US Affordability Data

Group 9
Quinton Mays
Matt Suozzi
Eric Pratsch

Github repository: https://github.com/q-maze/location-affordability-tool

Overview

- Data Background
  - HUD & DOT Data
  - Affordability index based on Census data
- Project Objectives
  - Data Cleaning
  - Exploratory Data Analysis
  - Key Questions
  - User Input
- Future Enhancements
- Conclusions

The Dataset: Location Affordability Index

- The Location Affordability Index (LAI) combines data from the US Department of Housing and Urban Development (HUD) and Department of Transportation (DOT) to provide standardized housing and transportation costs.
- LAI provides eight different household profiles to describe affordability and living costs based on family size, home ownership, auto ownership, and job situation.
- Other factors include residential density, employment access, job diversity, commute distance, and income.
- The dataset was stored as a pandas dataframe with 219,829 rows and 123 columns.
- The LAI includes Census Federal Information Processing Standard (FIPS) as location IDs, but does not include the city or county names, so the LAI was merged with a geopandas shapefile to provide names and plot the regions on a heatmap.

Household Types

- The LAI provides eight household types based with different attributes measured and/or modeled for each region: income, family size, number of workers, number of vehicles, transportation costs, housing costs, vehicle miles traveled, transit trips
    - Housing costs are provided for owners and renters separately and on a combined basis. For example, in New York City the combined housing cost index is 27.95, while it is 44.58 for owners and 23.96 for renters.

Household Type	Income	Family Size	Number of Commuters
Median	Median Household Income for an area	4	2
Moderate	80% of Median	3	1
Dual	150% of Median	4	2
Low	50% of Median	1	1
Very Low*	National Poverty Line ($11,880)	1	1
Single Professional	135% of Median	1	1
Single Worker	50% of Median	1	1
Retirees	80% of Median	2	0

Data Cleaning

1. Location IDs had dropped leading zeros, so applied zfill to properly merge ID's with the county names from the geopandas file
2. Dropped Very Low household profile as it is consistent for every region
3. Dropped Hawaii, Alaska, and District of Columbia as a way to remove potential outliers and to focus on states in the continental United States

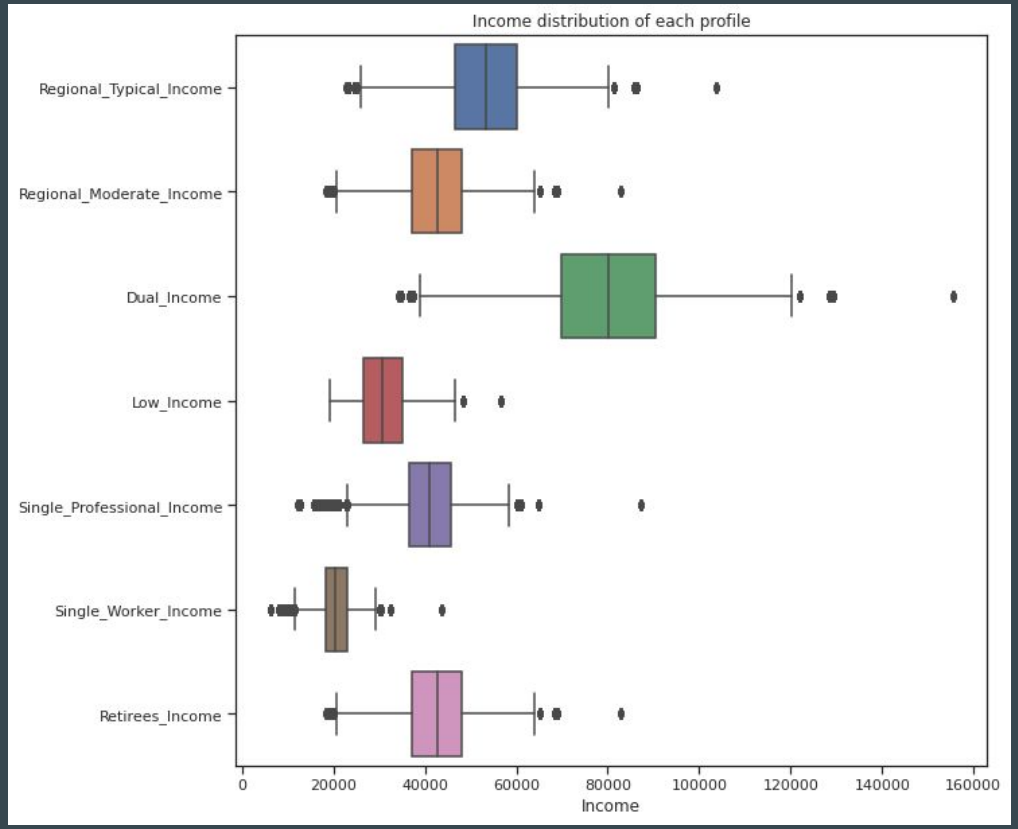
Before

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 219829 entries, 0 to 219828
Columns: 123 entries, FID to SHAPE_Area
dtypes: float64(115), int64(6), object(2)
memory usage: 206.3+ MB
None
```

After

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 196564 entries, 0 to 217221
Columns: 111 entries, Object_ID to State_County_ID
dtypes: float64(102), int32(1), int64(6), object(2)
memory usage: 167.2+ MB
None
```

Household Types: Income distribution



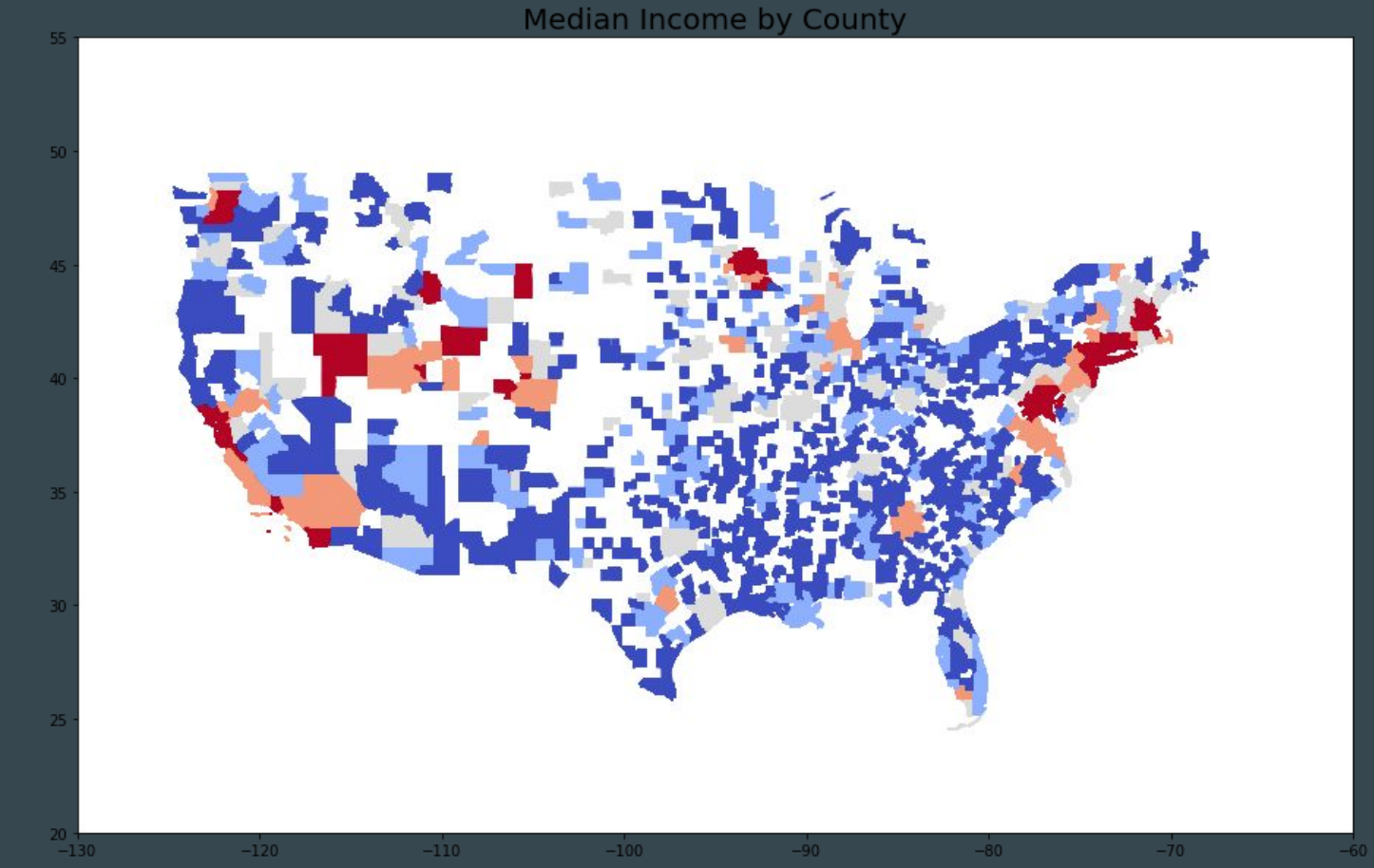
The box plot shows the variability of each household profile's income throughout the country.

Note: Regional Moderate Income and Retirees Income are similarly based on 80% of Regional Typical Income.

Key Questions

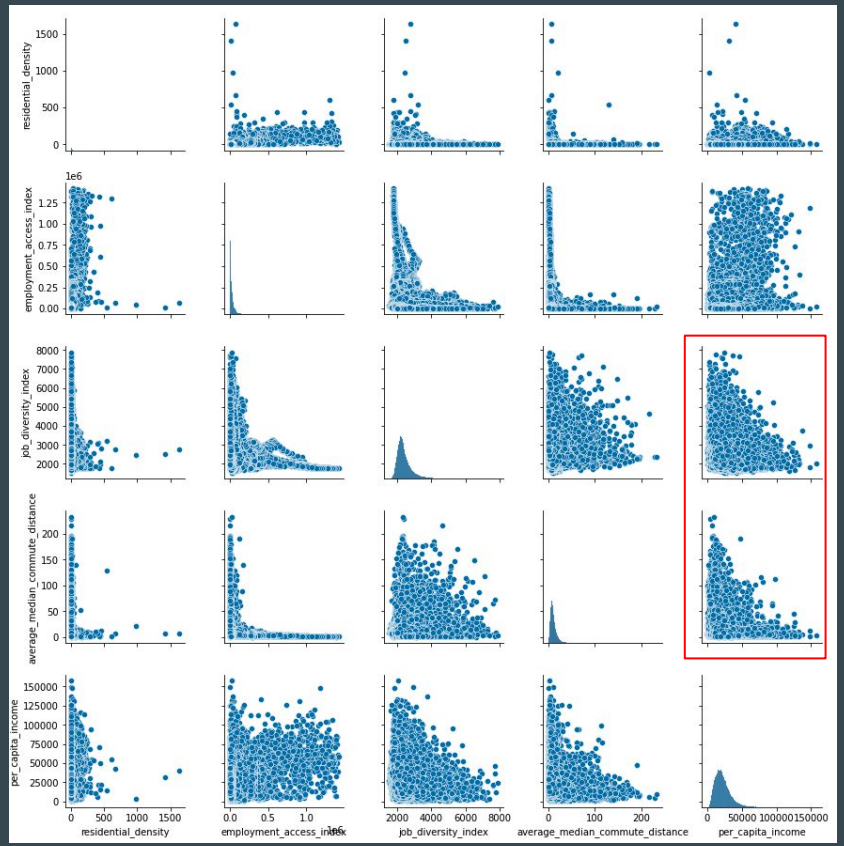
1. What is the relationship between income and housing and transportation costs?
2. What regions have significant wealth gaps or highly variable income distributions?
3. What locations have the highest and lowest income?
4. What is an affordable location in a state based on income and other factors?

Exploratory Data Analysis



Used the geopandas library to merge the LAI data set with a county shapefile and create the heatmap visualization for median income by county.

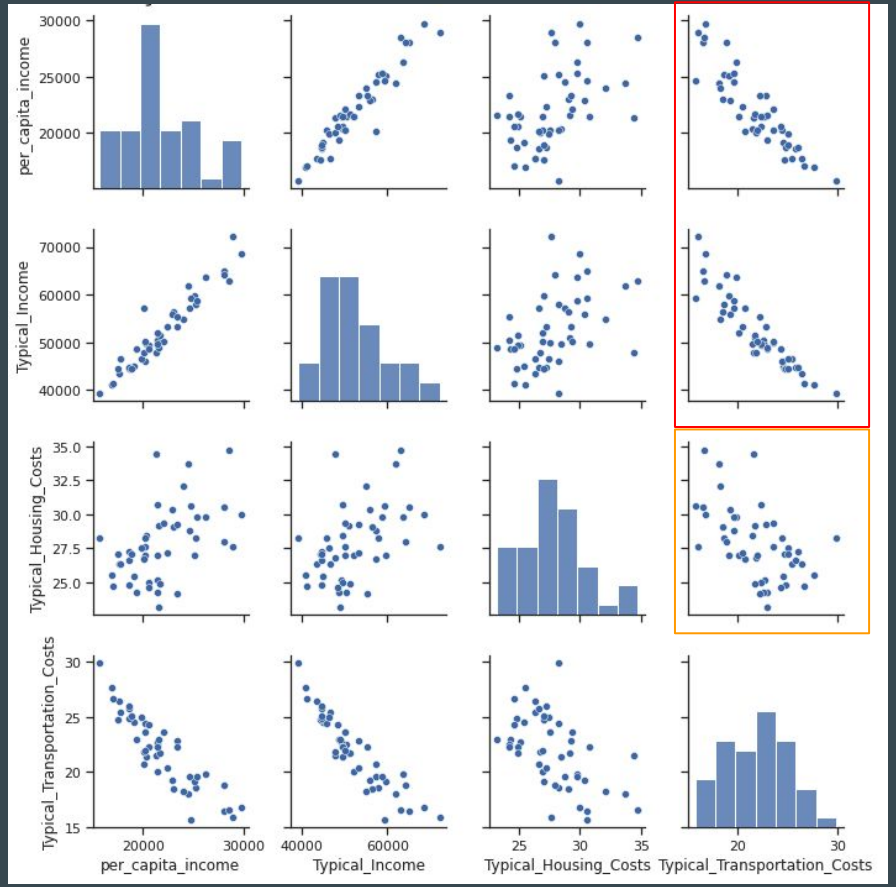
Exploratory Data Analysis



The plot shows the relationship between the key variables - residential density, employment access index, job diversity index, average median commute distance, and per capita income - for the entire country, so there will be high variability.

Per capita income seems to have a weak negative correlation with job diversity and commute distance, while the other variables are not meaningfully related.

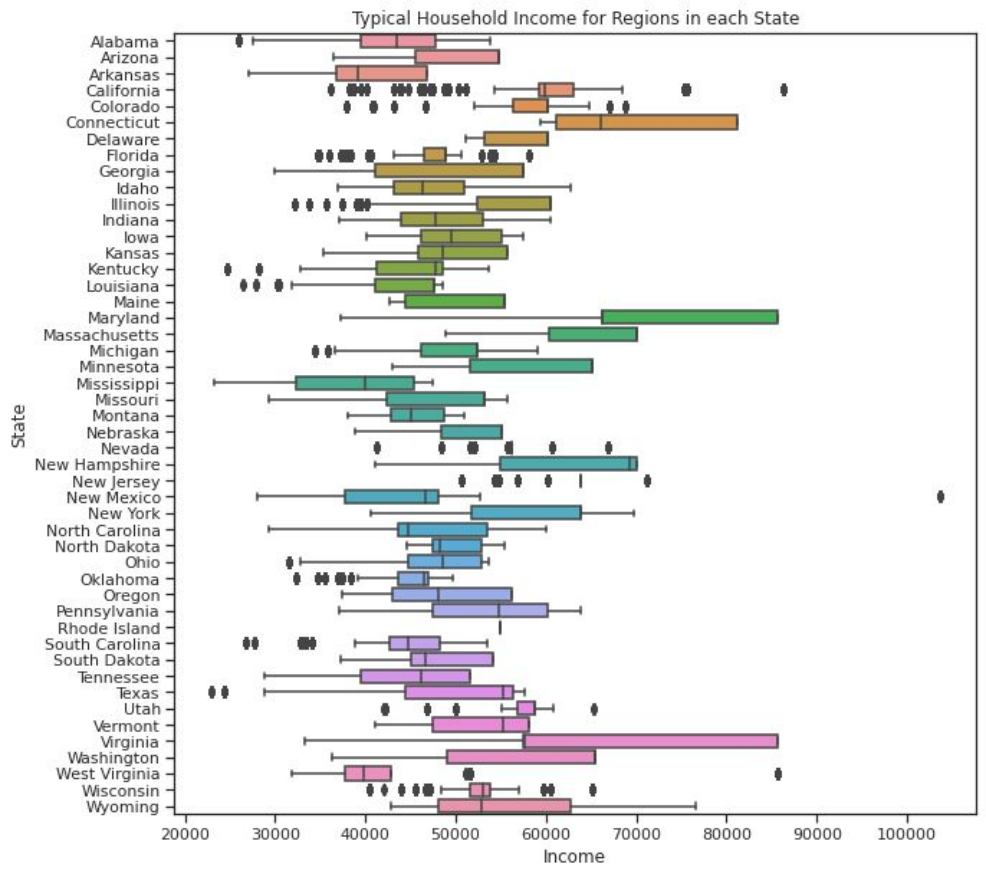
What is the relationship between income and housing and transportation costs?



Transportation Cost vs. Income & Housing

- Clear negative correlation between transportation cost and income (-.917) and housing cost (-.532).
- Houses further from cities tend to be cheaper, but cost more in transportation
- Higher percentage of income dedicated to transportation cost vs. equity in a house
- As income increases, higher cost housing in better locations becomes more affordable and transportation costs decrease

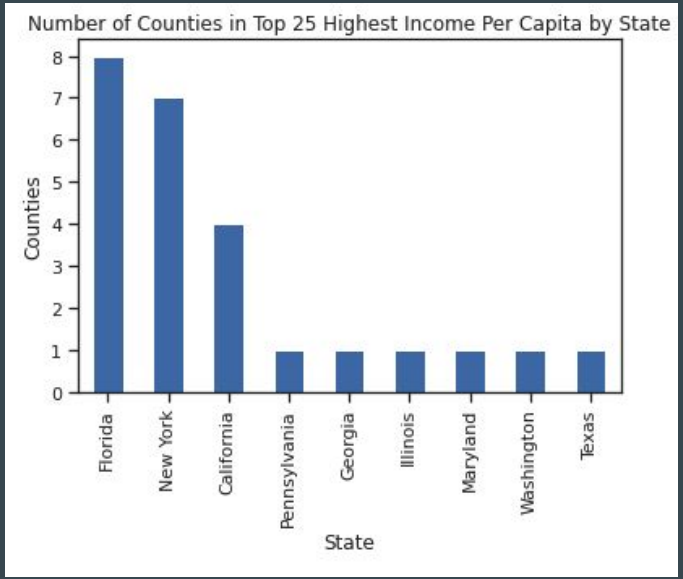
What regions have significant wealth gaps or highly variable income distributions?

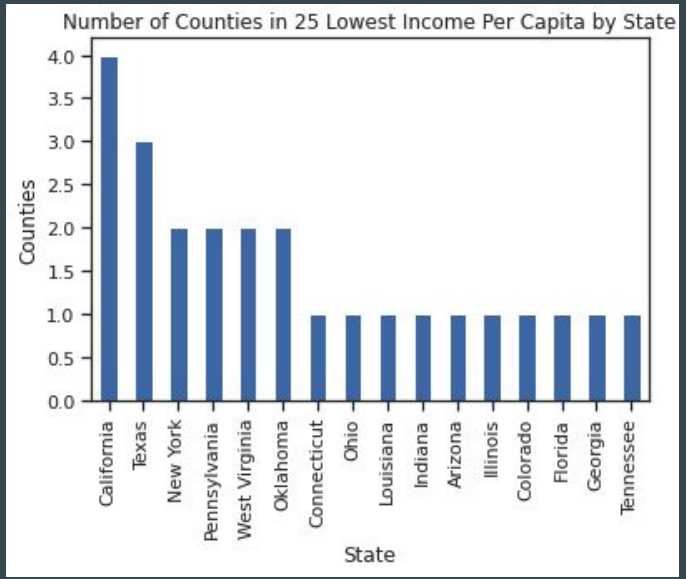


California and certain Northeast and Mid-Atlantic states show large differences in high and low income.

The median income for Southern states is lower compared to other regions, while the variability is tighter.

What locations have the highest and lowest income?





Highest Income Per Capita is highly concentrated in three states - Florida, New York, and California

The lowest income per capita by state is more distributed across more states

What is an affordable location in a state based on income and other factors?

Developed a user interface to classify a profile as a household type based on the inputs:

```
def get_filters():
    state_data = {'alabama': 'AL', 'alaska': 'AK', 'arizona': 'AZ', 'arkansas': 'AZ',
                  "colorado": 'CO', 'connecticut': 'CT', 'delaware': 'DE', 'florida':
                  'hawaii': 'HI', 'idaho': 'ID', 'illinois': 'IL', 'indiana': 'IN', 'io
                  'kentucky': 'KY', 'louisiana': 'LA', 'maine': 'ME', 'maryland': 'MD',
                  'massachusetts': 'MA', 'michigan': 'MI', 'minnesota': 'MN', 'mississi
                  'montana': 'MT', 'nebraska': 'NE', 'nevada': 'NV', 'New Hampshire':
                  'new jersey': 'NJ', 'new mexico': 'NM', 'new york': 'NY', 'north caro
                  'north dakota': 'ND', 'ohio': 'OH', 'oklahoma': 'OK', 'oregon': 'OR',
                  'pennsylvania': 'PA', 'rhode island': 'RI', 'south carolina': 'SC',
                  'tennesee': 'TN', 'texas': 'TX', 'utah': 'UT', 'vermont': 'VT', 'virg
                  'west virginia': 'WV', 'wisconsin': 'WI', 'wyoming': 'WY', 'washingto
    # Grab all inputs of the user
    print('Hello! Let\'s find you an affordable place!')
    # Get user's State
    while True:
        try:
            state = str(input('Please enter a State you are interested in: ').lower())
            if state in state_data.keys():
                state = state_data.get(state)
                break
            print("Please enter a valid state")
        except Exception as e:
            print(e)

    # Get user income level
    while True:
        try:
            income = int(input('Please enter your income: '))
            if income >= 0:
                break
            print("Please enter a valid income")
        except Exception as e:
            print(e)
```

```
LAI = LocationAffordabilityIndex()
```

```
user = get_filters()
LAI.add_user_prof(user)
result = LAI.show_affordable_locations()
```

```
Hello! Let's find you an affordable place!
Please enter a State you are interested in: new york
Please enter your income: 100000
Are you planning on Renting or Owning a Home?
Enter Renting or Owning: renting
Are you Retired?
Enter Yes or No: no
Single or Dual Income?
Enter Single or Dual: dual
Do you prefer Public Transit or Driving?
Enter Public Transit or Driving: public transit
----------------------------------------
```

Affordable Locations based on user input (continued)

Based on the inputs provided, the user is classified as one of the household types and a list of counties in the state is provided that meets the affordability criteria.

county	State	residential_density	employment_access_index
New York	NY	97.591965	687321.959284
Westchester	NY	7.592967	49182.624461
Nassau	NY	4.843407	56971.558445
Putnam	NY	1.347188	9606.000190
Suffolk	NY	2.318005	23835.146624
Rockland	NY	3.087681	28656.086172
Dutchess	NY	2.204580	11057.457169
Saratoga	NY	2.074100	8315.271697
Richmond	NY	9.284151	60885.204684
Orange	NY	2.784179	10985.478555

Prediction Model

OLS Regression Results

Dep. Variable:	employment_access_index	**R-squared:**	0.423
Model:	OLS	**Adj. R-squared:**	0.423
Method:	Least Squares	**F-statistic:**	7.192e+04
Date:	Thu, 22 Apr 2021	**Prob (F-statistic):**	0.00
Time:	14:55:53	**Log-Likelihood:**	-2.4232e+06
No. Observations:	196564	**AIC:**	4.846e+06
Df Residuals:	196561	**BIC:**	4.846e+06
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.611e+04	209.773	124.473	0.000	2.57e+04	2.65e+04
residential_density	3452.8542	9.597	359.772	0.000	3434.044	3471.665
average_median_commute_distance	-933.5539	13.325	-70.063	0.000	-959.670	-907.438

Omnibus:	193767.273	**Durbin-Watson:**	0.564
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	10217344317.490
Skew:	-3.019	**Prob(JB):**	0.00
Kurtosis:	1119.905	**Cond. No.**	27.3

Predict employment access index based on residential density and commute distance.

While the adjusted R-squared of 0.423 suggests that the model does not explain the variability in employment access well, the moderate score is acceptable for country-wide data.

Perhaps the model could be improved by including a region variable.

Unit Testing

Unit testing was conducted on the LAIUser class to test that users were classified appropriately.

```python
class userInputCase(unittest.TestCase):
  def test_income_zero(self):
    user1 = LAIUser('NJ', 0, 'renting', 'no', 'single', 'public transit')
    print("What is the user's income? " + str(user1.income))
    self.assertEqual(user1.income, 0)
  def test_state(self):
    user1 = LAIUser('NJ', 0, 'renting', 'no', 'single', 'public transit')
    print("What is the user's state? " + str(user1.state))
    self.assertEqual(user1.state, 'NJ')
  def test_retired(self):
    user2 = LAIUser('VA', 60000, 'owning', 'yes', 'single', 'driving')
    user2.classify_user()
    print("What is the user's classification? " + str(user2.classification))
    self.assertEqual(user2.classification, 'retired')
  def test_retired2(self):
    user2 = LAIUser('VA', 60000, 'owning', 'yes', 'single', 'driving')
    user2.classify_user()
    self.assertNotEqual(user2.classification, 'single')
  def test_single(self):
    user3 = LAIUser('WV', 60000, 'renting', 'no', 'single', 'driving')
    user3.classify_user()
    print("What is the user's classification? " + str(user3.classification))
    self.assertEqual(user3.classification, 'single')
  def test_dual(self):
    user4 = LAIUser('CA', 120000, 'owning', 'no', 'dual', 'driving')
    user4.classify_user()
    print("What is the user's classification? " + str(user4.classification))
    self.assertEqual(user4.classification, 'dual')
```

```python
# Run all unit tests
if __name__ == '__main__':
  unittest.main(argv=[''], verbosity=2, exit=False)
```

```
test_dual (__main__.userInputCase) ... ok
test_income_zero (__main__.userInputCase) ... ok
test_retired (__main__.userInputCase) ... ok
test_retired2 (__main__.userInputCase) ... ok
test_single (__main__.userInputCase) ... ok
test_state (__main__.userInputCase) ... What is the user's classification? dual
What is the user's income? 0
What is the user's classification? retired
What is the user's classification? single
What is the user's state? NJ
ok

----------------------------------------------------------------------
Ran 6 tests in 0.014s

OK
```

Unit Testing

Unit testing on LocationAffordabilityIndex to check for correct instantiation of pandas DataFrames and interpretation of user input

```
Ran 3 tests in 43.136s

OK
```

```python
from locationaffordabilityindex import *
import laiuser
import unittest


class TestLAI(unittest.TestCase):
    def setUp(self):
        self.LAI = LocationAffordabilityIndex()

        self.LAI2 = LocationAffordabilityIndex()

        self.user = laiuser.LAIUser('utah', 100000, 'renting', 'yes', 'retired', 'driving')
        self.user.classify_user()

        self.LAI2.add_user_prof(self.user)

    def test_init(self):
        # check to make sure the file locations are stored properly
        self.assertNotEqual(self.LAI.lai_csv_file_path, None)
        self.assertNotEqual(self.LAI.fips_data_file_path, None)
        # check to make sure the df instance variable is initialized correctly
        self.assertNotEqual(len(self.LAI.df.columns), 0)
        self.assertEqual(len(self.LAI.df.columns), 127)
        # check to make sure the user and by_county are blank until a profile is added
        self.assertEqual(self.LAI.user, None)
        self.assertEqual(self.LAI.by_county, None)

    def test_add_user_prof(self):
        # check to make sure the user object has been added
        self.assertNotEqual(self.LAI2.user, None)
        # check to make sure the length of the by_counties df is correct
        self.assertNotEqual(len(self.LAI2.by_county.columns), 0)

    def test_show_affordable_locations(self):
        # check to make sure the length of the by_counties df is correct
        self.assertNotEqual(len(self.LAI2.show_affordable_locations().columns), 0)
        self.assertEqual(len(self.LAI2.show_affordable_locations().columns), 24)
```

Future Enhancements

1. Refine User Classification
2. Refine prediction model
3. Integrate prediction model with user interface
4. Utilize K-Means clusters to identify comparable regions in different states
5. Integrate existing classes into live application with GUI for user input

Conclusions

- Higher income areas and higher housing costs are clustered around major cities
- Transportation costs increase as housing costs decrease
- While higher residential density is associated with more jobs, it does not necessarily lead to higher income for the region
- New York City is a unique example with a mix of high residential density and high income