# **ETL Project Report**

**Title:** GDP, Unemployment and Health Insurance Analysis for 2008-2018

**Members**: Ju A Han, Lakshmi Prasanna Rangam

### Extraction

#### Data Sources:

We used the three datasets from three different websites. We collected the data by each county in the States from 2008 to 2018. In terms of the timeframe, the plan was to collect data for recent 10 years, but we decided to extend one more year so that we include the great recession data in the analysis.

* GDP growth Data: Bureau of Economic Analysis (<https://www.bea.gov/data/gdp/gdp-county-metro-and-other-areas>)
* Unemployment: US Bureau of Labor Statistics (<https://www.bls.gov/lau/#cntyaa>)
* Health Insurance Coverage: US Census Bureau (<https://www.census.gov/data-tools/demo/sahie/#/>)

#### Data Format:

Datasets from the US Census Bureau, Bureau of Economic Analysis and US Bureau of Labor Statistics are in csv format. We downloaded a total of 15 files from the three websites.

### Transformation

We used *pandas* for the transformation of data. Main types of transformation used are filtering, renaming, splitting, dropping, merging, etc.

All the datasets had a column in common, “county-state”, which will later be used for merging. However, they were not in the same format, so we needed to modify them using replace() method. Moreover, all three datasets had NA values for some counties since they were newly created or removed during the years. We had to fill the NA values with zero to make sure that no error occurs later. Finally, we dropped all the unnecessary columns and rows in the datasets.

#### 

#### 

#### GDP Dataset:

1. Downloaded two csv files.

A screenshot of a social media post

Description automatically generated

Figure 1 GDP-Data: Reading csv file

1. Dropped a few rows and columns that aren’t needed.
2. Replaced a few NA values with zeros in both csv’s.

A screenshot of a cell phone

Description automatically generated

Figure 2 GDP-data: Dropping Rows and Replacing NAs

1. Split the county-state column into two different columns county and state.
2. State abbreviations from the state column have some asterisk symbols that need to be removed.

A screenshot of a cell phone

Description automatically generated

Figure 3 GDP-Data: Creating County and State Columns

1. Renamed the columns as we needed.

A screenshot of a social media post

Description automatically generated

Figure 4 GDP-Data: Renaming and Reordering

1. Two csv’s are merged into the final gdp dataframe.

A screenshot of a cell phone

Description automatically generated

Figure 5 GDP-Data: Merging

#### Health Insurance Coverage Dataset:

1. Downloaded one csv file which contained all the data

A close up of a piece of paper

Description automatically generated

Figure 6 Health-Insurance-Data: Reading CSV

1. Dropped the state-level data

A screenshot of a cell phone

Description automatically generated

Figure 7 Health-Insurance-Data: Dropping State-level Data

1. Extracted the yearly data into each data frame

A screenshot of a social media post

Description automatically generated

Figure 8 Health-Insurance-Data: Extracting Yearly Data

1. Replaced NA values with zero
2. Split the county-state column into two different columns county and state.
3. Renamed the columns
4. Merge them since the yearly data was not stated in columns but in rows

A screenshot of a social media post

Description automatically generated

Figure 9 Health-Insurance-Data: Merging and Renaming

#### Unemployment Dataset:

1. Downloaded 11 csv files for yearly data.

A screenshot of a social media post

Description automatically generated

Figure 10 Unemployment-Data: Reading CSV

1. Dropped unnecessary columns and rows.
2. Replaced NA values with zero.
3. Split the county-state column into two different columns county and state.

A screenshot of a social media post

Description automatically generated

Figure 11 Unemployment-Data: Dropping, Renaming, Extracting

1. Merged datasets and renamed the columns.

A screenshot of a social media post

Description automatically generated

Figure 12 Unemployment-Data: Merging and Renaming

### Loading

#### Data Storage:

We used a relational database (PostgreSQL) to store and link the three datasets using County\_State column as our common identifier. We used PostgreSQL over MongoDB as PostgreSQL stores data in table and supports relational databases while MongoDB stores data like documents and does not support well-defined relationships. Using MongoDB may end up with having a lot of duplicate data which might result in having corrupted data.

#### Data Loading:

Finally, we created a connection from SQL database, and loaded the final data in a jupyter notebook using *SQLAlchemy*.