Notebook 4

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

Research Questions

- **Q1: On the county level, did housing costs change between 2014 and 2018? If so, what county experience increase in housing price? And what county experience decrease in housing price?**
  - **Expected Exploration:**
    - We expect to explore and analyze the housing value data in New York Metro Region on the county level in 2014 and 2018. We expect to compare the housing value between the two years on the county level to find out changes in housing value. We hope to capture the geographic shift of housing value if there was any change.
    - We expect to create maps and charts to visually represent our findings of housing value comparison in 2014 and 2018.
- **Q2: On the county level, did housing affordability change bewteen 2014 and 2018? If so, what county experience increase in housing afforability? And what county experience decrease in housing affordability?**
  - **Expected Exploration:**
    - We expect to capture changes of housing affordability in the New York Metro Area on the county level between 2014 and 2018 by exploring and analyzing the housing affordability data.
    - We expect to create maps and charts to visually represent our findings of housing affordability comparison in 2014 and 2018.
- **Purpose of this notebook:** I conducted data exploration and analysis of the New York Metro Area housing affordability and value data in 2014 and 2018. I used the ACS 5-year data from 2010 to 2014, and from 2014 to 2018. I created bar graphs with plotly.express to compare housing values and affordability in counties across NYMA between 2014 and 2018.

Data sources

1. 2010-2014 5-Years ACS: https://www.census.gov/acs/www/data/data-tables-and-tools/data-profiles/2014/ (https://www.census.gov/acs/www/data/data-tables-and-tools/data-profiles/2014/)

2. 2014-2018 5-Years ACS: https://www.census.gov/acs/www/data/data-tables-and-tools/data-profiles/2018/ (https://www.census.gov/acs/www/data/data-tables-and-tools/data-profiles/2018/)

Importing libraries

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

Data Exploration of Housing Affordability in 2014 and 2018

Data exploration of 2014 housing affordability data

In [2]:
```
#use .read_csv command to import 2014 ACS survey housing affordability data
hao2014 = pd.read_csv('hao2014.csv')
```

In [3]:
```
#look at the frist 5 rows of the dataset
hao2014.head()
```

...

In [4]:
```
#drop the first row (row 0)
hao2014 = hao2014.drop([0])
```

In [5]:
```
#check datatypes and columns
hao2014.dtypes
```

...

In [6]:
```
#convert FIPS from object to integer so that I can merge the 2014 housing affordability dataframe with coun
hao2014['FIPS'] = hao2014['FIPS'].astype(str).astype(int)
hao2014.dtypes
```

...

Data exploration of 2018 housing afforability data

In [7]:
```
#import 2018 ACS survey housing affordability data
hao2018 = pd.read_csv('hao2018.csv')
```

In [8]:
```
#looking at the first 5 rows of the dataframe
hao2018.head()
```

...

In [9]:
```
#drop the first row (row 0)
hao2018 = hao2018.drop([0])
```

In [10]:
```
#check datatypes
hao2018.dtypes
```

...

In [11]:
```
#convert FIPS from object to integer so that I can merge the 2018 housing affordability dataframe with coun
hao2018['FIPS'] = hao2018['FIPS'].astype(str).astype(int)
hao2018.dtypes
```

...

In [12]:
```
list(hao2018)
```

...

In [13]:

```
hao2018.columns=['GEO_ID',
 'NAME',
 'FIPS',
 'DP04_0080E',
 'DP04_0081E',
 'DP04_0081PE',
 'DP04_0082E',
 'DP04_0082PE',
 'DP04_0083E',
 'DP04_0083PE',
 'DP04_0084E',
 'DP04_0084PE',
 'DP04_0085E',
 'DP04_0085PE',
 'DP04_0086E',
 'DP04_0086PE',
 'DP04_0087E',
 'DP04_0087PE',
 'DP04_0088E',
 'DP04_0088PE',
 'DP04_0089E',
 'DP04_0090PE',
 'DP04_0091E',
 'DP04_0091PE',
 'DP04_0092E',
 'DP04_0092PE',
 'DP04_0093E',
 'DP04_0093PE',
 'DP04_0094E',
 'DP04_0094PE',
 'DP04_0098E',
 'DP04_0098PE',
 'DP04_0099E',
 'DP04_0099PE',
 'DP04_0100E',
 'DP04_0100PE',
 'DP04_0098E.1',
 'DP04_0098PE.1',
 'DP04_0099E.1',
 'DP04_0099PE.1',
 'DP04_0100E.1',
 'DP04_0100PE.1',
```

```
'DP04_0102E',
'DP04_0103E',
'DP04_0103PE',
'DP04_0104E',
'DP04_0104PE',
'DP04_0105E',
'DP04_0105PE',
'DP04_0106E',
'DP04_0106PE',
'DP04_0107E',
'DP04_0107PE',
'DP04_0108E_y',
'DP04_0108PE_y',
'DP04_0109E_y',
'DP04_0110E_y',
'DP04_0110PE_y',
'DP04_0111E_y',
'DP04_0111PE_y',
'DP04_0112E_y',
'DP04_0112PE_y',
'DP04_0113E_y',
'DP04_0113PE_y',
'DP04_0114E_y',
'DP04_0114PE_y',
'DP04_0115E_y',
'DP04_0115PE_y',
'DP04_0116E_y',
'DP04_0117E_y',
'DP04_0120E_y',
'DP04_0120PE_y',
'DP04_0121E',
'DP04_0121PE',
'DP04_0122E',
'DP04_0122PE',
'DP04_0123E',
'DP04_0123PE',
'DP04_0124E',
'DP04_0124PE',
'DP04_0125E',
'DP04_0126E',
'DP04_0127E',
'DP04_0127PE',
'DP04_0128E',
```

```
 'DP04_0128PE',
 'DP04_0129E',
 'DP04_0129PE',
 'DP04_0130E',
 'DP04_0130PE',
 'DP04_0131E',
 'DP04_0131PE',
 'DP04_0132E',
 'DP04_0132PE',
 'DP04_0133E',
 'DP04_0133PE',
 'DP04_0134E',
 'DP04_0135E',
 'Unnamed: 98',
 'DP04_0137E',
 'DP04_0137PE',
 'DP04_0138E',
 'DP04_0138PE',
 'DP04_0139E',
 'DP04_0139PE',
 'DP04_0140E',
 'DP04_0140PE',
 'DP04_0141E',
 'DP04_0141PE',
 'DP04_0142E',
 'DP04_0142PE',
 'DP04_0143E']
```

Data analysis of the housing affordability data in 2014 and 2018

In [14]:
```
#import county boundary data set
tracts=gpd.read_file('NYMRcb/county.shp')
tracts.head()
```

...

In [15]:
```
#check if the format of statefp column is same
tracts.statefp.unique
```

...

In [16]:
```
#check if the format of countyfp column is same
tracts.countyfp.unique
```

...

In [17]:
```
#I noticed there are two rows with "-" between numbers, drop them to combine statefp and countyfp
tracts = tracts.drop(7)
tracts = tracts.drop(32)
```

In [62]:
```
#make a new column "FIPS" by combining statefp and countyfp
tracts["FIPS"]=tracts["statefp"]+tracts["countyfp"]
#transform FIPS into an integer to merge with hao2014 and hao2018
tracts["FIPS"] = tracts["FIPS"].astype(str).astype(int)
tracts.head
```

...

In [19]:
```
#merge tracts with hao2018 on FIPS plot it on the map
tracts2014=tracts.merge(hao2014,on="FIPS")
pd.set_option('display.max_columns', None)
tracts2014.head()
```

...

In [20]:
```
#merge the county boundary dataframe wiht 2018 housing affordability dataframe on FIPS
tracts2018=tracts.merge(hao2018,on="FIPS")
pd.set_option('display.max_columns', None)
tracts2018.head()
```

...

In [21]:

```
#convert the data type of columns in 2014 housing affordability dataframe from object to integers for mappi
tracts2014["DP04_0109E"] =tracts2014["DP04_0109E"].astype(str).astype(int)
tracts2014["DP04_0109PE"] =tracts2014["DP04_0109PE"].astype(float).astype(int)
tracts2014["DP04_0110E"] =tracts2014["DP04_0110E"].astype(str).astype(int)
tracts2014["DP04_0110PE"] =tracts2014["DP04_0110PE"].astype(float).astype(int)
tracts2014["DP04_0111E"] =tracts2014["DP04_0111E"].astype(float).astype(int)
tracts2014["DP04_0111PE"] =tracts2014["DP04_0111PE"].astype(float).astype(int)
tracts2014["DP04_0112E"] =tracts2014["DP04_0112E"].astype(float).astype(int)
tracts2014["DP04_0112PE"] =tracts2014["DP04_0112PE"].astype(float).astype(int)
tracts2014["DP04_0113E"] =tracts2014["DP04_0113E"].astype(float).astype(int)
tracts2014["DP04_0113PE"] =tracts2014["DP04_0113PE"].astype(float).astype(int)
tracts2014["DP04_0114E"] =tracts2014["DP04_0114E"].astype(float).astype(int)
```

In [22]:

```
#convert the data type of columns in 2018 housing affordability dataframe from object to integers for mappi
tracts2018["DP04_0111E_y"] =tracts2018["DP04_0111E_y"].astype(str).astype(int)
tracts2018["DP04_0111PE_y"] =tracts2018["DP04_0111PE_y"].astype(float).astype(int)
tracts2018["DP04_0112E_y"] =tracts2018["DP04_0112E_y"].astype(str).astype(int)
tracts2018["DP04_0112PE_y"] =tracts2018["DP04_0112PE_y"].astype(float).astype(int)
tracts2018["DP04_0113E_y"] =tracts2018["DP04_0113E_y"].astype(float).astype(int)
tracts2018["DP04_0113PE_y"] =tracts2018["DP04_0113PE_y"].astype(float).astype(int)
tracts2018["DP04_0114E_y"] =tracts2018["DP04_0114E_y"].astype(float).astype(int)
tracts2018["DP04_0114PE_y"] =tracts2018["DP04_0114PE_y"].astype(float).astype(int)
tracts2018["DP04_0115E_y"] =tracts2018["DP04_0115E_y"].astype(float).astype(int)
tracts2018["DP04_0115PE_y"] =tracts2018["DP04_0115PE_y"].astype(float).astype(int)
tracts2018["DP04_0116E_y"] =tracts2018["DP04_0116E_y"].astype(float).astype(int)
```

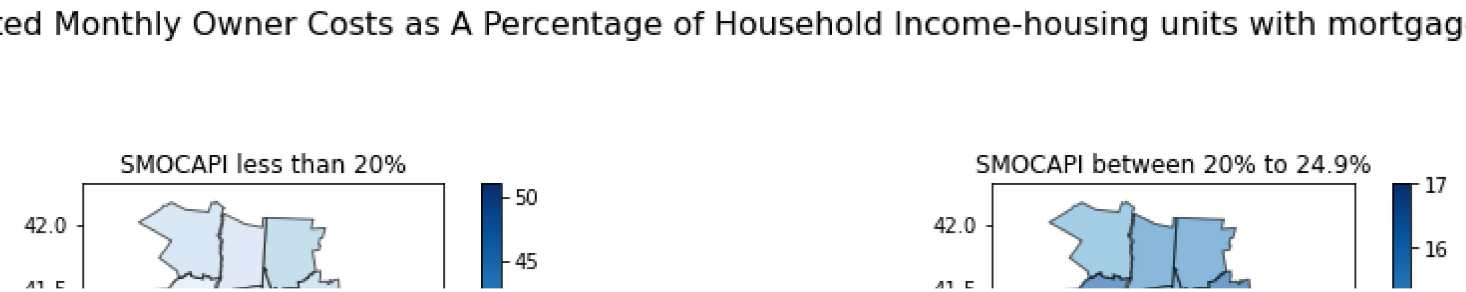
Mapping 2014 and 2018 housing affordability in NYMA

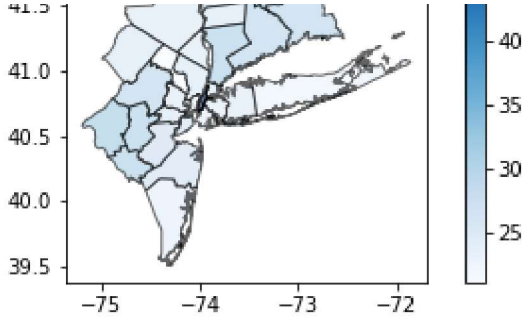
In [23]:

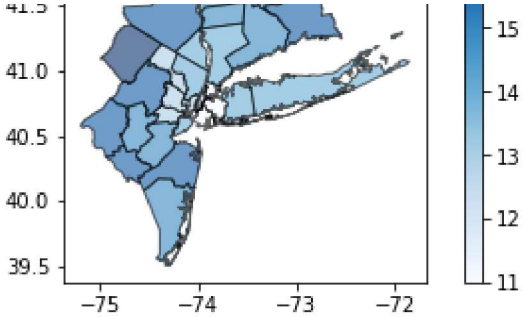
```
#plot 2014 SMOCAPI across counties in the NYMA region on the map
fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(15, 12)) #3 rows and 2 columns
(ax1, ax2), (ax3, ax4),(ax5, ax6) = axs

tracts2014.plot(
    figsize=(15,12),
    cmap='Blues',
    ax=ax1,
    column='DP04_0109PE',
    legend = True,
    edgecolor="Black",
    alpha=0.6
    )
ax1.set_title("SMOCAPI less than 20%")

tracts2014.plot(
    figsize=(15, 12),
     ax=ax2,
    cmap='Blues',
    column='DP04_0110PE',
    legend = True,
    edgecolor="Black",
    alpha=0.6
    )
ax2.set_title("SMOCAPI between 20% to 24.9%")

tracts2014.plot(
    figsize=(15, 12),
    cmap='Blues',
    column='DP04_0111PE',
    legend = True,
    ax=ax3,
    edgecolor="Black",
    alpha=0.6
)
ax3.set_title("SMOCAPI between 25% to 29.99%")

tracts2014.plot(
    figsize=(15, 12),
    cmap='Blues',
    column='DP04_0112PE',
    legend = True,
```

```
    ax=ax4,
    edgecolor="Black",
    alpha=0.6
)
ax4.set_title("SMOCAPI between 30% to 34.99%")

tracts2014.plot(
    figsize=(15, 12),
    cmap='Blues',
    column='DP04_0113PE',
    legend = True,
    ax=ax5,
    edgecolor="Black",
    alpha=0.6
)
ax5.set_title("SMOCAPI greater than 35%")

tracts2014.plot(
    figsize=(15, 12),
    cmap='Blues',
    column='DP04_0114E',
    legend = True,
    ax=ax6,
     edgecolor="Black",
    alpha=0.6
)
ax6.set_title("SMOCAPI not computed")


fig.suptitle('2014: Selected Monthly Owner Costs as A Percentage of Household Income-housing units with mor
```

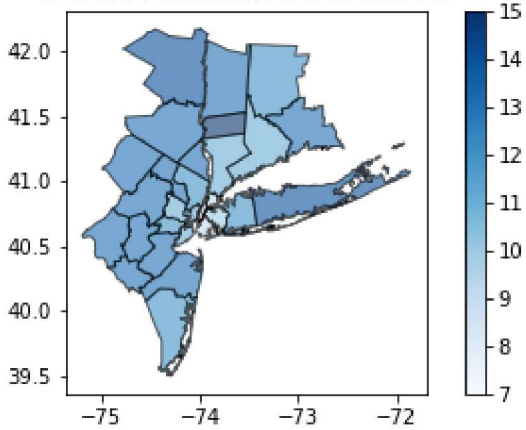
Out[23]: Text(0.5, 0.98, '2014: Selected Monthly Owner Costs as A Percentage of Household Income-housing units with mortgage')



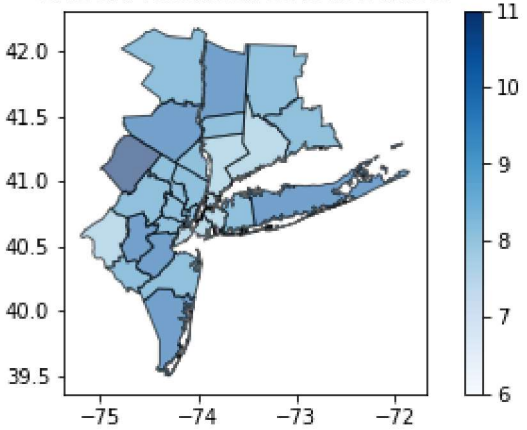




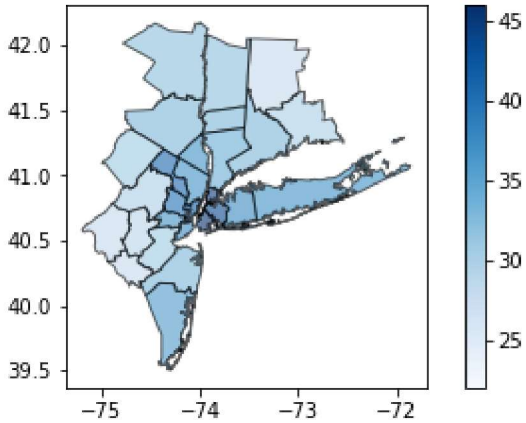
SMOCAPI between 25% to 29.99%



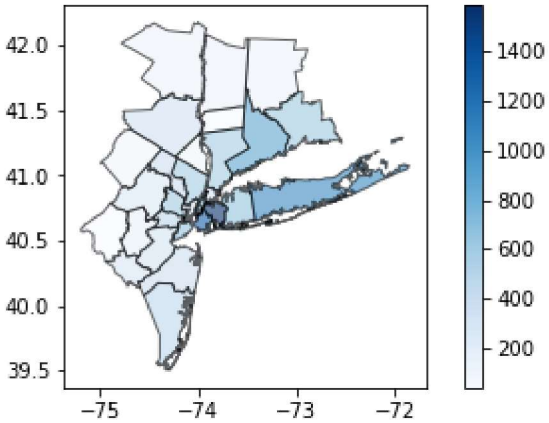
SMOCAPI between 30% to 34.99%



SMOCAPI greater than 35%



SMOCAPI not computed

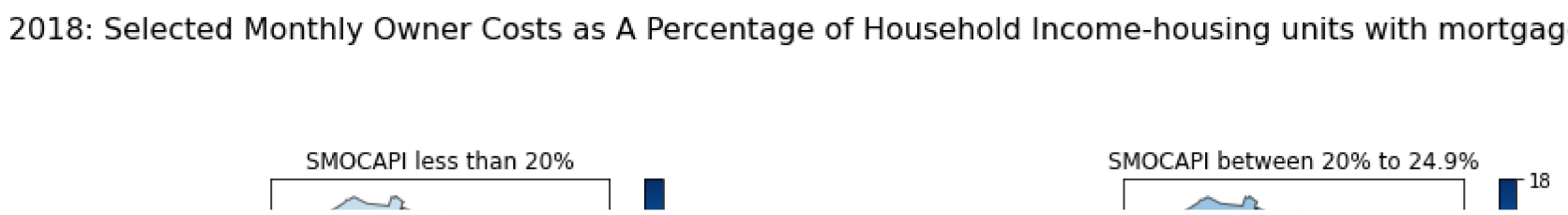


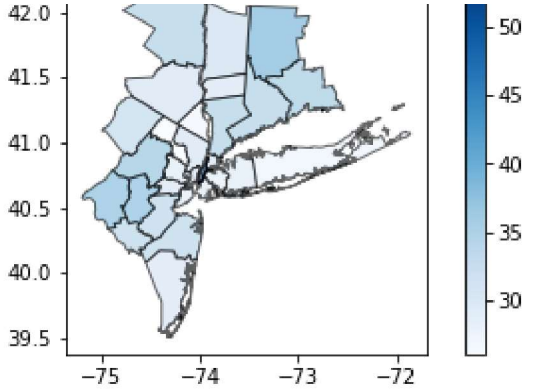
In [24]:

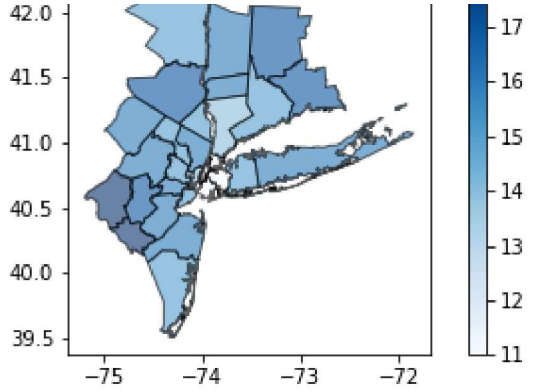
```
#plot 2018 SMOCAPI across counties in the NYMA region on the map

fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(15, 12))
(ax1, ax2), (ax3, ax4),(ax5, ax6) = axs


tracts2018.plot(
    figsize=(15, 12),
    cmap='Blues',
    column='DP04_0111PE_y',
    legend = True,
    edgecolor="Black",
    alpha=0.6,
    ax=ax1,
)
ax1.set_title("SMOCAPI less than 20%")

tracts2018.plot(
    figsize=(15, 12),
    cmap='Blues',
    column='DP04_0112PE_y',
    legend = True,
    edgecolor="Black",
    alpha=0.6,
    ax=ax2,
)
ax2.set_title("SMOCAPI between 20% to 24.9%")

tracts2018.plot(
    figsize=(15, 12),
    cmap='Blues',
    column='DP04_0113PE_y',
    legend = True,
    edgecolor="Black",
    alpha=0.6,
    ax=ax3,
)
ax3.set_title("SMOCAPI between 25% to 29.9%")

tracts2018.plot(
    figsize=(15, 12),
    cmap='Blues',
```

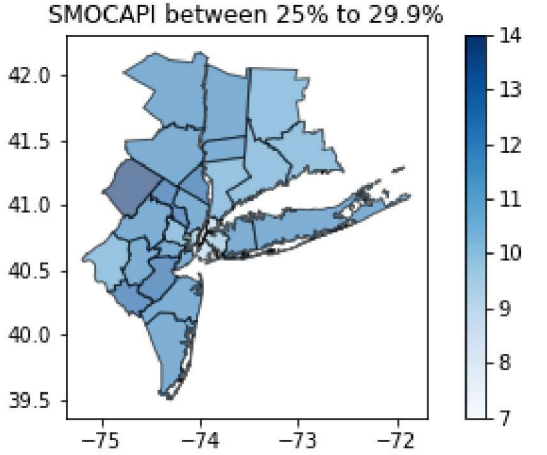
```
    column='DP04_0114PE_y',
    legend = True,
    ax=ax4,
    edgecolor="Black",
    alpha=0.6
)
ax4.set_title("SMOCAPI between 30% to 34.9%")

tracts2018.plot(
    figsize=(15, 12),
    cmap='Blues',
    column='DP04_0115PE_y',
    legend = True,
    ax=ax5,
    edgecolor="Black",
    alpha=0.6
)
ax5.set_title("SMOCAPI greater than 35%")

tracts2018.plot(
    figsize=(15, 12),
    cmap='Blues',
    column='DP04_0116E_y',
    legend = True,
    ax=ax6,
    edgecolor="Black",
    alpha=0.6
)
ax6.set_title("SMOCAPI not computed")


fig.suptitle('2018: Selected Monthly Owner Costs as A Percentage of Household Income-housing units with mor
```

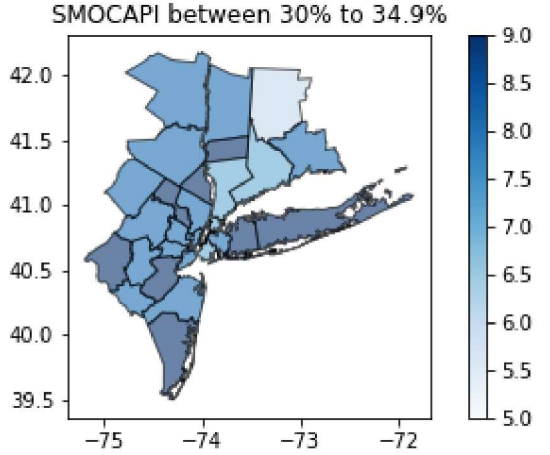
Out[24]: Text(0.5, 0.98, '2018: Selected Monthly Owner Costs as A Percentage of Household Income-housing units with mortgage')

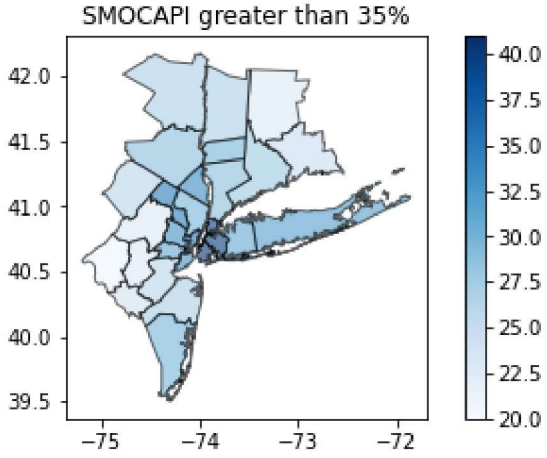


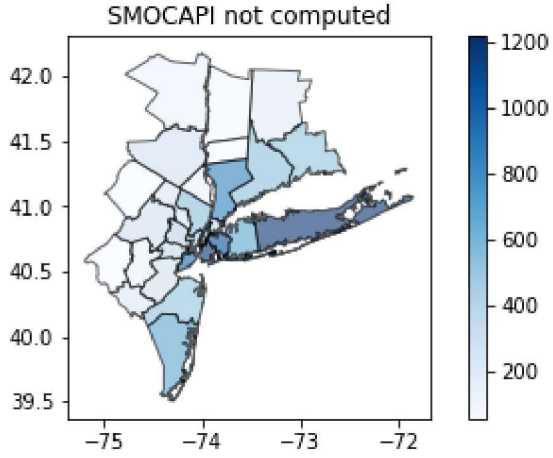












Comparing housing affordability in 2014 and 2018 with Plotly.Express

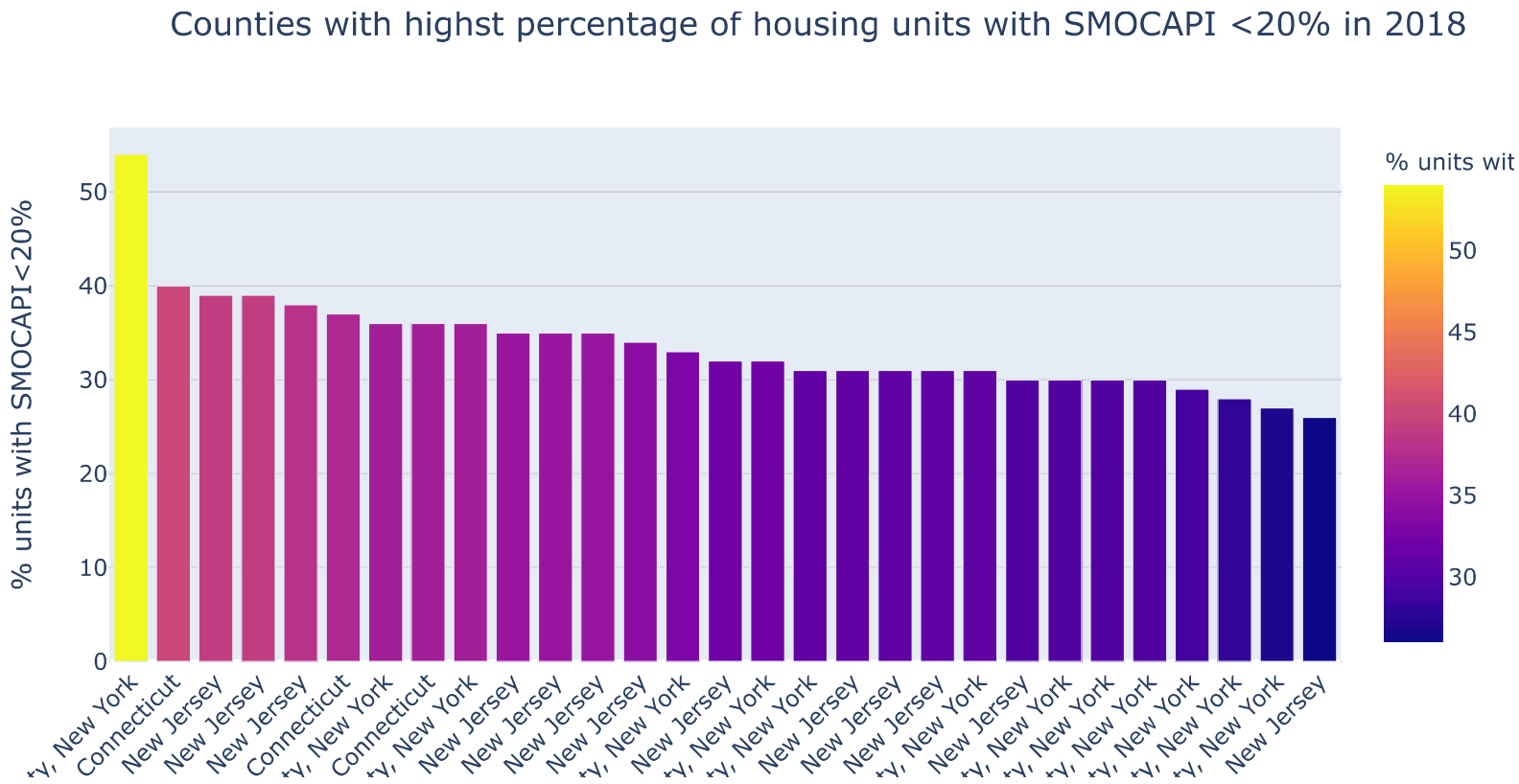
In [25]:

```
#US Census Bureau defined housing units with less than 20% SMOCAPI as affordable.
#create a new dataframe "SMOCAPI2018 by sorting the 2018housing units with less than 20% SMOCAPI
SMOCAPI2018=tracts2018.sort_values(by=['DP04_0111PE_y'], ascending=False)
SMOCAPI2018
```

...

In [26]:

```
#plot it
fig2=px.bar(
     SMOCAPI2018,
      x='name',
      y='DP04_0111PE_y',
    color='DP04_0111PE_y',
    labels={"DP04_0111PE_y": "Percent of housing units with SMOCAPI<20%", #change labels of x and y axis, a
        "name": "Counties",
        "DP04_0111PE_y": "% units with SMOCAPI<20%"
                }
)
fig2.update_layout(xaxis_tickangle=-45,title={
        'text': "Counties with highst percentage of housing units with SMOCAPI <20% in 2018", #add title
        'y':1,#change position of the title
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'}
)
```



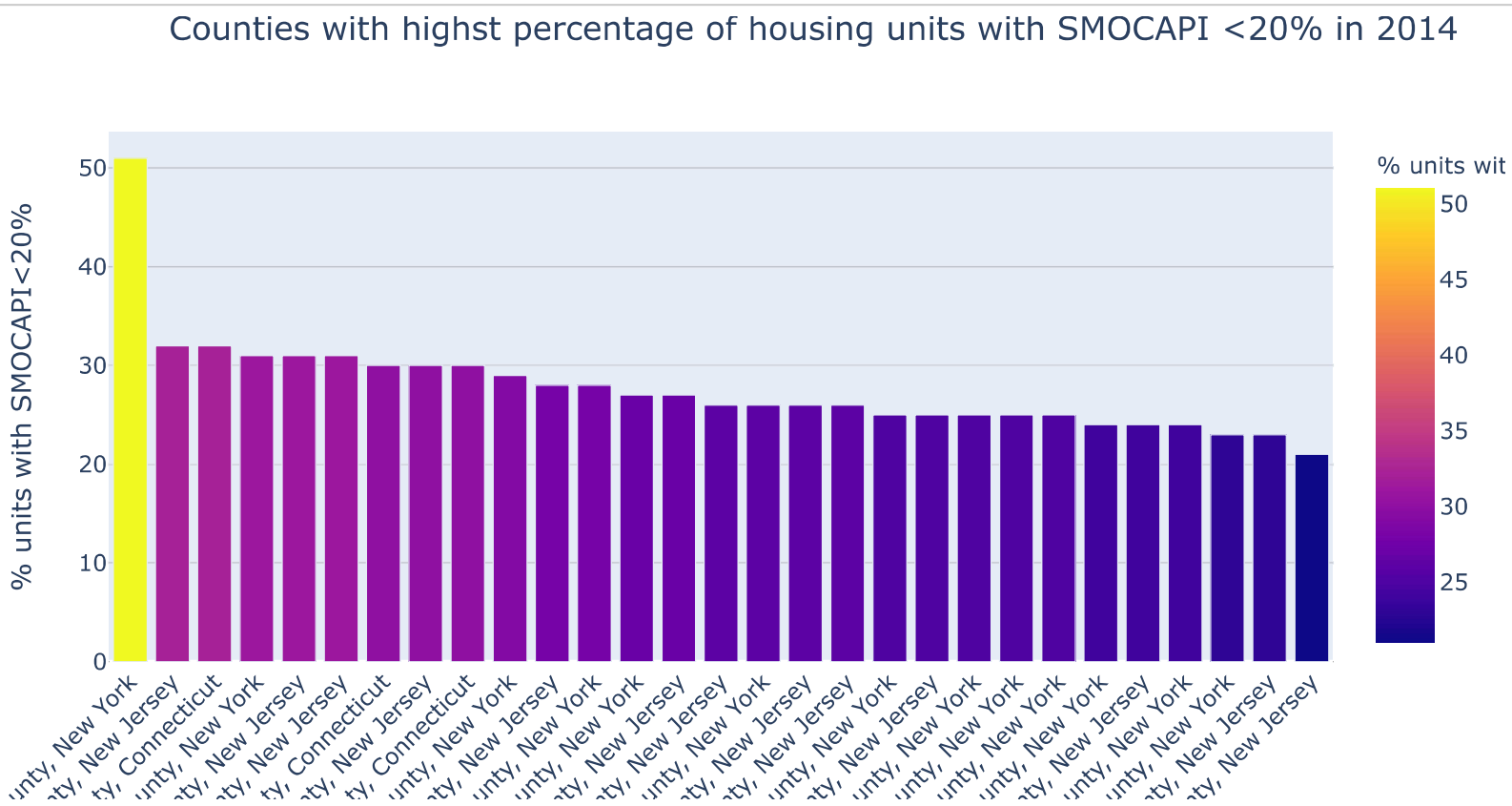
In [27]:

```
#create a new dataframe "SMOCAPI2014" by sorting the 2014 housing units with less than 20% SMOCAPI
SMOCAPI2014=tracts2014.sort_values(by=['DP04_0109PE'], ascending=False)
SMOCAPI2014
```

...

In [28]:

```
#plot it
fig2=px.bar(
     SMOCAPI2014,
      x='name',
      y='DP04_0109PE',
    color='DP04_0109PE',
    labels={"DP04_0109PE": "Percent of housing units with SMOCAPI<20%", #change labels of x and y axis, and
        "name": "Counties",
        "DP04_0109PE": "% units with SMOCAPI<20%"
                }
)
fig2.update_layout(xaxis_tickangle=-45,title={
        'text': "Counties with highst percentage of housing units with SMOCAPI <20% in 2014", #add title
        'y':1,#change position of the title
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
)
```



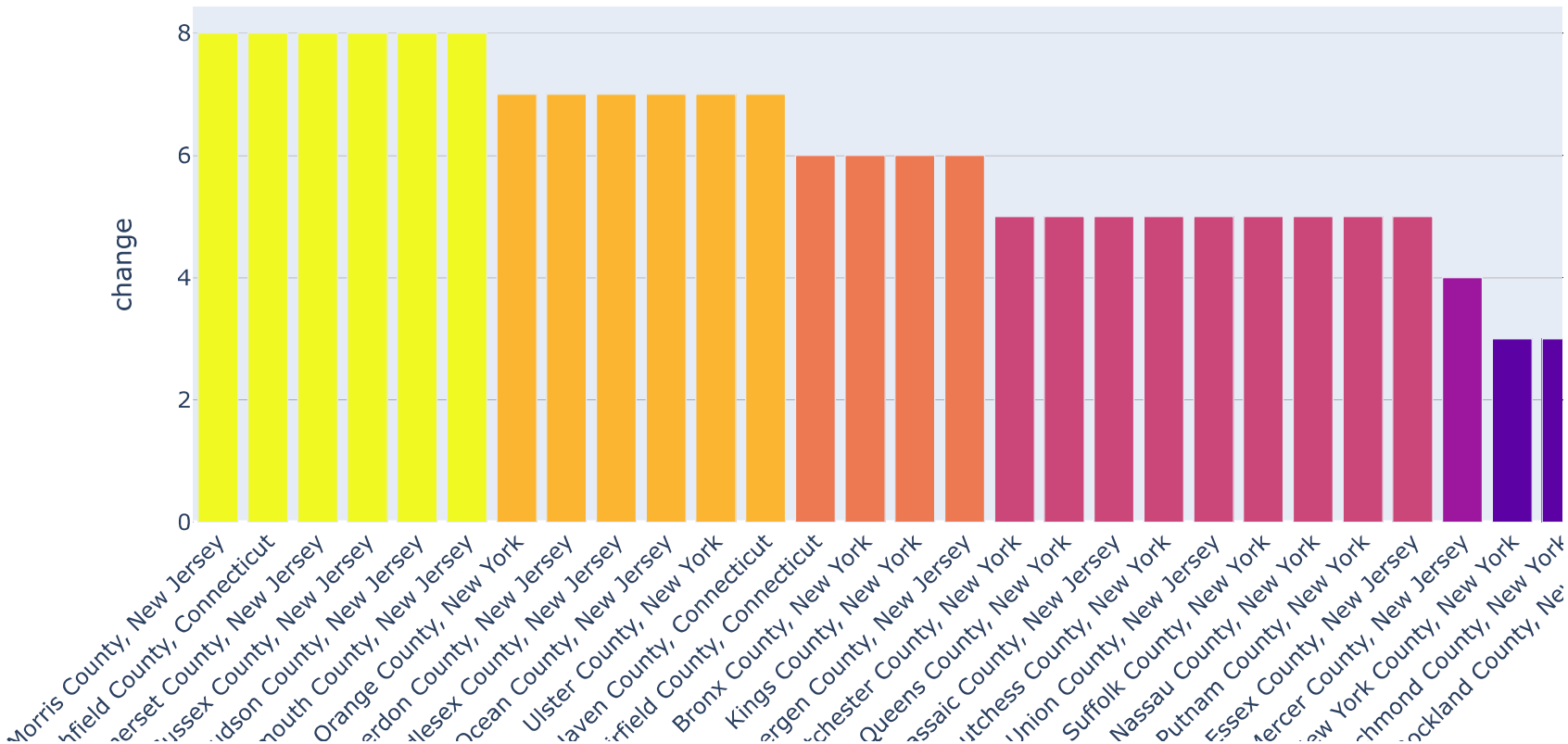
In [29]:

```
#I want to measure the change between 2014 and 2018 housing units with SMOCAPI<20% across counties in NYMA
tracts2018["change"]=SMOCAPI2018['DP04_0111PE_y']-SMOCAPI2014['DP04_0109PE'] #create new category "change" (
tracts2018
tracts2018=tracts2018.sort_values(by=['change'], ascending=False)
tracts2018
```

...

In [30]:

```
# Plot the change between the number of housing units with SMOCAPI<20% in 2014 and 2018
fig3=px.bar(
     tracts2018,
      x='name',
      y='change',
    color='change'
)
fig3.update_layout(xaxis_tickangle=-45,title={
        'text': "Percent change of housing units with SMOCAPI <20% between 2014 and 2018", #add title
        'y':1,#change position of the title
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
)
```

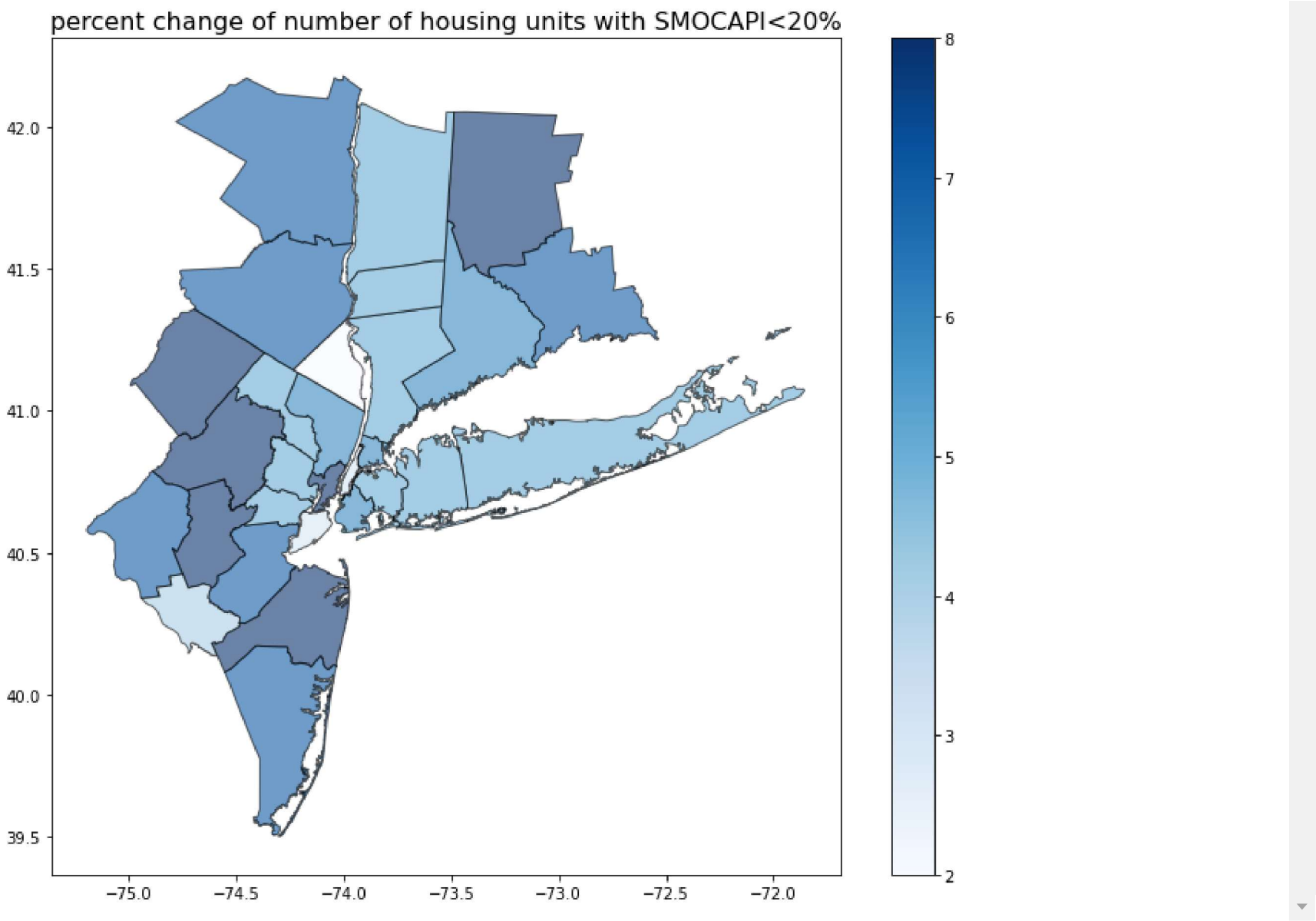


In [31]:

```
#plot it on the map
fig3=tracts2018.plot(figsize=(12,10),
    cmap='Blues',
    column='change',
    legend = True,
    edgecolor="Black",
    alpha=0.6)

plt.title('percent change of number of housing units with SMOCAPI<20%',fontsize=16)
```

Out[31]:

```
Text(0.5, 1.0, 'percent change of number of housing units with SMOCAPI<20%')
```



Conclusion

From the data exploration and analysis of housing affordability of owner occupied housing units with mortgage in NYMA, there is an overall increase in housing afforability in NYMA counties, ranging from two to eight percent. The highest increase is in counties further away from New York County (NYC), but also in Hudson County, which is located across from New York Conty.

Next step of the affordability analysis will be focused on the relationship between migration, housing affordability, and transit density on the county level.

Data exploration and analysis of housing value in 2014 and 2018

Data exploration

In [32]:
```
#importing 2014 and 2018 housing value data from ACS survey
value2014 = pd.read_csv('Book1.csv')
```

In [33]:
```
#looking at first 5 rows of the dataframe
value2014.head()
```

Out[33]:

	Value	Name	count1	year1	count2	year2	FIPS	change
0	Less than $50k	Fairfield County, Connecticut	4664	2014	4667	2018	901	0.000643
1	Less than $50k	Litchfield County, Connecticut	1248	2014	1214	2018	905	-0.027244
2	Less than $50k	New Haven County, Connecticut	5668	2014	4853	2018	909	-0.143790
3	Less than $50k	Bergen County, New Jersey	4207	2014	3109	2018	34003	-0.260994
4	Less than $50k	Essex County, New Jersey	2951	2014	3318	2018	34013	0.124365

In [34]:
```
#Check data types
value2014.dtypes
```

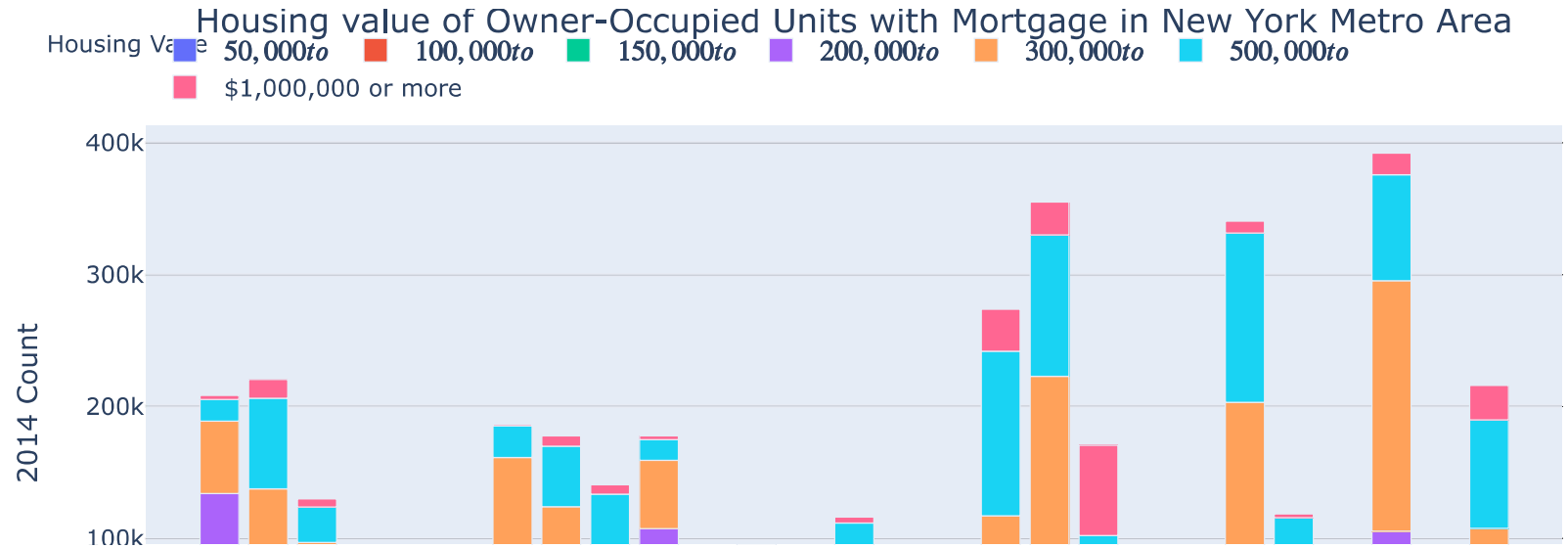
Out[34]:
```
Value      object
Name       object
count1      int64
year1       int64
count2      int64
year2       int64
FIPS        int64
change    float64
dtype: object
```

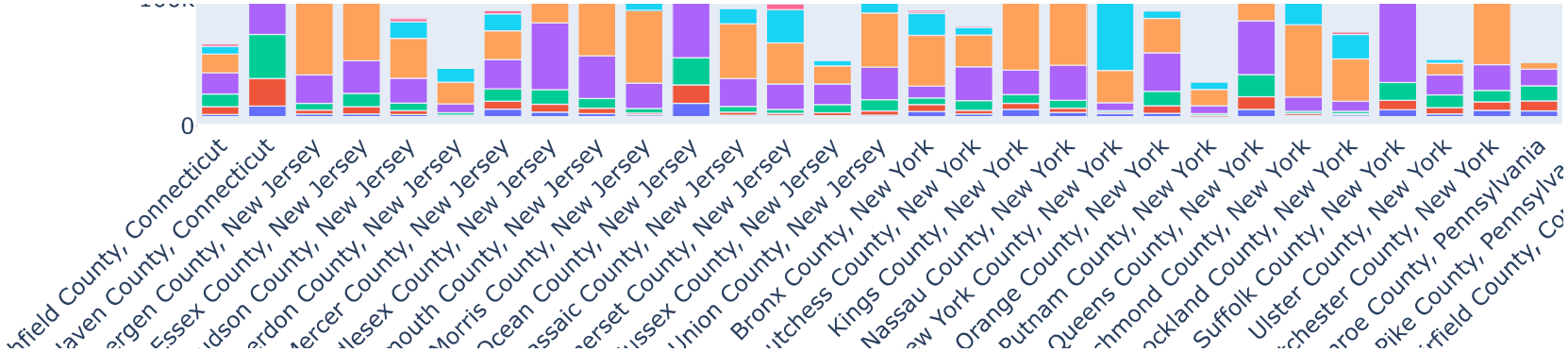
In [35]:
```
1 list(value2014)
```

Out[35]:
```
['Value', 'Name', 'count1', 'year1', 'count2', 'year2', 'FIPS', 'change']
```

In [36]:

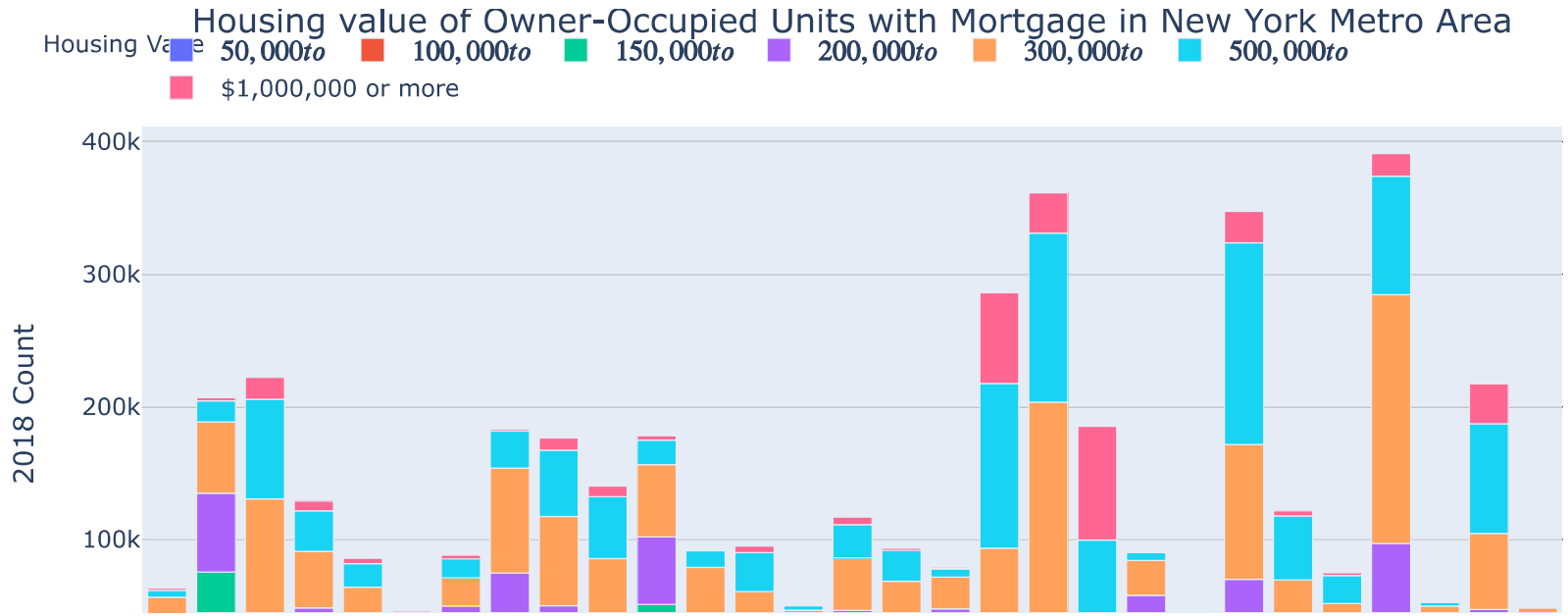
```
#Use plotly.express to create stacked bar chart of housing values in 2014 across counties in NYMA
Count = value2014.query("Value == ['Less than $50,000','$50,000 to $99,999','$100,000 to $149,999','$150,000
fig=px.bar(Count,
      x='Name',
      y='count1',
      color='Value',
      labels={"count1": "2014 Count", #change labels of x and y axis, and legend
              "Name": "Counties",
              "Value": "Housing Vaue"
                 })

fig.update_layout(xaxis_tickangle=-45,
      title={
        'text': "Housing value of Owner-Occupied Units with Mortgage in New York Metro Area",
        'y':1,#change position of the title
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
    legend=dict(
    orientation="h", #change the orientation of legend to horizontal
    yanchor="bottom",#change the position of the legend
    y=1.02,
    xanchor="right",
    x=.8))
fig
```

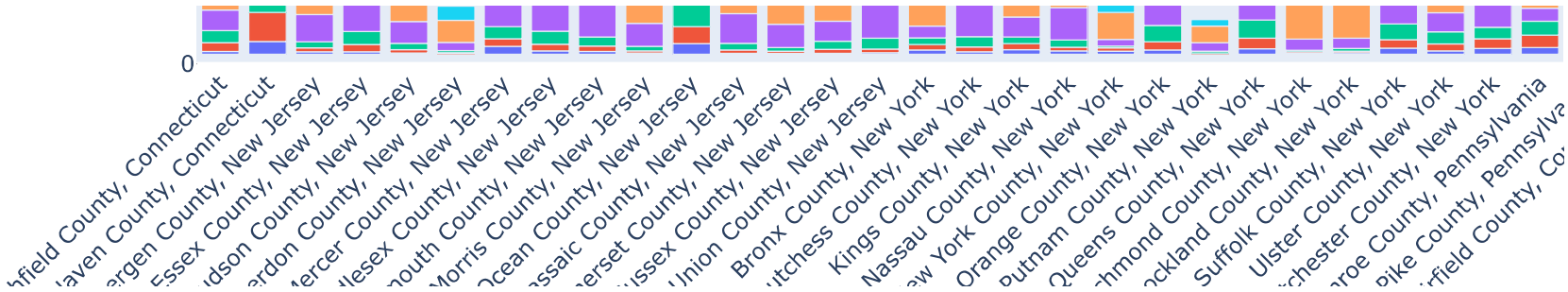



100k
0
hfield County, Connecticut
aven County, Connecticut
ergen County, New Jersey
Essex County, New Jersey
dson County, New Jersey
erdon County, New Jersey
Mercer County, New Jersey
Middlesex County, New Jersey
outh County, New Jersey
Morris County, New Jersey
Ocean County, New Jersey
assaic County, New Jersey
erset County, New Jersey
ussex County, New Jersey
Union County, New Jersey
Bronx County, New York
tchess County, New York
Kings County, New York
Nassau County, New York
ew York County, New York
Orange County, New York
Putnam County, New York
Queens County, New York
hmond County, New York
ockland County, New York
Suffolk County, New York
Ulster County, New York
chester County, New York
roe County, Pennsylvania
Pike County, Pennsylva
rfield County, Co

In [37]:

```
#Use plotly.express to create stacked bar chart of housing values in 2018 across counties in NYMA
Count = value2014.query("Value == ['Less than $50,000','$50,000 to $99,999','$100,000 to $149,999','$150,000
fig=px.bar(Count,
      x='Name',
      y='count2',
      color='Value',
      labels={"count2": "2018 Count", #change labels of x and y axis, and legend
        "Name": "Counties",
        "Value": "Housing Vaue"
                })
fig.update_layout(xaxis_tickangle=-45,
      title={
        'text': "Housing value of Owner-Occupied Units with Mortgage in New York Metro Area", #add title
        'y':1,#change position of the title
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
    legend=dict(
    orientation="h", #change the orientation of legend to horizontal
    yanchor="bottom", #change the position of the legend
    y=1.02,
    xanchor="right",
    x=.8))
```





Data analysis

In [38]:
```
#I want to measure the change between 2014 and 2018 housing values across value categories and counties in I
value2014["change"]=1-value2014["count1"] / value2014['count2'] #create new category "change" and calculate
value2014
```

Out[38]:

	Value	Name	count1	year1	count2	year2	FIPS	change
0	Less than $50k	Fairfield County, Connecticut	4664	2014	4667	2018	901	0.000643
1	Less than $50k	Litchfield County, Connecticut	1248	2014	1214	2018	905	-0.028007
2	Less than $50k	New Haven County, Connecticut	5668	2014	4853	2018	909	-0.167937
3	Less than $50k	Bergen County, New Jersey	4207	2014	3109	2018	34003	-0.353168
4	Less than $50k	Essex County, New Jersey	2951	2014	3318	2018	34013	0.110609
...	...	...	...	...	...	...	...	...
243	$1,000,000 or more	Suffolk County, New York	16345	2014	17122	2018	36103	0.045380
244	$1,000,000 or more	Ulster County, New York	654	2014	644	2018	36111	-0.015528
245	$1,000,000 or more	Westchester County, New York	26146	2014	30215	2018	36119	0.134668
246	$1,000,000 or more	Monroe County, Pennsylvania	234	2014	288	2018	42089	0.187500
247	$1,000,000 or more	Pike County, Pennsylvania	127	2014	154	2018	42103	0.175325

248 rows × 8 columns

In [39]:
```
#Group the data by counties and sort the change from high to low.
value2014.groupby("Name").change.describe().sort_values(by=["max"], ascending=False)
```

...

In [40]:
```
#create a new dataframe "top5change" to look at the top 5 counties across value categories that experienced
top10change=value2014.sort_values(by=['change'], ascending=False).head(10)
top10change
```

...

In [41]:

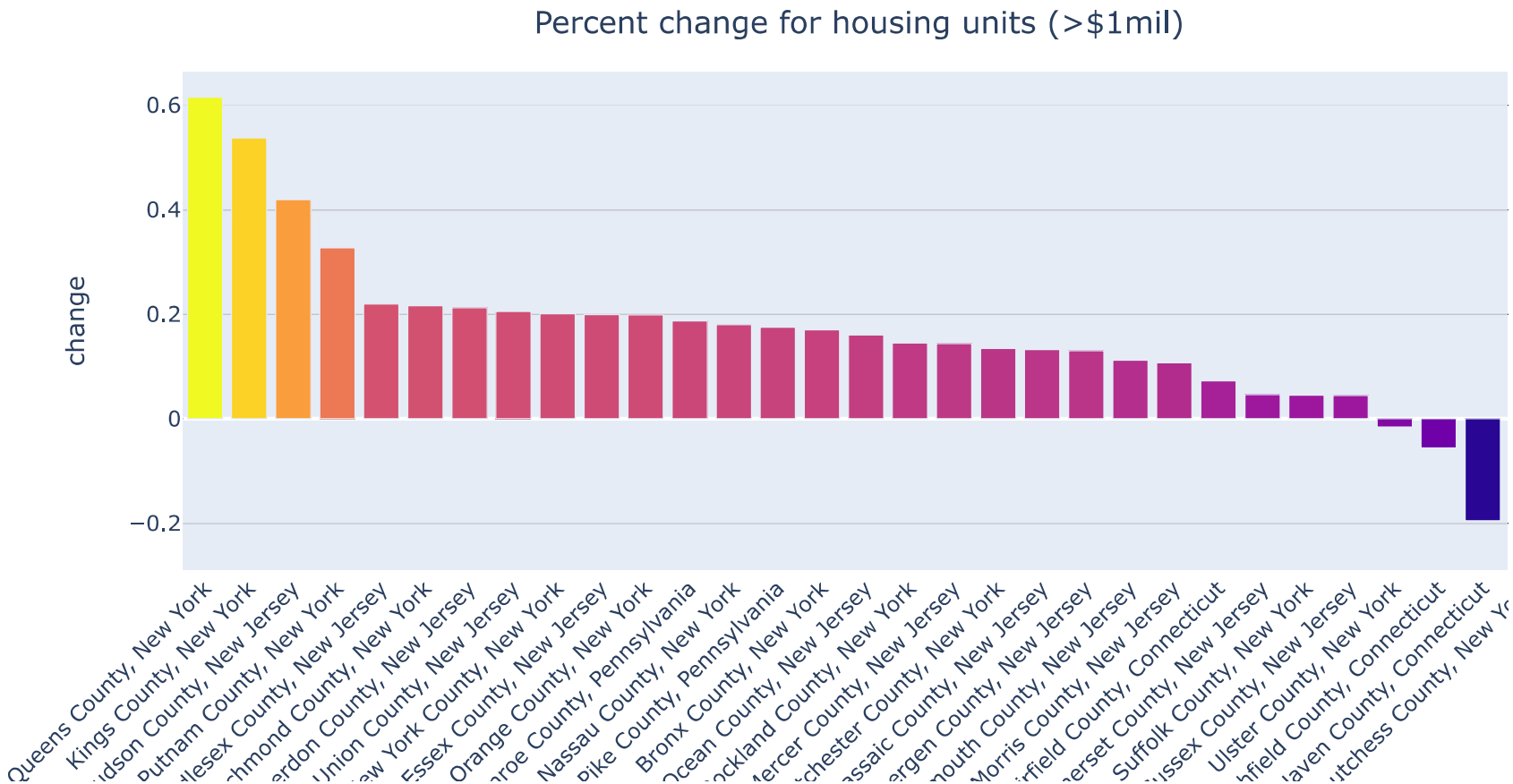
```
#create new dataframe topchange to see all the changes between 2014 and 2018 housing units count in each cou
topchange=value2014.sort_values(by=['change'], ascending=False)
topchange
```

...

Present the housing price change on the county level

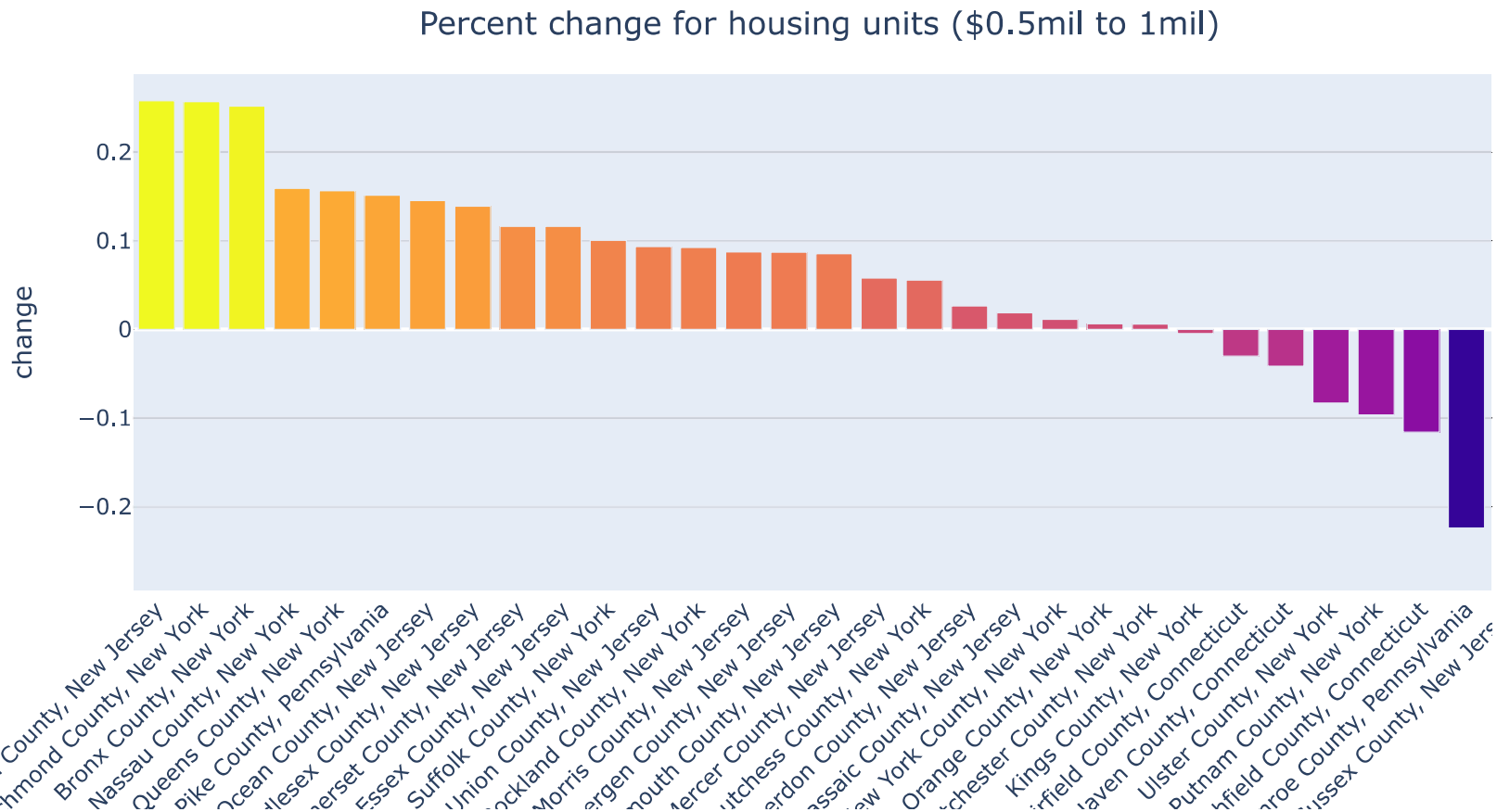
In [42]:

```
#I want to see the change of housing units in the value category "$1,000,000 or more"
temp = topchange.query("Value == ['$1,000,000 or more']")
fig5 = px.bar(temp,
       x='Name',
       y='change',
       color='change',
       labels={'Name':'County'})
fig5.update_layout(xaxis_tickangle=-45,title={
        'text': "Percent change for housing units (>$1mil)", #add title
        'y':0.95,#change position of the title
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'})
fig5
```



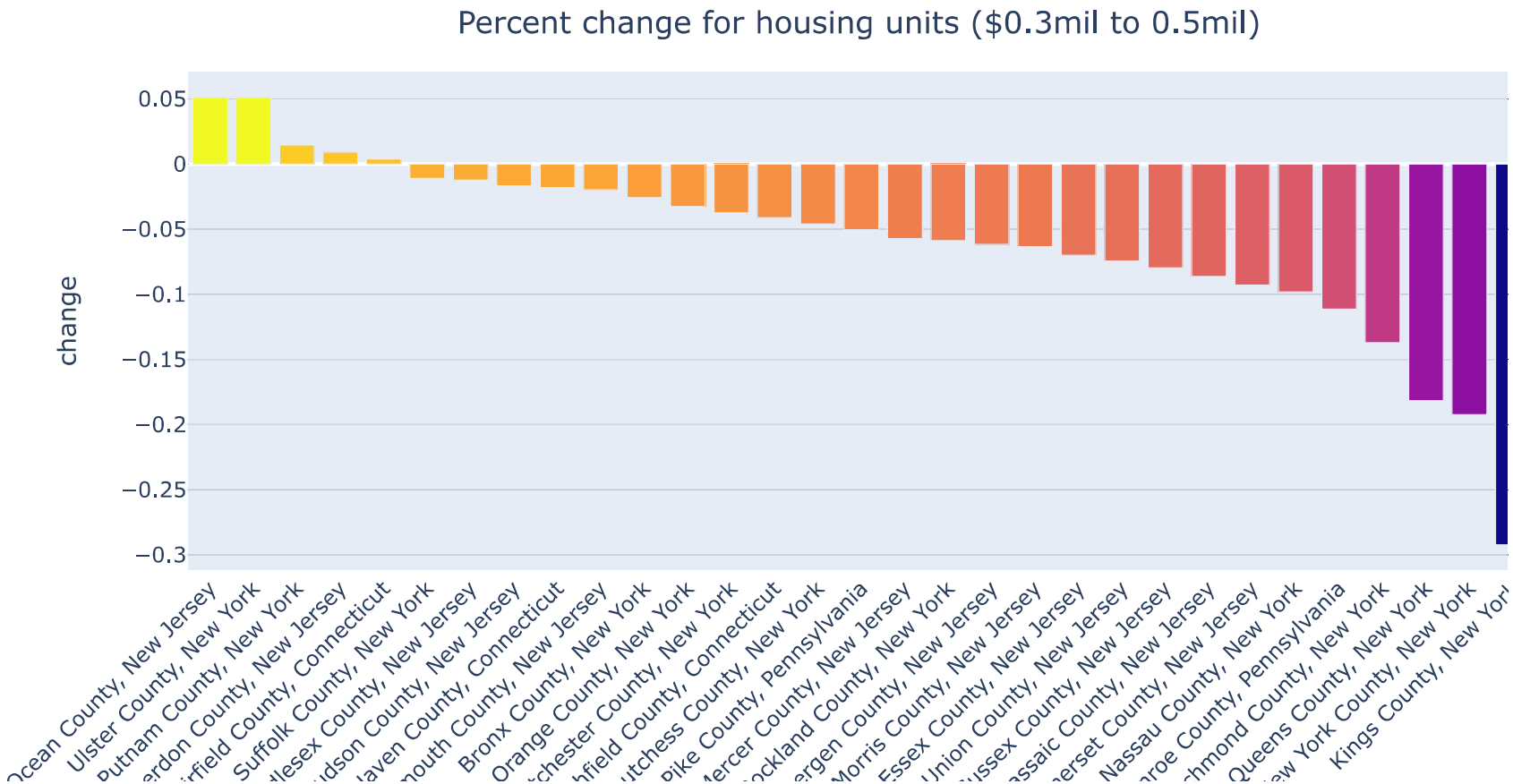
In [43]:

```
#I want to see the change of housing units in the value category "$500,000 to $999,999"
temp2 = topchange.query("Value == ['$500,000 to $999,999']")
fig6 = px.bar(temp2,
       x='Name',
       y='change',
       color='change',
       labels={'Name':'County'})
fig6.update_layout(xaxis_tickangle=-45,title={
        'text': "Percent change for housing units ($0.5mil to 1mil)", #add title
        'y':0.95,#change position of the title
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'})
fig6
```



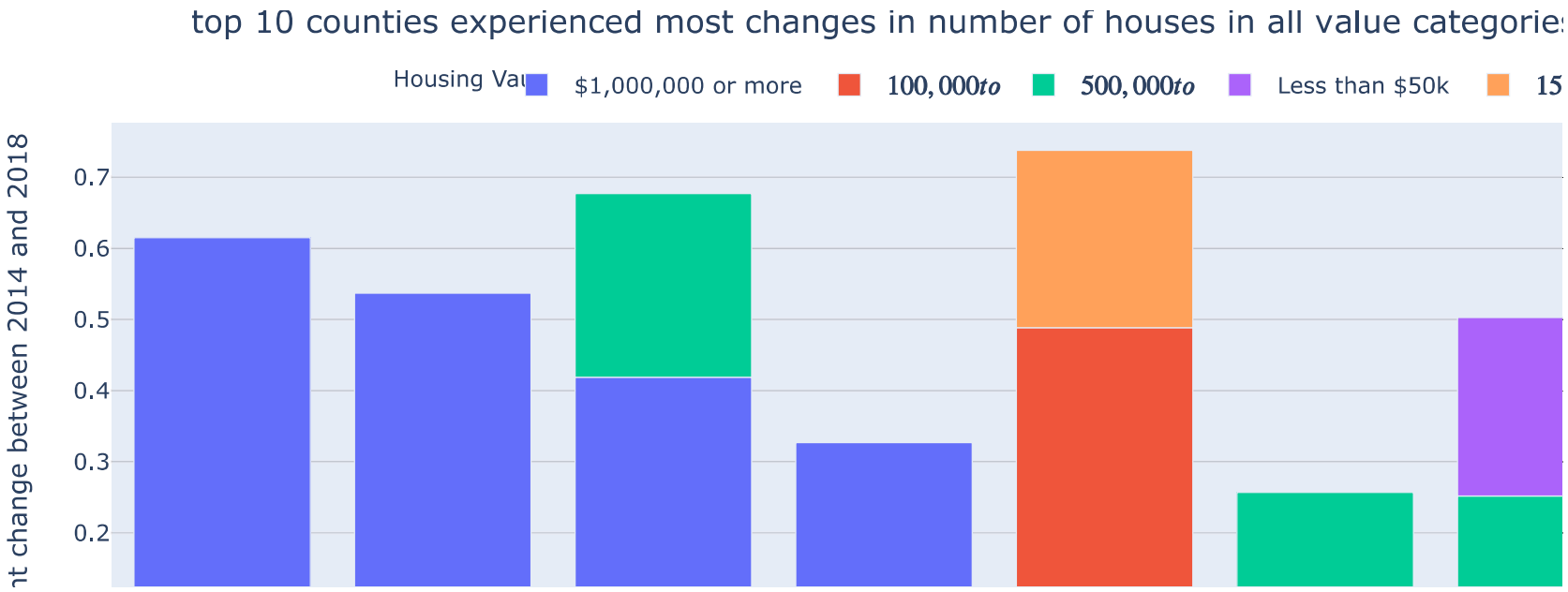
In [44]:

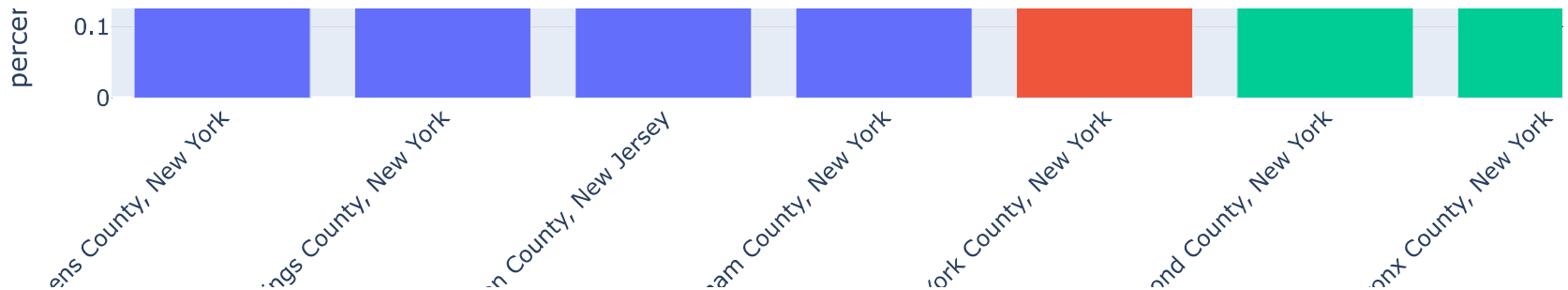
```
#I want to see the change of housing units in the value category "$300,000 to $499,999"
temp3 = topchange.query("Value == ['$300,000 to $499,999']")
fig7 = px.bar(temp3,
       x='Name',
       y='change',
       color='change',
       labels={'Name':'County'})
fig7.update_layout(xaxis_tickangle=-45,title={
        'text': "Percent change for housing units ($0.3mil to 0.5mil)", #add title
        'y':0.95,#change position of the title
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'})
fig7
```



In [45]:

```python
#create interactive bar graphs for the top 10 counties and distinguish the value categories
fig2=px.bar(
      top10change,
      x='Name',
      y='change',
      color='Value',
      labels={"change": "percent change between 2014 and 2018", #change labels of x and y axis, and legend
        "Name": "Counties",
        "Value": "Housing Vaue"
                }
)
fig2.update_layout(xaxis_tickangle=-45,title={
        'text': "top 10 counties experienced most changes in number of houses in all value categories", #ad
        'y':1,#change position of the title
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
    legend=dict(
    orientation="h",
    yanchor="bottom",
    y=1.02,
    xanchor="right",
    x=1
))
```





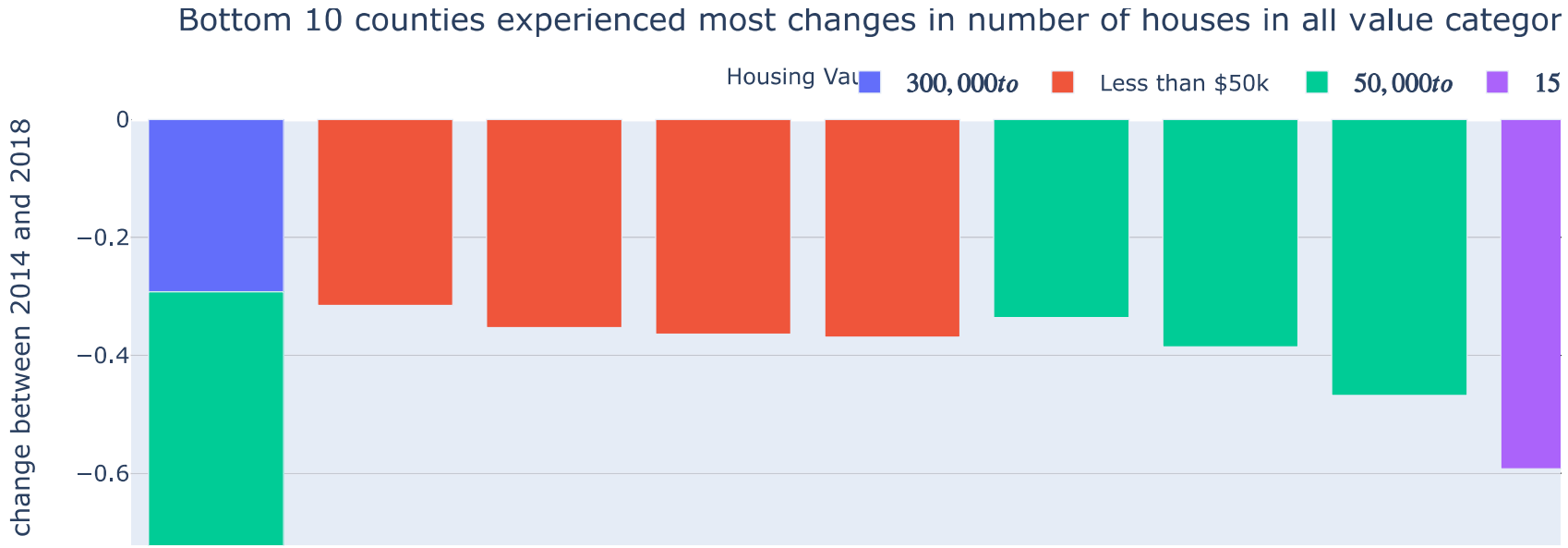
In [46]:

```
#look at the counties with decline in housing units within each value category
bottomchange=value2014.sort_values(by=['change'], ascending=False).tail(10)
bottomchange
```

...

In [47]:

```
#create interactive bar graphs for the botton 10 counties and distinguish the value categories
fig2=px.bar(
      bottomchange,
      x='Name',
      y='change',
      color='Value',
      labels={"change": "percent change between 2014 and 2018", #change labels of x and y axis, and legend
        "Name": "Counties",
        "Value": "Housing Vaue"
                }
)
fig2.update_layout(xaxis_tickangle=-45,title={
        'text': "Bottom 10 counties experienced most changes in number of houses in all value categories",
        'y':1,#change position of the title
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
    legend=dict(
    orientation="h",
    yanchor="bottom",
    y=1.02,
    xanchor="right",
    x=1
))
```



Map the changes of housing price in NYMA between 2014 and 2018

In [48]:
```
#import county boundary data set
tracts=gpd.read_file('NYMRcb/county.shp')
tracts.head()
```

...

In [49]:
```
#looking at datatypes
tracts.dtypes
```

...

In [50]:
```
#I knew I need to drop row 7 and 32 from the previous analysis because they contain different geographic reg
tracts = tracts.drop(7)
```

In [51]:
```
tracts = tracts.drop([32])
```

In [52]:
```
# create new column "FIPS" by combining the "statefp" column anhd "countyfp" column
tracts["FIPS"]=tracts["statefp"]+tracts["countyfp"]
```

In [53]:
```
#convert FIPS from object to integer to merge this dataframe with housing affordability dataframe later
tracts["FIPS"] = tracts["FIPS"].astype(str).astype(int)
```

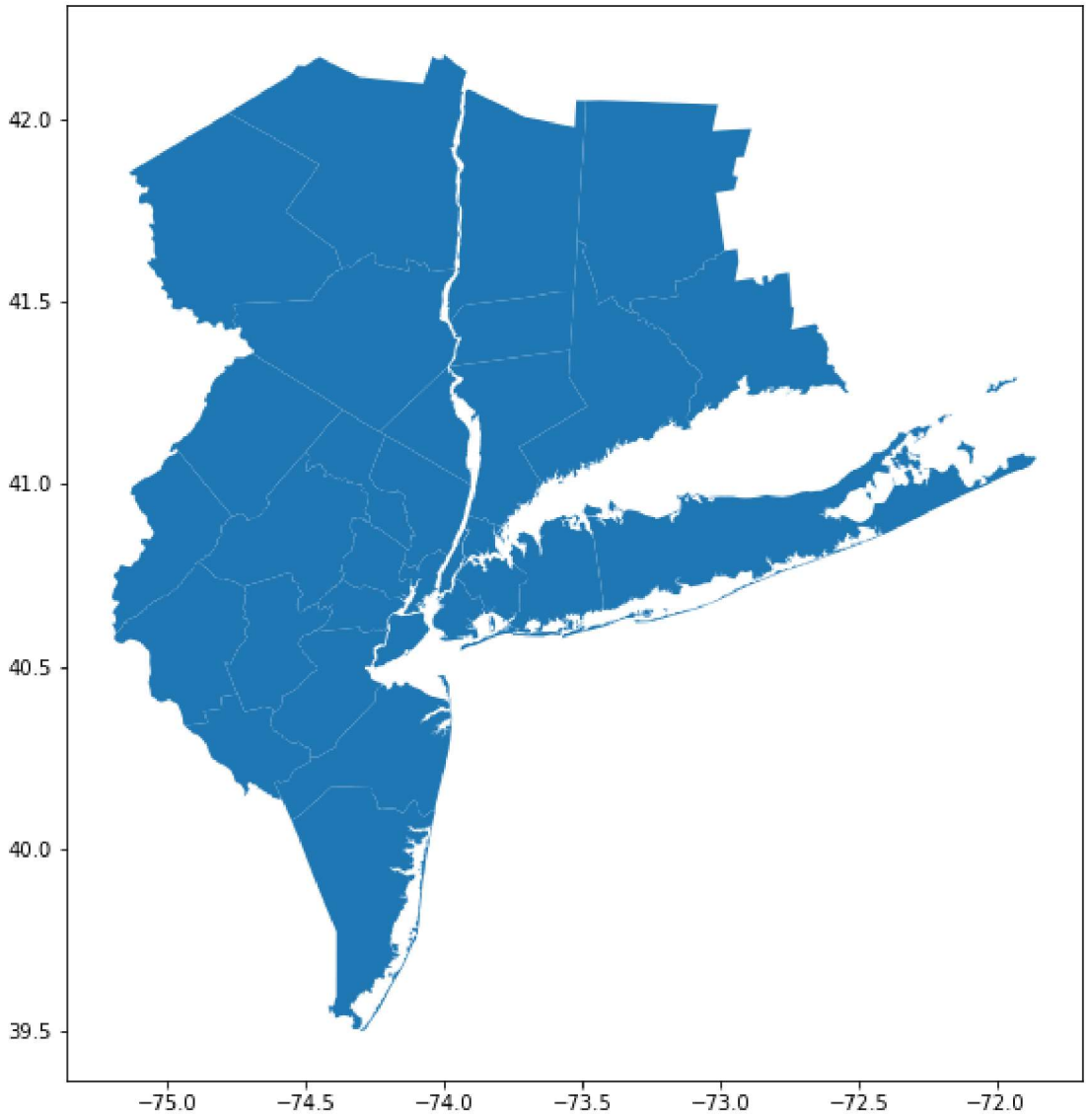
In [54]:

```
1 tracts.head()
```

...

In [55]:
```
#plot the county boundary
tracts.plot(figsize=(12,10))
```

Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0x7f0f4e8df5b0>



In [56]:

```
#map changing of housing units in the value category of more than $1,000,000 by merging the dataframe with c
cbtop1m=tracts.merge(temp,on="FIPS")
cbtop1m.head()
```

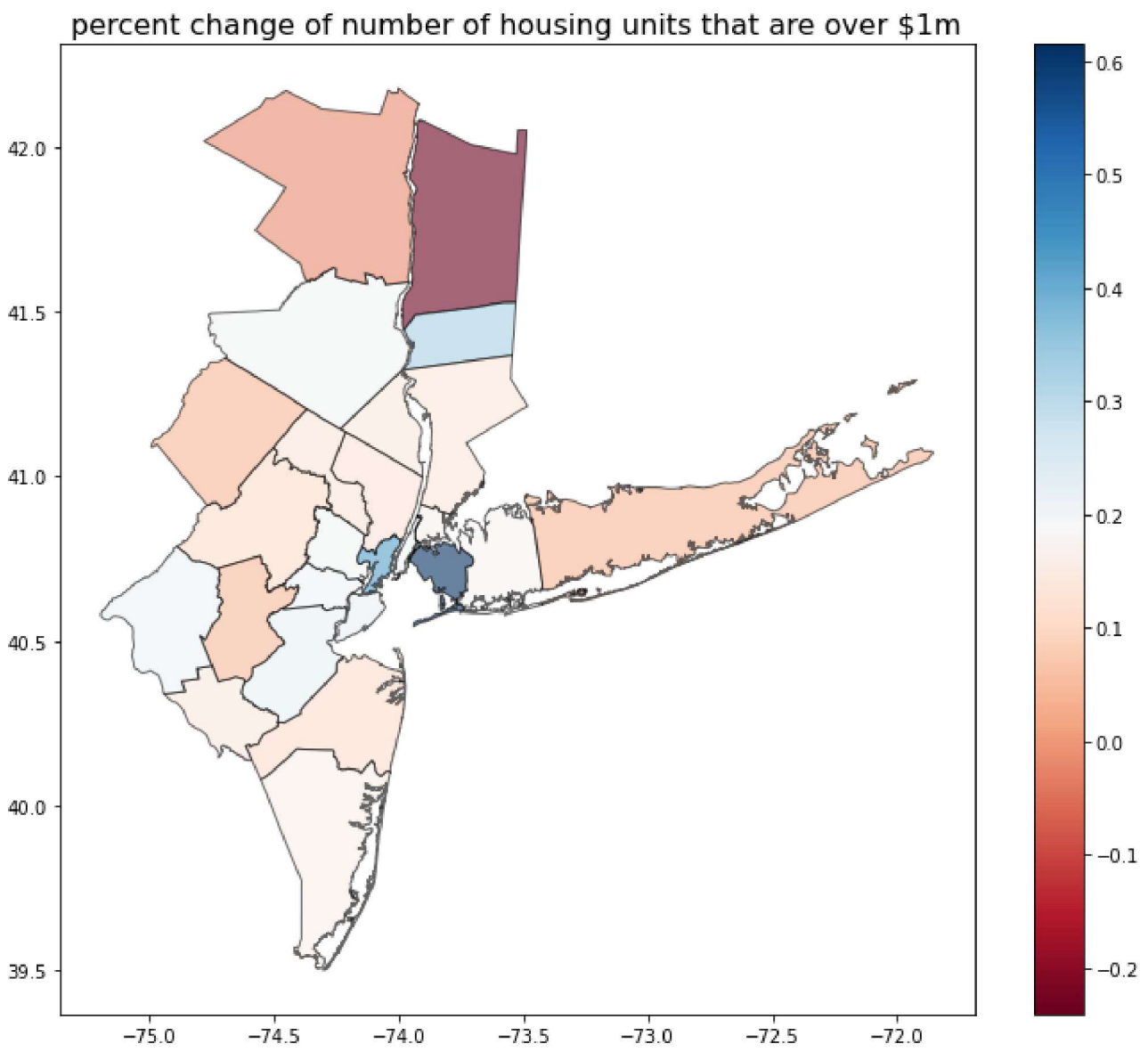
...

In [57]:

```
#plot it
figtop1m=cbtop1m.plot(figsize=(12,10),
    cmap='RdBu',
    column='change',
    legend = True,
    edgecolor='Black',
    alpha=0.6)

plt.title('percent change of number of housing units that are over $1m',fontsize=16)
```

Out[57]:

```
Text(0.5, 1.0, 'percent change of number of housing units that are over $1m')
```



In [58]:

```
1 #map changing of housing units in the value category between $500,000 to $999,999 by merging the dataframe
2 cbtophalf=tracts.merge(temp2,on="FIPS")
3 cbtop1m.head()
```

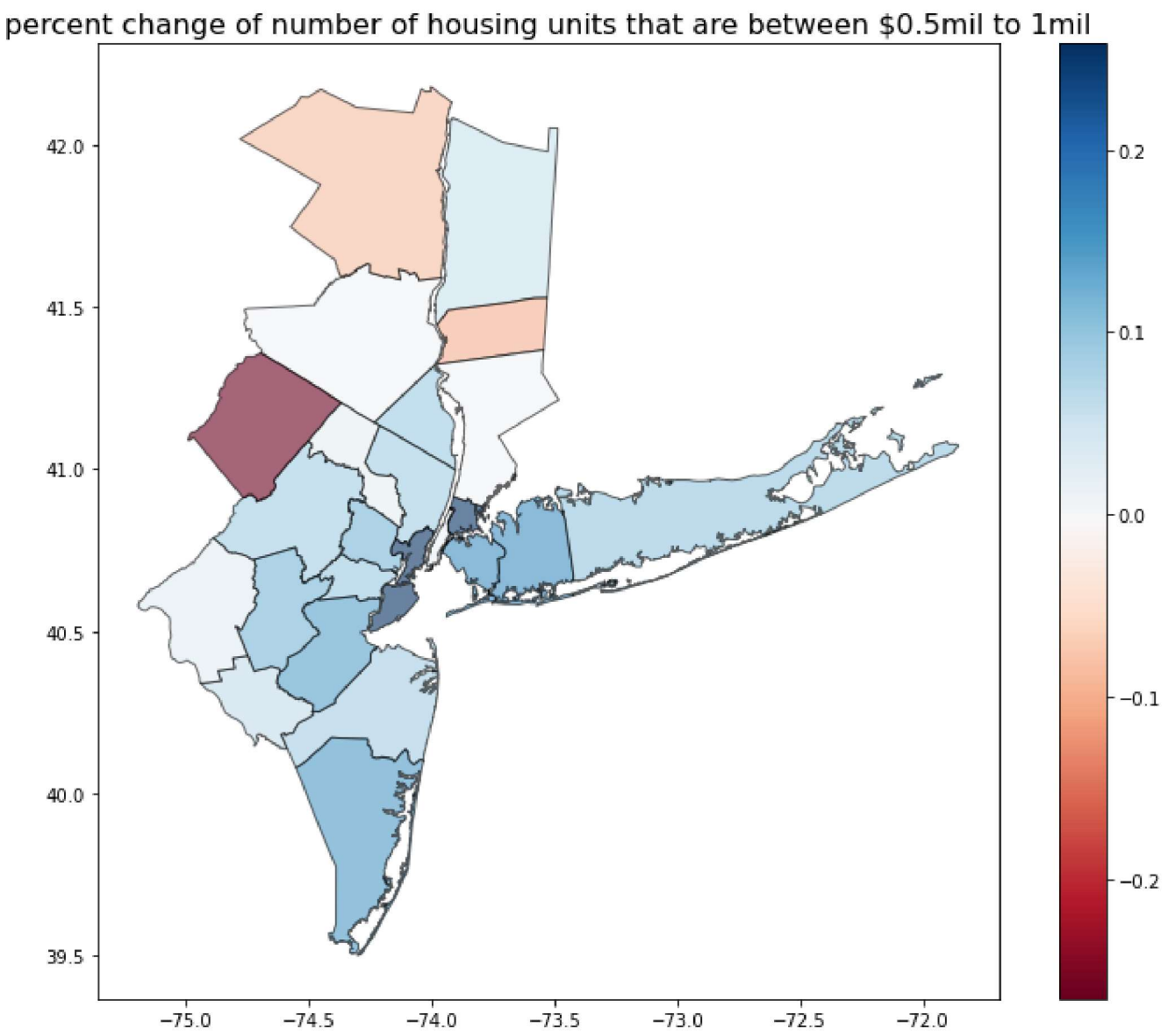
...

In [59]:

```
figtophalf=cbtophalf.plot(figsize=(12,10),
    cmap='RdBu',
    column='change',
    legend = True,
    edgecolor='Black',
    alpha=0.6)

plt.title('percent change of number of housing units that are between $0.5mil to 1mil',fontsize=16)
```

Out[59]:

```
Text(0.5, 1.0, 'percent change of number of housing units that are between $0.5mil to 1mil')
```



In [60]:

```
1 cbtop3to5=tracts.merge(temp3,on="FIPS")
2 cbtop3to5.head()
```

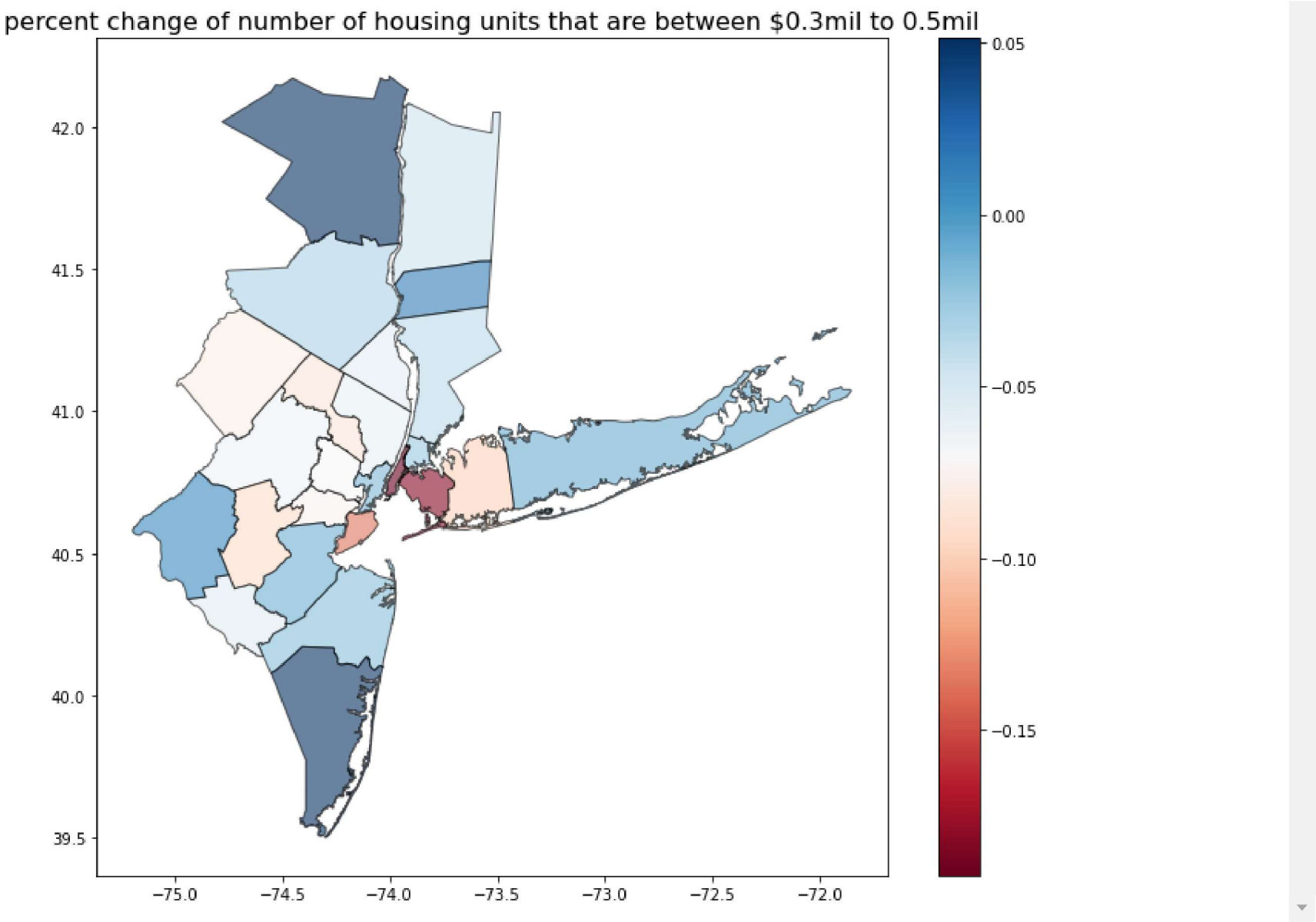
...

In [61]:

```
figtophalf=cbtop3to5.plot(figsize=(12,10),
    cmap='RdBu',
    column='change',
    legend = True,
    edgecolor='Black',
    alpha=0.6)

plt.title('percent change of number of housing units that are between $0.3mil to 0.5mil',fontsize=16)
```

Out[61]:

```
Text(0.5, 1.0, 'percent change of number of housing units that are between $0.3mil to 0.5mil')
```



Conclusion about housing price

From the data exploration and analysis of housing value in 2014 and 2018, all counties experienced some changes in housing price. For counties close to New York County (NYC), there is an increase in housing units with higher price (higher than $500,000$), $and decrease in housing units with lower price (below 500,000$).

Specifically, counties near New York County (NYC) experienced the highest increase in housing units in the more than $1 million value category. Queens county increased more than 60%, Kings county increased 53.7%, Hudson County(across from NYC) increased 41.9%, and the further aweay Putnam County increased 32.7%.

Similar geographic pattern of increase holds true for the value category between $500,000 to$ 999,999. In Hudson County, which is located across from New York County (NYC), housing units between $500,000 to$ 999,999 increased 25.8%, Richmond County, which is located mainly in Staten Island (below NYC), units increased 25.7%, and in Bronx, which is above NYC, housing units increased 25.2%. In counties further away from New York County, there is also a significant but smaller increase in housing units in the $500,000 to$ 999,999 value category.

For lower value categories, the housing units near New York County experienced an overall decine in number of units, ranging from minor to significant decrease. Kings County (Brooklyn) experienced 29.2% decrease in housing units within this value category, New York county itself experience 19.2% decrese, Queens County experience 18.2% decrease, and Richmond County experience 13.7% decrease. However, there are some counties nearby that experience a minor descrease as well. For counties further away from NYC, some experienced a minor decrease and some experienced a minor increase of housing units in this value category.

Conclusion on housing affordability and housing value in NYMA between 2014 and 2018

In this data exploration and analysis notebook, I conducted basic data exploration of housing affordability datasets, housing value datasets and county boundary datasets. I analyzed and mapped housing affordability anbd housing value in 2014 and 2018 across 31 counties in New York Metro Area. In the future weeks during this quarter, I will conduct more analysis on the impact of migration on housing price and affordability in NYMA.