ETL Project

Final Report

—

Omari Blockton

Nicholas McCarty

Hypothetical Use Case

In order to compare housing affordability by region for a specified time period (i.e., 2018), it is useful to create an index. One such index could be the ratio of a selected proxy for property value to a selected proxy for per capita income. That ratio can easily be calculated and visualized once all of the requisite datasets are in a centralized database.

Data Sources

As mentioned above, the two features required to arrive at an 'affordability index' by region for a specified time period are **house price index** and **regional income per capita** for said region. In this case, we will be looking at these features by state for 2018, and our data sources are:

- Federal Housing Finance Agency (FHFA) house price index (source)
- Bureau of Economic Analysis (BEA) per capita regional income (source: BEA API)
- Wikipedia state names and abbreviations (source)

Data Transformation

House Price Index

The house price index is available for download from the FHFA website as either a .txt or an .xls file. We chose to simply download the .xls file and save it as a UTF-8-encoded .csv file to be read into memory using Pandas. Upon doing so, the resulting dataframe is returned with the following structure:

```
housing_price_index
```

	state	yr	qtr	index_nsa	index_sa	Warning
0	AK	1991	1	100.00	100.00	* Note that this state has fewer than 15,000 t...
1	AK	1991	2	100.80	100.18	* Note that this state has fewer than 15,000 t...
2	AK	1991	3	102.05	101.16	* Note that this state has fewer than 15,000 t...
3	AK	1991	4	102.21	102.15	* Note that this state has fewer than 15,000 t...
4	AK	1992	1	102.45	102.39	* Note that this state has fewer than 15,000 t...

After having already inspected the regional income data (described in the next section), we decided to create a column identical to the one in that dataset called **TimePeriod**, into which we would essentially insert a concatenation of the **yr** and **qtr** columns, as shown

here (shout out to Andi Rubin-Schwarz for providing such an elegant solution to the problem). We then dropped every column except **state**, **index_sa** (seasonally adjusted house price index), and **TimePeriod**, before exporting the result as a .csv file for use in the creation of our database.

Regional Income

The regional income data was pulled using the BEA API. Upon doing so, the resulting dataframe is immediately returned (thanks to the fabulously written Python BEA API wrapper, PyBEA) with the following structure:

```
regional_income_2018
```

	CL_UNIT	Code	DataValue	GeoFips	GeoName	NoteRef	TimePeriod	UNIT_MULT
0	Dollars	SQ1-3	53037.0	00000	United States	NaN	2018Q1	0
1	Dollars	SQ1-3	53411.0	00000	United States	NaN	2018Q2	0
2	Dollars	SQ1-3	53892.0	00000	United States	NaN	2018Q3	0
3	Dollars	SQ1-3	54484.0	00000	United States	NaN	2018Q4	0
4	Dollars	SQ1-3	41854.0	01000	Alabama	NaN	2018Q1	0
5	Dollars	SQ1-3	42109.0	01000	Alabama	NaN	2018Q2	0
6	Dollars	SQ1-3	42397.0	01000	Alabama	NaN	2018Q3	0
7	Dollars	SQ1-3	42956.0	01000	Alabama	NaN	2018Q4	0
8	Dollars	SQ1-3	58814.0	02000	Alaska *	*	2018Q1	0
9	Dollars	SQ1-3	59182.0	02000	Alaska *	*	2018Q2	0

In addition to immediately identifying several columns that we would want to drop, we noticed that the listed **GeoName** for Alaska was followed by a space and an asterisk. Upon inspecting the unique values in the **GeoName** series, we discovered that all observations for Hawaii were recorded similarly. Consequently, we reformatted the **GeoName** series thusly. We then dropped every column except **DataValue** (per capita regional income), **GeoName** (state name), and **TimePeriod**, before exporting the result as a .csv file for use in the creation of our database.

State Names and Abbreviations

The state names and abbreviations were read into memory using Pandas' read_html method. The resulting dataframe was returned with the structure seen at the top of the next page. Clearly, we had some cleaning up to do, which we did thusly. The result was exported as a .csv file for use in the creation of our database, which will be described next.

```
states
```

	0	1	2	3	4	5	6	
0	Codes: ISO ISO 3166 codes (2-letter, 3-letter...	Codes:	ISO	ISO 3166 codes (2-letter, 3-letter, and 3-digi...	ANSI	2-letter and 2-digit codes from the ANSI stand...	USPS	2-letter c used by the U States Po
1	Codes:	NaN	NaN	NaN	NaN	NaN	NaN	
2	ISO	ISO 3166 codes (2-letter, 3-letter, and 3-digi...	NaN	NaN	NaN	NaN	NaN	
3	ANSI	2-letter and 2-digit codes from the ANSI stand...	NaN	NaN	NaN	NaN	NaN	
4	USPS	2-letter codes used by the United States Posta...	NaN	NaN	NaN	NaN	NaN	
5	USCG	2-letter codes used by the United States Coast...	NaN	NaN	NaN	NaN	NaN	

Database Creation

We chose to create a relational database using MySQL Workbench, due to the fact that each dataset has either a state name or abbreviation as a feature. This would allow us to join the tables together using our freshly-created table of state names and abbreviations. Upon creating the database, we were then tasked with writing scripts to create the requisite tables before using the data import wizard to populate them with the .csv files we previously exported after performing the necessary transformations. Finally, we wrote a query to join the house price index and regional income data for 2018 (by state) together using state name and abbreviation and **TimePeriod**.

House Price Index Table

The house price index table in our housing database was created using this script.

Regional Income Table

The house price index table in our housing database was created using this script.

State Names and Abbreviations Table

The house price index table in our housing database was created using this script.

Join Query

The house price index table and regional income table were joined using this script.

Final Result

The result was exported as a .csv file to be read into memory for final formatting using Pandas. These steps can be seen at the end of our Jupyter Notebook, which was used to document the workflow. The resulting dataframe is structured as shown below (the result also having been exported as a .csv file, for good measure) and can be used for further analysis.

```
pd.read_csv('Joined Output (Transformed).csv', index_col = 0)
```

	Housing Price Index	Regional Income	State	Yr. & Qtr.	Affordability Ratio
0	213.84	41854	Alabama	2018Q1	0.510919
1	219.90	42109	Alabama	2018Q2	0.522216
2	222.40	42397	Alabama	2018Q3	0.524565
3	223.41	42956	Alabama	2018Q4	0.520090
4	214.66	42073	Arkansas	2018Q1	0.510208