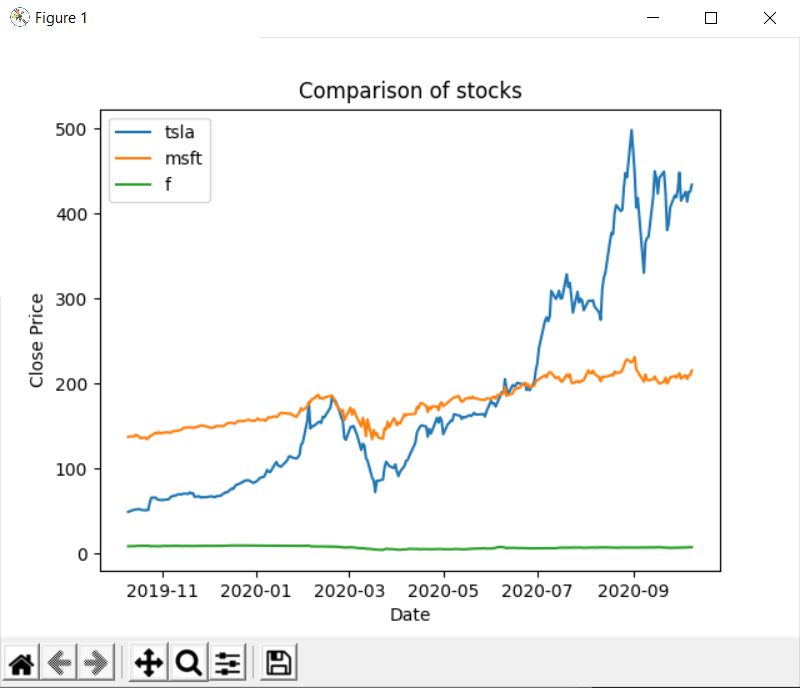
Nicholas DiGiacomo

CS 110 Project Report

12/2/20

**Section One: “Overview and Summary of Project”**

For this project, I decided to create a code that can determine, out of a number of stocks, which one is the best based on percent change. It first asks for the number of stocks you like to enter. After that, it asks for the ticker of the first stock and repeats to ask until the last stock. This follows by asking for a specific time window to the first date and then the last date that someone would like to compare the stocks. After all those inputs, the code calculates the percent change of each stock and prints all of them out. It then determines which is the best one out of the group. Lastly, it plots and shows a graph of all the stocks closing prices over the time window that was given.

SAMPLE OUTPUT:

How many stocks are in your portfolio: 3

Enter ticker name: tsla

Enter ticker name: msft

Enter ticker name: f

Enter start date in form YEAR-MONTH-DAY: 2019-10-10

Enter end date in form YEAR-MONTH-DAY: 2020-10-10

Date is valid

Here's the first close for the start date: 48.95

Here's the last close for the end date: 434.0

This is the percentage change for TSLA in the period you listed: 786.66 %

Here's the first close for the start date: 137.18

Here's the last close for the end date: 215.25

This is the percentage change for MSFT in the period you listed: 56.91 %

Here's the first close for the start date: 8.34

Here's the last close for the end date: 7.25

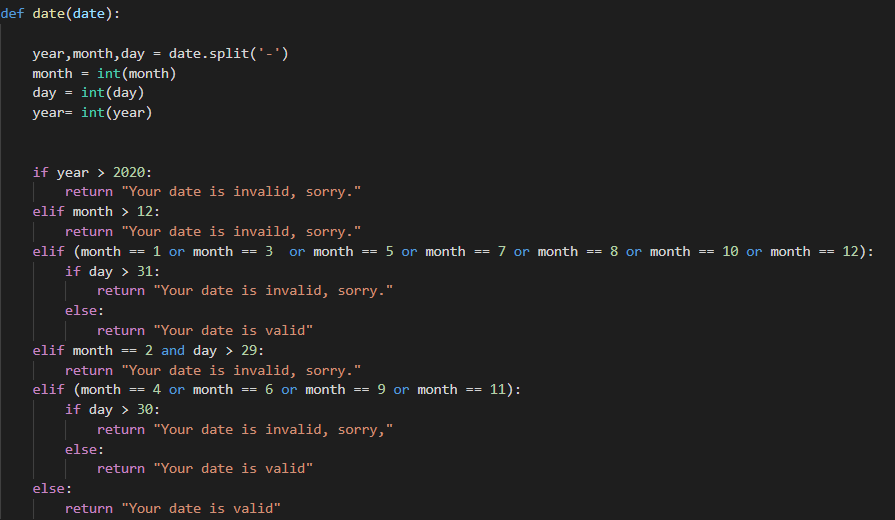
This is the percentage change for F in the period you listed: -13.06 %

The best performer of the stocks you selected was tsla with a 786.66 %

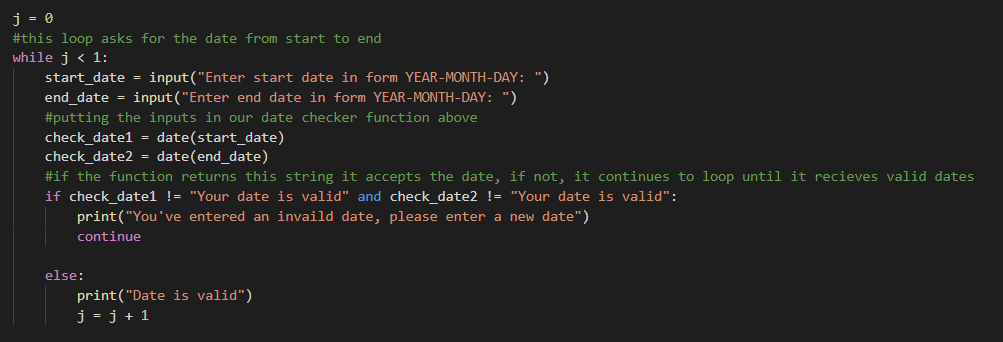
**Section Two: “Target Audience**

The specific audience for this type of program will probably be investors and portfolio managers in order to look at overall how their investments are doing as a whole and individually. Furthermore, it could help determine which stocks to invest in next. If someone had Tesla as a stock, they’ll be able to see by using this code that the stock is doing extremely well compared to others visually and numerically. It also could be used to look at entire indexes or sectors and see what are the best companies to invest in those massive groups.

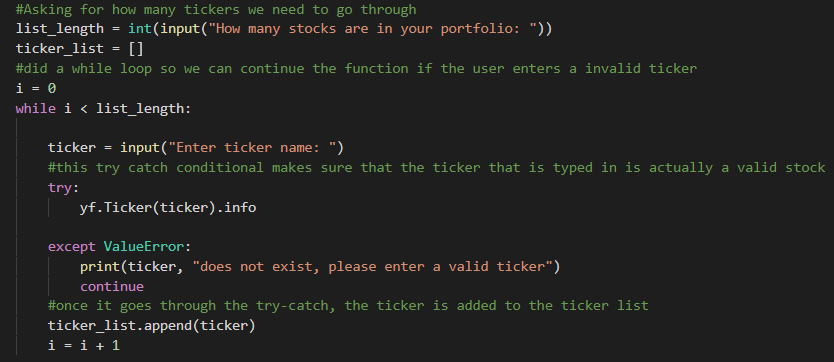
**Section Three: “Specific Programming techniques used”**

Multiple techniques were used in order to make this program. I made a main function, a date function, and a graph function. The date function helps determine if the date that’s entered is a valid date and is not out of the time zone, avoiding a possible error message at the end of the code:

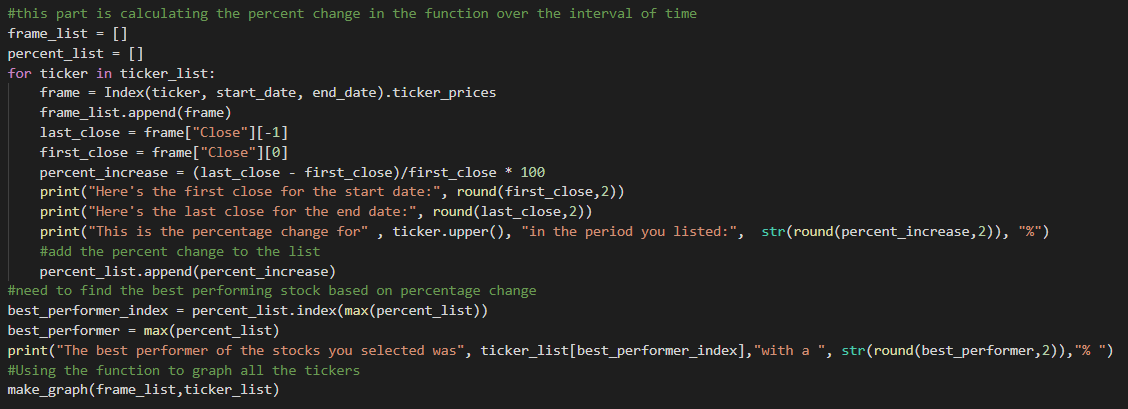
The function gives back a string value which is used in the main function of the code. In the main function I had to use multiple new methods in order to make sure the code doesn’t end if a invalid date is entered including a while loop and a continue expression. So, now when you run the code and enter a date that either hasn’t happened yet or is not an actual date, it will re-run the input and ask you again:



I did this for multiple inputs throughout the code and also used the try catch process for possible invaild ticker entries. I had to again use the while loop so that the code will restart automatically if the input is invaild :



I got my data from a library called yfinance, which when called can search up the ticker and give the data in the inputted time frame needed for the code to run. This helped me immensely in figuring out how the code can work and made the process easier. Once that was done, each dataset was analyzed by picking out the last closing price and the first closing price, then putting those values into an equation which calculates the percent change. Not only that, but the datasets are also graphed by date and closing prices afterwards using the graph function:



**Section Four: “Challenge(s)”**

There were many challenges that were encountered during this project. One was figuring out how I could make sure the code wouldn’t end if an input was invalid. I realized that while loops with a continue expression were a way to do this, yet we never did this in class, so it was challenging learning how they work and how a while loop is different from a for loop. I eventually figured it out. Another one was figuring out how to get the data that was needed for the project. I realized that there are many libraries in python that aren’t just math and matplotlib and assumed there could be one that could help me analyze stock data. I found yfinance which analyzes data from yahoo finance and imports that data to the code. This made the data finding process much easier and helped me focus on other parts of the project.

**Section Five: “Future Extensions”**

If I were to add new features, I would definitely add more metrics that the stocks could be looked at aside from just the percent change. I would maybe add the multiple p/e, p/b, and other market metrics. Then, I can create multiple graphs for users to look at based on these new numbers. Furthermore, I would also identify which sector a certain stock is in. For example, if Tesla was entered, I can label it as being in the auto sector. Lastly, I would try to see if the code can put the numbers such as p/e, percent change into a spreadsheet automatically so all the data is saved and in a file organized. There is a lot to build on within this code.