INSTRUCTIONS: Once you have identified your datasets, perform ETL on the data. Make sure to plan and document the following:

The sources of data that you will extract from.

The type of transformation needed for this data (cleaning, joining, filtering, aggregating, etc).

The type of final production database to load the data into (relational or non-relational).

The final tables or collections that will be used in the production database.

You will be required to submit a final technical report with the above information and steps required to reproduce your ETL process.

Project Report

At the end of the week, your team will submit a Final Report that describes the following:

Extract: your original data sources and how the data was formatted (CSV, JSON, MySQL, etc).

Transform: what data cleaning or transformation was required.

Load: the final database, tables/collections, and why this was chosen.

Please upload the report to Github and submit a link to Bootcampspot.

ETL Project: Heather Leek, Theodore Moreland, Adam Feldstein

**Data Sources**

<https://www.kaggle.com/cityofLA/la-restaurant-market-health-data>

<https://www.yelp.com/developers/documentation/v3>

<https://www.yelp.com/developers/documentation/v3/business_search>

<https://www.yelp.com/developers/documentation/v3/business_reviews>

<https://api.yelp.com>

The Kaggle data source provided a list of all restaurants in Los Angeles, CA including their name, address and health inspection information such as the health grade and health grade percentage. The Yelp API was utilized to pull LA restaurant reviews and star rating for 1,000 restaurants.

**Data Cleanup & Analysis**

**EXTRACTION**

Data was extracted from the data sources listed above. The team utilized restaurant\_inspections\_original.csv file from the Kaggle which was downloaded directly in the CSV format. Additionally, the team utilized the Yelp API to call data for 1,000 LA restaurants (data set limited per the API). All API data was called in a JSON format, loaded into a dataframe and then output into a CSV file.

**TRANSFORMATION**

The first transformation was performed on the restaurant\_inspections\_original.csv file. The file contained 20 columns. The entire file was read into a Jupyter notebook. The OS path, Pandas and CSV libraries were imported for use into the Jupyter notebook. The entire file was read into an inspection dataframe. The team discerned that only eight columns within the file were relevant to the analytics process. Therefore, the team created a variable to rename eight columns through a dictionary that contained the old column name and a new column name. The new column names were more descriptive of the data being utilized. For example, facility name was renamed restaurant name.

The new column names were inserted into the existing inspection dataframe. Next, the eight columns were inserted into a list that was later utilized to reduce the inspection dataframe to a new inspection cleaned dataframe with eight columns rather than twenty.

Next the team dropped all duplicates in the file as the original dataset contained multiple inspection histories for each restaurant. Only the first inspection record for each restaurant was kept within the dataframe as the original file was organized with the latest inspection first. Finally, the team cleaned the zip code field by keeping only the first five digits as the original file contained a mix of five digit and nine-digit zip codes. The last step in the transformation of the inspection cleaned dataframe was to output it to a clean CSV file (restaurant\_inspections\_clean.csv).

The next transformation was performed on the Yelp restaurant and review data that was extracted through the API calls. For the Yelp reviews and ratings data, the team imported the OS, Pandas, JSON, Api\_key, Requests and Rate Limit libraries into the Jupyter notebook. The team had to utilize an API key to perform the API call within the extract.

Data from the API was stored within two dictionaries. The restaurant dictionary stored the restaurant id, name, rating, review count, address, city, state and zip. The restaurant review dictionary stored the restaurant id, name, star rating, 3 randomly selected reviews and the date time for each review.

Data from the API calls was appended within each of the two dictionaries. Next, the data from each dictionary was put into a Pandas dataframe. Finally, the team dropped any duplicates in each dataframe and exported the contents of each dataframe into a CSV file. Two CSV files were output including yelp\_reviews.csv and yelp\_la\_restaurants.csv.

**LOAD**