Notebook 7: Interactive Map for the Changes of All Metrics

One of the most important focus in our project is to compare the changes of each metric. This notebook is devoted to make an interactive map (with dropdown menu) that contains all changes on one single map. The notebook still starts with preparing geodata and census data. However, the challenging part this time is to use geojson file, which we converted from panda dataframe earlier. The actual map-making cell engages with for loop to form map traces for each cateogry.

In [1]:
```
# Import all modules I will be using in this note book.

import pandas as pd
import geopandas as gpd
import contextily as ctx
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import numpy as np
import fiona
```

```
/opt/conda/lib/python3.8/site-packages/geopandas/_compat.py:106: UserWarning: The Shapely GEOS version (3.8.1-CAPI-1.13.3) is i
ncompatible with the GEOS version PyGEOS was compiled with (3.9.0-CAPI-1.16.2). Conversions between both will be slow.
  warnings.warn(
```

In [ ]:

In [2]:
```
# Import the raw data that contains geo information. It is a SHP file.

countyborder = gpd.read_file('GeoData/02_Basemap_countyborder/cb_2018_us_county_500k.shp')
```

In [3]:
```
countyborder.head()
```

Out[3]:

	STATEFP	COUNTYFP	COUNTYNS	AFFGEOID	GEOID	NAME	LSAD	ALAND	AWATER	geometry
0	21	007	00516850	0500000US21007	21007	Ballard	06	639387454	69473325	POLYGON ((-89.18137 37.04630, -89.17938 37.053...
1	21	017	00516855	0500000US21017	21017	Bourbon	06	750439351	4829777	POLYGON ((-84.44266 38.28324, -84.44114 38.283...
2	21	031	00516862	0500000US21031	21031	Butler	06	1103571974	13943044	POLYGON ((-86.94486 37.07341, -86.94346 37.074...
3	21	065	00516879	0500000US21065	21065	Estill	06	655509930	6516335	POLYGON ((-84.12662 37.64540, -84.12483 37.646...
4	21	069	00516881	0500000US21069	21069	Fleming	06	902727151	7182793	POLYGON ((-83.98428 38.44549, -83.98246 38.450...

In [4]:
```
# Clean out the dataset by keeping the columns I need.

columns_to_keep1 = ['GEOID','geometry','STATEFP']
countyborder_trimmed1 = countyborder [columns_to_keep1]
```

In [5]:
```
countyborder_trimmed1.head()
```

Out[5]:

	GEOID	geometry	STATEFP
0	21007	POLYGON ((-89.18137 37.04630, -89.17938 37.053...	21
1	21017	POLYGON ((-84.44266 38.28324, -84.44114 38.283...	21
2	21031	POLYGON ((-86.94486 37.07341, -86.94346 37.074...	21
3	21065	POLYGON ((-84.12662 37.64540, -84.12483 37.646...	21
4	21069	POLYGON ((-83.98428 38.44549, -83.98246 38.450...	21

In [6]:
```
NYC_5County = ['36005','36047','36061','36081','36085']
NonNYC_Metro = ['09001','09005','09009','34003','34013','34017','34019','34021','34023',
                '34025','34027','34029','34031','34035','34037','34039','36027','36059',
                '36071','36079','36087','36103','36111','36119','42089','42103']
NonContiguous = ['72','02','15','66','69','78','60']
```

In [7]:
```
countyborder_trimmed1['Region'] = 'Non_Metro_the_contiguous_US'
```

```
/opt/conda/lib/python3.8/site-packages/geopandas/geodataframe.py:1322: 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-ve
rsus-a-copy
  super(GeoDataFrame, self).__setitem__(key, value)
```

In [8]:
```
def regionbyGEOID_NYC(name):
    countyborder_trimmed1.loc[countyborder_trimmed1['GEOID'] == name,'Region'] = 'NYC'

def regionbyGEOID_NonNYC_Metro(name):
    countyborder_trimmed1.loc[countyborder_trimmed1['GEOID'] == name,'Region'] = 'NonNYC_Metro'

def regionbyGEOID_NonContiguous(name):
    countyborder_trimmed1.loc[countyborder_trimmed1['STATEFP'] == name,'Region'] = 'Non_the_contiguous_US'
```

In [9]:
```
countyborder_trimmed1.head()
```

Out[9]:

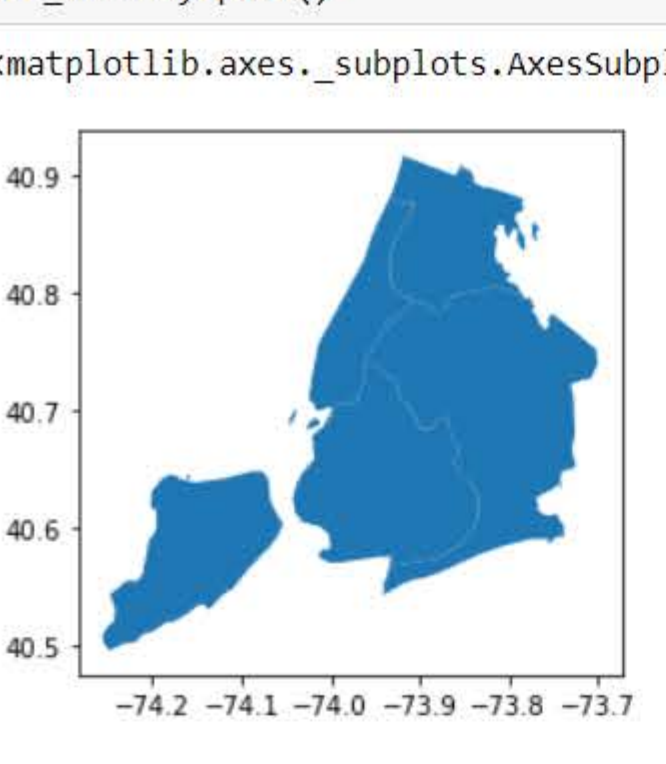
	GEOID	geometry	STATEFP	Region
0	21007	POLYGON ((-89.18137 37.04630, -89.17938 37.053...	21	Non_Metro_the_contiguous_US
1	21017	POLYGON ((-84.44266 38.28324, -84.44114 38.283...	21	Non_Metro_the_contiguous_US
2	21031	POLYGON ((-86.94486 37.07341, -86.94346 37.074...	21	Non_Metro_the_contiguous_US
3	21065	POLYGON ((-84.12662 37.64540, -84.12483 37.646...	21	Non_Metro_the_contiguous_US
4	21069	POLYGON ((-83.98428 38.44549, -83.98246 38.450...	21	Non_Metro_the_contiguous_US

In [10]:
```
for GEOID in NYC_5County:
    regionbyGEOID_NYC(GEOID)

NYC_5county = countyborder_trimmed1[countyborder_trimmed1.Region == 'NYC']
```

```
/opt/conda/lib/python3.8/site-packages/geopandas/geodataframe.py:1322: 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-ve
rsus-a-copy
  super(GeoDataFrame, self).__setitem__(key, value)
```

In [11]:
```
NYC_5county.plot()
```

Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7f746ad790d0>

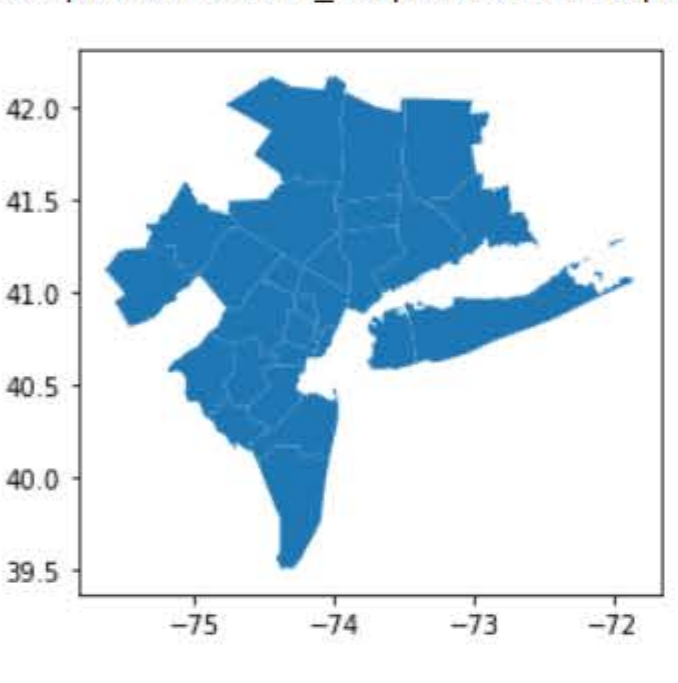


In [12]:
```
for GEOID in NonNYC_Metro:
    regionbyGEOID_NonNYC_Metro(GEOID)

NonNYC_Metro = countyborder_trimmed1[countyborder_trimmed1.Region == 'NonNYC_Metro']
```

```
/opt/conda/lib/python3.8/site-packages/geopandas/geodataframe.py:1322: 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-ve
rsus-a-copy
  super(GeoDataFrame, self).__setitem__(key, value)
```

In [13]:
```
NonNYC_Metro.plot()
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7f746ae16220>



In [14]:
```
for GEOID in NonContiguous:
    regionbyGEOID_NonContiguous(GEOID)

NonCountiguous = countyborder_trimmed1[countyborder_trimmed1.Region == 'Non_the_contiguous_US']
```

```
/opt/conda/lib/python3.8/site-packages/geopandas/geodataframe.py:1322: 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-ve
rsus-a-copy
  super(GeoDataFrame, self).__setitem__(key, value)
```

In [15]:
```
NonCountiguous.head()
```

Out[15]:

	GEOID	geometry	STATEFP	Region
26	02016	MULTIPOLYGON (((179.48246 51.98283, 179.48656 ...	02	Non_the_contiguous_US
27	02130	MULTIPOLYGON (((-130.98311 55.36598, -130.9809...	02	Non_the_contiguous_US
28	02180	MULTIPOLYGON (((-161.31946 64.12363, -161.3183...	02	Non_the_contiguous_US
29	02282	MULTIPOLYGON (((-139.51201 59.70289, -139.5095...	02	Non_the_contiguous_US
86	15007	MULTIPOLYGON (((-159.78794 22.03010, -159.7864...	15	Non_the_contiguous_US

In [ ]:

In [ ]:

In [ ]:

In [16]:
```
County_Demographics_Raw = pd.read_csv('Data/Combined_Census.csv',
                                      dtype={'AFFGEOID':str})
```

In [17]:
```
County_Demographics_Raw.head()
```

Out[17]:

	AFFGEOID	Geographic Area Name	Out-Migration	Transit Density	Railway Transportation	Total Population	Work From Home	GDP	Number of Jobs	Income Level	Housing affordability % change for owner-occupied units	Housing affordability % change for rental units	Median Number of Owned-Units	Median Rent
0	0500000US34017	Hudson County, New Jersey	2849	0.6851	0.0330	13753	0.0602	0.1475	0.0441	0.6609	8.0	0.0	-0.0077	0.1178
1	0500000US34029	Ocean County, New Jersey	-371	0.0031	-0.0491	10526	0.0507	0.1504	0.0385	0.0795	7.0	5.0	-0.0089	0.0710
2	0500000US09001	Fairfield County, Connecticut	801	0.0432	0.3556	10133	0.3090	0.1395	0.0571	0.4526	6.0	-1.0	-0.0045	0.0692
3	0500000US34003	Bergen County, New Jersey	1359	0.1284	0.1696	9543	0.1746	0.1450	0.0584	0.6739	8.0	3.0	0.0135	0.1148
4	0500000US34039	Union County, New Jersey	1201	0.1541	-0.0957	7830	0.0495	0.2400	0.0723	1.0136	8.0	4.0	0.0674	0.1094

In [18]:
```
County_Demographics_Raw ['GEOID'] = County_Demographics_Raw['AFFGEOID'].str.strip().str[-5:]
```

In [19]:
```
County_Demographics_Raw.head()
```

Out[19]:

	AFFGEOID	Geographic Area Name	Out-Migration	Transit Density	Railway Transportation	Total Population	Work From Home	GDP	Number of Jobs	Income Level	Housing affordability % change for owner-occupied units	Housing affordability % change for rental units	Median Number of Owned-Units	Median Rent
0	0500000US34017	Hudson County, New Jersey	2849	0.6851	0.0330	13753	0.0602	0.1475	0.0441	0.6609	8.0	0.0	-0.0077	0.1178
1	0500000US34029	Ocean County, New Jersey	-371	0.0031	-0.0491	10526	0.0507	0.1504	0.0385	0.0795	7.0	5.0	-0.0089	0.0710
2	0500000US09001	Fairfield County, Connecticut	801	0.0432	0.3556	10133	0.3090	0.1395	0.0571	0.4526	6.0	-1.0	-0.0045	0.0692
3	0500000US34003	Bergen County, New Jersey	1359	0.1284	0.1696	9543	0.1746	0.1450	0.0584	0.6739	8.0	3.0	0.0135	0.1148
4	0500000US34039	Union County, New Jersey	1201	0.1541	-0.0957	7830	0.0495	0.2400	0.0723	1.0136	8.0	4.0	0.0674	0.1094

In [20]:
```
County_Demographics_Merge = NonNYC_Metro.merge(County_Demographics_Raw,
                                               on='GEOID',
                                               how='left')
```

In [21]:
```
County_Demographics_Merge
```

Out[21]:

	GEOID	geometry	STATEFP	Region	AFFGEOID	Geographic Area Name	Out-Migration	Transit Density	Railway Transportation	Total Population	Work From Home	GDP	Numb of Jo
0	09009	MULTIPOLYGON (((-72.76143 41.24233, -72.75973 ...	09	NonNYC_Metro	0500000US09009	New Haven County, Connecticut	506	0.0161	0.0893	-3809	0.1997	0.1238	0.06
1	34003	POLYGON ((-74.27066 41.02103, -74.25046 41.060...	34	NonNYC_Metro	0500000US34003	Bergen County, New Jersey	1359	0.1284	0.1696	9543	0.1746	0.1450	0.05
2	34013	POLYGON ((-74.37623 40.76275, -74.37389 40.762	34	NonNYC_Metro	0500000US34013	Essex County, New Jersey	2411	0.3058	0.1165	3939	0.1739	0.1722	0.04

In [22]:
```
list(County_Demographics_Merge)
```

Out[22]:
```
['GEOID',
 'geometry',
 'STATEFP',
 'Region',
 'AFFGEOID',
 'Geographic Area Name',
 'Out-Migration',
 'Transit Density',
 'Railway Transportation',
 'Total Population',
 'Work From Home',
 'GDP',
 'Number of Jobs',
 'Income Level',
 'Housing affordability % change for owner-occupied units',
 'Housing affordability % change for rental units',
 'Median Number of Owned-Units',
 'Median Rent']
```

In [23]:
```
County_Demographics_Analysis1 = County_Demographics_Merge
```

Plotly Map

Import Json File

In [24]:
```
import json

with open('County_Demographics_Analysis2.geojson') as f:
    d = json.load(f)
    print(d)
```


{'type': 'FeatureCollection', 'crs': {'type': 'name', 'properties': {'name': 'urn:ogc:def:crs:EPSG::4269'}}, 'features': [{'ty
pe': 'Feature', 'properties': {'GEOID': '09009', 'STATEFP': '09', 'Region': 'NonNYC_Metro', 'AFFGEOID': '0500000US09009', 'Geo
graphic Area Name': 'New Haven County, Connecticut', 'Out-Migration': 506, 'Transit Density': 0.0161, 'Railway Transportatio
n': 0.0893, 'Total Population': -3809, 'Work From Home': 0.1997, 'GDP': 0.1238, 'Number of Jobs': 0.0684, 'Income Level': 0.38
06, 'Housing affordability % change for owner-occupied units': 6.0, 'Housing affordability % change for rental units': 5.0, 'M
edian Number of Owned-Units': -0.0138, 'Median Rent': 0.1056}, 'geometry': {'type': 'MultiPolygon', 'coordinates': [[[[-72.761
427, 41.242333], [-72.759733, 41.248454], [-72.75886, 41.253843], [-72.756353, 41.25548], [-72.754658, 41.255929], [-72.74767
8, 41.255855], [-72.740299, 41.256454], [-72.740099, 41.255105], [-72.741058, 41.252691], [-72.742891, 41.252631], [-72.74548
4, 41.250982], [-72.745661, 41.249723], [-72.749871, 41.247308], [-72.755655, 41.245059], [-72.757135, 41.244305], [-72.75966,
41.242012], [-72.760341, 41.241235], [-72.761427, 41.242333]]], [[[-72.764251, 41.259212], [-72.763066, 41.260377], [-72.76234
4, 41.260356], [-72.761187, 41.258996], [-72.761419, 41.258643], [-72.763115, 41.258037], [-72.763977, 41.258511], [-72.76425
1, 41.259212]]], [[[-73.326842, 41.485967], [-73.324874, 41.489567], [-73.326842, 41.492703], [-73.325508, 41.495306], [-73.32
3114, 41.497903], [-73.321578, 41.500415], [-73.320874, 41.503615], [-73.318889, 41.505189], [-73.307897, 41.505999], [-73.254
471, 41.511999], [-73.252265, 41.512134], [-73.249559, 41.512498], [-73.242247, 41.512952], [-73.219724, 41.510225], [-73.2080
06, 41.509007], [-73.20684, 41.508803], [-73.204197, 41.508527], [-73.170371, 41.512587], [-73.155492, 41.514304], [-73.15573,
41.515279], [-73.162523, 41.547083], [-73.164676, 41.557088], [-73.145036, 41.557822], [-73.130878, 41.560917], [-73.130221, 4
1.561055], [-73.129082, 41.561293], [-73.098529, 41.568976], [-73.093853, 41.570196], [-73.092353, 41.570496], [-73.089406, 4
1.575211], [-73.086755, 41.579592], [-73.085145, 41.582108], [-73.082955, 41.585616], [-73.080346, 41.586732], [-73.077825, 4
1.58748], [-73.064209, 41.588491], [-73.05801, 41.588647], [-73.057424, 41.589324], [-73.059458, 41.591605], [-73.060386, 41.5

In [25]:
```
d['type']
```

Out[25]: 'FeatureCollection'

In [26]:
```
d['features'][0].keys()
```

Out[26]: dict_keys(['type', 'properties', 'geometry'])

In [27]:
```
d['features'][0]['properties']
```

Out[27]:
```
{'GEOID': '09009',
 'STATEFP': '09',
 'Region': 'NonNYC_Metro',
 'AFFGEOID': '0500000US09009',
 'Geographic Area Name': 'New Haven County, Connecticut',
 'Out-Migration': 506,
 'Transit Density': 0.0161,
 'Railway Transportation': 0.0893,
 'Total Population': -3809,
 'Work From Home': 0.1997,
 'GDP': 0.1238,
 'Number of Jobs': 0.0684,
 'Income Level': 0.3806,
 'Housing affordability % change for owner-occupied units': 6.0,
 'Housing affordability % change for rental units': 5.0,
 'Median Number of Owned-Units': -0.0138,
 'Median Rent': 0.1056}
```

In [28]:
```
L = len(d['features'])
```

In [29]:
```
for k in range(L):
    d['features'][k]['id'] = f'{k}'
```

In [30]:
```
GEOID = []
for feat in d['features']:
    GEOID.append(feat['properties']['GEOID'])
GEOID
```

Out[30]:
```
['09009',
 '34003',
 '34013',
 '34023',
 '34019',
 '34021',
 '34025',
 '34029',
 '34035',
 '36103',
 '36119',
 '42103',
 '09001',
 '42089',
 '36059',
 '34027',
 '34031',
 '36071',
 '09005',
 '36079',
 '34037',
 '36027',
 '34039',
 '34017',
 '36111',
 '36087']
```

In [31]:
```
k=5
d['features'][k]['id']
```

Out[31]: '5'

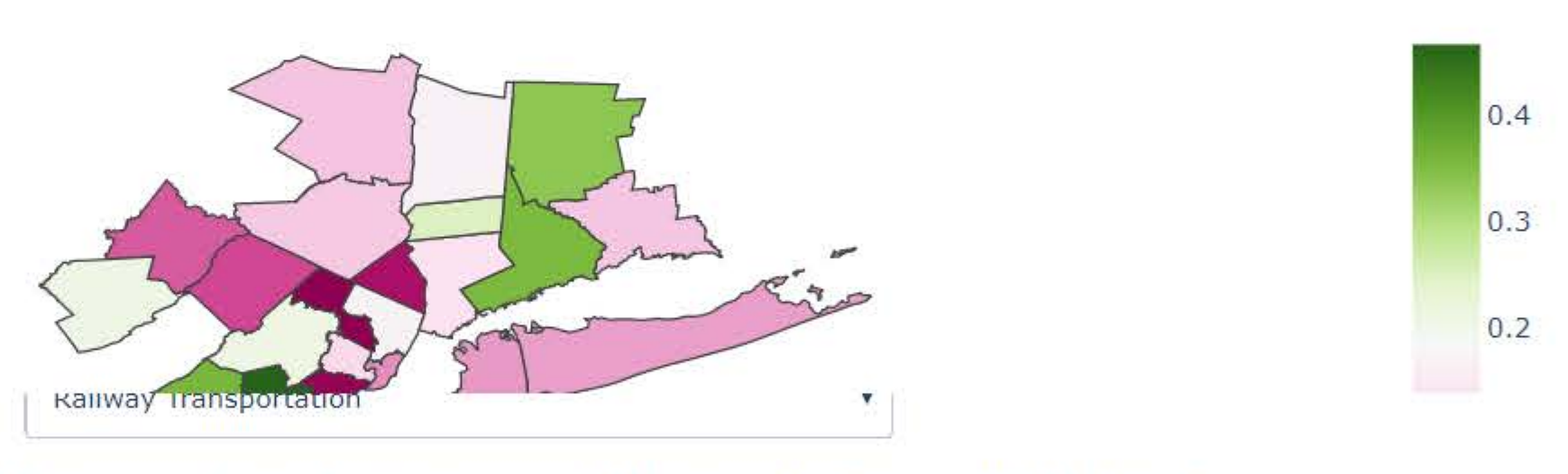
Produce Map

In [32]:
```
Topics = [
            'Railway Transportation',
            'Total Population',
            'Work From Home',
            'GDP',
            'Number of Jobs',
            'Income Level',
            'Housing affordability % change for owner-occupied units',
            'Housing affordability % change for rental units',
            'Median Number of Owned-Units',
            'Median Rent']
```

In [33]:
```
# define traces and buttons at once
traces = []
buttons = []

visible = np.array(Topics)

for topic in Topics:

    traces.append(go.Choropleth(geojson = d,
                                locations=[d['features'][k]['id'] for k in range(L)],
                                locationmode = 'geojson-id',
                                z = County_Demographics_Analysis1[topic].astype(float),
                                visible= True if topic == Topics[0] else False,
                                showlegend=False,
                                showscale = True
                                )
                  )


    buttons.append(dict(label=topic,
                        method="update",
                        args=[{"visible":list(visible==topic)},
                              {"title":f"<b>{topic}</b>"}]))

updatemenus = [{"active":0,
                "buttons":buttons,
                "x":0.5,
                "y":1.5
               }]

# Show figure


fig = go.Figure(data = traces,
                layout=dict(updatemenus=updatemenus))

# This is in order to get the first title displayed correctly
first_title = Topics[0]

fig.update_layout(title=f"<b>{first_title}</b>",title_x=0.5)


fig.update_geos(fitbounds="locations")

fig.update_layout(
    updatemenus=[
        dict(type = "dropdown",
            direction="down",
             xanchor="center",
             yanchor="top"
            )])

fig.update_layout(

    autosize=True,
)

fig.update_traces(marker_coloraxis='coloraxis', selector=dict(type='scatter'))
fig.update_traces(marker_colorbar_xanchor='left', selector=dict(type='scatter'))

fig.update_layout(title_xanchor='center')

fig.update_geos(visible=False)
fig.update_layout(title_text="The Change of Demographics, Economic, and Housing Metrics in NY Metro Area")
fig.show()
```

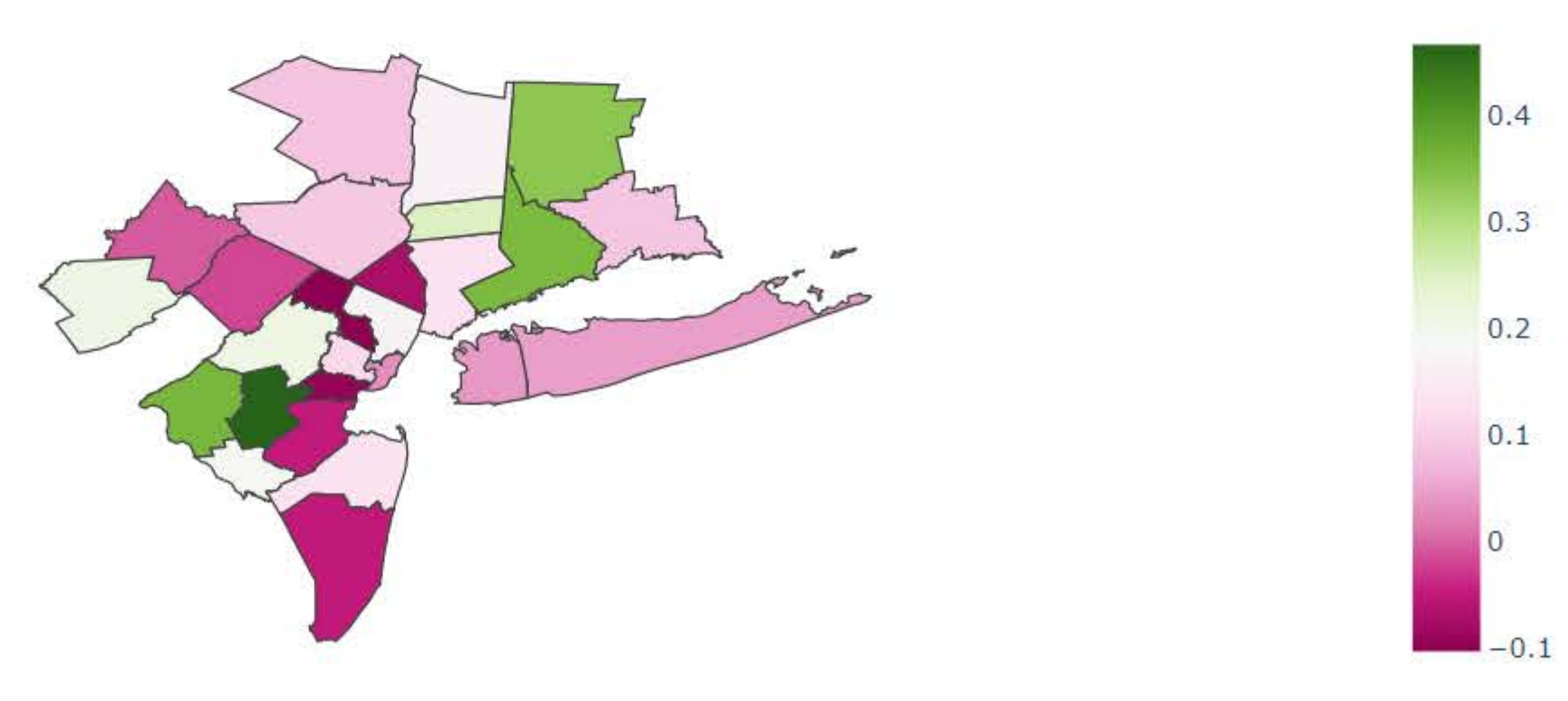
Railway Transportation

The Change of Demographics, Economic, and Housing Metrics in NY Metro Area



Railway Transportation

The Change of Demographics, Economic, and Housing Metrics in NY Metro Area



In [34]:
```
# fig.write_html("ExportData/AllFactorComparisonMap.html")
```

The resulted interactive map looks okay. We wish we could add more esthetic stuff into it, but figuring out the way to play around the button/dropdown menu is hard and is something this notebook focus on exploring.

In [ ]: