Jacob Buurman

Professor Ryan

CS110

13 December 2020

Final Project Report

For my final project in this course, I decided to make an extension of previous research done for my CAL103 research paper. In a group with other QF/Finance majors, we tackled the following question: how has COVID-19 impacted certain sectors in the stock market? Albeit a vague question, we narrowed down each of our sections to a specific area of focus, mine being stocks that actually profited because of the pandemic. These profits occurred for various reasons, such as an increased reliance on tech in daily life, the race to develop a vaccine, or simply due to an increase in demand for certain items a company already produced. Throughout my portion of the essay I mentioned eighteen different stocks, so I thought making a program comparing these stock charts to the index(es) they belong to over the course of 2020 would compliment my paper with visual data nicely.

To begin my program, I commented out a section describing the premise behind the program, explaining its connection to my CAL103 group research paper. This could be interesting to people who may have read the essay or are simply interested in stocks that saw significant gains during the pandemic. I then imported *datetime*, *matplotlib.pyplot*, and *pandas\_datareader* modules to be utilized later in the program. I also imported *style* from *matplotlib* in order to construct the stock charts, change the colors, etc. In the first visible part of the program, I once again mention that this program will be taking a look at some stocks that profited because of COVID-19 and prompt the user to notice the general positive trends and/or speedy recoveries of these stocks during the early stages of the pandemic (March, April), while the S&P500 and NASDAQ had large losses. I then provided the alphabetical list of stocks mentioned in the essay, assigning each a number from 1-18.

I assigned the *num* variable to an input asking the user for one of these numbers. Before this, I assigned the *start* variable to *dt.datetime*(2020, 1, 1) and the *end* variable to *dt.datetime*(2020, 12, 31) to show the entire year’s worth of data in each chart (done using the *datetime* module). I then performed a long if-elif-else statement with the same general format, the only thing changing being the *num* variable and it’s associated stock ticker. Using the imported *pandas\_datareader* and the *web.get\_data\_yahoo* function, the program will get the Yahoo Finance data for whatever ticker is entered, with the second and third arguments being the start and end variables (i.e. *web.get\_data\_yahoo*("AMZN", *start*, *end*). I assigned this function to the variable *df*. In the line below that I entered *df*[“Adj Close”]*.plot*(color = “red”) in order to tell the program to graph the adjusted close prices and plot it on a line graph in red (this uses *matplotlib.pyplot* and *style*). I repeated this with all eighteen stocks, ending with an else statement that prints an error message prompting the user to enter a number from 1-18.

I then constructed another if-else statement, which prints out a graph of the NASDAQ Composite index over the year 2020 if the *num* variable is equal to that of a stock listed on the index. I used the same format for getting the charts of the tickers, except I used the ticker ^IXIC and changed the color to blue in the *web.get\_data\_yahoo* function. I also entered messages under this if statement telling the user that the entered *num* variable is red, the NASDAQ is blue, and that the stock they entered is listed on the NASDAQ. Under the else statement, I entered a message telling the user that the stock they entered is not listed on the NASDAQ. I then did the same with the S&P500 in another if-else statement, using the ticker ^GSPC and changing the color to green. To conclude the program, I had another if statement with the numbers associated to stocks that are listed on both indexes, printing a message telling the user so. I did the same with another if statement and the one stock in my program (17, Weis Markets, WMK) that was listed on neither index, also printing a message telling the user so.

While the program runs smoothly, I did run into a few problems and there are definitely some improvements I could make if given more time. When the program runs, it only shows one stock chart at a time and you have to X out of the first chart to see the next one. This isn’t a huge problem, but I would like it if it showed the two/three charts at the same time for easier access to make comparisons. I originally coded the program so that the stock and index were on the same chart, but I quickly realized that this wouldn’t work because the large price difference between some of these stocks and the indexes makes the stock graph look like a flat line. Also, since the code for making the graphs of each of the eighteen stocks is very similar, I would imagine I could make a *chart* class to make the program cleaner and simpler.

