Jill McDonald

Professor Ryan

CS110A

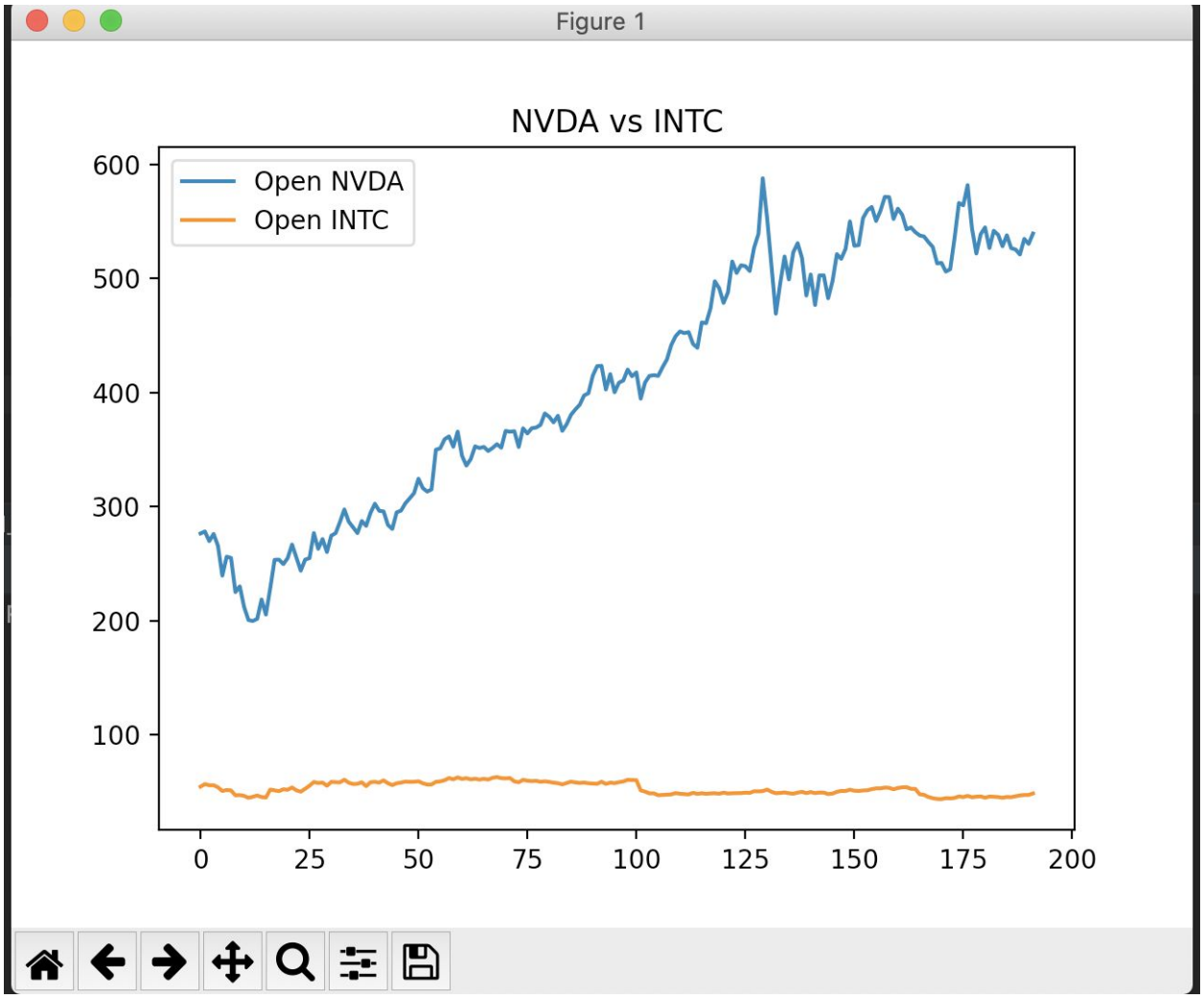
13 December 2020

Section 1:

My program compares stocks in the semiconductor industry to show how they have performed throughout the pandemic. The program takes the stocks of Nvidia, Intel, and Advanced Micro Devices and shows how these key companies have performed over the time period of March 1st to December 2nd, so the user can see the implications of the pandemic on the top semiconductor companies in the market. I created a function called “load_ticker” that created an instance of the finance ticker object, and it then takes ticker symbols as parameters. The program uses the ticker object to give us the stock history of prices, and returns the stock history for the specific range of dates. The infinite loop “while True:” runs until the user chooses option 4, which is exit, and this will break out of the infinite loop. This code will present the user with a menu to display a graph of two stock prices on the same plot. It reads the user's input, and then the program executes the action based on the user's input, and will display the graph of the number the user picked. It will load the history for two ticker symbols, and will plot the open prices for the symbols on the same axis. If the user chooses 4, nothing will be displayed. For my sample input, I input “1” to be able to get an output of a graph of NVDA and INTC on the same plot, as seen in figure 1. Below these images, I have two images of my entire code for my program.

```
/usr/local/bin/python3.8 /Users/jillianmcdonald/PycharmProjects/CSfinalproject/CSfinalproject.py
Please choose one of the following choices:
1. Display graph for NVDA and INTC
2. Display graph for INTC and AMD
3. Display graph for AMD and NVDA
4. Exit.
>>>
```

```
/usr/local/bin/python3.8 /Users/jillianmcdonald/PycharmProjects/CSfinalproject/CSfinalproject.py
Please choose one of the following choices:
1. Display graph for NVDA and INTC
2. Display graph for INTC and AMD
3. Display graph for AMD and NVDA
4. Exit.
>>> 1
```



```python
import yfinance as yf
from matplotlib import pyplot as plt


def load_ticker(symbol):

    ticker = yf.Ticker(symbol)
    hist = ticker.history(start="2020-03-01", end="2020-12-02")
    hist = hist.reset_index()
    for i in ['Open', 'High', 'Close', 'Low']:
        hist[i] = hist[i].astype('float64')

    return hist

def main():
    while True:
        print("Please choose one of the following choices: ")
        print("1. Display graph for NVDA and INTC")
        print("2. Display graph for INTC and AMD")
        print("3. Display graph for AMD and NVDA")
        print("4. Exit.")
        resp = input(">>> ")
        if resp == "1":
            h1 = load_ticker("NVDA")
            h2 = load_ticker("INTC")
            ax = h1[['Open']].plot(title="NVDA vs INTC")
            h2[['Open']].plot(ax=ax)
            plt.legend(["Open NVDA", "Open INTC"])
```

```python
        if resp == "2":
            h1 = load_ticker("INTC")
            h2 = load_ticker("AMD")
            ax = h1[['Open']].plot(title="INTC vs AMD")
            h2[['Open']].plot(ax=ax)
            plt.legend(["Open INTC", "Open AMD"])
        if resp == "3":
            h1 = load_ticker("AMD")
            h2 = load_ticker("NVDA")
            ax = h1[['Open']].plot(title="AMD vs NVDA")
            h2[['Open']].plot(ax=ax)
            plt.legend(["Open AMD", "Open NVDA"])
        if resp == "4":
            break


        plt.show()


main()
```

Section 2:

The target audience would be any investor interested in the semiconductor industry or in stocks in general. This program could give them an understanding of what stocks in the industry

are best to invest in during the pandemic, and to see which stocks in the industry are on an upward trend or a downturn. The user can see which stocks have momentum and if there is a possibility for alpha in those stocks.

Section 3:

Some of the specific programming techniques I used were functions, lists, loops and if statements. I imported yfinance modules and matplotlib modules to use their classes. I also used a user defined function “load_ticker” so I could pass ticker symbols into this function, which returns the ticker price history object. I used a menu driven loop that gives the user control of the number of times it iterates. As well, I used a list for column names in the price history data frame. Lastly, I used “if statements” to process the user input.

Section 4:

The major challenges I faced during this project were trying to get two graphs on the same plot. At first, the two stocks would graph on their own separate plot, so I had to figure out how to get them on the same axis. I was able to code one stock so it would open and graph on a plot, and then assign a variable to that code “ax”. And then I used the same code but modified it to represent another stock, and then used “ax=ax”, which made the two stocks graph on the same axis.

```
ax = h1[['Open']].plot(title="NVDA vs INTC")
h2[['Open']].plot(ax=ax)
```

Another issue I encountered was with naming the stocks on the graph and altering the legend so the stocks were named properly on the plots. I looked up how to do it on google and was able to find some code to make it work by using the plt.legend.

Section 5:

If I had more time to develop this program, I would include more stocks from the semiconductor industry, and import their data from yfinance. I would include these stocks as to compare them to the others, and create more graphs to give more information about the industry. I would also try to adjust my program so the user could input any two semiconductor stocks from the entire industry, and get a graph of their stock prices. I think this would definitely be a more advanced feature, but it would be interesting to see how it would work.