CS-GY 6313 / CUSP-GX 6006: Data Visualization - Spring '24

Homework #2

In [1]:

```
!pip install pandas
!pip install geopandas
!pip install geoplot
!pip install pyogrio
```

```
Requirement already satisfied: pandas in /Users/fengcharles/anaconda3/lib/pyth
on3.11/site-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in /Users/fengcharles/an
aconda3/lib/python3.11/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /Users/fengcharles/anaconda3/li
b/python3.11/site-packages (from pandas) (2022.7)
Requirement already satisfied: numpy>=1.21.0 in /Users/fengcharles/anaconda3/l
ib/python3.11/site-packages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in /Users/fengcharles/anaconda3/lib/py
thon3.11/site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Requirement already satisfied: geopandas in /Users/fengcharles/anaconda3/lib/p
ython3.11/site-packages (0.14.1)
Requirement already satisfied: fiona>=1.8.21 in /Users/fengcharles/anaconda3/l
ib/python3.11/site-packages (from geopandas) (1.9.4.post1)
Requirement already satisfied: packaging in /Users/fengcharles/anaconda3/lib/p
ython3.11/site-packages (from geopandas) (23.0)
Requirement already satisfied: pandas>=1.4.0 in /Users/fengcharles/anaconda3/l
ib/python3.11/site-packages (from geopandas) (1.5.3)
Requirement already satisfied: pyproj>=3.3.0 in /Users/fengcharles/anaconda3/l
ib/python3.11/site-packages (from geopandas) (3.6.0)
Requirement already satisfied: shapely>=1.8.0 in /Users/fengcharles/anaconda3/
lib/python3.11/site-packages (from geopandas) (2.0.1)
Requirement already satisfied: attrs>=19.2.0 in /Users/fengcharles/anaconda3/l
ib/python3.11/site-packages (from fiona>=1.8.21->geopandas) (22.1.0)
Requirement already satisfied: certifi in /Users/fengcharles/anaconda3/lib/pyt
hon3.11/site-packages (from fiona>=1.8.21->geopandas) (2024.2.2)
Requirement already satisfied: click~=8.0 in /Users/fengcharles/anaconda3/lib/
python3.11/site-packages (from fiona>=1.8.21->geopandas) (8.0.4)
Requirement already satisfied: click-plugins>=1.0 in /Users/fengcharles/anacon
da3/lib/python3.11/site-packages (from fiona>=1.8.21->geopandas) (1.1.1)
Requirement already satisfied: cligj>=0.5 in /Users/fengcharles/anaconda3/lib/
python3.11/site-packages (from fiona>=1.8.21->geopandas) (0.7.2)
Requirement already satisfied: six in /Users/fengcharles/anaconda3/lib/python
3.11/site-packages (from fiona>=1.8.21->geopandas) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /Users/fengcharles/an
aconda3/lib/python3.11/site-packages (from pandas>=1.4.0->geopandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /Users/fengcharles/anaconda3/li
b/python3.11/site-packages (from pandas>=1.4.0->geopandas) (2022.7)
Requirement already satisfied: numpy>=1.21.0 in /Users/fengcharles/anaconda3/l
ib/python3.11/site-packages (from pandas>=1.4.0->geopandas) (1.24.3)
Requirement already satisfied: geoplot in /Users/fengcharles/anaconda3/lib/pyt
hon3.11/site-packages (0.5.1)
Requirement already satisfied: matplotlib>=3.1.2 in /Users/fengcharles/anacond
a3/lib/python3.11/site-packages (from geoplot) (3.7.1)
Requirement already satisfied: seaborn in /Users/fengcharles/anaconda3/lib/pyt
hon3.11/site-packages (from geoplot) (0.12.2)
Requirement already satisfied: pandas in /Users/fengcharles/anaconda3/lib/pyth
on3.11/site-packages (from geoplot) (1.5.3)
Requirement already satisfied: geopandas>=0.9.0 in /Users/fengcharles/anaconda
3/lib/python3.11/site-packages (from geoplot) (0.14.1)
Requirement already satisfied: cartopy in /Users/fengcharles/anaconda3/lib/pyt
hon3.11/site-packages (from geoplot) (0.22.0)
Requirement already satisfied: mapclassify>=2.1 in /Users/fengcharles/anaconda
3/lib/python3.11/site-packages (from geoplot) (2.6.0)
Requirement already satisfied: contextily>=1.0.0 in /Users/fengcharles/anacond
a3/lib/python3.11/site-packages (from geoplot) (1.6.0)
Requirement already satisfied: geopy in /Users/fengcharles/anaconda3/lib/pytho
n3.11/site-packages (from contextily>=1.0.0->geoplot) (2.4.1)
Requirement already satisfied: mercantile in /Users/fengcharles/anaconda3/lib/
python3.11/site-packages (from contextily>=1.0.0->geoplot) (1.2.1)
```

```
Requirement already satisfied: pillow in /Users/fengcharles/anaconda3/lib/pyth
on3.11/site-packages (from contextily>=1.0.0->geoplot) (9.4.0)
Requirement already satisfied: rasterio in /Users/fengcharles/anaconda3/lib/py
thon3.11/site-packages (from contextily>=1.0.0->geoplot) (1.3.9)
Requirement already satisfied: requests in /Users/fengcharles/anaconda3/lib/py
thon3.11/site-packages (from contextily>=1.0.0->geoplot) (2.31.0)
Requirement already satisfied: joblib in /Users/fengcharles/anaconda3/lib/pyth
on3.11/site-packages (from contextily>=1.0.0->geoplot) (1.2.0)
Requirement already satisfied: xyzservices in /Users/fengcharles/anaconda3/li
b/python3.11/site-packages (from contextily>=1.0.0->geoplot) (2022.9.0)
Requirement already satisfied: fiona>=1.8.21 in /Users/fengcharles/anaconda3/l
ib/python3.11/site-packages (from geopandas>=0.9.0->geoplot) (1.9.4.post1)
Requirement already satisfied: packaging in /Users/fengcharles/anaconda3/lib/p
ython3.11/site-packages (from geopandas>=0.9.0->geoplot) (23.0)
Requirement already satisfied: pyproj>=3.3.0 in /Users/fengcharles/anaconda3/l
ib/python3.11/site-packages (from geopandas>=0.9.0->geoplot) (3.6.0)
Requirement already satisfied: shapely>=1.8.0 in /Users/fengcharles/anaconda3/
lib/python3.11/site-packages (from geopandas>=0.9.0->geoplot) (2.0.1)
Requirement already satisfied: scipy>=1.0 in /Users/fengcharles/anaconda3/lib/
python3.11/site-packages (from mapclassify>=2.1->geoplot) (1.10.1)
Requirement already satisfied: numpy>=1.3 in /Users/fengcharles/anaconda3/lib/
python3.11/site-packages (from mapclassify>=2.1->geoplot) (1.24.3)
Requirement already satisfied: scikit-learn in /Users/fengcharles/anaconda3/li
b/python3.11/site-packages (from mapclassify>=2.1->geoplot) (1.3.0)
Requirement already satisfied: networkx in /Users/fengcharles/anaconda3/lib/py
thon3.11/site-packages (from mapclassify>=2.1->geoplot) (3.1)
Requirement already satisfied: contourpy>=1.0.1 in /Users/fengcharles/anaconda
3/lib/python3.11/site-packages (from matplotlib>=3.1.2->geoplot) (1.0.5)
Requirement already satisfied: cycler>=0.10 in /Users/fengcharles/anaconda3/li
b/python3.11/site-packages (from matplotlib>=3.1.2->geoplot) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /Users/fengcharles/anacond
a3/lib/python3.11/site-packages (from matplotlib>=3.1.2->geoplot) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/fengcharles/anacond
a3/lib/python3.11/site-packages (from matplotlib>=3.1.2->geoplot) (1.4.4)
Requirement already satisfied: pyparsing>=2.3.1 in /Users/fengcharles/anaconda
3/lib/python3.11/site-packages (from matplotlib>=3.1.2->geoplot) (2.4.7)
Requirement already satisfied: python-dateutil>=2.7 in /Users/fengcharles/anac
onda3/lib/python3.11/site-packages (from matplotlib>=3.1.2->geoplot) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /Users/fengcharles/anaconda3/li
b/python3.11/site-packages (from pandas->geoplot) (2022.7)
Requirement already satisfied: pyshp>=2.1 in /Users/fengcharles/anaconda3/lib/
python3.11/site-packages (from cartopy->geoplot) (2.3.1)
Requirement already satisfied: attrs>=19.2.0 in /Users/fengcharles/anaconda3/l
ib/python3.11/site-packages (from fiona>=1.8.21->geopandas>=0.9.0->geoplot) (2
2.1.0)
Requirement already satisfied: certifi in /Users/fengcharles/anaconda3/lib/pyt
hon3.11/site-packages (from fiona>=1.8.21->geopandas>=0.9.0->geoplot) (2024.2.
2)
Requirement already satisfied: click~=8.0 in /Users/fengcharles/anaconda3/lib/
python3.11/site-packages (from fiona>=1.8.21->geopandas>=0.9.0->geoplot) (8.0.
4)
Requirement already satisfied: click-plugins>=1.0 in /Users/fengcharles/anacon
da3/lib/python3.11/site-packages (from fiona>=1.8.21->geopandas>=0.9.0->geoplo
t) (1.1.1)
Requirement already satisfied: cligj>=0.5 in /Users/fengcharles/anaconda3/lib/
python3.11/site-packages (from fiona>=1.8.21->geopandas>=0.9.0->geoplot) (0.7.
2)
Requirement already satisfied: six in /Users/fengcharles/anaconda3/lib/python
3.11/site-packages (from fiona>=1.8.21->geopandas>=0.9.0->geoplot) (1.16.0)
Requirement already satisfied: geographiclib<3,>=1.52 in /Users/fengcharles/an
```

```
aconda3/lib/python3.11/site-packages (from geopy->contextily>=1.0.0->geoplot)
(2.0)
Requirement already satisfied: affine in /Users/fengcharles/anaconda3/lib/pyth
on3.11/site-packages (from rasterio->contextily>=1.0.0->geoplot) (2.4.0)
Requirement already satisfied: snuggs>=1.4.1 in /Users/fengcharles/anaconda3/l
ib/python3.11/site-packages (from rasterio->contextily>=1.0.0->geoplot) (1.4.
7)
Requirement already satisfied: setuptools in /Users/fengcharles/anaconda3/lib/
python3.11/site-packages (from rasterio->contextily>=1.0.0->geoplot) (68.0.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /Users/fengcharles/
anaconda3/lib/python3.11/site-packages (from requests->contextily>=1.0.0->geop
lot) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /Users/fengcharles/anaconda3/li
b/python3.11/site-packages (from requests->contextily>=1.0.0->geoplot) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Users/fengcharles/anacon
da3/lib/python3.11/site-packages (from requests->contextily>=1.0.0->geoplot)
(1.26.16)
Requirement already satisfied: threadpoolctl>=2.0.0 in /Users/fengcharles/anac
onda3/lib/python3.11/site-packages (from scikit-learn->mapclassify>=2.1->geopl
ot) (2.2.0)
Requirement already satisfied: pyogrio in /Users/fengcharles/anaconda3/lib/pyt
hon3.11/site-packages (0.7.2)
Requirement already satisfied: certifi in /Users/fengcharles/anaconda3/lib/pyt
hon3.11/site-packages (from pyogrio) (2024.2.2)
Requirement already satisfied: numpy in /Users/fengcharles/anaconda3/lib/pytho
n3.11/site-packages (from pyogrio) (1.24.3)
Requirement already satisfied: packaging in /Users/fengcharles/anaconda3/lib/p
ython3.11/site-packages (from pyogrio) (23.0)
```

In [2]:

```python
import pandas as pd
import geopandas as gpd
import geoplot
import matplotlib.pyplot as plt
```

Data Pre-Processing (3/15 points)

In [3]:

```python
# ----------------------- #
# DO NOT MODIFY THIS CODE #
# ----------------------- #

trips_df = pd.read_csv('./datasets/202007-divvy-tripdata.csv')
community_df = gpd.read_file('./datasets/chicago-community-areas.geojson')
stations_df = pd.read_csv('./datasets/station-locations.csv')
```

Bike Trip Pre-processing (1 point)

In [4]:

```python
"""
TODO:
Within the bike trip data that we loaded (`trips_df`), get rid of missing (`Na
start and end station ids, and convert those columns to integer columns.
Make sure the modified dataframe is referenced as `trips_pr_df`.
"""
#Drop NAs
trips_pr_df = trips_df.dropna(subset=['start_station_id','end_station_id'])
#Convert column to integers
trips_pr_df[['start_station_id','end_station_id']] = trips_pr_df[['start_statio
```

```
/var/folders/qp/9y56mfxx3zq2c_cjbvf9xg_w0000gn/T/ipykernel_28575/218670856.py:
10: 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/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  trips_pr_df[['start_station_id','end_station_id']] = trips_pr_df[['start_sta
tion_id','end_station_id']].astype(int)
```

In [5]:
```
trips_pr_df.head()
```

Out[5]:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_i
0	762198876D69004D	docked_bike	2020-07-09 15:22:02	2020-07-09 15:25:52	Ritchie Ct & Banks St	18
1	BEC9C9FBA0D4CF1B	docked_bike	2020-07-24 23:56:30	2020-07-25 00:20:17	Halsted St & Roscoe St	29
2	D2FD8EA432C77EC1	docked_bike	2020-07-08 19:49:07	2020-07-08 19:56:22	Lake Shore Dr & Diversey Pkwy	32
3	54AE594E20B35881	docked_bike	2020-07-17 19:06:42	2020-07-17 19:27:38	LaSalle St & Illinois St	18
4	54025FDC7440B56F	docked_bike	2020-07-04 10:39:57	2020-07-04 10:45:05	Lake Shore Dr & North Blvd	26

Community Areas Pre-processing (1 point)

In [6]:
```
"""
TODO:
Within the geojson data for the Chicago community areas (`community_df`), rena
column `area_numbe` to `area_number`, and convert that column to an integer
column. Make sure to reference the modified geojson data as `community_pr`.
"""
#rename the column
community_df.rename(columns={'area_numbe': 'area_number'}, inplace=True)
```

In [7]:
```
#Convert column to intergers
community_df[['area_number']] = community_df[['area_number']].astype(int)
#rename the df
community_pr_df = community_df
```

In [8]:
```
community_pr_df.head()
```

Out[8]:

	community	area	shape_area	perimeter	area_num_1	area_number	comarea_id	comar
0	DOUGLAS	0	46004621.1581	0	35	35	0	
1	OAKLAND	0	16913961.0408	0	36	36	0	
2	FULLER PARK	0	19916704.8692	0	37	37	0	
3	GRAND BOULEVARD	0	48492503.1554	0	38	38	0	
4	KENWOOD	0	29071741.9283	0	39	39	0	

Stations Pre-processing (1 point)

In [9]:

```
import geopandas as gpd
from shapely.geometry import Point
```

In [10]:

```
"""
TODO:
Within the bike station location data (`stations_df`), convert it to a
`GeoDataFrame` and set its geometry to the point specified by the longitude
and latitude pair. Make sure to reference the modified data as
`stations_pr_df`.
"""

from shapely.geometry import Point
#add column 'geometry'
stations_df['geometry'] = stations_df.apply(lambda row: Point(row['lon'], row[
#Convert df to geodataframe
stations_df = gpd.GeoDataFrame(stations_df, geometry='geometry')
#rename the dataframe
stations_pr_df = stations_df
```

In [11]:

```
stations_pr_df.head()
```

Out [11]:

	has_kiosk	lat	lon	external_id	rental_uris	short_
0	True	41.876511	-87.620548	a3a36d9e-a135-11e9-9cda-0a87ae2ba916	{'ios': 'https://chi.lft.to/lastmile_qr_scan',...	
1	True	41.867226	-87.615355	a3a37378-a135-11e9-9cda-0a87ae2ba916	{'ios': 'https://chi.lft.to/lastmile_qr_scan',...	
2	True	41.856268	-87.613348	a3a378ca-a135-11e9-9cda-0a87ae2ba916	{'ios': 'https://chi.lft.to/lastmile_qr_scan',...	
3	True	41.874053	-87.627716	a3a37e26-a135-11e9-9cda-0a87ae2ba916	{'ios': 'https://chi.lft.to/lastmile_qr_scan',...	S
4	True	41.886976	-87.612813	a3a38363-a135-11e9-9cda-0a87ae2ba916	{'ios': 'https://chi.lft.to/lastmile_qr_scan',...	KA15030

5 rows × 22 columns

Geographical Visualization (12/15 points)

Spatial Join (2 points)

In [12]:

```
"""
TODO:
Given points from station locations, we want to find out which
community areas those points are in. This can be accomplished
using an `sjoin` (https://geopandas.org/en/stable/gallery/spatial_joins.html)
in `geopandas`. After joining the two datasets, you should be
able to find the area_number for each `station_id`.
"""
station_community_df = gpd.sjoin(left_df=community_pr_df, right_df=stations_pr_
```

```
/var/folders/qp/9y56mfxx3zq2c_cjbvf9xg_w0000gn/T/ipykernel_28575/4057284314.p
y:9: UserWarning: CRS mismatch between the CRS of left geometries and the CRS
of right geometries.
Use `to_crs()` to reproject one of the input geometries to match the CRS of th
e other.

Left CRS: EPSG:4326
Right CRS: None

  station_community_df = gpd.sjoin(left_df=community_pr_df, right_df=stations_
pr_df)
```

Add Community Areas to Trips (4 points)

In [13]:

```
"""
TODO:
Use the updated dataframe from the previous part with the bike trip dataset to
columns specifying the start and end community area numbers (`start_ca_num` and
`end_ca_num`) for each trip. Remove any entries in your final results that have
`NaN` values for either `start_ca_num` or `end_ca_num`.Save your results in
`trips_community_df`.
"""
# Merge, selecting only the specified columns from the right dataframe
#Merge the START community number to the df
trips_community_df = pd.merge(left=trips_pr_df,
                              right=station_community_df[['area_number','static
                              left_on='start_station_id',
                              right_on='station_id',
                              how='left')
#Rename it to 'start_ca_num'
trips_community_df.rename(columns={'area_number': 'start_ca_num'}, inplace=True
```

In [14]:

```
#Merge the END community number to the df
trips_community_df = pd.merge(left=trips_community_df,
                              right=station_community_df[['area_number','static
                              left_on='end_station_id',
                              right_on='station_id',
                              how='left')
#Rename it to 'end_ca_num'
trips_community_df.rename(columns={'area_number': 'end_ca_num'}, inplace=True)
```

In [15]:

```
#Dropna
trips_community_df = trips_community_df.dropna(subset=['start_ca_num','end_ca_
trips_community_df.head()
```

Out[15]:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_
0	762198876D69004D	docked_bike	2020-07-09 15:22:02	2020-07-09 15:25:52	Ritchie Ct & Banks St	18
1	BEC9C9FBA0D4CF1B	docked_bike	2020-07-24 23:56:30	2020-07-25 00:20:17	Halsted St & Roscoe St	29
2	D2FD8EA432C77EC1	docked_bike	2020-07-08 19:49:07	2020-07-08 19:56:22	Lake Shore Dr & Diversey Pkwy	32
3	54AE594E20B35881	docked_bike	2020-07-17 19:06:42	2020-07-17 19:27:38	LaSalle St & Illinois St	18
4	54025FDC7440B56F	docked_bike	2020-07-04 10:39:57	2020-07-04 10:45:05	Lake Shore Dr & North Blvd	26

Explaining the Joins (2 points)

In a short (no more than a paragraph) description, please briefly answer the following inquiries. You can write either in Markdown or in code comments in the space provided in the notebook file.

1. For each join conducted in steps 1 and 2, what was your rationale for using these particular join types?
2. Did your final `trips_community_df` end up a different size from the original `trips_pr_df` dataframe? If so, what do you think caused this difference in size?

```
In [16]: len(trips_pr_df)
```

```
Out[16]: 550425
```

```
In [17]: len(trips_community_df)
```

```
Out[17]: 545513
```

""" OPTIONAL: Use this space for either your answers for the above prompt or to run additional code. """

1. For the first step, I use the sjoin, sjoin match up geometry infomation within 2 dataframe, and make the join happens to include points geometry in the multipolygon geometry. It automatically fullfill our requirment that station_id surjection on area_number. For the step 2, I use the regular left join for the table, becuase 1 area number can have multiple station, and we need to make sure the station id will only appear once but area number can appear multiple times in a dataframe.
2. Yes. Some start or end station is not in the recorded community with community number.

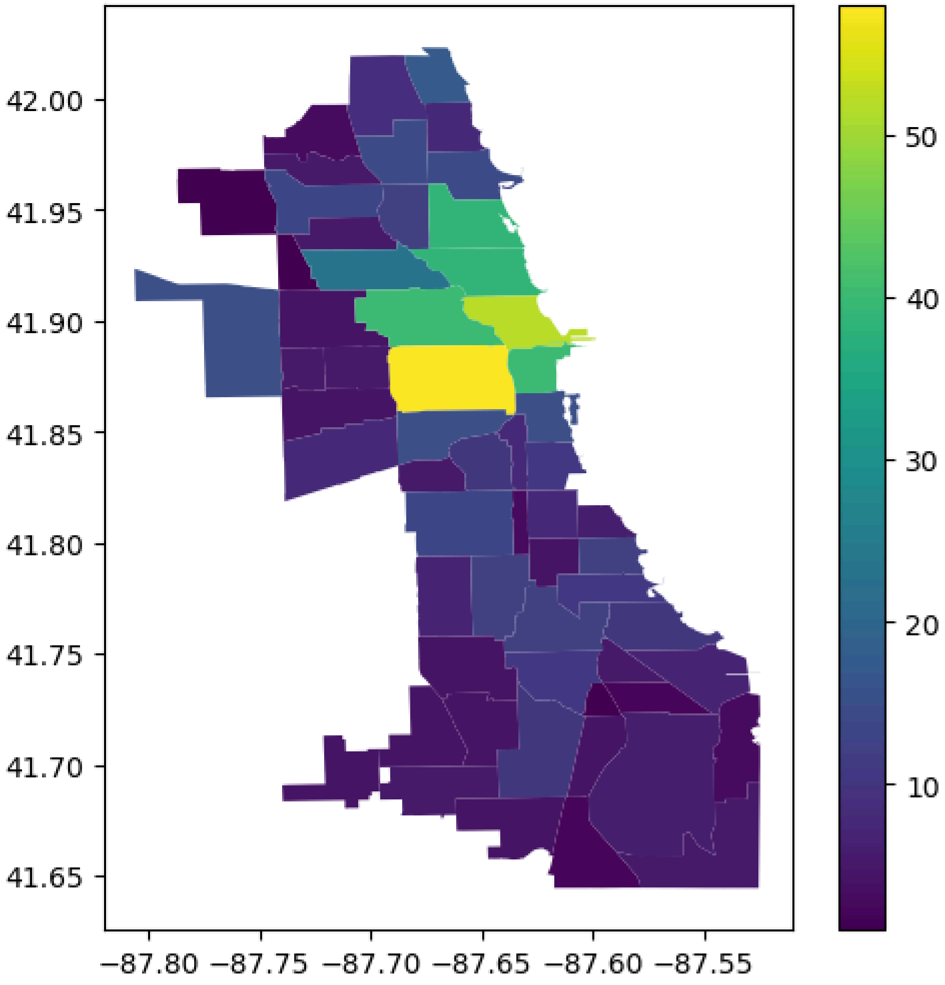
Visualize Station Distribution (4 points)

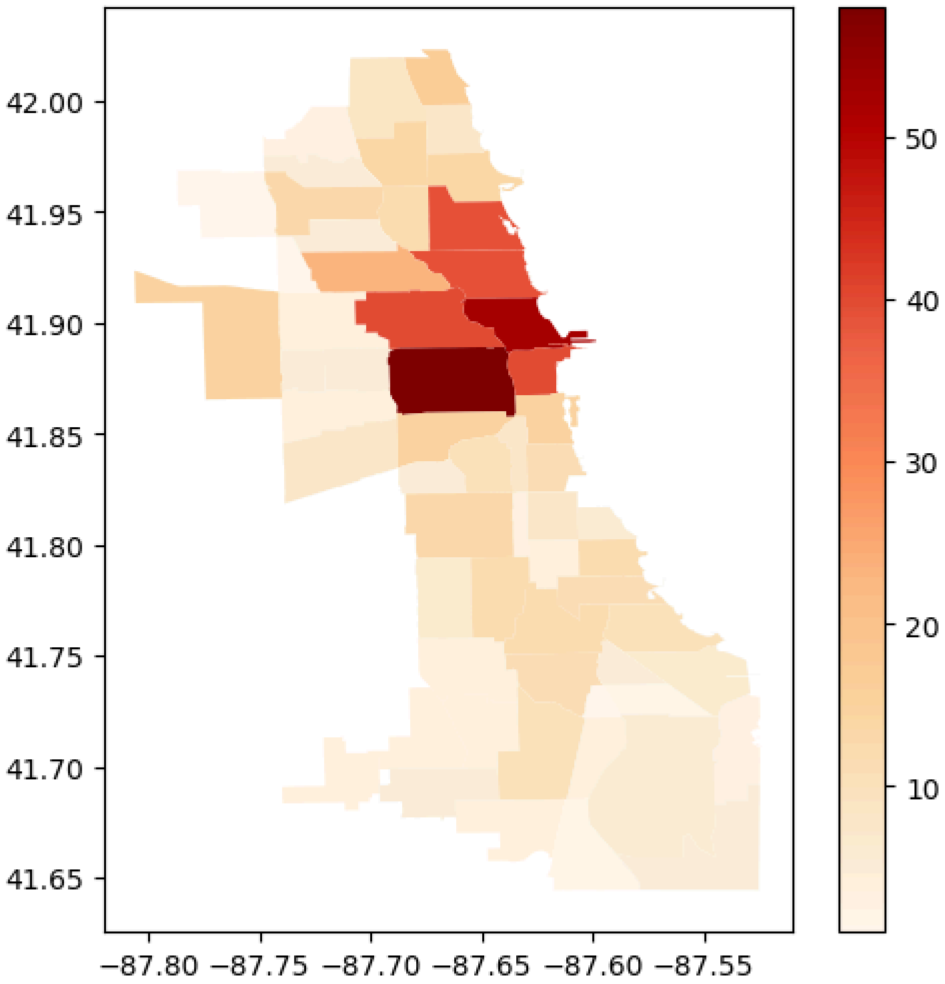
```
In [18]: #build the dictionary matching up the area_umber and geometry so each of the a
         area_geometry_dict = station_community_df.set_index('area_number')['geometry']
```

```
In [19]: #aggregate the station count in each area number
         station_count = station_community_df.groupby(['area_number']).agg({'station_id
         station_count=station_count.reset_index()
```

```
In [20]: #map the area_number with geometry information
         station_count['geometry'] = station_count['area_number'].map(area_geometry_dic
         #convert a df to gdf
         station_count = gpd.GeoDataFrame(station_count, geometry='geometry')
```

```
In [21]: """
         TODO:
         We want to understand which community areas have bike stations. Using `geopanda
         generate a plot of the number of stations per community area. This can be
         accomplished by aggregating the stations by community area. Then use the `plot
         command to generate a chloropleth map. You are allowed to define a colormap fo
         your chloropleth map via the `cmap` parameter
         """
         station_count.plot(figsize=(6,6), column='station_id',legend = True)
         station_count.plot(figsize=(6,6), cmap='OrRd', column='station_id',legend = Tru
Out[21]: <Axes: >
```


42.00
41.95
41.90
41.85
41.80
41.75
41.70
41.65
−87.80
−87.75
−87.70
−87.65
−87.60
−87.55
50
40
30
20
10


42.00
41.95
41.90
41.85
41.80
41.75
41.70
41.65
−87.80
−87.75
−87.70
−87.65
−87.60
−87.55
50
40
30
20
10

In [ ]: