

Rachel Potter

Dr. Kevin Ryan

CS 110A: Creative Problem Solving in Computing

13 December 2020

Stocks of South Korean Entertainment Companies

Section One: Overview and Summary

My program aims to take the past stock prices of five popular South Korean entertainment companies (specifically, Cube, JYP, YG, FNC, and SM Entertainment), and see which one is doing better in the stock market. The information is given as a csv file, and my program takes the price of stocks and the date from each company, and graphs the information side-by-side in the same graph to be easily compared. The y-axis shows the price (in USD), the x-axis shows the date, and the line showing the stock prices for each company is color coded, explained by a key. A sample of the user interface can be found below in section 6. The user is given the option to graph the stocks since 2016 (by month), for 2020 (by week), or for up to a month back (by day) by inputting 1, 2, or 3 accordingly. The maximum date for now is December 11, 2020. If the user closes the graph, they are allowed a new input.

Section 2: Target Audience

My program could be useful to anyone who is considering investing in any of these companies, to see which one might be the better investment. It is particularly helpful for people living in the US, since most other data tables for these companies show the prices in KRW. My program automatically converts this information to USD. I also think that fans of the groups these companies produce would be interested in the output as well. Each company takes a different approach to their artists (for example, SM artists are always highly polished while JYP

artists tend to go for a more “self-made” image), so fans may be interested in how that translates to success in the stock market.

Section 3: Specific Programming Techniques Used

Functions were used for each possible user input, as well as one for the graph. For the graphing function, arguments were also passed through, depending on the input. *If-elif-else* statements allowed different functions to be called based on user input. Since this is a graphing program that intakes csv files, both matplotlib and csv were imported into the program. The files first had to be opened. From there, lists were used several times: to store the files that were to be read, to store the labels for the lines on the graph, as well as the lines in each csv file also being stored as a list. Utilizing nested for loops, each line of the file could be read individually. In a second loop, new lists were also created, with elements in the date and price column in the csv lists being appended to their own date list and price lists respectively. Some string manipulation (`string.replace()`) was used to get rid of unnecessary characters like commas in the numbers in the price list. Converting some elements in a list into a floating-point number was also necessary, in order to use the conversion factor of 0.00092 KRW to USD. From there, the information was ready to be graphed using the matplotlib library. Cases where the user did not input a number 1-3 were also accounted for, with an *else* statement.

Section 4: Challenges

The first challenge I faced was that the dates in my datasets were originally in the wrong order. This was fixed through some reformatting in Excel. Then, I realized that my numeric values in the csv were stored as strings, and contained commas, so could not be directly converted to floating point numbers. A for loop had to be created for each element in the price list, which used `string.replace()` to replace the comma with no space. The number could then be

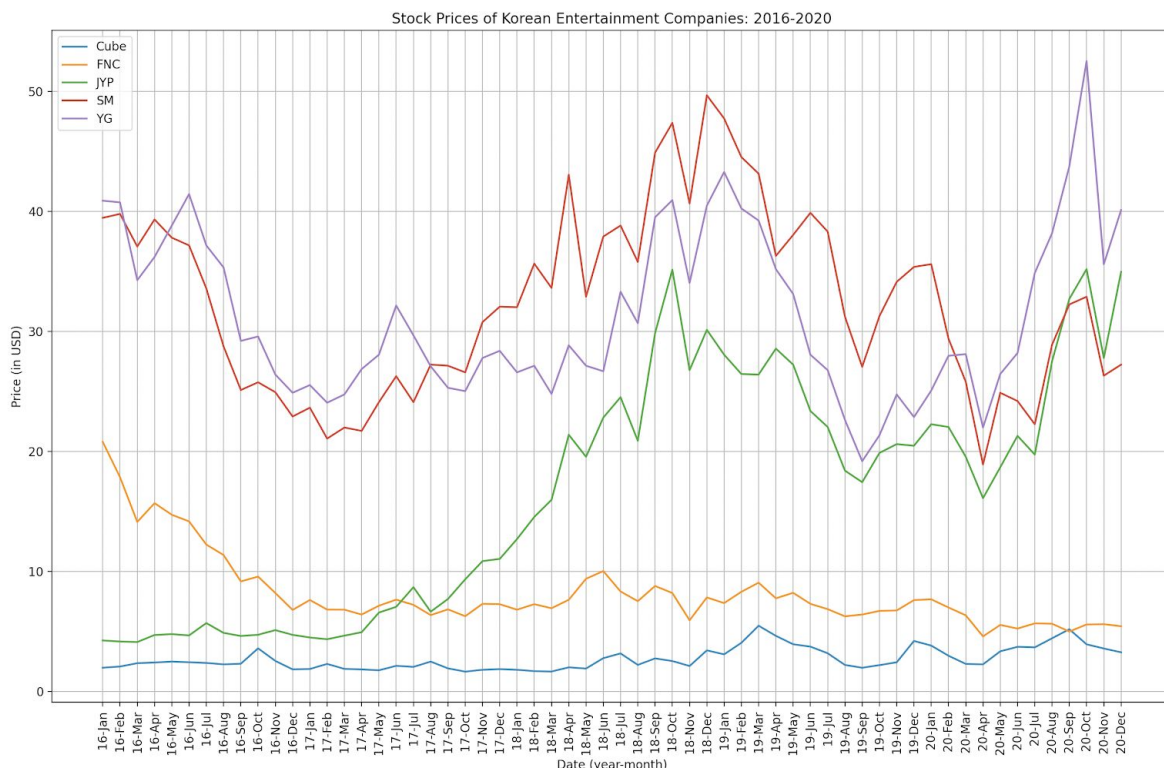
made into a floating point number. However, since my dataset also had the prices in Korean currency, each price then had to be multiplied by the conversion factor (one South Korean won being equal to 0.00092 USD) before the price list could be graphed. Finally, decision structures had to be implemented a certain way so that the program would be re-run when the user was ready for a new input. A *while* loop was used for this purpose, with more input being asked for after the *else-elif-else* statements for the menu.

Section 5: Future Extensions

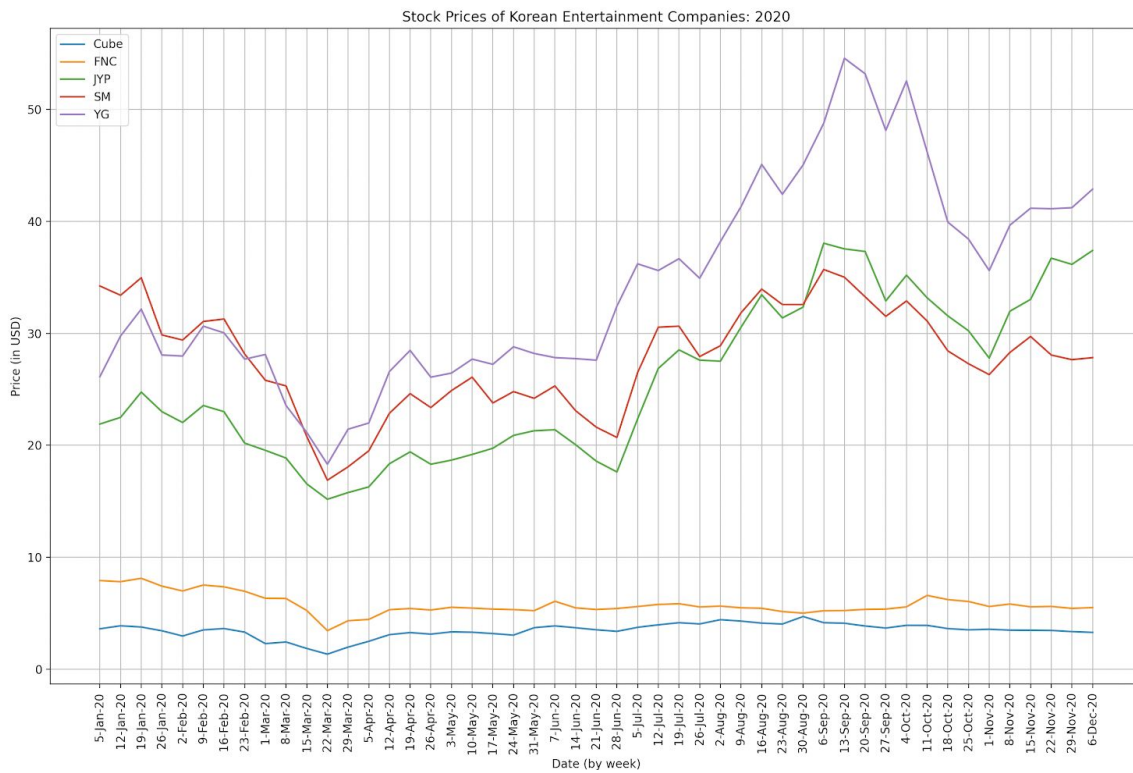
Currently, the maximum present time that my csv files hold is limited to a set date, December 11, 2020. A possible extension could be to grab updated data directly from another website, so that it can expand to show future dates, or even other companies. I could also add more options to the user menu, to get different combinations of companies to compare. Finally, an option could also be created to convert everything back to the original currency.

Section 6: Sample Inputs and Outputs

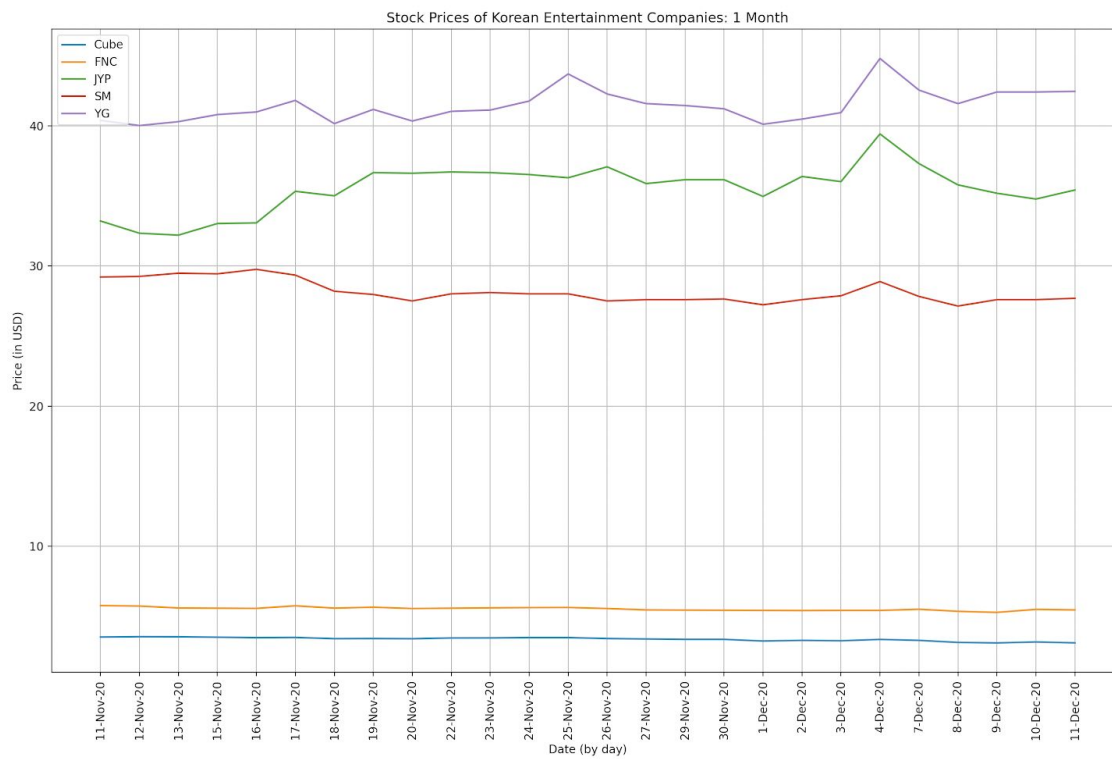
Graph with user input 1:



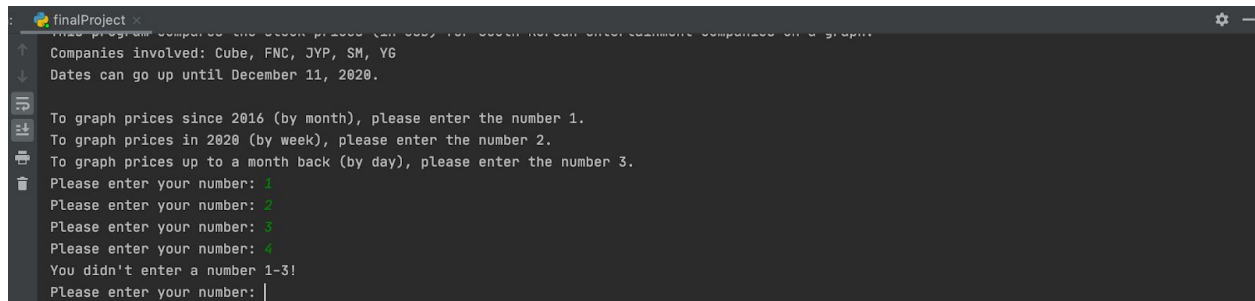
Graph with user input 2:



Graph with user input 3:



User menu:



```
finalProject
This program computes the stock prices (in $$$) for each stock entered, and displays them on a graph.
Companies involved: Cube, FNC, JYP, SM, YG
Dates can go up until December 11, 2020.

To graph prices since 2016 (by month), please enter the number 1.
To graph prices in 2020 (by week), please enter the number 2.
To graph prices up to a month back (by day), please enter the number 3.
Please enter your number: 1
Please enter your number: 2
Please enter your number: 3
Please enter your number: 4
You didn't enter a number 1-3!
Please enter your number: |
```

I pledge my honor that I have abided by the Stevens Honor System.

Rachel Potter