Explanation of Options Trading Assistant Code

"Options are a type of derivative security. Derivative implies that the price of the option is intrinsically linked to something else." – Investopedia

In writing our program, we aimed to build an assistant for options trading. A program that would be able to be your first stop en-route to figuring out which companies will be the most promising companies to buy options on. Many day traders know of options and understand the risk in dealing with these types of trades. The average investor usually does not have time for this type risky behavior since in some cases it can be a quick way to lose a chunk of money.

In giving a brief explanation of Options it is, in layman's terms, a contract that grants the investor the right, but not the obligation to buy or sell an underlying asset at a set price on or before a certain date. The right to buy is called a call option, and the right to sell is called a put option. What the general audience might want to know, is that a very basic level, the call option is betting that the stock will go above a certain strike price and the put option is betting that the stock will go below a certain strike price. The investor can sell their option at any point during the duration that the investor is holding the option. For example; if I buy an option on NIKE for January 18. I can sell the option at any point between now and January 18th. If I buy a call option, and the stock goes above the strike price that I estimated, I can sell to make a quick profit and vice versa for put options. However, if the stock goes in the opposite direction, you may lose up to all of the amount that you have purchased the option for. That's why you need to keep a close watch for added security or set alerts for every time the stock goes above or below a certain point, or even automatically set the option to sell itself!

From picking up day-trading this summer and finding the mutual interest of Zach Ventrice, we have decided to build this program as our final project. It is my theory as well as many other's that the volatility of stocks are much higher during a stocks earnings announcements. For this reason, this is the best time to invest in options, since there is much data circulating about the company, and since there is often an expected jump one way or the other. Our program aims to give you the upper edge during this time.I have searched around this summer to find a program that would output simple and relevant company data, and have not found it all on one platform. Our program uses the Google Finance API for company data, pulls performance information from Yahoo Finance for the graph, and pulls both the earnings announcements dataframe as well as analyst recommendation from NASDAQ (Yahoo and Nasdaq are not APIs).

The comments in the code give insight into what is actually happening, but let me run through that with you now.

Lines 5 through 19 will are import statements and will import all of the modules we will be using throughout or code. There are various different imports due to the functionality we needed to pull data from the web. These imports range from importing json to regex and many modules in between.

Lines 24 through 36 are the introduction to the program, these lines will print out our messages to the user, from instructions to what the code will do. The inputs ask the user for a date so that they can decide when they would like to view earnings releases for.

Line 42 formats the user input to replace it into the NASDAG (website) link in a "Dec-12-2017" format so that will load the correct information.

Lines 48 through 59 check the website for valid dataframe. It does this by checking for a specific column [‘Time’] in the dataframe, since it is the only dataframe with that format on the NASDAQ webpage. If it does not exist, it is due to the date too far in advance or the date being a weekend. If this is the case, an “Oops” statement will appear.

Lines 67 through 83 will take the data from the dataframe and split the desired column into the list. I wrote a regular expression script - '\(([A-Z]+)\)' - that will take A-Z in parenthesis out of the company names so that only this list is available to be used as valid input in the GetStockData() function.

Lines 89 through 149 are defining the extensive function known as GetStockData. This function searches the Google Finance API for all relevant stock information from trading price to profit margin. If this code does not run correctly and “Oops” statement will also come up.

Lines 157 through 169 define the GraphPostOrPass() function. This function searches Plotly to pull an image of a graph that is linked to data in Yahoo. This graph will only post if the stock is valid and the corresponding graph exists, otherwise an “Oops” statement will be the output.

Lines 175 through 183 define the AnalystRecommendation() function. This function pulls the analyst recommendation from the corresponding stock straight from NASDAQ or an “Oops” statement will output with “No Data.”

Lines 188 through 195 define the GetAllData() function which will run all three of the previous functions (GetStockData(), GraphPostOrPass(), and AnalystRecommendation()) if and only if stock_choice of the user is in the list valid_choices.

Line 205 calls df which is the original dataframe taken from the NASDAQ website for earnings announcements. *Unfortunately, we were forced to write our code this way since Jupyter would not allow us to display the df in a conditional statement. Due to this reason, we were unable to add the desired loops to our code. If there is a name error when trying to run code for first couple times, we are not able to fix that since we cannot display "df" in a try and except function or any other conditional method due to issues with Jupyter.

At the end of this first block, after the dataframe is printed, the user will run the second block of code (Lines 1 though 3). These lines print valid_choices for the user’s readability of the list, ask for stock_choice of the user, and calls the function GetAllData() in order to output all of

the relevant information for the stock (General performance, performance graph, and an analyst recommendation).

Space in between specified lines are used for in-code comments (This is why there are jumps).