NYU INFO 2335 “Freestyle” Project
Planning and Requirements Document
Vinicius Moreira (vgm236@stern.nyu.edu)

Downloading price data from the Bureau of Labor Statistics

Background

Financial market asset prices follow the news, being it a company specific fundamental, policy changes or economic data. To be ahead of the markets, economists like me track and analyze macroeconomic data with the intent of drawing a scenario for the future. That exercise allows us to try to predict where prices will be and profit from it.

One type of data that is particularly important to economists is the Consumer Prices Index (CPI henceforth) and its components. There are two main reasons why this data is important.

First, the Federal Reserve has a (informal) 2% target for the CPI. As such, higher (lower) CPI changes may lead the monetary authority to raise (reduce) rates. This has direct implication in all market prices, from the T-bills market to the S&P 500 and even into other countries asset prices.

Second, one type of asset available in the financial market are the Treasury Inflation-Protected Securities (TIPS). Those bonds return an interest rate plus the CPI. Therefore, all else constant, higher CPI means higher prices of TIPS.

However, at these days, speed in drawing a scenario with incoming data is a differential to beat the market in a world full of data watchers. Therefore, I’d like to create a tool to speed up the process of downloading and

Problem Statement

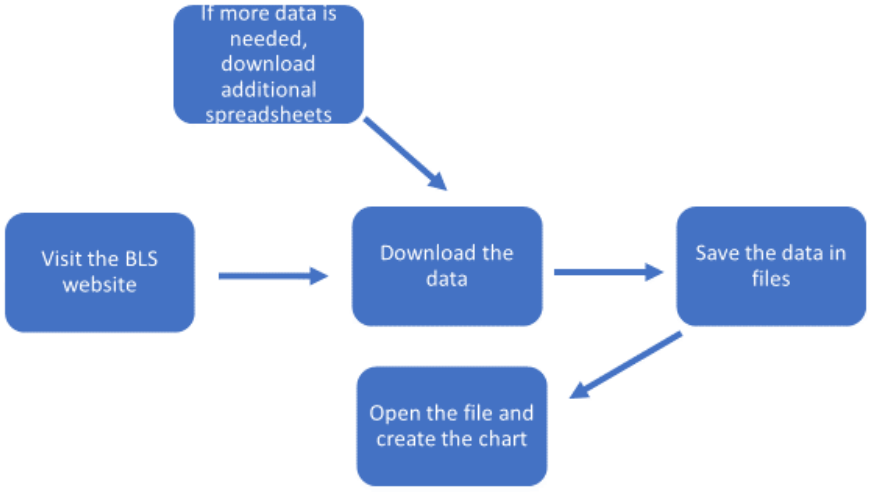
User Need

- As an economist, I want to download the CPI data quickly so I can be ahead of markets and lead my company and my clients to profit from my strategies

Current State of Processes

The data is available in the Bureau of Labor Statistics (BLS henceforth) in this link. The source contains a table in the excel format with the CPI and its breakdown only for the past year. Therefore, to download the full data (including the details of the report), one needs to download a ZIP file with the whole history of the CPI. The process, described below, can take hours depending on how much data you already have and how much you need.

1. Visit the BLS website, find the appropriate path to the CPI database
2. Download most recent data (example: May-2019)
   a) If more data is needed, one needs to download additional spreadsheets available there and link then together with the most recent data
3. Save the CPI information in your files
4. Finally, to visualize the information properly, a chart needs to be created and formatted.



Proposed solution

System Objectives

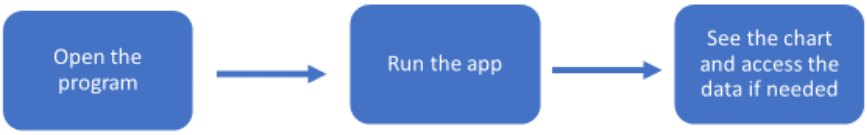
My proposed solution is called “Downloading price data from the Bureau of Labor Statistics with Python”. Its job will be to access the Bureau of Labor Statistics data, download the information in a user-friendly way (.xlsx file), and generate and quickly display the data downloaded in different formats (level, monthly changes and yearly changes).

Future State of Processes

The system should streamline the current state process by automating the process of downloading and visualizing the data.

After the system is developed and deployed, my future process will involve the following steps:

1. Open and run the program. Done! I have the data chart I need



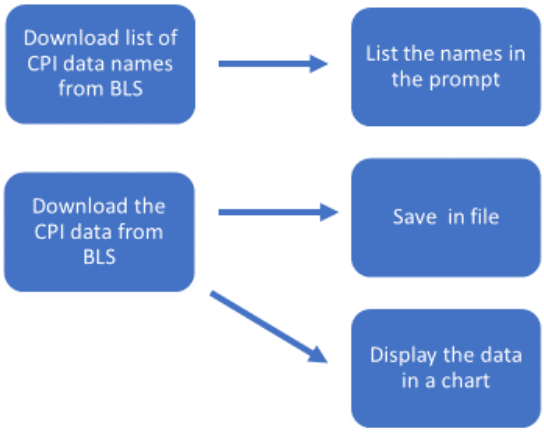
Information Requirements

Information Inputs

- The CPI data you want (Headline CPI, Core CPI, apparel prices, etc), which will be download and listed whenever you run the app
- The date you want your data to begin and end
- The transformation you want to see in the chart – original series, month-on-month changes or year-over-year changes

Information Outputs

- Excel file with the data you selected
- Chart with the data (transformed or not) you selected



Funcionality Requirements

Reading Contact Info

REQUIRED: The system requires an API from the BLS website, which can be requested in this link and will be emailed to you. The API Key should read this information from an

environment variable, to keep it out of the source code, and to prevent me from having to type it every time I use the program.

Listing data

Before anything, the system should list all types of Consumer Price Index data available in BLS.

Understanding which series of data the user wants

The system should understand the data you need with the code provided in the list above (if not, an error message should appear). It should also understand the beginning and ending dates for the downloaded time series (if not, an error message should appear).

Saving the data downloaded in excel format

The system should save an excel file (in .xlsx format) in a general data folder wherever the program is running from.

Displaying a Chart with the data

The system should return a Chart with the data (transformed or not), formatted appropriately.

Technology Requirements

The system will implement its logic using the Python programming language. It will use at least the following Python modules and packages:

- The “datetime” module for determining dates in the series of data
- The “requests” package to get data from BLS site
- The “json” package to read the data coming from BLS
- The “csv” package to create a file in excel format
- The “dotenv” package to securely read the API Key of the user
- The “os” package to generally save the excel file in a folder (named “data) wherever you are using the app
- The “pandas” package to easily work with the data series

- The “matplotlib” package to draw the chart

Development Plan

Feature-Specific Development Plan

Collecting Contact Info

First, I will use the general API provided by BLS (which does not need an API Key). If can request data and the list from this general API, I will then create and store my own specific API Key (that gives me access to more data more times).

Reading Schedules

I will store the list of CPI items in a general file in the same folder. If I can do that, I will use the “os.path” code to save in a specific data file. I will do the same for the data access.

I will read the data and transform it in two lists, one displaying the name of the CPI items and the other displaying the codes.

Determining Data To Be Downloaded And Displayed

I will create an input code requesting the user to define which specific data he wants from the list. If I am successful at that, I will create an option in which, if the code is incorrect, the program will display an error message. I will do the same for the initial and end date of the data.

I will not use the inputs the user set at first. I will initially use a general code to see if I can access the data. I will see if it is downloading the data properly this way, saving in the same folder the program is. Again, if this works, I will start using the “os.path” to save the file in .xlsx format.

Presenting The Chart

With the data saved (derived from all the inputs of the user), I will then try to build a chart. If working properly, I will then make sure I edit the chart with general

configurations (Ex: fixed title as “Inflation” and y-label = “label”). Then, I will define that the title of the chart will be the name of the CPI item I chose.

Alternative Transformations Of The Data In The Chart

But I want also to display the data in month-on-month and year-on-year formats. As such, I will take a step back and define two different series using the “df.pct_change” function. Those will be the month-on-month and year-on-year series.

I will then write an if condition for the chart creation. Depending on the transformation of the data selected, the y-label will have that name (ex: “MoM”) and the chart will be displayed with that transformation.