Creation of a Fixed Income Dashboard using Python Dash Library

Environment: JupyterLab in Anaconda

# Desired Output

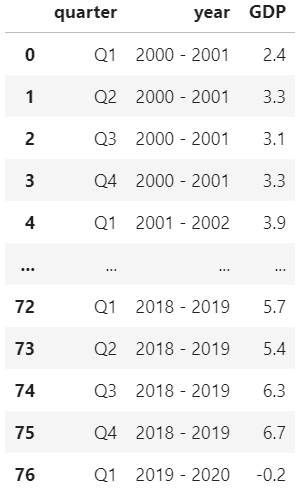
The objective is to have a guide on creating a fixed income market view (go long or short duration, which area of the curve is most attractive) without the need for a Bloomberg Terminal.

In particular, the output will include the following: (1) local GDP growth and inflation, (2) chart of historical yield curve per maturity, (3) Term Spread Z-scores, and (4) Real Yield Z-scores.

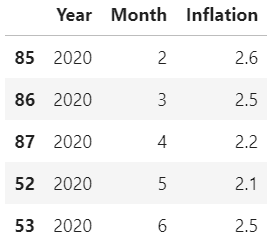
# Data

Local GDP Growth and Inflation, ***One-Time Extract***

For GDP growth, since I could not find a way to show last column (even with pd.read\_excel() and requests), I had to manually download the file and just use Pandas \_\_.read\_csv(). The cleaned dataframe output is saved as a csv as well using \_\_.to\_csv().



For inflation data, I used BeautifulSoup, requests, and Regex to extract the information from html table.



We perform the same saving of the dataframe stored using \_\_.to\_csv().

Unfortunately, as I was about to update for the following month, I noticed the website changed the “td class” associated with the inflation data which changes the pattern I set to search. As such, I put in place another procedure for singular updates.

Local GDP Growth and Inflation, ***Moving Forward***

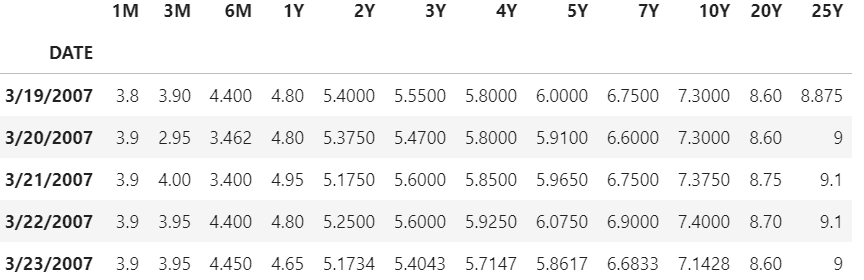
Since data points are added only quarterly for GDP and monthly for inflation, instead of repeating the entire process again which unfortunately included some manual parts due to problems encountered, we just type down the updated information in a dataframe format then append this to the historical csv created. After which, we just use pd.to\_csv() to update the historical dataframe stored.

Philippine Yields, ***One-Time Extract***

Since PH yields (BVAL) are saved as pdf files in the website, I first downloaded these in a local folder using requests and BeautifulSoup.

Before October 2018, PH local yields were PDST-R2 yields and these historical rates are also saved as a pdf which require a one-time download for the creation of our historical PH yield database. For this, I used tabula library since it is a clear table stored in a pdf file. I then saved it as a csv.

I then perform some manual arrangements of both datasets.



Philippine Yields, ***Moving Forward***

We only get the dates we need. The code creates the links to the pdf files and if it doesn’t exist then it skips for that date, but if it does, then it downloads the pdf and extracts the data into a dataframe. This dataframe containing updated yields then gets appended to the historical dataframe named PHreferencerates.csv.

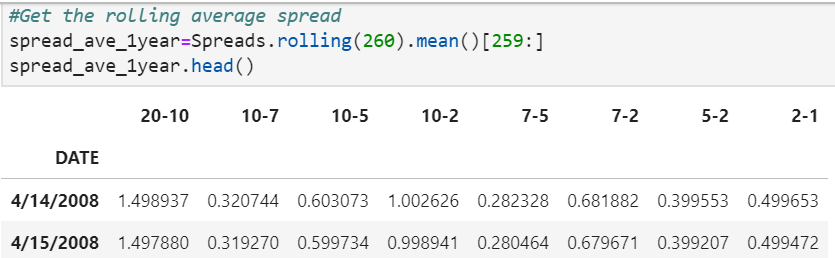
Term Spreads Z-score Computation

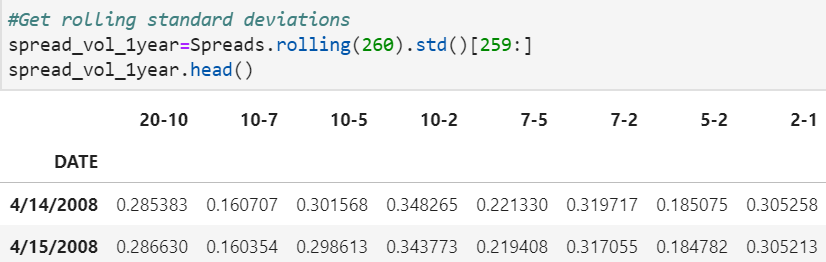
I load the PHreferencerates file and create Spreads columns from there.



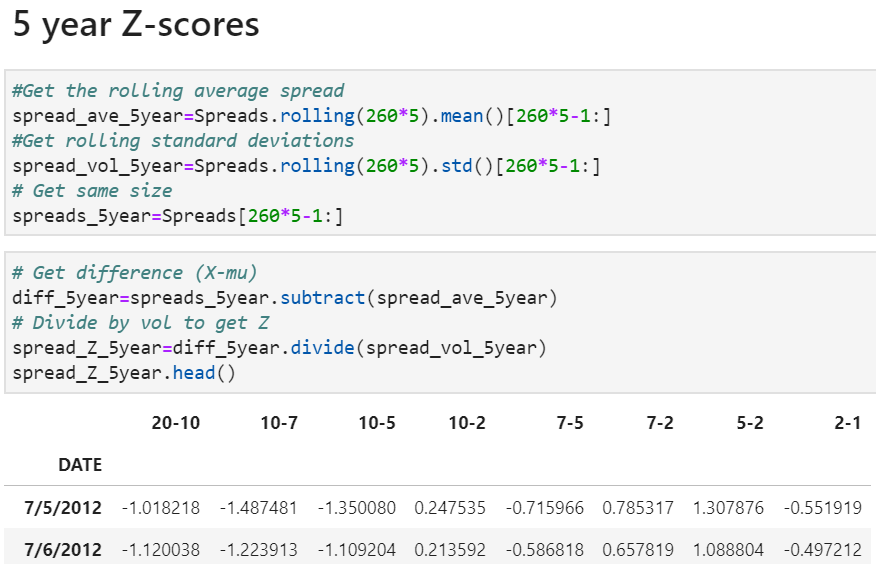
From each spread, I compute a rolling 1-year and 5-year mean and standard deviation to eventually compute the latest Z-score using (X-mu)/vol.

Computing rolling mean and standard deviation:

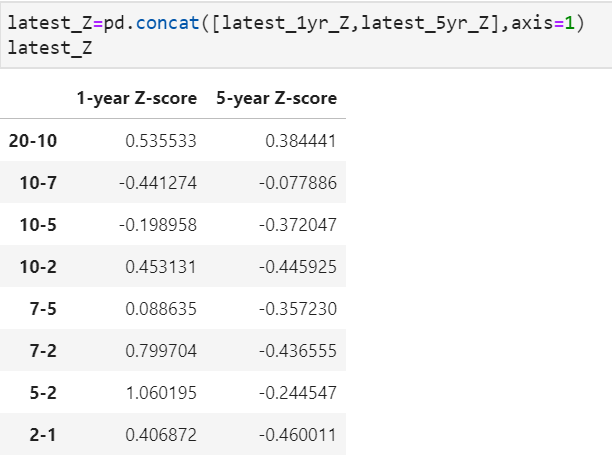




The resulting start date for the 1-year vs. 5-year will be different. To get the difference(X-mu), I cut the initial “X” dataframe, i.e. spreads, such that it has the same starting date as the rolling average.



Just getting the latest Z-scores to check attractiveness:



Sample Interpretation:

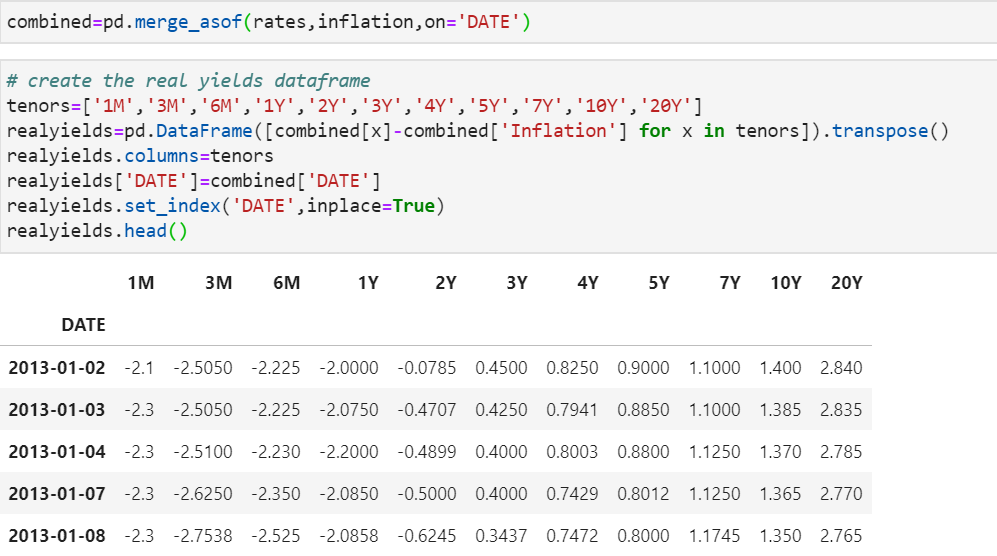
Only the 5-year relative to the 2-year appears attractive with term spread Z-score > 1 for the 1-year horizon.

Real Yields Z-score Computation

After loading inflation and PH yields data, I add a DATE column to inflation data.



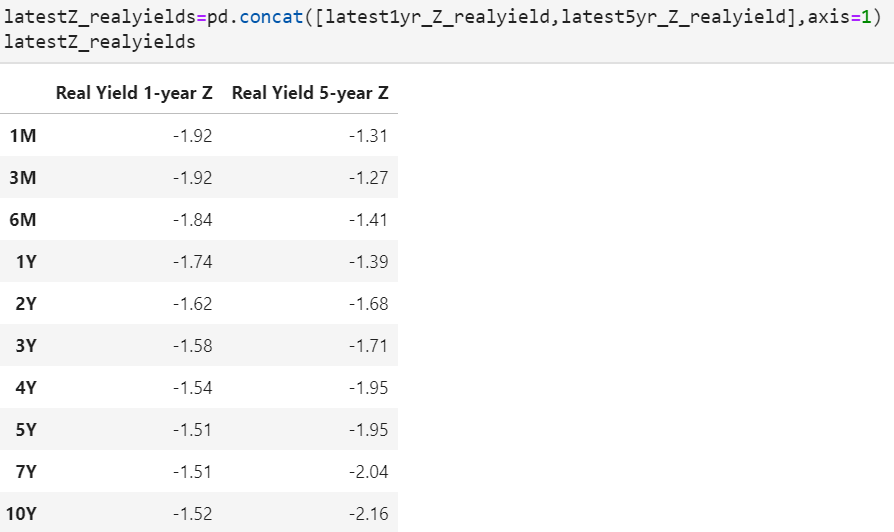
I then merge this with the yields data on the “DATE” column and from there create “real yield” columns as yield minus applicable inflation.



I then save this as PHrealyields.csv.



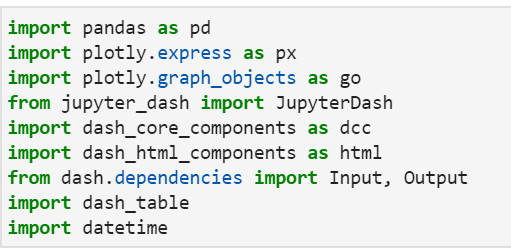
I then follow the same process as Term Spreads in computing the Z-scores of real yields.





Dashboard using Dash Library

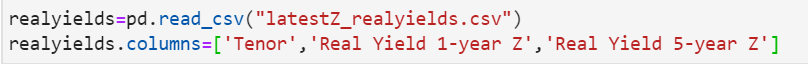
I first import all the necessary libraries:

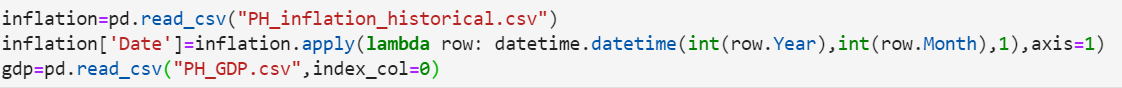


Then I load all the necessary data:







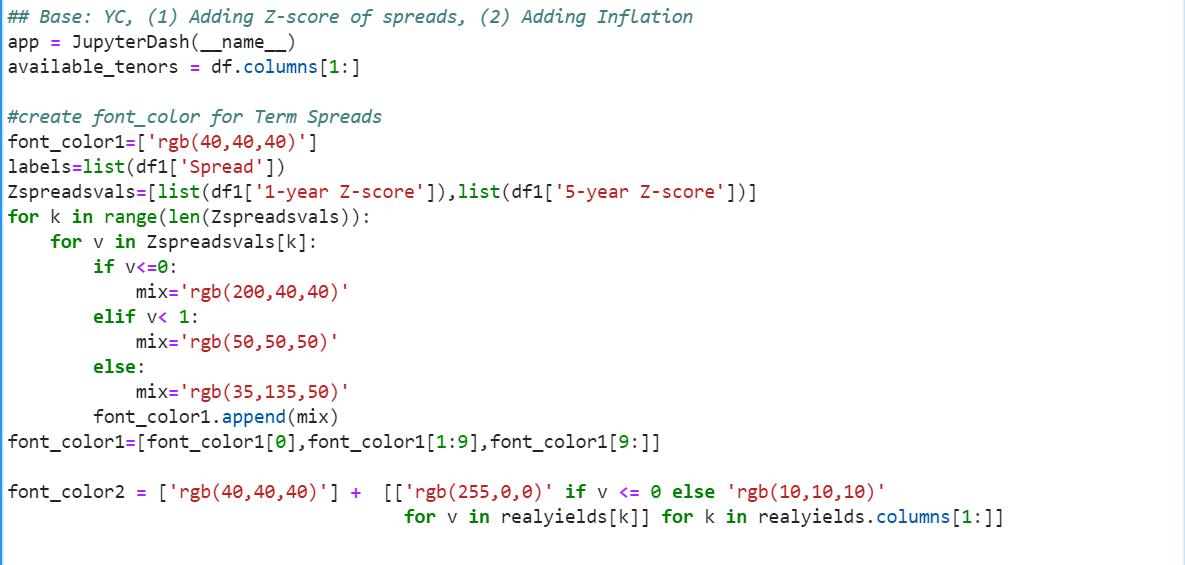


To chart GDP properly, I add a date column:

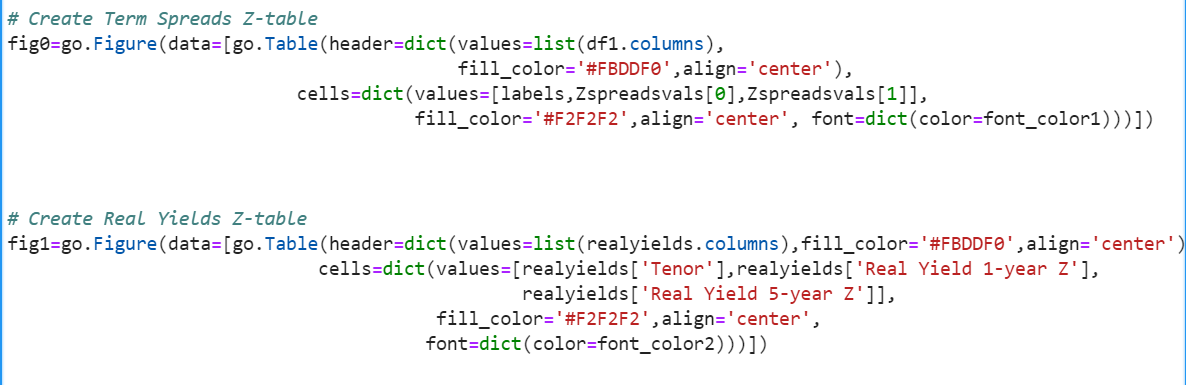


Finally, to output all of these in dash:

* Available tenors is stored as a list for our dcc.dropdown later on
* Create base font color for the term spreads with changes depending on the values
* The same is done for real yields but with only 2 conditions.



Additional formatting on the Z-score tables along with inputting of values under “cells” portion:



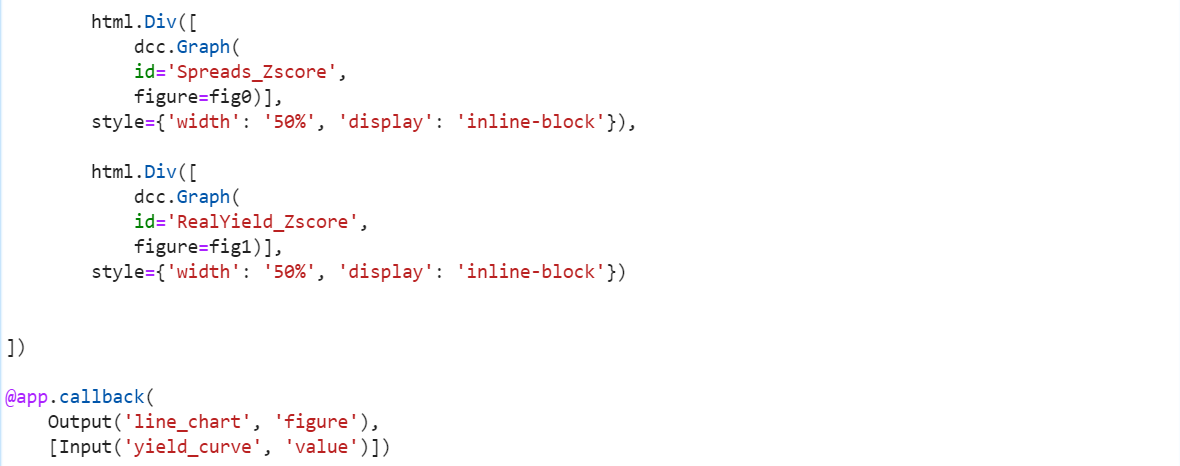
The layout for the graphs where inflation and GDP split the first area in-line each with 50% of the width.



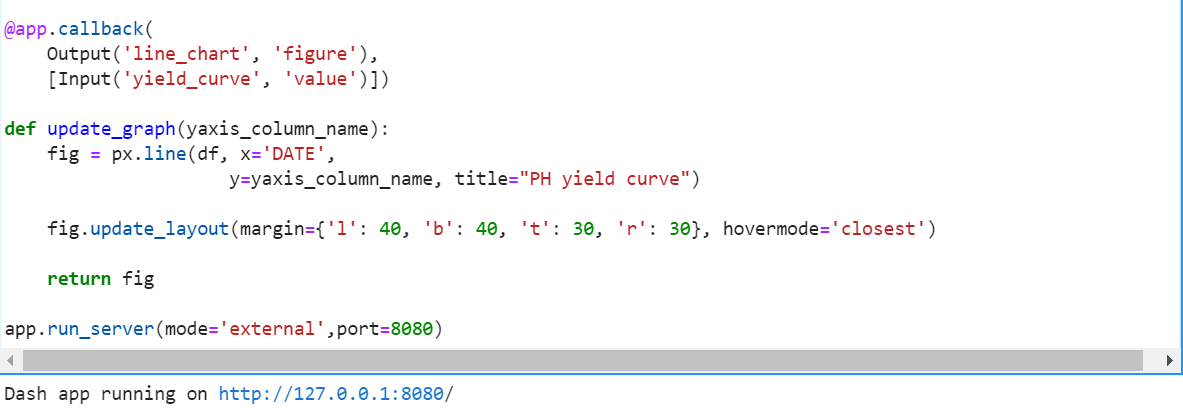
This is followed by the yield curve chart with special “dropdown” option allowing for the viewing of multiple maturities at a time:



Finally, the two Z-score tables each splitting in-line 50% width:



This portion allows the updating of the **line\_chart** as identified above depending on input **value**. Finally, some margins are added (left, bottom, top, right):



The final dash app appears as below:

