Ryan Jonnada

Professor Kevin Ryan

CS 110

December 13th, 2020

I pledge my honor that I have abided by the Steven's Honor System

Overview and Summary of Project

This program presents five stocks (Apple, Microsoft, Nvidia, Adobe, and Cloudflare) that performed well during the pandemic. First the program informs the reader that it will present those five stocks, and asks the reader if they are ready to continue. The reader can input any answer that they want but the code will only continue when "yes" is inputted, and prompts the reader to do so. After this, a plot is defined to graph the five stocks mentioned above along with the S&P which is used as a benchmark to be compared to. Seven graphs are imported, five of them are the individual stocks, one is the S&P 500, and then the last one is one with all five stocks compared to the S&P 500. Each graph displays the respective stock prices from January 1st, 2020 to December 1st, 2020. The code then gives details about the graphs that will be shown and how to understand that. Along with the graphs, the code will provide reasoning in the console for why these stocks persevered during the pandemic and did so well despite the conditions. After this, the program will thank the reader for viewing the code.

Target Audience

This program would be targeted to those who are new to learning about the stock market and would like to understand basic information about how certain events can affect the price of a stock.

Specific Programming Techniques Used

First, I imported yfinance, matplotlib.pyplot, and time. Yfinance downloads the market data from Yahoo Finance which is used to generate the graphs in this program. Matplotlib.pyplot analyzes the data from yfinance and uses that data to plot the graphs of the stocks. Time is imported to give breaks and allow the program user to read all the instructions before continuing. After importing these packages, I am defining a class to process all the variables used to graph the stock data. A while loop is then implemented to ensure that the user inputs the correct answer to continue the program. I then define the function "plot_stock" with the parameters that will be used to facilitate the creation of the graphs. Within this function I am downloading yfinance data between the variables of start to end (January 1st, 2020 to December 1st, 2020). Then the function plots the graph, after this the x axis is set to date, and the y axis is set to price. Furthermore, I am using a list variable for the tickers that will be used by the function "plot_stock", as well as the start and end date. I am then entering print information that is followed by a sleep function that uses the imported time package to ensure that the reader has enough time to read all of the given info. Then I created an object that allows me to call the "plot_stock" function which is where the graphs will be plotted. I do this for the S&P 500, Apple, Microsoft, Nvidia, Adobe, Cloudflare, and one with all stocks included. After following the directions to call the graph, you must close that graph to view the information on that stock

as well accessing the next graph. Once all graphs have been called the program then closes with a closing print statement.

Challenges

I was not able to open all graphs within the same window, therefore I graphed them all in one function call which created 7 graphs on separate windows. Because of this I implemented a block of print statements with instructions on how to access these graphs. However, this brought up another issue. When these instructions were printed, the graphs would be imported immediately leaving no time for the user to read the instructions. I overcame this by importing the time package which gave the reader time to understand how to access the graphs.

Future Extensions

I would like to program a website to add this information to so viewers could see the graphs along with the data at the same time. I would also implement a more in depth explanation about why these stocks did so well during the pandemic. Along with this, I would like to include start and end dates as input for the program to allow for more in depth analysis which will need extensive input validation. I would also like to include an option where the user can input any two stock prices which will both be graphed on the same graph which can be used for comparison.