Final-Report.md

CS 5010 Project

- Quinton Mays (rub9ez)
- Matt Suozzi (mds5dd)
- Eric Pratsch (ewp4x)

- GitHub Repo
- Presentation

Introduction

Our team was interested in learning more about patterns in housing affordability within the continental United States. Our goal for this project was to learn about overall trends in housing affordability across different geographic regions and to use this knowledge to create an application for users to identify affordable places based on a target state at the county level.

The Data

The first dataset that our team selected for this application was the Location Affordability Index from the United States Department of Housing and Urban Development (United States Department of Housing and Urban Development, 2019). The data set contains data on differenet affordability criteria for each census tract in the United States including:

- Transportation Cost Data
- Housing Cost Data
- Population and Geoprofile Data
- Employment Data
- Detailed Household Profile Data

The location affordability index contains over 200,000 rows and 123 columns and is over 200MB in size. More information, including a data dictionary and download access, can be found here: Location Affordability Index v.3 page.

The second dataset, the 2016 County and Place Federal Information Processing Standard (FIPS) codes, used comes from the United States Census Bureau (United States Census Bureau, 2017) and contains the county names FIPs codes from the Location Affordability Index dataset. The dataset can be found here: 2016 FIPS Codes. This dataset was joined to the Location Affordability Index on the county level FIPS code.

Based on initial analysis of the data there are eight types of households that are classified in the data set:

- type 1 - Typical HH
- type 2 - Moderate HH
- type 3 - Dual Income HH
- type 4 - Low Income HH
- type 5 - Single Very Low Income
- type 6 - Single Professional HH
- type 7 - Single Worker HH
- type 8 - Retirees

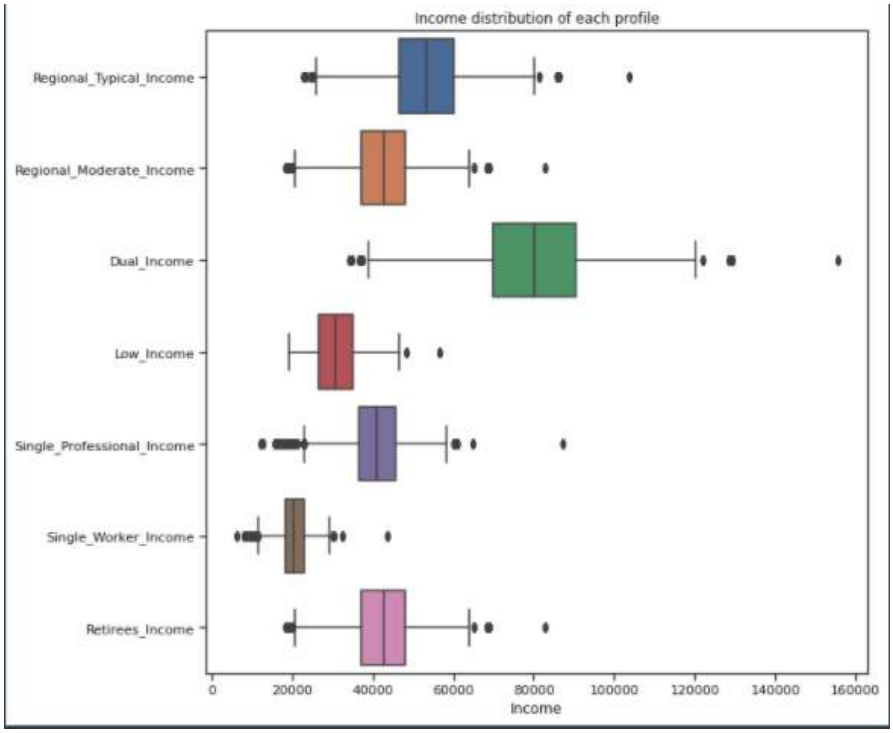
See technical documentation on HUD website for more information on household types: Location Affordability Index v.3 page.

The dataset provides 13 columns that contain data unique to each household classifcation including:

- Household Income
- Household Size
- Household Number of Commuters
- Household Housing-Transportation Costs
- Household Housing Costs
- Household Owners Housing-Transportation Cost
- Household Owners Housing Costs
- Household Renters Housing-Transportation Costs
- Household Renters Housing Costs
- Household Transportation Costs
- Autos Per Household
- Annual Miles Per Household
- Annual Transit Trips Per Household

For data cleaning and processing we executed the following steps:

- The data for the Single Very Low Income (type 5) had little variability due to the fact that is based off of the national poverty level, which is consistent across all states, and therefore was not helpful in the analysis so it was removed. `<data_cont = data.drop(data[(data['State']=='Hawaii') | (data['State']== 'Alaska') | (data['State']== 'District of Columbia')].index)>`
- The data in states like Hawaii, Alaska, and Washington DC were also removed due to being outliers in population density and other factors. `<data = data.drop(data.columns[69:82], axis=1)>`
- Filtered for NaN Data `<data = data.dropna()>`



Experimental Design

The following is our step-by-step process from obtaining the data to finding results.

1. Obtain datasets from Department of Housing and Urban Development (HUD) (United States Department of Housing and Urban Development, 2019) and Census Bureau (United States Census Bureau, 2017) websites via webscraping download links.
2. Read the two CSV files into 2 pandas (McKinney, W., & others. 2010) dataframes in Python
3. Merge county names column from the Census Bureau into the HUD dataset on the FIPS code column.
4. Examine & clean the data
5. Query and analyze the data (including Unit Testing of our code) in the following ways:
    - Visualizations
    - Descriptive Statistics

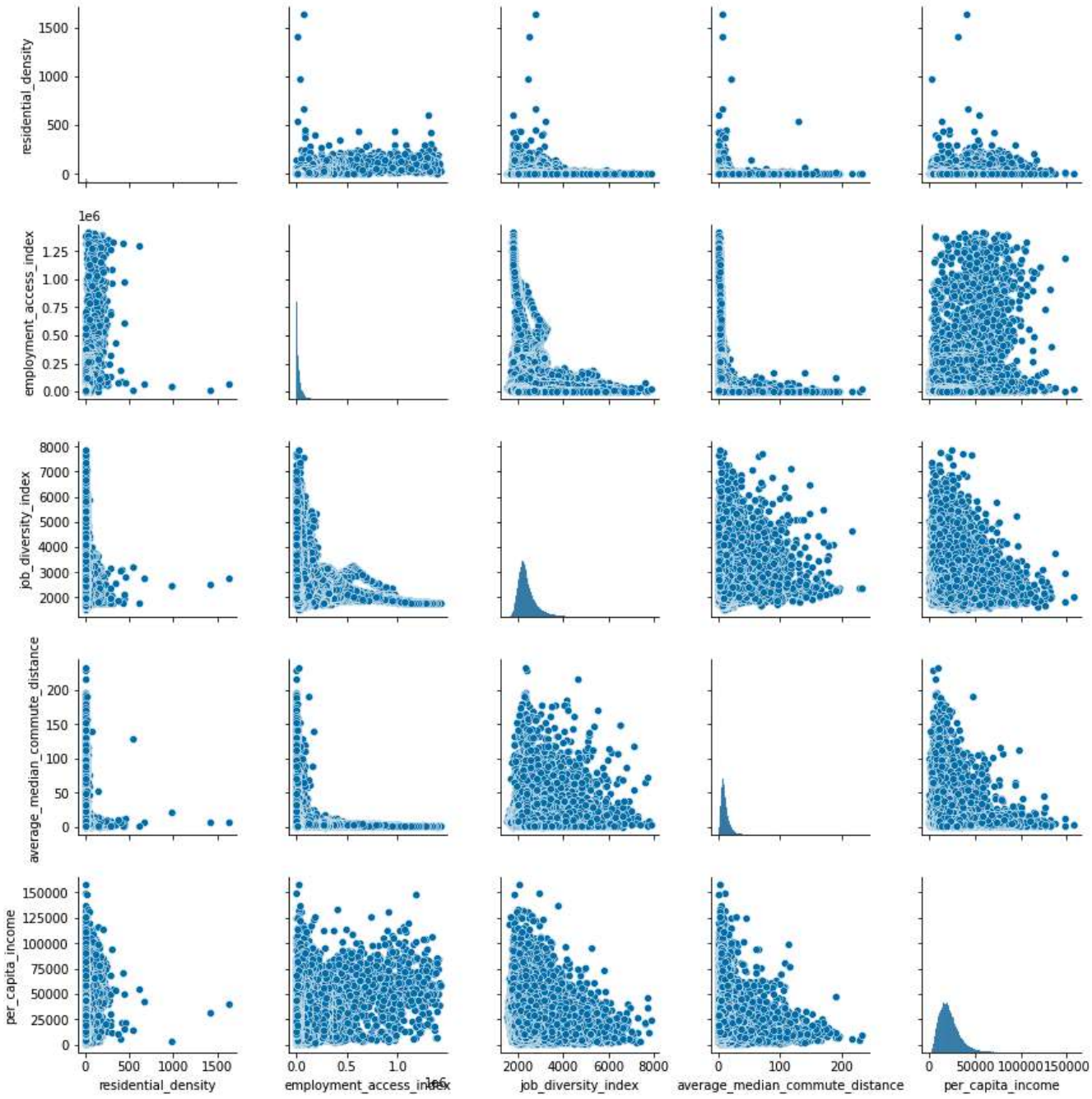
Once the analysis was complete, we used our knowledge to build an application to visualize affordability based on user input. See section Beyond the Original Specifications.

Results

There are multiple fields that are characteristics of the locations:

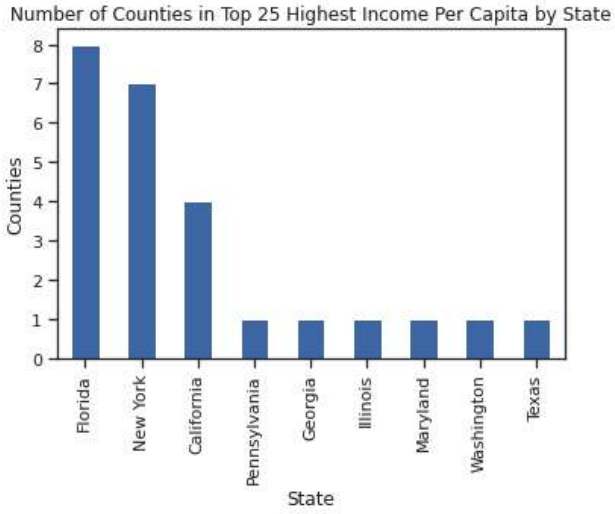
- residential density
- employment access index
- job diversity index
- Median commute distance
- Per capita Income

Per capita Income *This does not have a high correlation with any of the other variables which seems to indicate there is income avaiability regardless of things like employment access, job diversity, and residential density

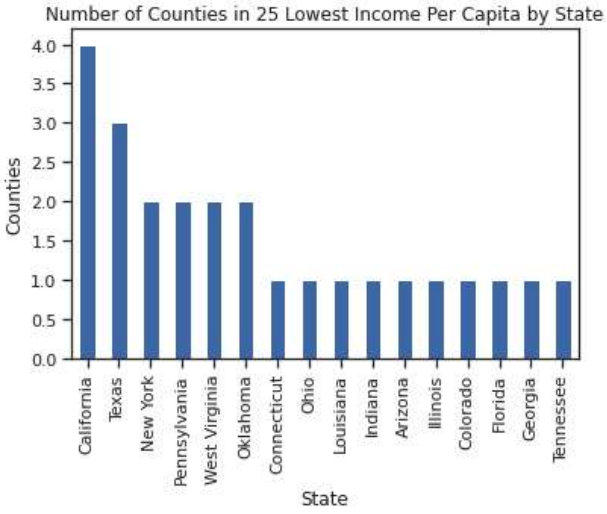


Highest Per Capita Income -The top 25 areas with the highest income appear to be concentrated in a few states.

- FL - 8
- NY - 7
- CA - 4



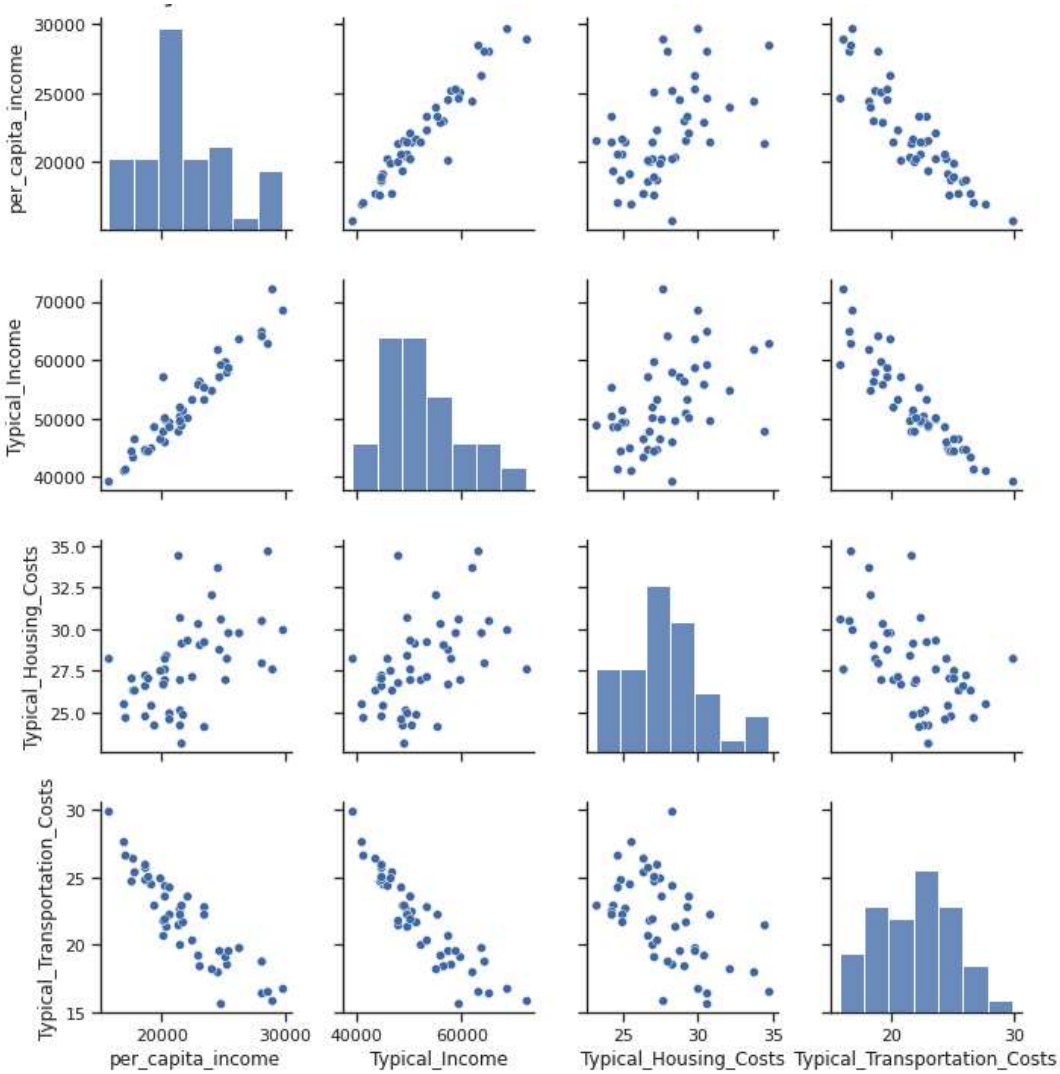
Lowest Per Capita Income - The lowest 25 areas with smallest income are more spread out across more states.



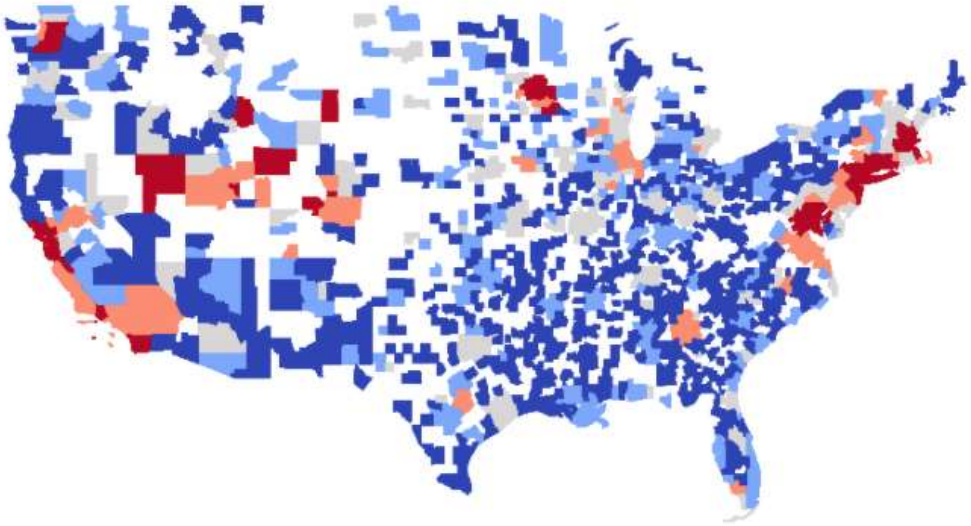
Average of Per Capita Income, Typical Income, Housing Cost, Transportation Cost:

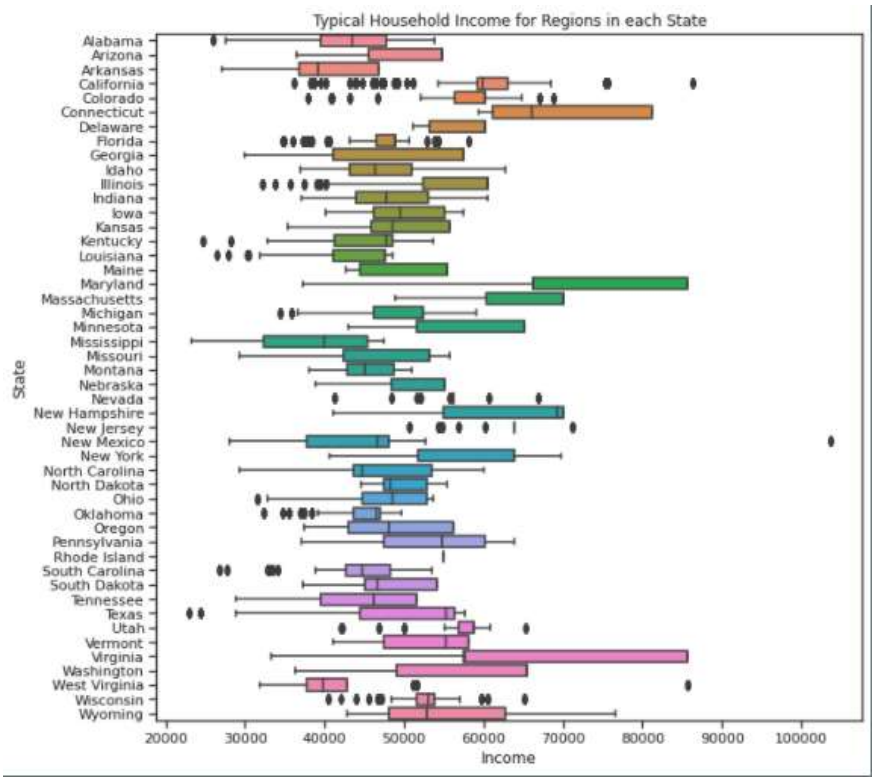
- Transportation Cost vs. Income & Housing
  - Clear negative correlation between Transportation cost and income (-.917433) and housing cost (-.532232).
  - Houses further from cities tend to be cheaper so more cost in transportation
  - Higher percentage of income dedicated to trasnportation cost vs. equity in a house

	per_capita_income	Typical_Income	Typical_Housing_Costs	Typical_Transportation_Costs
per_capita_income	1.000000	0.957106	0.505561	-0.899523
Typical_Income	0.957106	1.000000	0.462198	-0.917433
Typical_Housing_Costs	0.505561	0.462198	1.000000	-0.532232
Typical_Transportation_Costs	-0.899523	-0.917433	-0.532232	1.000000



Wealth appears to be concentrated on the coasts and higher in mid-atlantic and Northeast.





Beyond the Original Specifications

For this project, we wanted to create a tool that provided the user to visualize the affordable locations in a given area. To accomplish this we created an application composed of two classes:

- class LocationAffordabilityIndex
  - Application class which contains the dataset stored in a pandas (McKinney, W., & others. 2010) dataframe and different methods to allow the user to manipulate and visualize this data, including the creation of a user-specific location affordability index.
- class LAIUser
  - Representation of the user's profile including desired state, income, work status, household profile, and transportation profile.

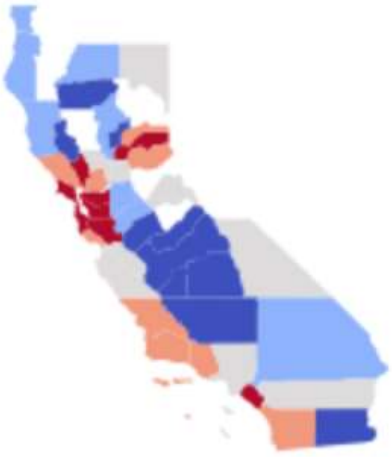
The application begins by creating an instance of the LocationAffordabilityIndex using data scraped from the download links, or from a local file. It then calls a method to prompt the user to supply information to build their profile. It then classifies the user based on their input and returns a pandas (McKinney, W., & others. 2010) dataframe that contains the affordable areas for the user to live in. This dataframe was then visualized using geopandas (Jordahl, K. 2014), which uses the state and country FIPS codes from the LocationAffordabilityIndex dataframe, and combines them with a county shape file that allows for visualization of the affordability of each location.

```
# 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)
```

This is an export for California based resident making $100,000. The chart shows the least affordable locations based on income per capita.

	county	State	residential_density	employment_access_index	job_diversity_index	average_median_commute_distance	per_capita_income
0	Marin	CA	3.539425	17822.673815	2207.741580	14.055059	42372.507557
1	San Mateo	CA	6.287540	36876.134590	2025.197985	11.111265	36793.876572
2	San Francisco	CA	23.346612	146367.492785	2404.598999	4.193474	35209.771190
3	Santa Clara	CA	5.664153	44888.156073	2059.957132	9.205184	33911.361850
4	Contra Costa	CA	4.249327	22707.919295	2201.025335	16.276580	31348.426644
5	Napa	CA	3.254543	12182.007951	2645.502728	13.297292	30072.470345
6	Placer	CA	2.450399	12713.882087	2282.069285	21.683346	29597.735996
7	Orange	CA	5.924068	57044.334551	2161.988014	11.350541	28864.357539

Red represents highest median income and blue is lowest



Testing

The following testing was conducted on this project:

1. Testing on the LAIUser class to ensure that the user profile is correctly set-up based on the provided user input.
2. Testing on the LocationAffordabilityIndex class to ensure that the application is properly intialized from the supplied data source (scraped or from local file) and that the methods for assigning a user profile object (LAIUser) and returning the correct result dataframe are written correctly.

Sample output from our testing can be seen below:

```
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')
```

```
# 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
```

Conclusions

1. Key Findings and Use Cases

    i. Income is negatively correlated with Transportation costs (-.917433) meaning wealthier people tend to live closer to the city center and have less transportation costs overall.

    ii. Transportation is also negatively correlated with housing cost (-.532232) just not as closely correlated. Cheaper houses are further out which then require more transportation cost.

    iii. There are certain states that tend to have pockets of some of the wealthiest areas like Florida, New York, and California.

    iv. Regions like Mid-Atlantic and Northeast are wealthier versus many southern or central states.

    v. This tool could be very useful to users that are interested in relocating to different areas of the country for work or personal reasons. It could also be useful to public officials that are interested in visualizing the affordable locations in their district, in order to promote new residents moving to those areas.

2. Future Improvements

    i. Find more recent data to do a comparison of how specific metropolitan areas fluctuated since 2016.

    ii. Create a more accurate user input system to map users to the optimal household type. Currently we only map to 3 of the 8 household types.

    iii. Utilize algorithms to identify similar affordability areas close to the user supplied desired location.

    iv. Create affordability mapping down to the census tract level. Currently we only map to the county level, but if different shape files were acquired and utilized, we could map affordability to the census tract level.

Bibliography

Jordahl, K. (2014). GeoPandas: Python tools for geographic data. URL: Https://Github. Com/Geopandas/Geopandas.

McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).

United States Department of Housing and Urban Development. (2019). Location Affordability Index V3.0. United States Department of Housing and Urban Development. https://www.hudexchange.info/programs/location-affordability-index/

United States Census Bureau. (2017, May 5). 2016 FIPS Codes. https://www.census.gov/geographies/reference-files/2016/demo/popest/2016-fips.html