Introduction:

This project begins within the dynamic landscape of a fictitious wholesale club meticulously crafted for analytical exploration. Drawing inspiration from the financial nuances of BJ's Wholesale Club in 2022 and 2021, the core financial framework of our wholesale club emerges. However, every other facet of information is generated through the artistry of randomization functions, with defined ranges and weights.

At its essence, this endeavor is driven by the ambition to simulate a comprehensive dataset reflective of a wholesale club company. The journey ahead involves a deep dive into the outcomes of this deliberate randomness. Following the initial exploration, the project transforms into a simulated scenario, prompting a crucial question: what recommendations would be made for the wholesale club's expansion strategy if tasked with a preliminary analysis? Nestled exclusively along the east coast, the company faces the challenge of strategically choosing from six selected states in its surrounding geography for potential expansion.

The narrative unfolds as we navigate through the simulated experience, seamlessly transitioning from scrutinizing the dataset for analytical insights to embarking on a journey of internet research and census data exploration. This project encapsulates the synergy between data exploration and real-world decision-making, encapsulating the intricate process of transforming raw data into actionable recommendations.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns

store_data = pd.read_csv("wholesale_store_data.csv")
```

In [2]:
```
store_data.columns
```

Out[2]:
```
Index(['Store Numbers', 'City', 'State', 'City Code', ' 2022 Gross Revenue ',
       ' 2021 Gross Revenue ', ' 2022 Gross Profit ', ' 2021 Gross Profit ',
       ' 2022 Expenses ', ' 2021 Expenses ', ' 2022 Net Income ',
       ' 2021 Net Income ', 'Total Members', 'Avg_Mbr_Length',
       'Premium Member', 'Budget Member', 'Mainstream Member', 'Column1',
       'Column2', 'Column3', 'Column4', 'Column5', 'Column6', 'Column7',
       'RETIREES', 'OLDER SINGLES/COUPLES', 'OLDER FAMILIES', 'YOUNG FAMILIES',
       'YOUNG SINGLES/COUPLES', 'MIDAGE FAMILIES', 'MIDAGE SINGLES/COUPLES'],
      dtype='object')
```

In [3]:
```
#dropping unneeded helper columns from original dataset.
columns_dropping =['City Code','Column1',
       'Column2', 'Column3', 'Column4', 'Column5', 'Column6', 'Column7']

store_data.drop(columns=columns_dropping, inplace=True)
```

In [4]:
```
store_data.describe()
```

Out[4]:

	Store Numbers	2022 Gross Revenue	2021 Gross Revenue	2022 Gross Profit	2021 Gross Profit	2022 Expenses	2021 Expenses	202 Ir
count	272.000000	2.720000e+02	2.720000e+02	2.720000e+02	2.720000e+02	2.720000e+02	2.720000e+02	2.72000
mean	136.500000	4.322046e+07	3.943003e+07	7.606801e+06	6.860824e+06	6.465020e+06	5.852283e+06	1.14178
std	78.663842	7.037043e+06	6.419894e+06	1.238520e+06	1.117062e+06	1.052618e+06	9.528535e+05	1.85901
min	1.000000	2.804482e+07	2.558529e+07	4.935889e+06	4.451841e+06	4.195012e+06	3.797420e+06	7.40877

25%	68.750000	3.928894e+07	3.584330e+07	6.914853e+06	6.236734e+06	5.876934e+06	5.319934e+06	1.03791
50%	136.500000	4.420396e+07	4.032727e+07	7.779897e+06	7.016946e+06	6.612135e+06	5.985455e+06	1.16776
75%	204.250000	4.851512e+07	4.426034e+07	8.538661e+06	7.701300e+06	7.257008e+06	6.569209e+06	1.28165
max	272.000000	5.489216e+07	5.007811e+07	9.661020e+06	8.713592e+06	8.210901e+06	7.432694e+06	1.45011

8 rows × 21 columns

In [5]:
```
store_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 272 entries, 0 to 271
Data columns (total 23 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Store Numbers           272 non-null    int64  
 1   City                    272 non-null    object 
 2   State                   272 non-null    object 
 3    2022 Gross Revenue     272 non-null    int64  
 4    2021 Gross Revenue     272 non-null    float64
 5    2022 Gross Profit      272 non-null    float64
 6    2021 Gross Profit      272 non-null    float64
 7    2022 Expenses          272 non-null    float64
 8    2021 Expenses          272 non-null    float64
 9    2022 Net Income        272 non-null    float64
 10   2021 Net Income        272 non-null    float64
 11  Total Members           272 non-null    int64  
 12  Avg_Mbr_Length          272 non-null    float64
 13  Premium Member          272 non-null    int64  
 14  Budget Member           272 non-null    int64  
 15  Mainstream Member       272 non-null    int64  
 16  RETIREES                272 non-null    int64  
 17  OLDER SINGLES/COUPLES   272 non-null    int64  
 18  OLDER FAMILIES          272 non-null    int64  
 19  YOUNG FAMILIES          272 non-null    int64  
 20  YOUNG SINGLES/COUPLES   272 non-null    int64  
 21  MIDAGE FAMILIES         272 non-null    int64  
 22  MIDAGE SINGLES/COUPLES  272 non-null    int64  
dtypes: float64(8), int64(13), object(2)
memory usage: 49.0+ KB
```

In [6]:
```
store_data
```

Out[6]:

	Store Numbers	City	State	2022 Gross Revenue	2021 Gross Revenue	2022 Gross Profit	2021 Gross Profit	2022 Expenses	2021 Expenses	202 In
0	1	Miami	Florida	49361095	45032126.97	8687552.72	7835590.09	7383551.06	6683758.35	13040
1	2	Port St. Lucie	Florida	32868396	29985837.67	5784837.70	5217535.75	4916533.56	4450558.00	8683
2	3	Port St. Lucie	Florida	42454094	38730869.96	7471920.54	6739171.37	6350385.27	5748513.18	11215
3	4	Miami	Florida	48098343	43880118.32	8465308.37	7635140.59	7194665.58	6512774.92	12706
4	5	Port St. Lucie	Florida	44277169	40394061.28	7792781.74	7028566.66	6623085.20	5995367.36	11696
...	...	...	...	...	...	...	...	...	...	
267	268	Oyster Bay	New York	30389086	27723963.16	5348479.14	4823969.59	4545672.42	4114846.06	8028

268	269	New York City	New York	41493487	37854508.19	7302853.71	6586684.43	6206695.37	5618441.81	10961
269	270	Brookhaven	New York	42168955	38470737.65	7421736.08	6693908.35	6307733.49	5709903.82	11140
270	271	Brookhaven	New York	39609983	36136187.49	6971357.01	6287696.62	5924956.32	5363405.22	10464
271	272	New York City	New York	44932780	40992175.19	7908169.28	7132638.48	6721153.07	6084140.63	11870

272 rows × 23 columns

In [7]:
```
#cleaning column names

column_name_mapping = {'Store Numbers': 'Store_Numbers',' 2022 Gross Revenue ': '2022_Gr
       ' 2021 Gross Revenue ': '2021_Gross_Revenue', ' 2022 Gross Profit ': '2022_Gross_
       ' 2022 Expenses ': '2022_Expenses', ' 2021 Expenses ': '2021_Expenses', ' 2022 Ne
       ' 2021 Net Income ': '2021_Net_Income', 'Total Members': 'Total_Members',
       'Premium Member': 'Premium_Member', 'Budget Member': 'Budget_Member', 'Mainstream
       'RETIREES': 'Retirees', 'OLDER SINGLES/COUPLES': 'Older_Single/Couples', 'OLDER F
       'YOUNG FAMILIES': 'Young_Families', 'YOUNG SINGLES/COUPLES': 'Young_Single/Couple
       'MIDAGE SINGLES/COUPLES': 'Midage_Singles/Couples'}

store_data.rename(columns=column_name_mapping, inplace=True)
```

In [8]:
```
#grouping store data by states
store_data_states = store_data.groupby("State")

#viewing number of stores in each state
store_data_states['Store_Numbers'].count()
```

Out[8]:
```
State
Florida           30
Georgia           30
Maryland          30
New Jersey        30
New York          32
North Carolina    30
Pennsylvania      30
South Carolina    30
Virginia          30
Name: Store_Numbers, dtype: int64
```

In [9]:
```
#number of stores in each city and state
store_data_states["City"].value_counts()
```

Out[9]:
```
State           City
Florida         Miami                7
                Orlando              6
                Port St. Lucie       6
                Jacksonville         4
                St. Petersburg       4
                Tampa                3
Georgia         Savannah             8
                Atlanta              7
                Columbus             7
                Augusta              6
                Athens               2
Maryland        Frederick           10
                Silver Spring        7
                Germantown           5
                Waldorf              4
                Baltimore            2
```

```
                  Columbia              2
New Jersey        Elizabeth             9
                  Newark                7
                  Jersey City           5
                  Lakewood              4
                  Paterson              3
                  Edison                2
New York          New York City        10
                  Brookhaven            6
                  Buffalo               5
                  Oyster Bay            5
                  Hempstead             3
                  Islip                 3
North Carolina    Raleigh               7
                  Greensboro            6
                  Durham                5
                  Charlotte             4
                  Fayetteville          4
                  Winston-Salem         4
Pennsylvania      Reading              11
                  Philadelphia          6
                  Erie                  4
                  Allentown             3
                  Pittsburgh            3
                  Upper Darby           3
South Carolina    North Charleston      8
                  Columbia              5
                  Mount Pleasant        5
                  Charleston            4
                  Greenville            4
                  Rock Hill             4
Virginia          Arlington             8
                  Richmond              6
                  Virginia Beach        6
                  Chesapeake            5
                  Norfolk               5
Name: City, dtype: int64
```

In [10]:

```python
#looking at gross revenue for previous year
store_data_states["2022_Gross_Revenue"].sum().round().sort_values(ascending=False)
```

Out[10]:

```
State
Georgia           1397109994
New York          1362754835
New Jersey        1361378117
South Carolina    1296937859
Maryland          1281107230
Virginia          1280417154
Florida           1278295926
North Carolina    1249760389
Pennsylvania      1248203454
Name: 2022_Gross_Revenue, dtype: int64
```

In [11]:

```python
#looking at gross revenue for previous year
store_data_states["2021_Gross_Revenue"].sum().round().sort_values(ascending=False)
```

Out[11]:

```
State
Georgia           1.274583e+09
New York          1.243241e+09
New Jersey        1.241985e+09
South Carolina    1.183196e+09
Maryland          1.168754e+09
Virginia          1.168125e+09
Florida           1.166189e+09
North Carolina    1.140156e+09
```

```
Pennsylvania      1.138736e+09
Name: 2021_Gross_Revenue, dtype: float64
```

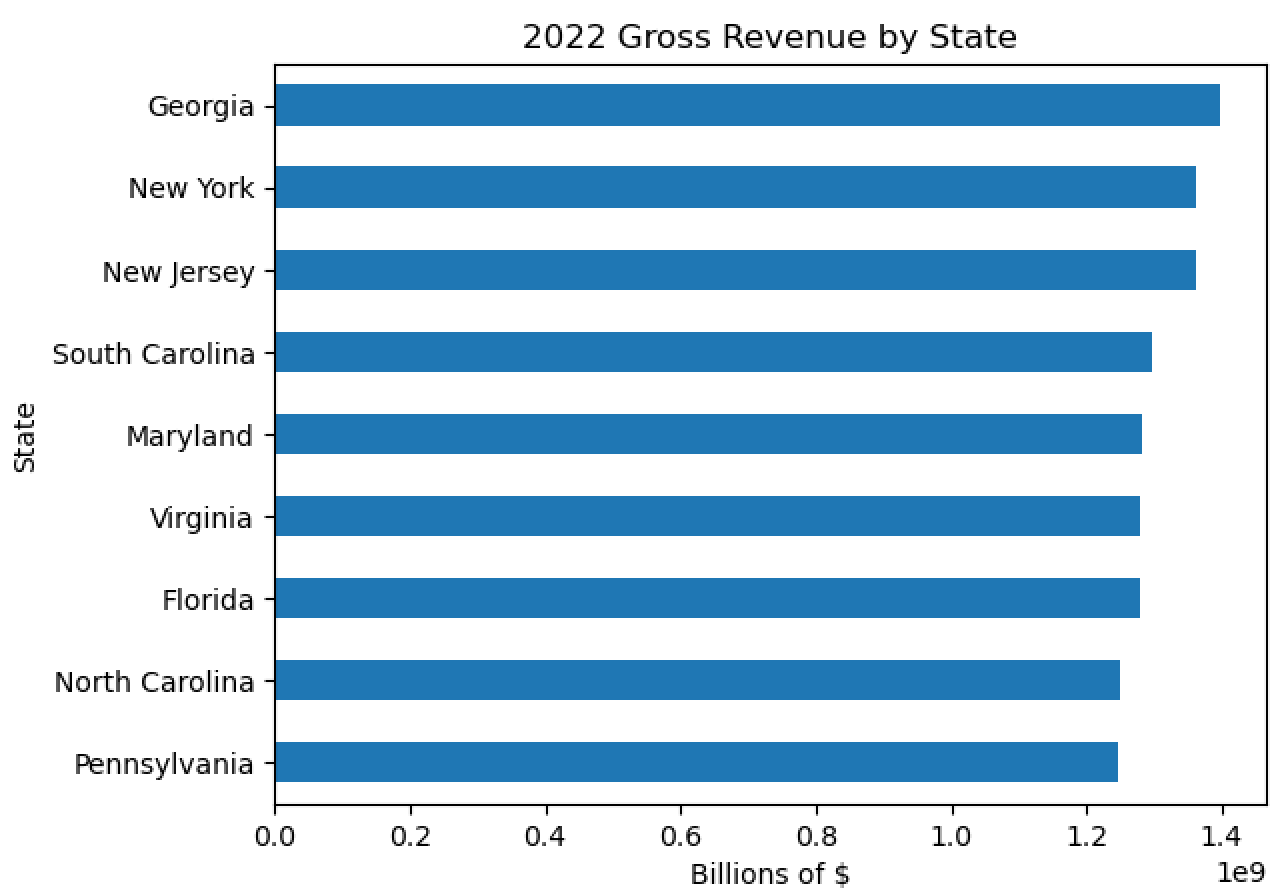
In [12]:
```
#looking at net income for current year
store_data_states["2022_Net_Income"].sum().round().sort_values(ascending=False)
```

Out[12]:
```
State
Georgia           36908293.0
New York          36000712.0
New Jersey        35964343.0
South Carolina    34261986.0
Maryland          33843778.0
Virginia          33825548.0
Florida           33769510.0
North Carolina    33015670.0
Pennsylvania      32974540.0
Name: 2022_Net_Income, dtype: float64
```

In [13]:
```
#looking at net income for previous  year
store_data_states["2021_Net_Income"].sum().round().sort_values(ascending=False)
```

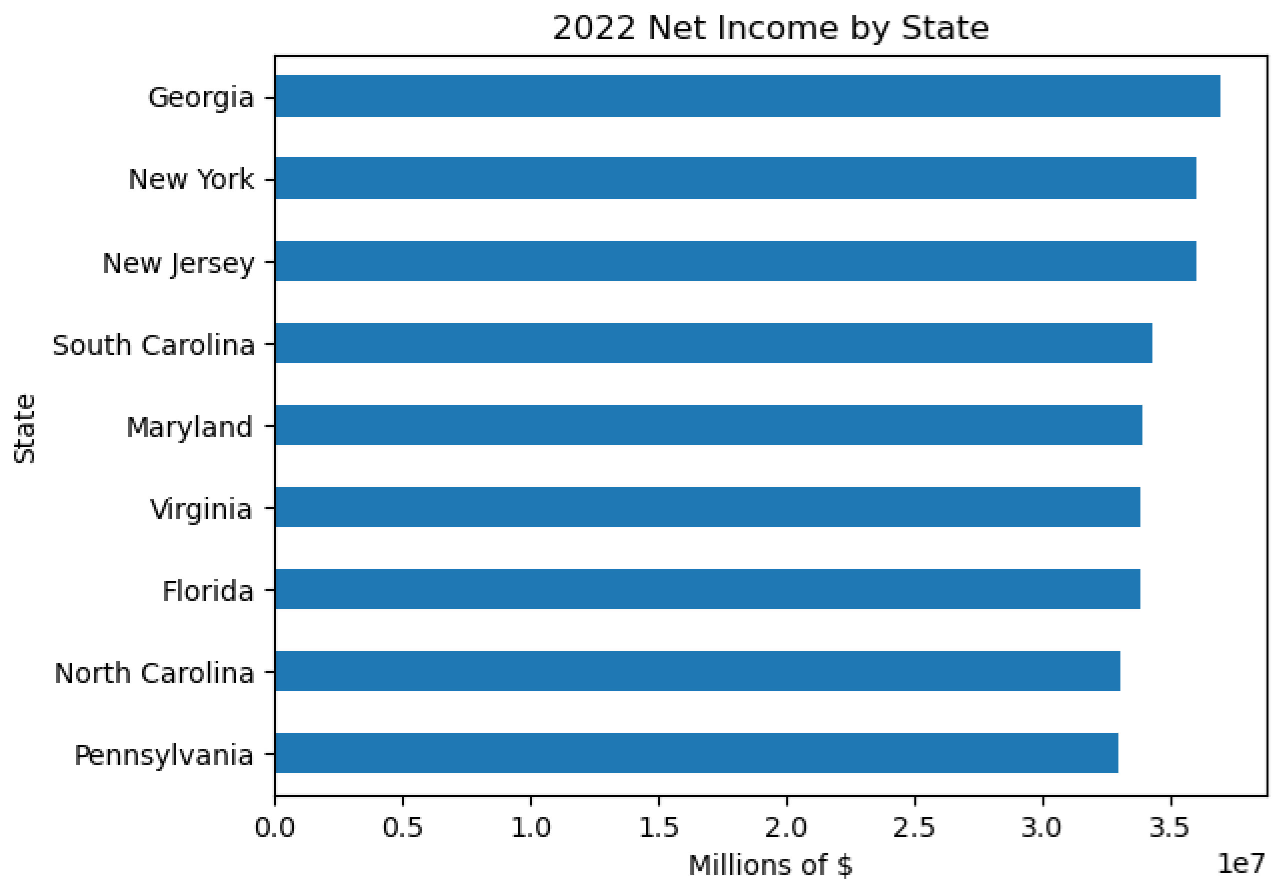
Out[13]:
```
State
Georgia           32601295.0
New York          31799624.0
New Jersey        31767499.0
South Carolina    30263798.0
Maryland          29894393.0
Virginia          29878290.0
Florida           29828792.0
North Carolina    29162920.0
Pennsylvania      29126590.0
Name: 2021_Net_Income, dtype: float64
```

In [14]:
```
store_data_states["2022_Gross_Revenue"].sum().sort_values(ascending=True).plot(kind='bar
plt.xlabel('Billions of $')
plt.title("2022 Gross Revenue by State")
plt.show()
```



In [15]:

```
store_data_states["2022_Net_Income"].sum().sort_values(ascending=True).plot(kind='barh')
plt.xlabel('Millions of $')
plt.title("2022 Net Income by State")
plt.show()
```



Notes:

Based on gross revenue and net income for 2022 and 2021 the states that performed the best were Georgia, New York, and New Jersey.

Lets explore these states further to see what the receipe for the top performing sales are.

Top 3 States:

In [16]:

```
#extracting the 3 states
sd_ga = store_data[store_data["State"] == "Georgia"]
sd_ny = store_data[store_data["State"] == "New York"]
sd_nj = store_data[store_data["State"] == "New Jersey"]

#merging 3 states into a new dataframe
top3_states = pd.concat([sd_ga, sd_ny, sd_nj], ignore_index=True)

top3_states
```

Out[16]:

	Store_Numbers	City	State	2022_Gross_Revenue	2021_Gross_Revenue	2022_Gross_Profit	2021_Gross_I
0	31	Augusta	Georgia	37285840	34015871.83	6562307.84	59187
1	32	Atlanta	Georgia	52975509	48329556.86	9323689.58	84093
2	33	Atlanta	Georgia	54892157	50078114.83	9661019.63	87135
3	34	Augusta	Georgia	42679190	38936225.04	7511537.44	67749

4	35	Columbus	Georgia	54574630	49788434.95	9605134.88	86631
...	...	...	...	...	...	...	
87	236	Elizabeth	New Jersey	54331407	49566542.61	9562327.63	86245
88	237	Newark	New Jersey	53844329	49122181.35	9476601.90	85472
89	238	Elizabeth	New Jersey	44377034	40485168.12	7810357.98	70444
90	239	Elizabeth	New Jersey	48692162	44421859.39	8569820.51	77294
91	240	Newark	New Jersey	44058586	40194648.01	7754311.14	69938

92 rows × 23 columns

In [17]:

```
#Top 10 stores by net income
top3_states[['Store_Numbers','City','State','2022_Net_Income']].sort_values(by="2022_Net
```

Out[17]:

	Store_Numbers	City	State	2022_Net_Income
2	33	Atlanta	Georgia	1450119.05
36	247	New York City	New York	1447710.77
55	266	New York City	New York	1446446.66
37	248	Hempstead	New York	1446055.15
4	35	Columbus	Georgia	1441730.75
87	236	Elizabeth	New Jersey	1435305.38
29	60	Atlanta	Georgia	1430768.55
27	58	Savannah	Georgia	1427432.70
88	237	Newark	New Jersey	1422437.95
85	234	Newark	New Jersey	1422159.29

In [18]:

```
#Top 10 stores by Gross Revenue

top3_states[['Store_Numbers','City','State','2022_Gross_Revenue']].sort_values(by="2022_
```

Out[18]:

	Store_Numbers	City	State	2022_Gross_Revenue
2	33	Atlanta	Georgia	54892157
36	247	New York City	New York	54800995
55	266	New York City	New York	54753144
37	248	Hempstead	New York	54738324
4	35	Columbus	Georgia	54574630
87	236	Elizabeth	New Jersey	54331407
29	60	Atlanta	Georgia	54159672
27	58	Savannah	Georgia	54033398
88	237	Newark	New Jersey	53844329

85	234	Newark	New Jersey	53833781

Notes:

Based on the net income and gross revenue the top 10 stores listed all seem to be in heavily populated areas in each state.

> Savannah and Atlanta are both highly populated in Georgia. Georgia is interesting what stands out is there are 3 different cities on the top 10.
>
> > New York has 2 cities on the list while New Jersey has 1.
> >
> > > New York City and Elizabeth New Jersey are both highly populated areas. Elizabeth is also very close to New York City geographically. Hempstead is located in Long Island New York which is outside of New York City. Even so it still remains close to New York City and is heavily populated.

Revenue and Income by City:

In [19]:
```
#top 3 states 2022 Revenue grouped by city

city_grouped = top3_states.groupby('City')

city_grouped['2022_Gross_Revenue'].sum(numeric_only=True).sort_values(ascending=False)
```

Out[19]:
```
City
New York City    479298554
Elizabeth        419860465
Savannah         392181871
Columbus         354990777
Atlanta          351240328
Newark           344338566
Buffalo          256613955
Jersey City      235064050
Augusta          230510110
Brookhaven       213732419
Oyster Bay       165801269
Lakewood         154480271
Hempstead        141117837
Paterson         116953383
Islip            106190801
Edison            90681382
Athens            68186908
Name: 2022_Gross_Revenue, dtype: int64
```

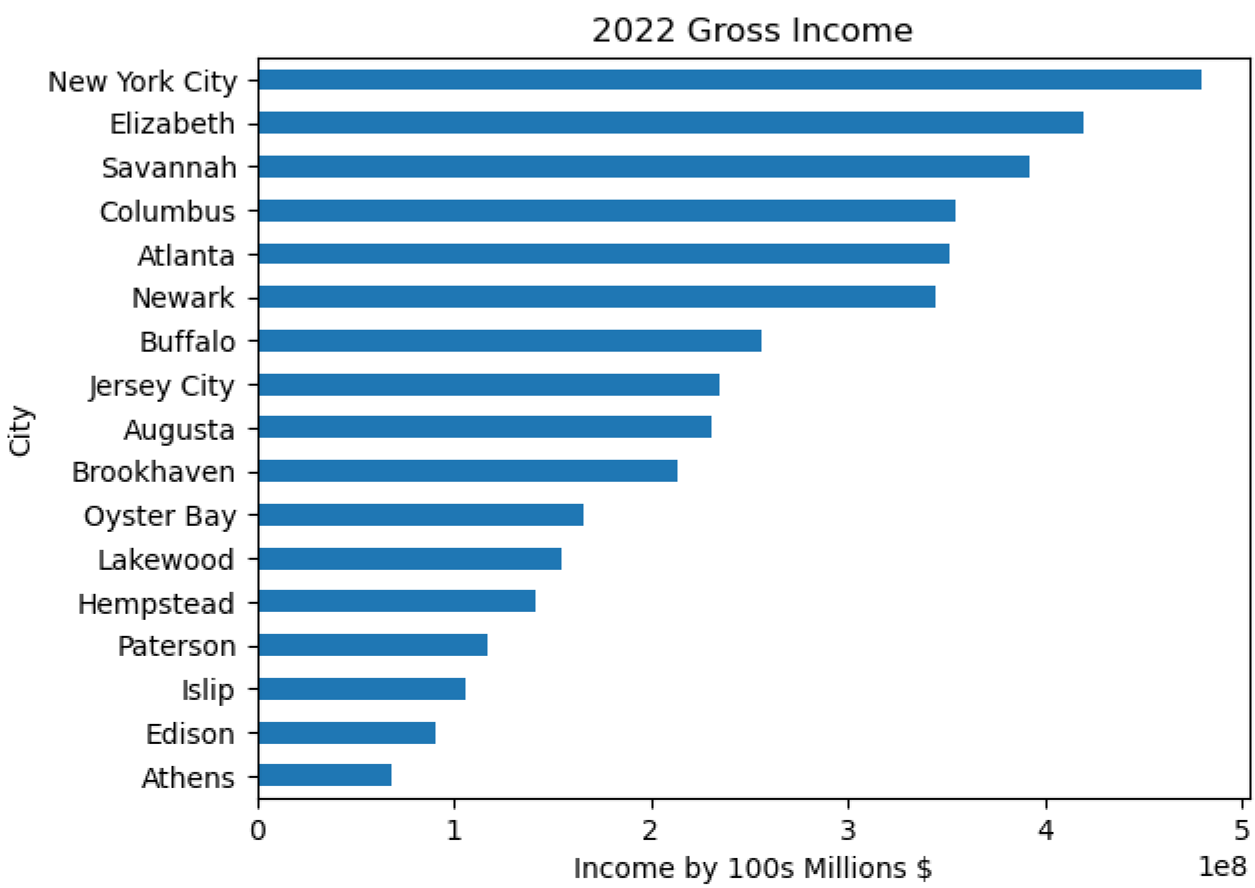
In [20]:
```
#2022 Net Income grouped by city

city_grouped = top3_states.groupby('City')

city_grouped['2022_Net_Income'].sum(numeric_only=True).sort_values(ascending=False)
```

Out[20]:
```
City
New York City    12661917.49
Elizabeth        11091705.81
Savannah         10360503.80
Columbus          9378004.36
Atlanta           9278926.50
Newark            9096598.50
Buffalo           6779124.82
```

```
Jersey City        6209828.05
Augusta            6089523.88
Brookhaven         5646297.56
Oyster Bay         4380071.62
Lakewood           4080998.01
Hempstead          3727994.57
Paterson           3089627.69
Islip              2805306.11
Edison             2395584.47
Athens             1801334.46
Name: 2022_Net_Income, dtype: float64
```

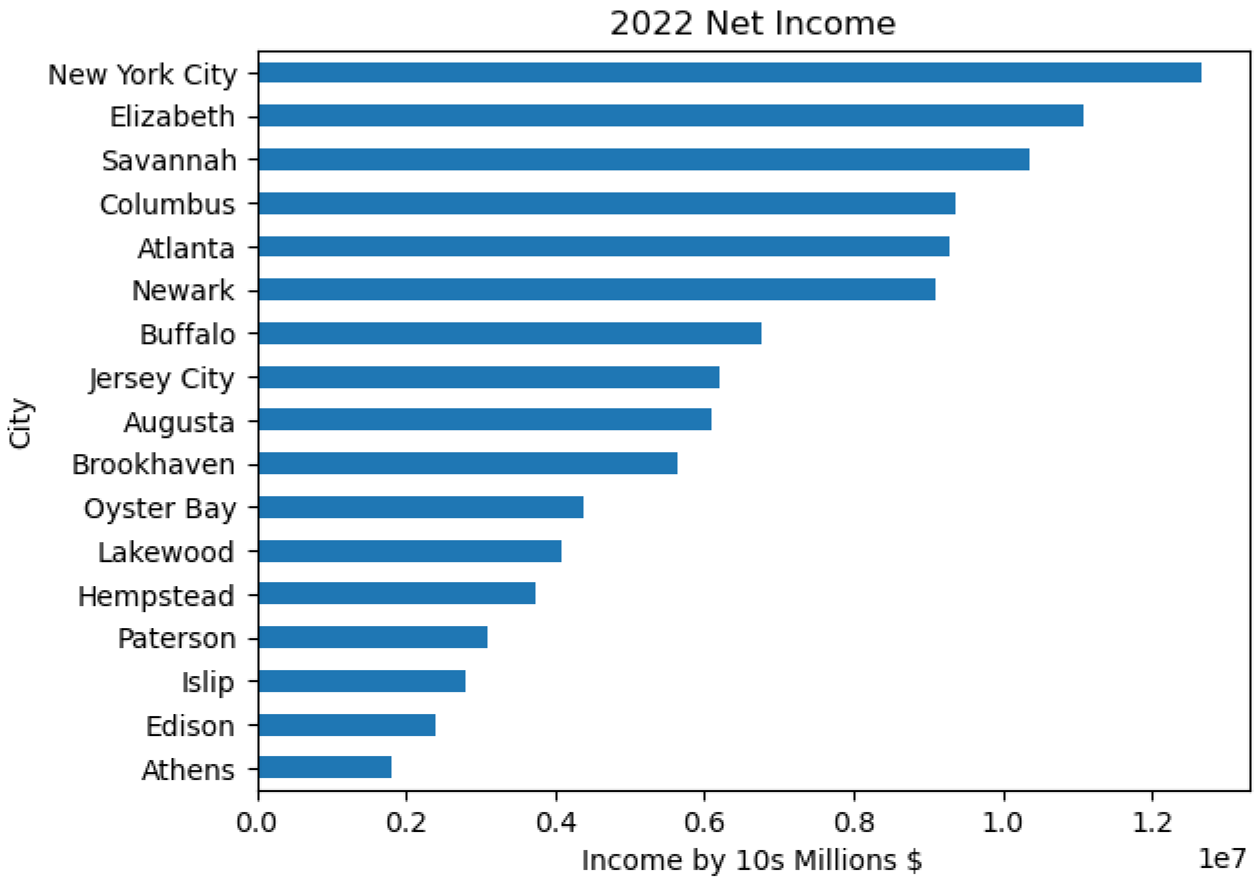
In [21]:

```
city_grouped['2022_Gross_Revenue'].sum(numeric_only=True).sort_values(ascending=True).pl
plt.xlabel('Income by 100s Millions $')
plt.title("2022 Gross Income")
plt.show()
```



In [22]:

```
city_grouped['2022_Net_Income'].sum(numeric_only=True).sort_values(ascending=True).plot(
plt.xlabel('Income by 10s Millions $')
plt.title("2022 Net Income")
plt.show()
```



Notes:

As I can see above the top 3 cities match the top 10 store list except for Hempstead NY. Hempstead NY Comes in 12th place by city. The hempstead store is 1 of 3 stores in the city. Every other city has more stores in the region or are generally more heavily populated that is why Hempstead has fallen from the store performance vs city performance.

I would not count out the hempstead store as an outlier as it may provide interesting insights on the customer segmentation.

Exploring the Average Member Length column:

In [23]:
```
store_data["Avg_Mbr_Length"].describe()
```

Out[23]:
```
count    272.000000
mean       7.231507
std        0.671246
min        4.000000
25%        7.060000
50%        7.220000
75%        7.320000
max       13.000000
Name: Avg_Mbr_Length, dtype: float64
```

In [24]:
```
#top 10 stores by average member length in years
top3_states[['Store_Numbers','City','State','Avg_Mbr_Length']].sort_values(by='Avg_Mbr_L
```

Out[24]:

	Store_Numbers	City	State	Avg_Mbr_Length
0	31	Augusta	Georgia	10.00
41	252	New York City	New York	9.50

32	243	New York City	New York	7.66
69	218	Jersey City	New Jersey	7.55
8	39	Athens	Georgia	7.55
62	211	Newark	New Jersey	7.50
87	236	Elizabeth	New Jersey	7.43
68	217	Newark	New Jersey	7.41
9	40	Atlanta	Georgia	7.40
14	45	Savannah	Georgia	7.40

In [25]:

```
#exploring to see if average member length at the wholesale club is statistically signif
#will use the pearson correlation test with a significance level of .05 or 5%
from scipy.stats import pearsonr

correlation, p_value = pearsonr(store_data["Avg_Mbr_Length"], store_data["2022_Net_Incom

print("Correlation:", correlation)
print("P Value:", p_value)
```

```
Correlation: -0.08008570968516467
P Value: 0.18789203620533126
```

In [26]:

```
#splitting dataframe into 2 for a t-test of independence

store_data_2 = store_data.sort_values(by="2022_Net_Income")

#splitting by net income down the middle top 136 vs bottom 136
low_income = store_data_2.iloc[0:135]
high_income = store_data_2.iloc[135:271]
```

In [27]:

```
from scipy.stats import ttest_ind
#going to use the Welchs t-test with a significance level of .05 or 5%
t_stat, p_value = ttest_ind(low_income['Avg_Mbr_Length'], high_income['Avg_Mbr_Length'],

print('T-statistic:', t_stat)
print('P-value:', p_value)
```

```
T-statistic: 1.1217521554033845
P-value: 0.26376200665532323
```

Notes:

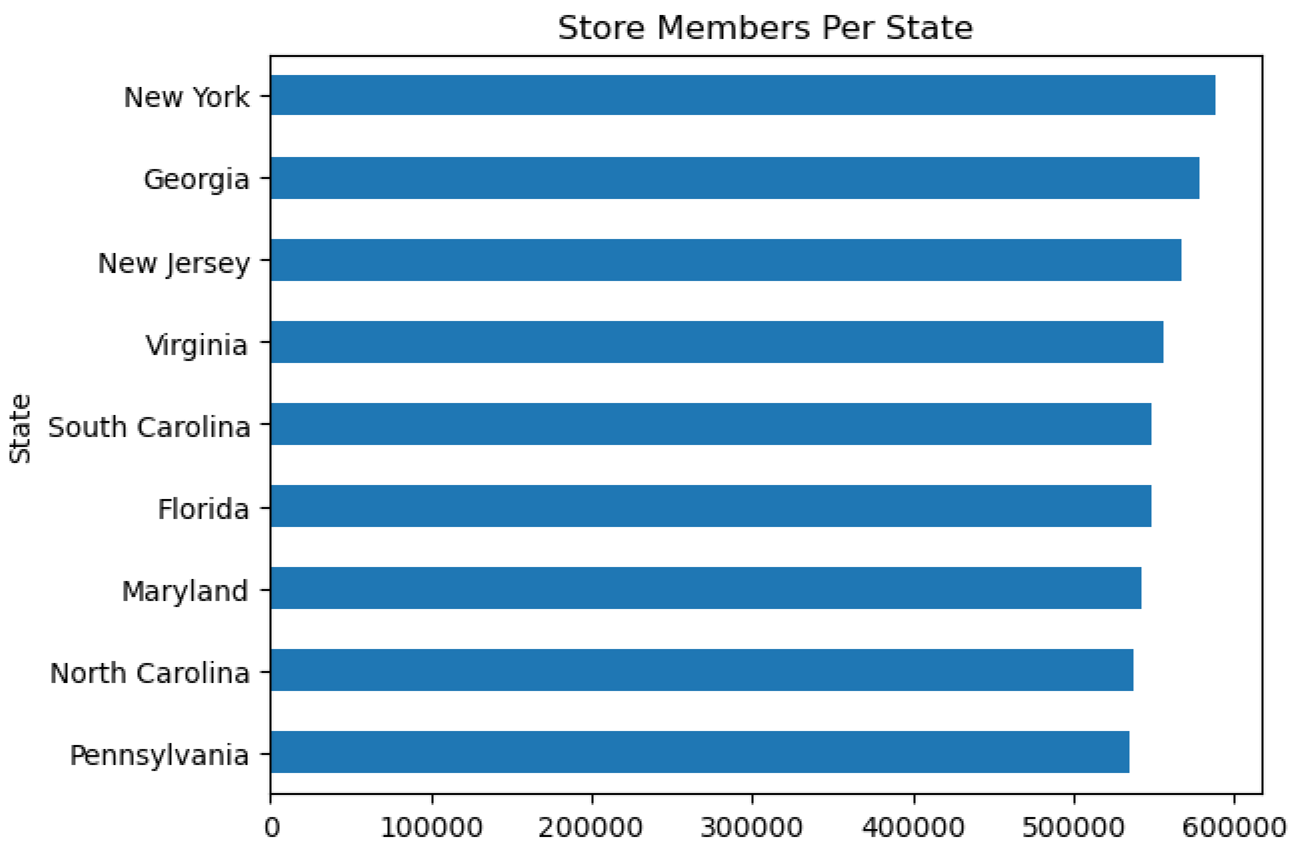
Based on the results of the correlation test and the t-test I do not have enough evidence to reject the null hypothesis. The null hypothesis is that there is no significant difference between the net income and the average length of membership. This is not to say that no significant relationship exists between membership length and income but without more detailed breakdown of information I am unable to conclude the impact of membership length has on a stores performance.

To Recap so far:

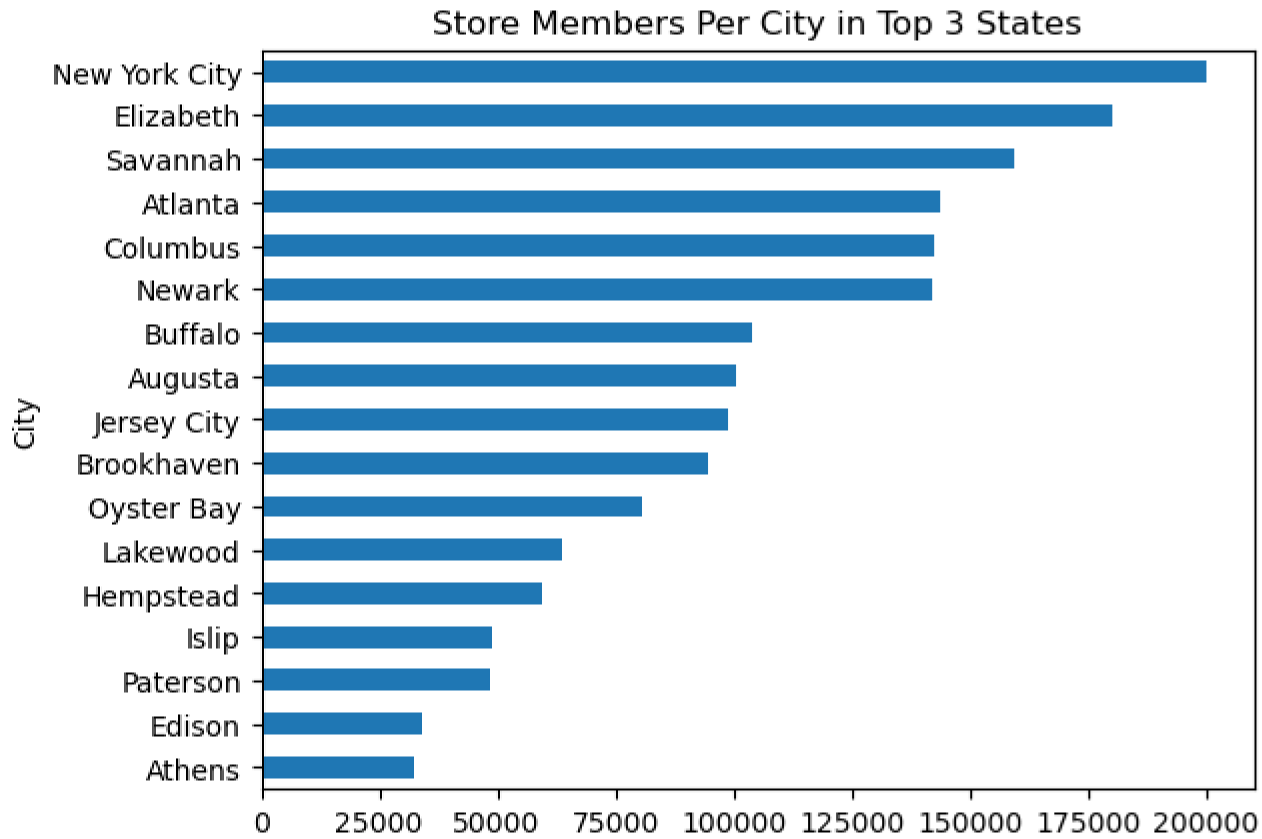
- I have explored the data by state, city, store, looking at net income and gross revenue.
- Determined the top 3 States are Georgia, New York, and New Jersey
- Top 3 cities are Savannah GA, NYC NY, and Elizabeth, NJ
- Performed 2 statistical tests on the Average Membership Length and net income.

I will continue now to explore the demographical columns starting with the membership tiers and membership population.

In [28]:
```
#viewing membership population by state

store_data_states['Total_Members'].sum().sort_values(ascending=True).plot(kind='barh')
plt.title("Store Members Per State")
plt.show()
```



In [29]:
```
#viewing membership population by city in the top 3 states

city_grouped['Total_Members'].sum().sort_values(ascending=True).plot(kind='barh')
plt.title("Store Members Per City in Top 3 States")
plt.show()
```



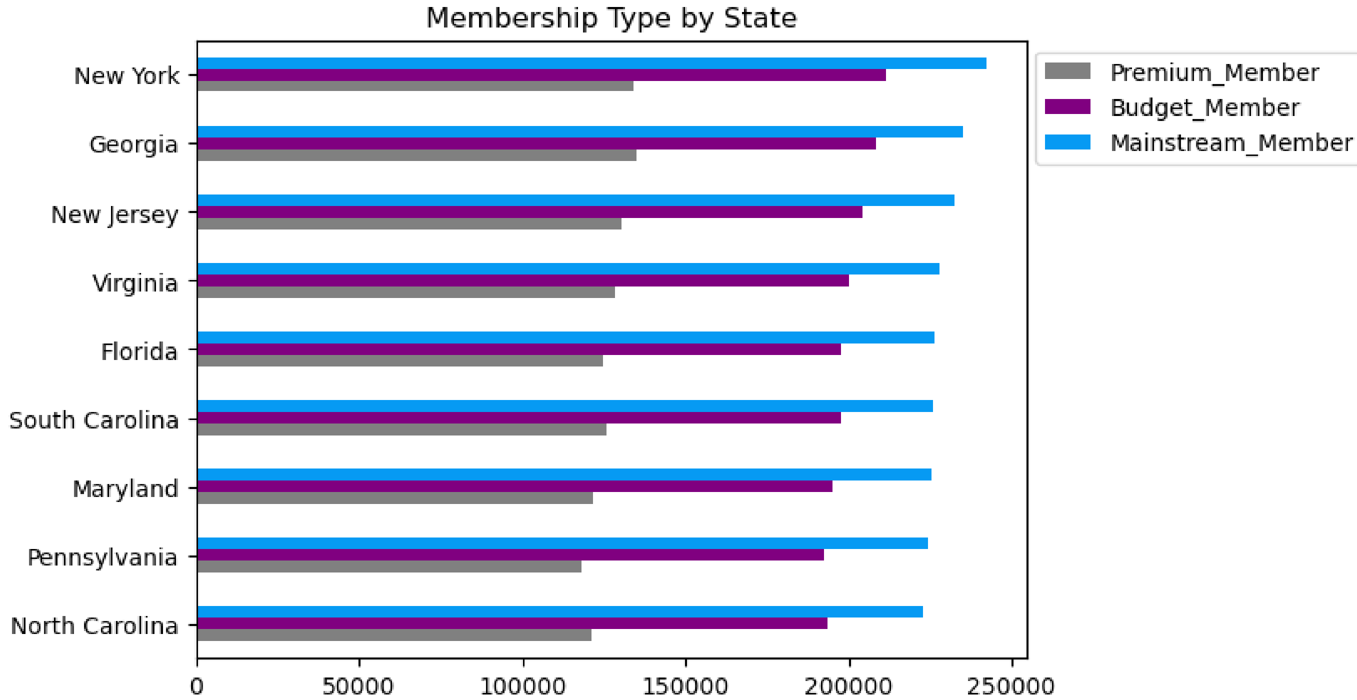
Notes:

As I can see the cities that are performing the highest in regards to income also have the most membership. This would also correlate with city sizes.

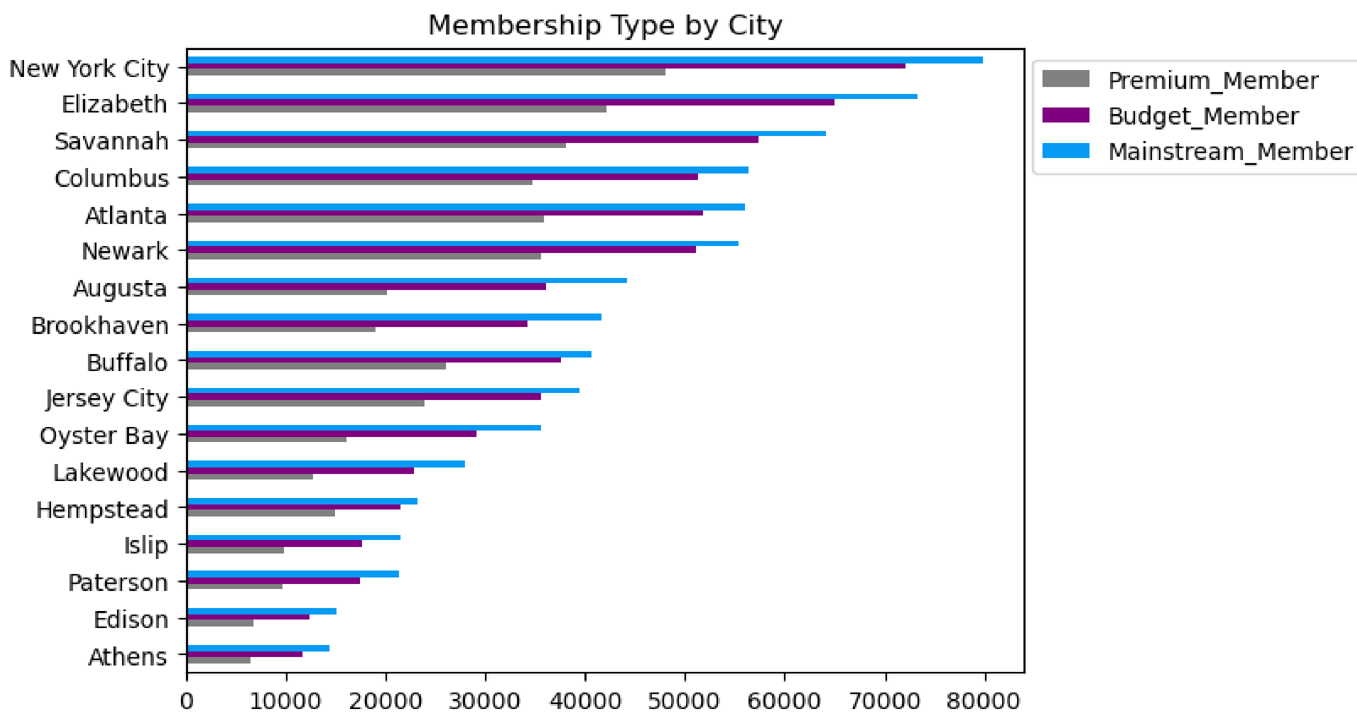
Exploring the membership types:

In [30]:

```
colors = ['#808080', '#800080', '#069AF3']
store_data_states[["Premium_Member","Budget_Member","Mainstream_Member"]].sum().sort_val
plt.legend(loc="upper right", bbox_to_anchor=(1.42,1))
plt.title("Membership Type by State")
plt.ylabel("")
plt.show()
```



In [31]:
```
colors = ['#808080', '#800080', '#069AF3']
city_grouped[["Premium_Member","Budget_Member","Mainstream_Member"]].sum().sort_values(b
plt.legend(loc="upper right", bbox_to_anchor=(1.42,1))
plt.title("Membership Type by City")
plt.ylabel("")
plt.show()
```



In [32]:
```
#percentage of premium membership in each state
premium_members = store_data_states["Premium_Member"].sum() / store_data_states["Total_M

premium_members.sort_values(ascending= False)
```

Out[32]:
```
State
Georgia           0.233689
Virginia          0.230516
New Jersey        0.230314
South Carolina    0.228796
New York          0.227680
```

```
Florida           0.227342
North Carolina    0.225129
Maryland          0.224311
Pennsylvania      0.220627
dtype: float64
```

In [33]:
```python
#percentage of budget members in each state
budget_members = store_data_states["Budget_Member"].sum() / store_data_states["Total_Mem

budget_members.sort_values(ascending= False)
```

Out[33]:
```
State
Florida           0.360004
South Carolina    0.360002
North Carolina    0.360001
New Jersey        0.360000
New York          0.359998
Virginia          0.359998
Pennsylvania      0.359996
Maryland          0.359996
Georgia           0.359994
dtype: float64
```

In [34]:
```python
#percentage of mainstream in each state
mainstream_members = store_data_states["Mainstream_Member"].sum() / store_data_states["T

mainstream_members.sort_values(ascending= False)
```

Out[34]:
```
State
Pennsylvania      0.419382
Maryland          0.415691
North Carolina    0.414874
Florida           0.412660
New York          0.412332
South Carolina    0.411208
New Jersey        0.409698
Virginia          0.409489
Georgia           0.406318
dtype: float64
```

State	Premium	Budget	Mainstream	P Rank	B Rank	M Rank
Georgia	23.4%	35.9%	40.6%	1st	9th	9th
New York	22.7%	36%	41.2%	5th	5th	5th
New Jersey	23.0%	36%	40.9%	3rd	4th	7th
Average	23.0%	36%	40.9%			

As we can see Georgia has the most premium member share amongst its total memberships. New York is balanced and New Jersey ranks higher on Premium and Budget members than NY.

Based on percentages it doesnt look like major differences between member types but its probably safe to conclude that it does have an impact. Compared to the other states the premium membership and mainstream members might be giving these states the push they need. Espcially if considering the population size of each state; Georgia is performing very well against larger populated states that have similar or more amounts of members.

Exploring Membership type on a city level:

In [35]:
```python
#percentage of premium in by city
premium_members_city = city_grouped["Premium_Member"].sum() / city_grouped["Total_Member
```

```
premium_members_city.sort_values(ascending= False)
```

Out[35]:
```
City
Hempstead        0.250013
Newark           0.250009
Atlanta          0.250009
Buffalo          0.249995
Columbus         0.243658
Jersey City      0.240663
New York City    0.240424
Savannah         0.238155
Elizabeth        0.233664
Islip            0.200025
Edison           0.200023
Brookhaven       0.200011
Paterson         0.200004
Augusta          0.200004
Oyster Bay       0.200002
Lakewood         0.200000
Athens           0.200000
dtype: float64
```

In [36]:
```
#percentage of budget in each city
budget_members_city = city_grouped["Budget_Member"].sum() / city_grouped["Total_Members"

budget_members_city.sort_values(ascending= False)
```

Out[36]:
```
City
Hempstead        0.360019
Paterson         0.360012
Brookhaven       0.360006
Newark           0.360004
Jersey City      0.360003
Edison           0.360001
Elizabeth        0.359999
Atlanta          0.359999
Buffalo          0.359998
Savannah         0.359995
New York City    0.359993
Oyster Bay       0.359992
Columbus         0.359991
Augusta          0.359991
Athens           0.359988
Islip            0.359987
Lakewood         0.359984
dtype: float64
```

In [37]:
```
#percentage of mainstream in each city
mainstream_members_city = city_grouped["Mainstream_Member"].sum() / city_grouped["Total_

mainstream_members_city.sort_values(ascending= False)
```

Out[37]:
```
City
Paterson         0.440026
Brookhaven       0.440015
Athens           0.440012
Islip            0.440009
Oyster Bay       0.440005
Augusta          0.440005
Edison           0.440005
Lakewood         0.440000
Elizabeth        0.406342
Savannah         0.401851
New York City    0.399583
Jersey City      0.399345
```

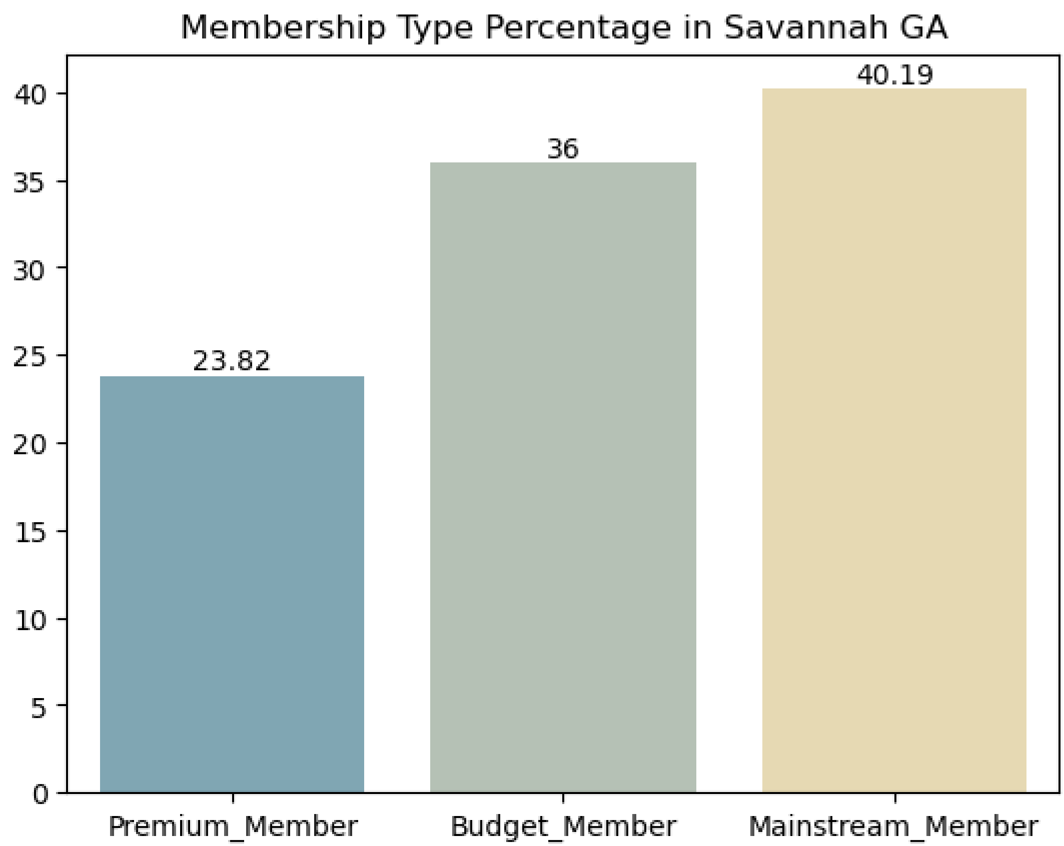
```
Columbus          0.396351
Newark            0.390009
Buffalo           0.390007
Hempstead         0.390002
Atlanta           0.389999
dtype: float64
```

In [38]:

```
#extracting top 3 cities
savannah = store_data[store_data["City"] == "Savannah"]
ny_ny = store_data[store_data["City"] == "New York City"]
elizabeth = store_data[store_data["City"] == "Elizabeth"]
```

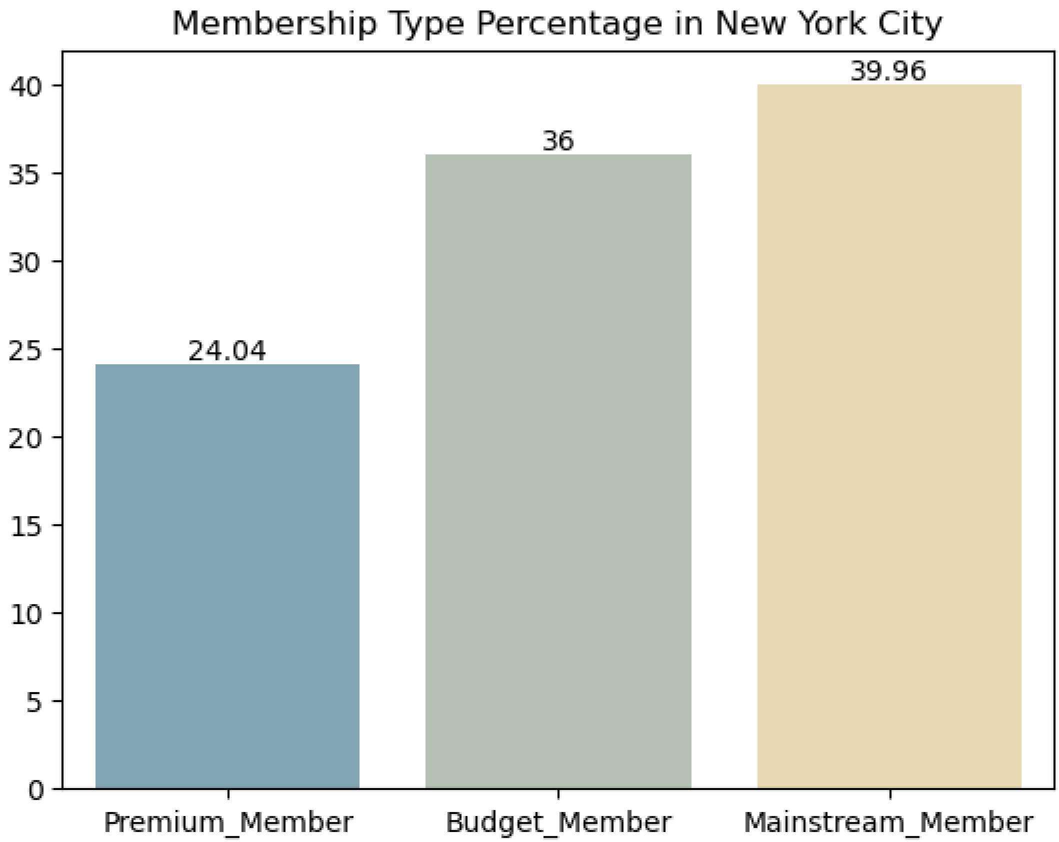
In [39]:

```
#cacluating percentages for membership type in Savannah GA
s_g = round(savannah[["Premium_Member","Budget_Member","Mainstream_Member" ]].sum() / sa

s_gplot = sns.barplot(x=s_g.index, y=s_g.values, palette = "blend:#7AB,#EDA")
plt.bar_label(s_gplot.containers[0])
plt.title("Membership Type Percentage in Savannah GA")
plt.show()
```

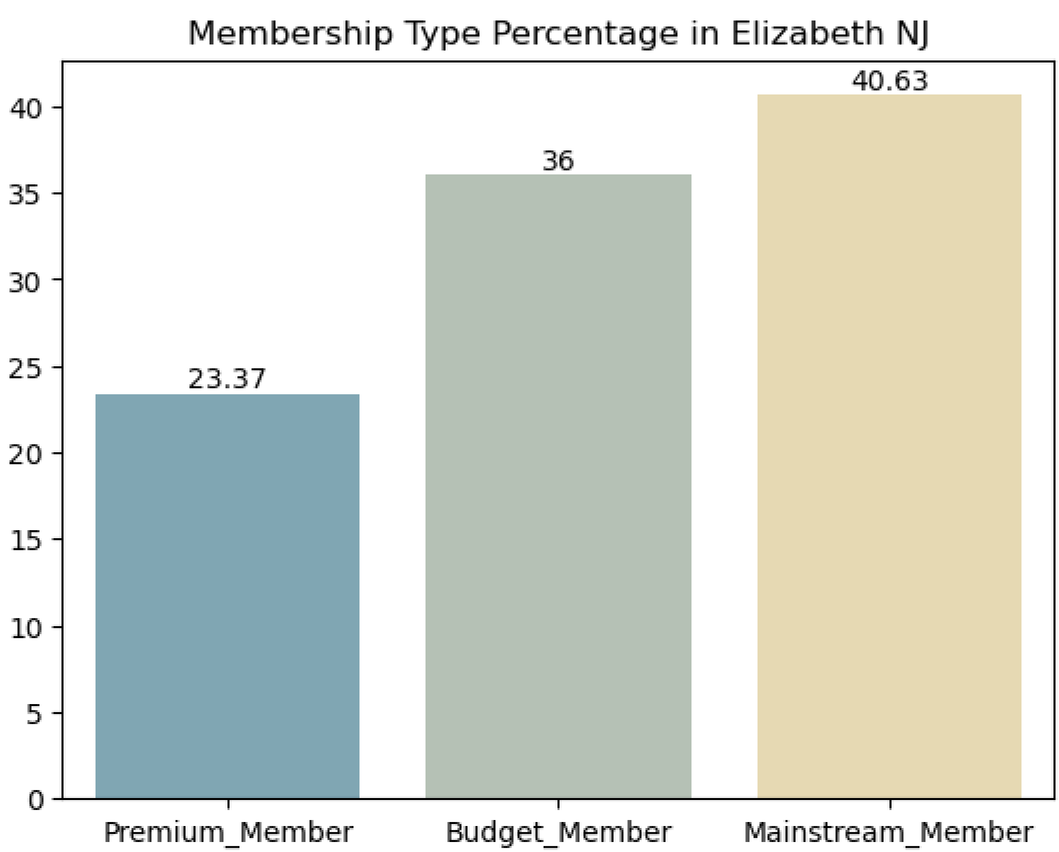


In [40]:

```
#cacluating percentages for membership type in New York City
ny_g = round(ny_ny[["Premium_Member","Budget_Member","Mainstream_Member" ]].sum() / ny_n

s_gplot = sns.barplot(x=ny_g.index, y=ny_g.values, palette = "blend:#7AB,#EDA")
plt.bar_label(s_gplot.containers[0])
plt.title("Membership Type Percentage in New York City")
plt.show()
```



In [41]:
```
#cacluating percentages for membership type in Elizabeth NJ

e_g = round(elizabeth[["Premium_Member","Budget_Member","Mainstream_Member" ]].sum() / e

s_gplot = sns.barplot(x=e_g.index, y=e_g.values, palette = "blend:#7AB,#EDA")
plt.bar_label(s_gplot.containers[0])
plt.title("Membership Type Percentage in Elizabeth NJ")
plt.show()
```



Notes

Looking at the city level it seems Mainstream is the dominate group type followed by Budget and than Premium. The higher performing cities with sales seem to have a more balanced membership leaning towards Mainstream the most. The interesting obversation is that the 3 top performing cities have more Premium and Budget members. It seems a balanced leaning membership type might yield the best results.

Exploring Age and Family demographics:

In [42]:
```
store_data.columns
```

Out[42]:
```
Index(['Store_Numbers', 'City', 'State', '2022_Gross_Revenue',
       '2021_Gross_Revenue', '2022_Gross_Profit', '2021_Gross_Profit',
       '2022_Expenses', '2021_Expenses', '2022_Net_Income', '2021_Net_Income',
       'Total_Members', 'Avg_Mbr_Length', 'Premium_Member', 'Budget_Member',
       'Mainstream_Member', 'Retirees', 'Older_Single/Couples',
       'Older_Families', 'Young_Families', 'Young_Single/Couples',
       'Midage_Families', 'Midage_Singles/Couples'],
      dtype='object')
```

In [43]:
```
#sum of each demographic category in each state
store_data_states[['Retirees', 'Older_Single/Couples',
       'Older_Families', 'Young_Families', 'Young_Single/Couples',
       'Midage_Families', 'Midage_Singles/Couples']].sum()
```

Out[43]:

State	Retirees	Older_Single/Couples	Older_Families	Young_Families	Young_Single/Couples	Midage_Famili
Florida	80530	78799	90582	72551	49664	950
Georgia	83090	82701	95480	77371	52706	1010
Maryland	80239	78010	89523	71381	48954	936
New Jersey	81780	81146	93614	75701	51612	989
New York	86214	84393	97019	77725	53198	1018
North Carolina	79296	77247	88697	70822	48541	928
Pennsylvania	80279	77163	88285	69816	48043	917
South Carolina	80579	78810	90584	72527	49654	950
Virginia	80480	79600	91750	74033	50523	968

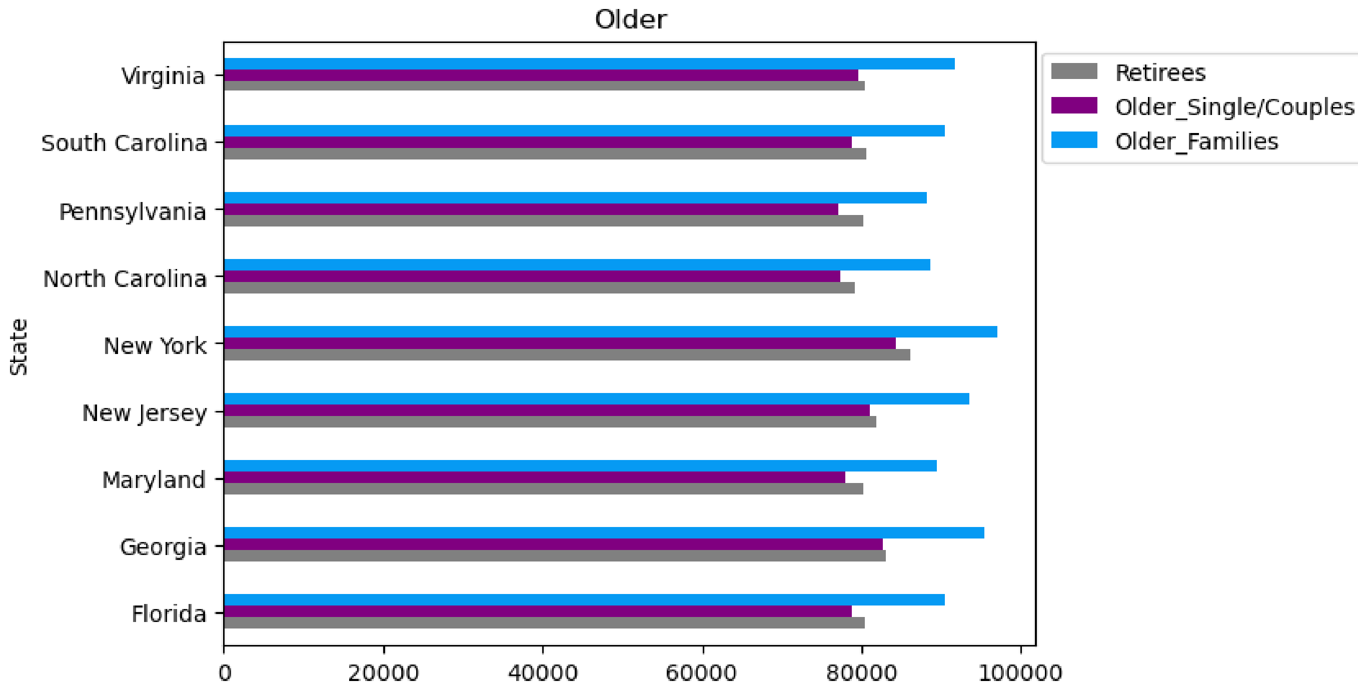
In [44]:
```
#grouping by age category
older = store_data_states["Retirees"].sum() + store_data_states["Older_Single/Couples"].
older_fam = store_data_states['Older_Families'].sum()
no_retirees = store_data_states["Older_Single/Couples"].sum()

midage = store_data_states['Midage_Singles/Couples'].sum()
midage_fam = store_data_states['Midage_Families'].sum()

young = store_data_states["Young_Single/Couples"].sum()
young_fam = store_data_states["Young_Families"].sum()

#grouping by family, non-family, and non family excluding retirees
non_families = older + midage + young
families_total = older_fam + midage_fam + young_fam
non_families_no_r =  no_retirees + midage + young
```

```python
print("Families")
print()
print(families_total)
print()
print("Non Families")
print()
print(non_families)
print()
print("None Families without Retirees")
print(non_families_no_r )
print()
print("Retirees")
print(store_data_states["Retirees"].sum())
```

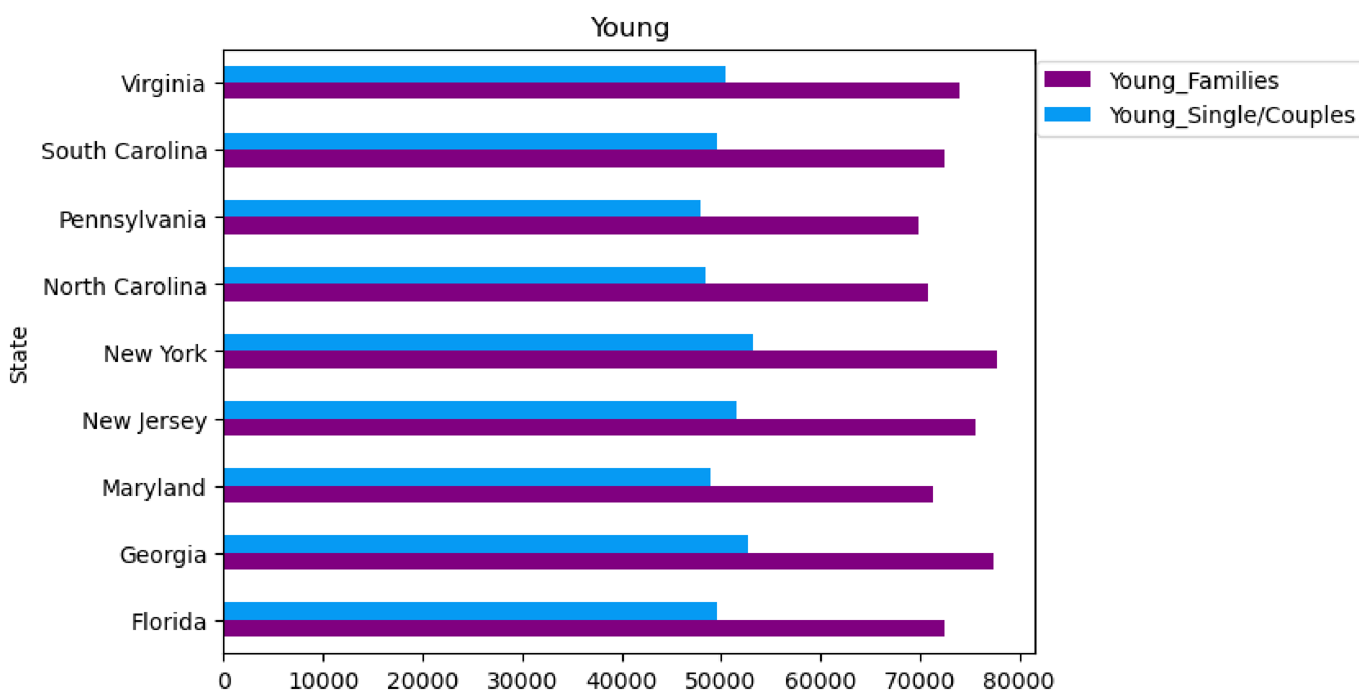
```
Families

State
Florida           258192
Georgia           273943
Maryland          254531
New Jersey        268275
New York          276576
North Carolina    252380
Pennsylvania      249855
South Carolina    258147
Virginia          262616
dtype: int64

Non Families

State
Florida           290792
Georgia           304718
Maryland          288048
New Jersey        299073
New York          311414
North Carolina    285182
Pennsylvania      285209
South Carolina    290842
Virginia          293463
dtype: int64

None Families without Retirees
State
Florida           210262
Georgia           221628
Maryland          207809
New Jersey        217293
New York          225200
North Carolina    205886
Pennsylvania      204930
South Carolina    210263
Virginia          212983
dtype: int64

Retirees
State
Florida           80530
Georgia           83090
Maryland          80239
New Jersey        81780
New York          86214
North Carolina    79296
Pennsylvania      80279
South Carolina    80579
```

```
Virginia          80480
Name: Retirees, dtype: int64
```

In [45]:
```python
store_data_states[['Retirees', 'Older_Single/Couples',
        'Older_Families']].sum().plot(kind="barh", color=colors)
plt.legend(loc="upper right", bbox_to_anchor=(1.42,1))
plt.title("Older")
plt.show()
```

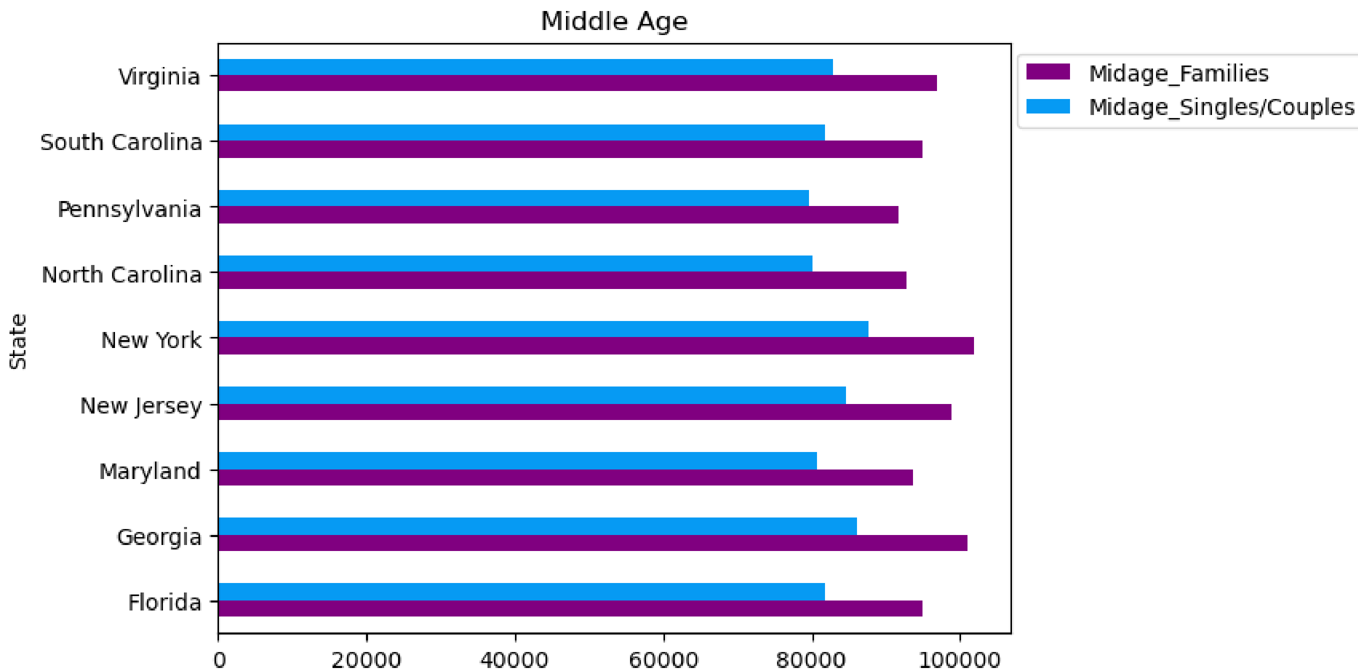


In [46]:
```python
colors2 = ['#800080', '#069AF3']

store_data_states[['Young_Families', 'Young_Single/Couples']].sum().plot(kind='barh', co
plt.legend(loc="upper right", bbox_to_anchor=(1.42,1))
plt.title("Young")
plt.show()
```

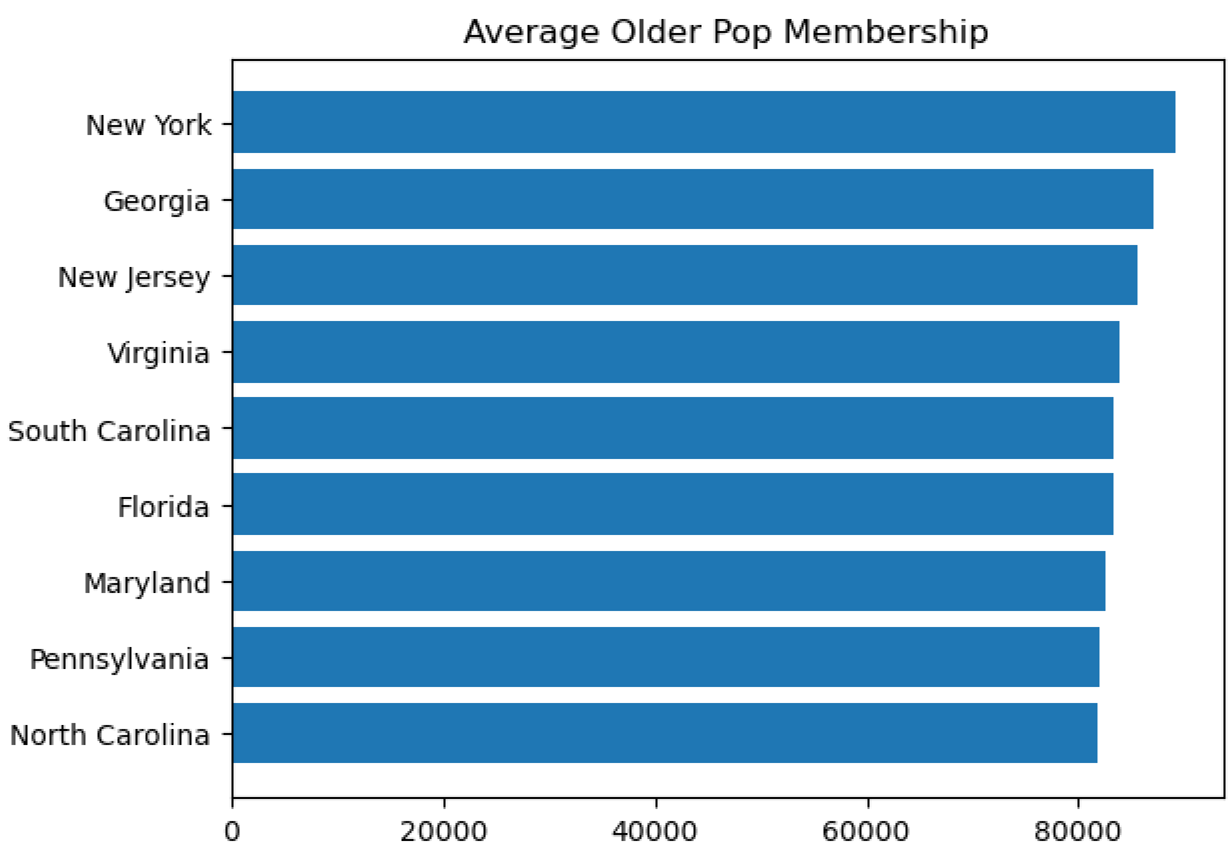


In [47]:
```python
store_data_states[['Midage_Families', 'Midage_Singles/Couples']].sum().plot(kind="barh",
plt.legend(loc="upper right", bbox_to_anchor=(1.46,1))
```

```
plt.title("Middle Age")
plt.show()
```

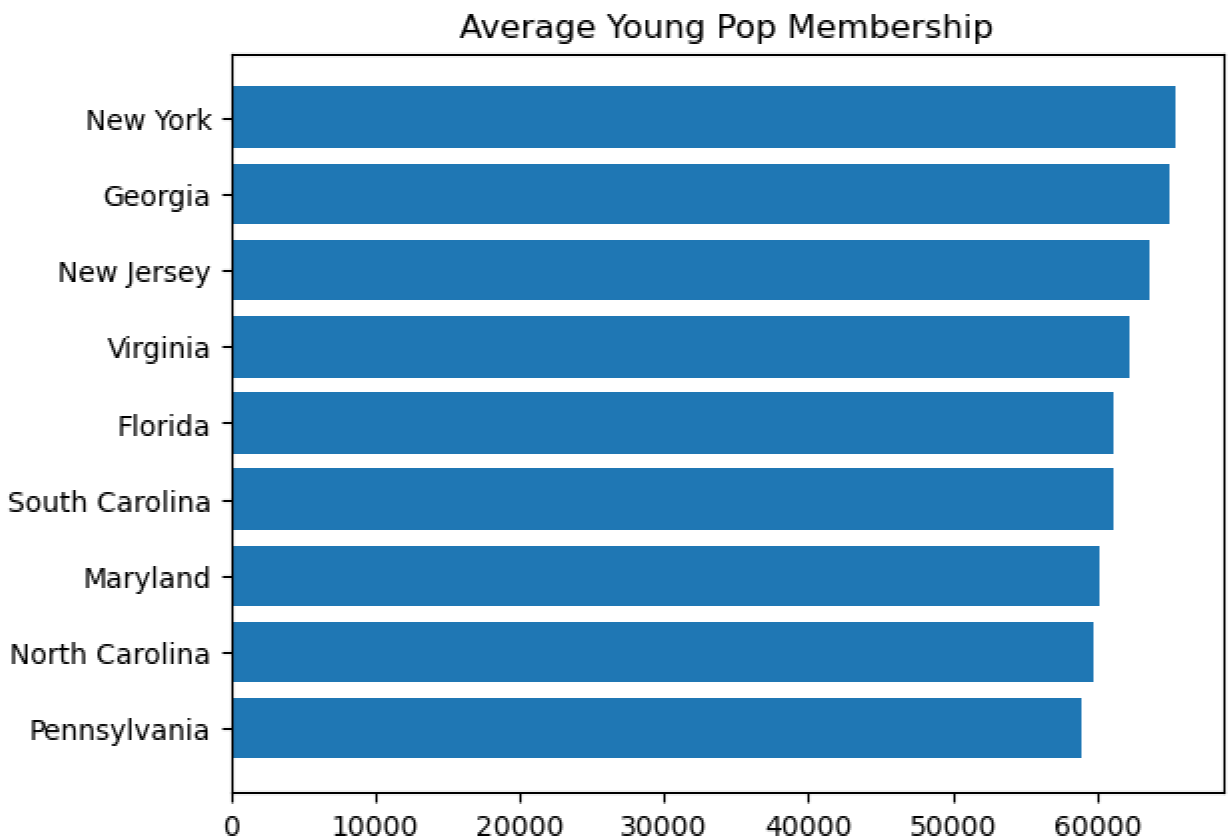


In [48]:
```
older_pop = store_data_states['Retirees'].sum() + store_data_states['Older_Single/Couple

older_pop_avg  = older_pop / 3

older_pop_avg = older_pop_avg.sort_values()

plt.barh(width = older_pop_avg.values, y=older_pop_avg.index)
plt.title("Average Older Pop Membership")
plt.show()
```

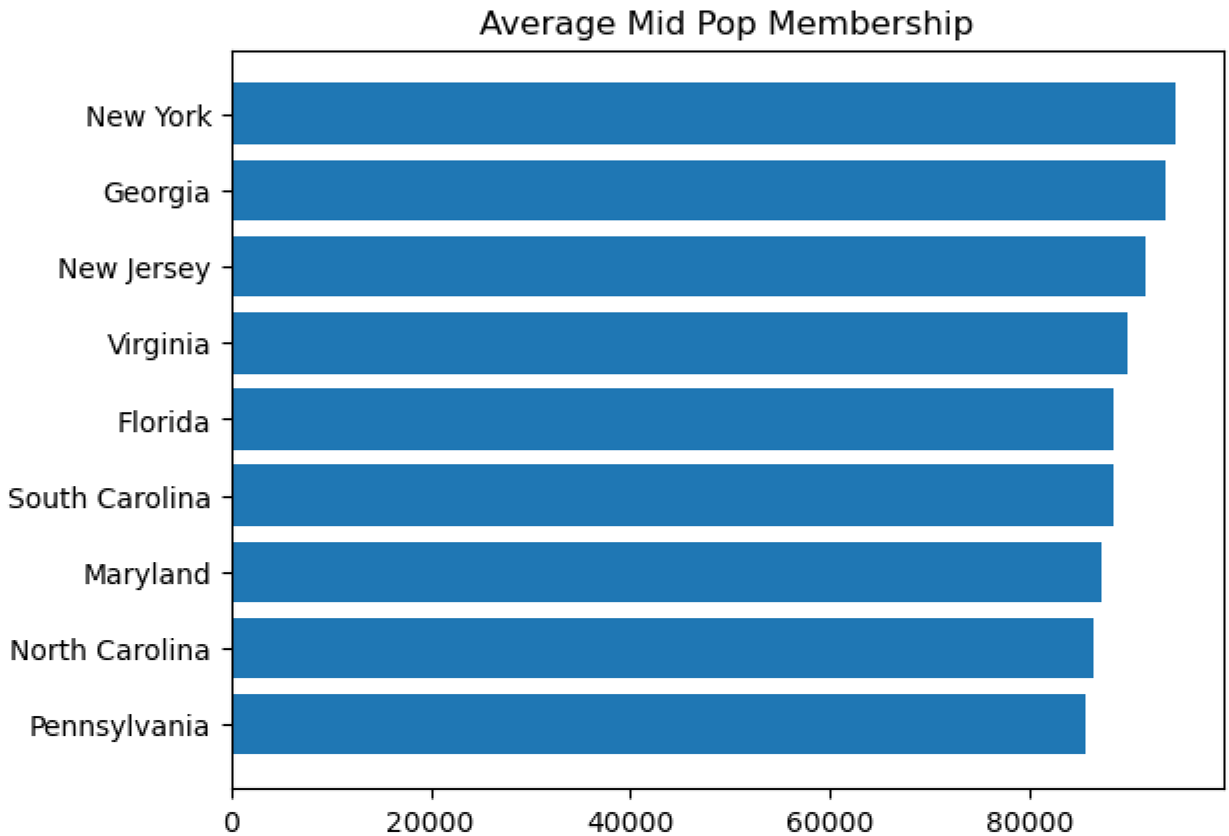


In [49]:
```
young_pop = store_data_states['Young_Families'].sum() + store_data_states['Young_Single/
young_pop_avg  = young_pop / 2
```

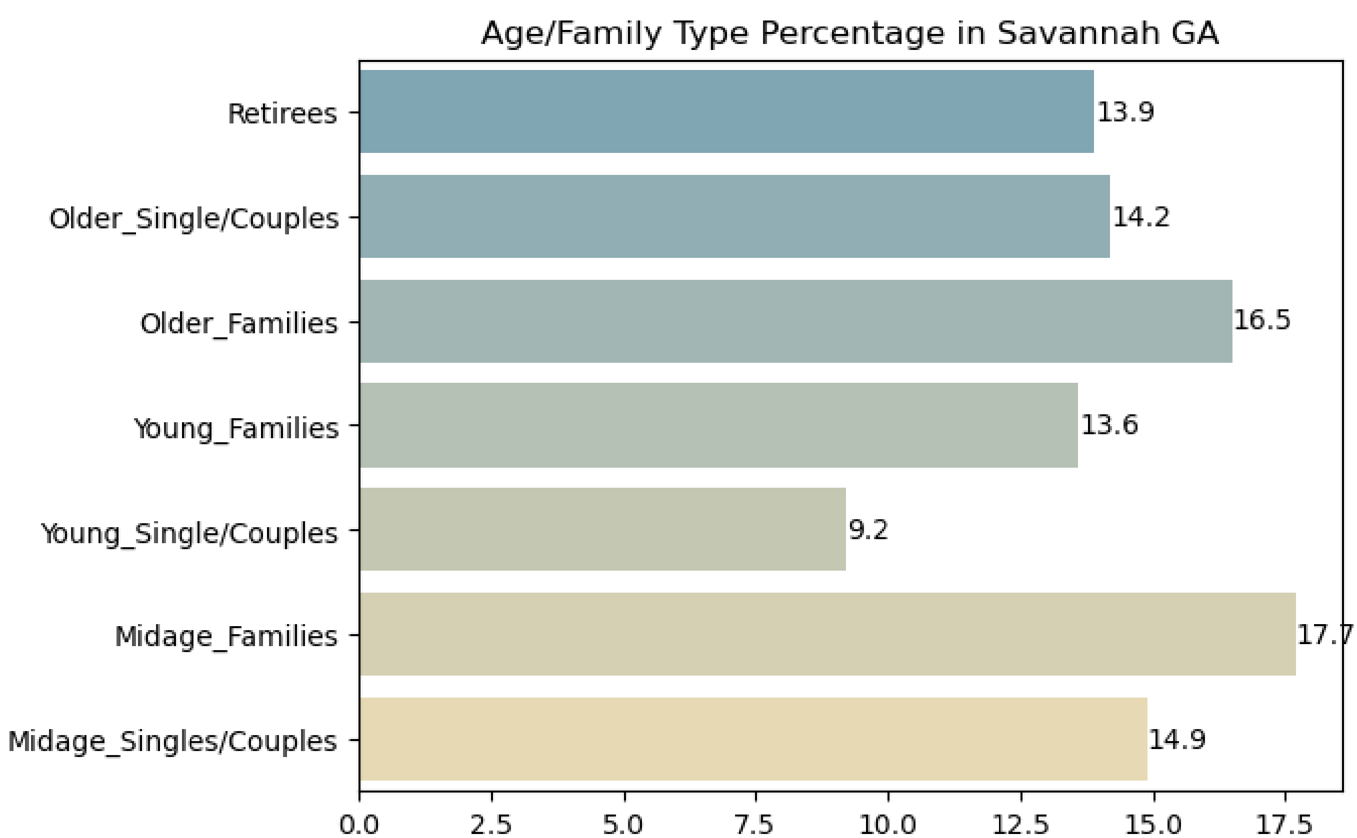
```
young_pop_avg = young_pop_avg.sort_values()

plt.barh(width = young_pop_avg.values, y=young_pop_avg.index)
plt.title("Average Young Pop Membership")
plt.show()
```



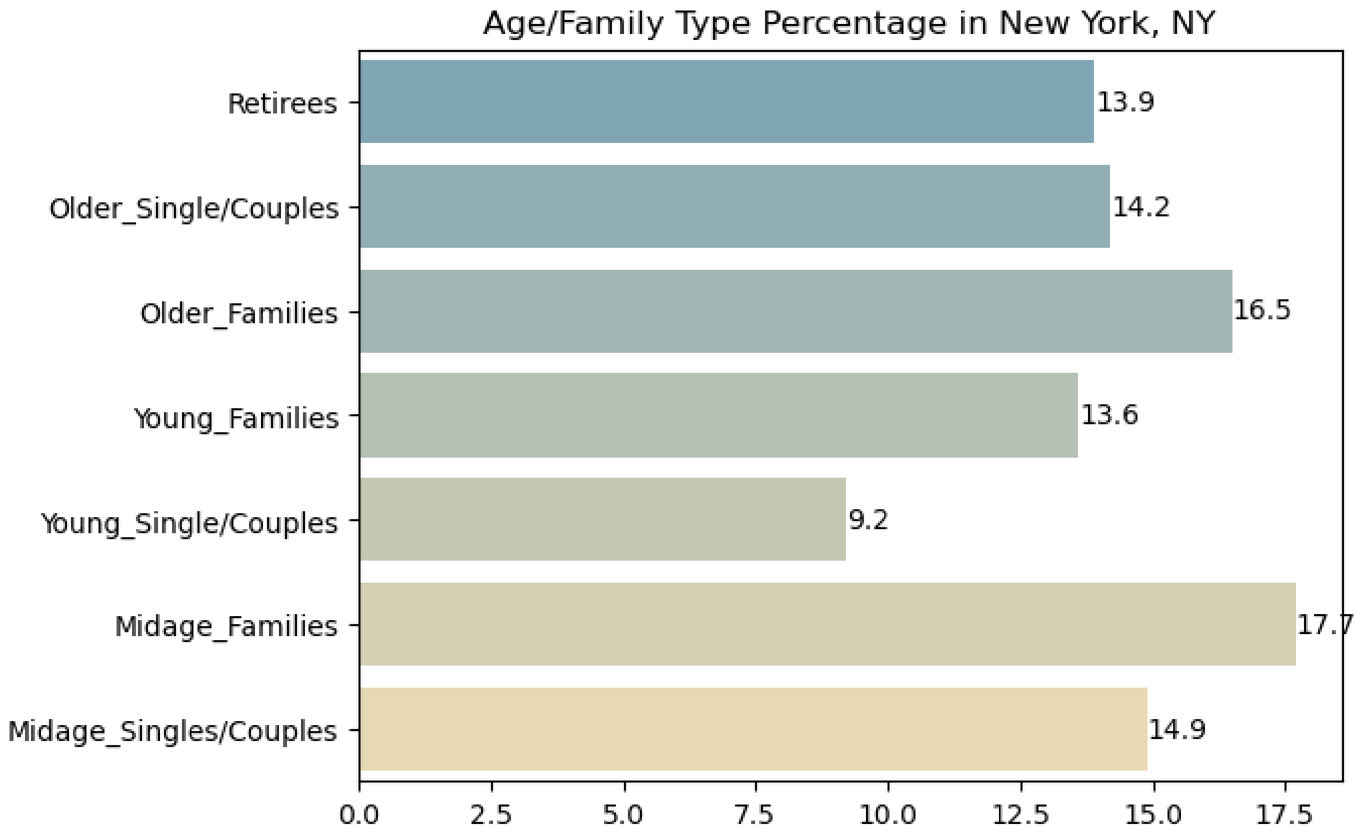
In [50]:
```
mid_pop = store_data_states['Midage_Families'].sum() + store_data_states['Midage_Singles
mid_pop_avg  = mid_pop / 2

mid_pop_avg = mid_pop_avg.sort_values()

plt.barh(width = mid_pop_avg.values, y=mid_pop_avg.index)
plt.title("Average Mid Pop Membership")
plt.show()
```



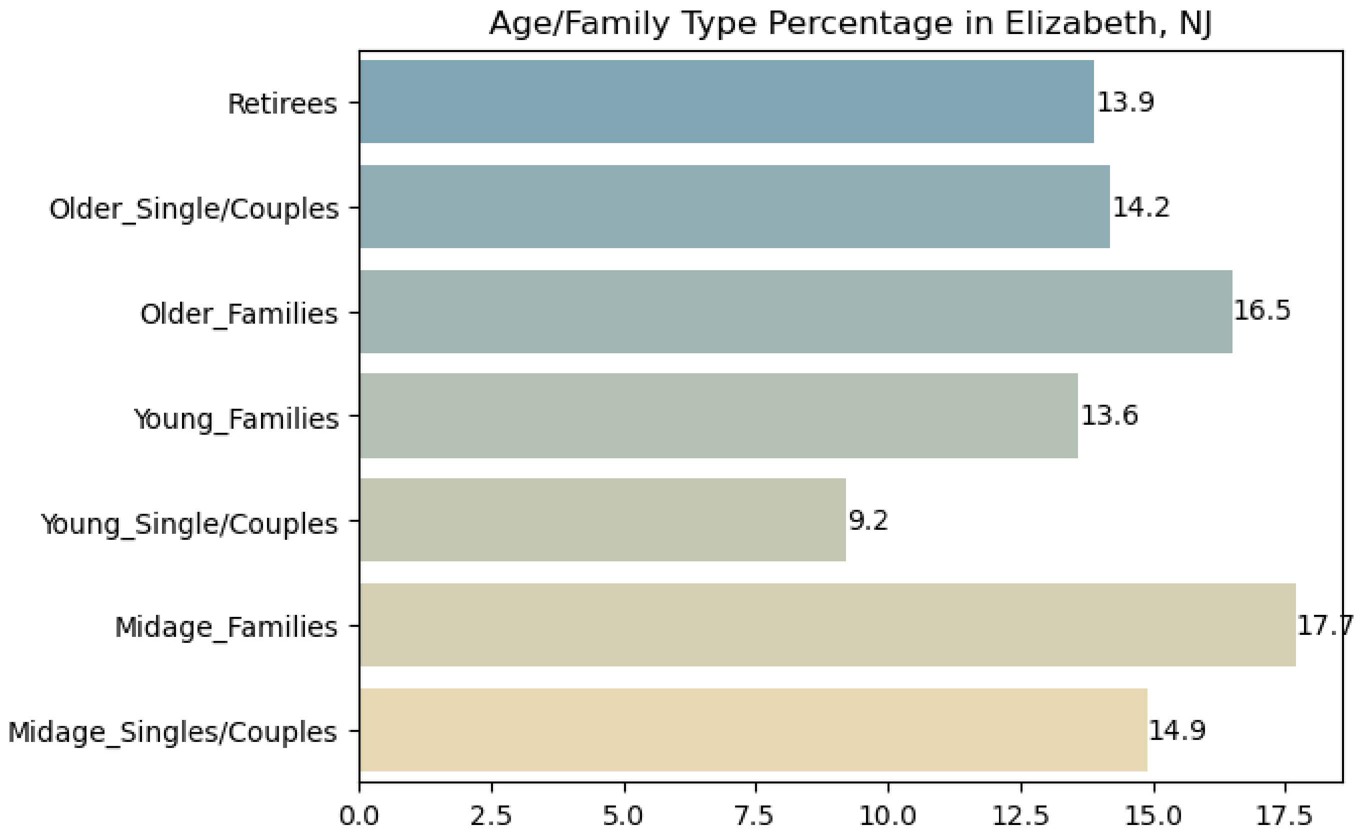
In [53]:
```
s_fm = round(savannah[["Retirees","Older_Single/Couples","Older_Families","Young_Familie
                         "Young_Single/Couples","Midage_Families", "Midage_Singles/Couples"]
                         / savannah["Total_Members"].sum() * 100,2)

s_fmplot = sns.barplot(x=s_fm.values, y=s_fm.index, palette = "blend:#7AB,#EDA")
plt.bar_label(s_fmplot.containers[0])
plt.title("Age/Family Type Percentage in Savannah GA")
plt.show()
```



In [54]:
```
ny_fm = round(ny_ny[["Retirees","Older_Single/Couples","Older_Families","Young_Families"
                     "Young_Single/Couples","Midage_Families", "Midage_Singles/Couples"]
                     / ny_ny["Total_Members"].sum() * 100,2)

ny_fmplot = sns.barplot(x=ny_fm.values, y=ny_fm.index, palette = "blend:#7AB,#EDA")
plt.bar_label(ny_fmplot.containers[0])
plt.title("Age/Family Type Percentage in New York, NY")
plt.show()
```



In [55]:
```
nj_fm = round(elizabeth[["Retirees","Older_Single/Couples","Older_Families","Young_Famil
                     "Young_Single/Couples","Midage_Families", "Midage_Singles/Couples"]
                     / elizabeth["Total_Members"].sum() * 100,2)

nj_fmplot = sns.barplot(x=nj_fm.values, y=nj_fm.index, palette = "blend:#7AB,#EDA")
plt.bar_label(nj_fmplot.containers[0])
plt.title("Age/Family Type Percentage in Elizabeth, NJ")
plt.show()
```



Notes:

The largest population of members are families between all age groups. Singles and couples tend to be drastically lower. Regarding by age segement it seems that middle age is the largest segment between all age groupes with the older age being a very close second. The Younger age group comes last in all segments. Viewing the top 3 cities the breakout is almost exact the differences between each are a few 100ths of percent.

Summary:

- I have explored the data by state, city, store, looking at net income and gross revenue.
- Determined the top 3 States are Georgia, New York, and New Jersey
- Top 3 cities are Savannah GA, NYC NY, and Elizabeth, NJ
- Performed 2 statistical tests on the Average Membership Length and net income with no conclusive determination that a statistical signficance exists based on the data at hand.
- Larger cities and metros outperformed smaller less populated cities.
- The average membership length is 7.2 years
- The largest share of membership type is mainstream members followed by budget, and than premium.
- Among the top 3 performing states Georgia has the largest population of premium members
- Looking at membership at the city level the best performing cities have a slightly higher premium membership and a more balance closer balance between budget and mainstream.
- The largest population of members are families between all age groups.
- Age segement it seems that middle age is the largest segment between all age groupes with the older age being a very close second.

Part 2:

The wholesale club has reached back out to ask if I can do some initial analysis to help on which state they should expand into next. They have narrowed down their choices to Connecticut, Delaware, Kentucky, Ohio, Tenesesse, and West Virginia.They want us to use publicly available data to help start their research.

- They want the Pros and Cons that we can gather on each state.
- To use the previous dataset and results to come up with a recommendation.
- They also want to use their competitors locations to assist in the decision making process.

Competitor Data:

Compiled store data of competitors and geographic information to assist in the analysis.

In [56]:
```
#compiled store data of competitors and geographic information
comp_data = pd.read_csv('wholesale_stores.csv', encoding='latin-1')

comp_data
```

Out[56]:

	Competitor	Address	City	County	State	ZipCode	Latitude	Longitude
0	Sam's Club	2500 Mountaineer Boulevard	South Charleston	Kanawha County	WV	25309	38.322039	-81.712078
1	Sam's Club	1100 Grand Central Avenue	Vienna	Wood County	WV	26105	39.309899	-81.550965
2	Sam's Club	5045 University Town Centre Drive	Morgantown	Monongalia	WV	26501	39.639221	-80.002206
3	Sam's Club	1220 N Eisenhower Drive	Beckley	Raleigh County	WV	25801	37.802850	-81.174016
4	Sam's Club	200 Emily Drrive	Clarksburg	Harrison County	WV	26301	39.275356	-80.279399
...	...	...	...	...	...	...	...	...
111	Costco Wholesale	3600 East Main Street	Waterbury	New Haven County	CT	6705	41.539500	-72.967654
112	Costco Wholesale	284 Flanders Road	East Lyme	New London County	CT	6333	41.359156	-72.213360
113	Costco Wholesale	1718 Boston Post Road	Milford	New Haven County	CT	6460	41.249140	-73.023454
114	Costco Wholesale	200 Federal Road	Brookfield	Fairfield County	CT	6804	41.442213	-73.406258
115	Costco Wholesale	779 Connecticut Avenue	Norwalk	Fairfield County	CT	6854	41.092447	-73.452720

116 rows × 8 columns

In [57]:
```
comp_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 116 entries, 0 to 115
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Competitor  116 non-null    object
 1   Address     116 non-null    object
 2   City        116 non-null    object
```

```
 3   County      116 non-null    object
 4   State       116 non-null    object
 5   ZipCode     116 non-null    int64
 6   Latitude    116 non-null    float64
 7   Longitude   116 non-null    float64
dtypes: float64(2), int64(1), object(5)
memory usage: 7.4+ KB
```

In [58]:
```
#competitor names
comp_data["Competitor"].unique()
```

Out[58]:
```
array(["Sam's Club", "BJ's Wholesale Club", 'Costco Wholesale '],
      dtype=object)
```

In [59]:
```
comp_grouped = comp_data.groupby("Competitor")

#count of stores
comp_grouped["State"].count()
```

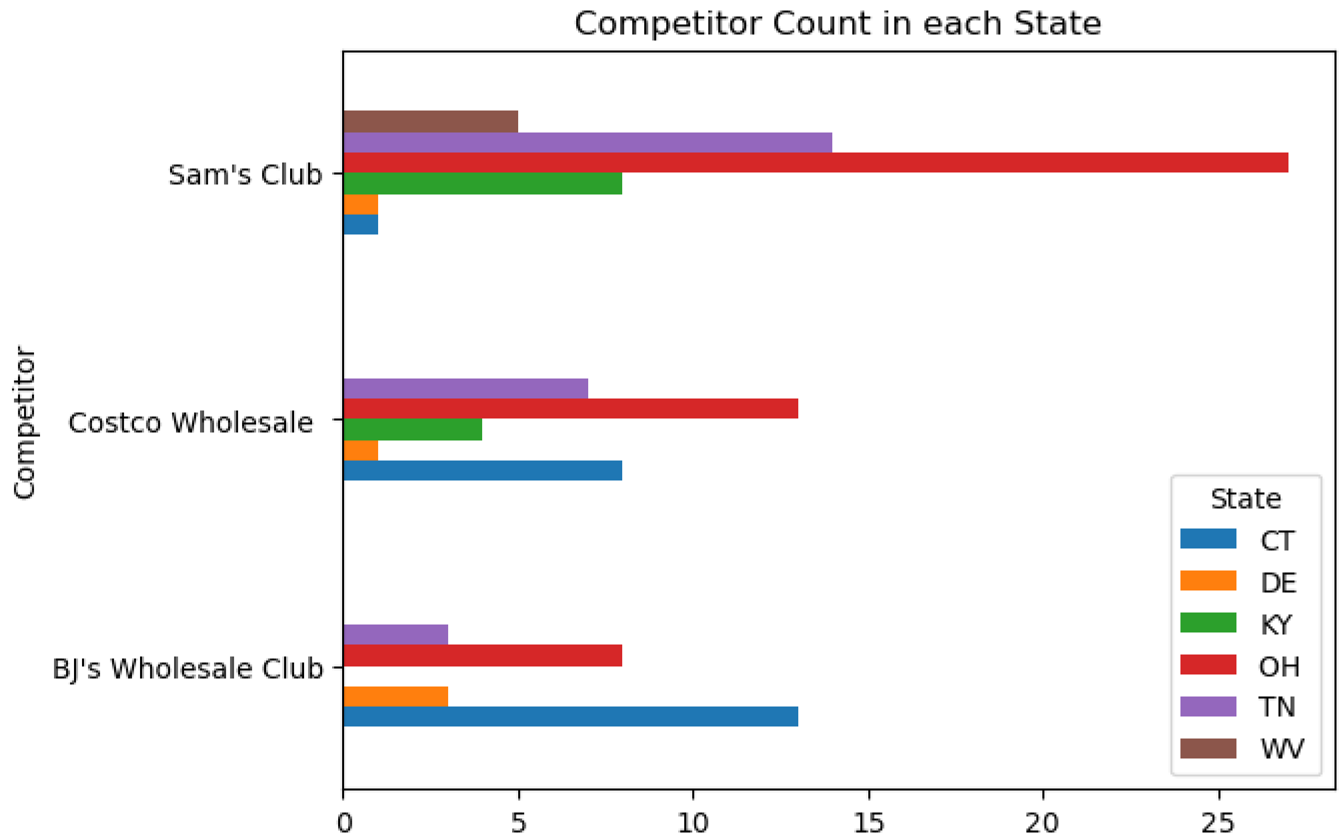
Out[59]:
```
Competitor
BJ's Wholesale Club    27
Costco Wholesale       33
Sam's Club             56
Name: State, dtype: int64
```

In [60]:
```
comp_state = comp_grouped["State"].value_counts()
```

In [61]:
```
comp_state
```

Out[61]:
```
Competitor           State
BJ's Wholesale Club  CT       13
                     OH        8
                     DE        3
                     TN        3
Costco Wholesale     OH       13
                     CT        8
                     TN        7
                     KY        4
                     DE        1
Sam's Club           OH       27
                     TN       14
                     KY        8
                     WV        5
                     CT        1
                     DE        1
Name: State, dtype: int64
```

In [62]:
```
comp_state.unstack().plot(kind='barh')
plt.title("Competitor Count in each State")
plt.show()
```



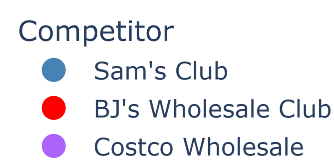
In [63]:
```python
comp_county = comp_data.groupby(["County"])
comp_count = comp_county[["State"]].value_counts()

comp_count
```

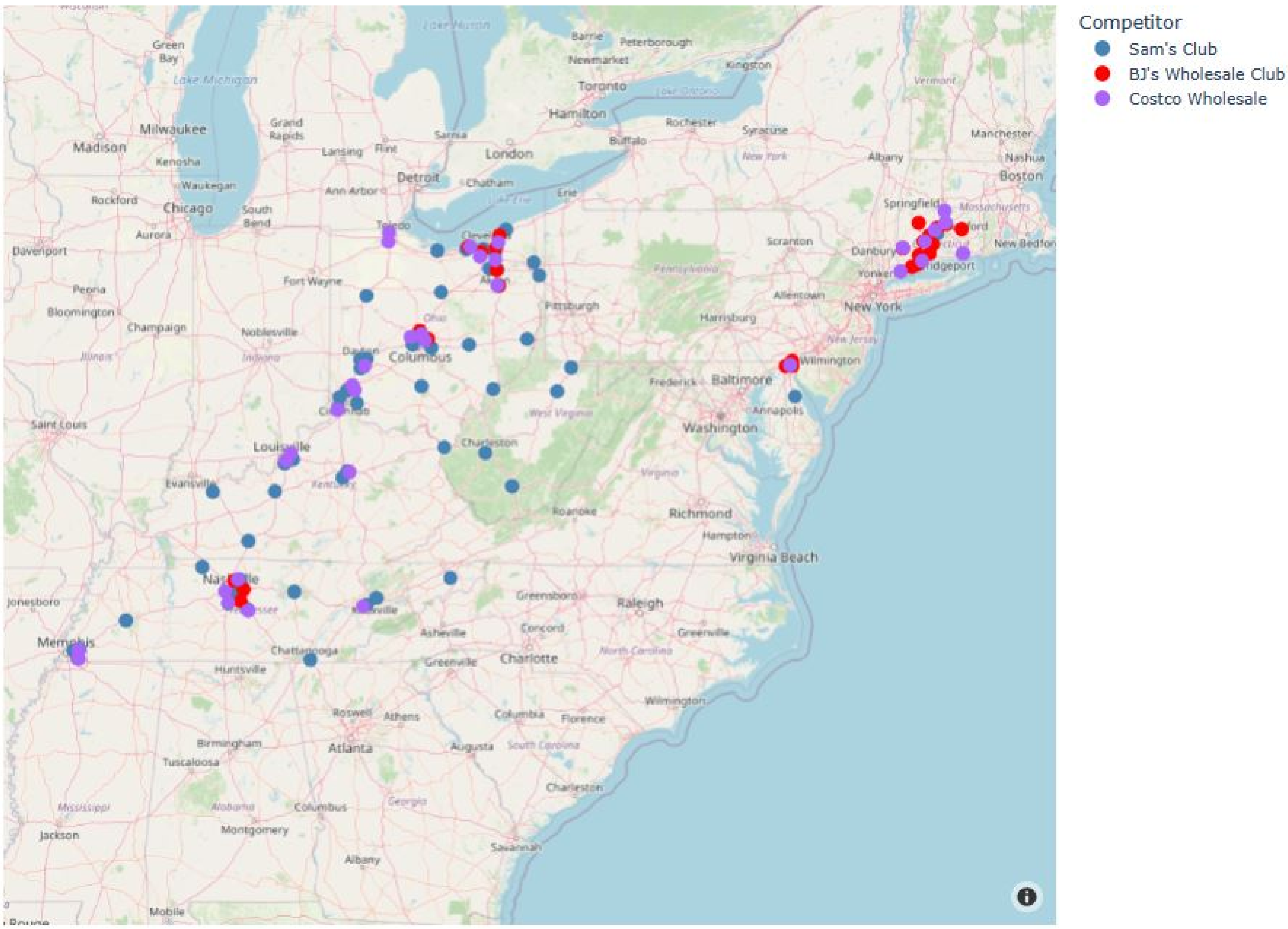
Out[63]:
```
County              State
Allen County        OH       1
Belmont County      OH       1
Boone County        KY       2
Butler County       OH       1
Cuyahoga County     OH       7
Davidson County     TN       3
Daviess County      KY       1
Delaware County     OH       1
Erie County         OH       1
Fairfield County    CT       6
Fayette County      KY       2
Franklin County     OH       7
Greene County       OH       1
Hamilton County     OH       5
                    TN       1
Hardin County       KY       1
Harrison County     WV       1
Hartford County     CT       7
Jefferson County    KY       4
Jessamine County    KY       1
Kanawha County      WV       1
Kent County         DE       1
Knox County         TN       3
Lake County         OH       2
Litchfield County   CT       1
Lorain County       OH       3
Lucas County        OH       1
Madison County      TN       1
Mahoning County     OH       1
Monongalia          WV       1
Montgomery County   OH       3
```

```
                     TN       1
Muskingum County   OH       1
New Castle County  DE       4
New Haven County   CT       6
New London County  CT       1
Putnam County      TN       1
Raleigh County     WV       1
Richland County    OH       1
Ross County        OH       1
Rutherford County  TN       3
Shelby County      TN       5
Stark County       OH       3
Summit County      OH       4
Sumner County      TN       2
Trumbull County    OH       1
Union County       OH       1
Warren County      KY       1
Washington County  TN       1
Williamson County  TN       2
Wilson County      TN       1
Windham County     CT       1
Wood County        OH       1
                   WV       1
dtype: int64
```

In [65]:

```
fig = px.scatter_mapbox(comp_data, lat="Latitude", lon="Longitude", hover_name="Competit
                        color="Competitor",
                        color_discrete_map={"Sam's Club":"steelblue","BJ's Wholesale Club
                        zoom=4, height=700)
fig.update_layout(mapbox_style='open-street-map')
fig.update_layout(margin= {"r":0,"t":0, "l":0,"b":0})
fig.update_traces(marker={"size":12})
fig.show()
```





To view and interact with virtual map current version of plotly.express needs to be installed.

Notes:

- Sam's Club is heavily concentrated in Ohio and Tennessee.
- BJ's is heavily concentrated in Connecticut and Ohio.
- Costco is heavily concentrated in Connecticut and Ohio.

- The only competitor in West Virginia is Sam's Club

- All are located in Delaware and have a store prescence in Wilmington but Sam's club is the only competitor in Dover.
- Bj's is not in Kentucky leaving only 2 competitors
- BJ's is also only in Nashville Tennessee which leaves only 2 competitors in the rest of the state.

Census Data:

I have collected American Community and Decennial Survey data from the Census Bureau website. I have cleaned and reformatted most of the data in excel. The information I have gathered are Age, Sex, Income, Race, Homeownership, Household demographics.

I have split the data into 2 portions. The top 3 performing states and the 6 states the company is looking to possibly enter. The top 3 states are New York, New Jersey, and Georgia. For each of these states I took the entire states data and the county which each of the top performing city is located in. With this information I plan to create key characteristics from each category to support selecting the next state to enter.

To recall Savannah, New York City, and Elizabeth were the top 3 cities. Savannah is located in Chatham County. NYC consists of 5 counties: New York, Richmond, Kings, Queens, and Bronx counties. Elizabeth is located in Richmond county.

I will be using this data to compare to the 6 other states, the current competitor locations, and the companies demographic information to conclude which state will be the best fit based on a ranking system. I will also give pros and cons of the 6 states.

In [473…

```
#loading in the data for top 3 states.
top_3_race = pd.read_csv("top_3_race.csv")
top_3_age  = pd.read_csv("top_3_age_sex.csv")
top_3_income = pd.read_csv("top_3_income.csv")
top_3_housing = pd.read_csv("top_3_housing.csv")
top_3_household = pd.read_csv("top_3_household.csv")
```

In [67]:

```
#correcting column name
top_3_race.rename(columns={"Total":"Total_Population"}, inplace=True)

top_3_race
```

Out[67]:

	County	State	Total_Population	Hispanic or Latino	Not Hispanic or Latino	Population of one race	White alone	Black or African American alone	American Indian and Alaska Native alone	Asi alo
0	Georgia	Georgia	10,711,908	1,123,457	9,588,451	9,198,318	5,362,156	3,278,119	20,375	475,6
1	Chatham County	Georgia	295,291	23,790	271,501	260,538	139,433	108,011	619	10,6
2	New Jersey	New Jersey	9,288,994	2,002,575	7,286,419	6,996,948	4,816,381	1,154,142	11,206	942,9
3	Union County	New Jersey	575,345	195,519	379,826	362,289	211,245	112,261	552	31,9
4	New York	New York	20,201,249	3,948,032	16,253,217	15,532,370	10,598,907	2,759,022	54,908	1,916,3
5	Bronx County	New York	1,472,654	806,463	666,191	637,821	130,796	419,393	3,087	67,7
6	Kings County	New York	2,736,074	516,426	2,219,648	2,106,478	968,427	729,696	3,964	370,7
7	New York County	New York	1,694,251	402,640	1,291,611	1,228,622	793,294	199,592	1,895	219,6
8	Queens County	New York	2,405,464	667,861	1,737,603	1,653,491	549,358	381,375	9,576	656,5

9	Richmond County	New York	495,747	96,960	398,787	387,469	277,981	46,835	624	58,7

In [68]:
```
#viewing just the state data.
top_3_race.iloc[[0,2,4]]
```

Out[68]:

	County	State	Total_Population	Hispanic or Latino	Not Hispanic or Latino	Population of one race	White alone	Black or African American alone	American Indian and Alaska Native alone	Asian alone
0	Georgia	Georgia	10,711,908	1,123,457	9,588,451	9,198,318	5,362,156	3,278,119	20,375	475,68
2	New Jersey	New Jersey	9,288,994	2,002,575	7,286,419	6,996,948	4,816,381	1,154,142	11,206	942,92
4	New York	New York	20,201,249	3,948,032	16,253,217	15,532,370	10,598,907	2,759,022	54,908	1,916,32

Notes:

Based on the racial data we cant give that much consideration to population since NY came in second and Georgia came in first. What is interesting Georgia has the largest Black/African American population and Native Hawaiian/Pacific Islander. Lets drill down to the county level.

In [69]:
```
top_3_race.iloc[[1,3,5,6,7,8,9]]
```

Out[69]:

	County	State	Total_Population	Hispanic or Latino	Not Hispanic or Latino	Population of one race	White alone	Black or African American alone	American Indian and Alaska Native alone	Asian alone
1	Chatham County	Georgia	295,291	23,790	271,501	260,538	139,433	108,011	619	10,620
3	Union County	New Jersey	575,345	195,519	379,826	362,289	211,245	112,261	552	31,963
5	Bronx County	New York	1,472,654	806,463	666,191	637,821	130,796	419,393	3,087	67,766
6	Kings County	New York	2,736,074	516,426	2,219,648	2,106,478	968,427	729,696	3,964	370,776
7	New York County	New York	1,694,251	402,640	1,291,611	1,228,622	793,294	199,592	1,895	219,624
8	Queens County	New York	2,405,464	667,861	1,737,603	1,653,491	549,358	381,375	9,576	656,583
9	Richmond County	New York	495,747	96,960	398,787	387,469	277,981	46,835	624	58,753

Notes:

- The majority population of Chatham county seems to be white and black/african american. Which makes up 83% of their population.
- The racial split is 47% white and 36.5% black/african american.
- New Jersey is more disbursed between races 34% Hispanic, 37% white, 20% black/african american.
- New York's counties vary with one dominating race. For example Richmond County which is Staten Island is 78% white. While the Bronx is dominated by Hispanic at 54%.

What this data is showing me looking at race is that it seems diversity is a contributing factor to these stores. The top 3 states are a perfect example of large city diversity. I will have to keep this in mind when reviewing the 6 potential states. The best chance of success seems to be populations not dominated by a single racial background.

Age Data:

The age data we have from the original dataset is old, midage, and young.

We can break this up into 3 age ranges. young = 20-39, midage = 40-59,and older = 60+ Using this format we can pair up the companies demographics with the census age demographics. To recall midage was the companies largest age segement with older being a very close 2nd. Young being the lowest of age demographics.

In [70]:
```
top_3_age
```

Out[70]:

	County	State	Category	Total population	Under 5 years	5 to 9 years	10 to 14 years	15 to 19 years	20 to 24 years	25 to 29 years	...
0	Georgia	Georgia	Total	10,912,876	621,126	683,215	741,043	762,949	771,563	730,956	...
1	Georgia	Georgia	Percent	(X)	5.7%	6.3%	6.8%	7.0%	7.1%	6.7%	...
2	Georgia	Georgia	Male	5,323,951	319,188	349,200	377,740	390,500	392,098	365,255	...
3	Georgia	Georgia	Percent Male	(X)	6.0%	6.6%	7.1%	7.3%	7.4%	6.9%	...
4	Georgia	Georgia	Female	5,588,925	301,938	334,015	363,303	372,449	379,465	365,701	...
5	Georgia	Georgia	Percent Female	(X)	5.4%	6.0%	6.5%	6.7%	6.8%	6.5%	...
6	Chatham County	Georgia	Total	301,107	17,355	16,217	17,373	19,488	23,572	23,242	...
7	Chatham County	Georgia	Percent	(X)	5.8%	5.4%	5.8%	6.5%	7.8%	7.7%	...
8	Chatham County	Georgia	Male	144,407	8,744	7,976	9,340	9,311	11,413	11,676	...
9	Chatham County	Georgia	Percent Male	(X)	6.1%	5.5%	6.5%	6.4%	7.9%	8.1%	...
10	Chatham County	Georgia	Female	156,700	8,611	8,241	8,033	10,177	12,159	11,566	...
11	Chatham County	Georgia	Percent Female	(X)	5.5%	5.3%	5.1%	6.5%	7.8%	7.4%	...
12	New Jersey	New Jersey	Total	9,261,699	513,333	533,608	585,993	576,961	569,581	575,079	...

13	New Jersey	New Jersey	Percent	(X)	5.5%	5.8%	6.3%	6.2%	6.1%	6.2%	...
14	New Jersey	New Jersey	Male	4,564,704	261,922	274,175	299,362	296,411	288,273	293,852	...
15	New Jersey	New Jersey	Percent Male	(X)	5.7%	6.0%	6.6%	6.5%	6.3%	6.4%	...
16	New Jersey	New Jersey	Female	4,696,995	251,411	259,433	286,631	280,550	281,308	281,227	...
17	New Jersey	New Jersey	Percent Female	(X)	5.4%	5.5%	6.1%	6.0%	6.0%	6.0%	...
18	Union County	New Jersey	Total	569,815	34,256	35,470	38,722	36,580	35,044	33,212	...
19	Union County	New Jersey	Percent	(X)	6.0%	6.2%	6.8%	6.4%	6.2%	5.8%	...
20	Union County	New Jersey	Male	281,970	17,805	17,058	20,791	18,923	17,815	16,759	...
21	Union County	New Jersey	Percent Male	(X)	6.3%	6.0%	7.4%	6.7%	6.3%	5.9%	...
22	Union County	New Jersey	Female	287,845	16,451	18,412	17,931	17,657	17,229	16,453	...
23	Union County	New Jersey	Percent Female	(X)	5.7%	6.4%	6.2%	6.1%	6.0%	5.7%	...
24	New York	New York	Total	19,677,151	1,055,455	1,070,033	1,161,685	1,198,745	1,298,992	1,349,368	...
25	New York	New York	Percent	(X)	5.4%	5.4%	5.9%	6.1%	6.6%	6.9%	...
26	New York	New York	Male	9,628,899	543,601	545,792	598,202	609,954	649,835	673,807	...
27	New York	New York	Percent Male	(X)	5.6%	5.7%	6.2%	6.3%	6.7%	7.0%	...
28	New York	New York	Female	10,048,252	511,854	524,241	563,483	588,791	649,157	675,561	...
29	New York	New York	Percent Female	(X)	5.1%	5.2%	5.6%	5.9%	6.5%	6.7%	...
30	Bronx County	New York	Total	1,379,946	90,674	88,115	99,932	92,854	96,715	99,220	...
31	Bronx County	New York	Percent	(X)	6.6%	6.4%	7.2%	6.7%	7.0%	7.2%	...
32	Bronx County	New York	Male	653,279	46,478	46,498	49,270	47,377	48,876	48,089	...
33	Bronx County	New York	Percent Male	(X)	7.1%	7.1%	7.5%	7.3%	7.5%	7.4%	...
34	Bronx County	New York	Female	726,667	44,196	41,617	50,662	45,477	47,839	51,131	...
35	Bronx County	New York	Percent Female	(X)	6.1%	5.7%	7.0%	6.3%	6.6%	7.0%	...
36	Kings County	New York	Total	2,590,516	166,970	154,625	160,796	139,691	153,230	212,419	...

37	Kings County	New York	Percent	(X)	6.4%	6.0%	6.2%	5.4%	5.9%	8.2%	...
38	Kings County	New York	Male	1,235,007	84,944	79,698	81,904	71,192	74,467	100,953	...
39	Kings County	New York	Percent Male	(X)	6.9%	6.5%	6.6%	5.8%	6.0%	8.2%	...
40	Kings County	New York	Female	1,355,509	82,026	74,927	78,892	68,499	78,763	111,466	...
41	Kings County	New York	Percent Female	(X)	6.1%	5.5%	5.8%	5.1%	5.8%	8.2%	...
42	New York County	New York	Total	1,596,273	66,445	54,049	67,184	72,265	111,696	163,335	...
43	New York County	New York	Percent	(X)	4.2%	3.4%	4.2%	4.5%	7.0%	10.2%	...
44	New York County	New York	Male	763,019	33,790	25,473	36,043	33,567	47,715	79,256	...
45	New York County	New York	Percent Male	(X)	4.4%	3.3%	4.7%	4.4%	6.3%	10.4%	...
46	New York County	New York	Female	833,254	32,655	28,576	31,141	38,698	63,981	84,079	...
47	New York County	New York	Percent Female	(X)	3.9%	3.4%	3.7%	4.6%	7.7%	10.1%	...
48	Queens County	New York	Total	2,278,029	122,662	120,419	126,359	116,177	128,273	160,284	...
49	Queens County	New York	Percent	(X)	5.4%	5.3%	5.5%	5.1%	5.6%	7.0%	...
50	Queens County	New York	Male	1,114,721	63,121	59,348	67,254	60,066	63,882	79,371	...
51	Queens County	New York	Percent Male	(X)	5.7%	5.3%	6.0%	5.4%	5.7%	7.1%	...
52	Queens County	New York	Female	1,163,308	59,541	61,071	59,105	56,111	64,391	80,913	...
53	Queens County	New York	Percent Female	(X)	5.1%	5.2%	5.1%	4.8%	5.5%	7.0%	...
54	Richmond County	New York	Total	491,133	25,920	28,469	31,015	29,650	30,699	31,319	...
55	Richmond County	New York	Percent	(X)	5.3%	5.8%	6.3%	6.0%	6.3%	6.4%	...
56	Richmond County	New York	Male	241,330	13,774	14,081	16,488	15,121	15,968	15,813	...
57	Richmond County	New York	Percent Male	(X)	5.7%	5.8%	6.8%	6.3%	6.6%	6.6%	...
58	Richmond County	New York	Female	249,803	12,146	14,388	14,527	14,529	14,731	15,506	...
59	Richmond County	New York	Percent Female	(X)	4.9%	5.8%	5.8%	5.8%	5.9%	6.2%	...

60 rows × 34 columns

In [71]:
```
#viewing total percentages for Georgia and Chatham County in Georgia
ga_age = top_3_age.iloc[[1,7]]

ga_age.iloc[:,4:22]
```

Out[71]:

	Under 5 years	5 to 9 years	10 to 14 years	15 to 19 years	20 to 24 years	25 to 29 years	30 to 34 years	35 to 39 years	40 to 44 years	45 to 49 years	50 to 54 years	55 to 59 years	60 to 64 years	65 to 69 years	70 to 74 years	75 to 79 years
1	5.7%	6.3%	6.8%	7.0%	7.1%	6.7%	6.9%	6.7%	6.8%	6.2%	6.6%	6.2%	6.0%	5.0%	4.2%	2.9%
7	5.8%	5.4%	5.8%	6.5%	7.8%	7.7%	7.7%	6.6%	6.6%	5.4%	5.8%	6.1%	6.0%	5.5%	4.4%	3.4%

In [72]:
```
#removing the percentage symbol
ga_age = ga_age.replace("%","", regex=True)
```

In [73]:
```
#converting data into a float
ga_age_float = ga_age.iloc[:,4:22].astype(float)

#combining by age groups
ga_young = ga_age_float["20 to 24 years"] + ga_age_float["25 to 29 years"] + ga_age_floa

ga_mid = ga_age_float["40 to 44 years"] + ga_age_float["45 to 49 years"] + ga_age_float[

ga_older = ga_age_float["60 to 64 years"] + ga_age_float["65 to 69 years"] + ga_age_floa

print("State = 1 County = 7")
print()
print("Young Adults")
print(ga_young)
print()
print("Midage Adults")
print(ga_mid)
print()
print("Older Adults")
print(ga_older)
```

```
State = 1 County = 7

Young Adults
1    27.4
7    29.8
dtype: float64

Midage Adults
1    25.8
7    23.9
dtype: float64

Older Adults
1    21.1
7    22.8
dtype: float64
```

In [74]:
```
#viewing total percentages for New Jersey and Union County in New Jersey
nj_age = top_3_age.iloc[[13,19]]

nj_age.iloc[:,4:22]
```

Out[74]:

	Under 5 years	5 to 9 years	10 to 14 years	15 to 19 years	20 to 24 years	25 to 29 years	30 to 34 years	35 to 39 years	40 to 44 years	45 to 49 years	50 to 54 years	55 to 59 years	60 to 64 years	65 to 69 years	70 to 74 years	75 to 79 years

13	5.5%	5.8%	6.3%	6.2%	6.1%	6.2%	6.6%	6.7%	6.6%	6.2%	6.7%	6.7%	6.8%	5.5%	4.5%	3.4%
19	6.0%	6.2%	6.8%	6.4%	6.2%	5.8%	6.5%	7.0%	7.1%	6.7%	7.0%	7.4%	5.6%	5.0%	3.8%	2.6%

In [75]:
```
#removing the percentage symbol
nj_age = nj_age.replace("%","", regex=True)
```

In [76]:
```
#converting data into a float
nj_age_float = nj_age.iloc[:,4:22].astype(float)

#combining by age groups
nj_young = nj_age_float["20 to 24 years"] + nj_age_float["25 to 29 years"] + nj_age_floa

nj_mid = nj_age_float["40 to 44 years"] + nj_age_float["45 to 49 years"] + nj_age_float[

nj_older = nj_age_float["60 to 64 years"] + nj_age_float["65 to 69 years"] + nj_age_floa

print("State = 13 County = 19")
print()
print("Young Adults")
print(nj_young)
print()
print("Midage Adults")
print(nj_mid)
print()
print("Older Adults")
print(nj_older)
```

```
State = 13 County = 19

Young Adults
13    25.6
19    25.5
dtype: float64

Midage Adults
13    26.2
19    28.2
dtype: float64

Older Adults
13    24.3
19    20.8
dtype: float64
```

In [77]:
```
#viewing total percentages for New York and all counties in NY
ny_age = top_3_age.iloc[[25,31,37,43,49,55]]

ny_age.iloc[:,4:22]
```

Out[77]:

	Under 5 years	5 to 9 years	10 to 14 years	15 to 19 years	20 to 24 years	25 to 29 years	30 to 34 years	35 to 39 years	40 to 44 years	45 to 49 years	50 to 54 years	55 to 59 years	60 to 64 years	65 to 69 years	70 to 74 years	75 t 7 year
25	5.4%	5.4%	5.9%	6.1%	6.6%	6.9%	7.2%	6.6%	6.3%	5.8%	6.4%	6.6%	6.7%	5.6%	4.7%	3.4
31	6.6%	6.4%	7.2%	6.7%	7.0%	7.2%	7.6%	6.2%	6.6%	5.8%	6.1%	6.2%	5.9%	4.7%	3.5%	2.6
37	6.4%	6.0%	6.2%	5.4%	5.9%	8.2%	9.1%	7.8%	6.5%	5.8%	5.8%	5.7%	5.5%	4.9%	4.0%	2.8
43	4.2%	3.4%	4.2%	4.5%	7.0%	10.2%	10.0%	8.1%	6.3%	5.9%	6.2%	6.2%	5.4%	5.0%	4.7%	3.8
49	5.4%	5.3%	5.5%	5.1%	5.6%	7.0%	7.8%	6.7%	6.8%	6.3%	6.8%	6.9%	6.7%	5.8%	4.6%	3.0

55	5.3%	5.8%	6.3%	6.0%	6.3%	6.4%	6.8%	6.1%	6.6%	6.3%	6.9%	7.0%	6.9%	4.9%	5.3%	3.3

In [78]:
```
#removing the percentage symbol
ny_age = ny_age.replace("%","", regex=True)
```

In [79]:
```
#converting data into a float
ny_age_float = ny_age.iloc[:,4:22].astype(float)

#combining by age groups
ny_young = ny_age_float["20 to 24 years"] + ny_age_float["25 to 29 years"] + ny_age_floa

ny_mid = ny_age_float["40 to 44 years"] + ny_age_float["45 to 49 years"] + ny_age_float[

ny_older = ny_age_float["60 to 64 years"] + ny_age_float["65 to 69 years"] + ny_age_floa

print("State = 25 Bronx = 31 Kings = 37 NY = 43 Queens = 49 Richmond = 55")
print()
print("Young Adults")
print(ny_young)
print()
print("Midage Adults")
print(ny_mid)
print()
print("Older Adults")
print(ny_older)
```

```
State = 25 Bronx = 31 Kings = 37 NY = 43 Queens = 49 Richmond = 55

Young Adults
25    27.3
31    28.0
37    31.0
43    35.3
49    27.1
55    25.6
dtype: float64

Midage Adults
25    25.1
31    24.7
37    23.8
43    24.6
49    26.8
55    26.8
dtype: float64

Older Adults
25    24.8
31    20.4
37    21.1
43    23.7
49    24.7
55    24.3
dtype: float64
```

Notes:

As I can see the middle age demographic is higher than older which follows the demographics of the companies data set when it comes to member population. Georgia and New York have a larger young adult population than any other category. This does not support the companies membership statistics but nonethless it is good regardless since everyone ages. The company data also shows that families are more likely to shop with them. So as the younger adults start families and age this will open up potential customers.

Income_Data:

To recall the membership type that was 1st was mainstream, 2nd was budget, and 3rd was premium.

We can break this up into 3 categories by income level as well. budget = 0-49,000, mainstream 50,000-149,000, premium = 150,000+

Whats nice is this data has families, non families, and married couples. Which will really supplement the original dataset.

In [80]:

```
top_3_income
```

Out[80]:

	County	State	Category	Total	Less than $10,000	10, 000*to* 14,999	15, 000*to* 24,999	25, 000*to* 34,999	35, 000*to* 49,999	50, 000*to* 74,999	75, 000*to* 99,999	100, 000*to* 149,999
0	Georgia	Georgia	Households	4,092,467	5.6%	3.6%	6.9%	7.3%	11.2%	16.8%	13.3%	16.9%
1	Georgia	Georgia	Families	2,733,234	3.8%	1.8%	4.5%	5.9%	10.2%	16.6%	14.3%	19.4%
2	Georgia	Georgia	Married-couple families	1,917,471	1.5%	0.8%	2.6%	3.8%	8.0%	14.9%	15.1%	22.7%
3	Georgia	Georgia	Nonfamily households	1,359,233	10.2%	7.6%	12.3%	11.0%	13.7%	17.4%	10.4%	10.5%
4	Chatham County	Georgia	Households	121,527	5.6%	4.3%	8.2%	8.8%	13.6%	15.8%	14.2%	15.2%
5	Chatham County	Georgia	Families	70,429	3.8%	4.1%	4.3%	6.9%	11.3%	13.6%	18.2%	17.2%
6	Chatham County	Georgia	Married-couple families	48,070	2.0%	2.4%	1.3%	3.4%	8.7%	11.8%	21.2%	22.1%
7	Chatham County	Georgia	Nonfamily households	51,098	8.5%	5.8%	15.0%	11.4%	16.2%	18.3%	8.8%	11.2%
8	New Jersey	New Jersey	Households	3,516,978	4.4%	2.9%	5.0%	5.9%	7.9%	13.7%	11.6%	18.1%
9	New Jersey	New Jersey	Families	2,378,459	2.2%	1.4%	3.4%	4.2%	6.8%	12.4%	11.5%	20.0%
10	New Jersey	New Jersey	Married-couple families	1,753,523	1.2%	0.7%	1.9%	2.6%	4.9%	10.5%	10.7%	20.9%
11	New Jersey	New Jersey	Nonfamily households	1,138,519	9.7%	6.4%	8.7%	10.2%	11.2%	17.2%	11.6%	13.1%
12	Union County	New Jersey	Households	202,575	2.3%	3.2%	4.9%	6.3%	8.5%	14.2%	11.1%	17.2%
13	Union County	New Jersey	Families	145,607	1.5%	1.5%	2.9%	5.1%	8.5%	13.7%	10.1%	17.5%
14	Union County	New Jersey	Married-couple families	103,296	1.1%	1.0%	2.3%	2.8%	5.6%	10.9%	9.0%	18.6%

15	Union County	New Jersey	Nonfamily households	56,968	5.2%	8.5%	9.8%	12.0%	11.3%	17.0%	12.9%	13.2%
16	New York	New York	Households	7,774,308	6.8%	4.1%	6.7%	6.5%	9.3%	14.3%	11.6%	16.4%
17	New York	New York	Families	4,738,232	4.2%	2.2%	4.7%	5.2%	8.2%	13.6%	12.3%	19.0%
18	New York	New York	Married-couple families	3,288,930	1.6%	1.1%	3.1%	3.7%	6.5%	11.9%	12.2%	21.1%
19	New York	New York	Nonfamily households	3,036,076	11.6%	7.5%	10.3%	8.9%	11.5%	15.7%	10.5%	11.4%
20	Bronx County	New York	Households	533,035	14.4%	8.2%	9.7%	8.8%	12.0%	15.5%	10.6%	11.0%
21	Bronx County	New York	Families	329,608	10.4%	5.0%	9.0%	9.1%	12.8%	17.1%	11.3%	12.9%
22	Bronx County	New York	Married-couple families	139,814	3.4%	2.2%	6.2%	6.6%	11.1%	16.6%	14.0%	19.6%
23	Bronx County	New York	Nonfamily households	203,427	22.0%	14.2%	11.9%	8.9%	11.4%	12.8%	8.4%	6.7%
24	Kings County	New York	Households	1,026,361	8.1%	5.2%	7.5%	6.8%	9.1%	13.7%	11.7%	14.9%
25	Kings County	New York	Families	584,460	5.9%	2.9%	6.8%	6.6%	9.3%	14.3%	12.3%	15.7%
26	Kings County	New York	Married-couple families	361,735	2.8%	1.8%	5.9%	6.0%	7.8%	12.1%	11.7%	17.0%
27	Kings County	New York	Nonfamily households	441,901	12.0%	8.6%	9.1%	7.4%	9.4%	12.5%	10.9%	13.2%
28	New York County	New York	Households	803,844	9.1%	5.3%	6.1%	4.5%	7.1%	10.3%	8.9%	13.2%
29	New York County	New York	Families	322,646	5.8%	3.2%	4.6%	5.1%	6.3%	8.7%	7.8%	11.8%
30	New York County	New York	Married-couple families	217,775	1.6%	1.1%	2.9%	3.1%	4.3%	6.5%	6.3%	11.8%
31	New York County	New York	Nonfamily households	481,198	11.5%	6.9%	7.1%	4.2%	7.5%	11.5%	9.7%	13.9%
32	Queens County	New York	Households	839,853	5.9%	3.7%	6.3%	6.8%	8.5%	16.0%	12.0%	18.3%
33	Queens County	New York	Families	556,133	3.6%	2.2%	5.4%	6.6%	8.8%	15.7%	12.8%	19.1%
34	Queens County	New York	Married-couple families	364,485	2.3%	2.1%	4.1%	5.3%	7.9%	14.6%	12.0%	20.1%
35	Queens County	New York	Nonfamily households	283,720	11.1%	6.7%	8.9%	8.5%	8.9%	17.8%	11.4%	14.0%
36	Richmond County	New York	Households	169,946	6.1%	3.0%	5.7%	6.3%	7.6%	12.5%	11.6%	19.5%

37	Richmond County	New York	Families	124,667	3.8%	1.6%	3.9%	4.2%	6.8%	12.1%	12.0%	21.8%
38	Richmond County	New York	Married-couple families	91,618	2.1%	0.9%	3.5%	4.0%	4.9%	10.5%	11.0%	22.8%
39	Richmond County	New York	Nonfamily households	45,279	13.4%	6.7%	11.0%	13.0%	11.0%	15.1%	9.4%	13.1%

In [81]:
```
#extracting Georgia and Chatham County
ga_income = top_3_income.iloc[0:8]

#viewing just total population in each Category
ga_income.iloc[:,0:4]
```

Out[81]:

	County	State	Category	Total
0	Georgia	Georgia	Households	4,092,467
1	Georgia	Georgia	Families	2,733,234
2	Georgia	Georgia	Married-couple families	1,917,471
3	Georgia	Georgia	Nonfamily households	1,359,233
4	Chatham County	Georgia	Households	121,527
5	Chatham County	Georgia	Families	70,429
6	Chatham County	Georgia	Married-couple families	48,070
7	Chatham County	Georgia	Nonfamily households	51,098

In [82]:
```
#extracting last 2 columns
ga_med_mean = ga_income.iloc[:,14:16]
```

In [83]:
```
#stripping percent symbol
ga_income = ga_income.replace("%","", regex=True)

#capturing just the percent columns
ga_income_range = ga_income.iloc[:,5:14]
```

In [84]:
```
#converting object to float
ga_income_range = ga_income_range.astype(float)
```

In [85]:
```
#viewing column names
ga_income_range.columns
```

Out[85]:
```
Index(['$10,000 to $14,999', '$15,000 to $24,999', '$25,000 to $34,999',
       '$35,000 to $49,999', '$50,000 to $74,999', '$75,000 to $99,999',
       '$100,000 to $149,999', '$150,000 to $199,999', '$200,000 or more'],
      dtype='object')
```

In [86]:
```
#combining into new categories
ga_budget = ga_income_range["$10,000 to $14,999"] + ga_income_range["$15,000 to $24,999"
ga_main = ga_income_range["$50,000 to $74,999"] + ga_income_range["$75,000 to $99,999"]
ga_premium = ga_income_range["$150,000 to $199,999"] + ga_income_range["$200,000 or more
```

In [87]:
```
#extracting first 3 columns
first_3 = ga_income.iloc[:,0:3]

#creating 3 new columns
first_3["Budget/Low Income"] = ga_budget
```

```
first_3["Mainstream/Middle Income"] = ga_main
first_3["Premium/High Income"] = ga_premium

#merging last 2 columns
combined_ga = pd.merge(first_3, ga_med_mean, left_index=True, right_index=True, how="lef

combined_ga
```

Out[87]:

	County	State	Category	Budget/Low Income	Mainstream/Middle Income	Premium/High Income	Median income (dollars)	Mean income (dollars)
0	Georgia	Georgia	Households	29.0	47.0	18.4	72,837	99,863
1	Georgia	Georgia	Families	22.4	50.3	23.5	86,642	116,323
2	Georgia	Georgia	Married-couple families	15.2	52.7	30.6	105,880	137,787
3	Georgia	Georgia	Nonfamily households	44.6	38.3	6.8	44,656	61,965
4	Chatham County	Georgia	Households	34.9	45.2	14.1	64,157	87,053
5	Chatham County	Georgia	Families	26.6	49.0	20.6	79,961	106,853
6	Chatham County	Georgia	Married-couple families	15.8	55.1	27.2	98,174	N
7	Chatham County	Georgia	Nonfamily households	48.4	38.3	5.0	43,452	56,886

Notes:

As I can see Georgia follows the companies data almost exactly.

- Mainstream/Middle Income is the largest group, followed by Budget/Low Income, and in last Premium/High Income.
- Families also have a higher income amongst all other segments. Which works well since the companies largest membership group is families.

This data correlates with the findings in part 1. I will continue with the next 2 states.

In [88]:
```
#seperating New Jersey and Union County
nj_income = top_3_income.iloc[8:16]

#extracting last 2 columns
nj_med_mean = nj_income.iloc[:,14:16]

#stripping percent symbol
nj_income = nj_income.replace("%","", regex=True)

#capturing just the percent columns
nj_income_range = nj_income.iloc[:,5:14]
```

In [89]:
```
#converting object to float
nj_income_range = nj_income_range.astype(float)
```

In [90]:
```
#combining into new categories
nj_budget = nj_income_range["$10,000 to $14,999"] + nj_income_range["$15,000 to $24,999"
nj_main = nj_income_range["$50,000 to $74,999"] + nj_income_range["$75,000 to $99,999"]
nj_premium = nj_income_range["$150,000 to $199,999"] + nj_income_range["$200,000 or more
```

In [91]:
```
#extracting first 3 columns
first_3_nj = nj_income.iloc[:,0:3]

#creating 3 new columns
first_3_nj["Budget/Low Income"] = nj_budget
first_3_nj["Mainstream/Middle Income"] = nj_main
first_3_nj["Premium/High Income"] = nj_premium

#merging last 2 columns
combined_nj = pd.merge(first_3_nj, nj_med_mean, left_index=True, right_index=True, how="

combined_nj
```

Out[91]:

	County	State	Category	Budget/Low Income	Mainstream/Middle Income	Premium/High Income	Median income (dollars)	Mean income (dollars)
8	New Jersey	New Jersey	Households	21.7	43.4	30.4	96,346	134,191
9	New Jersey	New Jersey	Families	15.8	43.9	38.1	117,988	157,601
10	New Jersey	New Jersey	Married-couple families	10.1	42.1	46.6	140,500	181,939
11	New Jersey	New Jersey	Nonfamily households	36.5	41.9	12.1	54,589	78,512
12	Union County	New Jersey	Households	22.9	42.5	32.1	98,028	145,267
13	Union County	New Jersey	Families	18.0	41.3	39.3	116,775	166,679
14	Union County	New Jersey	Married-couple families	11.7	38.5	48.6	144,966	N
15	Union County	New Jersey	Nonfamily households	41.6	43.1	10.0	53,646	75,150

Notes:

According to the data NJ has a higher population of Premium/High Income earners 6 out of 8 categories over Budget members. Families are again the strongest income class aswell as Mainstream/Middle Income being the largest percentage of all 3 income groups.

In [92]:
```
#seperating New York and all counties
ny_income = top_3_income.iloc[16:]

#extracting last 2 columns
ny_med_mean = ny_income.iloc[:,14:16]

#stripping percent symbol
ny_income = ny_income.replace("%","", regex=True)
```

```
#capturing just the percent columns
ny_income_range = ny_income.iloc[:,5:14]
```

In [93]:
```
#converting object to float
ny_income_range = ny_income_range.astype(float)
```

In [94]:
```
#combining into new categories
ny_budget = ny_income_range["$10,000 to $14,999"] + ny_income_range["$15,000 to $24,999"
ny_main = ny_income_range["$50,000 to $74,999"] + ny_income_range["$75,000 to $99,999"]
ny_premium = ny_income_range["$150,000 to $199,999"] + ny_income_range["$200,000 or more
```

In [95]:
```
#extracting first 3 columns
first_3_ny = ny_income.iloc[:,0:3]

#creating 3 new columns
first_3_ny["Budget/Low Income"] = ny_budget
first_3_ny["Mainstream/Middle Income"] = ny_main
first_3_ny["Premium/High Income"] = ny_premium

#merging last 2 columns
combined_ny = pd.merge(first_3_ny, ny_med_mean, left_index=True, right_index=True, how="

combined_ny
```

Out[95]:

	County	State	Category	Budget/Low Income	Mainstream/Middle Income	Premium/High Income	Median income (dollars)	Mean income (dollars)
16	New York	New York	Households	26.6	42.3	24.4	79,557	119,130
17	New York	New York	Families	20.3	44.9	30.5	99,066	141,334
18	New York	New York	Married-couple families	14.4	45.2	38.7	121,320	168,776
19	New York	New York	Nonfamily households	38.2	37.6	12.6	50,181	78,600
20	Bronx County	New York	Households	38.7	37.1	9.8	45,517	66,878
21	Bronx County	New York	Families	35.9	41.3	12.5	54,583	75,943
22	Bronx County	New York	Married-couple families	26.1	50.2	20.3	83,256	102,243
23	Bronx County	New York	Nonfamily households	46.4	27.9	3.7	27,038	45,071
24	Kings County	New York	Households	28.6	40.3	23.0	73,951	115,625
25	Kings County	New York	Families	25.6	42.3	26.1	82,936	131,096
26	Kings County	New York	Married-couple families	21.5	40.8	35.0	104,812	162,893
27	Kings County	New York	Nonfamily households	34.5	36.6	17.0	56,244	88,830

28	New York County	New York	Households	23.0	32.4	35.5	95,866	175,743
29	New York County	New York	Families	19.2	28.3	46.8	133,880	251,514
30	New York County	New York	Married-couple families	11.4	24.6	62.3	205,490	330,393
31	New York County	New York	Nonfamily households	25.7	35.1	27.6	78,193	123,291
32	Queens County	New York	Households	25.3	46.3	22.4	80,557	106,667
33	Queens County	New York	Families	23.0	47.6	25.9	89,732	116,575
34	Queens County	New York	Married-couple families	19.4	46.7	31.7	104,208	131,613
35	Queens County	New York	Nonfamily households	33.0	43.2	12.7	57,749	78,159
36	Richmond County	New York	Households	22.6	43.6	27.9	93,164	119,550
37	Richmond County	New York	Families	16.5	45.9	33.8	110,820	137,058
38	Richmond County	New York	Married-couple families	13.3	44.3	40.1	126,008	N
39	Richmond County	New York	Nonfamily households	41.7	37.6	7.3	44,540	62,115

Notes:

NY is interesting due to the diversity of each county. Each county varies on what is 2nd or 3rd place amongst income groups. Overall Mainstream/Middle income is still number 1. Married couples and families still exceed non family and household income.

What I can conclude from the top 3 states exactly matches the companies data. To look at areas where mainstream/middle income are the dominate income group and families and married couples.

Housing Data:

In [96]:
```
top_3_housing
```

Out[96]:

	County	State	Label	Total housing units	Occupied housing units	Vacant housing units	Homeowner vacancy rate	Rental vacancy rate	Total housing units2	1-unit, detached	...	i t
0	Georgia	Georgia	Estimate	4,539,156	4,092,467	446,689	1.1	6.1	4,539,156	3,017,661	...	1,(
1	Georgia	Georgia	Percent	4,539,156	90.20%	9.80%	(X)	(X)	4,539,156	66.50%	...	
2	Chatham County	Georgia	Estimate	137,606	121,527	16,079	1.5	6.5	137,606	83,799	...	

3	Chatham County	Georgia	Percent	137,606	88.30%	11.70%	(X)	(X)	137,606	60.90%	...
4	New Jersey	New Jersey	Estimate	3,785,097	3,516,978	268,119	0.7	3	3,785,097	2,007,394	...
5	New Jersey	New Jersey	Percent	3,785,097	92.90%	7.10%	(X)	(X)	3,785,097	53.00%	...
6	Union County	New Jersey	Estimate	211,906	202,575	9,331	1.5	3.5	211,906	103,842	...
7	Union County	New Jersey	Percent	211,906	95.60%	4.40%	(X)	(X)	211,906	49.00%	...
8	New York	New York	Estimate	8,585,784	7,774,308	811,476	1	3.5	8,585,784	3,528,524	...
9	New York	New York	Percent	8,585,784	90.50%	9.50%	(X)	(X)	8,585,784	41.10%	...
10	Bronx County	New York	Estimate	557,985	533,035	24,950	1.1	2.5	557,985	36,082	...
11	Bronx County	New York	Percent	557,985	95.50%	4.50%	(X)	(X)	557,985	6.50%	...
12	Kings County	New York	Estimate	1,101,429	1,026,361	75,068	1.4	2.5	1,101,429	49,541	...
13	Kings County	New York	Percent	1,101,429	93.20%	6.80%	(X)	(X)	1,101,429	4.50%	...
14	New York County	New York	Estimate	923,239	803,844	119,395	3.6	4.3	923,239	11,268	...
15	New York County	New York	Percent	923,239	87.10%	12.90%	(X)	(X)	923,239	1.20%	...
16	Queens County	New York	Estimate	911,913	839,853	72,060	0.9	3	911,913	177,198	...
17	Queens County	New York	Percent	911,913	92.10%	7.90%	(X)	(X)	911,913	19.40%	...
18	Richmond County	New York	Estimate	184,497	169,946	14,551	1.2	5.3	184,497	65,328	...
19	Richmond County	New York	Percent	184,497	92.10%	7.90%	(X)	(X)	184,497	35.40%	...

20 rows × 35 columns

In [97]:
```
top_3_housing.columns
```

Out[97]:
```
Index(['County', 'State', 'Label', 'Total housing units',
       'Occupied housing units', 'Vacant housing units',
       'Homeowner vacancy rate', 'Rental vacancy rate', 'Total housing units2',
       '1-unit, detached', '1-unit, attached', '2 units', '3 or 4 units',
       '5 to 9 units', '10 to 19 units', '20 or more units', 'Mobile home',
       'Boat, RV, van, etc.', 'Occupied housing units3', 'Owner-occupied',
       'Renter-occupied', 'Average household size of owner-occupied unit',
       'Average household size of renter-occupied unit',
       'Occupied housing units4', 'Moved in 2021 or later',
       'Moved in 2018 to 2020', 'Moved in 2010 to 2017',
       'Moved in 2000 to 2009', 'Moved in 1990 to 1999',
       'Moved in 1989 and earlier', 'Occupied housing units5',
       'No vehicles available', '1 vehicle available', '2 vehicles available',
```

```
       '3 or more vehicles available'],
      dtype='object')
```

In [98]:

```
#splitting into estimated numbers and percents
housing_estimate = top_3_housing.iloc[[0,2,4,6,8,10,12,14,16,18]]
housing_pct = top_3_housing.iloc[[1,3,5,7,9,11,13,15,17,19]]

#splitting into more easily readable chunks of data

#county state
housing_pct_2 = housing_pct.iloc[:,0:2]

#housing types
housing_type = housing_pct.iloc[:,10:18]

#renter and owner
pct_occupied = housing_pct.iloc[:,19:21]

#car and when moved
pct_moved_car = housing_pct.iloc[:,24:]

# combining county and state with each segment
combined_housing = pd.merge(housing_pct_2, housing_type, left_index=True, right_index=Tr

combined_or = pd.merge(housing_pct_2, pct_occupied, left_index=True, right_index=True, h

combined_moved_car = pd.merge(housing_pct_2, pct_moved_car, left_index=True, right_index
```

In [99]:

```
combined_housing
```

Out[99]:

	County	State	1-unit, attached	2 units	3 or 4 units	5 to 9 units	10 to 19 units	20 or more units	Mobile home	Boat, RV, van, etc.
1	Georgia	Georgia	4.50%	2.20%	3.10%	4.30%	4.10%	7.30%	8.00%	0.10%
3	Chatham County	Georgia	6.00%	5.40%	5.90%	6.50%	4.80%	7.80%	2.60%	0.00%
5	New Jersey	New Jersey	10.30%	8.30%	5.90%	4.60%	4.80%	12.10%	1.00%	0.00%
7	Union County	New Jersey	5.60%	16.80%	7.30%	3.10%	4.20%	13.70%	0.30%	0.00%
9	New York	New York	5.40%	9.60%	6.90%	5.20%	4.40%	25.20%	2.10%	0.00%
11	Bronx County	New York	5.90%	7.90%	8.20%	5.10%	6.70%	59.40%	0.20%	0.10%
13	Kings County	New York	8.20%	16.40%	15.60%	10.20%	6.50%	38.40%	0.20%	0.00%
15	New York County	New York	0.60%	0.90%	2.00%	5.30%	10.60%	79.30%	0.00%	0.00%
17	Queens County	New York	9.00%	19.10%	9.90%	5.70%	3.90%	32.60%	0.30%	0.00%
19	Richmond County	New York	25.30%	21.90%	3.70%	2.50%	1.60%	9.30%	0.30%	0.00%

In [100…

```
#stripping the Percent symbol
combined_housing = combined_housing.replace("%","", regex=True)
```

In [107…

```
#converting columns to float
```

```
combined_housing[combined_housing.columns[2:]] = combined_housing[combined_housing.colum
```

In [108…

```
#calculating amounts to see which is the most popular housing type
combined_housing[["1-unit, attached","2 units","3 or 4 units", "5 to 9 units", "10 to 19
```

Out[108]:

```
1-unit, attached      80.8
2 units              108.5
3 or 4 units          68.5
5 to 9 units          52.5
10 to 19 units        51.6
20 or more units     285.1
dtype: float64
```

In [109…

```
combined_or
```

Out[109]:

	County	State	Owner-occupied	Renter-occupied
1	Georgia	Georgia	65.90%	34.10%
3	Chatham County	Georgia	56.40%	43.60%
5	New Jersey	New Jersey	64.60%	35.40%
7	Union County	New Jersey	57.00%	43.00%
9	New York	New York	54.10%	45.90%
11	Bronx County	New York	21.20%	78.80%
13	Kings County	New York	29.50%	70.50%
15	New York County	New York	24.30%	75.70%
17	Queens County	New York	44.60%	55.40%
19	Richmond County	New York	68.70%	31.30%

In [110…

```
combined_moved_car
```

Out[110]:

	County	State	Moved in 2021 or later	Moved in 2018 to 2020	Moved in 2010 to 2017	Moved in 2000 to 2009	Moved in 1990 to 1999	Moved in 1989 and earlier	Occupied housing units5	No vehicles available	1 vehicle available	vehicl availab
1	Georgia	Georgia	18.80%	24.70%	24.00%	16.20%	8.80%	7.40%	4,092,467	5.70%	32.20%	38.40
3	Chatham County	Georgia	26.70%	26.80%	19.80%	12.70%	6.60%	7.40%	121,527	5.90%	36.20%	43.90
5	New Jersey	New Jersey	14.40%	21.40%	24.10%	17.90%	10.90%	11.30%	3,516,978	10.90%	35.40%	36.10
7	Union County	New Jersey	14.00%	22.50%	23.30%	17.30%	9.90%	13.00%	202,575	10.40%	36.70%	35.60
9	New York	New York	14.90%	18.80%	24.00%	17.50%	11.40%	13.40%	7,774,308	29.10%	33.90%	25.60
11	Bronx County	New York	10.10%	18.80%	29.70%	20.20%	10.70%	10.40%	533,035	61.10%	29.00%	8.50
13	Kings County	New York	17.80%	19.30%	25.00%	17.20%	10.40%	10.20%	1,026,361	55.20%	36.20%	7.00
15	New York County	New York	25.20%	18.80%	19.00%	13.20%	10.70%	13.10%	803,844	78.20%	19.10%	2.10
17	Queens	New	13.80%	17.90%	25.90%	18.30%	11.60%	12.50%	839,853	36.80%	41.00%	16.60

	County	York									
19	Richmond County	New York	9.50%	17.00%	26.00%	20.80%	13.10%	13.60%	169,946	15.50%	38.50%	33.90

In [105…

```
#nyc moved in rate for all 5 counties
(10.10 + 17.80 + 25.20 + 13.8 + 9.5) / 5
```

Out[105]:

```
15.279999999999998
```

Notes:

The trend between the housing types between the 3 states seem to be multiple unit homes. Leaning toward large condo buildings and multi familiy homes.

Regarding homeownership rates 2 of the states favor homeownership. New York city completely leans toward renters instead of homeowners.

The moved in rate has been decreasing between all areas from 2020 to 2021. The only county with positive growth was Manhattan New York.

Georgia has increased over its historical trend by large percentages.

The lowest moved in rate that should be considered is roughly 14%. This would be a very nice growing pace for a county/state.

66% have access to a car. 54% Own a home

Also I can see that vehicle ownership is high in NJ and Georgia while lower in NY. This makes sense since New York city is a densly populated city with a large population and robust transportation system.

Household data:

In [106…

```
top_3_household
```

Out[106]:

	County	State	Label	Total households	Married-couple household	With children of the householder under 18 years	Cohabiting couple household	With children of the householder under 18 years2	Male householder, no spouse/partner present
0	Georgia	Georgia	Estimate	4,092,467	1,917,471	739,349	246,640	86,380	696,443
1	Georgia	Georgia	Percent	4,092,467	46.90%	18.10%	6.00%	2.10%	17.00%
2	Chatham County	Georgia	Estimate	121,527	48,070	14,486	9,786	2,626	21,749
3	Chatham County	Georgia	Percent	121,527	39.60%	11.90%	8.10%	2.20%	17.90%
4	New Jersey	New Jersey	Estimate	3,516,978	1,753,523	707,557	245,942	82,379	565,098
5	New Jersey	New Jersey	Percent	3,516,978	49.90%	20.10%	7.00%	2.30%	16.10%
6	Union County	New Jersey	Estimate	202,575	103,296	44,338	19,179	8,716	29,267

7	Union County	New Jersey	Percent	202,575	51.00%	21.90%	9.50%	4.30%	14.40%
8	New York	New York	Estimate	7,774,308	3,288,930	1,242,696	586,369	163,494	1,471,496
9	New York	New York	Percent	7,774,308	42.30%	16.00%	7.50%	2.10%	18.90%
10	Bronx County	New York	Estimate	533,035	139,814	58,055	36,140	17,698	122,094
11	Bronx County	New York	Percent	533,035	26.20%	10.90%	6.80%	3.30%	22.90%
12	Kings County	New York	Estimate	1,026,361	361,735	148,223	85,281	20,049	199,765
13	Kings County	New York	Percent	1,026,361	35.20%	14.40%	8.30%	2.00%	19.50%
14	New York County	New York	Estimate	803,844	217,775	82,160	54,842	4,902	205,225
15	New York County	New York	Percent	803,844	27.10%	10.20%	6.80%	0.60%	25.50%
16	Queens County	New York	Estimate	839,853	364,485	137,423	50,605	14,936	161,991
17	Queens County	New York	Percent	839,853	43.40%	16.40%	6.00%	1.80%	19.30%
18	Richmond County	New York	Estimate	169,946	91,618	40,057	8,931	3,519	23,199
19	Richmond County	New York	Percent	169,946	53.90%	23.60%	5.30%	2.10%	13.70%

20 rows × 39 columns

In [112…

```
top_3_household.columns
```

Out[112]:

```
Index(['County', 'State', 'Label', 'Total households',
       'Married-couple household',
       'With children of the householder under 18 years',
       'Cohabiting couple household',
       'With children of the householder under 18 years2',
       'Male householder, no spouse/partner present',
       'With children of the householder under 18 years3',
       'Householder living alone', '65 years and over',
       'Female householder, no spouse/partner present',
       'With children of the householder under 18 years4',
       'Householder living alone5', '65 years and over6',
       'Households with one or more people under 18 years',
       'Households with one or more people 65 years and over',
       'Average household size', 'Average family size',
       'Population in households', 'Householder', 'Spouse',
       'Unmarried partner', 'Child', 'Other relatives', 'Other nonrelatives',
       'Males 15 years and over', 'Never married',
       'Now married, except separated', 'Separated', 'Widowed', 'Divorced',
       'Females 15 years and over', 'Never married7',
       'Now married, except separated8', 'Separated9', 'Widowed10',
       'Divorced11'],
      dtype='object')
```

Notes:

Since family types are our main customer base we will isolate some columns to view.

- 'Married-couple household', 'With children of the householder under 18 years',
- 'Cohabiting couple household','With children of the householder under 18 years2',
- 'Average household size', 'Average family size','Population in households'

In [129…

```
#dropping unwanted columns and making a new variable
top_3_hh = top_3_household[['County', 'State','Label','Married-couple household', 'With
'Cohabiting couple household','With children of the householder under 18 years2',
'Average household size', 'Average family size','Population in households']]
```

In [133…

```
#extracting state seperately

#extracting georgia

ga_head = top_3_hh.iloc[:4,:]

#extracting new jersey

nj_head = top_3_hh.iloc[4:8,:]

#extracting new york
ny_head = top_3_hh.iloc[8:,:]
```

In [138…

```
ga_head
```

Out[138]:

	County	State	Label	Married-couple household	With children of the householder under 18 years	Cohabiting couple household	With children of the householder under 18 years2	Average household size	Average family size	Populatic householc
0	Georgia	Georgia	Estimate	1,917,471	739,349	246,640	86,380	2.61	3.19	10,662,5
1	Georgia	Georgia	Percent	46.90%	18.10%	6.00%	2.10%	(X)	(X)	10,662,5
2	Chatham County	Georgia	Estimate	48,070	14,486	9,786	2,626	2.37	3.05	287,9
3	Chatham County	Georgia	Percent	39.60%	11.90%	8.10%	2.20%	(X)	(X)	287,9

In [136…

```
nj_head
```

Out[136]:

	County	State	Label	Married-couple household	With children of the householder under 18 years	Cohabiting couple household	With children of the householder under 18 years2	Average household size	Average family size	Population in households
4	New Jersey	New Jersey	Estimate	1,753,523	707,557	245,942	82,379	2.59	3.16	9,092,231
5	New Jersey	New Jersey	Percent	49.90%	20.10%	7.00%	2.30%	(X)	(X)	9,092,231
6	Union County	New Jersey	Estimate	103,296	44,338	19,179	8,716	2.79	3.26	564,759
7	Union County	New Jersey	Percent	51.00%	21.90%	9.50%	4.30%	(X)	(X)	564,759

In [139…
```
ny_head
```

Out[139]:

	County	State	Label	Married-couple household	With children of the householder under 18 years	Cohabiting couple household	With children of the householder under 18 years2	Average household size	Average family size	Populatic househol
8	New York	New York	Estimate	3,288,930	1,242,696	586,369	163,494	2.45	3.12	19,062,96
9	New York	New York	Percent	42.30%	16.00%	7.50%	2.10%	(X)	(X)	19,062,96
10	Bronx County	New York	Estimate	139,814	58,055	36,140	17,698	2.5	3.2	1,332,23
11	Bronx County	New York	Percent	26.20%	10.90%	6.80%	3.30%	(X)	(X)	1,332,23
12	Kings County	New York	Estimate	361,735	148,223	85,281	20,049	2.48	3.25	2,546,47
13	Kings County	New York	Percent	35.20%	14.40%	8.30%	2.00%	(X)	(X)	2,546,47
14	New York County	New York	Estimate	217,775	82,160	54,842	4,902	1.89	2.83	1,522,38
15	New York County	New York	Percent	27.10%	10.20%	6.80%	0.60%	(X)	(X)	1,522,38
16	Queens County	New York	Estimate	364,485	137,423	50,605	14,936	2.67	3.26	2,245,54
17	Queens County	New York	Percent	43.40%	16.40%	6.00%	1.80%	(X)	(X)	2,245,54
18	Richmond County	New York	Estimate	91,618	40,057	8,931	3,519	2.85	3.36	483,92
19	Richmond County	New York	Percent	53.90%	23.60%	5.30%	2.10%	(X)	(X)	483,92

Notes: all of the top 3 states seem roughly similar so I think the best method is to use an average to create a household profile.

In [162…
```
#extracting number columns
top_3_hh_numbers = top_3_hh.iloc[[0,2,4,6,8,10,12,14,16,18]]

#removing comma
top_3_hh_numbers = top_3_hh_numbers.replace(",","", regex=True)
```

In [172…
```
#converting from object to float and running the mean

top_3_hh_numbers.iloc[:,7:9].astype(float).mean(numeric_only=True)
```

Out[172]:
```
Average household size     2.520
Average family size        3.168
dtype: float64
```

In [173…
```
top_3_pct = top_3_hh.iloc[[1,3,5,7,9,11,13,15,17,19]]

top_3_pct = top_3_pct.replace("%","", regex=True)
```

In [176…

```
#pulling average
top_3_pct["Married-couple household"].astype(float).mean()
```

Out[176]:

```
41.55
```

In [178…

```
#average
top_3_pct["With children of the householder under 18 years"].astype(float).mean()
```

Out[178]:

```
16.35
```

In [179…

```
#Avg
top_3_pct["Cohabiting couple household"].astype(float).mean()
```

Out[179]:

```
7.13
```

In [181…

```
top_3_pct["With children of the householder under 18 years2"].astype(float).mean()
```

Out[181]:

```
2.2800000000000002
```

- Average household size 2.5
- Average family size 3.2
- Average Married Couple Houshold Pct 41.6%
- Average With children of the householder under 18 years Pct 16.6%
- Average Cohabiting couple household Pct 7.1%
- Average With children of the householder under 18 years2 Pct 2.3%

Top 3 States Profile:

- Diversity among races
- Large Middle Aged population followed by young.
- Mainstream/middle income are the dominate income group and families and married couples.
- 2021 Moved in rate that should be considered is roughly 14%
- Large population of homeownership
- Large population of vehicle ownership
- Average household size 2.5
- Average family size 3.2
- Average Married Couple Houshold Pct 41.6%
- Average With children of the householder under 18 years Pct 16.6%
- Average Cohabiting couple household Pct 7.1%
- Average With children of the householder under 18 years2 Pct 2.3%

This seems to be the most profitable recipe besides just population size. Time to load in 6 potential states data from the census to see which state best fits this profile

Potential 6 states:

In [569…

```
#loading in the data for the 6 states.
six_household = pd.read_csv("Household Census.csv")
six_housing  = pd.read_csv("Housing Census.csv")
six_race = pd.read_csv("Race Census.csv")
six_age = pd.read_csv("Age Census.csv")
six_income = pd.read_csv("Income Census.csv")
```

In [477…

```
#starting in same order I will begin with race

six_race
```

Out[477]:

	County	State	Total:	Hispanic or Latino	Not Hispanic or Latino:	Population of one race:	White alone	Black or African American alone	American Indian and Alaska Native alone	Asian alone
0	Connecticut	Connecticut	3,605,944	623,293	2,982,651	2,845,082	2,279,232	360,937	6,404	170,459
1	Fairfield County	Connecticut	957,419	205,351	752,068	715,131	552,125	99,992	858	50,751
2	Hartford County	Connecticut	899,498	166,275	733,223	701,581	523,105	118,154	1,166	53,325
3	Litchfield County	Connecticut	185,186	14,580	170,606	163,254	155,601	2,957	268	3,434
4	Middlesex County	Connecticut	164,245	11,928	152,317	145,879	131,954	8,001	214	4,923
...	...	...	...	...	...	...	...	...	...	...
370	Webster County	West Virginia	8,378	50	8,328	8,123	8,086	10	14	4
371	Wetzel County	West Virginia	14,442	148	14,294	13,817	13,704	34	18	50
372	Wirt County	West Virginia	5,194	22	5,172	5,021	4,999	6	4	0
373	Wood County	West Virginia	84,296	1,174	83,122	79,672	77,718	1,016	148	564
374	Wyoming County	West Virginia	21,382	170	21,212	20,708	20,539	107	30	11

375 rows × 13 columns

In [478…

```
#seperating states
wv_race = six_race[six_race["County"] == "West Virginia"]
oh_race = six_race[six_race["County"] == "Ohio"]
ky_race = six_race[six_race["County"] == "Kentucky"]
tn_race = six_race[six_race["County"] == "Tennessee"]
de_race = six_race[six_race["County"] == "Delaware"]
ct_race = six_race[six_race["County"] == "Connecticut"]
```

In [479…

```
wv_race
```

Out[479]:

	County	State	Total:	Hispanic or Latino	Not Hispanic or Latino:	Population of one race:	White alone	Black or African American alone	American Indian and Alaska Native alone	Asian alone	Nativ Hawaiia an Othe Pacifi Islande alon
319	West Virginia	West Virginia	1,793,716	34,827	1,758,889	1,686,754	1,598,834	64,749	3,187	14,903	42

In [480…

```
wv_race = wv_race.replace(",","", regex=True)
```

In [481…

```
wv_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
wv_race.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1 entries, 319 to 319
Data columns (total 13 columns):
 #   Column                                             Non-Null Count  Dtype
---  ------                                             --------------  -----
 0   County                                             1 non-null      object
 1   State                                              1 non-null      object
 2   Total:                                             1 non-null      int32
 3   Hispanic or Latino                                 1 non-null      int32
 4   Not Hispanic or Latino:                            1 non-null      int32
 5   Population of one race:                            1 non-null      object
 6   White alone                                        1 non-null      int32
 7   Black or African American alone                    1 non-null      int32
 8   American Indian and Alaska Native alone            1 non-null      object
 9   Asian alone                                        1 non-null      int32
 10  Native Hawaiian and Other Pacific Islander alone   1 non-null      object
 11  Some Other Race alone                              1 non-null      object
 12  Population of two or more races:                   1 non-null      object
dtypes: int32(6), object(7)
memory usage: 88.0+ bytes
```

In [482…

```
#calculating the percentage of diversity in West Virginia
wv_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
```

Out[482]:

	Hispanic or Latino	Not Hispanic or Latino:	White alone	Black or African American alone	Asian alone
319	1.941612	98.058388	89.135292	3.609769	0.830845

In [483…

```
#removing comma
oh_race = oh_race.replace(",","", regex=True)
#converting columns to integer
oh_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
```

In [484…

```
#calculating the percentage of diversity in Ohio
oh_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
```

Out[484]:

	Hispanic or Latino	Not Hispanic or Latino:	White alone	Black or African American alone	Asian alone
134	4.418071	95.581929	75.88605	12.349561	2.513711

In [485…

```
#removing comma
ky_race = ky_race.replace(",","", regex=True)
#converting columns to integer
ky_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
```

In [486…

```
#calculating the percentage of diversity in Kentucky
ky_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
```

Out[486]:

	Hispanic or Latino	Not Hispanic or Latino:	White alone	Black or African American alone	Asian alone
13	4.612995	95.387005	81.333719	7.940014	1.63883

In [487…

```
#removing comma
tn_race = tn_race.replace(",","", regex=True)
#converting columns to integer
tn_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
```

In [488…

```
#calculating the percentage of diversity in Tenessee
tn_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
```

Out[488]:

	Hispanic or Latino	Not Hispanic or Latino:	White alone	Black or African American alone	Asian alone
223	6.933846	93.066154	70.906663	15.682204	1.943353

In [489…

```
#removing comma
de_race = de_race.replace(",","", regex=True)
#converting columns to integer
de_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
```

In [490…

```
#calculating the percentage of diversity in Delaware
de_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
```

Out[490]:

	Hispanic or Latino	Not Hispanic or Latino:	White alone	Black or African American alone	Asian alone
9	10.534897	89.465103	58.573885	21.512241	4.282851

In [491…

```
#removing comma
ct_race = ct_race.replace(",","", regex=True)
#converting columns to integer
ct_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
```

In [492…

```
#calculating the percentage of diversity in Delaware
ct_race[["Hispanic or Latino","Not Hispanic or Latino:","White alone","Black or African
```

Out[492]:

	Hispanic or Latino	Not Hispanic or Latino:	White alone	Black or African American alone	Asian alone
0	17.285155	82.714845	63.207637	10.009501	4.727167

Notes:

Based on the diversity make up Delaware will come in with the best balance and than Connecticut.

Diversity Ranked:

1) Delaware 2) Connecticut 3) Tennessee 4) Ohio 5) Kentucky 6) West Virginia

Age demographics:

In [493…

```
six_age
```

Out[493]:

	County	State	Category Type	Total population	Under 5 years	5 to 9 years	10 to 14 years	15 to 19 years	20 to 24 years	25 to 29 years	...	Under 18 years
0	West Virginia	West Virginia	Total	1,775,156	87,469	96,217	102,819	111,760	114,325	102,223	...	351,543
1	West Virginia	West Virginia	Percent	(X)	4.90%	5.40%	5.80%	6.30%	6.40%	5.80%	...	19.80%
2	West Virginia	West Virginia	Male	882,101	43,466	49,137	52,066	57,593	56,408	55,461	...	177,564
3	West Virginia	West Virginia	Percent Male	(X)	4.90%	5.60%	5.90%	6.50%	6.40%	6.30%	...	20.10%

4	West Virginia	West Virginia	Female	893,055	44,003	47,080	50,753	54,167	57,917	46,762	...	173,979
...	...	...	...	...	...	...	...	...	...	...	...	...
577	Windham County	Connecticut	Percent	(X)	4.40%	6.10%	5.20%	6.70%	6.70%	6.70%	...	19.30%
578	Windham County	Connecticut	Male	57,917	2,755	3,517	3,574	3,846	3,878	3,927	...	11,974
579	Windham County	Connecticut	Percent Male	(X)	4.80%	6.10%	6.20%	6.60%	6.70%	6.80%	...	20.70%
580	Windham County	Connecticut	Female	58,501	2,363	3,569	2,485	3,966	3,934	3,816	...	10,534
581	Windham County	Connecticut	Percent Female	(X)	4.00%	6.10%	4.20%	6.80%	6.70%	6.50%	...	18.00%

582 rows × 34 columns

In [494…

```
#removing comma and percentage symbol
six_age = six_age.replace({",":"", "%":""}, regex=True)
```

In [495…

```
#seperating states
wv_age = six_age[six_age["County"] == "West Virginia"]
oh_age = six_age[six_age["County"] == "Ohio"]
ky_age = six_age[six_age["County"] == "Kentucky"]
tn_age = six_age[six_age["County"] == "Tennessee"]
de_age = six_age[six_age["County"] == "Delaware"]
ct_age = six_age[six_age["County"] == "Connecticut"]
```

In [496…

```
#isolating just the percent row
wv_age_pct = wv_age.iloc[1:2,:]

#isolating just the first 4 columns
wv_age_head = wv_age_pct.iloc[:,:3]

#converting slice into a dataframe
wv_head = pd.DataFrame(wv_age_head)

#isolating just the number columns
wv_age_num = wv_age_pct.iloc[:,4:22]

#converting object type to float
wv_age_float = wv_age_num.astype(float)
```

In [497…

```
#combining by age groups
wv_young = wv_age_float["20 to 24 years"] + wv_age_float["25 to 29 years"] + wv_age_floa

wv_mid = wv_age_float["40 to 44 years"] + wv_age_float["45 to 49 years"] + wv_age_float[

wv_older = wv_age_float["60 to 64 years"] + wv_age_float["65 to 69 years"] + wv_age_floa

#creating columns
wv_head["Young"] = wv_young
wv_head["Midage"] = wv_mid
wv_head["Older"] = wv_older

wv_head
```

Out[497]:

	County	State	Category Type	Young	Midage	Older

1	West Virginia	West Virginia	Percent	23.8	25.4	28.3

In [498…

```
#isolating just the percent row
oh_age_pct = oh_age.iloc[1:2,:]

#isolating just the first 4 columns
oh_age_head = oh_age_pct.iloc[:,:3]

#converting slice into a dataframe
oh_head = pd.DataFrame(oh_age_head)

#isolating just the number columns
oh_age_num = oh_age_pct.iloc[:,4:22]

#converting object type to float
oh_age_float = oh_age_num.astype(float)
```

In [499…

```
#combining by age groups
oh_young = oh_age_float["20 to 24 years"] + oh_age_float["25 to 29 years"] + oh_age_floa

oh_mid = oh_age_float["40 to 44 years"] + oh_age_float["45 to 49 years"] + oh_age_float[

oh_older = oh_age_float["60 to 64 years"] + oh_age_float["65 to 69 years"] + oh_age_floa

#creating columns
oh_head["Young"] = oh_young
oh_head["Midage"] = oh_mid
oh_head["Older"] = oh_older

oh_head
```

Out[499]:

	County	State	Category Type	Young	Midage	Older
181	Ohio	Ohio	Percent	25.8	24.6	25.3

In [500…

```
#isolating just the percent row
ky_age_pct = ky_age.iloc[1:2,:]

#isolating just the first 4 columns
ky_age_head = ky_age_pct.iloc[:,:3]

#converting slice into a dataframe
ky_head = pd.DataFrame(ky_age_head)

#isolating just the number columns
ky_age_num = ky_age_pct.iloc[:,4:22]

#converting object type to float
ky_age_float = ky_age_num.astype(float)
```

In [501…

```
#combining by age groups
ky_young = ky_age_float["20 to 24 years"] + ky_age_float["25 to 29 years"] + ky_age_floa

ky_mid = ky_age_float["40 to 44 years"] + ky_age_float["45 to 49 years"] + ky_age_float[

ky_older = ky_age_float["60 to 64 years"] + ky_age_float["65 to 69 years"] + ky_age_floa

#creating columns
ky_head["Young"] = ky_young
ky_head["Midage"] = ky_mid
ky_head["Older"] = ky_older

ky_head
```

Out[501]:

	County	State	Category Type	Young	Midage	Older
439	Kentucky	Kentucky	Percent	25.9	25.0	24.3

In [502…

```
#isolating just the percent row
tn_age_pct = tn_age.iloc[1:2,:]

#isolating just the first 4 columns
tn_age_head = tn_age_pct.iloc[:,:3]

#converting slice into a dataframe
tn_head = pd.DataFrame(tn_age_head)

#isolating just the number columns
tn_age_num = tn_age_pct.iloc[:,4:22]

#converting object type to float
tn_age_float = tn_age_num.astype(float)
```

In [503…

```
#combining by age groups
tn_young = tn_age_float["20 to 24 years"] + tn_age_float["25 to 29 years"] + tn_age_floa

tn_mid = tn_age_float["40 to 44 years"] + tn_age_float["45 to 49 years"] + tn_age_float[

tn_older = tn_age_float["60 to 64 years"] + tn_age_float["65 to 69 years"] + tn_age_floa

#creating columns
tn_head["Young"] = tn_young
tn_head["Midage"] = tn_mid
tn_head["Older"] = tn_older

tn_head
```

Out[503]:

	County	State	Category Type	Young	Midage	Older
49	Tennessee	Tennessee	Percent	26.9	25.1	23.9

In [504…

```
#isolating just the percent row
de_age_pct = de_age.iloc[1:2,:]

#isolating just the first 4 columns
de_age_head = de_age_pct.iloc[:,:3]

#converting slice into a dataframe
de_head = pd.DataFrame(de_age_head)

#isolating just the number columns
de_age_num = de_age_pct.iloc[:,4:22]

#converting object type to float
de_age_float = de_age_num.astype(float)
```

In [505…

```
#combining by age groups
de_young = de_age_float["20 to 24 years"] + de_age_float["25 to 29 years"] + de_age_floa

de_mid = de_age_float["40 to 44 years"] + de_age_float["45 to 49 years"] + de_age_float[

de_older = de_age_float["60 to 64 years"] + de_age_float["65 to 69 years"] + de_age_floa

#creating columns
de_head["Young"] = de_young
de_head["Midage"] = de_mid
```

```
de_head["Older"] = de_older

de_head
```

Out[505]:

	County	State	Category Type	Young	Midage	Older
415	Delaware	Delaware	Percent	24.7	23.7	28.6

In [506…

```
#isolating just the percent row
ct_age_pct = ct_age.iloc[1:2,:]

#isolating just the first 4 columns
ct_age_head = ct_age_pct.iloc[:,:3]

#converting slice into a dataframe
ct_head = pd.DataFrame(ct_age_head)

#isolating just the number columns
ct_age_num = ct_age_pct.iloc[:,4:22]

#converting object type to float
ct_age_float = ct_age_num.astype(float)
```

In [507…

```
#combining by age groups
ct_young = ct_age_float["20 to 24 years"] + ct_age_float["25 to 29 years"] + ct_age_floa

ct_mid = ct_age_float["40 to 44 years"] + ct_age_float["45 to 49 years"] + ct_age_float[

ct_olctr = ct_age_float["60 to 64 years"] + ct_age_float["65 to 69 years"] + ct_age_floa

#creating columns
ct_head["Young"] = ct_young
ct_head["Midage"] = ct_mid
ct_head["Olctr"] = ct_olctr

ct_head
```

Out[507]:

	County	State	Category Type	Young	Midage	Olctr
529	Connecticut	Connecticut	Percent	25.5	26.1	25.3

Notes:

Based on the age demographics make up Connecticut will come in with the best midage. Tennessee Would come in second if we are looking at midage and young together. As previously mentioned middle age was the largest segement and than young. Even though West Virginia comes in second for midage adults they have the lowest percentage of young adults.

Diversity Ranked by Midage:

1) Connecticut 2) Tennessee 3) West Virginia 4) Kentucky 5) Ohio 6) Delaware

Income demographics:

To recall the average makeup of the top 3 states were 23% Premium, 40.9% Mainstream, and 36% Budget. I would like to find a match as close as possible.

In [508…

```
#starting with income
```

```
six_income
```

Out[508]:

	County	State	Family Type	Total	Less than $10,000	10, 000to 14,999	15, 000to 24,999	25, 000to 34,999	35, 000to 49,999	50, 000to 74,999	75, 000to 99,999	100, 000 149,
0	West Virginia	West Virginia	Households	736,341	7.10%	6.30%	10.50%	9.90%	12.50%	17.40%	12.20%	14.
1	West Virginia	West Virginia	Families	463,064	4.70%	3.20%	6.90%	8.00%	11.70%	18.90%	14.70%	18.
2	West Virginia	West Virginia	Married-couple families	338,510	1.70%	1.90%	4.70%	5.90%	10.70%	18.90%	16.70%	22.
3	West Virginia	West Virginia	Nonfamily households	273,277	12.50%	12.10%	17.10%	14.00%	13.70%	14.30%	7.30%	5.
4	Berkeley County	West Virginia	Households	51,145	4.10%	3.10%	6.10%	9.10%	10.90%	20.00%	14.40%	17.
...	...	...	...	...	...	...	...	...	...	...	...	
383	Tolland County	Connecticut	Nonfamily households	22,004	11.30%	8.10%	17.40%	10.70%	19.10%	8.00%	8.70%	5.
384	Windham County	Connecticut	Households	45,724	5.20%	3.70%	8.10%	7.60%	7.20%	19.80%	15.70%	17.
385	Windham County	Connecticut	Families	29,938	3.60%	3.20%	4.80%	5.40%	6.50%	16.00%	19.40%	22.
386	Windham County	Connecticut	Married-couple families	N	N	N	N	N	N	N	N	
387	Windham County	Connecticut	Nonfamily households	15,786	9.80%	5.90%	17.60%	12.60%	11.40%	22.40%	7.10%	10.

388 rows × 16 columns

In [509…

```
#removing comma and percentage symbol
six_income = six_income.replace({",":"", "%":""}, regex=True)
```

In [510…

```
#seperating states
wv_income = six_income[six_income["County"] == "West Virginia"]
oh_income = six_income[six_income["County"] == "Ohio"]
ky_income = six_income[six_income["County"] == "Kentucky"]
tn_income = six_income[six_income["County"] == "Tennessee"]
de_income = six_income[six_income["County"] == "Delaware"]
ct_income = six_income[six_income["County"] == "Connecticut"]
```

In [511…

```
wv_income
```

Out[511]:

	County	State	Family Type	Total	Less than $10,000	10, 000to 14,999	15, 000to 24,999	25, 000to 34,999	35, 000to 49,999	50, 000to 74,999	75, 000to 99,999	100, 000to 149,999	150, 000 199,
0	West Virginia	West Virginia	Households	736341	7.10	6.30	10.50	9.90	12.50	17.40	12.20	14.20	
1	West Virginia	West Virginia	Families	463064	4.70	3.20	6.90	8.00	11.70	18.90	14.70	18.40	
2	West Virginia	West Virginia	Married-couple	338510	1.70	1.90	4.70	5.90	10.70	18.90	16.70	22.50	

			families									
3	West Virginia	West Virginia	Nonfamily households	273277	12.50	12.10	17.10	14.00	13.70	14.30	7.30	5.90

In [512…

```
#isolating just the first 4 columns
wv_income_head = wv_income.iloc[:,:4]

#converting slice into a dataframe
wv_inc = pd.DataFrame(wv_income_head)

#isolating just the number columns
wv_income_num = wv_income.iloc[:,4:14]

#isolating the last 2 columns

wv_income_end = wv_income.iloc[:,14:]

#converting slic to dataframe
wv_income_back = pd.DataFrame(wv_income_end)

#converting object type to float
wv_income_num = wv_income_num.astype(float)
```

In [513…

```
#combining into new categories
wv_budget = wv_income_num["$10,000 to $14,999"] + wv_income_num["$15,000 to $24,999"] +
wv_main = wv_income_num["$50,000 to $74,999"] + wv_income_num["$75,000 to $99,999"] + wv
wv_premium = wv_income_num["$150,000 to $199,999"] + wv_income_num["$200,000 or more"]
```

In [514…

```
#adding new categories as a column
wv_inc["Budget"] = wv_budget
wv_inc["Mainstream"] = wv_main
wv_inc["Premium"] = wv_premium

#merging mean and median income to dataframe
wv_final = pd.merge(wv_inc, wv_income_end, left_index=True, right_index=True, how="left"
```

In [515…

```
wv_final
```

Out[515]:

	County	State	Family Type	Total	Budget	Mainstream	Premium	Median income (dollars)	Mean income (dollars)
0	West Virginia	West Virginia	Households	736341	39.2	43.8	9.9	54329	75265
1	West Virginia	West Virginia	Families	463064	29.8	52.0	13.5	70318	89306
2	West Virginia	West Virginia	Married-couple families	338510	23.2	58.1	17.0	83915	102844
3	West Virginia	West Virginia	Nonfamily households	273277	56.9	27.5	3.2	31082	48384

In [516…

```
budget_wv = round(wv_final["Budget"].sum() / 4, 2)

mainstream_wv = round(wv_final["Mainstream"].sum() / 4, 2)

premium_wv = round(wv_final["Premium"].sum() / 4, 2)

print("Budget Percent Total")
print(budget_wv)
```

```
print()
print("Mainstream Percent Total")
print(mainstream_wv)
print()
print("Premium Percent Total")
print(premium_wv)
```

```
Budget Percent Total
37.28

Mainstream Percent Total
45.35

Premium Percent Total
10.9
```

In [517…

```
#ohio state
#isolating just the first 4 columns
oh_income_head = oh_income.iloc[:,:4]

#converting slice into a dataframe
oh_inc = pd.DataFrame(oh_income_head)

#isolating just the number columns
oh_income_num = oh_income.iloc[:,4:14]

#isolating the last 2 columns

oh_income_end = oh_income.iloc[:,14:]

#converting slic to dataframe
oh_income_back = pd.DataFrame(oh_income_end)

#converting object type to float
oh_income_num = oh_income_num.astype(float)
```

In [518…

```
#combining into new categories
oh_budget = oh_income_num["$10,000 to $14,999"] + oh_income_num["$15,000 to $24,999"] +
oh_main = oh_income_num["$50,000 to $74,999"] + oh_income_num["$75,000 to $99,999"] + oh
oh_premium = oh_income_num["$150,000 to $199,999"] + oh_income_num["$200,000 or more"]
```

In [519…

```
#adding new categories as a column
oh_inc["Budget"] = oh_budget
oh_inc["Mainstream"] = oh_main
oh_inc["Premium"] = oh_premium

#merging mean and median income to dataframe
oh_final = pd.merge(oh_inc, oh_income_end, left_index=True, right_index=True, how="left"
```

In [520…

```
oh_final
```

Out[520]:

	County	State	Family Type	Total	Budget	Mainstream	Premium	Median income (dollars)	Mean income (dollars)
120	Ohio	Ohio	Households	4878206	32.1	47.0	15.1	65720	90109
121	Ohio	Ohio	Families	2983145	22.2	53.1	21.3	86001	110719
122	Ohio	Ohio	Married-couple families	2173755	14.4	57.1	27.4	103290	129527
123	Ohio	Ohio	Nonfamily households	1895061	49.8	35.4	4.3	40164	54221

In [521…

```
budget_oh = round(oh_final["Budget"].sum() / 4, 2)

mainstream_oh = round(oh_final["Mainstream"].sum() / 4, 2)

premium_oh = round(oh_final["Premium"].sum() / 4, 2)

print("Budget Percent Total")
print(budget_oh)
print()
print("Mainstream Percent Total")
print(mainstream_oh)
print()
print("Premium Percent Total")
print(premium_oh)
```

```
Budget Percent Total
29.62

Mainstream Percent Total
48.15

Premium Percent Total
17.03
```

In [522…

```
#isolating just the first 4 columns
ky_income_head = ky_income.iloc[:,:4]

#converting slice into a dataframe
ky_inc = pd.DataFrame(ky_income_head)

#isolating just the number columns
ky_income_num = ky_income.iloc[:,4:14]

#isolating the last 2 columns

ky_income_end = ky_income.iloc[:,14:]

#converting slic to dataframe
ky_income_back = pd.DataFrame(ky_income_end)

#converting object type to float
ky_income_num = ky_income_num.astype(float)

#combining into new categories
ky_budget = ky_income_num["$10,000 to $14,999"] + ky_income_num["$15,000 to $24,999"] +
ky_main = ky_income_num["$50,000 to $74,999"] + ky_income_num["$75,000 to $99,999"] + ky
ky_premium = ky_income_num["$150,000 to $199,999"] + ky_income_num["$200,000 or more"]


#adding new categories as a column
ky_inc["Budget"] = ky_budget
ky_inc["Mainstream"] = ky_main
ky_inc["Premium"] = ky_premium

#merging mean and median income to dataframe
ky_final = pd.merge(ky_inc, ky_income_end, left_index=True, right_index=True, how="left"
```

In [523…

```
ky_final
```

Out[523]:

	County	State	Family Type	Total	Budget	Mainstream	Premium	Median income (dollars)	Mean income (dollars)
276	Kentucky	Kentucky	Households	1828680	36.0	45.0	12.2	59341	82614
277	Kentucky	Kentucky	Families	1172125	27.2	51.9	16.6	76119	99631

278	Kentucky	Kentucky	Married-couple families	860710	19.9	57.1	21.1	91212	115427
279	Kentucky	Kentucky	Nonfamily households	656555	54.2	30.2	3.3	33993	48151

In [524…

```
budget_ky = round(ky_final["Budget"].sum() / 4, 2)

mainstream_ky = round(ky_final["Mainstream"].sum() / 4, 2)

premium_ky = round(ky_final["Premium"].sum() / 4, 2)

print("Budget Percent Total")
print(budget_ky)
print()
print("Mainstream Percent Total")
print(mainstream_ky)
print()
print("Premium Percent Total")
print(premium_ky)
```

```
Budget Percent Total
34.33

Mainstream Percent Total
46.05

Premium Percent Total
13.3
```

In [525…

```
#isolating just the first 4 columns
tn_income_head = tn_income.iloc[:,:4]

#converting slice into a dataframe
tn_inc = pd.DataFrame(tn_income_head)

#isolating just the number columns
tn_income_num = tn_income.iloc[:,4:14]

#isolating the last 2 columns

tn_income_end = tn_income.iloc[:,14:]

#converting slic to dataframe
tn_income_back = pd.DataFrame(tn_income_end)

#converting object type to float
tn_income_num = tn_income_num.astype(float)

#combining into new categories
tn_budget = tn_income_num["$10,000 to $14,999"] + tn_income_num["$15,000 to $24,999"] +
tn_main = tn_income_num["$50,000 to $74,999"] + tn_income_num["$75,000 to $99,999"] + tn
tn_premium = tn_income_num["$150,000 to $199,999"] + tn_income_num["$200,000 or more"]


#adding new categories as a column
tn_inc["Budget"] = tn_budget
tn_inc["Mainstream"] = tn_main
tn_inc["Premium"] = tn_premium

#merging mean and median income to dataframe
tn_final = pd.merge(tn_inc, tn_income_end, left_index=True, right_index=True, how="left"

tn_final
```

Out[525]:

	County	State	Family Type	Total	Budget	Mainstream	Premium	Median income (dollars)	Mean income (dollars)
32	Tennessee	Tennessee	Households	2846684	32.9	47.0	14.7	65254	89799
33	Tennessee	Tennessee	Families	1846572	24.3	52.5	19.3	80910	105555
34	Tennessee	Tennessee	Married-couple families	1342153	17.1	56.8	24.5	96141	123035
35	Tennessee	Tennessee	Nonfamily households	1000112	50.5	35.0	5.2	40285	56860

In [526…

```
budget_tn = round(tn_final["Budget"].sum() / 4, 2)

mainstream_tn = round(tn_final["Mainstream"].sum() / 4, 2)

premium_tn = round(tn_final["Premium"].sum() / 4, 2)

print("Budget Percent Total")
print(budget_tn)
print()
print("Mainstream Percent Total")
print(mainstream_tn)
print()
print("Premium Percent Total")
print(premium_tn)
```

```
Budget Percent Total
31.2

Mainstream Percent Total
47.82

Premium Percent Total
15.92
```

In [527…

```
#isolating just the first 4 columns
de_income_head = de_income.iloc[:,:4]

#converting slice into a dataframe
de_inc = pd.DataFrame(de_income_head)

#isolating just the number columns
de_income_num = de_income.iloc[:,4:14]

#isolating the last 2 columns

de_income_end = de_income.iloc[:,14:]

#converting slic to dataframe
de_income_back = pd.DataFrame(de_income_end)

#converting object type to float
de_income_num = de_income_num.astype(float)

#combining into new categories
de_budget = de_income_num["$10,000 to $14,999"] + de_income_num["$15,000 to $24,999"] +
de_main = de_income_num["$50,000 to $74,999"] + de_income_num["$75,000 to $99,999"] + de
de_premium = de_income_num["$150,000 to $199,999"] + de_income_num["$200,000 or more"]


#adding new categories as a column
de_inc["Budget"] = de_budget
```

```python
de_inc["Mainstream"] = de_main
de_inc["Premium"] = de_premium

#merging mean and median income to dataframe
de_final = pd.merge(de_inc, de_income_end, left_index=True, right_index=True, how="left"

de_final
```

Out[527]:

	County	State	Family Type	Total	Budget	Mainstream	Premium	Median income (dollars)	Mean income (dollars)
336	Delaware	Delaware	Households	402334	25.9	49.8	20.2	82174	105438
337	Delaware	Delaware	Families	263885	16.9	54.2	26.6	100128	124756
338	Delaware	Delaware	Married-couple families	197223	11.7	54.9	32.3	112712	139819
339	Delaware	Delaware	Nonfamily households	138449	45.2	40.7	6.2	46579	62657

In [528…

```python
budget_de = round(de_final["Budget"].sum() / 4, 2)

mainstream_de = round(de_final["Mainstream"].sum() / 4, 2)

premium_de = round(de_final["Premium"].sum() / 4, 2)

print("Budget Percent Total")
print(budget_de)
print()
print("Mainstream Percent Total")
print(mainstream_de)
print()
print("Premium Percent Total")
print(premium_de)
```

```
Budget Percent Total
24.92

Mainstream Percent Total
49.9

Premium Percent Total
21.32
```

In [529…

```python
#isolating just the first 4 columns
ct_income_head = ct_income.iloc[:,:4]

#converting slice into a dataframe
ct_inc = pd.DataFrame(ct_income_head)

#isolating just the number columns
ct_income_num = ct_income.iloc[:,4:14]

#isolating the last 2 columns

ct_income_end = ct_income.iloc[:,14:]

#converting slic to dataframe
ct_income_back = pd.DataFrame(ct_income_end)

#converting object type to float
ct_income_num = ct_income_num.astype(float)

#combining into new categories
```

```
ct_budget = ct_income_num["$10,000 to $14,999"] + ct_income_num["$15,000 to $24,999"] +
ct_main = ct_income_num["$50,000 to $74,999"] + ct_income_num["$75,000 to $99,999"] + ct
ct_premium = ct_income_num["$150,000 to $199,999"] + ct_income_num["$200,000 or more"]


#adding new categories as a column
ct_inc["Budget"] = ct_budget
ct_inc["Mainstream"] = ct_main
ct_inc["Premium"] = ct_premium

#merging mean and median income to dataframe
ct_final = pd.merge(ct_inc, ct_income_end, left_index=True, right_index=True, how="left"

ct_final
```

Out[529]:

	County	State	Family Type	Total	Budget	Mainstream	Premium	Median income (dollars)	Mean income (dollars)
352	Connecticut	Connecticut	Households	1428313	25.8	44.3	24.7	83771	120009
353	Connecticut	Connecticut	Families	916362	17.2	47.4	32.7	106576	146203
354	Connecticut	Connecticut	Married-couple families	664848	10.4	47.3	41.3	129296	172356
355	Connecticut	Connecticut	Nonfamily households	511951	43.0	37.8	8.7	45211	68045

In [530…

```
budget_ct = round(ct_final["Budget"].sum() / 4, 2)

mainstream_ct = round(ct_final["Mainstream"].sum() / 4, 2)

premium_ct = round(ct_final["Premium"].sum() / 4, 2)

print("Budget Percent Total")
print(budget_ct)
print()
print("Mainstream Percent Total")
print(mainstream_ct)
print()
print("Premium Percent Total")
print(premium_ct)
```

```
Budget Percent Total
24.1

Mainstream Percent Total
44.2

Premium Percent Total
26.85
```

State	Budget Percent Total	Mainstream Percent Total	Premium Percent Total
WV	37.28	45.35	10.9
OH	29.62	48.15	17.3
KY	34.33	46.05	13.3
TN	31.20	47.82	15.9
DE	24.92	49.90	21.3
CT	24.10	44.20	26.8

Each average state income. The average for top 3 states were 36% Budget, 40.9% Mainstream, and 23% Premium.

If I take the difference percent from every category for each state and minus the average from the top 3 states than add them back together for a total score of difference.

formula used = Budget Pecent - Average Budget, Mainstream Percent - Average Mainstream, Premium Percent - Average Premium than sum the results to get a score.

The states would be ranked in this order.

1) Connecticut 2) Tennessee 3) West Virginia 4) Kentucky 5) Delaware 6) Ohio

Housing Data:

In [531…

```
six_housing
```

Out[531]:

	County	State	Counts	Total housing units	Occupied housing units	Vacant housing units	Homeowner vacancy rate	Rental vacancy rate	Total housing units2	1-unit, detached
0	Connecticut	Connecticut	Estimate	1,536,327	1,428,313	108,014	0.6	4.1	1,536,327	907,419
1	Connecticut	Connecticut	Percent	1,536,327	93.00%	7.00%	(X)	(X)	1,536,327	59.10%
2	Fairfield County	Connecticut	Estimate	380,697	357,271	23,426	0.6	4.3	380,697	213,069
3	Fairfield County	Connecticut	Percent	380,697	93.80%	6.20%	(X)	(X)	380,697	56.00%
4	Hartford County	Connecticut	Estimate	386,152	360,140	26,012	0.3	4.9	386,152	220,472
...	...	...	...	...	...	...	...	...	...	...
189	Monongalia County	West Virginia	Percent	49,952	89.60%	10.40%	(X)	(X)	49,952	48.80%
190	Raleigh County	West Virginia	Estimate	34,630	28,043	6,587	1	5.9	34,630	26,366
191	Raleigh County	West Virginia	Percent	34,630	81.00%	19.00%	(X)	(X)	34,630	76.10%
192	Wood County	West Virginia	Estimate	40,324	37,220	3,104	1.4	4.4	40,324	30,687
193	Wood County	West Virginia	Percent	40,324	92.30%	7.70%	(X)	(X)	40,324	76.10%

194 rows × 35 columns

In [532…

```
#removing comma and percentage symbol
six_housing = six_housing.replace({",":"", "%":""}, regex=True)

#seperating states
wv_housing = six_housing[six_housing["County"] == "West Virginia"]
oh_housing = six_housing[six_housing["County"] == "Ohio"]
ky_housing = six_housing[six_housing["County"] == "Kentucky"]
tn_housing = six_housing[six_housing["County"] == "Tennessee"]
```

```
de_housing = six_housing[six_housing["County"] == "Delaware"]
ct_housing = six_housing[six_housing["County"] == "Connecticut"]
```

In [533…

```
#extracting only percent row
wv_housing_pct = wv_housing.iloc[1:2,:]

wv_housing_pct
```

Out[533]:

	County	State	Counts	Total housing units	Occupied housing units	Vacant housing units	Homeowner vacancy rate	Rental vacancy rate	Total housing units2	1-unit, detached	...	Move i 201 t 201
179	West Virginia	West Virginia	Percent	861686	85.50	14.50	(X)	(X)	861686	70.70	...	19.3

1 rows × 35 columns

In [534…

```
#splitting into more easily readable chunks of data

#county state
wv_housing_head = wv_housing_pct.iloc[:,0:2]

#housing types
wv_housing_type = wv_housing_pct.iloc[:,9:18]

#renter and owner
wv_occupied = wv_housing_pct.iloc[:,19:21]

#moved
wv_moved = wv_housing_pct.iloc[:,24:25]

#car
wv_car = wv_housing_pct.iloc[:,31:]
```

In [535…

```
# combining county and state with each segment
wv_combined_housing = pd.merge(wv_housing_head, wv_housing_type, left_index=True, right_

#housing data
wv_combined_or = pd.merge(wv_combined_housing, wv_occupied, left_index=True, right_index

#moved in pct and car ownership
wv_combined_moved_car = pd.merge(wv_moved, wv_car, left_index=True, right_index=True, ho
```

In [536…

```
wv_combined_or
```

Out[536]:

	County	State	1-unit, detached	1-unit, attached	2 units	3 or 4 units	5 to 9 units	10 to 19 units	20 or more units	Mobile home	Boat, RV, van, etc.	Owner-occupied	Renter-occupied
179	West Virginia	West Virginia	70.70	2.70	2.00	3.00	2.60	1.90	3.20	13.80	0.20	74.50	25.50

In [537…

```
wv_combined_moved_car
```

Out[537]:

	Moved in 2019 or later	No vehicles available	1 vehicle available	2 vehicles available	3 or more vehicles available
179	13.10	8.00	34.40	37.20	20.30

In [538...

```
#extracting only percent row
oh_housing_pct = oh_housing.iloc[1:2,:]

oh_housing_pct
```

Out[538]:

	County	State	Counts	Total housing units	Occupied housing units	Vacant housing units	Homeowner vacancy rate	Rental vacancy rate	Total housing units2	1-unit, detached	...	Moved in 2015 to 2018
57	Ohio	Ohio	Percent	5293227	92.20	7.80	(X)	(X)	5293227	68.80	...	21.60

1 rows × 35 columns

In [539...

```
#splitting into more easily readable chunks of data

#county state
oh_housing_head = oh_housing_pct.iloc[:,0:2]

#housing types
oh_housing_type = oh_housing_pct.iloc[:,9:18]

#renter and owner
oh_occupied = oh_housing_pct.iloc[:,19:21]

#moved
oh_moved = oh_housing_pct.iloc[:,24:25]

#car
oh_car = oh_housing_pct.iloc[:,31:]

# combining county and state with each segment
oh_combined_housing = pd.merge(oh_housing_head, oh_housing_type, left_index=True, right_

#housing data
oh_combined_or = pd.merge(oh_combined_housing, oh_occupied, left_index=True, right_index

#moved in pct and car ownership
oh_combined_moved_car = pd.merge(oh_moved, oh_car, left_index=True, right_index=True, ho
```

In [540...

```
oh_combined_or
```

Out[540]:

	County	State	1-unit, detached	1-unit, attached	2 units	3 or 4 units	5 to 9 units	10 to 19 units	20 or more units	Mobile home	Boat, RV, van, etc.	Owner-occupied	Renter-occupied
57	Ohio	Ohio	68.80	5.00	3.70	4.20	4.50	3.90	6.50	3.40	0.10	67.30	32.70

In [541...

```
oh_combined_moved_car
```

Out[541]:

	Moved in 2019 or later	No vehicles available	1 vehicle available	2 vehicles available	3 or more vehicles available
57	16.40	7.20	34.30	37.70	20.80

In [542...

```
#extracting only percent row
ky_housing_pct = ky_housing.iloc[1:2,:]

ky_housing_pct
```

Out[542]:

	County	State	Counts	Total housing units	Occupied housing units	Vacant housing units	Homeowner vacancy rate	Rental vacancy rate	Total housing units2	1-unit, detached	...	Mov 20 20
27	Kentucky	Kentucky	Percent	2023679	90.40	9.60	(X)	(X)	2023679	67.20	...	21

1 rows × 35 columns

In [543…

```
#splitting into more easily readable chunks of data

#county state
ky_housing_head = ky_housing_pct.iloc[:,0:2]

#housing types
ky_housing_type = ky_housing_pct.iloc[:,9:18]

#renter and owner
ky_occupied = ky_housing_pct.iloc[:,19:21]

#moved
ky_moved = ky_housing_pct.iloc[:,24:25]

#car
ky_car = ky_housing_pct.iloc[:,31:]

# combining county and state with each segment
ky_combined_housing = pd.merge(ky_housing_head, ky_housing_type, left_index=True, right_

#housing data
ky_combined_or = pd.merge(ky_combined_housing, ky_occupied, left_index=True, right_index

#moved in pct and car ownership
ky_combined_moved_car = pd.merge(ky_moved, ky_car, left_index=True, right_index=True, ho
```

In [544…

```
ky_combined_or
```

Out[544]:

	County	State	1-unit, detached	1-unit, attached	2 units	3 or 4 units	5 to 9 units	10 to 19 units	20 or more units	Mobile home	Boat, RV, van, etc.	Owner-occupied	Renter-occupied
27	Kentucky	Kentucky	67.20	2.90	2.70	4.30	4.70	3.40	3.70	10.90	0.20	68.80	31.20

In [545…

```
ky_combined_moved_car
```

Out[545]:

	Moved in 2019 or later	No vehicles available	1 vehicle available	2 vehicles available	3 or more vehicles available
27	17.40	6.40	32.70	37.60	23.40

In [546…

```
#extracting only percent row
tn_housing_pct = tn_housing.iloc[1:2,:]

tn_housing_pct
```

Out[546]:

	County	State	Counts	Total housing units	Occupied housing units	Vacant housing units	Homeowner vacancy rate	Rental vacancy rate	Total housing units2	1-unit, detached	...	M

135	Tennessee	Tennessee	Percent	3144583	90.50	9.50	(X)	(X)	3144583	68.40	...

1 rows × 35 columns

In [547…

```python
#splitting into more easily readable chunks of data

#county state
tn_housing_head = tn_housing_pct.iloc[:,0:2]

#housing types
tn_housing_type = tn_housing_pct.iloc[:,9:18]

#renter and owner
tn_occupied = tn_housing_pct.iloc[:,19:21]

#moved
tn_moved = tn_housing_pct.iloc[:,24:25]

#car
tn_car = tn_housing_pct.iloc[:,31:]

# combining county and state with each segment
tn_combined_housing = pd.merge(tn_housing_head, tn_housing_type, left_index=True, right_

#housing data
tn_combined_or = pd.merge(tn_combined_housing, tn_occupied, left_index=True, right_index

#moved in pct and car ownership
tn_combined_moved_car = pd.merge(tn_moved, tn_car, left_index=True, right_index=True, ho
```

In [548…

```python
tn_combined_or
```

Out[548]:

	County	State	1-unit, detached	1-unit, attached	2 units	3 or 4 units	5 to 9 units	10 to 19 units	20 or more units	Mobile home	Boat, RV, van, etc.	Owner-occupied	Rent occupi
135	Tennessee	Tennessee	68.40	3.90	2.40	3.10	4.20	3.50	6.10	8.30	0.20	67.20	32.

In [549…

```python
tn_combined_moved_car
```

Out[549]:

	Moved in 2019 or later	No vehicles available	1 vehicle available	2 vehicles available	3 or more vehicles available
135	18.20	5.10	30.60	38.60	25.80

In [550…

```python
#extracting only percent row
de_housing_pct = de_housing.iloc[1:2,:]

de_housing_pct
```

Out[550]:

	County	State	Counts	Total housing units	Occupied housing units	Vacant housing units	Homeowner vacancy rate	Rental vacancy rate	Total housing units2	1-unit, detached	...	Mov 20 20
19	Delaware	Delaware	Percent	465804	86.40	13.60	(X)	(X)	465804	60.60	...	22

1 rows × 35 columns

In [551…

```
#splitting into more easily readable chunks of data

#county state
de_housing_head = de_housing_pct.iloc[:,0:2]

#housing types
de_housing_type = de_housing_pct.iloc[:,9:18]

#renter and owner
de_occupied = de_housing_pct.iloc[:,19:21]

#moved
de_moved = de_housing_pct.iloc[:,24:25]

#car
de_car = de_housing_pct.iloc[:,31:]

# combining county and state with each segment
de_combined_housing = pd.merge(de_housing_head, de_housing_type, left_index=True, right_

#housing data
de_combined_or = pd.merge(de_combined_housing, de_occupied, left_index=True, right_index

#moved in pct and car ownership
de_combined_moved_car = pd.merge(de_moved, de_car, left_index=True, right_index=True, ho


de_combined_or
```

Out[551]:

	County	State	1-unit, detached	1-unit, attached	2 units	3 or 4 units	5 to 9 units	10 to 19 units	20 or more units	Mobile home	Boat, RV, van, etc.	Owner-occupied	Renter-occupied
19	Delaware	Delaware	60.60	16.20	1.00	2.10	3.60	5.00	5.00	6.50	0.10	74.10	25.90

In [552…

```
de_combined_moved_car
```

Out[552]:

	Moved in 2019 or later	No vehicles available	1 vehicle available	2 vehicles available	3 or more vehicles available
19	15.10	5.90	32.50	40.60	20.90

In [553…

```
#extracting only percent row
ct_housing_pct = ct_housing.iloc[1:2,:]

ct_housing_pct
```

Out[553]:

	County	State	Counts	Total housing units	Occupied housing units	Vacant housing units	Homeowner vacancy rate	Rental vacancy rate	Total housing units2	1-unit, detached	...
1	Connecticut	Connecticut	Percent	1536327	93.00	7.00	(X)	(X)	1536327	59.10	...

1 rows × 35 columns

In [554…

```
#splitting into more easily readable chunks of data
```

```
#county state
ct_housing_head = ct_housing_pct.iloc[:,0:2]

#housing types
ct_housing_type = ct_housing_pct.iloc[:,9:18]

#renter and owner
ct_occupied = ct_housing_pct.iloc[:,19:21]

#moved
ct_moved = ct_housing_pct.iloc[:,24:25]

#car
ct_car = ct_housing_pct.iloc[:,31:]

# combining county and state with each segment
ct_combined_housing = pd.merge(ct_housing_head, ct_housing_type, left_index=True, right_

#housing data
ct_combined_or = pd.merge(ct_combined_housing, ct_occupied, left_index=True, right_index

#moved in pct and car ownership
ct_combined_moved_car = pd.merge(ct_moved, ct_car, left_index=True, right_index=True, ho

ct_combined_or
```

Out[554]:

	County	State	1-unit, detached	1-unit, attached	2 units	3 or 4 units	5 to 9 units	10 to 19 units	20 or more units	Mobile home	Boat, RV, van, etc.	Owner-occupied	Ren occup
1	Connecticut	Connecticut	59.10	6.80	7.20	7.80	4.90	3.80	9.70	0.80	0.10	66.60	3:

In [555…

```
ct_combined_moved_car
```

Out[555]:

	Moved in 2019 or later	No vehicles available	1 vehicle available	2 vehicles available	3 or more vehicles available
1	23.90	8.40	33.60	37.50	20.40

State	Homeownership Rate	Dominate Type	Moved in Rate	Car Ownership Rate
Top 3 States	54%	Condo/Coop	14%	66%
WV	74%	Single Family	13%	92%
OH	67%	Single Family	16%	92.8%
KY	69%	Single Family	17%	93.6%
TN	67%	Single Family	18%	94.9%
DE	74%	Single Family	15%	94.1%
CT	66%	Single Family	24%	91.6%

All six states have high car ownership over 90%. Homeownership are all over 66% Single Family homes are the most dominate housing type in the 6 states. Population Growth or moved in rate is above the 14% except for West Virginia I think the only way to measure is by the population growth.

Ranking:

1) Connecticut 2) Tennessee 3) Kentucky 4) Ohio 5) Delaware 6) West Virginia

Household Data:

In [570…

```
six_household
```

Out[570]:

	County	State	Label	Total households	Married-couple household	With children of the householder under 18 years	Cohabiting couple household	With children of the householder under 18 years2	househ… spouse/pa… pr…
0	Connecticut	Connecticut	Estimate	1,428,313	664,848	249,151	104,776	31,427	24
1	Connecticut	Connecticut	Percent	1,428,313	46.50%	17.40%	7.30%	2.20%	1
2	Fairfield County	Connecticut	Estimate	357,271	182,947	75,190	21,997	6,474	5
3	Fairfield County	Connecticut	Percent	357,271	51.20%	21.00%	6.20%	1.80%	1.
4	Hartford County	Connecticut	Estimate	360,140	158,481	64,991	26,119	7,457	6
...	...	...	...	...	...	...	...	...	
189	Monongalia County	West Virginia	Percent	44,767	34.90%	13.80%	9.00%	2.50%	2
190	Raleigh County	West Virginia	Estimate	28,043	12,513	3,853	928	322	
191	Raleigh County	West Virginia	Percent	28,043	44.60%	13.70%	3.30%	1.10%	1
192	Wood County	West Virginia	Estimate	37,220	15,549	4,616	3,228	1,347	
193	Wood County	West Virginia	Percent	37,220	41.80%	12.40%	8.70%	3.60%	2

194 rows × 40 columns

In [571…

```
six_household.columns
```

Out[571]:

```
Index(['County', 'State', 'Label', 'Total households',
       'Married-couple household',
       'With children of the householder under 18 years',
       'Cohabiting couple household',
       'With children of the householder under 18 years2',
       'Male householder, no spouse/partner present',
       'With children of the householder under 18 years3',
       'Householder living alone', '65 years and over',
       'Female householder, no spouse/partner present',
       'With children of the householder under 18 years4',
       'Householder living alone5', '65 years and over6',
       'Households with one or more people under 18 years',
       'Households with one or more people 65 years and over',
       'Average household size', 'Average family size',
       'Population in households', 'Householder', 'Spouse',
       'Unmarried partner', 'Child', 'Other relatives', 'Other nonrelatives',
       'Males 15 years and over', 'Never married',
       'Now married, except separated', 'Separated', 'Widowed', 'Divorced',
```

```
       'Females 15 years and over', 'Never married7',
       'Now married, except separated8', 'Separated9', 'Widowed10',
       'Divorced11', 'Column12'],
      dtype='object')
```

In [576…

```
#removing comma and percentage symbol
six_household = six_household.replace({",":"", "%":""}, regex=True)

#dropping unwanted columns into a new variable
six_hh = six_household[['County', 'State','Label','Total households','Married-couple hou
'Cohabiting couple household','With children of the householder under 18 years2',
'Average household size', 'Average family size']]


#seperating states
wv_hh = six_hh[six_hh["County"] == "West Virginia"]
oh_hh = six_hh[six_hh["County"] == "Ohio"]
ky_hh = six_hh[six_hh["County"] == "Kentucky"]
tn_hh = six_hh[six_hh["County"] == "Tennessee"]
de_hh = six_hh[six_hh["County"] == "Delaware"]
ct_hh = six_hh[six_hh["County"] == "Connecticut"]
```

In [577…

```
wv_hh
```

Out[577]:

	County	State	Label	Total households	Married-couple household	With children of the householder under 18 years	Cohabiting couple household	With children of the householder under 18 years2	Average household size	Avera fam s
178	West Virginia	West Virginia	Estimate	736341	338510	107763	49938	16403	2.34	2
179	West Virginia	West Virginia	Percent	736341	46.00	14.60	6.80	2.20	(X)	

In [580…

```
oh_hh
```

Out[580]:

	County	State	Label	Total households	Married-couple household	With children of the householder under 18 years	Cohabiting couple household	With children of the householder under 18 years2	Average household size	Average family size
56	Ohio	Ohio	Estimate	4878206	2173755	777371	380688	119414	2.35	2.98
57	Ohio	Ohio	Percent	4878206	44.60	15.90	7.80	2.40	(X)	(X)

In [581…

```
ky_hh
```

Out[581]:

	County	State	Label	Total households	Married-couple household	With children of the householder under 18 years	Cohabiting couple household	With children of the householder under 18 years2	Average household size	Ave fa
26	Kentucky	Kentucky	Estimate	1828680	860710	313826	132618	47167	2.4	
27	Kentucky	Kentucky	Percent	1828680	47.10	17.20	7.30	2.60	(X)	

In [582…
```
tn_hh
```

Out[582]:

	County	State	Label	Total households	Married-couple household	With children of the householder under 18 years	Cohabiting couple household	With children of the householder under 18 years2	Average household size	A
134	Tennessee	Tennessee	Estimate	2846684	1342153	479805	195854	61916	2.43	
135	Tennessee	Tennessee	Percent	2846684	47.10	16.90	6.90	2.20	(X)	

In [584…
```
de_hh
```

Out[584]:

	County	State	Label	Total households	Married-couple household	With children of the householder under 18 years	Cohabiting couple household	With children of the householder under 18 years2	Average household size	Ave fa
18	Delaware	Delaware	Estimate	402334	197223	60960	28592	10999	2.47	
19	Delaware	Delaware	Percent	402334	49.00	15.20	7.10	2.70	(X)	

In [586…
```
ct_hh
```

Out[586]:

	County	State	Label	Total households	Married-couple household	With children of the householder under 18 years	Cohabiting couple household	With children of the householder under 18 years2	Average household size
0	Connecticut	Connecticut	Estimate	1428313	664848	249151	104776	31427	2.45
1	Connecticut	Connecticut	Percent	1428313	46.50	17.40	7.30	2.20	(X)

Notes:

Comparing the 6 states household data the order of ranking would be the following:

1) Delaware 2) Connecticut 3) Kentucky 4) Tennessee 5) Ohio 6) West Virginia

All Ranks:

Diversity Ranks:

1) Delaware 2) Connecticut 3) Tennessee 4) Ohio 5) Kentucky 6) West Virginia

Age Ranks:

1) Connecticut 2) Tennessee 3) West Virginia 4) Kentucky 5) Delaware 6) Ohio

Income Ranks:

1) Connecticut 2) Tennessee 3) West Virginia 4) Kentucky 5) Delaware 6) Ohio

Housing Ranks:

1) Connecticut 2) Tennessee 3) Kentucky 4) Ohio 5) Delaware 6) West Virginia

Houshold Ranks:

1) Delaware 2) Connecticut 3) Kentucky 4) Tennessee 5) Ohio 6) West Virginia

Based on the State rankings combined:

1) Connecticut 1.4 2) Tennessee 2.6 3) Delaware 3.4 4) Kentucky 3.8 5) West Virginia 4.8 6) Ohio 5

Insights and Recommendation:

Based on all census demographics and comparing it to the top 3 performing states Connecticut ranks the most likely state for the wholesale club to have success in. The negative with Connecticut is there is a high concentration of competitors within the state. Even with the competition it is spread out enough for the wholesale club to penetrate the market. Especially Bridgeport which has the most population. The only competitor directly in the city is Bj's.

The second state is Tennessee. Tennessee only has 2 competitors but they are concentrated in the 3 largest cities. The largest city Nashville has the largest concentration of competition. Knoxville or Memphis would be a better choice.

Delaware would be the fourth choice in my opinion based on solely on population. The downside of Delaware is that it is a small state and most of the competition has a presence in Wilmington which is the largest city. All of the other cities have a drastically smaller population than any other state.

Kentucky would probably be a better choice than Delaware based on population size. Kentucky has only 2 of 3 competitors within the state but they are both in the largest populated cities. This would make it a little hard to gain traction in the beginning.

West Virginia's only benefit is that it would have 1 competitor which is Sam's Club. Other than that its a small state and scored 2nd to last on the rankings.

Ohio scores last in the rankings and all 3 competitors are present in the state. The competition is stacked in all major urban areas. This would prove difficult to penetrate this market and gain traction. This would be the worst state to enter based on the competitor presence and ranking.

Based on all of the information I would recommend Connecticut. Its scores the highest on the state rankings and it has the largest growth in population amongst the other 6 states as well.