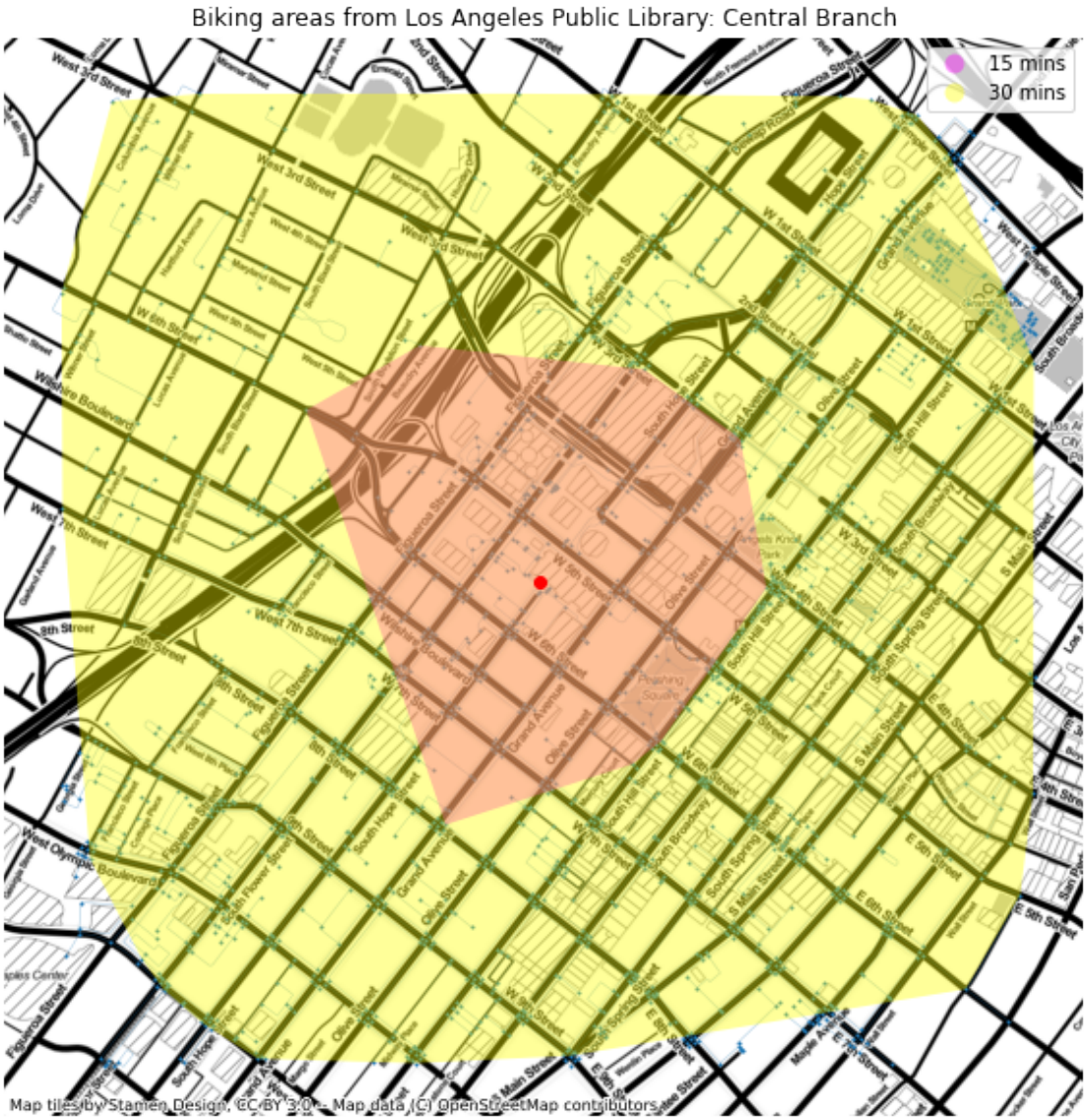
[252]:
```
isochrones.plot(figsize=(10,10),alpha=0.2,cmap='spring')
```

[252]:
```
<AxesSubplot:>
```



[253]:
```
ig, ax = plt.subplots(figsize=(10,15))

isochrones.plot(alpha=0.4,
                ax=ax,
                column='time',
                cmap='spring',
                legend=True,
                zorder=20)

gdf_nodes.loc[[center_node]].plot(ax=ax,color='r', zorder=30)

gdf_nodes.plot(ax=ax,
               markersize=1,
```

```
                zorder=10)

gdf_edges.plot(ax=ax,
                linewidth=0.5,
                alpha=0.2,
                zorder=10)

ax.axis('off')


ax.set_title('Biking areas from Los Angeles Public Library: Central Branch')

ctx.add_basemap(ax, source=ctx.providers.Stamen.Toner)
```



Michael Criste: Development & Analysis