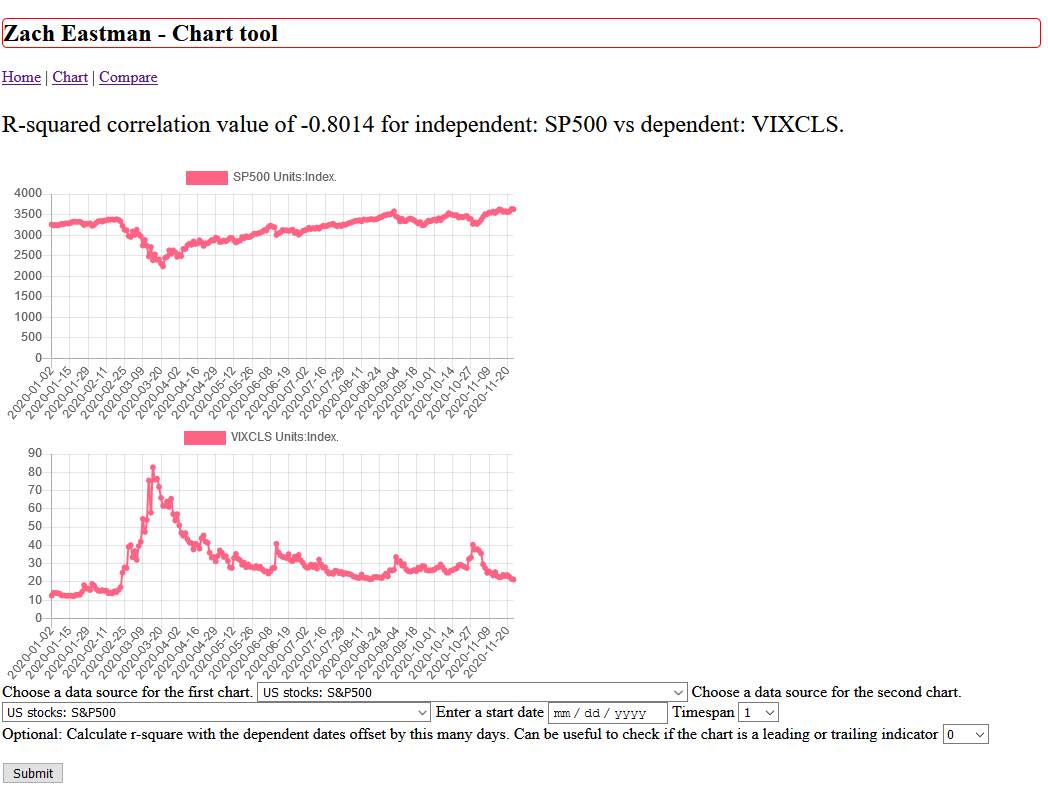
For the most part, the old artifact was REST API trying to be Yahoo finance lite. Rather than be a useful out a program, the code implements a API a developer could hypothetically include in their code to access stock data. Again, I’ve reworked that to focus more on the end user experience. I will include some screenshots and video of the end-product in the final because this code requires the external libraries fredapi, Chart.js, and Flask to run, but here’s what it looks like right now.



I enjoy charts and think they would make a lot more visual appealing portfolio website. I can’t do it on Github Pages, because it only takes static content, so again I’ll demonstrate these pages with pictures and video.

To be extremely reductive, I feel that a lot of popular finance boils down to “how much does this set of numbers look like this other set numbers?” Math conveniently has an algorithm for that, R-squared.

I have been dissuaded combining the two charts into a double y-axis charts after reading Tim Duy’s blog, in which he convincingly argues that humans are bad at estimating how correlated to charts are. Instead I just put two charts side by side.

I got better and Flask and it’s built-in templating language Jinja over the course of this. Turns out Jinja implements a clone of Python dicts, including the items() function and being insertion ordered. That reduced a lot of headaches from me. I find working with lists of tuples containing a date and price to be needlessly more complicated than a dict with key: date, value: price. I will be using Flask next time I want to make a website, it’s quite enjoyable to use.

I also spent hours debugging my rsquare function. It kept outputting values in the range of -6122, when it’s supposed to give a value between -1 and 1 in most cases. Several print statements later, it dawned on me that ^ is in fact not the exponentiation operator, but bitwise XOR. Exponentiation is star star. I’m going to be repeating that and afterward the function tested against Excel’s RSQ and handling some null and random data perfectly.