**ETL Project**

**Group 12 team members:**

Ming Lei

Denise Person

**Objective**

Create a database that analysts can use to determine what affect the shape of the yield curve has on economic growth (represented by GDP.)

**Extraction**

Two data sets were used for this project.

Monthly yield curve data for the period June 1990 to June 2019 was obtained from the U.S. Department of the Treasury. Data for the period 1970 to 2019 was sought; however, yield curve data earlier than 1990 was not available. The data was scraped off the website which presented a tabular format for each year’s data.

Quarterly real and current GDP growth rates for the second quarter of 1947 to the first quarter 2019 were obtained from the U.S. Bureau of Economic Analysis. This data was available in CSV format.

Pandas was used to read the data into two separate dataframes using Jupyter Notebook.

**Transformation**

The first step of the transformation process was to delete any columns and rows that were not needed. Columns and rows deleted included summary information about the data set and NaN values. Deletion of the summary information was completed in EXCEL prior to reading the data into a Pandas dataframe. Formatting of date values were adjusted for both data sets as the dates were going to be used to join the two data sets.

**Yield Curve Data Set**

The yield curve data was scraped from Treasury.gov which offers a visual table for each year’s data set. While there is an API for Treasury.gov, it is horribly documented with no specific API for treasury yield curves.



See comments in Jupyter Notebook Yield\_Curve\_Scraping.ipynb file.

The result of this scrape is already produced in the yield.csv file.

The schema is provided in Yield\_Table\_Schema\_SQL.sql for copy paste into whatever DB you wish.

The initial dataframe of the GDP data is in the table below.

**GDP Data Set**

**Initial GDP DataFrame of CSV File**



In the GDP data set, the column headings were the quarterly date values. As the dates were going to be used to join the two data sets, the GDP dataframe was transposed so that the quarterly date values were in a column.

**Transposed GDP DataFrame**

****

The transposed GDP dataframe, which included other economic data in addition to GDP, was filtered with a new dataframe being created that included only the date, real GDP, and current GDP columns.

**Filtered GDP DataFrame**

****

**Load**

The final step was to load the data into a Database. The yield curve and GDP Pandas dataframes were imported from separate Jupyter Notebook files to populate the tables of two separate databases created in SQL.

You can combine the two sets of data by running a join of the two sets by date (which are both varchar(10) datatype) in SQL:

**select \* from GDP**

**join yield**

**on GDP.date = yield.date;**

or you can connect to the DB and join them as Dataframes in Python using Pandas:

**merge\_table = pd.merge(gdp\_df, yield\_df, on="date")**

We used SQLAlchemy to connect to both SQL databases and created a combined file from which queries can be performed.

**Instructions to replicate:**

Go to https://www.bea.gov/data/gdp/gross-domestic-product and download ??????

Open and run all lines of code in **gdpQ.ipynb** in Jupyter Notebook

This will read in /resources/ t10101q.csv, clean it, and create

**gdpQDB.csv** in /resources to be used for importing later.

Open and run all lines of code in **Yield\_Curve\_Scraping.ipynb** in Jupyter Notebook.

This will scrape data off Treasury.gov and create **yield.csv** to be used

for importing into your tables later.

Run **Table\_Schema\_SQL.sql** in PGAdmin.

This will create a table called gdp and a table called yield in your database.

Import the **gdpQDB.csv** into the gdp table we just created using PGAdmin

Import the **yield.csv** into the yield table we just created using PGAdmin

You now have two tables in your database that you may use.