ETL processes.

Data source:

Geojson - These data contain all 50 states with zip code as boarders. The data is extracted from <https://github.com/OpenDataDE/State-zip-code-GeoJSON>. This data is use in conjunction with Zillow data to map out housing prices.

Zillow api – We use this api to extract US housing price data per zip code. This data is merge with the Geojson data. This will allow us to create a heat map that outlines various area base on median single-family home.

Census api – We use this api to extract US population data base on zip code. This data is merge with the Geojson data and Zillow to have a complete picture of the Population demographics and housing prices.

Processes:

Extract, Transform:

\*\*\*geopandas is require for this process. Otherwise the data will not merge correctly\*\*\*

1. Extract all data from the above data sources
2. Clean up column replace any spaces on the column name with underscore
3. Rename any column with zip code to ZCTA5CE10

\*\*Note\*\* Step 3 may not be necessary but since we are down a person and had no time to test. We just assume geopandas will be looking for that column, so we opted rename all zip code column.

1. Run merge.py to merge the 3 data into one large json file
   1. Read data from geojson file of state with geojson
   2. Read Zillow csv with pandas
   3. Read census data with pandas
   4. Create a data frame for both Zillow data and census data
   5. Convert data type to string for column (ZCTA5CE10) \*\*without this step data will not merge\*\*
   6. Merge Zillow data frame to geojson data
   7. Convert data type to string for column (ZCTA5CE10) in cenus dataframe
   8. Merge census date frame to geojson data
   9. Write new file with Geojson driver. Not using this will cause issue
2. Load data into postgres database (json2postgres.py)
   1. Read the merge json file (ga\_merge.json)
   2. Normalize the json data
   3. Clean up data by dropping geojson columns such as geometry. Coordinates
   4. Create an engine to connect to postgres db
   5. Use the .to\_sql method to to upload the database table
   6. Clear connection
3. Generate visual with data (sqltodf.py)
   1. Create an engine to connect to the database
   2. Use the .execute method get data( select \* form ga\_merge)
   3. Run the .execute method again with .key() \*\*Note\*\* Need both data and header. Otherwise you will numbers on your column name
   4. Normalize the data
   5. Create a data frame for the data
   6. Rename the column from the results form .key() method
   7. Create a data frame with poverty rate of over 20 percent
   8. Run .dropna() no get rid of NaN
   9. Remove crazy numbers such as (-666666666)
   10. Get data frame stats. Use for sanity check
   11. Create scatter plot with projection of housing prices of population with over 20% of poverty rate for the last 4 years.
   12. Create 4 data frame with median housing price from the above data
   13. Merge and create a bar graph to compare data
   14. Re-run steps g-m for poverty rate under 10 percent (the rich people)
   15. Now compare the 2 bar charts

Visual map layering

1. Need the following js preloaded – choropleth.js, leaflet-heat.js
2. Logic.js
   1. Create a monthly payment function to check for mortgage
   2. Create a createmap function to generate background maps
   3. Create legend for the map and load
   4. Load geojson data to d3.json to create function that will display city, county name, poverty rate into pop up.
   5. Create list with data need to do mortgage math
   6. Create heat map layer
   7. Create popup for both layers