Ryan Malani

IT 129

Final Project Diary

**May 24:**

I simply read the project description thoroughly to get a good understanding of the task at hand and headed to the discussion board to answer any questions my peers may have had.

**May 25:**

I finished reading all of the chapters in our textbook regarding this week’s content in order to get a broader understanding of the use of files and functions in Python. This allowed me to conceptualize the use of a file or API for data in a script.

**May 26:**

I began thinking about what type of project to do, but I struggled to think of anything original and fairly complex or useful in my personal life that would be considered a “small project” or could be comprehensively addressed in 30-50 lines of code. After brainstorming for a little while I ultimately decided to seek the assistance of Google the following day.

**May 27:**

I began searching for potential projects today, starting with a simple and quick Google search of “python small projects”. However, this was to no avail as the majority of the projects were extremely redundant or boring to me. I knew that if I wanted to get the full value out of this project, I would need to be working on something related to another one of my passions. I had to drive to Philadelphia which cut short my research time, but it allowed me to think during my drive what I might like to start looking for and working on.

**May 28:**

I recognized that two of my main passions lie in the automotive industry and the stock market. I have identified that while I would love to work on automating a process in the automotive field, I don’t have an extremely large window to work on this project as you mentioned. Therefore, I decided to begin searching for a programming project related to the stock market. Being that the project requires the use of a file or API, I recognized that I wouldn’t be able to import market data into a .txt file, at least easily, so I began my search for a free and intuitive API. That search led me [here.](https://towardsdatascience.com/python-how-to-get-live-market-data-less-than-0-1-second-lag-c85ee280ed93) This article outlines the free Yahoo Finance API, which collects and outputs live market data, and how to integrate it in the script. I then decided how I was going to add my spin to the project with the buy/sell signals. Similarly to [this project](https://medium.com/analytics-vidhya/python-i-have-tested-a-trading-mathematical-technic-in-realtime-658a80381151) by Sajid Lhessani, I will be using a technical tool. However, in my experience trading, I prefer the use of the volume profile (more information on that tool and the technicalities behind it can be found [here](https://www.tradingview.com/support/solutions/43000502040-volume-profile/)) which I will be swapping out for the Bollinger Bands that he used. The volume profile tool has three main parts – the value area low, value area high, and point of control. I will be using these three parts to send, or output, alerts and buy/sell signals. I will further personalize this project by prompting the user to select a time frame and ticker which will be entered into the Yahoo Finance API.

**May 29-31:**

I didn’t work on the project at all as I was awaiting the approval/rejection. However, I did continue to look for different small projects to have a backup plan in my spare time.

**June 1:**

Today I began to dig deeper into the mathematical background behind the Volume Profile tool in order to get a better understanding of how I will build out the parameters for the value area low/high and point of control. I still haven’t begun coding, but I received confirmation that my project was good to begin working on so I plan to start coding and playing with the API tomorrow.

**June 2:**

Began to write my script and tried running my program with just the imported packages in terminal to see if everything was installed correctly and it appears I’ve run into my first snag. I went back and ran the pip install numpy command but it kept failing due to PEP 517 and building the numpy wheels. I remedied this by redownloading Anaconda3 which includes the numpy and pandas packages. I then opened a new terminal window and ran the pip install yfinance and pip install plotly commands.

**June 3:**

Continued to work on the project, this time gathering user input as arguments to pass into the API. Was running into some issues using the user input in place of the API setup as I was getting tracebacks for the time data not matching the format of the API. However, my issue was that I wasn’t following the API argument format correctly at all, and it wasn’t a data type issue. I resolved this issue by changing “data = yf.download(tick, per, inv)” to “data = yf.download(tickers=tick, period=per, interval=inv)”. The script worked perfectly after this simple fix. I then played with different configurations of the plotly graph to show time frames that would work best for showing my desired technical indicator.