

This is the MVP for my project that analyzes the density of crimes near busy subway stations. Below is a copy of the jupyter notebook that merges the cleaned MTA data with a subway location csv based on the station name. The stations are then plotted using geopandas. The MVP is the final plot of each station with 0.25 mile buffer which will be used to calculate and create a kernel density plot. I will plot the lat/longs of the crimes on top of this and calculate kernel density within the buffers of each station.

In [108...]

```
In [1]: ## Import the packages I need
import geopandas
import matplotlib
import numpy as np
import matplotlib.pyplot as plt
from shapely import wkt
import pandas as pd

%matplotlib inline
```

```
In [2]: ## Import the packages I need
import pandas as pd
import geopandas as gpd
from shapely.geometry import Point
```

```
In [3]: ## Import SQLAlchemy

import sqlalchemy
```

```
In [4]: ## Import SQLAlchemy pac
from sqlalchemy import create_engine
import numpy as np
import pandas as pd
```

```
In [55]: ## Import my shapefile data

subway_stations_locations = pd.read_csv("Stations.csv")
```

```
In [56]: ## Importing a toolbox so I can extend the number of rows visible in the head
from functools import partialmethod
```

```
In [57]: ## Extending the number of rows visible in the head() output
pd.DataFrame.head = partialmethod(pd.DataFrame.head, n=10)
subway_stations_locations.head()
```

```
Out[57]:
```

	Station ID	Complex ID	GTFS Stop ID	Division	Line	Stop Name	Borough	Daytime Routes	Structure	Latitude
0	1	1	R01	BMT	Astoria	Astoria-Ditmars Blvd	Q	N W	Elevated	40.77
1	2	2	R03	BMT	Astoria	Astoria Blvd	Q	N W	Elevated	40.77
2	3	3	R04	BMT	Astoria	30 Av	Q	N W	Elevated	40.76
3	4	4	R05	BMT	Astoria	Broadway	Q	N W	Elevated	40.76
4	5	5	R06	BMT	Astoria	36 Av	Q	N W	Elevated	40.75
5	6	6	R08	BMT	Astoria	39 Av-Dutch Kills	Q	N W	Elevated	40.75
6	7	613	R11	BMT	Astoria	Lexington Av/59 St	M	N W R	Subway	40.76
7	8	8	R13	BMT	Astoria	5 Av/59 St	M	N W R	Subway	40.76
8	9	9	R14	BMT	Broadway - Brighton	57 St-7 Av	M	N Q R W	Subway	40.76
9	10	10	R15	BMT	Broadway - Brighton	49 St	M	N R W	Subway	40.75

```
In [59]: ## Check out my table
##subway_stations_locations.head()
```

```
In [ ]:
```

In [ ]:

In [62]:

```
## Add a new column with the upper case names
subway_stations_locations['NAME_UPPER'] = subway_stations_locations['Stop Name
```

In [63]:

```
## Checking that the names actually changed to upper
subway_stations_locations.head()
```

Out[63]:

	Station ID	Complex ID	GTFS Stop ID	Division	Line	Stop Name	Borough	Daytime Routes	Structure	Latitude
0	1	1	R01	BMT	Astoria	Astoria-Ditmars Blvd	Q	N W	Elevated	40.77
1	2	2	R03	BMT	Astoria	Astoria Blvd	Q	N W	Elevated	40.77
2	3	3	R04	BMT	Astoria	30 Av	Q	N W	Elevated	40.76
3	4	4	R05	BMT	Astoria	Broadway	Q	N W	Elevated	40.76
4	5	5	R06	BMT	Astoria	36 Av	Q	N W	Elevated	40.75
5	6	6	R08	BMT	Astoria	39 Av-Dutch Kills	Q	N W	Elevated	40.75
6	7	613	R11	BMT	Astoria	Lexington Av/59 St	M	N W R	Subway	40.76
7	8	8	R13	BMT	Astoria	5 Av/59 St	M	N W R	Subway	40.76
8	9	9	R14	BMT	Broadway - Brighton	57 St-7 Av	M	N Q R W	Subway	40.76
9	10	10	R15	BMT	Broadway - Brighton	49 St	M	N R W	Subway	40.75

```
In [67]: ## Drop all columns except name and geometry

##subway_stations_locations_line = subway_stations_locations.drop(['name', 'no
subway_stations_locations_line= subway_stations_locations[['NAME_UPPER', 'GTFS
```

```
In [68]: subway_stations_locations_line.head()
```

```
Out[68]:
```

	NAME_UPPER	GTFS Latitude	GTFS Longitude
0	ASTORIA-DITMARS BLVD	40.775036	-73.912034
1	ASTORIA BLVD	40.770258	-73.917843
2	30 AV	40.766779	-73.921479
3	BROADWAY	40.761820	-73.925508
4	36 AV	40.756804	-73.929575
5	39 AV-DUTCH KILLS	40.752882	-73.932755
6	LEXINGTON AV/59 ST	40.762660	-73.967258
7	5 AV/59 ST	40.764811	-73.973347
8	57 ST-7 AV	40.764664	-73.980658
9	49 ST	40.759901	-73.984139

```
In [69]: ## Export this table as a csv
# Write DataFrame to CSV
subway_stations_locations_line.to_csv('tables_to_join/subway_stations_lines_a
```

```
In [123... ## Exported the cleaned datafame of the 50 busiest subway stations from the o
## cleaned the mta data
```

```
In [70]: ## Import the two CSV's to be read in thsi jupyter notebook

df_subway_stations_lines_and_locations = pd.read_csv("subway_stations_lines_a
```

```
In [71]: ## Import the two CSV's to be read in thsi jupyter notebook

df_busiest_subway_stations = pd.read_csv("busiest_subway_stations.csv")
```

```
In [72]: ## Check that jupyter is reading the csvs
df_subway_stations_lines_and_locations.head()
```

```
Out[72]:
```

	NAME_UPPER	GTFS Latitude	GTFS Longitude
0	ASTORIA-DITMARS BLVD	40.775036	-73.912034
1	ASTORIA BLVD	40.770258	-73.917843
2	30 AV	40.766779	-73.921479
3	BROADWAY	40.761820	-73.925508
4	36 AV	40.756804	-73.929575
5	39 AV-DUTCH KILLS	40.752882	-73.932755
6	LEXINGTON AV/59 ST	40.762660	-73.967258
7	5 AV/59 ST	40.764811	-73.973347
8	57 ST-7 AV	40.764664	-73.980658
9	49 ST	40.759901	-73.984139

```
In [73]: ## Check that jupyter is reading the csvs
df_busiest_subway_stations.head()
```

```
Out[73]:
```

	STATION	DAILY_ENTRIES
0	34 ST-PENN STA	5487455.0
1	34 ST-HERALD SQ	4187187.0
2	FLUSHING-MAIN	3974386.0
3	GRD CNTRL-42 ST	3886202.0
4	23 ST	3556540.0
5	EUCLID AV	3462777.0
6	FULTON ST	3355790.0
7	96 ST	3320869.0
8	125 ST	3223380.0
9	86 ST	3168741.0

```
In [74]: ## Do a left join the tables with mta to identify which names need to be repl

merged_left = pd.merge(left=df_subway_stations_lines_and_locations,
                        right=df_busiest_subway_stations,
                        how='left', left_on='NAME_UPPER', right_on='STATION')

merged_left
```

```
Out[74]:
```

	NAME_UPPER	GTFS Latitude	GTFS Longitude	STATION	DAILY_ENTRIES
0	ASTORIA-DITMARS BLVD	40.775036	-73.912034	NaN	NaN
1	ASTORIA BLVD	40.770258	-73.917843	ASTORIA BLVD	356863.0
2	30 AV	40.766779	-73.921479	30 AV	556075.0
3	BROADWAY	40.761820	-73.925508	BROADWAY	701403.0
4	36 AV	40.756804	-73.929575	36 AV	279020.0
...	...	...	...	...	...
491	PRINCE'S BAY	40.525507	-74.200064	NaN	NaN
492	PLEASANT PLAINS	40.522410	-74.217847	NaN	NaN
493	RICHMOND VALLEY	40.519631	-74.229141	NaN	NaN
494	TOTTENVILLE	40.512764	-74.251961	NaN	NaN
495	ARTHUR KILL	40.516578	-74.242096	NaN	NaN

496 rows × 5 columns

```
In [75]: ## Create a new dataframe with only the rows where the joining worked

Successfully_merged_names=merged_left.dropna(axis=0)
```

```
In [130... ##The mta data statio names do not have the integer strings "th, nd, etc." Fo
## 49 ST not 49TH ST like it is in the other dataset. That is why we are gett
```

```
In [76]: Successfully_merged_names
```

Out[76]:

	NAME_UPPER	GTFS Latitude	GTFS Longitude	STATION	DAILY_ENTRIES
1	ASTORIA BLVD	40.770258	-73.917843	ASTORIA BLVD	356863.0
2	30 AV	40.766779	-73.921479	30 AV	556075.0
3	BROADWAY	40.761820	-73.925508	BROADWAY	701403.0
4	36 AV	40.756804	-73.929575	36 AV	279020.0
7	5 AV/59 ST	40.764811	-73.973347	5 AV/59 ST	506209.0
...	...	...	...	...	...
469	TIMES SQ-42 ST	40.755983	-73.986229	TIMES SQ-42 ST	2542974.0
472	96 ST	40.784318	-73.947152	96 ST	3320869.0
473	86 ST	40.777891	-73.951787	86 ST	3168741.0
474	72 ST	40.768799	-73.958424	72 ST	1534080.0
476	TOMPKINSVILLE	40.636949	-74.074835	TOMPKINSVILLE	33156.0

353 rows × 5 columns

In [85]:

```
Successfully_merged_names.sort_values('DAILY_ENTRIES')
```

Out[85]:

	NAME_UPPER	GTFS Latitude	GTFS Longitude	STATION	DAILY_ENTRIES
199	BROAD CHANNEL	40.608382	-73.815925	BROAD CHANNEL	13723.0
202	BEACH 105 ST	40.583209	-73.827559	BEACH 105 ST	14276.0
201	BEACH 98 ST	40.585307	-73.820558	BEACH 98 ST	21811.0
476	TOMPKINSVILLE	40.636949	-74.074835	TOMPKINSVILLE	33156.0
206	BEACH 44 ST	40.592943	-73.776013	BEACH 44 ST	43790.0
...	...	...	...	...	...
228	23 ST	40.742878	-73.992821	23 ST	3556540.0
405	23 ST	40.739864	-73.986599	23 ST	3556540.0
13	23 ST	40.741303	-73.989344	23 ST	3556540.0
11	34 ST-HERALD SQ	40.749567	-73.987950	34 ST-HERALD SQ	4187187.0
227	34 ST-HERALD SQ	40.749719	-73.987823	34 ST-HERALD SQ	4187187.0

353 rows × 5 columns

In [95]:

```
## Rename the GTFS Lat/Long columns to headers with no space.

Successfully_merged_names.rename(columns={'GTFS Latitude': 'Latitude',
                                         'GTFS Longitude': 'Longitude'},
                                inplace=True, errors='raise')
```

/Users/rachelbryn/opt/anaconda3/envs/geo\_env/lib/python3.9/site-packages/pandas/core/frame.py:4441: 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](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
return super().rename()

In [96]:

```
Successfully_merged_names
```

Out[96]:

	NAME_UPPER	Latitude	Longitude	STATION	DAILY_ENTRIES
1	ASTORIA BLVD	40.770258	-73.917843	ASTORIA BLVD	356863.0
2	30 AV	40.766779	-73.921479	30 AV	556075.0
3	BROADWAY	40.761820	-73.925508	BROADWAY	701403.0
4	36 AV	40.756804	-73.929575	36 AV	279020.0
7	5 AV/59 ST	40.764811	-73.973347	5 AV/59 ST	506209.0
...	...	...	...	...	...
469	TIMES SQ-42 ST	40.755983	-73.986229	TIMES SQ-42 ST	2542974.0
472	96 ST	40.784318	-73.947152	96 ST	3320869.0
473	86 ST	40.777891	-73.951787	86 ST	3168741.0
474	72 ST	40.768799	-73.958424	72 ST	1534080.0
476	TOMPKINSVILLE	40.636949	-74.074835	TOMPKINSVILLE	33156.0

353 rows × 5 columns

In [98]:

```
## create the geodataframe from the dataframe
## https://geopandas.org/gallery/create_geopandas_from_pandas.html

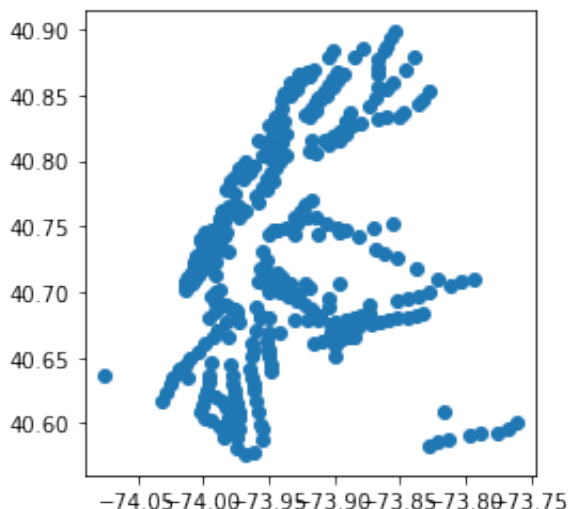
test_gdf = geopandas.GeoDataFrame(
    Successfully_merged_names, geometry=geopandas.points_from_xy(Successfully_
```

In [101]:

```
# We can now plot our GeoDataFrame of the 353 stations with corresponding loc
test_gdf.plot()
```



Out[101... &lt;AxesSubplot:&gt;

In [114... `import contextily as ctx`In [111... `from shapely import wkt`  
`import pandas as pd`In [102... `## Create a buffer polyhon list`  
`buf = test_gdf.buffer(5)`  
`buf.head()`Out[102... 

```
1    POLYGON ((-68.91784 40.77026, -68.94192 40.280...
2    POLYGON ((-68.92148 40.76678, -68.94556 40.276...
3    POLYGON ((-68.92551 40.76182, -68.94958 40.271...
4    POLYGON ((-68.92957 40.75680, -68.95365 40.266...
7    POLYGON ((-68.97335 40.76481, -68.99742 40.274...
dtype: geometry
```

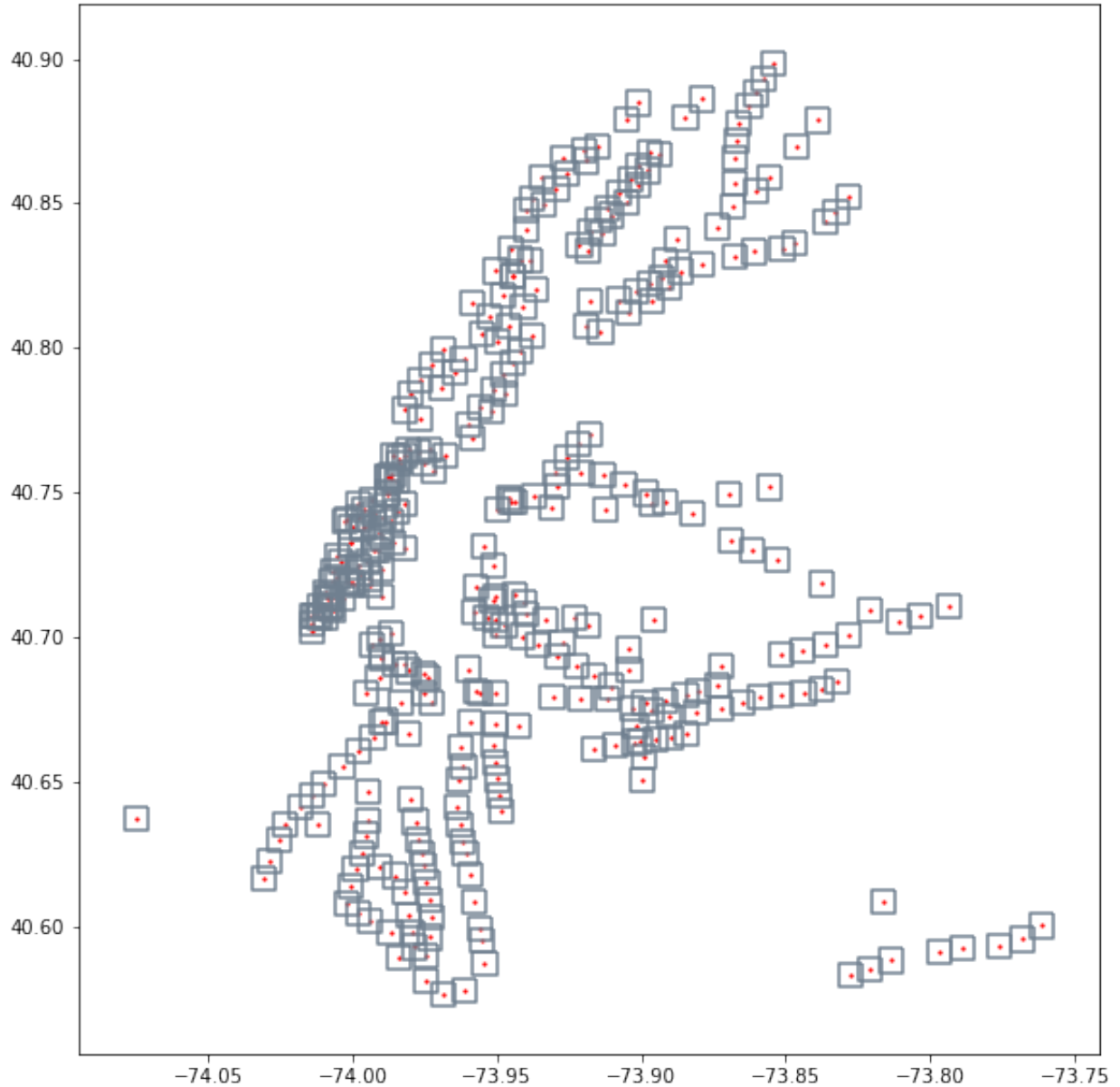
In [123... `# Buffer the points using a square cap style`  
`# Note cap_style: round = 1, flat = 2, square = 3`  
`buffer = test_gdf.buffer(0.00405, cap_style = 3)`  
`## NOTE: for "Attempt_to_map.buffer(0.00400, cap_style = 3)" the 0.008 is the`  
`## 0.001 = 111 meters, so 0.004 is equal to 400 meters or about 0.25 miles.`  
  
`# It doesn't come with a CRS because it's a CSV, so let's`  
`# say "hey, let's use the standard shape of the earth etc"`  
`test_gdf.crs = {'init': 'epsg:3857'}`  
  
`# Plot the results`  
`fig, ax1 = plt.subplots(1, figsize=(10,10))`  
`buffer.boundary.plot(ax=ax1, color = 'slategrey')`  
`test_gdf.plot(ax = ax1, color = 'red', markersize= 1.5)`

```

/Users/rachelbryn/opt/anaconda3/envs/geo_env/lib/python3.9/site-packages/pyproj/crs/crs.py:53: FutureWarning: '+init=<authority>:<code>' syntax is deprecated. '<authority>:<code>' is the preferred initialization method. When making the change, be mindful of axis order changes: https://pyproj4.github.io/pyproj/s
table/gotchas.html#axis-order-changes-in-proj-6
    return _prepare_from_string(" ".join(pjargs))

```

Out[123... <AxesSubplot:>



In [124... *## Add a basemap to my plot above. The NYC borough map is built into Geopandas*

In [113... *## Set the dataframe CRS*

In [112]:

In [ ]:

In [ ]:

In [ ]:

In [99]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [2]:

```
## Using isna() to select all rows with NaN under an entire DataFrame to expl  
## not get joined  
##merged_left[merged_left.isna().any(axis=1)].head()
```

In [ ]:

```
## open the database with the subway station locations/lat/long (which was do  
## for some of the names in the null df above to see how they may differ.
```

In [ ]:

```
## Replace/edit names
```

```
In [ ]: ##Now save this nice clean table to the same database as the mta data
```

```
In [ ]: ## Join tables on NAME column
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```