Notebook3_Midterm Data Exploration and Analysis-Transit System

February 22, 2021

1 Notebook 3

Project:" Intra-Regional Migration and Transportation in New York Metro Area"

Due to the large data our team is working with, there are a total of four notebooks submitted for this midterm (two from each team member)

I also outlined the notebook into the Table of Content

2 Research Questions

- **In this notebook: What's the transit network look alike in New York Metro Area?**
  - **Expected Exploration:**
    - We expect to combine three transit rail network shapefiles to create a regional transit line map, and we expect to combine three transit station network shapefiles to create a regional transit station map on New York Metro Region.
    - We expect to create an interactive map with the regional transit stations.
  - **Purpose of this notebook:**
    - In this notebook, I first introduced research questions regarding transit network and data sources, and I conducted data explorationa and analysis of transit network in NYMA. I combined and mapped the transit stations and lines of three dominating transit systems, including the New Jersey Rail, Long Island Railroad and Metro-North Railroad. We also created maps for transit density in New York Metro Area (density=number of stations in each county/county land area).
- **We are expecting to analyze and answer the following research questions in the next few weeks after midterm:**
  - **Q1: What's the transit density of each county in the New York Metro Area?**
  - **Q2: Are counties with higher transit density popular migration destinations in the New York Metro Area?**

3 Data source

1. NJ Rail Lines: https://njogis-newjersey.opendata.arcgis.com/datasets/passenger-railroad-lines-in-nj
2. NJ Rail Stations: https://njogis-newjersey.opendata.arcgis.com/datasets/railroad-stations-in-nj
3. LIRR: https://catalog.data.gov/es_AR/dataset/long-island-railroad-map
4. Metro North Lines: https://maps.princeton.edu/catalog/nyu-2451-34755
5. Metro North Stations: https://maps.princeton.edu/catalog/nyu-2451-34756.

4 Importing libraries

[1]:
```
import geopandas as gpd
import matplotlib.pyplot as plt
import plotly.express as px
import pandas as pd
```

5 Data exploration

5.1 Data exploration for transit lines

[2]:
```
njrail=gpd.read_file('NJRail_line/Passenger_Railroad_Lines_in_NJ.shp')
njstation=gpd.read_file('NJRail_station/Railroad_Stations_in_NJ.shx')
lirail=gpd.read_file('nyu-2451-34753-geojson.json')
listation=gpd.read_file('nyu-2451-34754-geojson.json')
mnrail=gpd.read_file('MNStation/mnline.json')
mnstation=gpd.read_file('MNStation/stops.json')
```

[3]:
```
#looking at first 5 rows of the njrail, lirail and mnrail datasets.
njrail.head()
```

[3]:

	OBJECTID	RAIL_LINE	SERVICE	Shape_Leng	DATE_STAMP	geometry
0	1	ATLANTIC CITY RAIL LINE	None	356957.019400	2016-08-30	LINESTRING (508669.853 193016.598, 505026.387 …
1	2	BERGEN COUNTY LINE	HOBOKEN	155780.575084	2013-11-04	LINESTRING (622908.638 692949.426, 620720.513 …
2	3	MAIN LINE	HOBOKEN	161721.051881	2013-11-04	LINESTRING (587611.064 830753.567, 588462.132 …
3	4	MEADOWLANDS RAIL LINE	None	56328.562168	2013-11-04	LINESTRING (622908.638 692949.426, 620720.513 …
4	5	MONTCLAIR BOONTON LINE	NEW YORK CITY	328910.733006	2013-11-04	LINESTRING (399357.761 735253.132, 399513.899 …

[4]:
```
lirail.head()
```

[4]:
```
                  id  route_id      route_long  \
0  nyu_2451_34753.1        11         Belmont
1  nyu_2451_34753.2        10  Port Jefferson
2  nyu_2451_34753.3        12       City Zone
3  nyu_2451_34753.4         1         Babylon
4  nyu_2451_34753.5         3      Oyster Bay

                                            geometry
0  MULTILINESTRING ((-73.99309 40.75074, -73.9924…
1  MULTILINESTRING ((-73.90300 40.74607, -73.9034…
2  MULTILINESTRING ((-73.80933 40.69955, -73.8100…
3  MULTILINESTRING ((-73.99309 40.75074, -73.9924…
4  MULTILINESTRING ((-73.99309 40.75074, -73.9924…
```

[5]:
```
mnrail.head()
```

[5]:
```
                  id  route_id  route_long  \
0  nyu_2451_34755.1         1      Hudson
1  nyu_2451_34755.2         3   New Haven
2  nyu_2451_34755.3         2      Harlem
3  nyu_2451_34755.4         5     Danbury
4  nyu_2451_34755.5         4  New Canaan

                                            geometry
0  MULTILINESTRING ((-73.93795 41.70584, -73.9472…
1  MULTILINESTRING ((-72.92175 41.30498, -72.9282…
2  MULTILINESTRING ((-73.56220 41.81472, -73.5582…
3  MULTILINESTRING ((-73.45016 41.39636, -73.4181…
4  MULTILINESTRING ((-73.49563 41.14630, -73.4981…
```

As you can see from the first few rows of the three rail line datasets, they all have different columns, however, they all have a few things in common, including the name of each rail line and geometry, but those two columns in the three datasets have different names, so I need to change column names to merge them together into one new dataframe.

[6]:
```
list(njrail)
```

[6]:
```
['OBJECTID', 'RAIL_LINE', 'SERVICE', 'Shape_Leng', 'DATE_STAMP', 'geometry']
```

[7]:
```
njrail.columns=['id','linename','service','shape_leng','date_stamp','geometry']
njrail.head()
```

[7]:
```
   id                 linename        service     shape_leng  date_stamp  \
0   1  ATLANTIC CITY RAIL LINE           None  356957.019400  2016-08-30
1   2       BERGEN COUNTY LINE       HOBOKEN  155780.575084  2013-11-04
2   3                MAIN LINE       HOBOKEN  161721.051881  2013-11-04
3   4    MEADOWLANDS RAIL LINE           None   56328.562168  2013-11-04
```

```
4   5   MONTCLAIR BOONTON LINE  NEW YORK CITY  328910.733006  2013-11-04

                                                 geometry
0  LINESTRING (508669.853 193016.598, 505026.387 …
1  LINESTRING (622908.638 692949.426, 620720.513 …
2  LINESTRING (587611.064 830753.567, 588462.132 …
3  LINESTRING (622908.638 692949.426, 620720.513 …
4  LINESTRING (399357.761 735253.132, 399513.899 …
```

[8]:
```
#Let's add a new column called "operating" to the mnrail dataframe to␣
 ↪distinguish it from two other rail line dataframes.
njrail['Operating'] = 'New Jersey Railroad'
njrail.head()
```

[8]:
```
   id                 linename        service     shape_leng  date_stamp  \
0   1  ATLANTIC CITY RAIL LINE           None  356957.019400  2016-08-30
1   2       BERGEN COUNTY LINE        HOBOKEN  155780.575084  2013-11-04
2   3                MAIN LINE        HOBOKEN  161721.051881  2013-11-04
3   4    MEADOWLANDS RAIL LINE           None   56328.562168  2013-11-04
4   5   MONTCLAIR BOONTON LINE  NEW YORK CITY  328910.733006  2013-11-04

                                                 geometry            Operating
0  LINESTRING (508669.853 193016.598, 505026.387 …  New Jersey Railroad
1  LINESTRING (622908.638 692949.426, 620720.513 …  New Jersey Railroad
2  LINESTRING (587611.064 830753.567, 588462.132 …  New Jersey Railroad
3  LINESTRING (622908.638 692949.426, 620720.513 …  New Jersey Railroad
4  LINESTRING (399357.761 735253.132, 399513.899 …  New Jersey Railroad
```

[9]:
```
list(lirail)
```

[9]:
```
['id', 'route_id', 'route_long', 'geometry']
```

[10]:
```
lirail.columns=['id','number','linename','geometry']
lirail.head()
```

[10]:
```
                id  number        linename  \
0  nyu_2451_34753.1      11         Belmont
1  nyu_2451_34753.2      10  Port Jefferson
2  nyu_2451_34753.3      12       City Zone
3  nyu_2451_34753.4       1         Babylon
4  nyu_2451_34753.5       3      Oyster Bay

                                                 geometry
0  MULTILINESTRING ((-73.99309 40.75074, -73.9924…
1  MULTILINESTRING ((-73.90300 40.74607, -73.9034…
2  MULTILINESTRING ((-73.80933 40.69955, -73.8100…
3  MULTILINESTRING ((-73.99309 40.75074, -73.9924…
```

```
4  MULTILINESTRING ((-73.99309 40.75074, -73.9924…
```

[11]:
```
#Let's add a new column called "operating" to the mnrail dataframe to␣
↪distinguish it from two other rail line dataframes.
lirail['Operating'] = 'Long Island Railroad'
lirail.head()
```

[11]:
```
                 id  number        linename  \
0  nyu_2451_34753.1      11         Belmont
1  nyu_2451_34753.2      10  Port Jefferson
2  nyu_2451_34753.3      12       City Zone
3  nyu_2451_34753.4       1         Babylon
4  nyu_2451_34753.5       3      Oyster Bay

                                            geometry             Operating
0  MULTILINESTRING ((-73.99309 40.75074, -73.9924…  Long Island Railroad
1  MULTILINESTRING ((-73.90300 40.74607, -73.9034…  Long Island Railroad
2  MULTILINESTRING ((-73.80933 40.69955, -73.8100…  Long Island Railroad
3  MULTILINESTRING ((-73.99309 40.75074, -73.9924…  Long Island Railroad
4  MULTILINESTRING ((-73.99309 40.75074, -73.9924…  Long Island Railroad
```

[12]:
```
list(mnrail)
```

[12]:
```
['id', 'route_id', 'route_long', 'geometry']
```

[13]:
```
mnrail.columns=['id','number','linename','geometry']
mnrail.head()
```

[13]:
```
                 id  number    linename  \
0  nyu_2451_34755.1       1      Hudson
1  nyu_2451_34755.2       3   New Haven
2  nyu_2451_34755.3       2      Harlem
3  nyu_2451_34755.4       5     Danbury
4  nyu_2451_34755.5       4  New Canaan

                                            geometry
0  MULTILINESTRING ((-73.93795 41.70584, -73.9472…
1  MULTILINESTRING ((-72.92175 41.30498, -72.9282…
2  MULTILINESTRING ((-73.56220 41.81472, -73.5582…
3  MULTILINESTRING ((-73.45016 41.39636, -73.4181…
4  MULTILINESTRING ((-73.49563 41.14630, -73.4981…
```

Now we have changed the names of shared columns (names and geometry) in three dataframe to the same names.

I also noticed that in the geometry columns of the three transit line datasets that they do not use the same geometry coordination system, so I'm going to change them to share the same coordination system.

[14]:
```
lirail=lirail.to_crs('epsg:3424')
lirail.head()
```

[14]:

	id	number	linename	geometry	Operating
0	nyu_2451_34753.1	11	Belmont	MULTILINESTRING ((632561.894 698796.549, 63274…	Long Island Railroad
1	nyu_2451_34753.2	10	Port Jefferson	MULTILINESTRING ((657530.427 697249.579, 65739…	Long Island Railroad
2	nyu_2451_34753.3	12	City Zone	MULTILINESTRING ((683616.747 680493.214, 68342…	Long Island Railroad
3	nyu_2451_34753.4	1	Babylon	MULTILINESTRING ((632561.894 698796.549, 63274…	Long Island Railroad
4	nyu_2451_34753.5	3	Oyster Bay	MULTILINESTRING ((632561.894 698796.549, 63274…	Long Island Railroad

[15]:
```
mnrail=mnrail.to_crs('epsg:3424')
mnrail.head()
```

[15]:

	id	number	linename	geometry
0	nyu_2451_34755.1	1	Hudson	MULTILINESTRING ((645588.804 1046859.351, 6433…
1	nyu_2451_34755.2	3	New Haven	MULTILINESTRING ((925725.242 904249.608, 92398…
2	nyu_2451_34755.3	2	Harlem	MULTILINESTRING ((747752.029 1087426.164, 7489…
3	nyu_2451_34755.4	5	Danbury	MULTILINESTRING ((780145.485 935346.610, 78901…
4	nyu_2451_34755.5	4	New Canaan	MULTILINESTRING ((768726.204 844093.227, 76815…

[16]:
```
#Let's add a new column called "operating" to the mnrail dataframe to␣
↪distinguish it from two other rail line dataframes.
mnrail['Operating'] = 'Metro North'
mnrail.head()
```

[16]:

	id	number	linename	geometry	Operating
0	nyu_2451_34755.1	1	Hudson	MULTILINESTRING ((645588.804 1046859.351, 6433…	Metro North
1	nyu_2451_34755.2	3	New Haven	MULTILINESTRING ((925725.242 904249.608, 92398…	Metro North
2	nyu_2451_34755.3	2	Harlem	MULTILINESTRING ((747752.029 1087426.164, 7489…	Metro North
3	nyu_2451_34755.4	5	Danbury	MULTILINESTRING ((780145.485 935346.610, 78901…	Metro North
4	nyu_2451_34755.5	4	New Canaan	MULTILINESTRING ((768726.204 844093.227, 76815…	Metro North

```
0  MULTILINESTRING ((645588.804 1046859.351, 6433…  Metro North
1  MULTILINESTRING ((925725.242 904249.608, 92398…  Metro North
2  MULTILINESTRING ((747752.029 1087426.164, 7489…  Metro North
3  MULTILINESTRING ((780145.485 935346.610, 78901…  Metro North
4  MULTILINESTRING ((768726.204 844093.227, 76815…  Metro North
```

Now the geometry columns of the three dataframes share the same coordination system, and we can merge them into one dataframe now. But before we do that, let's just look at the data types of the three dataframes.

[17]:
```
#check datatypes of columns
njrail.dtypes
```

[17]:
```
id               int64
linename        object
service         object
shape_leng     float64
date_stamp      object
geometry      geometry
Operating       object
dtype: object
```

[18]:
```
lirail.dtypes
```

[18]:
```
id             object
number          int64
linename       object
geometry     geometry
Operating      object
dtype: object
```

[19]:
```
mnrail.dtypes
```

[19]:
```
id             object
number          int64
linename       object
geometry     geometry
Operating      object
dtype: object
```

Let's combine the three rail line dataframe into a new one.

[20]:
```
linjrail = lirail.append(njrail)
nymarail=linjrail.append(mnrail)
```

[21]:
```
#Let's sample the new datagframe with the combined rail lines to make sure␣
 ↪there are values from all three dataframes.
nymarail.sample(10)
```

[21]:
```
                   id  number                 linename  \
14                 15     NaN      PASCACK VALLEY LINE
11                 12     NaN  NORTH JERSEY COAST LINE
6    nyu_2451_34753.7     5.0                  Montauk
2    nyu_2451_34753.3    12.0                City Zone
0    nyu_2451_34755.1     1.0                   Hudson
5    nyu_2451_34753.6     2.0                Hempstead
0    nyu_2451_34753.1    11.0                  Belmont
26                 27     NaN                     PATH
2    nyu_2451_34755.3     2.0                   Harlem
9   nyu_2451_34753.10     6.0               Long Beach

                                                 geometry             Operating  \
14  LINESTRING (622908.638 692949.426, 620720.513 …   New Jersey Railroad
11  LINESTRING (618407.609 452679.772, 618718.882 …   New Jersey Railroad
6   MULTILINESTRING ((632561.894 698796.549, 63274…  Long Island Railroad
2   MULTILINESTRING ((683616.747 680493.214, 68342…  Long Island Railroad
0   MULTILINESTRING ((645588.804 1046859.351, 6433…           Metro North
5   MULTILINESTRING ((632561.894 698796.549, 63274…  Long Island Railroad
0   MULTILINESTRING ((632561.894 698796.549, 63274…  Long Island Railroad
26  MULTILINESTRING ((621237.511 692081.826, 62123…   New Jersey Railroad
2   MULTILINESTRING ((747752.029 1087426.164, 7489…           Metro North
9   MULTILINESTRING ((683924.444 680574.535, 68361…  Long Island Railroad

                 service     shape_leng  date_stamp
14                  None  166462.999112  2013-11-04
11               HOBOKEN  338263.278633  2013-11-04
6                    NaN            NaN         NaN
2                    NaN            NaN         NaN
0                    NaN            NaN         NaN
5                    NaN            NaN         NaN
0                    NaN            NaN         NaN
26  HOBOKEN - 33 STREET   18974.559217  2016-02-29
2                    NaN            NaN         NaN
9                    NaN            NaN         NaN
```

[22]:
```
#Let's clean up the data a little more and get rid of the columns where there␣
↪are some rows without value.
columns_to_keep=['linename','geometry','Operating']
nymarail=nymarail[columns_to_keep]
nymarail.sample(10)
```

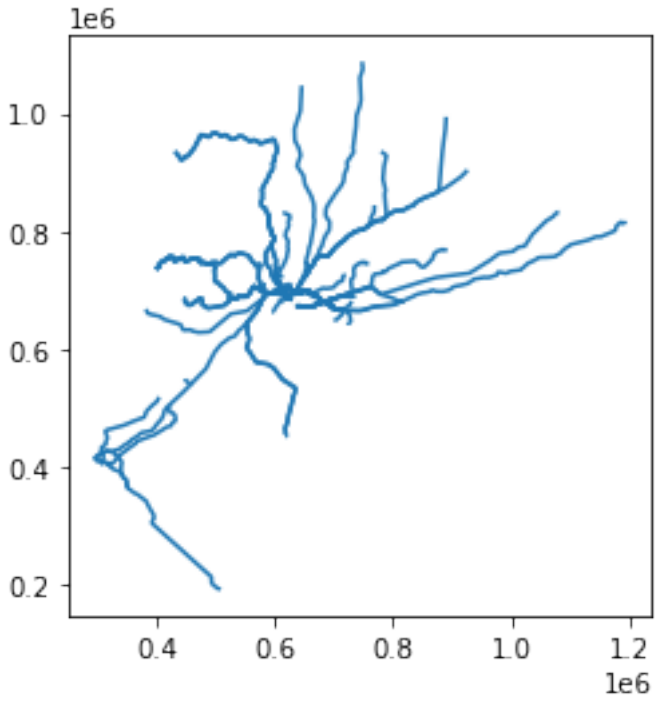
[22]:
```
                    linename  \
7              MORRIS & ESSEX
1          BERGEN COUNTY LINE
21  HUDSON BERGEN LIGHT RAIL
24                      PATH
```

```
13   NORTHEAST CORRIDOR LINE
4     MONTCLAIR BOONTON LINE
15           PATCO SPEEDLINE
5                  Hempstead
6             MORRIS & ESSEX
9                 Long Beach
```

	geometry	Operating
7	LINESTRING (399357.761 735253.132, 399513.899 …	New Jersey Railroad
1	LINESTRING (622908.638 692949.426, 620720.513 …	New Jersey Railroad
21	MULTILINESTRING ((614528.455 683433.565, 61438…	New Jersey Railroad
24	MULTILINESTRING ((622045.451 686297.289, 62205…	New Jersey Railroad
13	LINESTRING (458356.477 540953.196, 458004.667 …	New Jersey Railroad
4	LINESTRING (399357.761 735253.132, 399513.899 …	New Jersey Railroad
15	LINESTRING (351629.897 364774.694, 351391.896 …	New Jersey Railroad
5	MULTILINESTRING ((632561.894 698796.549, 63274…	Long Island Railroad
6	LINESTRING (446012.803 687337.456, 446094.132 …	New Jersey Railroad
9	MULTILINESTRING ((683924.444 680574.535, 68361…	Long Island Railroad

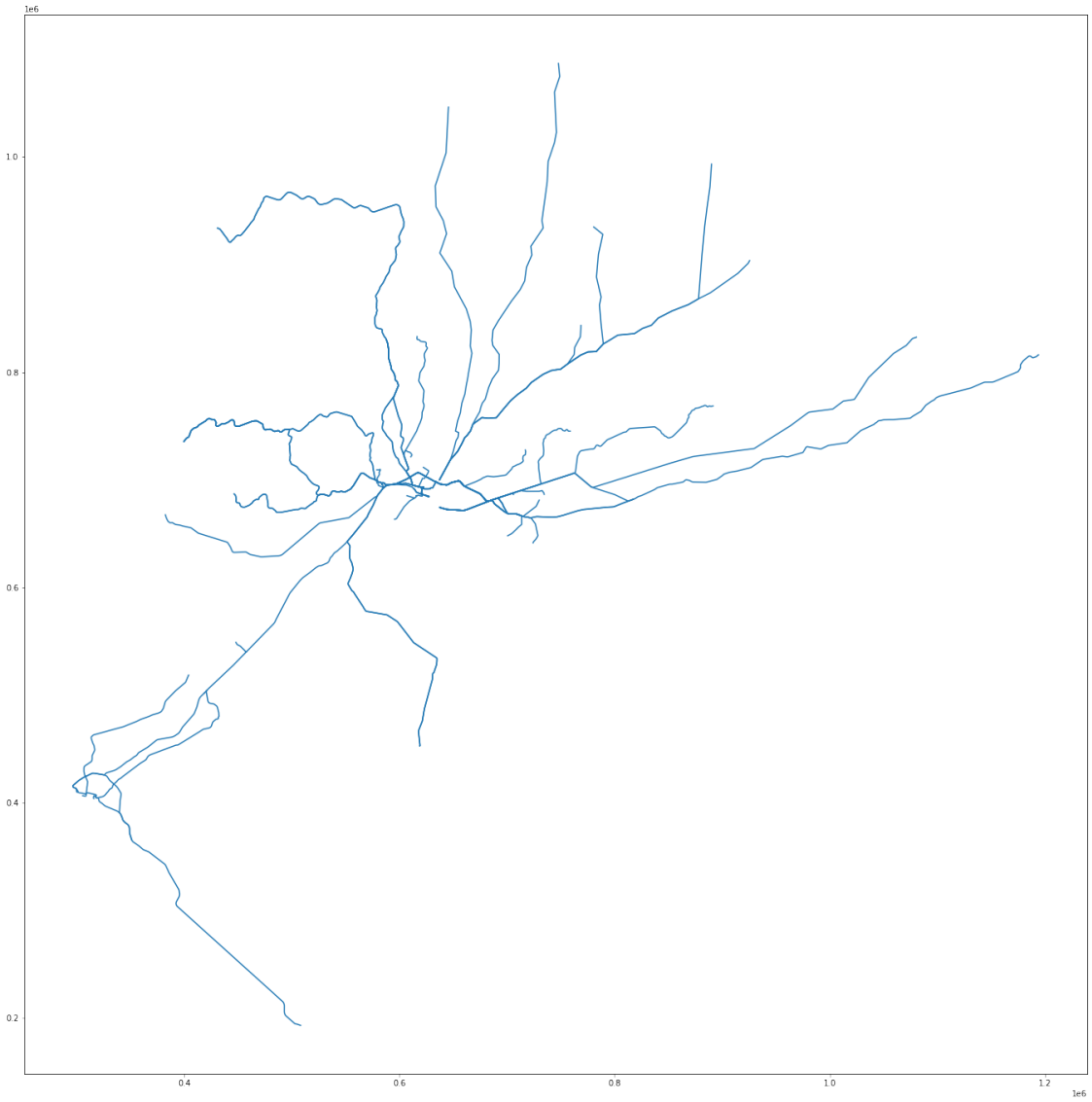
[23]:
```
#Let's plot the combined rail lines onto the map
nymarail.plot()
```

[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f78ed214e20>
```



[24]:
```
#Let's make it look better
nymarail.plot(figsize=(40,24))
```

[24]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f78ed214ee0>
```



5.2 Data exploration for transit stations

Let's do the same thing with the station datasets.

[25]:
```
#looking at first 5 rows of the njstation, listation and mnstation datasets.
njstation.head()
```

[25]:
```
   OBJECTID    COUNTY   LATITUDE  LONGITUDE         STATION  \
0         1     OCEAN  40.092718 -74.048192  Point Pleasant
1         2  MONMOUTH  40.150567 -74.035460     Spring Lake
2         3  MONMOUTH  40.180589 -74.027296          Belmar
3         4  MONMOUTH  40.203775 -74.018956   Bradley Beach
4         5  MONMOUTH  40.215360 -74.014788     Asbury Park

                 RAIL_LINE                      MUN_LABEL   ATIS_ID AMTRAK  \
0  North Jersey Coast Line  Point Pleasant Beach Borough  RAIL0122      N
1  North Jersey Coast Line            Spring Lake Borough  RAIL0141      N
2  North Jersey Coast Line                 Belmar Borough  RAIL0015      N
3  North Jersey Coast Line          Bradley Beach Borough  RAIL0022      N
4  North Jersey Coast Line               Asbury Park City  RAIL0008      N

                        geometry
0  POINT (618521.134 459008.903)
1  POINT (621972.996 480099.144)
2  POINT (624196.751 491047.221)
3  POINT (626480.961 499505.650)
4  POINT (627622.290 503731.988)
```

[26]:
```
listation.head()
```

[26]:
```
                 id  stop_id            stop_name  stop_lat  stop_lon  geoid  \
0  nyu_2451_34754.1        1     Long Island City  40.74128 -73.95639  36081
1  nyu_2451_34754.2        2  Hunterspoint Avenue  40.74238 -73.94679  36081
2  nyu_2451_34754.3        8         Penn Station  40.75058 -73.99358  36061
3  nyu_2451_34754.4        9             Woodside  40.74584 -73.90297  36081
4  nyu_2451_34754.5       10         Forest Hills  40.71957 -73.84481  36081

          namelsad                     geometry
0    Queens County  POINT (-73.95639 40.74128)
1    Queens County  POINT (-73.94679 40.74238)
2  New York County  POINT (-73.99358 40.75058)
3    Queens County  POINT (-73.90297 40.74584)
4    Queens County  POINT (-73.84481 40.71957)
```

There are 277 stations in the NJ Rail network and some has multiple values, meaning there are more than 1 lines intersecting at those staions.

[27]:
```
mnstation.head()
```

[27]:
```
                 id  stop_id           stop_name   stop_lat   stop_lon  \
0  nyu_2451_34756.1        1       Grand Central  40.752998 -73.977056
1  nyu_2451_34756.2        4    Harlem-125th St.  40.805157 -73.939149
2  nyu_2451_34756.3      622    Yankees-E153 St.  40.825300 -73.929900
3  nyu_2451_34756.4        9      Morris Heights  40.854252 -73.919583
```

```
4  nyu_2451_34756.5       10  University Heights  40.862248 -73.913120

   wheelchair  geoid         namelsad                     geometry
0           1  36061  New York County  POINT (-73.97706 40.75300)
1           1  36061  New York County  POINT (-73.93915 40.80516)
2           1  36005     Bronx County  POINT (-73.92990 40.82530)
3           1  36005     Bronx County  POINT (-73.91958 40.85425)
4           1  36005     Bronx County  POINT (-73.91312 40.86225)
```

Again, three dataframes share the names and the geometry columns, and let's change the names of those two columns. From the data exploration and analysis for the rail line dataframes, I know I don't need to keep all the columns as well, so I will drop the columns that are not shared across the dataframes.

[28]:
```
list(njstation)
```

[28]:
```
['OBJECTID',
 'COUNTY',
 'LATITUDE',
 'LONGITUDE',
 'STATION',
 'RAIL_LINE',
 'MUN_LABEL',
 'ATIS_ID',
 'AMTRAK',
 'geometry']
```

[29]:
```
columns_to_keep=['STATION','LATITUDE','LONGITUDE','geometry']
njstation=njstation[columns_to_keep]
njstation.head()
```

[29]:
```
          STATION   LATITUDE  LONGITUDE                       geometry
0  Point Pleasant  40.092718 -74.048192  POINT (618521.134 459008.903)
1     Spring Lake  40.150567 -74.035460  POINT (621972.996 480099.144)
2          Belmar  40.180589 -74.027296  POINT (624196.751 491047.221)
3   Bradley Beach  40.203775 -74.018956  POINT (626480.961 499505.650)
4     Asbury Park  40.215360 -74.014788  POINT (627622.290 503731.988)
```

[30]:
```
njstation.columns=['stationname','lat','lon','geometry']
njstation
```

[30]:
```
        stationname        lat        lon                       geometry
0    Point Pleasant  40.092718 -74.048192  POINT (618521.134 459008.903)
1       Spring Lake  40.150567 -74.035460  POINT (621972.996 480099.144)
2            Belmar  40.180589 -74.027296  POINT (624196.751 491047.221)
3     Bradley Beach  40.203775 -74.018956  POINT (626480.961 499505.650)
4       Asbury Park  40.215360 -74.014788  POINT (627622.290 503731.988)
```

```
..              …        …        …                                  …
280         Bristol  40.105037 -74.854642  POINT (392929.771 463373.005)
281         Croydon  40.093575 -74.906575  POINT (378384.438 459260.172)
282       Eddington  40.082994 -74.933703  POINT (370776.718 455441.562)
283          Tacony  40.023226 -75.039024  POINT (341176.327 433831.875)
284       Levittown  40.140259 -74.817016  POINT (403499.492 476163.107)

[285 rows x 4 columns]
```

[31]:
```
njstation['Operating'] = 'New Jersey Railroad'
njstation.head()
```

[31]:
```
      stationname        lat        lon                        geometry  \
0  Point Pleasant  40.092718 -74.048192  POINT (618521.134 459008.903)
1     Spring Lake  40.150567 -74.035460  POINT (621972.996 480099.144)
2          Belmar  40.180589 -74.027296  POINT (624196.751 491047.221)
3   Bradley Beach  40.203775 -74.018956  POINT (626480.961 499505.650)
4     Asbury Park  40.215360 -74.014788  POINT (627622.290 503731.988)

             Operating
0  New Jersey Railroad
1  New Jersey Railroad
2  New Jersey Railroad
3  New Jersey Railroad
4  New Jersey Railroad
```

[32]:
```
list(listation)
```

[32]:
```
['id',
 'stop_id',
 'stop_name',
 'stop_lat',
 'stop_lon',
 'geoid',
 'namelsad',
 'geometry']
```

[33]:
```
columns_to_keep=['stop_name','stop_lat','stop_lon','geometry']
listation=listation[columns_to_keep]
listation.head()
```

[33]:
```
             stop_name  stop_lat  stop_lon                    geometry
0    Long Island City  40.74128 -73.95639  POINT (-73.95639 40.74128)
1  Hunterspoint Avenue  40.74238 -73.94679  POINT (-73.94679 40.74238)
2        Penn Station  40.75058 -73.99358  POINT (-73.99358 40.75058)
3            Woodside  40.74584 -73.90297  POINT (-73.90297 40.74584)
4        Forest Hills  40.71957 -73.84481  POINT (-73.84481 40.71957)
```

[34]:
```
listation.columns=['stationname','lat','lon','geometry']
listation.head()
```

[34]:

	stationname	lat	lon	geometry
0	Long Island City	40.74128	-73.95639	POINT (-73.95639 40.74128)
1	Hunterspoint Avenue	40.74238	-73.94679	POINT (-73.94679 40.74238)
2	Penn Station	40.75058	-73.99358	POINT (-73.99358 40.75058)
3	Woodside	40.74584	-73.90297	POINT (-73.90297 40.74584)
4	Forest Hills	40.71957	-73.84481	POINT (-73.84481 40.71957)

[35]:
```
listation['Operating'] = 'Long Island Railroad'
listation.head()
```

[35]:

	stationname	lat	lon	geometry	Operating
0	Long Island City	40.74128	-73.95639	POINT (-73.95639 40.74128)	Long Island Railroad
1	Hunterspoint Avenue	40.74238	-73.94679	POINT (-73.94679 40.74238)	Long Island Railroad
2	Penn Station	40.75058	-73.99358	POINT (-73.99358 40.75058)	Long Island Railroad
3	Woodside	40.74584	-73.90297	POINT (-73.90297 40.74584)	Long Island Railroad
4	Forest Hills	40.71957	-73.84481	POINT (-73.84481 40.71957)	Long Island Railroad

[36]:
```
columns_to_keep=['stop_name','stop_lat','stop_lon','geometry']
mnstation=mnstation[columns_to_keep]
mnstation.head()
```

[36]:

	stop_name	stop_lat	stop_lon	geometry
0	Grand Central	40.752998	-73.977056	POINT (-73.97706 40.75300)
1	Harlem-125th St.	40.805157	-73.939149	POINT (-73.93915 40.80516)
2	Yankees-E153 St.	40.825300	-73.929900	POINT (-73.92990 40.82530)
3	Morris Heights	40.854252	-73.919583	POINT (-73.91958 40.85425)
4	University Heights	40.862248	-73.913120	POINT (-73.91312 40.86225)

[37]:
```
mnstation.columns=['stationname','lat','lon','geometry']
mnstation.head()
```

[37]:

	stationname	lat	lon	geometry
0	Grand Central	40.752998	-73.977056	POINT (-73.97706 40.75300)
1	Harlem-125th St.	40.805157	-73.939149	POINT (-73.93915 40.80516)
2	Yankees-E153 St.	40.825300	-73.929900	POINT (-73.92990 40.82530)
3	Morris Heights	40.854252	-73.919583	POINT (-73.91958 40.85425)
4	University Heights	40.862248	-73.913120	POINT (-73.91312 40.86225)

[38]:
```
mnstation['Operating'] = 'Metro North Railroad'
mnstation.head()
```

[38]:
```
           stationname        lat        lon                     geometry  \
0       Grand Central  40.752998 -73.977056  POINT (-73.97706 40.75300)
1    Harlem-125th St.  40.805157 -73.939149  POINT (-73.93915 40.80516)
2    Yankees-E153 St.  40.825300 -73.929900  POINT (-73.92990 40.82530)
3      Morris Heights  40.854252 -73.919583  POINT (-73.91958 40.85425)
4  University Heights  40.862248 -73.913120  POINT (-73.91312 40.86225)

              Operating
0  Metro North Railroad
1  Metro North Railroad
2  Metro North Railroad
3  Metro North Railroad
4  Metro North Railroad
```

Now we have changed the column names for station and geometry to the same across the three dataframes, but I noticed that the coordination system for geometry columns are different, I need to change them to the same coordination system for mapping purpose

[39]:
```
listation=listation.to_crs('epsg:3424')
listation.head()
```

[39]:
```
            stationname       lat       lon                         geometry  \
0     Long Island City  40.74128 -73.95639  POINT (642750.072 695409.903)
1  Hunterspoint Avenue  40.74238 -73.94679  POINT (645407.555 695827.249)
2         Penn Station  40.75058 -73.99358  POINT (632425.769 698736.255)
3             Woodside  40.74584 -73.90297  POINT (657540.623 697167.271)
4         Forest Hills  40.71957 -73.84481  POINT (673726.190 687712.061)

              Operating
0  Long Island Railroad
1  Long Island Railroad
2  Long Island Railroad
3  Long Island Railroad
4  Long Island Railroad
```

[40]:
```
mnstation=mnstation.to_crs('epsg:3424')
mnstation.head()
```

[40]:
```
           stationname        lat        lon                         geometry  \
0       Grand Central  40.752998 -73.977056  POINT (636998.419 699643.976)
1    Harlem-125th St.  40.805157 -73.939149  POINT (647378.517 718710.419)
2    Yankees-E153 St.  40.825300 -73.929900  POINT (649891.090 726065.105)
3      Morris Heights  40.854252 -73.919583  POINT (652676.295 736631.220)
4  University Heights  40.862248 -73.913120  POINT (654444.539 739556.129)
```

```
              Operating
0  Metro North Railroad
1  Metro North Railroad
2  Metro North Railroad
3  Metro North Railroad
4  Metro North Railroad
```

Time to combine the three dataframes.

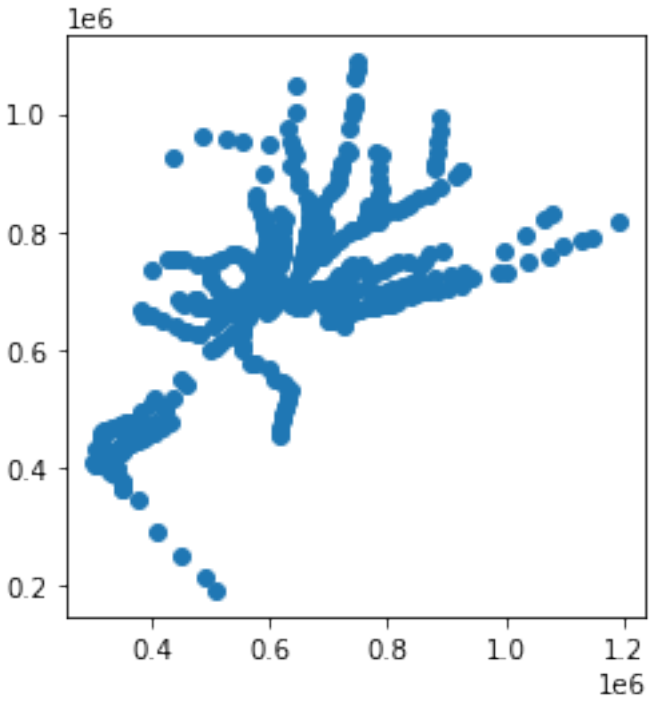
[41]:
```
linjstation= listation.append(njstation)
nymastation=linjstation.append(mnstation)
nymastation.sample(10)
```

[41]:
```
                 stationname        lat        lon  \
55                     Purdy's  41.325775 -73.659061
131  Essex Street-Hackensack  40.878996 -74.051883
157        Montclair St Univ  40.869784 -74.197434
53                   Millburn  40.725635 -74.303811
210               Orange St.  40.750623 -74.184941
54             Goldens Bridge  41.294338 -73.677655
216          Washington Park  40.743944 -74.169849
88                Bloomfield  40.792708 -74.200065
32                  Dunellen  40.590868 -74.463038
229                Riverside  40.039260 -74.958890

                          geometry             Operating
55   POINT (723083.636 909003.136)  Metro North Railroad
131  POINT (616033.964 745430.746)   New Jersey Railroad
157  POINT (575799.077 741902.150)   New Jersey Railroad
53   POINT (546498.360 689304.806)   New Jersey Railroad
210  POINT (579410.224 698503.984)   New Jersey Railroad
54   POINT (718085.391 897500.508)  Metro North Railroad
216  POINT (583600.447 696085.889)   New Jersey Railroad
88   POINT (575167.770 713820.782)   New Jersey Railroad
32   POINT (502389.527 640151.874)   New Jersey Railroad
229  POINT (363647.145 439546.730)   New Jersey Railroad
```

Time to plot the combined station dataframe onto the map

[42]:
```
nymastation.plot()
```

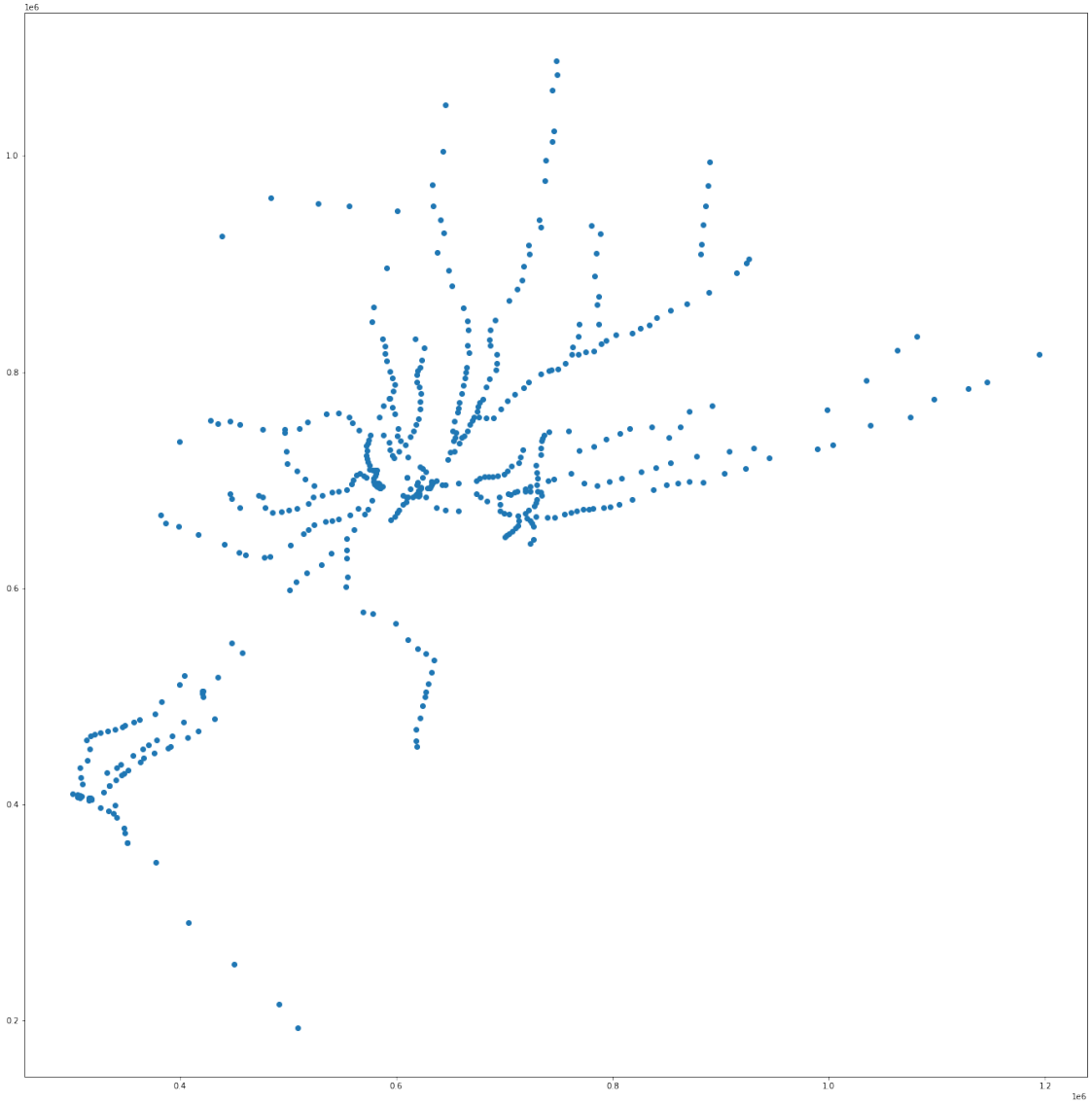
[42]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f78ed1a0910>
```



Let's make it look better:

[43]:
```
nymastation.plot(
            figsize=(40,24))
```

[43]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f78e9f90460>
```



6 Data analysis

Now I have the two new dataframes with combines lines and stations, I want to plot them onto the county boundary map.

[44]:
```
#import county boundary shapefile
cb=gpd.read_file('Countyborder')
```

[45]:
```
#quick look at the dataframe--I'm looking at geometry column especially to see␣
 ↪if there is a need to convert the coordination system
cb.head()
```

[45]:
```
  STATEFP COUNTYFP  COUNTYNS  GEOID       NAME          NAMELSAD LSAD CLASSFP  \
0      31       039  00835841  31039     Cuming     Cuming County   06      H1
1      53       069  01513275  53069  Wahkiakum  Wahkiakum County   06      H1
2      35       011  00933054  35011    De Baca    De Baca County   06      H1
3      31       109  00835876  31109  Lancaster  Lancaster County   06      H1
4      31       129  00835886  31129   Nuckolls   Nuckolls County   06      H1

   MTFCC CSAFP CBSAFP METDIVFP FUNCSTAT       ALAND    AWATER     INTPTLAT  \
0  G4020  None   None     None        A  1477641638  10701538  +41.9158651
1  G4020  None   None     None        A   680956787  61588406  +46.2946377
2  G4020  None   None     None        A  6016761648  29147345  +34.3592729
3  G4020   339  30700     None        A  2169252486  22867561  +40.7835474
4  G4020  None   None     None        A  1489645186   1718484  +40.1764918

       INTPTLON                                           geometry
0  -096.7885168  POLYGON ((-97.01952 42.00410, -97.01952 42.004…
1  -123.4244583  POLYGON ((-123.43639 46.23820, -123.44759 46.2…
2  -104.3686961  POLYGON ((-104.56739 33.99757, -104.56772 33.9…
3  -096.6886584  POLYGON ((-96.91060 40.95841, -96.91060 40.958…
4  -098.0468422  POLYGON ((-98.27367 40.08940, -98.27367 40.089…
```

[46]:
```
#convert the coordination system to match with the line and station dataframes
cb=cb.to_crs('epsg:3424')
cb.head()
```

[46]:
```
  STATEFP COUNTYFP  COUNTYNS  GEOID       NAME          NAMELSAD LSAD CLASSFP  \
0      31       039  00835841  31039     Cuming     Cuming County   06      H1
1      53       069  01513275  53069  Wahkiakum  Wahkiakum County   06      H1
2      35       011  00933054  35011    De Baca    De Baca County   06      H1
3      31       109  00835876  31109  Lancaster  Lancaster County   06      H1
4      31       129  00835886  31129   Nuckolls   Nuckolls County   06      H1

   MTFCC CSAFP CBSAFP METDIVFP FUNCSTAT       ALAND    AWATER     INTPTLAT  \
0  G4020  None   None     None        A  1477641638  10701538  +41.9158651
1  G4020  None   None     None        A   680956787  61588406  +46.2946377
2  G4020  None   None     None        A  6016761648  29147345  +34.3592729
3  G4020   339  30700     None        A  2169252486  22867561  +40.7835474
4  G4020  None   None     None        A  1489645186   1718484  +40.1764918

       INTPTLON                                           geometry
0  -096.7885168  POLYGON ((-5641359.566 1984269.368, -5641275.3…
1  -123.4244583  POLYGON ((-11631254.865 6939959.122, -11630633…
2  -104.3686961  POLYGON ((-8768757.056 -321918.300, -8768861.1…
3  -096.6886584  POLYGON ((-5716403.584 1592485.371, -5716390.7…
4  -098.0468422  POLYGON ((-6187603.199 1378867.881, -6187601.9…
```

[47]:
```
#check data types
cb.dtypes
```

[47]:
```
STATEFP       object
COUNTYFP      object
COUNTYNS      object
GEOID         object
NAME          object
NAMELSAD      object
LSAD          object
CLASSFP       object
MTFCC         object
CSAFP         object
CBSAFP        object
METDIVFP      object
FUNCSTAT      object
ALAND          int64
AWATER         int64
INTPTLAT      object
INTPTLON      object
geometry    geometry
dtype: object
```

[48]:
```
#trim couunty border data set according to FIPS code of the counties in NYMA
newcb = cb[cb.GEOID.isin(["34037",
 "36111",
 "36103",
 "34039",
 "36027",
 "36059",
 "34023",
 "36119",
 "09009",
 "34017",
 "42089",
 "36085",
 "36079",
 "34025",
 "34035",
 "34029",
 "09001",
 "09005",
 "34027",
 "34013",
 "36081",
 "34003",
 "36047",
```

```
 "36061",
 "34031",
 "36087",
 "34019",
 "42103",
 "36071",
 "36005",
 "34021"])]
newcb.head()
```

[48]:

	STATEFP	COUNTYFP	COUNTYNS	GEOID	NAME	NAMELSAD	LSAD	CLASSFP	\
111	34	037	00882236	34037	Sussex	Sussex County	06	H1	
211	36	111	00974153	36111	Ulster	Ulster County	06	H1	
444	36	103	00974149	36103	Suffolk	Suffolk County	06	H1	
476	34	039	00882235	34039	Union	Union County	06	H1	
544	36	027	00974112	36027	Dutchess	Dutchess County	06	H1	

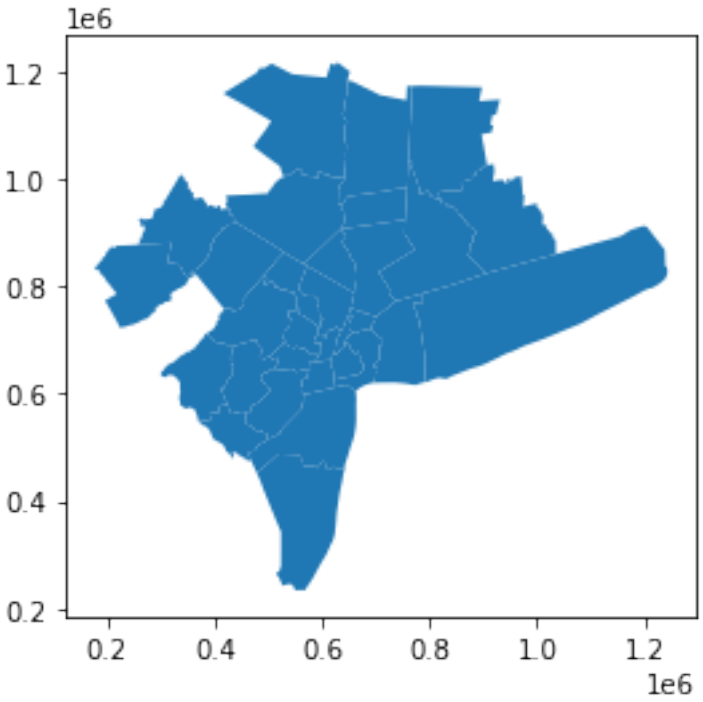
	MTFCC	CSAFP	CBSAFP	METDIVFP	FUNCSTAT	ALAND	AWATER	\
111	G4020	408	35620	35084	A	1343552956	43234734	
211	G4020	408	28740	None	A	2911757797	94596954	
444	G4020	408	35620	35004	A	2360846288	3785546967	
476	G4020	408	35620	35084	A	266170662	7046286	
544	G4020	408	35620	20524	A	2060678182	76956282	

	INTPTLAT	INTPTLON	\
111	+41.1374609	-074.6919141	
211	+41.9472124	-074.2654582	
444	+40.9435540	-072.6922184	
476	+40.6598707	-074.3086957	
544	+41.7547699	-073.7400411	

	geometry
111	POLYGON ((370993.193 814703.071, 370800.006 81…
211	POLYGON ((444806.717 1143268.270, 441642.444 1…
444	POLYGON ((942985.170 834736.238, 943040.019 83…
476	POLYGON ((511928.653 679591.091, 511951.923 67…
544	POLYGON ((644545.526 1107670.941, 644425.218 1…

[49]:
```
newcb.plot()
```

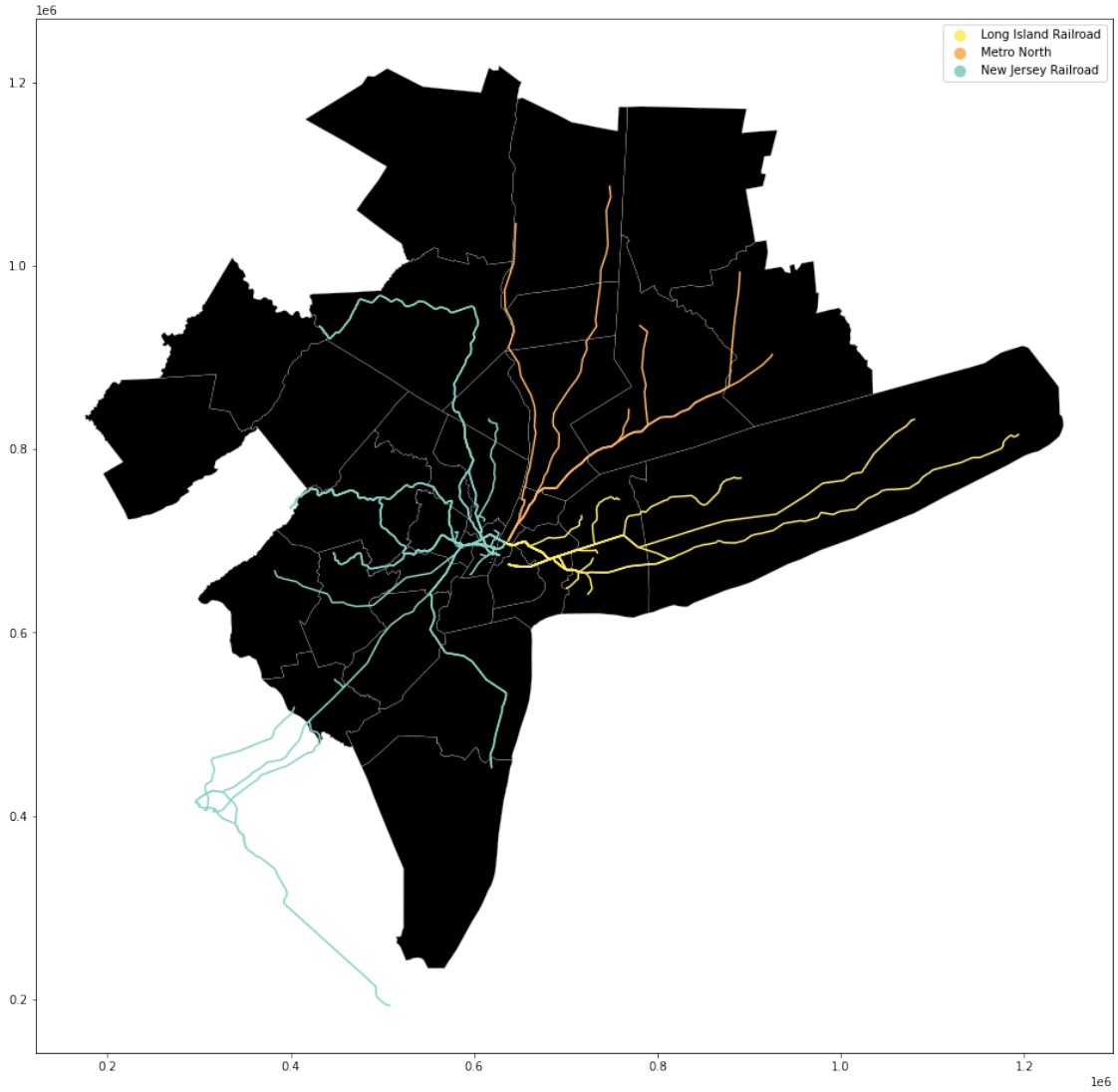
[49]: <matplotlib.axes._subplots.AxesSubplot at 0x7f78e9f3ff10>



Let's make it look better and put the line dataframe onto the county border map

[50]:
```
#plot transit lines onto the counties
fig, ax = plt.subplots(figsize = (20,16))
nymarail.plot(column = 'Operating',cmap = 'Set3_r', legend=True, ax=ax)
newcb.geometry.plot(color='black',edgecolor='white',linewidth = 0.2,ax=ax)
```

[50]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f78e9f16a90>
```



[51]:
```
#Let's add the stations on to the county boundary map
stationincounty = gpd.sjoin(nymastation, cb)
```

[52]:
```
stationincounty.head()
```

[52]:

	stationname	lat	lon	geometry	\
0	Long Island City	40.74128	-73.95639	POINT (642750.072 695409.903)	
1	Hunterspoint Avenue	40.74238	-73.94679	POINT (645407.555 695827.249)	
3	Woodside	40.74584	-73.90297	POINT (657540.623 697167.271)	
4	Forest Hills	40.71957	-73.84481	POINT (673726.190 687712.061)	
5	Kew Gardens	40.70964	-73.83089	POINT (677612.026 684123.665)	

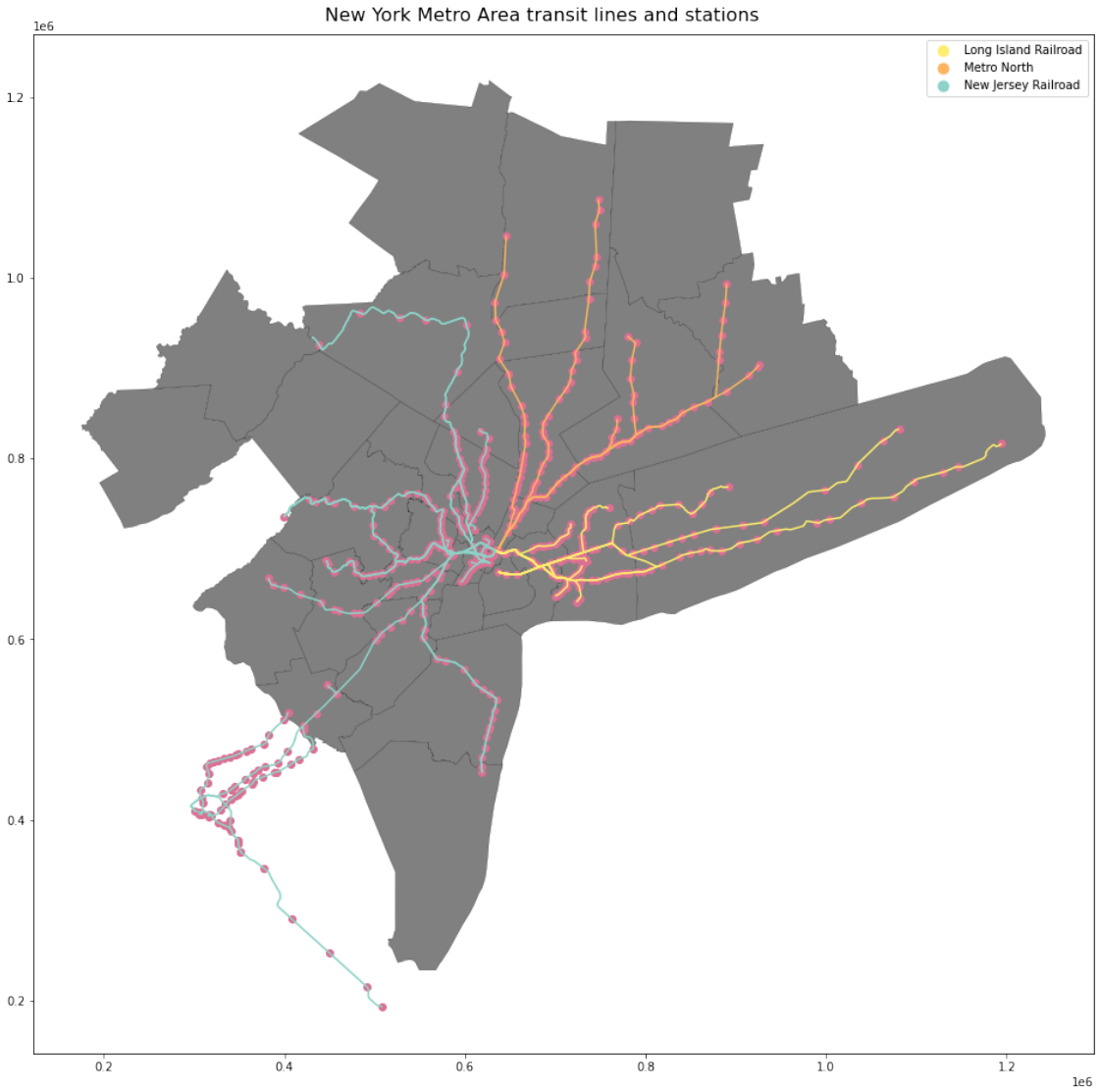
	Operating	index_right	STATEFP	COUNTYFP	COUNTYNS	GEOID	…	\

```
0  Long Island Railroad         2333      36      081  00974139  36081  …
1  Long Island Railroad         2333      36      081  00974139  36081  …
3  Long Island Railroad         2333      36      081  00974139  36081  …
4  Long Island Railroad         2333      36      081  00974139  36081  …
5  Long Island Railroad         2333      36      081  00974139  36081  …

  CLASSFP  MTFCC CSAFP CBSAFP METDIVFP FUNCSTAT      ALAND     AWATER  \
0      H6  G4020   408  35620    35614        C  281697156  179401845
1      H6  G4020   408  35620    35614        C  281697156  179401845
3      H6  G4020   408  35620    35614        C  281697156  179401845
4      H6  G4020   408  35620    35614        C  281697156  179401845
5      H6  G4020   408  35620    35614        C  281697156  179401845

      INTPTLAT      INTPTLON
0  +40.6585662  -073.8380168
1  +40.6585662  -073.8380168
3  +40.6585662  -073.8380168
4  +40.6585662  -073.8380168
5  +40.6585662  -073.8380168

[5 rows x 23 columns]
```

[53]:
```
fig, ax = plt.subplots(figsize = (20,16))

nymarail.plot(column = 'Operating',cmap = 'Set3_r', legend=True, ax=ax)
newcb.geometry.plot(color='Grey', edgecolor='black',linewidth = 0.2,ax=ax)
stationincounty.plot(column = 'name',color='palevioletred', legend=True, ax=ax)
fig.suptitle('New York Metro Area transit lines and stations', fontsize=16,x=0.
↪5,y=0.9)
```

```
/opt/conda/lib/python3.8/site-packages/geopandas/plotting.py:572: UserWarning:
Only specify one of 'column' or 'color'. Using 'color'.
  warnings.warn(
```

[53]:
```
Text(0.5, 0.9, 'New York Metro Area transit lines and stations')
```



6.1 Put stations on the map with Plotly.Express

[54]:
```
nymastation= px.scatter_mapbox(nymastation,
                          lat='lat',
                          lon='lon',
                          color='Operating',
                          hover_name="stationname",
                          mapbox_style="stamen-toner")
nymastation.show()
```

7 Transit density on county level

We want to conduct analysis on transit station denstiy on the county level. I will calculate the number of transit stations in each county

[55]:
```
#how many transit stations does each county have?
Queens=stationincounty[stationincounty['GEOID']=="36081"]
print ("There are" + " "+  str(len(Queens)) +" "+ "stations in Queens County,␣
 ↪NY")
```

```
There are 22 stations in Queens County, NY
```

[56]:
```
Fairfield=stationincounty[stationincounty['GEOID']=="09001"]
print ("There are" + " "+  str(len(Fairfield)) +" "+ "stations in Fairfield␣
 ↪County, CT")
```

```
There are 28 stations in Fairfield County, CT
```

[57]:
```
Litchfield=stationincounty[stationincounty['GEOID']=="09005"]
print ("There are" + " "+  str(len(Litchfield)) +" "+ "stations in Litchfield␣
 ↪County, CT")
```

```
There are 0 stations in Litchfield County, CT
```

[58]:
```
NewHaven=stationincounty[stationincounty['GEOID']=="09009"]
print ("There are" + " "+  str(len(NewHaven)) +" "+ "stations in NewHaven␣
 ↪County, CT")
```

```
There are 10 stations in NewHaven County, CT
```

[59]:
```
Bergen=stationincounty[stationincounty['GEOID']=="34003"]
print ("There are" + " "+  str(len(Bergen)) +" "+ "stations in Bergen County,␣
 ↪NJ")
```

```
There are 31 stations in Bergen County, NJ
```

[60]:
```
Essex=stationincounty[stationincounty['GEOID']=="34013"]
print ("There are" + " "+  str(len(Essex)) +" "+ "stations in Essex County, NJ")
```

```
There are 39 stations in Essex County, NJ
```

[61]:
```
Hudson=stationincounty[stationincounty['GEOID']=="34017"]
print ("There are" + " "+  str(len(Hudson)) +" "+ "stations in Hudson County,␣
 ↪NJ")
```

```
There are 33 stations in Hudson County, NJ
```

[62]:
```
Hunterdon=stationincounty[stationincounty['GEOID']=="34019"]
```

```
print ("There are" + " "+  str(len(Hunterdon)) +" "+ "stations in Hunterdon␣
 ↪County, NJ")
```

```
There are 4 stations in Hunterdon County, NJ
```

[63]:
```
Mercer=stationincounty[stationincounty['GEOID']=="34021"]
print ("There are" + " "+  str(len(Mercer)) +" "+ "stations in Mercer County,␣
 ↪NJ")
```

```
There are 8 stations in Mercer County, NJ
```

[64]:
```
Middlesex=stationincounty[stationincounty['GEOID']=="34023"]
print ("There are" + " "+  str(len(Middlesex)) +" "+ "stations in Middlesex␣
 ↪County, NJ")
```

```
There are 10 stations in Middlesex County, NJ
```

[65]:
```
Monmouth=stationincounty[stationincounty['GEOID']=="34025"]
print ("There are" + " "+  str(len(Monmouth)) +" "+ "stations in Monmouth␣
 ↪County, NJ")
```

```
There are 14 stations in Monmouth County, NJ
```

[66]:
```
Morris=stationincounty[stationincounty['GEOID']=="34027"]
print ("There are" + " "+  str(len(Morris)) +" "+ "stations in Morris County,␣
 ↪NJ")
```

```
There are 19 stations in Morris County, NJ
```

[67]:
```
Ocean=stationincounty[stationincounty['GEOID']=="34029"]
print ("There are" + " "+  str(len(Ocean)) +" "+ "stations in Ocean County, NJ")
```

```
There are 2 stations in Ocean County, NJ
```

[68]:
```
Passaic=stationincounty[stationincounty['GEOID']=="34031"]
print ("There are" + " "+  str(len(Passaic)) +" "+ "stations in Passaic County,␣
 ↪NJ")
```

```
There are 9 stations in Passaic County, NJ
```

[69]:
```
Somerset=stationincounty[stationincounty['GEOID']=="34035"]
print ("There are" + " "+  str(len(Somerset)) +" "+ "stations in Somerset␣
 ↪County, NJ")
```

```
There are 11 stations in Somerset County, NJ
```

[70]:
```
Sussex=stationincounty[stationincounty['GEOID']=="34037"]
```

```
print ("There are" + " "+  str(len(Sussex)) +" "+ "stations in Sussex County,␣
 ↪NJ")
```

```
There are 0 stations in Sussex County, NJ
```

[71]:
```
Union=stationincounty[stationincounty['GEOID']=="34039"]
print ("There are" + " "+  str(len(Union)) +" "+ "stations in Union County, NJ")
```

```
There are 16 stations in Union County, NJ
```

[72]:
```
Bronx=stationincounty[stationincounty['GEOID']=="36005"]
print ("There are" + " "+  str(len(Bronx)) +" "+ "stations in Bronx County, NY")
```

```
There are 12 stations in Bronx County, NY
```

[73]:
```
Dutchess=stationincounty[stationincounty['GEOID']=="36027"]
print ("There are" + " "+  str(len(Dutchess)) +" "+ "stations in Dutchess␣
 ↪County, NY")
```

```
There are 10 stations in Dutchess County, NY
```

[74]:
```
Kings=stationincounty[stationincounty['GEOID']=="36047"]
print ("There are" + " "+  str(len(Kings)) +" "+ "stations in Kings County, NY")
```

```
There are 3 stations in Kings County, NY
```

[75]:
```
Nassau=stationincounty[stationincounty['GEOID']=="36059"]
print ("There are" + " "+  str(len(Nassau)) +" "+ "stations in Nassau County,␣
 ↪NY")
```

```
There are 57 stations in Nassau County, NY
```

[76]:
```
NewYork=stationincounty[stationincounty['GEOID']=="36061"]
print ("There are" + " "+  str(len(NewYork)) +" "+ "stations in New" + " "␣
 ↪+"York County, NY")
```

```
There are 11 stations in New York County, NY
```

[77]:
```
Orange=stationincounty[stationincounty['GEOID']=="36071"]
print ("There are" + " "+  str(len(Orange)) +" "+ "stations inOrange County,␣
 ↪NY")
```

```
There are 7 stations inOrange County, NY
```

[78]:
```
Putnam=stationincounty[stationincounty['GEOID']=="36079"]
print ("There are" + " "+  str(len(Putnam)) +" "+ "stations in Putnam County,␣
 ↪NY")
```

```
There are 6 stations in Putnam County, NY
```

[79]:
```
Queens=stationincounty[stationincounty['GEOID']=="36081"]
print ("There are" + " "+  str(len(Queens)) +" "+ "stations in Queens County,␣
 ↪NY")
```

```
There are 22 stations in Queens County, NY
```

[80]:
```
Richmond=stationincounty[stationincounty['GEOID']=="36085"]
print ("There are" + " "+  str(len(Richmond)) +" "+ "stations in Richmond␣
 ↪County, NY")
```

```
There are 0 stations in Richmond County, NY
```

[81]:
```
Rockland=stationincounty[stationincounty['GEOID']=="36087"]
print ("There are" + " "+  str(len(Rockland)) +" "+ "stations in Rockland␣
 ↪County, NY")
```

```
There are 5 stations in Rockland County, NY
```

[82]:
```
Suffolk=stationincounty[stationincounty['GEOID']=="36103"]
print ("There are" + " "+  str(len(Suffolk)) +" "+ "stations in Suffolk County,␣
 ↪NY")
```

```
There are 41 stations in Suffolk County, NY
```

[83]:
```
Ulster=stationincounty[stationincounty['GEOID']=="36111"]
print ("There are" + " "+  str(len(Ulster)) +" "+ "stations in Ulster County,␣
 ↪NY")
```

```
There are 0 stations in Ulster County, NY
```

[84]:
```
Westchester=stationincounty[stationincounty['GEOID']=="36119"]
print ("There are" + " "+  str(len(Westchester)) +" "+ "stations in Westchester␣
 ↪County, NY")
```

```
There are 43 stations in Westchester County, NY
```

[85]:
```
Monroe=stationincounty[stationincounty['GEOID']=="42089"]
print ("There are" + " "+  str(len(Monroe)) +" "+ "stations in Monroe County,␣
 ↪PA")
```

```
There are 0 stations in Monroe County, PA
```

[86]:
```
Pike=stationincounty[stationincounty['GEOID']=="42103"]
print ("There are" + " "+  str(len(Pike)) +" "+ "stations in Pike County, PA")
```

```
There are 0 stations in Pike County, PA
```

[87]:
```
#create a new column of transit station number and add to the dataframe
newcb['stationcount'] = [
    str(len(stationincounty[stationincounty['GEOID']=="09001"])),
    str(len(stationincounty[stationincounty['GEOID']=="34025"])),
    str(len(stationincounty[stationincounty['GEOID']=="36081"])),
    str(len(stationincounty[stationincounty['GEOID']=="09005"])),
    str(len(stationincounty[stationincounty['GEOID']=="36027"])),
    str(len(stationincounty[stationincounty['GEOID']=="36087"])),
    str(len(stationincounty[stationincounty['GEOID']=="36059"])),
    str(len(stationincounty[stationincounty['GEOID']=="09009"])),
    str(len(stationincounty[stationincounty['GEOID']=="36079"])),
    str(len(stationincounty[stationincounty['GEOID']=="36103"])),
    str(len(stationincounty[stationincounty['GEOID']=="36071"])),
    str(len(stationincounty[stationincounty['GEOID']=="34039"])),
    str(len(stationincounty[stationincounty['GEOID']=="34029"])),
    str(len(stationincounty[stationincounty['GEOID']=="34013"])),
    str(len(stationincounty[stationincounty['GEOID']=="34027"])),
    str(len(stationincounty[stationincounty['GEOID']=="36005"])),
    str(len(stationincounty[stationincounty['GEOID']=="34023"])),
    str(len(stationincounty[stationincounty['GEOID']=="36047"])),
    str(len(stationincounty[stationincounty['GEOID']=="34003"])),
    str(len(stationincounty[stationincounty['GEOID']=="34017"])),
    str(len(stationincounty[stationincounty['GEOID']=="36085"])),
    str(len(stationincounty[stationincounty['GEOID']=="36119"])),
    str(len(stationincounty[stationincounty['GEOID']=="34037"])),
    str(len(stationincounty[stationincounty['GEOID']=="34035"])),
    str(len(stationincounty[stationincounty['GEOID']=="36061"])),
    str(len(stationincounty[stationincounty['GEOID']=="34021"])),
    str(len(stationincounty[stationincounty['GEOID']=="34019"])),
    str(len(stationincounty[stationincounty['GEOID']=="36111"])),
    str(len(stationincounty[stationincounty['GEOID']=="42089"])),
    str(len(stationincounty[stationincounty['GEOID']=="42103"])),
    str(len(stationincounty[stationincounty['GEOID']=="34031"]))
                    ]
```

```
/opt/conda/lib/python3.8/site-packages/geopandas/geodataframe.py:853:
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
```

[88]:
```
newcb.head()
```

[88]:
```
     STATEFP COUNTYFP  COUNTYNS  GEOID      NAME         NAMELSAD LSAD CLASSFP  \
111      34      037  00882236  34037    Sussex    Sussex County   06      H1
211      36      111  00974153  36111    Ulster    Ulster County   06      H1
444      36      103  00974149  36103   Suffolk   Suffolk County   06      H1
476      34      039  00882235  34039     Union     Union County   06      H1
544      36      027  00974112  36027  Dutchess  Dutchess County   06      H1

     MTFCC CSAFP CBSAFP METDIVFP FUNCSTAT       ALAND      AWATER  \
111  G4020   408  35620    35084        A  1343552956    43234734
211  G4020   408  28740     None        A  2911757797    94596954
444  G4020   408  35620    35004        A  2360846288  3785546967
476  G4020   408  35620    35084        A   266170662     7046286
544  G4020   408  35620    20524        A  2060678182    76956282

        INTPTLAT      INTPTLON  \
111  +41.1374609  -074.6919141
211  +41.9472124  -074.2654582
444  +40.9435540  -072.6922184
476  +40.6598707  -074.3086957
544  +41.7547699  -073.7400411

                                              geometry stationcount
111  POLYGON ((370993.193 814703.071, 370800.006 81…           28
211  POLYGON ((444806.717 1143268.270, 441642.444 1…           14
444  POLYGON ((942985.170 834736.238, 943040.019 83…           22
476  POLYGON ((511928.653 679591.091, 511951.923 67…            0
544  POLYGON ((644545.526 1107670.941, 644425.218 1…           10
```

[89]:
```
cb.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 3233 entries, 0 to 3232
Data columns (total 18 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   STATEFP   3233 non-null   object
 1   COUNTYFP  3233 non-null   object
 2   COUNTYNS  3233 non-null   object
 3   GEOID     3233 non-null   object
 4   NAME      3233 non-null   object
 5   NAMELSAD  3233 non-null   object
 6   LSAD      3233 non-null   object
 7   CLASSFP   3233 non-null   object
 8   MTFCC     3233 non-null   object
 9   CSAFP     1231 non-null   object
 10  CBSAFP    1899 non-null   object
 11  METDIVFP  113 non-null    object
```

```
 12  FUNCSTAT  3233 non-null   object
 13  ALAND     3233 non-null   int64
 14  AWATER    3233 non-null   int64
 15  INTPTLAT  3233 non-null   object
 16  INTPTLON  3233 non-null   object
 17  geometry  3233 non-null   geometry
dtypes: geometry(1), int64(2), object(15)
memory usage: 454.8+ KB
```

[90]:
```
#change data dypte for "TransitDensity" to calculate transit density
newcb['stationcount'] = newcb['stationcount'].astype(str).astype(int)
newcb.dtypes
```

```
/opt/conda/lib/python3.8/site-packages/geopandas/geodataframe.py:853:
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
```

[90]:
```
STATEFP           object
COUNTYFP          object
COUNTYNS          object
GEOID             object
NAME              object
NAMELSAD          object
LSAD              object
CLASSFP           object
MTFCC             object
CSAFP             object
CBSAFP            object
METDIVFP          object
FUNCSTAT          object
ALAND              int64
AWATER             int64
INTPTLAT          object
INTPTLON          object
geometry        geometry
stationcount       int64
dtype: object
```

[91]:
```
#calculate transit density of each county: stationcount/land area*10000000, and␣
↪add a new column "TD" for transit density
```

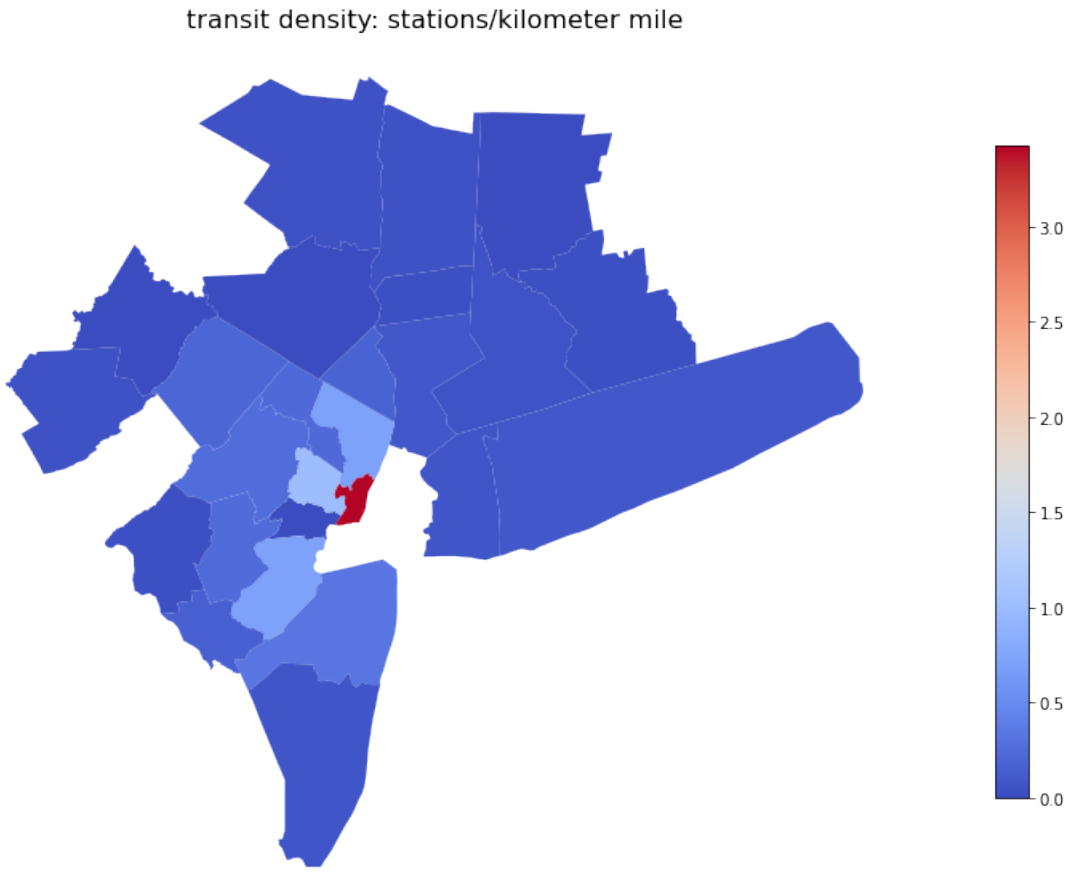
```
newcb['TD'] = newcb['stationcount']/newcb['ALAND']*10000000
```

[92]:
```
newcb.GEOID.unique()
```

[92]:
```
array(['34037', '36111', '36103', '34039', '36027', '36059', '34023',
       '36119', '09009', '34017', '42089', '36085', '36079', '34025',
       '34035', '34029', '09001', '09005', '34027', '34013', '36081',
       '34003', '36047', '36061', '34031', '36087', '34019', '42103',
       '36071', '36005', '34021'], dtype=object)
```

[107]:
```
#exclude 5 NYC counties from analysis and plotting since they are most likely␣
 ↪to be outliers
newcb=newcb.loc[newcb['GEOID'] != '36081']
newcb=newcb.loc[newcb['GEOID'] != '36047']
newcb=newcb.loc[newcb['GEOID'] != '36061']
newcb=newcb.loc[newcb['GEOID'] != '36005']
newcb=newcb.loc[newcb['GEOID'] != '36085']

#plot transit density "TD"
fig, ax = plt.subplots(figsize = (20,10))

newcb.plot(column = 'TD',
        cmap = 'coolwarm',
        legend=True,
        ax=ax,
        legend_kwds={'shrink': 0.75},
)
plt.axis("off")
plt.title("transit density: stations/kilometer mile", fontsize=16)
```

[107]:
```
Text(0.5, 1.0, 'transit density: stations/kilometer mile')
```



[97]:
```
#Get counties with top 5 and bottom 5 transit density
tdt5=newcb.sort_values(by='TD',ascending = False).head(5)
tdb5=newcb.sort_values(by='TD',ascending = False).tail(5)
#create new dataframe by combining the top 5 and bottom 5 counties for plotting
transitdensity=tdt5.append(tdb5)
```

[100]:
```
#plot transit density
figtd=px.bar(
      transitdensity,
      x='TD',
      y='NAME',
      orientation='h', #change orientation of bar chart
      color='TD',
      labels={'TD':'Transit Density','NAME':'County Names'},#change labels
      color_continuous_scale='Bluered' #change color of the bars
)
#update bar charts
figtd.update_layout(title={
```

```
        'text': "counties with top 5 and bottom 5 transit density", #add title
        'y':1,#change position of the title
        'x':0.5}
)
figtd
```

8 Conlusion

In this notebook, I combined three transit lines files and three transit stations files into two dataframs, and explore some mapping options for future analysis on transit system and migration, housing value and economic factors.