Ethan Marcano

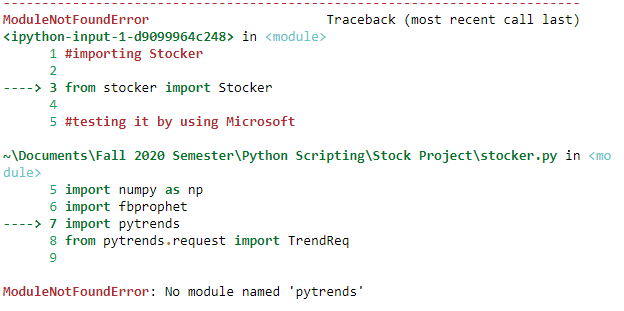
Dr. Pendleton

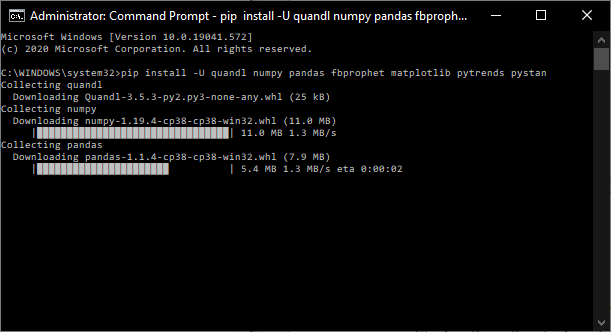
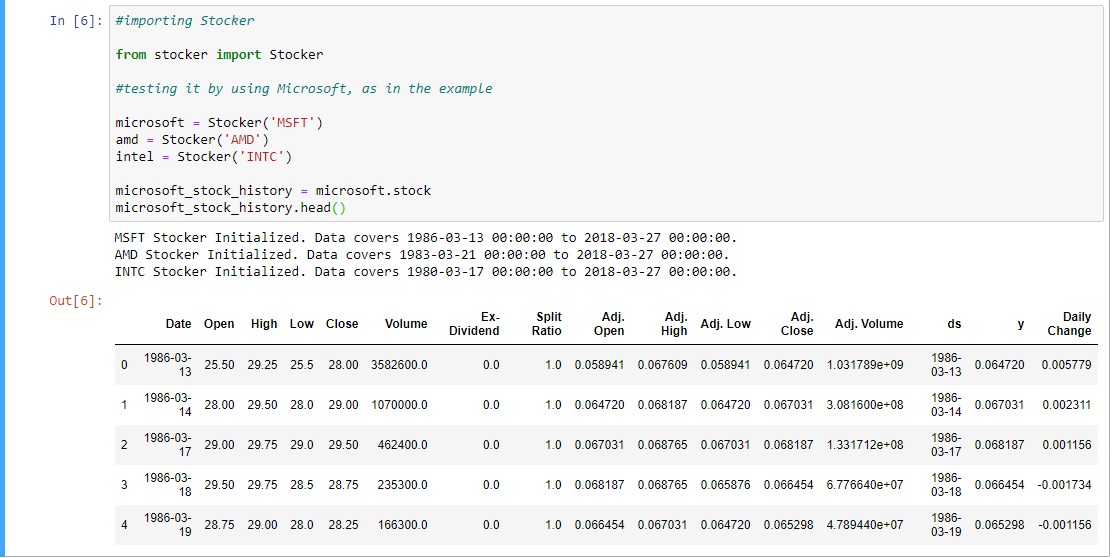
Python Scripting

Project Log

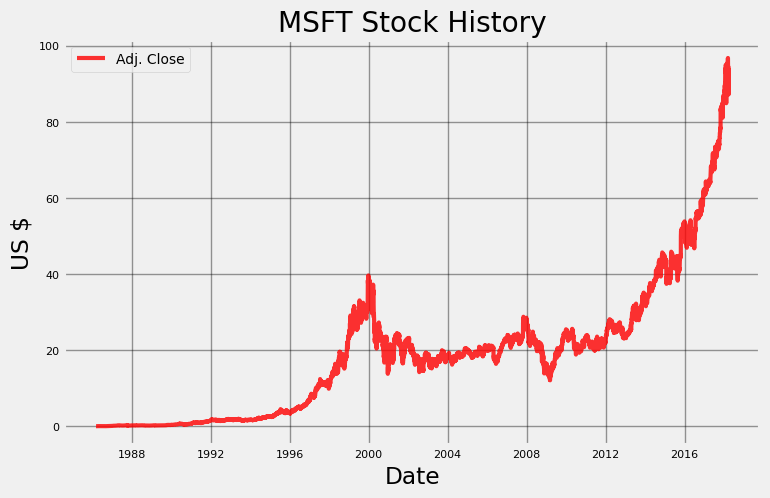
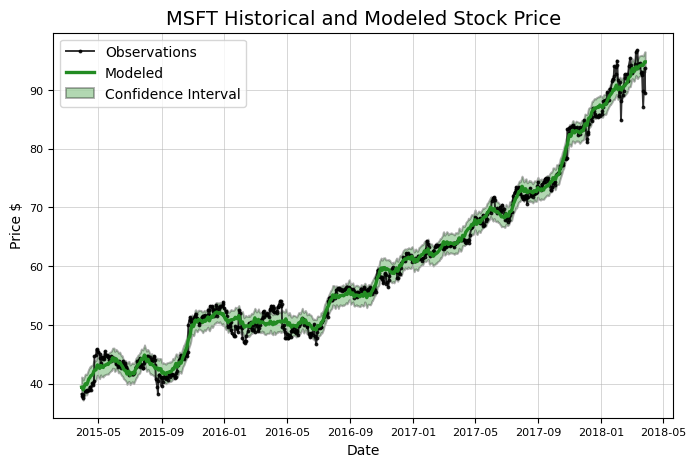
**October 29, 2020**

* The first entry of this project will detail the project finding process. Wanting to specialize in data-wrangling with a data science specialization from the university, I decided to look for a project that would enhance my understanding of python and its uses there.
* As for lines of code that I think the project will take, my estimate would be in the 50s to 70s range depending on how in-depth I want my analysis to be.
* This site is the main inspiration for the project: <https://towardsdatascience.com/stock-analysis-in-python-a0054e2c1a4c>
* This project details stock analysis in python, using libraries to grab data and model it, both historically and into the future. This would be a valuable experience in understanding python and its data-wrangling capabilities.
* I will work on this project initially 2-3 hours a week, with my time increasing depending on if I run into a roadblock or if I believe that the project is more time-consuming than initially thought.
* I will do all of the work on my own and list all sources of help used in this project, such as a classmate or stackoverflow. This includes linking the stackoverflow question that helped me solve a problem. I will only ask classmates for help.

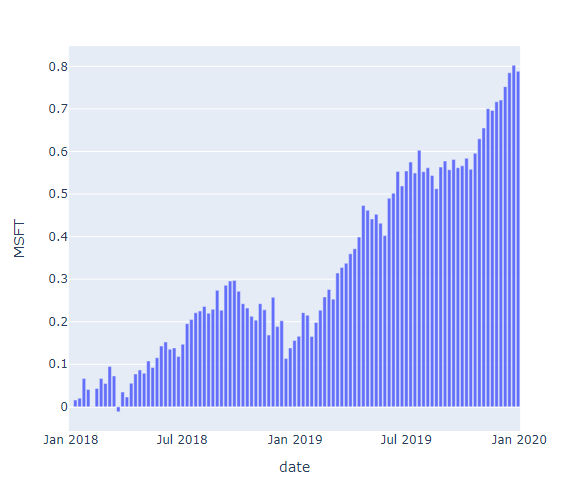
**Week of November 5, 2020**

* This week involved reading the basic documentation and installing the packages necessary for Stocker, the python tool needed to analyze and predict stock trends.
* This involved using Anaconda to get the right packages as well, and to ensure Jupyter Notebook integration (which Stocker accounts for)
* One of the packages involved required some troubleshooting to install properly: <https://github.com/facebook/prophet/issues/722>
* With all issues involving packages and dependencies solved, the tool has been installed.
* Testing the module, it seems that a package slipped through:
* After reinstalling pytrends in Anaconda, the error went away.
* I will do research in the future regarding this project on whether or not I can gather more current data (extending the range), but the current range is good for now (This will involve modifying Stocker as well as getting current data from Quandl).
* Once I get basic implementation down, I want try and compare data from competitors (in this case, AMD vs Intel), and plot their trends and analyze them as well.
* The foundation is there for future work regarding plotting and analysis.
* This week I worked for two hours, on set up and testing different parameters. I expect to maintain this pace barring any major roadblocks or changes in goals (in line with original project, of course).

**Week of 12th November, 2020**

* Much of this week’s work was dedicated to getting plotly to work, a graphing library, to work and produce the graphics I need for the project. I need to graph the data properly.
* I was able to get the example graphic working:
* I feel that I can experiment with more types of graphs if I continue to acquaint myself with plotly and Stocker to aid in my analysis of AMD vs Intel.
* Stocker has built-in arguments for the range and type of plot, and I may be able to expand on it with some studying.
* I also set up a quandl account; this will enable me to get more current data to plot. **UPDATE:** Unfortunately, it seems that the data feed is no longer actively supported. Some time may be taken to find a more up-to-date data feed, but the range should suffice.
* We are also able to model historical and modeled stock price

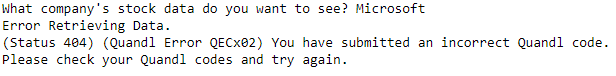
As the graph indicates, this is largely over a three year period. It can be argued that any extra past data would be unnecessary in analyzing “current” trends.

* Using plotly to produce a graph is more detailed, yet limited to the last two years. It can also be modified to compare data as well as zoom in on specific periods. If I can figure out how to overcome this limit, this might prove more useful than Stocker in that aspect and in the long run. I would like to be able to use this program without supplementary parts and excessive modification.
* I spent approximately two and a half hours this week working on the project. This is going as planned, although more time may need to be considered if I am to create a more detailed and aesthetically pleasing product.  
    
    
    
    
    
    
  **Week of 19th November, 2020**
* Following last class’s suggestion, I decided to refine my goals regarding the project, namely with a numerical list of what the program is to do.

1. User inputs stock name
   1. A function asking for the stock name
   2. Assign that stock name to a dictionary (or list?)
   3. The stock name then can be used to initialize and set up the data
2. The stock is initialized
3. The historical data is graphed
4. The predictive data is graphed
5. Continue until the user is done

* I want to try and make it so they can put in the colloquial name (Microsoft instead of MSFT, Google instead of GOOG).

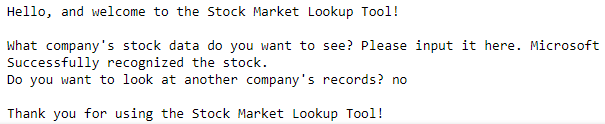
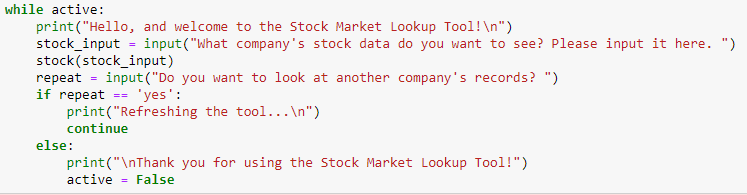
So far, much of my work has been testing and troubleshooting functions and classes, utilizing knowledge from class and occasionally consulting the Python Crash Course textbook for help.

The error I want to get rid of regarding using the improper name.

It has improved, but I lack repetition and actual results.

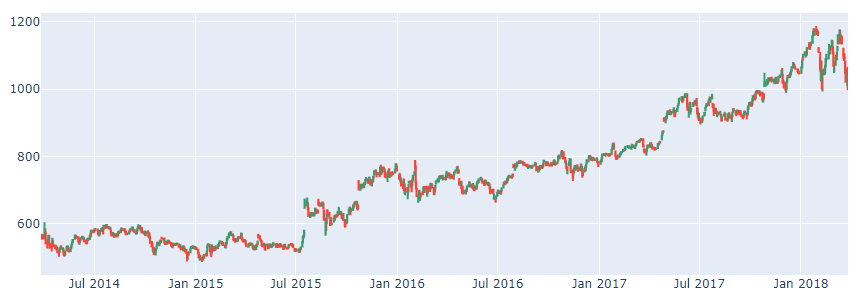
* Now that I have restated my intentions and written a basic plan down for my program, I feel better about my goals, and what I need to do.
* However, this does mean that I need to and that I *will* spend more time the following week tweaking what I have to make sure that it works consistently.
* I spent two hours working this week, and this is likely to increase by a couple hours or more in the next week as I understand what I need to do, as well as time opening up in general over the next couple weeks.

**Week of 26th November, 2020**

* The goal of this week was to get the meat of the program to work, including the looping function, as well as the colloquial-stock naming convention.
* The latter issue may be waived in lieu of asking the user to input the stock name if it cannot be solved in sufficient time. **UPDATE:** Going with asking the user to input the stock name in the first place.
* Although there was difficulty in getting the while loop to work originally (it getting stuck forever), there was a solution, in asking if the user wanted to continue instead of asking them if they wanted to quit.
* To bolster the viability and usability of the program, I will ask the user if they want to see alternative graphs, using plotly. The majority of the program is complete but polishing it and giving the user extra options in their analysis. This week I spent 3-4 hours polishing up the program, and I expect to spend around 2-3 hours before submitting my work. This will be complete by the final due date.

**Week of 2nd December, 2020 \*FINAL WEEK\***

* As detailed last week, this final week has been cleaning up the code and polishing existing features, as well as trying to compensate for different use cases.
  + Different use cases in this case meaning: Asking for a stock, not requesting extra graphs, asking for another stock; different patterns for different users
* I consulted the plotly documentation extensively while implementing the graphs, which enabled me to have a basic configuration that was easily customized. Again, plotly as a library comes with more current stock data, yet with a more limited span, hence its use as an extra option within the program. **UPDATE:** I have further integrated the quandl data with the program, and as such the program now consults the data instead of what comes with plotly (which was found to be lacking following testing of more obscure stock) <https://plotly.com/python/plotly-fundamentals/>.

Candlestick Graph implemented in the program

* I experimented with nested if-else statements within the program to deal with the different use cases. I found it to be useful, if ugly and/or inefficient. I am sure I can learn a better method in the future.
* A notable bug that I’ve encountered (not sure if related to jupyter’s outputs) is that the Stocker made graphs will not show up the first run through (with embedded text instead showing up), but will show up on subsequent runs.
  + **UPDATE:** This disappeared in later iterations of the program. Not sure what caused it. Occasionally comes back. Extended testing shows that it appears when the plotly graphs are loaded. Commenting out the graphs made them show on the first try. I am unsure if this is an issue with Stocker, plotly, or jupyter notebooks. Upon uncommenting the Stocker graphs work again. Again, this error could just be localized to jupyter.
* Polishing up the program, it worked as intended and was roughly ~53 lines of working code. It took roughly three hours to finish.
* Many of the resources consulted in this project were direct from the documentation of the libraries (Stocker, Plotly and Pandas), as well as from the textbook used in class, Python Crash Course 2nd Edition. Regrettably, I did not consult my classmates for the project, mainly because I was unsure of specific questions that would be useful for not only myself, but for others as well. I find that I like to work through a project myself thoroughly, and consulting documentation is more useful to me to remembering concepts and forces me to understand it myself before I ask for help. I understand that this will not be the case in the real world and that these things have a time limit (as well as having colleagues that may be working on a project with me) but the advantages of this project is that it enables me to work through different possible solutions at my own pace.
* As for the journal, I find that it was useful for writing down my thoughts and concepts regarding the project, as well as an alternate way of keeping tabs of my resources other than browser bookmarks. However, much of my thought process was internal, and as such some of my work did not reach the page other than the result of my thinking (shown through the UPDATES as well as different bullet points). Writing down what my program was to do (as suggested in one class) was a major help in understanding my goals for the project, as well as giving direction to my logging.
* The project was useful and interesting, and was useful in understanding and utilize APIs, and data visualization. In this way, it succeeded in getting my toes wet in that area of knowledge. I also found working with Stocker interesting, as it was a library and a toolkit specifically created for stock data visualization and required slight modification to get working. Working with graphing libraries forced me to look at coding in a different way, different than the standard exercises I had encountered before. If I were to come back to this project in the future, I would look at tidying up my code, as well as implementing dropped features, like comparing two stock values on the same graph, or somehow allowing the user to use the colloquial name in asking for the stock.