In [21]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import datasets
import seaborn as sns
from sqlalchemy import create_engine
import datetime
import geopandas as gpd
import geoplot as gplt
import contextily as ctx
from shapely.geometry import Point, Polygon
%matplotlib inline
```

In [22]:
```
engine = create_engine("sqlite:///stations.db")
```

In [23]:
```
engine2 = create_engine("sqlite:///mta_sample.db")
```

In [24]:
```
mta_data = pd.read_sql('SELECT SUM(ENTRIES), STATION FROM sample_table GROUP BY STATION ORDER BY SUM(ENTRIES) c
```

I am using two databases at the moment, one that contains a list of MTA stations with their latitude and longitude coordinates, and the other is the provided MTA dataset required for this project.

I organized the station entries below in descending order to know which have the most traffic. I used that information to filter out the lat/long coordinates of those stations within the dataset that contains the lat/long coordinates.

I have been attempting to map the stations using these coordinates, but I have been unable to produce a basemap thus far. My intention for the MVP was to show two maps, side by side. One map would show the locations of the subway stations with the highest traffic, and the other map would show the neighborhoods with the highest covid rates. I was hoping to provide a clear visual of whether or not there is a correlation between the foot traffic around these stations and the rate of covid infection -- which would ultimately provide the imaginary insurance company with a better idea of the impact subway travel has on the rates of infection.

In [25]:
```
mta_data.head()
```

Out[25]:

	SUM(ENTRIES)	STATION
0	360940318341	DEKALB AV
1	310478108930	42 ST-PORT AUTH
2	282166007877	125 ST
3	244912387597	TIMES SQ-42 ST
4	237764444181	23 ST

In [26]:
```
geo_data = pd.read_sql('SELECT STATION,LAT,LONG FROM stations_table;', engine)
```

In [27]:
```
geo_data.head()
```

Out[27]:

	STATION	LAT	LONG
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

In [28]:
```
top_stations = pd.read_sql("SELECT STATION, LAT, LONG FROM stations_table WHERE STATION IN ('DeKalb Av','42 St-
```

In [29]:
```
def make_point(row):
    return Point(row.LAT, row.LONG)
```

In [30]:
```
points = top_stations.apply(make_point, axis=1)
```

In [31]:
```
stationgeodf = gpd.GeoDataFrame(top_stations, geometry=points)
```

In [32]:
```
stationgeodf.crs = {'init': 'epsg:4326'}
```

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

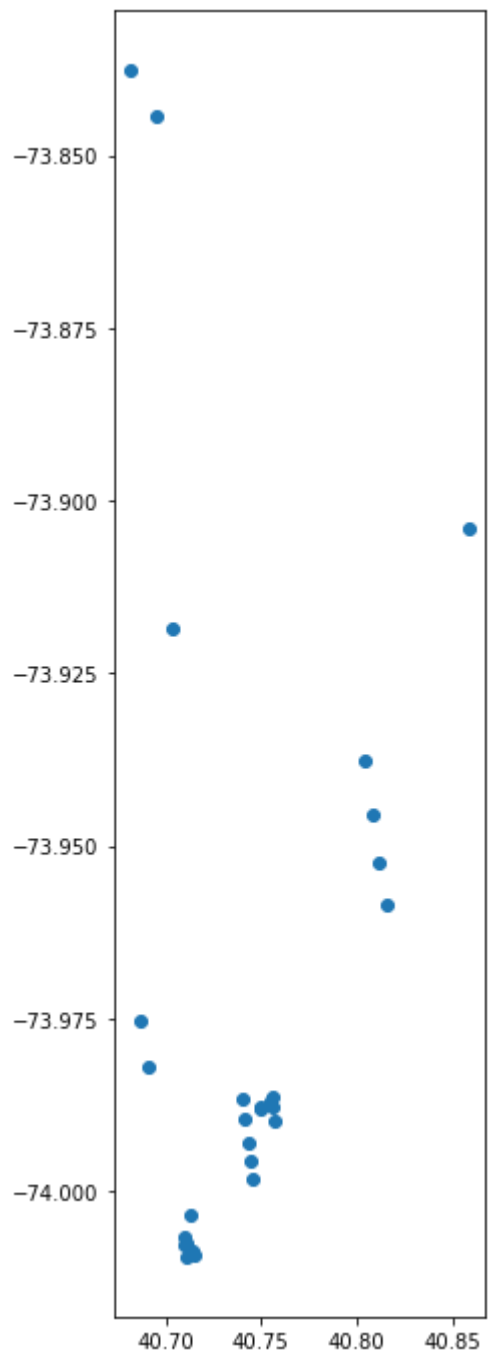
In [33]:
```
stationgeodf.head()
```

Out[33]:

	STATION	LAT	LONG	geometry
0	Times Sq-42 St	40.754672	-73.986754	POINT (40.75467 -73.98675)
1	34 St-Herald Sq	40.749567	-73.987950	POINT (40.74957 -73.98795)
2	23 St	40.741303	-73.989344	POINT (40.74130 -73.98934)
3	DeKalb Av	40.690635	-73.981824	POINT (40.69064 -73.98182)
4	104 St	40.695178	-73.844330	POINT (40.69518 -73.84433)

In [34]:
```
stationgeodf.plot(figsize=(12,12))
```

Out[34]:
```
<AxesSubplot:>
```



In [ ]:
```

```