Report that acts almost like a user guide

* Motivation for why it was gathered
* Data sources
* Justification for structure
* Use Cases of what someone could do
* No diagram, but code in Notebook and screenshots of data

ETL Project

*Motivation*

Economics is my passion. I have been concerned about government spending for as long as I can remember, and in both the 2008 Financial Crisis and the 2020-2021 COVID-19 Pandemic, government spending has accelerated rapidly. Many people across the country are very concerned about inflation, given the sheer amount of money the government and Federal Reserve have created. However, inflation is not simply a function of total dollars in circulation—there are many other factors that can apply both upward and downward pressure on the value of the dollar.

I have built a database that will serve as the foundation of a massive academic study into many of the factors that affect inflation. All of the data I use is publicly available, but much of it is hard to access, it lacks some key calculations and transformations, and it is not available in one source. I have pulled over 80 different data series, and combined them into several tables in a PostgreSQL Database.

*Data Sources*

So far, 100% of the data I have pulled has come from the St. Louis Federal Reserve’s Research Branch and the associated data warehouse, FRED (Federal Reserve Economic Data). I intend to add to the database in the coming weeks with data from other APIs.

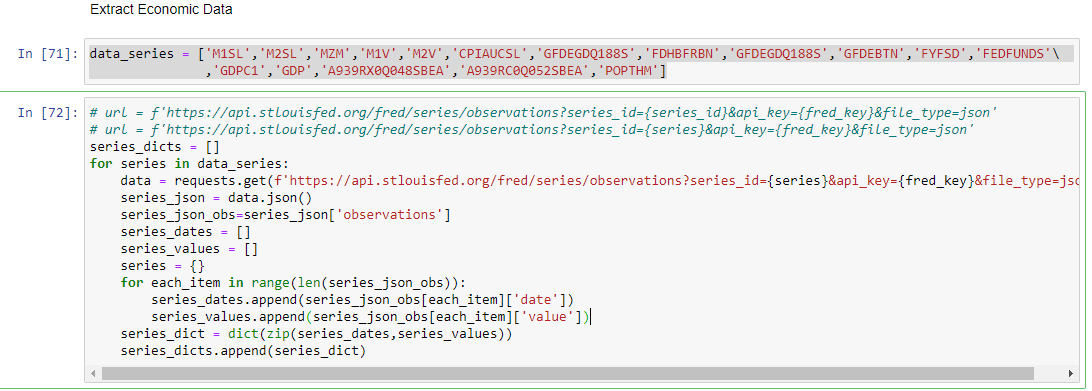
All of the data I extracted from FRED was through their API. I pulled 86 different data series on a wide variety of topics from different measures of GDP, CPI, PPI, M1, Mw, and many more.

*Justification for Database Structure*

All the data I extracted is in nicely formatted time-series tabular form. This lends itself well to a SQL-based database rather than an unstructured database like MongoDB. Despite the wide variety of data series I use, the database is actually not very complicated, since “Date” serves as the Primary Key for all tables, and all tables can relate to each other through their respective “Date” columns.

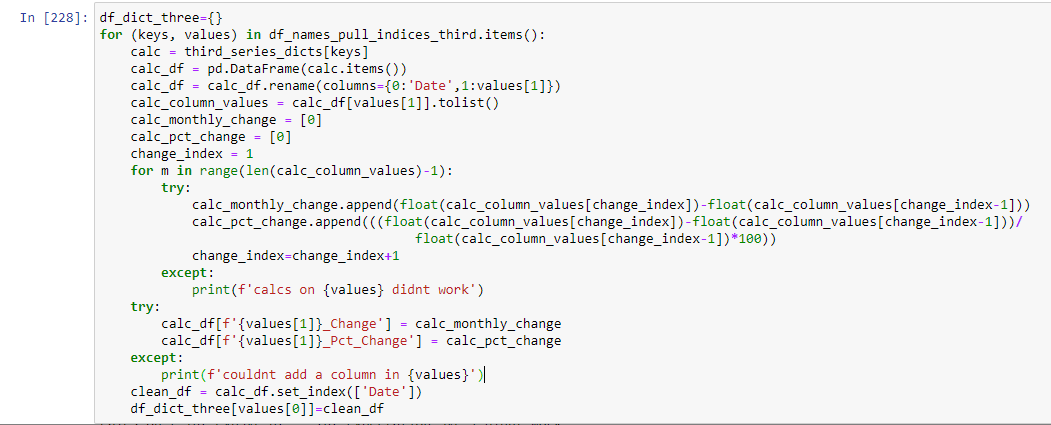
*Process*

To start, I read through the API documentation for FRED, and constructed the necessary query:



I extracted the data in several “chunks” so I wouldn’t ask too much of the FRED server at a time.

Once the data was extracted, I had to create pandas DataFrames and run some calculations. I was able to write loops to be able to process several DataFrames with a single block of code:



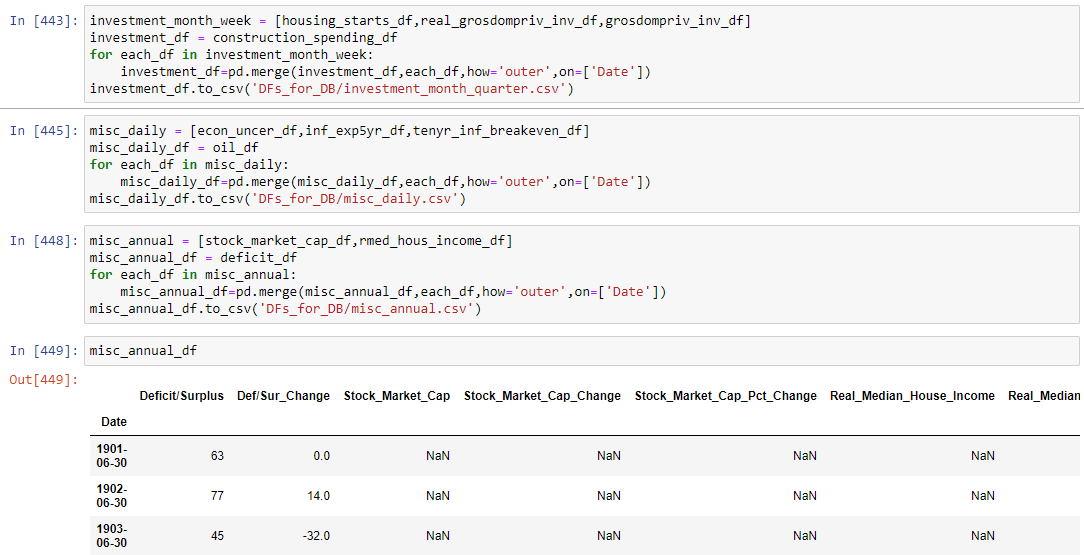
Once I had all of the DataFrames saved in dictionaries, I pulled them out individually to make merging easier later:



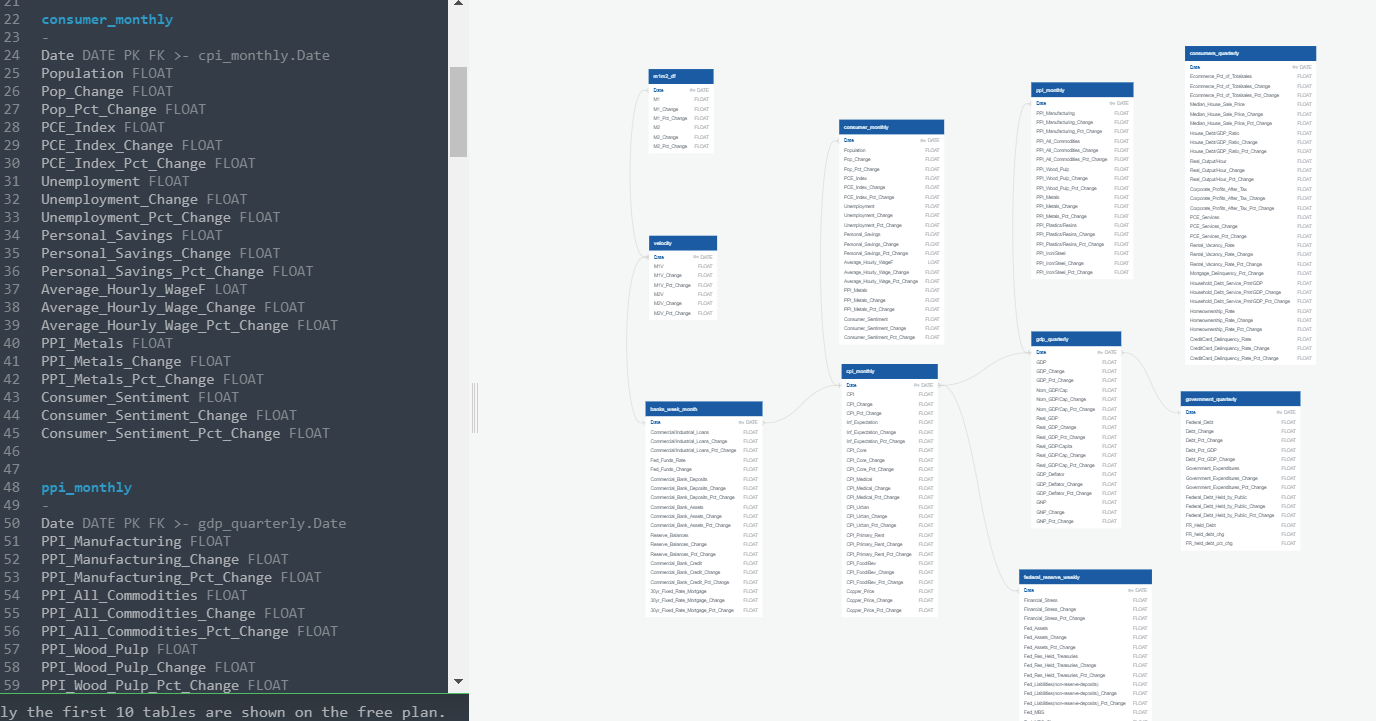
I ran into a few data sets that had null values or “.” as a placeholder, I had to do some further cleaning so I could then run similar calculations as above. It helped that I incorporated error handling in my code above to process the DataFrames:



I then merged DataFrames using more loops according to categories (Money, Banks, Government, CPI, etc.) as well as the time intervals of the data itself (Daily, Weekly, Monthly, Quarterly, Annual):



As I merged the DataFrames, I input the structure into a program that generates ERDs to later set up a SQL Database:



Finally, created the database in PostgreSQL, imported the CSVs that I exported from Jupyter Notebook, and ran some basic queries to ensure everything was working properly:

