## ETL project: Mark Clayton and Jenny Bui-

For our project, we decided to tackle US Income rates and US representation political party. All the information was separated based on State/zip code.

## Data Sources

The extraction process begins with the datasets that we gathered. Our sources were from Kaggle:

<https://www.kaggle.com/jayrav13/us-representation-by-zip-code>

<https://www.kaggle.com/goldenoakresearch/us-household-income-stats-geo-locations?select=kaggle_income.csv>

## Data Clean-up

The datasets included a CSV and JSON file in which we took to Python/Pandas to transform in our next step.

**kaggle\_income.csv > clean\_income\_db (Project 2 Data Clean-up Notebook)**

The first step of the data cleaning process was to load the kaggle\_income.csv into a jupyter notebook. We then created it into a Pandas DataFrame and reviewed the columns and content to determine what would be needed for our database. We then pulled out the 8 columns we wanted vs all 19 that were available. We then removed rows based on the content from the income columns that was 0 and only pulled the last of the unique zip code entries since it can have multiple cities with the same zip. We then saved the DataFrame as a csv file to use in PostgresSQL.

**us\_representation > officials complete (json with T Notebook)**

The json was much harder to clean then we expected. First step was to load the json file into the notebook. When trying to run such a large dataset of 228MB with 30k rows we had to learn to open jupyter notebook without the normal data limits (jupyter notebook -- NotebookApp.iopub\_data\_rate\_limit=1.0e10) for it to even work. We then turned the data into a DataFrame and reset the index so that the Zip Code was a column. Then there was the complication that in order to get the political party and officials we wanted were nested in each officials column cell. We then had to view the nested details to determine how we can pull out the info we want. Once we were able to identify which key data we needed we created a list to hold the info and pulled out the name and political party. That allowed us to create a new DataFrame “officials” that we then turned into a csv to use in PostgresSQL. During this clean-up process I tried a few different things that can be seen in the json fails folder of github. Using things like json\_normalize to figure out how to un-nest the information.

When loading these items into github we also ran into a problem because the file size limit is 100MB typically. We had to download git lfs in our terminal, track the files and have a gitattributes file. Once you set it up, you can push and pull like normal, it just takes a little more time.

## Database (PostgresSQL)

The database in PostgresSQL was added and used CREATE TABLE to get the framework from each of the 2 csv. Then we used the import option to bring in the values of the tables. Once both were created, we used INNER JOIN by zip code to view the tables together. We then used different combinations of queries to review the information for what we needed.

## Data Limitations

There were some limitations to our data. For example, we only had one variable that could be tested since there is no direct relationship between income and presidential party. We also only did the United States since the Income did cover other countries, but the US representation did not. Due to the different data sources, when joining it was only able to match to 40 of the 50 states.

## Conclusion

In the end we were able to see some trends and correlations between average income and Political Party. For example, the Top 10 states for average income had primarily Democratic Representation for 8 states and Primarily Republican for 2 states. Republicans primarily serve in the highest total states 26 of the 40 in our data. We could continue to manipulate this data and find percentages of political party to be more accurate.

|  |  |  |  |
| --- | --- | --- | --- |
| state\_name | avg | count | party |
| Alabama | 42162.54 | 4383 | Republican |
| Arizona | 36263 | 7 | Unknown |
| Arkansas | 39523.14 | 2244 | Republican |
| Connecticut | 59663.78 | 1474 | Democrat |
| Delaware | 53999.94 | 239 | Democrat |
| District of Columbia | 60209.35 | 100 | Democrat |
| Florida | 45622.09 | 5464 | Republican |
| Georgia | 44609.74 | 4772 | Republican |
| Illinois | 48067.86 | 3924 | Democrat |
| Indiana | 43700.6 | 3460 | Republican |
| Iowa | 44324.03 | 2799 | Republican |
| Kansas | 43128.89 | 2901 | Republican |
| Kentucky | 40241.26 | 2580 | Republican |
| Louisiana | 45579.53 | 2745 | Republican |
| Maine | 47677.4 | 656 | Republican |
| Maryland | 58819.31 | 1218 | Democrat |
| Massachusetts | 57753.63 | 1632 | Democrat |
| Michigan | 43720.28 | 4568 | Republican |
| Minnesota | 47650.52 | 2413 | Democrat |
| Mississippi | 40068.43 | 2114 | Republican |
| Missouri | 40430.34 | 4041 | Republican |
| Montana | 45505.52 | 692 | Republican |
| Nebraska | 42484.56 | 1736 | Republican |
| New Hampshire | 54712.28 | 747 | Republican |
| New Jersey | 61827.86 | 2903 | Republican |
| New York | 54921.03 | 5664 | Democrat |
| North Carolina | 43497.94 | 4359 | Republican |
| North Dakota | 47613.96 | 1428 | Republican |
| Ohio | 43126.19 | 6051 | Republican |
| Oklahoma | 41896.12 | 1213 | Republican |
| Pennsylvania | 45179 | 5172 | Democrat |
| Rhode Island | 52888.1 | 338 | Democrat |
| South Carolina | 41958.47 | 2829 | Republican |
| South Dakota | 41290.93 | 1257 | Republican |
| Tennessee | 42461.48 | 2030 | Unknown |
| Texas | 50972.59 | 1275 | Republican |
| Vermont | 47601.93 | 442 | Democrat |
| Virginia | 51743.49 | 1992 | Democrat |
| West Virginia | 41507.85 | 1166 | Republican |
| Wisconsin | 46015.06 | 2665 | Republican |