Introduction

My research question for this project is, "How are the locations of COVID-19 testing centers in Chicago related to COVID cases and demographic patterns?" I aimed to discover whether the distribution of COVID testing sites was related to COVID case rates or demographic trends (proxied by racial demographics).

I conducted this research using data from Chicago's Open Data portal. My data layers display COVID case rates by Chicago zip code, racial breakdowns by Census tract, and the locations of COVID testing sites. I loaded in sidewalk data that I hoped to use to create service areas of COVID testing sites, but I ultimately had to use buffers of the testing sites due to time and open source software constraints.

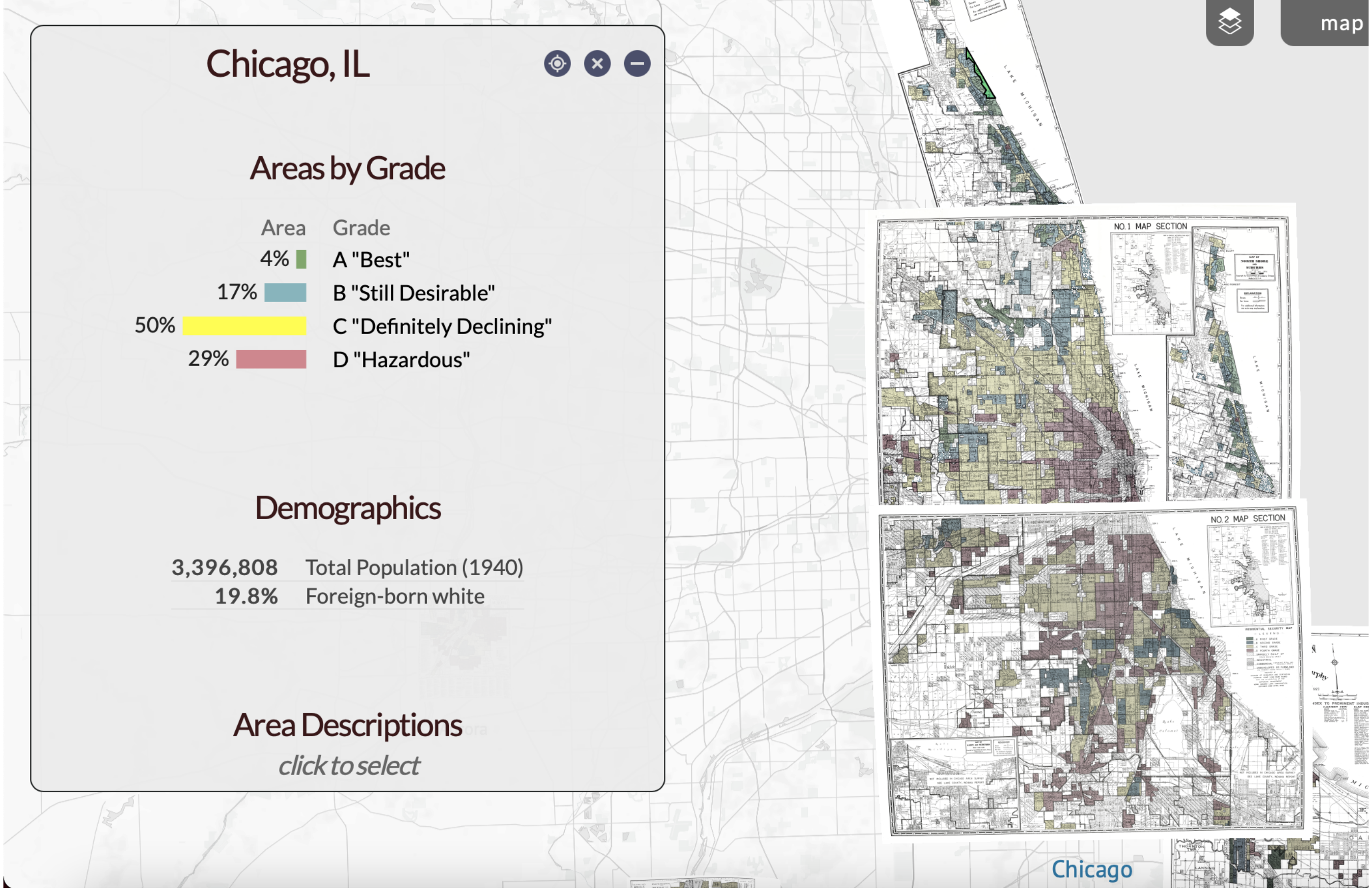
My method of statistical analysis was ANOVA (analysis of variance). This method illustrates the statistical significance of mean-differences between groups.

Background Literature

Literature throughout the pandemic has shown disparities in the impacts of COVID-19, particulary across racial and ethnic lines. A recent study looking at 6 cities in the U.S. (including Chicago) showed that a 10 percentage point increase in a zip code's Black population is associated with 9.2 additional COVID cases per 10,000 people, and a similar increase in Hispanic residents is associated with 20.6 additional cases. Potential explanations for this increase are disparities in types of jobs (frontline employment vs. employment that can move online), ability (or lack thereof) to move outside of the city during COVID peaks, access to healthcare and testing during the pandemic, and longterm preexisting health disparities (Benitez, Courtemanche, & Yelowitz 2020).

It is also important to examine literature about the history of segregation in the United States and Chicago specifically. Chicago has been used as a model for prototypical American cities (Park and Burgess 1925) and is quite segregated between its predominantly-white suburbs and its predominantly-Black "South Side". In recent decades, however, Chicago has become more diverse overall and has undergone complex racial change including an increase in Hispanic/Latino identifying residents (Onésimo Sandoval 2011).

One resource about segregation in the US that I find particularly interesting and important is the web map "Mapping Inequality" that was developed by scholars at the University of Richmond, Virginia Tech University, and the University of Maryland. The interactive mapping tool can be found here: https://dsl.richmond.edu/panorama/redlining/#loc=5/39.1/-94.58. Below is a screenshot of Chicago's redlining maps as found on the Mapping Inequality resource. It shows that most neighborhoods in southern and central Chicago were deemed "Definitely Declining" or "Hazardous" by redlining maps in the 1950s, whereas suburbs to the north and outskirts of the city were deemed "Best" or "Still Desirable". These designations were highly racially charged and illustrate the legacy of segregation on racial breakdowns in Chicago today.



Data and Data Processing

Import Packages

In [1]:

```python
#!pip install momepy
import geopandas as gpd
import fiona
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
from shapely import wkt
import networkx as nx
import osmnx as ox
import momepy
import scipy.stats as stats
#!pip install statsmodels
import statsmodels.api as sm
from statsmodels.formula.api import ols
!pip install bioinfokit
from bioinfokit.analys import stat
```

```
/Users/kearadennehy/miniconda3/lib/python3.9/site-packages/geopandas/_compat.py:106: UserW
arning: The Shapely GEOS version (3.8.0-CAPI-1.13.1 ) is incompatible with the GEOS versio
n PyGEOS was compiled with (3.9.1-CAPI-1.14.2). Conversions between both will be slow.
  warnings.warn(
Requirement already satisfied: bioinfokit in /Users/kearadennehy/miniconda3/lib/python3.9/
site-packages (2.0.6)
Requirement already satisfied: scikit-learn in /Users/kearadennehy/miniconda3/lib/python3.
9/site-packages (from bioinfokit) (0.24.2)
Requirement already satisfied: textwrap3 in /Users/kearadennehy/miniconda3/lib/python3.9/s
ite-packages (from bioinfokit) (0.9.2)
Requirement already satisfied: matplotlib in /Users/kearadennehy/miniconda3/lib/python3.9/
```

```
site-packages (from bioinfokit) (3.4.2)
Requirement already satisfied: scipy in /Users/kearadennehy/miniconda3/lib/python3.9/site-
packages (from bioinfokit) (1.7.1)
Requirement already satisfied: numpy in /Users/kearadennehy/miniconda3/lib/python3.9/site-
packages (from bioinfokit) (1.21.1)
Requirement already satisfied: pandas in /Users/kearadennehy/miniconda3/lib/python3.9/site
-packages (from bioinfokit) (1.3.1)
Requirement already satisfied: tabulate in /Users/kearadennehy/miniconda3/lib/python3.9/si
te-packages (from bioinfokit) (0.8.9)
Requirement already satisfied: seaborn in /Users/kearadennehy/miniconda3/lib/python3.9/sit
e-packages (from bioinfokit) (0.11.2)
Requirement already satisfied: adjustText in /Users/kearadennehy/miniconda3/lib/python3.9/
site-packages (from bioinfokit) (0.7.3)
Requirement already satisfied: matplotlib-venn in /Users/kearadennehy/miniconda3/lib/pytho
n3.9/site-packages (from bioinfokit) (0.11.6)
Requirement already satisfied: statsmodels in /Users/kearadennehy/miniconda3/lib/python3.
9/site-packages (from bioinfokit) (0.12.2)
Requirement already satisfied: cycler>=0.10 in /Users/kearadennehy/miniconda3/lib/python3.
9/site-packages (from matplotlib->bioinfokit) (0.10.0)
Requirement already satisfied: pyparsing>=2.2.1 in /Users/kearadennehy/miniconda3/lib/pyth
on3.9/site-packages (from matplotlib->bioinfokit) (2.4.7)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/kearadennehy/miniconda3/lib/pyt
hon3.9/site-packages (from matplotlib->bioinfokit) (1.3.1)
Requirement already satisfied: python-dateutil>=2.7 in /Users/kearadennehy/miniconda3/lib/
python3.9/site-packages (from matplotlib->bioinfokit) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in /Users/kearadennehy/miniconda3/lib/python
3.9/site-packages (from matplotlib->bioinfokit) (8.3.1)
Requirement already satisfied: six in /Users/kearadennehy/miniconda3/lib/python3.9/site-pa
ckages (from cycler>=0.10->matplotlib->bioinfokit) (1.16.0)
Requirement already satisfied: pytz>=2017.3 in /Users/kearadennehy/miniconda3/lib/python3.
9/site-packages (from pandas->bioinfokit) (2021.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /Users/kearadennehy/miniconda3/lib/
python3.9/site-packages (from scikit-learn->bioinfokit) (2.2.0)
Requirement already satisfied: joblib>=0.11 in /Users/kearadennehy/miniconda3/lib/python3.
9/site-packages (from scikit-learn->bioinfokit) (1.0.1)
Requirement already satisfied: patsy>=0.5 in /Users/kearadennehy/miniconda3/lib/python3.9/
site-packages (from statsmodels->bioinfokit) (0.5.1)
```

Data Wrangling

I loaded in Chicago's COVID data by zip code and merged it with geospatial data about each zip code.

In [2]:
```
covidCasesLink = "https://data.cityofchicago.org/api/geospatial/yhhz-zm2v?method=export&fo
casesdf = gpd.read_file(covidCasesLink)
```

In [3]:
```
casesdf = casesdf.dropna()
```

In [4]:
```
casesdf = casesdf.sort_values(by="date_week_", ascending=False)
```

In [5]:
```
casesdf = casesdf.drop(labels="geometry", axis=1)
```

In [73]:
```
casesdf.head()
```

Out[73]:

	zip_code	week_numbe	date_week_	time_week_	date_wee_2	time_wee_2	cases_week	cases_cum
4240	60622	33.0	2021-08-15	00:00:00.000	2021-08-21	00:00:00.000	76.0	5591

	zip_code	week_numbe	date_week_	time_week_	date_wee_2	time_wee_2	cases_week	cases_cun
4271	60633	33.0	2021-08-15	00:00:00.000	2021-08-21	00:00:00.000	12.0	1422
4268	60619	33.0	2021-08-15	00:00:00.000	2021-08-21	00:00:00.000	82.0	5348
4267	60653	33.0	2021-08-15	00:00:00.000	2021-08-21	00:00:00.000	49.0	3020
4265	60638	33.0	2021-08-15	00:00:00.000	2021-08-21	00:00:00.000	47.0	8265

5 rows × 22 columns

In [7]:

```
zipCodesLink = "https://data.cityofchicago.org/api/geospatial/gdcf-axmw?method=export&forn
zipcodesdf = gpd.read_file(zipCodesLink)
```

In [8]:

```
zipcodesdf = zipcodesdf.merge(casesdf, right_on="zip_code", left_on="zip")
```

In [9]:

```
zipcodesdf.head()
```

Out[9]:

	objectid	shape_area	shape_len	zip	geometry	zip_code	week_numbe	date_week_	time_we
0	33.0	1.060523e+08	42720.044406	60647	POLYGON ((-87.67762 41.91776, -87.67761 41.917...	60647	33.0	2021-08-15	00:00:00
1	33.0	1.060523e+08	42720.044406	60647	POLYGON ((-87.67762 41.91776, -87.67761 41.917...	60647	32.0	2021-08-08	00:00:00
2	33.0	1.060523e+08	42720.044406	60647	POLYGON ((-87.67762 41.91776, -87.67761 41.917...	60647	31.0	2021-08-01	00:00:00
3	33.0	1.060523e+08	42720.044406	60647	POLYGON ((-87.67762 41.91776, -87.67761 41.917...	60647	30.0	2021-07-25	00:00:00
4	33.0	1.060523e+08	42720.044406	60647	POLYGON ((-87.67762 41.91776, -87.67761 41.917...	60647	29.0	2021-07-18	00:00:00

5 rows × 27 columns

I then chose which columns I was interested in and created a new dataframe with those columns. I then isolated data from the first week of August (the most recent week when I started working on this project). I created a column for cumulative cases per 10,000 people and made the dataframe into a geodataframe.

In [10]:

```
geometryseries = zipcodesdf["geometry"]
zipseries = zipcodesdf["zip"]
areaseries = zipcodesdf["shape_area"]
lenseries = zipcodesdf["shape_len"]
cumucasesseries = zipcodesdf["cases_cumu"]
zipcodesseries = zipcodesdf["zip"]
datesseries = zipcodesdf["date_week_"]
populationseries = zipcodesdf["population"]
frame = {"zip": zipcodesseries, "week": datesseries, "shape_area": areaseries, "shape_len'
cumulativecasesdf = pd.DataFrame(frame)
```

In [11]:

```
cumulativecasesdf.set_geometry("geometry")
```

Out[11]:

	zip	week	shape_area	shape_len	geometry	cumu_cases	population
0	60647	2021-08-15	1.060523e+08	42720.044406	POLYGON ((-87.67762 41.91776, -87.67761 41.917...	9433.0	87509.0
1	60647	2021-08-08	1.060523e+08	42720.044406	POLYGON ((-87.67762 41.91776, -87.67761 41.917...	9324.0	87509.0
2	60647	2021-08-01	1.060523e+08	42720.044406	POLYGON ((-87.67762 41.91776, -87.67761 41.917...	9224.0	87509.0
3	60647	2021-07-25	1.060523e+08	42720.044406	POLYGON ((-87.67762 41.91776, -87.67761 41.917...	9140.0	87509.0
4	60647	2021-07-18	1.060523e+08	42720.044406	POLYGON ((-87.67762 41.91776, -87.67761 41.917...	9071.0	87509.0
...	...	...	...	...	...	...	...
4315	60619	2020-04-12	1.678720e+08	53040.907078	POLYGON ((-87.58592 41.75150, -87.58592 41.751...	432.0	61258.0
4316	60619	2020-04-05	1.678720e+08	53040.907078	POLYGON ((-87.58592 41.75150, -87.58592 41.751...	339.0	61258.0
4317	60619	2020-03-29	1.678720e+08	53040.907078	POLYGON ((-87.58592 41.75150, -87.58592 41.751...	242.0	61258.0
4318	60619	2020-03-22	1.678720e+08	53040.907078	POLYGON ((-87.58592 41.75150, -87.58592 41.751...	130.0	61258.0
4319	60619	2020-03-15	1.678720e+08	53040.907078	POLYGON ((-87.58592 41.75150, -87.58592 41.751...	31.0	61258.0

4320 rows × 7 columns

In [12]:

```
cumulativecasesdf = cumulativecasesdf[cumulativecasesdf["week"]=="2021-08-01"]
```

In [13]:

```
cumulativecasesdf["CasesPer10000"] = cumulativecasesdf["cumu_cases"] / cumulativecasesdf['
```

In [14]:

```
cumulativecasesgdf = gpd.GeoDataFrame(
    cumulativecasesdf, geometry=cumulativecasesdf["geometry"])
```

In [15]:

```
testingSitesLink = "https://data.cityofchicago.org/api/views/thdn-3grx/rows.csv?accessType
testingsitesdf = pd.read_csv(testingSitesLink)
```

In [16]:
```
testingsitesdf = testingsitesdf.dropna()
```

In [17]:
```
testingsitesdf['Location'] = gpd.GeoSeries.from_wkt(testingsitesdf['Location'])
testingsitesgdf = gpd.GeoDataFrame(testingsitesdf, geometry="Location")
```

In [72]:
```
cumulativecasesgdf.head()
```

Out[72]:

	zip	week	shape_area	shape_len	geometry	cumu_cases	population	CasesPer10000
2	60647	2021-08-01	1.060523e+08	42720.044406	POLYGON ((1162710.570 1913303.050, 1162715.690...	9224.0	87509.0	1054.063011
78	60639	2021-08-01	1.274761e+08	48103.782721	POLYGON ((1149304.490 1914985.850, 1149278.070...	14817.0	90517.0	1636.930079
193	60622	2021-08-01	7.085383e+07	42527.989679	POLYGON ((1165664.482 1902791.860, 1165664.480...	5438.0	52793.0	1030.060804
269	60651	2021-08-01	9.903962e+07	47970.140153	POLYGON ((1154894.670 1905152.880, 1154849.960...	8168.0	63218.0	1292.037078
344	60611	2021-08-01	2.350606e+07	34689.350631	POLYGON ((1180096.830 1904616.840, 1180186.790...	2434.0	32426.0	750.632209

I loaded sidewalk data from Cook County (Chicago's county) into a geodataframe as well.

In [19]:
```
sidewalksLink = 'https://datahub.cmap.illinois.gov/dataset/93394557-3315-4b93-ab53-9fd576b
sidewalksdf = gpd.read_file(sidewalksLink)
```

In [20]:
```
sidewalksgdf = gpd.GeoDataFrame(sidewalksdf, geometry="geometry")
```

I set all geodataframes to EPSG 3435 (NAD83 / Illinois East (ftUS)).

In [21]:
```
print(sidewalksgdf.crs)
print(cumulativecasesgdf.crs)
print(testingsitesgdf.crs)
```

```
epsg:3435
None
None
```

In [22]:
```
cumulativecasesgdf = cumulativecasesgdf.set_crs('EPSG:4326')
cumulativecasesgdf = cumulativecasesgdf.to_crs('EPSG:3435')

testingsitesgdf = testingsitesgdf.set_crs('EPSG:4326')
testingsitesgdf = testingsitesgdf.to_crs('EPSG:3435')
```

I clipped my sidewalk data by the COVID cases (Chicago boundaries).

In [23]:

```
sidewalksgdf = gpd.clip(sidewalksgdf, cumulativecasesgdf)
```

Then, I loaded Census tract geospatial data into a geodataframe.

In [24]:

```
censusTractsLink = "https://data.cityofchicago.org/api/geospatial/5jrd-6zik?method=export&
censustractsgdf = gpd.read_file(censusTractsLink)
```

In [74]:

```
censustractsgdf.head()
```

Out[74]:

	commarea	commarea_n	countyfp10	geoid10	name10	namelsad10	notes	statefp10	tractce10	
0	44	44.0	031	17031842400	8424	Census Tract 8424	None	17	842400	((1 11
1	59	59.0	031	17031840300	8403	Census Tract 8403	None	17	840300	((1 1 11
2	34	34.0	031	17031841100	8411	Census Tract 8411	None	17	841100	((1 11
3	31	31.0	031	17031841200	8412	Census Tract 8412	None	17	841200	((1 11
4	32	32.0	031	17031839000	8390	Census Tract 8390	None	17	839000	((1 11

Census demographic data was unable to load in via link. Instead, I found a table here: https://data.census.gov/cedsci/advanced?q=chicago. I downloaded tract-level race data for all of Chicago to my local computer.

In [26]:

```
racedf = pd.read_csv("/Users/kearadennehy/Downloads/DECENNIALPL2010/DECENNIALPL2010.P1_dat
```

In [27]:

```
racedf.columns = racedf.iloc[0]
racedf = racedf.drop([0])
racedf = racedf.reset_index()
racedf.insert(loc=73, column="PctWhite", value=1.0)
```

I created columns for percent of the population that is white and percent of the population that is of color. I made a new dataframe called "simpleracedf" with these columns, then merged that dataframe back into my Census geodataframe.

In [28]:

```
convert_dict = {"Total": float,
                "Total!!Population of one race!!White alone": float
               }

racedf = racedf.astype(convert_dict)

racedf["PctWhite"] = racedf["Total!!Population of one race!!White alone"] / racedf["Total'

racedf["PctPOC"] = 1.0 - racedf["PctWhite"]
```

In [29]:

```
idseries = racedf["id"].str.split(pat="1400000US", expand=True)[1]
nameseries = racedf["Geographic Area Name"]
totalpopseries = racedf["Total"]
pctwhiteseries = racedf["PctWhite"]
pctpocseries = racedf["PctPOC"]
frame2 = {"id": idseries, "Name": nameseries, "Population": totalpopseries, "PctWhite": pc
simpleracedf = pd.DataFrame(frame2)
simpleracedf
```

Out[29]:

	id	Name	Population	PctWhite	PctPOC
0	17031010100	Census Tract 101, Cook County, Illinois	4854.0	0.372888	0.627112
1	17031010201	Census Tract 102.01, Cook County, Illinois	6450.0	0.358450	0.641550
2	17031010202	Census Tract 102.02, Cook County, Illinois	2818.0	0.438964	0.561036
3	17031010300	Census Tract 103, Cook County, Illinois	6236.0	0.523894	0.476106
4	17031010400	Census Tract 104, Cook County, Illinois	5042.0	0.662634	0.337366
...	...	...	...	...	...
1314	17031843800	Census Tract 8438, Cook County, Illinois	2110.0	0.291943	0.708057
1315	17031843900	Census Tract 8439, Cook County, Illinois	3533.0	0.036513	0.963487
1316	17031980000	Census Tract 9800, Cook County, Illinois	0.0	NaN	NaN
1317	17031980100	Census Tract 9801, Cook County, Illinois	0.0	NaN	NaN
1318	17031990000	Census Tract 9900, Cook County, Illinois	0.0	NaN	NaN

1319 rows × 5 columns

In [30]:

```
simpleracedf
```

Out[30]:

	id	Name	Population	PctWhite	PctPOC
0	17031010100	Census Tract 101, Cook County, Illinois	4854.0	0.372888	0.627112
1	17031010201	Census Tract 102.01, Cook County, Illinois	6450.0	0.358450	0.641550
2	17031010202	Census Tract 102.02, Cook County, Illinois	2818.0	0.438964	0.561036
3	17031010300	Census Tract 103, Cook County, Illinois	6236.0	0.523894	0.476106
4	17031010400	Census Tract 104, Cook County, Illinois	5042.0	0.662634	0.337366
...	...	...	...	...	...
1314	17031843800	Census Tract 8438, Cook County, Illinois	2110.0	0.291943	0.708057
1315	17031843900	Census Tract 8439, Cook County, Illinois	3533.0	0.036513	0.963487

	id	Name	Population	PctWhite	PctPOC
1316	17031980000	Census Tract 9800, Cook County, Illinois	0.0	NaN	NaN
1317	17031980100	Census Tract 9801, Cook County, Illinois	0.0	NaN	NaN
1318	17031990000	Census Tract 9900, Cook County, Illinois	0.0	NaN	NaN

1319 rows × 5 columns

In [31]:

```
censustractsgdf = censustractsgdf.merge(simpleracedf, right_on="id", left_on="geoid10")
```

In [32]:

```
censustractsgdf = censustractsgdf.to_crs('EPSG:3435')
```

Below is my final Census tract geodataframe. It includes columns for percent of the population that is white and percent of the population that is of color.

In [33]:

```
censustractsgdf
```

Out[33]:

	commarea	commarea_n	countyfp10	geoid10	name10	namelsad10	notes	statefp10	tractce10
0	44	44.0	031	17031842400	8424	Census Tract 8424	None	17	842400
1	59	59.0	031	17031840300	8403	Census Tract 8403	None	17	840300
2	34	34.0	031	17031841100	8411	Census Tract 8411	None	17	841100
3	31	31.0	031	17031841200	8412	Census Tract 8412	None	17	841200
4	32	32.0	031	17031839000	8390	Census Tract 8390	None	17	839000
...	...	...	...	...	...	...	...	...	...
796	7	7.0	031	17031070400	704	Census Tract 704	None	17	070400

	commarea	commarea_n	countyfp10	geoid10	name10	namelsad10	notes	statefp10	tractce10
797	7	7.0	031	17031070500	705	Census Tract 705	None	17	070500
798	13	13.0	031	17031130300	1303	Census Tract 1303	None	17	130300
799	29	29.0	031	17031292200	2922	Census Tract 2922	None	17	292200
800	63	63.0	031	17031630900	6309	Census Tract 6309	None	17	630900

801 rows × 15 columns

Below is my final COVID cases geodataframe. It includes cumulative cases per 10,000 people for each zip code.

In [70]:

```
cumulativecasesgdf
```

Out[70]:

	zip	week	shape_area	shape_len	geometry	cumu_cases	population	CasesPer10000
2	60647	2021-08-01	1.060523e+08	42720.044406	POLYGON ((1162710.570 1913303.050, 1162715.690...	9224.0	87509.0	1054.063011
78	60639	2021-08-01	1.274761e+08	48103.782721	POLYGON ((1149304.490 1914985.850, 1149278.070...	14817.0	90517.0	1636.930079
193	60622	2021-08-01	7.085383e+07	42527.989679	POLYGON ((1165664.482 1902791.860, 1165664.480...	5438.0	52793.0	1030.060804
269	60651	2021-08-01	9.903962e+07	47970.140153	POLYGON ((1154894.670 1905152.880, 1154849.960...	8168.0	63218.0	1292.037078
344	60611	2021-08-01	2.350606e+07	34689.350631	POLYGON ((1180096.830 1904616.840, 1180186.790...	2434.0	32426.0	750.632209
419	60638	2021-08-01	1.661663e+08	67710.646739	POLYGON ((1145038.120 1877098.200, 1145038.830...	8119.0	58797.0	1380.852765

	zip	week	shape_area	shape_len	geometry	cumu_cases	population	CasesPer10000
494	60652	2021-08-01	1.283098e+08	48187.949880	POLYGON ((1161676.040 1854860.750, 1161677.220...	5605.0	43907.0	1276.561824
569	60626	2021-08-01	4.917058e+07	33983.913306	POLYGON ((1166067.060 1951048.980, 1166072.700...	4564.0	49730.0	917.755882
644	60615	2021-08-01	6.656545e+07	38321.313769	POLYGON ((1189361.370 1872142.680, 1189398.870...	2610.0	41563.0	627.962370
719	60621	2021-08-01	1.047468e+08	42299.920391	POLYGON ((1177515.860 1857917.930, 1177516.630...	2504.0	29042.0	862.199573
794	60645	2021-08-01	6.218147e+07	32075.599421	POLYGON ((1159502.470 1950363.100, 1159637.750...	5033.0	47732.0	1054.428895
869	60643	2021-08-01	1.830131e+08	59337.807440	POLYGON ((1173133.650 1825945.560, 1173155.640...	4715.0	49870.0	945.458191
944	60643	2021-08-01	3.136688e+06	7175.092168	POLYGON ((1173372.620 1818761.500, 1173322.720...	4715.0	49870.0	945.458191
1019	60660	2021-08-01	3.798439e+07	27307.813836	POLYGON ((1168745.010 1942680.340, 1168763.980...	3055.0	43242.0	706.489062
1094	60640	2021-08-01	7.730525e+07	48985.871304	POLYGON ((1169889.620 1937438.500, 1169879.180...	5117.0	69715.0	733.988381
1169	60614	2021-08-01	9.446063e+07	50587.347380	POLYGON ((1175249.530 1918946.000, 1175250.880...	6206.0	71308.0	870.309082
1245	60631	2021-08-01	1.071176e+08	68289.751577	POLYGON ((1113850.780 1937277.680, 1113851.310...	3359.0	29529.0	1137.525822
1320	60646	2021-08-01	1.178901e+08	62633.139821	POLYGON ((1131758.090 1943202.870, 1131787.480...	2900.0	27987.0	1036.195376
1395	60628	2021-08-01	3.452417e+08	83748.991990	POLYGON ((1188501.500 1842031.460, 1188493.500...	5831.0	66724.0	873.898447
1470	60625	2021-08-01	1.058309e+08	42524.110478	POLYGON ((1162224.950 1929224.820, 1162177.660...	7440.0	79243.0	938.884192

	zip	week	shape_area	shape_len	geometry	cumu_cases	population	CasesPer10000
1545	60641	2021-08-01	1.139033e+08	42682.982947	POLYGON ((1149241.150 1918317.690, 1149178.660...	9327.0	71023.0	1313.236557
1620	60657	2021-08-01	6.354776e+07	45646.488028	POLYGON ((1173407.570 1924468.300, 1173430.380...	5767.0	70052.0	823.245589
1696	60636	2021-08-01	1.041147e+08	42448.282844	POLYGON ((1169622.050 1854989.950, 1168296.990...	3549.0	32203.0	1102.071236
1771	60649	2021-08-01	8.052608e+07	55092.863293	POLYGON ((1199827.780 1853229.590, 1199131.600...	3801.0	46024.0	825.873457
1846	60617	2021-08-01	4.528374e+08	102528.036078	POLYGON ((1199893.750 1853181.290, 1199898.480...	8528.0	82534.0	1033.271137
1921	60633	2021-08-01	1.875267e+08	59703.215815	POLYGON ((1184469.530 1824765.270, 1189067.610...	1403.0	12871.0	1090.047393
1994	60612	2021-08-01	1.067189e+08	42663.196676	POLYGON ((1162929.220 1905244.080, 1162930.540...	3753.0	34311.0	1093.818309
2069	60604	2021-08-01	4.294902e+06	12245.808402	POLYGON ((1174762.920 1899362.320, 1174763.850...	110.0	782.0	1406.649616
2139	60624	2021-08-01	9.941812e+07	40394.410008	POLYGON ((1155213.880 1894533.120, 1155157.680...	3591.0	36158.0	993.141214
2214	60656	2021-08-01	8.951588e+07	84430.973216	POLYGON ((1108622.090 1933078.350, 1108639.180...	3343.0	27579.0	1212.154175
2289	60644	2021-08-01	9.909222e+07	40030.838529	POLYGON ((1145868.800 1894304.080, 1145867.800...	4769.0	47712.0	999.538900
2365	60655	2021-08-01	1.153801e+08	77392.322613	POLYGON ((1162125.840 1838922.770, 1162129.420...	3562.0	28804.0	1236.633801
2439	60603	2021-08-01	4.560229e+06	13672.682289	POLYGON ((1179499.190 1900447.100, 1179495.570...	84.0	1174.0	715.502555
2506	60605	2021-08-01	3.630128e+07	37973.346105	POLYGON ((1178341.150 1898593.570, 1179224.086...	2067.0	27519.0	751.117410

	zip	week	shape_area	shape_len	geometry	cumu_cases	population	CasesPer10000
2581	60653	2021-08-01	6.775983e+07	34316.361543	POLYGON ((1183378.140 1881936.210, 1183399.440...	2932.0	31972.0	917.052421
2656	60609	2021-08-01	2.134903e+08	58540.920413	POLYGON ((1176844.900 1881764.500, 1176846.710...	7539.0	61495.0	1225.953330
2731	60618	2021-08-01	1.412359e+08	47847.080382	POLYGON ((1162376.170 1923907.580, 1162377.950...	8873.0	94395.0	939.986228
2807	60616	2021-08-01	1.096671e+08	45823.539838	POLYGON ((1183367.490 1881980.950, 1182507.860...	3995.0	54464.0	733.512045
2882	60602	2021-08-01	4.847125e+06	14448.174993	POLYGON ((1181225.660 1901279.920, 1181224.750...	115.0	1244.0	924.437299
2953	60601	2021-08-01	9.166246e+06	19804.582109	POLYGON ((1177741.840 1902882.940, 1177849.890...	1145.0	14675.0	780.238501
3028	60608	2021-08-01	1.765055e+08	53169.217720	POLYGON ((1171293.470 1892289.680, 1171295.330...	8332.0	79205.0	1051.953791
3103	60607	2021-08-01	6.466429e+07	39143.639517	POLYGON ((1173173.090 1898448.860, 1173267.180...	2792.0	29591.0	943.530127
3178	60661	2021-08-01	9.357756e+06	13132.565490	POLYGON ((1173041.650 1902921.770, 1173037.700...	931.0	9926.0	937.940762
3252	60606	2021-08-01	6.766411e+06	12040.440235	POLYGON ((1174680.590 1902381.080, 1174681.080...	344.0	3101.0	1109.319574
3345	60630	2021-08-01	1.310738e+08	49198.940283	POLYGON ((1142186.810 1937012.780, 1142450.930...	6033.0	57344.0	1052.071708
3420	60642	2021-08-01	4.796149e+07	31296.227144	POLYGON ((1165664.482 1902791.860, 1165657.780...	2165.0	20201.0	1071.729122
3494	60659	2021-08-01	6.969841e+07	38931.526661	POLYGON ((1162029.240 1937153.010, 1159312.660...	4290.0	41068.0	1044.608941
3569	60634	2021-08-01	1.940626e+08	77647.318007	POLYGON ((1138557.480 1918057.610, 1138459.640...	10747.0	75995.0	1414.171985

	zip	week	shape_area	shape_len	geometry	cumu_cases	population	CasesPer10000
3644	60613	2021-08-01	5.399089e+07	31196.320656	POLYGON ((1173142.110 1925167.220, 1172089.300...	4083.0	50113.0	814.758645
3720	60610	2021-08-01	3.159816e+07	24208.698879	POLYGON ((1176224.670 1905728.050, 1176224.667...	3761.0	39019.0	963.889387
3796	60654	2021-08-01	1.586996e+07	17119.698877	POLYGON ((1176224.670 1905728.050, 1176227.120...	2198.0	19135.0	1148.680429
3871	60632	2021-08-01	2.117553e+08	63253.238669	POLYGON ((1158274.840 1881316.350, 1158294.680...	13623.0	91039.0	1496.391656
3946	60623	2021-08-01	1.552855e+08	53406.915617	POLYGON ((1158274.840 1881316.350, 1158257.320...	11337.0	85979.0	1318.577792
4021	60629	2021-08-01	2.111148e+08	58701.325375	POLYGON ((1161671.610 1855025.450, 1161672.220...	17757.0	111850.0	1587.572642
4096	60620	2021-08-01	2.116961e+08	58466.160298	POLYGON ((1177968.250 1841966.540, 1177926.720...	5932.0	68096.0	871.123120
4172	60637	2021-08-01	1.254243e+08	52377.854541	POLYGON ((1190513.200 1868834.400, 1190490.030...	3883.0	47454.0	818.266110
4247	60619	2021-08-01	1.678720e+08	53040.907078	POLYGON ((1188196.100 1852924.880, 1188196.210...	5178.0	61258.0	845.277352

Below is my final testing sites geodataframe.

In [71]:
```
testingsitesgdf
```

Out[71]:

	Facility	Phone	Address	Web Site	Location	buffered	
0	Heartland Health Center - Edgewater	(773) 751-7800	1300 W Devon Ave Chicago, IL 60660	https://www.heartlandhealthcenters.org/covid-1...	POINT (1166403.189 1942634.785)	POLYGON ((1169043.189 1942634.785, 1169030.477...	(1
1	University of Chicago Hospital - Hyde Park	(773) 702-2800	901 E 58th St Chicago, IL 60637	https://www.uchicagomedicine.org/patients-visi...	POINT (1183231.010 1866746.946)	POLYGON ((1185871.010 1866746.946, 1185858.298...	1

	Facility	Phone	Address	Web Site	Location	buffered
2	Near North Health Service Corporation: Chicago...	(773) 227-8022	1734 W Chicago Ave Chicago, IL 60622	https://www.nearnorthhealth.org/	POINT (1164527.323 1905394.543)	POLYGON ((1167167.323 1905394.543, 1167154.611...
3	Aayu Clinics - Lakeview	(773) 227-3669	1645 A W School St Chicago, IL 60657	https://www.aayuclinics.com/services-1	POINT (1164655.887 1921973.142)	POLYGON ((1167295.887 1921973.142, 1167283.175...
4	Loretto Hospital	(773) 854-5475	645 S Central Ave Chicago, IL 60644	https://www.lorettohospital.org/covid-19-testing/	POINT (1139191.328 1896475.545)	POLYGON ((1141831.328 1896475.545, 1141818.615...
...	...	...	...	...	...	...
142	Lurie Children's Hospital	(312) 227-5300	225 E Chicago Ave, IL 60611	https://www.luriechildrens.org/en/news-stories...	POINT (1177937.863 1905767.195)	POLYGON ((1180577.863 1905767.195, 1180565.150...
143	PCC Community Wellness Center - Salud	(773) 295-3347	5359 W Fullerton Ave Chicago, IL 60639	https://www.pccwellness.org/covid-19?id=177	POINT (1140143.113 1915425.287)	POLYGON ((1142783.113 1915425.287, 1142770.401...
144	PrimeCare - Northwest Health Center	(312) 633-5841	1649 N Pulaski Rd Chicago, IL 60639	https://primecarehealth.org/northwest-health-c...	POINT (1149453.628 1910853.567)	POLYGON ((1152093.628 1910853.567, 1152080.916...
145	John H. Stroger, Jr. Cook County Hospital	(312) 864-6000	1969 W Ogden Ave Chicago, IL 60612	https://cookcountyhealth.org/locations/john-h-...	POINT (1163307.419 1897094.229)	POLYGON ((1165947.419 1897094.229, 1165934.707...
146	Mile Square Health Center UIC - Back of the Yards	(312) 996-2000	4630 S Bishop St Chicago, IL 60609	https://hospital.uillinois.edu/patients-and-vi...	POINT (1167454.161 1873852.019)	POLYGON ((1170094.161 1873852.019, 1170081.449...

128 rows × 7 columns

Data Analysis

I pulled one testing site out of my testing sites geodataframe to try and complete my network analysis. I ultimately was unable to perform network analysis, but I kept the small sidewalk visualizations I made to give a glimpse into my process and to show the sidewalk data that I loaded into the notebook.

In [34]:

```
testingsitesgdf.loc[0]
```

```
Facility                    Heartland Health Center - Edgewater
```

```
Out[34]: Phone                                       (773) 751-7800
         Address                     1300 W Devon Ave Chicago, IL 60660
         Web Site    https://www.heartlandhealthcenters.org/covid-1...
         Location          POINT (1166403.189108892 1942634.785260455)
         Name: 0, dtype: object
```

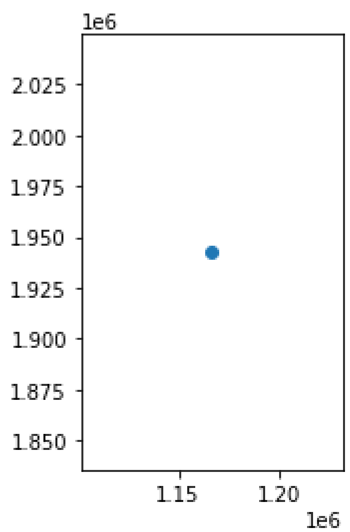
In [35]:
```
samplesite = gpd.GeoDataFrame(testingsitesgdf.loc[0])
samplesite = samplesite.transpose()
samplesite
```

Out[35]:

	Facility	Phone	Address	Web Site	Location
0	Heartland Health Center - Edgewater	(773) 751-7800	1300 W Devon Ave Chicago, IL 60660	https://www.heartlandhealthcenters.org/covid-1...	POINT (1166403.189108892 1942634.785260455)

In [36]:
```
samplesite = samplesite.set_geometry("Location")
samplesite = samplesite.set_crs('EPSG:3435')
samplesite.plot()
```

Out[36]: <AxesSubplot:>



In [37]:
```
samplesite.crs
```

Out[37]:
```
<Projected CRS: EPSG:3435>
Name: NAD83 / Illinois East (ftUS)
Axis Info [cartesian]:
- X[east]: Easting (US survey foot)
- Y[north]: Northing (US survey foot)
Area of Use:
- name: United States (USA) - Illinois - counties of Boone; Champaign; Clark; Clay; Coles; Cook; Crawford; Cumberland; De Kalb; De Witt; Douglas; Du Page; Edgar; Edwards; Effingham; Fayette; Ford; Franklin; Gallatin; Grundy; Hamilton; Hardin; Iroquois; Jasper; Jefferson; Johnson; Kane; Kankakee; Kendall; La Salle; Lake; Lawrence; Livingston; Macon; Marion; Massac; McHenry; McLean; Moultrie; Piatt; Pope; Richland; Saline; Shelby; Vermilion; Wabash; Wayne; White; Will; Williamson.
- bounds: (-89.28, 37.06, -87.02, 42.5)
Coordinate Operation:
- name: SPCS83 Illinois East zone (US Survey feet)
- method: Transverse Mercator
Datum: North American Datum 1983
- Ellipsoid: GRS 1980
- Prime Meridian: Greenwich
```

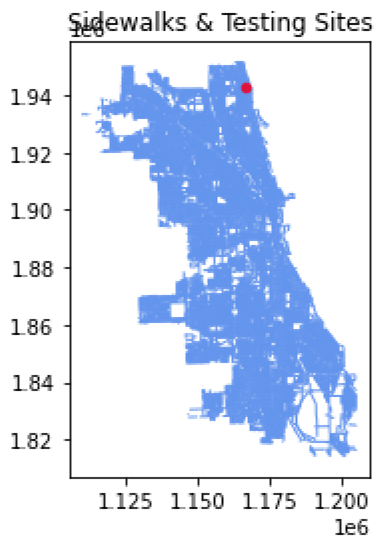
In [38]:
```
fig, ax = plt.subplots()

ax.set_aspect('equal')
ax.set_title('Sidewalks & Testing Sites')

sidewalksgdf.plot(ax=ax, color='cornflowerblue', linewidth=1, zorder=0)

#testingsitesgdf.plot(ax=ax, color='crimson', zorder=10, markersize=20)
samplesite.plot(ax=ax, color='crimson', zorder=10, markersize=20)

plt.show();
```



I tested a buffer on the sample site once I realized I would be unable to complete network analysis.

In [39]:
```
samplesite["buffered"] = samplesite.buffer(2640)
```

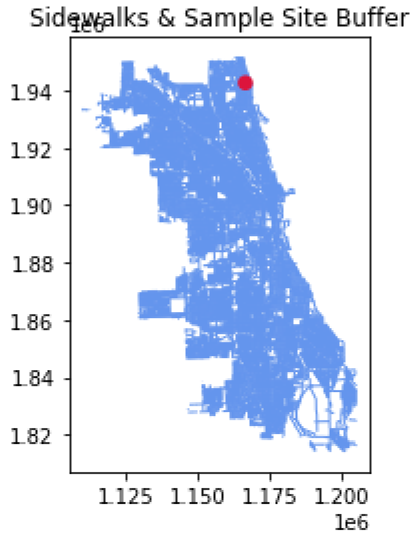
In [40]:
```
fig, ax = plt.subplots()

ax.set_aspect('equal')
ax.set_title('Sidewalks & Sample Site Buffer')

sidewalksgdf.plot(ax=ax, color='cornflowerblue', linewidth=1, zorder=0)

#testingsitesgdf.plot(ax=ax, color='crimson', zorder=10, markersize=20)
samplesite["buffered"].plot(ax=ax, color='crimson', zorder=10, markersize=20)

plt.show();
```



I decided to go with a half mile buffer for each testing site to get a sense of areas that have readily accessible testing sites.

In [41]:

```
testingsitesgdf["buffered"] = testingsitesgdf.buffer(2640)
buffered_sites = gpd.GeoDataFrame(data=testingsitesgdf, geometry=testingsitesgdf["bufferec
```

In [42]:

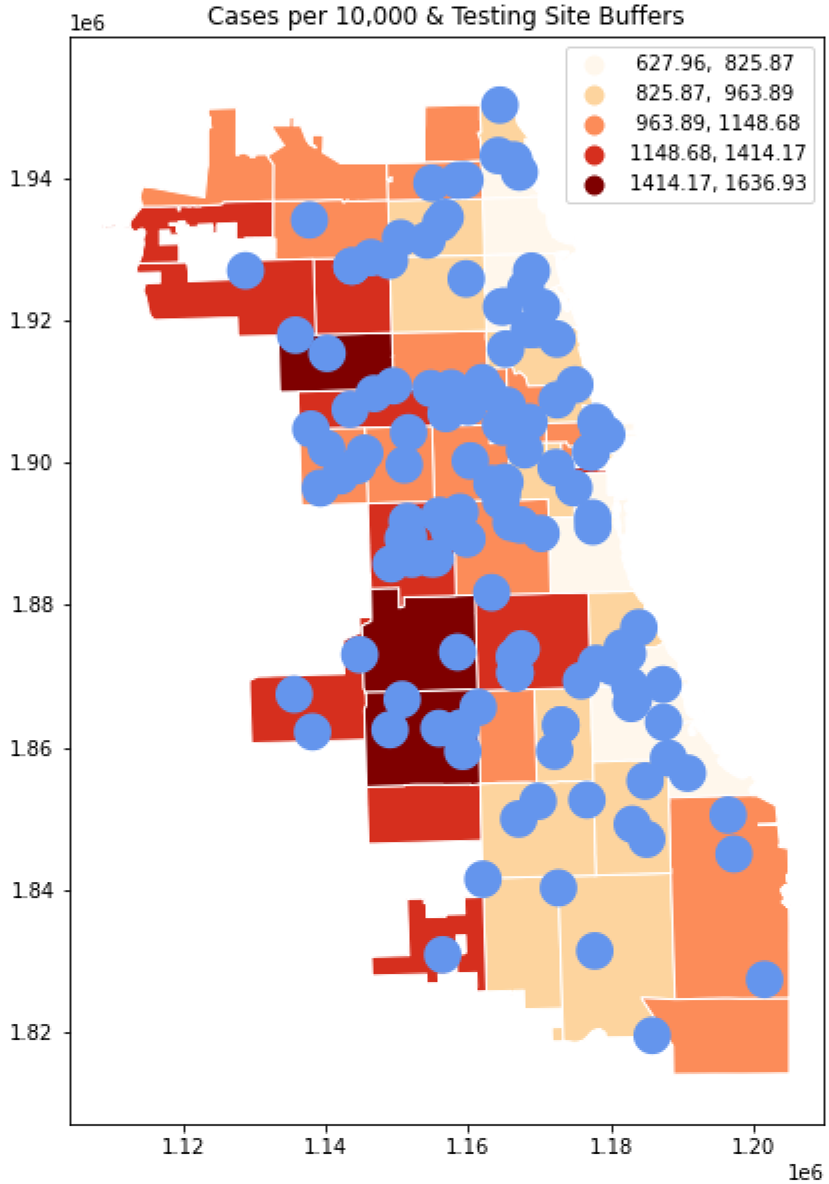
```
fig, ax = plt.subplots(figsize=(15, 10))

ax.set_aspect('equal')
ax.set_title('Cases per 10,000 & Testing Site Buffers')

cumulativecasesgdf.plot(ax=ax, column="CasesPer10000", legend=True, cmap="OrRd", scheme="N

#testingsitesgdf.plot(ax=ax, color='crimson', zorder=10, markersize=20)
buffered_sites.plot(ax=ax, color='cornflowerblue', zorder=10, markersize=20)

plt.show();
```



I then created 'near' and 'not near' layers based on the buffers and merged them together. That way, I have one layer with polygons that are near and not near testing sites. This layer will be used for ANOVA analysis.

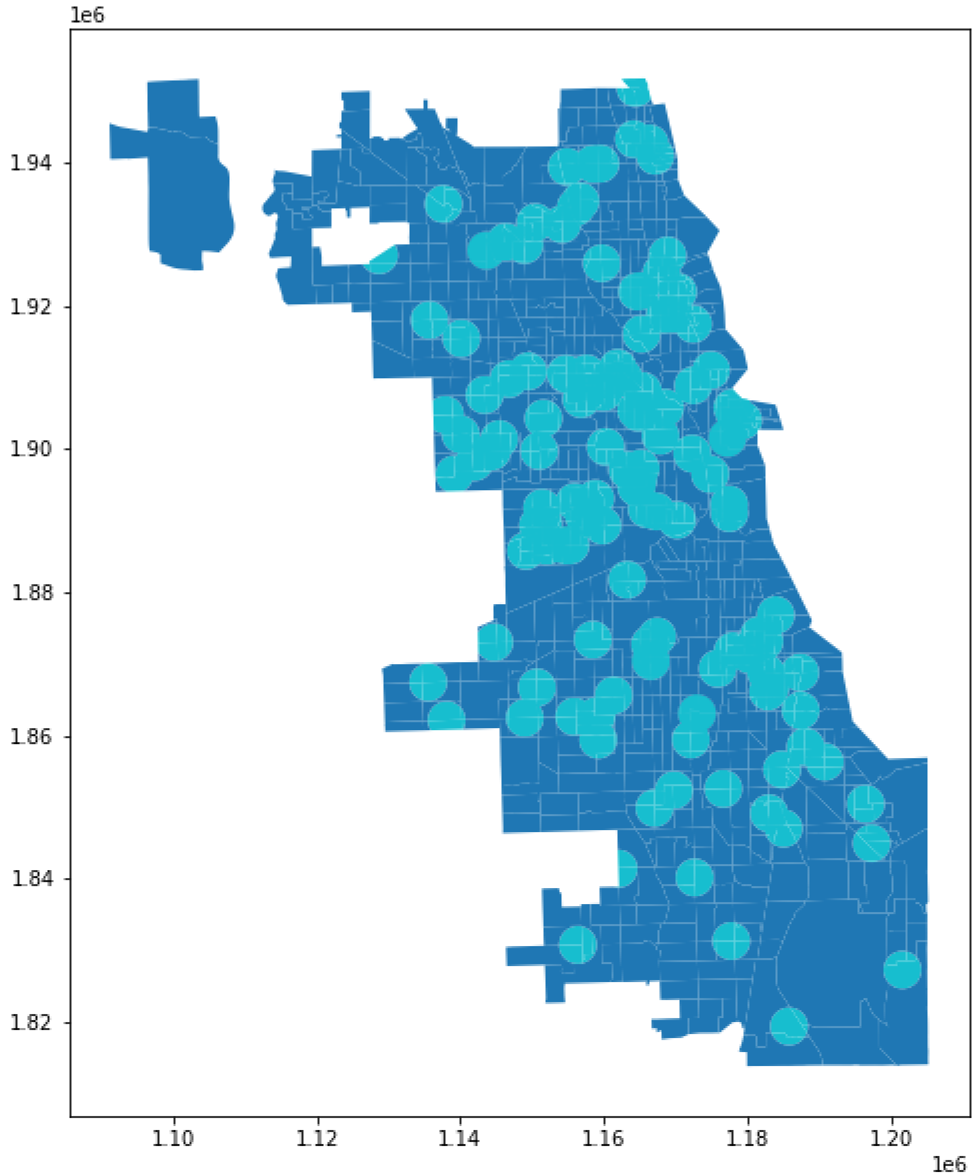
In [43]:
```
near_sites = gpd.overlay(censustractsgdf, buffered_sites, how='intersection')
not_near_sites = gpd.overlay(censustractsgdf, near_sites, how='difference')
```

In [44]:
```
near_sites["near_site"] = "Yes"
not_near_sites["near_site"] = "No"
```

In [45]:
```
merged = near_sites.append(not_near_sites)
```

In [46]:
```
merged.plot(column="near_site", figsize=(15,10))
```

Out[46]:
```
<AxesSubplot:>
```



In [47]:

```
near_site_pctpoc = near_sites["PctPOC"]
not_near_site_pctpoc = not_near_sites["PctPOC"]
frame3 = {"near": near_site_pctpoc, "not_near": not_near_site_pctpoc}
analysisdf = pd.DataFrame(frame3)

df_melt = pd.melt(analysisdf.reset_index(), id_vars=['index'], value_vars=['near', 'not_ne
df_melt = df_melt.drop("index", axis=1)
```

In [48]:

```
df_melt.columns
```

Out[48]:

```
Index(['variable', 'value'], dtype='object')
```

In [49]:

```
averages = df_melt.groupby('variable').mean()
averages
```

Out[49]:

	value
variable	
near	0.556351
not_near	0.590946

In [50]:
```
model = ols('value ~ variable', data=df_melt).fit()
anova_table = sm.stats.anova_lm(model)
anova_table
```

Out[50]:

	df	sum_sq	mean_sq	F	PR(>F)
variable	1.0	0.513134	0.513134	5.094665	0.024119
Residual	1803.0	181.598055	0.100720	NaN	NaN

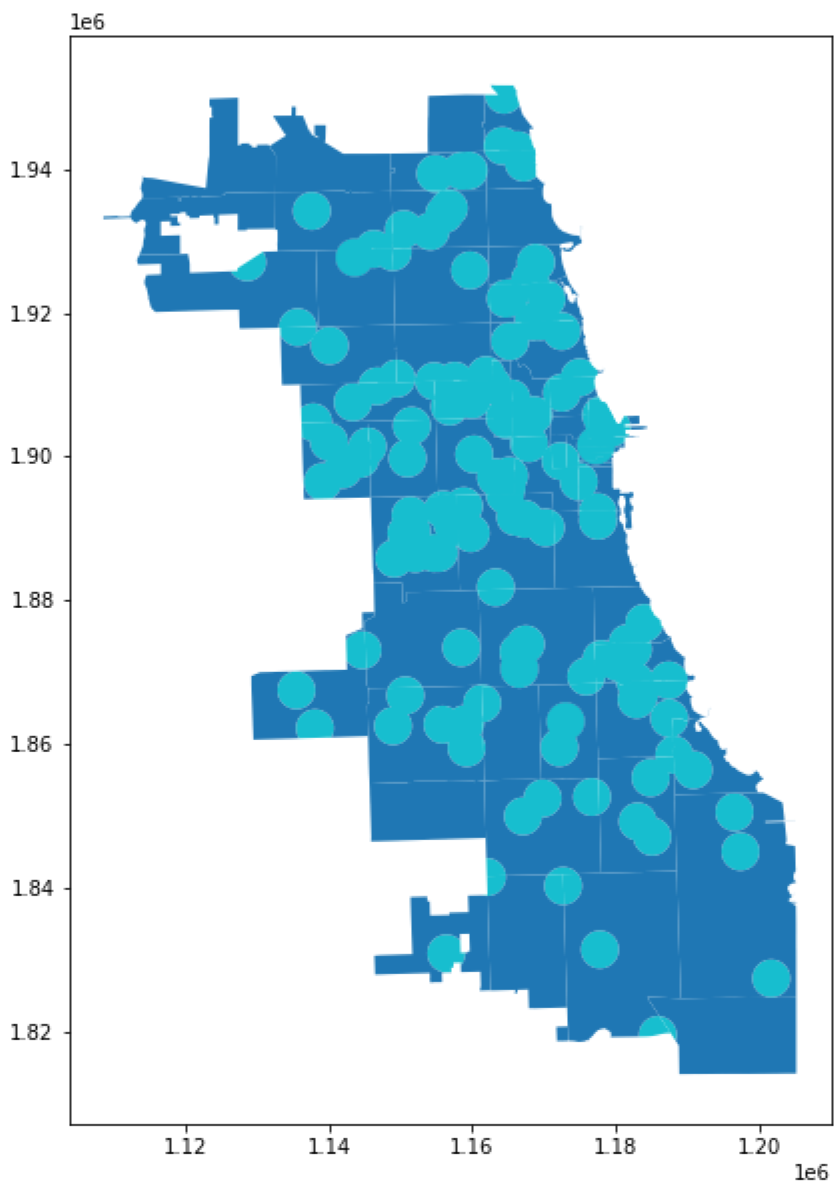
In [51]:
```
near_sites_cases = gpd.overlay(cumulativecasesgdf, buffered_sites, how='intersection')
not_near_sites_cases = gpd.overlay(cumulativecasesgdf, near_sites, how='difference')
```

In [52]:
```
near_sites_cases["near_site"] = "Yes"
not_near_sites_cases["near_site"] = "No"
```

In [53]:
```
merged_cases = near_sites_cases.append(not_near_sites_cases)
```

In [54]:
```
merged_cases.plot(column="near_site", figsize=(15,10))
```

Out[54]: <AxesSubplot:>



In [55]:

```
near_site_cases_pctpoc = near_sites_cases["CasesPer10000"]
not_near_site_pctpoc_cases = not_near_sites_cases["CasesPer10000"]
frame4 = {"near": near_site_cases_pctpoc, "not_near": not_near_site_pctpoc_cases}
analysisdf_cases = pd.DataFrame(frame4)

df_melt_cases = pd.melt(analysisdf_cases.reset_index(), id_vars=['index'], value_vars=['ne
df_melt_cases = df_melt_cases.drop("index", axis=1)
df_melt_cases
```

Out[55]:

	variable	value
0	near	1054.063011
1	near	1030.060804
2	near	1054.063011
3	near	1030.060804
4	near	870.309082
...	...	...
645	not_near	1318.577792
646	not_near	1587.572642
647	not_near	871.123120
648	not_near	818.266110
649	not_near	845.277352

650 rows × 2 columns

In [56]:

```
averages_cases = df_melt_cases.groupby('variable').mean()
averages_cases
```

Out[56]:

	value
variable	
near	1026.245390
not_near	1029.456764

In [57]:

```
model_cases = ols('value ~ variable', data=df_melt_cases).fit()
anova_table_cases = sm.stats.anova_lm(model_cases)
anova_table_cases
```

Out[57]:

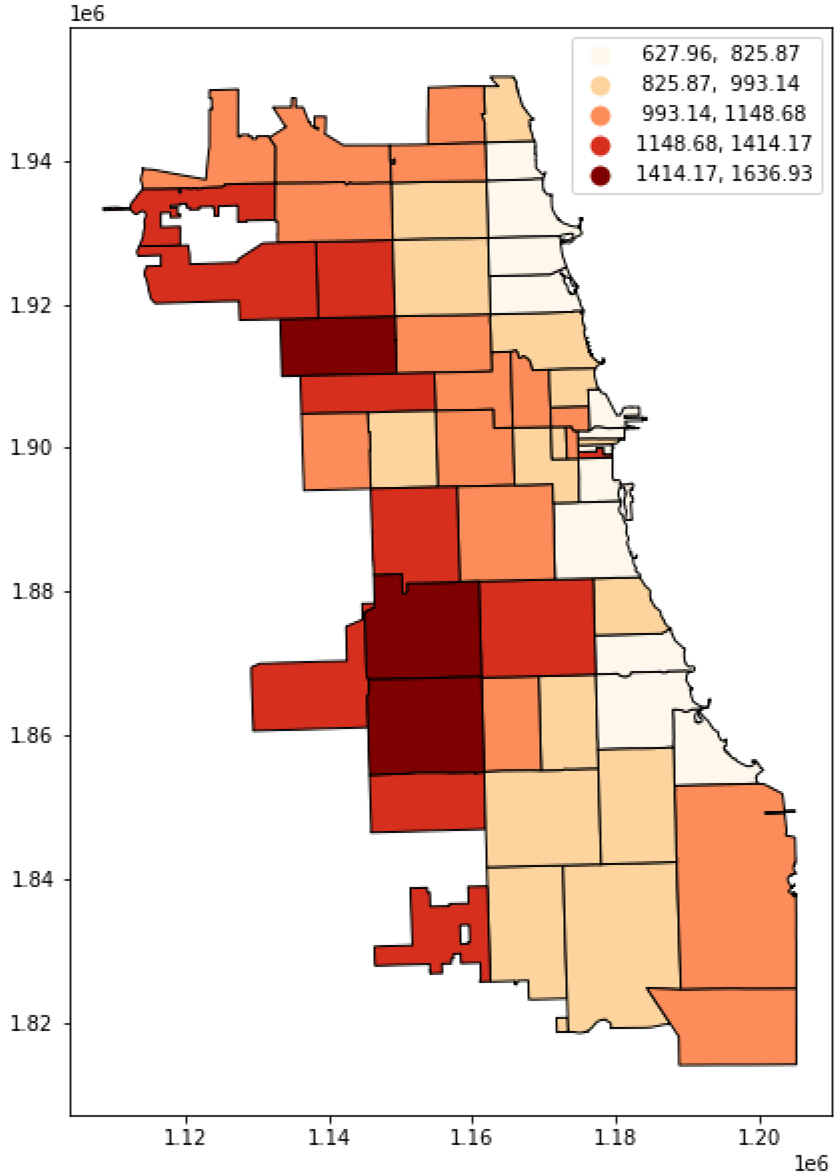
	df	sum_sq	mean_sq	F	PR(>F)
variable	1.0	4.859926e+02	485.992648	0.00959	0.922049
Residual	327.0	1.657140e+07	50677.058446	NaN	NaN

Visualization

In [76]:

```
cumulativecasesgdf.plot(column="CasesPer10000", figsize=(15,10), legend=True, edgecolor="k
```

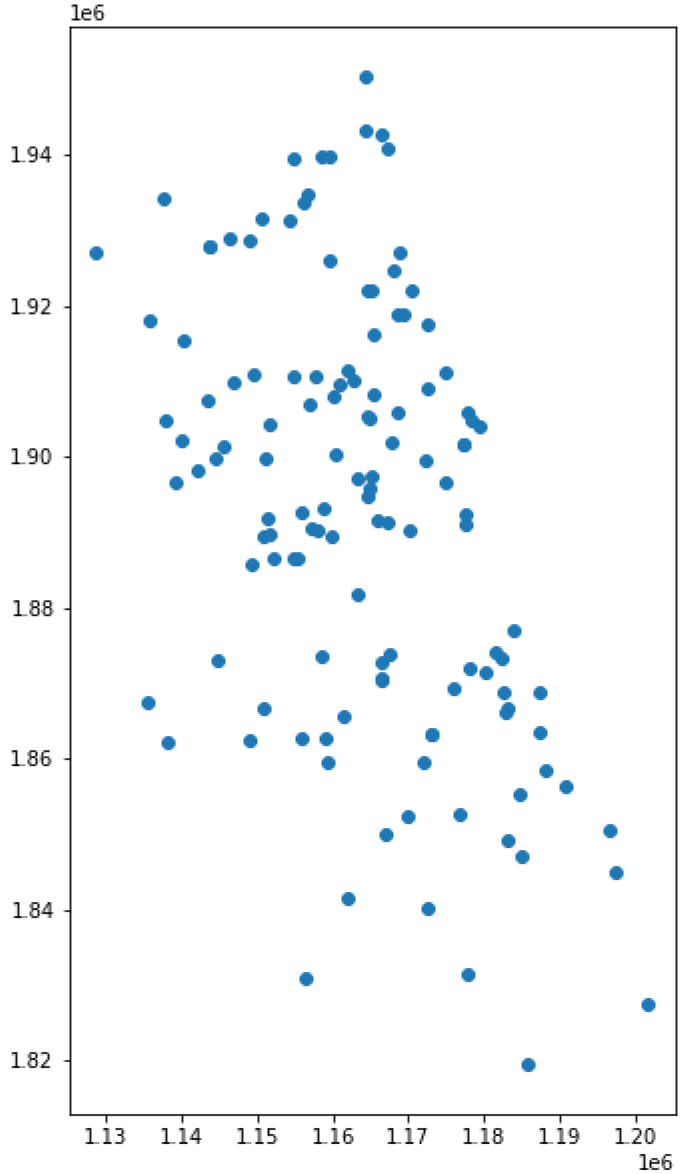
Out[76]: <AxesSubplot:>



In [60]:

```
testingsitesgdf.plot(figsize=(15,10), legend=True)
```

Out[60]: <AxesSubplot:>



In [77]:

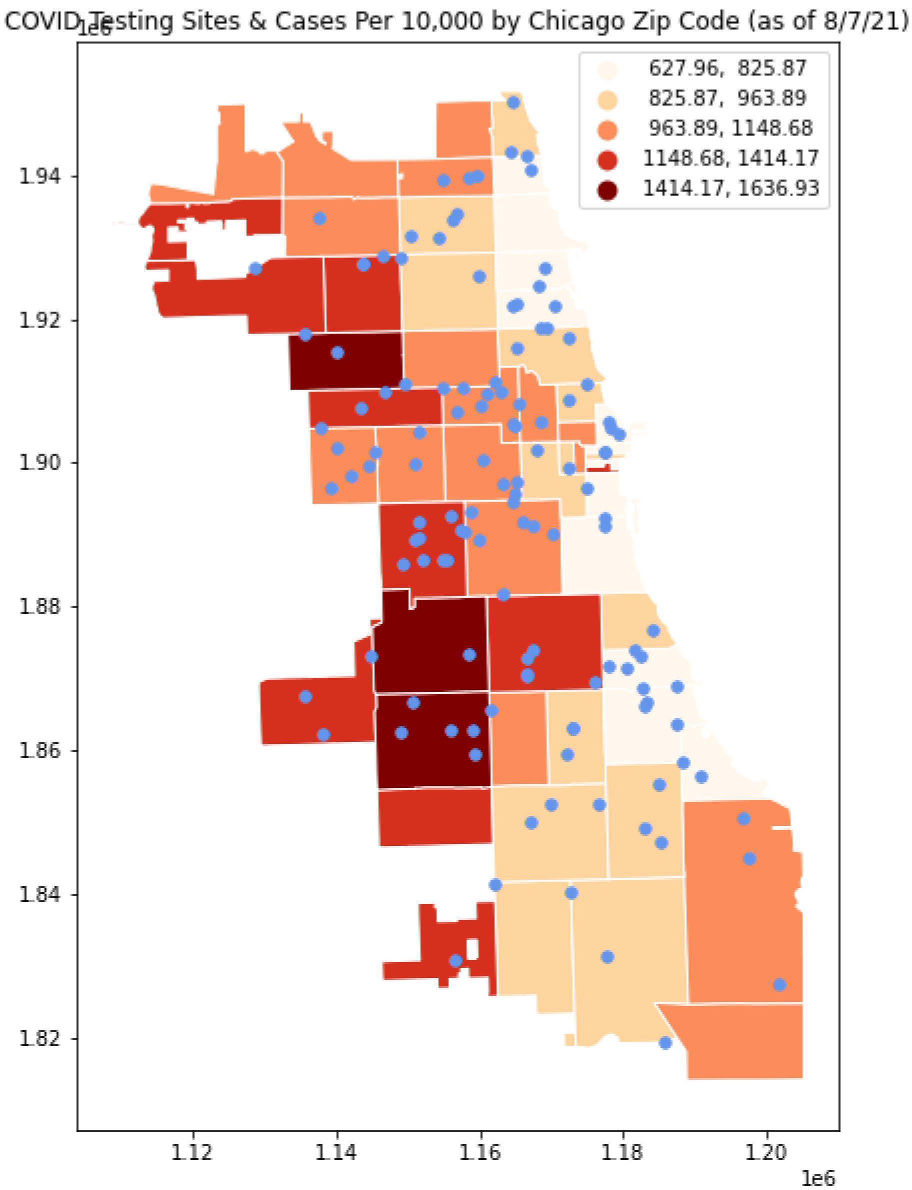
```
fig, ax = plt.subplots(figsize=(15,10))

ax.set_aspect('equal')
ax.set_title('COVID Testing Sites & Cases Per 10,000 by Chicago Zip Code (as of 8/7/21)')

cumulativecasesgdf.plot(ax=ax, column="CasesPer10000", legend=True, cmap="OrRd", scheme="N

testingsitesgdf.plot(ax=ax, marker='o', color='cornflowerblue', markersize=30)

plt.show();
```



In [62]:

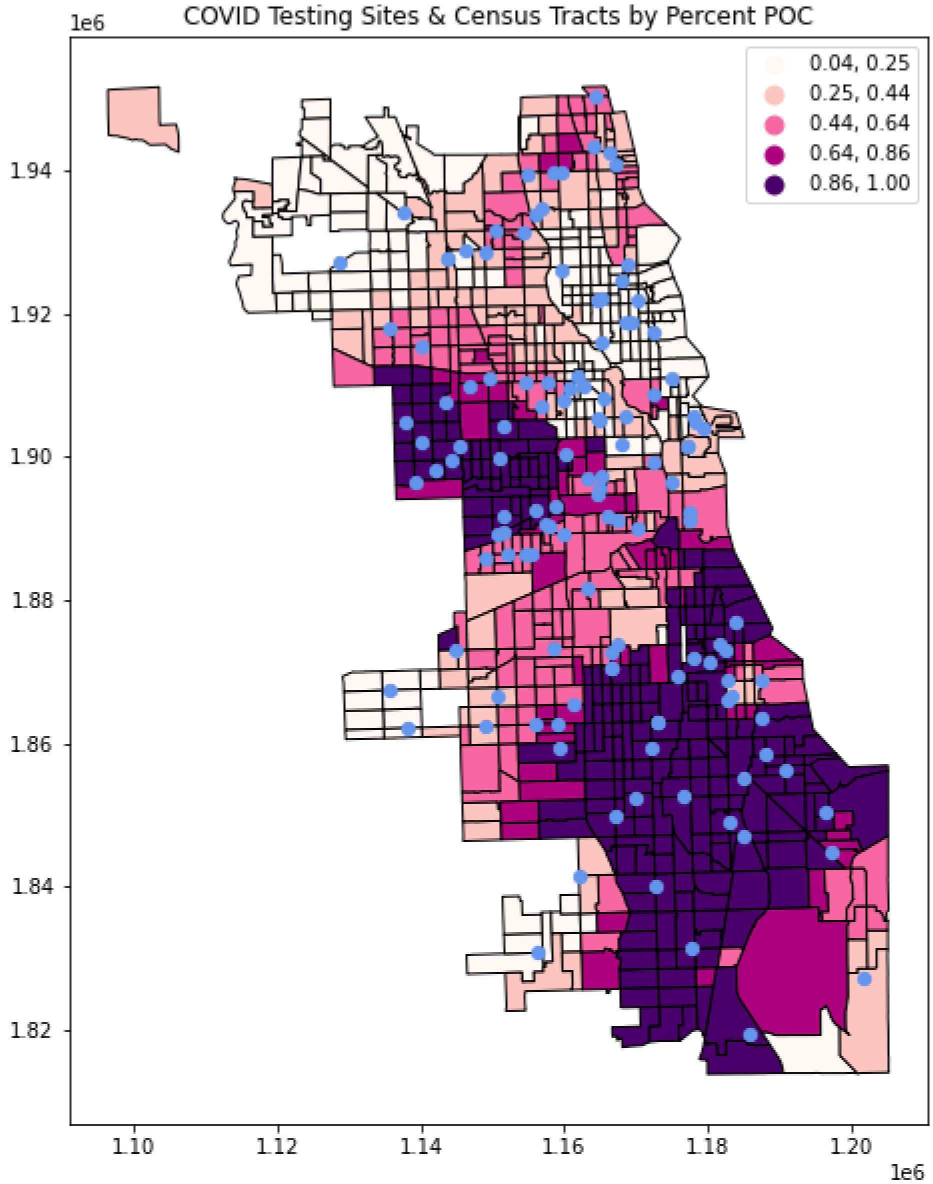
```
fig, ax = plt.subplots(figsize=(15,10))

ax.set_aspect('equal')
ax.set_title('COVID Testing Sites & Census Tracts by Percent POC')

censustractsgdf.plot(ax=ax, column="PctPOC", figsize=(20,15), legend=True, edgecolor="blac
                     cmap="RdPu", scheme="NaturalBreaks")

testingsitesgdf.plot(ax=ax, marker='o', color='cornflowerblue', markersize=40)

plt.show();
```



In [63]:

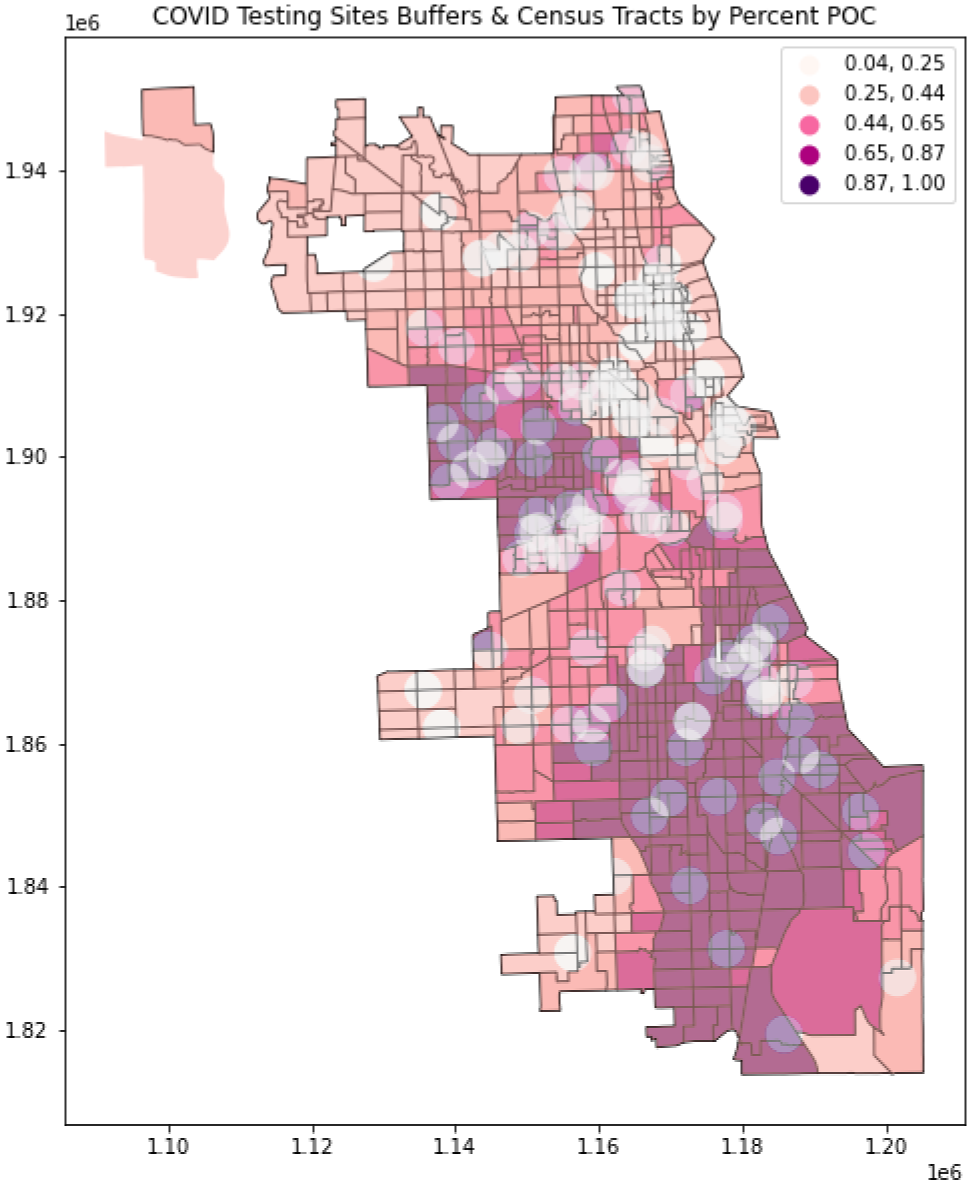
```
fig, ax = plt.subplots(figsize=(15,10))

ax.set_aspect('equal')
ax.set_title('COVID Testing Sites Buffers & Census Tracts by Percent POC')

censustractsgdf.plot(ax=ax, column="PctPOC", legend=True, edgecolor="black", cmap="RdPu",

merged.plot(ax=ax, column="near_site", alpha=0.6, cmap="Pastel1")

plt.show();
```



In [64]:

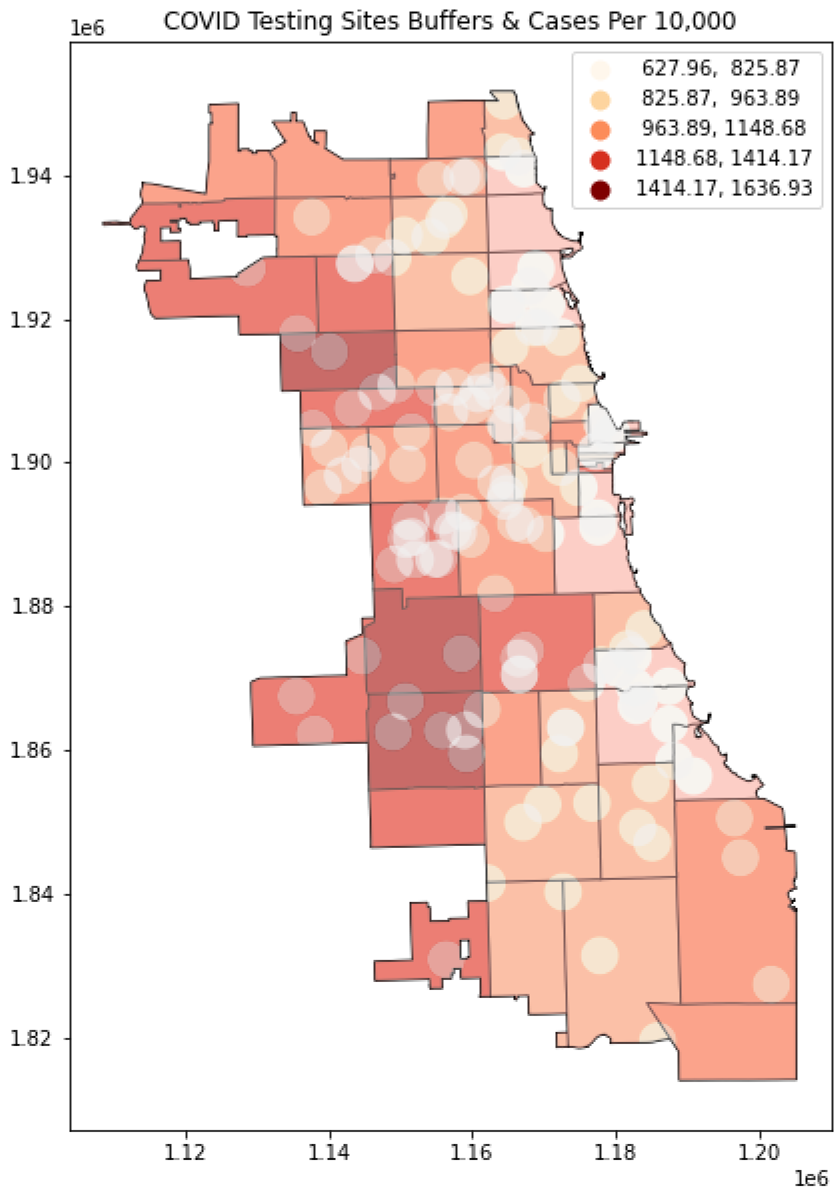
```
fig, ax = plt.subplots(figsize=(15,10))

ax.set_aspect('equal')
ax.set_title('COVID Testing Sites Buffers & Cases Per 10,000')

cumulativecasesgdf.plot(ax=ax, column="CasesPer10000", legend=True, edgecolor="black", cma

merged_cases.plot(ax=ax, column="near_site", alpha=0.6, cmap="Pastel1")

plt.show();
```



Analysis and Results

In [65]:

```
averages
```

Out[65]:

variable	value
near	0.556351
not_near	0.590946

In [66]:

```
anova_table
```

Out[66]:

	df	sum_sq	mean_sq	F	PR(>F)
variable	1.0	0.513134	0.513134	5.094665	0.024119
Residual	1803.0	181.598055	0.100720	NaN	NaN

My results indicate that the areas near Chicago's COVID-19 testing sites (within a half mile) are on average about 4% whiter than areas not near testing sites. This difference is small but statistically significant, with a p-value of 0.024.

In [67]:
```
averages_cases
```

Out[67]:

	value
variable	
near	1026.245390
not_near	1029.456764

In [68]:
```
anova_table_cases
```

Out[68]:

	df	sum_sq	mean_sq	F	PR(>F)
variable	1.0	4.859926e+02	485.992648	0.00959	0.922049
Residual	327.0	1.657140e+07	50677.058446	NaN	NaN

I did not find any statistically significant difference between the COVID case rate near testing sites and not near testing sites. There was an average of just 3 more cases per 10,000 people in the "not near" group, and the p-value of 0.922 clearly indicates no relationship.

Conclusions

A large benefit of this project was the discovery of a statistically significant difference between racial breakdowns near and not near COVID testing sites. I found that areas with accessible testing sites are 4% whiter than areas without accessible testing. This finding is an example of the allocation of health resources being disparate along racial lines, which has of course contributed to disparate impacts of the pandemic.

Though I did not find a statistically significant difference between COVID case rates near and not near testing sites, this finding is not too surprising. It illustrates that, perhaps, Chicago did not purposely add testing sites to areas that had more COVID cases. That may be an area for future work for health officials in cities.

One large limitation of this project was the method of calculating what is "near" a testing site. I wanted to create service areas using network analysis, but the timescale of the project and available open source packages did not allow for this method. Instead, I used half-mile buffers. These buffers can still give us a sense of what is within walking distance, especially since Chicago's sidewalks are close to a uniform grid, but they do not fully encompass what it means for a location to be accessible. People may access COVID-19 testing by driving, biking, using public transportation, getting a ride from a friend, or (now that over the counter rapid tests are available) by ordering pharmacy delivery. I acknowledge this limitation of my analysis but still find it useful to investigate the physical locations of testing sites and the demographics of people who live near them.

Another limitation is the way that I measured COVID distribution across Chicago. The most granular data available was at the zip code level, and it would have been helpful to have data at the Census tract level, especially to stay consistent with my demographic data. Additionally, I chose to use the measure of cumulative cases for each zip code and create a measure of cases-to-date per 10,000. This measure does not reflect fluctuations throughout the pandemic or movements of testing sites throughout the city. It is, however, a snapshot of the cumulative impact of COVID throughout the city.

References

Benitez, J., Courtemanche, C. & Yelowitz, A. (2020). Racial and Ethnic Disparities in COVID-19: Evidence from Six Large Cities. J Econ Race Policy 3, 243–261. https://doi.org/10.1007/s41996-020-00068-9.

Nelson, R., Madron, J., Ayers, N., Winling, L., Marciano, R., Jansen, G., Lee, M., Shah, S., Kending, M. & Connolly, N. D. B. Mapping Inequality: Redlining in New Deal America. University of Richmond's Digital Scholarship Lab. https://dsl.richmond.edu/panorama/redlining/#loc=5/39.1/-94.58.

Onésimo Sandoval, J. S. (2011). Neighborhood Diversity and Segregation in the Chicago Metropolitan Region, 1980-2000. Urban Geography, 32(5), 609-640. https://doi.org/10.2747/0272-3638.32.5.609.