**ETL Project**

Cicy Chen

Rob Stewart

***OBJECTIVE***

Often times it can be challenging to find all the data one needs in order to make informed investment decisions. That being said, our objective is to provide prospective investors with a single database from which they can pull *both* qualitative and quantitative stock related information. To do this, we created a single database that contains quantitative and qualitive stock data so that we, as investors, have more robust data to analyze before making investments decisions.

Our datasets include information related to stocks that are included in the S&P 500 Index as of 10/02/2019.

**EXTRACT**

We sourced our quantitative data from Yahoo Finance, which included stock data such as price, market cap, P/E Ratio, Beta, etc. Our Yahoo Finance dataset was formatted in a CSV file.

Our qualitative data was sourced from a Wikipedia page, from which we copied and pasted data from a table into a CSV file. Content for this dataset include stock symbol, stock name (long-form), and sector.

***TRANSFORM***

We began the exciting part of the project by reading in our CSVs with Jupyter Notebook. After we read the datasets, we quickly realized that we needed to make minor adjustments to one of the two datasets.

We created our first dataframe by only pulling in 50% of the columns from the original *quantitative* dataset, as we found much of the data to be useless when determining if stocks are worthy of our investment. Given the sensitive nature of using python *and* SQL, we found it would be best to change our column headers to be lowercase. In similar vein, we decided to remove all instances of ‘commas’ from price. Lastly, we assigned more appropriate datatypes to each of our columns, as all of the columns were originally being read as ‘objects’.

Fortunately for our group, our second dataset was rather clean and did not require any additional transformation, so we were able to read in the CSV and create our second dataframe.

***LOAD***

Our final steps of the project required us to push our two pandas dataframes into two separate SQL tables. To begin, we created two tables in postgres, and then used python to send our pandas dataframes to the applicable tables in postgres. Lastly, in postgres, we used an inner join to combine our two tables containing quantitative and qualitative stock data so that the two different types of datasets can be housed in a single database.