ETL Project – Final Report

Houston Statistics by Year

By David Nassif, Philip Wirth, Michael Dowlin

# Background

The group wanted to explore some of Houston’s oil history, mainly how correlated the oil price was with: population, employment and housing cost.

# Data

We pulled from two main sources for all of the data, the Federal Reserve Bank of St. Louis (FRED) and Macrotrends (website). FRED has lots of valuable datasets for various economic and demographic statistics. We used Macrotrends to get the oil price history table.

# Extraction

We downloaded the FRED datasets as csv’s. These csvs are included in the project, in the raw data folder. The Macrotrends data (oilprice\_usd) is scraped using Pandas “read\_html” function.

# Transform

The data transformation include aggregation (to 1 row per year), converting some #’s from thousands to millions (population) and renaming columns/indexes.

# Load

The data is loaded into five tables in a PostreSQL database called “houston\_statistics\_db”. We decided to use a relational database and have “year” be the primary key for each table. The data can then be combined in multiple ways, i.e. “all 5 tables together, or population and oilprice, etc.”

# Flask API

Our Flask program, app.py, can be used to pull a combined dataset (all five tables), or each table separately. The six calls can be accessed from the local host address on any browser. The index.html site lists out the addresses for the calls.

# Potential Uses

The data could be used to look for trends and correlations among any combination of the five datasets. Potential uses:

* Does a rising oil price mean: increased population, increased employment and an increase in house cost?
* Will a big decrease in oil price lower the housing cost? If so, is there a one year lag, or two?
* While we are pretty sure the Houston population has no impact on oil price, you could test that theory too.
* Which changes faster in reaction to large oil price swings: employment, housing cost or population?
* Has the impact of oil price on employment levels or population decreased/increased over time? Has Houston become more reliant on oil or less since 2000?

Schema



# Steps to Create

| **Step #** | **Step** | **Description** |
| --- | --- | --- |
| 1 | Open Postgres | The PosgreSQL service needs to be running |
| 2 | Create a PosgreSQL DB called "houston\_statistics\_db" | This will be the db used to store the data that is scraped/downloaded. |
| 3 | Run the following jupyter notebooks to import the CSVs, and scrape the oil prices | (order not important) |
| 3-a | fed\_employee.ipynb | This will import the houston\_employees.csv, which is all non-farm employees in the Houston metropoliton statistical area (MSA). The data is already by year, so the cleaning done here is column and index renaming. Data is stored in the nonfarm\_employees table. |
| 3-b | fed\_energy\_employee.ipynb | This will import the houston\_energy\_employees.csv, which is employees involved in the extraction of oil and gas in the Houston MSA. This data is also by year, and the cleaning is related to column and index renaming. Data is stored in the energy\_extraction\_employees table. |
| 3-c | fed\_housing.ipynb | This will import the houston\_housing.csv file, which has a quarterly housing cost index for the Houston MSA. The data is averaged by year, with some column/index renaming. The data is stored in the housing\_index table. |
| 3-d | oilprices.ipynb | This notebook uses the pandas read\_html function to scrape oil price history from macrotrends.net. The prices are by year, and we are only grabbing the average close price. The data is stored in the oilprice table. |
| 3-e | population.ipynb | This notebook imports the houston\_population.csv file, which has population data (by year) for the Houston MSA. There is some munging of the data, like converting the population which was reported in the thousands. The data is stored in the population table. |
| 4 | Run the app.py application | This will launch the flask application. To access, open a browser and go to the local host address |