ETL Final Report

Analyzing Stocks in the S&P 500

First, I scraped Wikipedia using pandas to get a table of all the stocks currently in the S&P 500 index. I then went to various websites intending to use an API and pull more data. However, most websites only give us 500 free pulls and the S&P 500 is obviously 500 stocks, hence, I’d use up my allotment on the first pull. Fortunately, the website Barchart.com produces daily csv files that are available for download for free.

After the web scraping and csv file uploads, I ended up with five files to which I loaded in to a pandas dataframe. Each file contained a stock symbol and other unique data including Fundamental data, Technical data, Performance data, Trading data and Descriptive data.

Before the data could be exported, I had to clean it up. So, I performed the following:

* I dropped the last row of each csv file to remove a note that was added from the website.
* I dropped certain columns that were not needed.
* I changed the column names to remove certain characters that PGAdmin did not like.
  + This involved some trial and error. Column heading had to be all lowercase, no spaces, numbers to start or symbols.
* I dropped a row in one of the csv files that contained Nan’s
* I dropped commas and “%” signs so that the columns in the tables could be created as floats and are now easy to sort.
* Finally, I had to make sure I dropped the index when exporting or I’d get an error message.

Once the data was cleaned, I set up a new database in PGAdmin called Stocks and created tables for each of the five dataframes using python to export all the data from Jupyter Notebook to PGAdmin. The tables are:

* SP\_Wiki which contains qualitative data.
* SP\_Main which contains pricing data
* SP\_Fundamental which contains fundamental data including PE ratios and dividend yields
* SP\_Performance which contains YTD and 52-week performance data
* SP\_Technical which contains 52 week High and Low price.

After transforming the data, my goal was to export it in to a relational database since the data is uniform in nature and can be easily cross-referenced or “joined” on the column “Symbol” in each table as a primary key.