# Problems

For my project I created an automated stock analysis tool. I got the idea while working on a project for my seminar in investments class. I found analysis steps such as downloading data, calculating beta, sharp ratios, and market covariance etc., to be a bit arduous and repetitive. Additionally, since I was manually repeating these tasks, I was exposed to more errors and ended up having to replicate work unnecessarily. Tasks like this tend to be lend themselves towards automation. Lastly, existing tools are locked behind prohibitively expensive paywalls and can be tricky to interpret.

# Goals

So since I had to make a project for this class, I decided to create my own tool. The overarching goal being to make a tool that retrieved stock data, automate the analysis of the data and to help the user make investment decisions. For this project to make any sense it had to be able to handle more than one security at a time and automate the analysis to prevent manual errors. Lastly, the tool had to have an accessible GUI that presented the data in an understandable way to prevent errors in interpretation of the data.

# Capabilities

While this Stonker the name for my tool is still very much in the “Alpha” stage of development I believe I’m well on the way to accomplishing these goals. Currently Stonker uses the Alpha Vantage API to retrieve the history of returns a user requested. As of right now, that only allows for over 20 years of data to be returned for any ticker symbol. It then successfully consolidates, analyzes and displays the data in tables and graphs. Users can change the tickers being analyzed, weight of each ticker in the portfolio and their own risk aversion level.

# Challenges

While this is going great so far, there have been several challenges developing this tool. The largest so far has been learning how to work with the Dash framework developed by Plotly. Dash is the engine that creates a local server and displays all the GUI elements. It does this by translating python code into HTML and Javascript, all while integrating a separate css file and your python project files. It also automatically deploys a flask application.

While this is all great, the logic behind dash is a bit unintuitive and it seems to be a bit buggy with hit and miss documentation. I chose it because it’s a framework purpose built for creating dashboards in python, which is essentially what this project is.

The other large challenge to overcome occurred about halfway through my project. I changed my approach from a linear mindset that was a chain of functions to an object-oriented approach. I did this because the code was becoming unwieldly, and integration with dash seemed like it was going to be even more complicated than it already was. While this change I believe ultimately paid off, it did cause me to rewrite about half of my code and wrap my head around new logic.

# Highlights

I think the interactivity of Stonker is one of the bigger highlights. Ability to change your risk aversion level and the weights of each security is somewhat unique in the market. Additionally, the ability to download all the data already formatted and accessible is unique likely due to licensing issues. The tool is entirely written in Python and CSS which makes it easy to modify. Lastly, thanks to Dash, it is easily distributable and deployable to other people if need be.

# Oppotunities

With all of that being said, there are still more things I’d like to accomplish and just did not have time for. Namely, more graphs and more analysis such as the efficiency frontier and recommended risk weighted weights for each portfolio. There are also some GUI tweaks such as selecting tables via dropdown vs the long scroll that exists currently.

**DEMO**