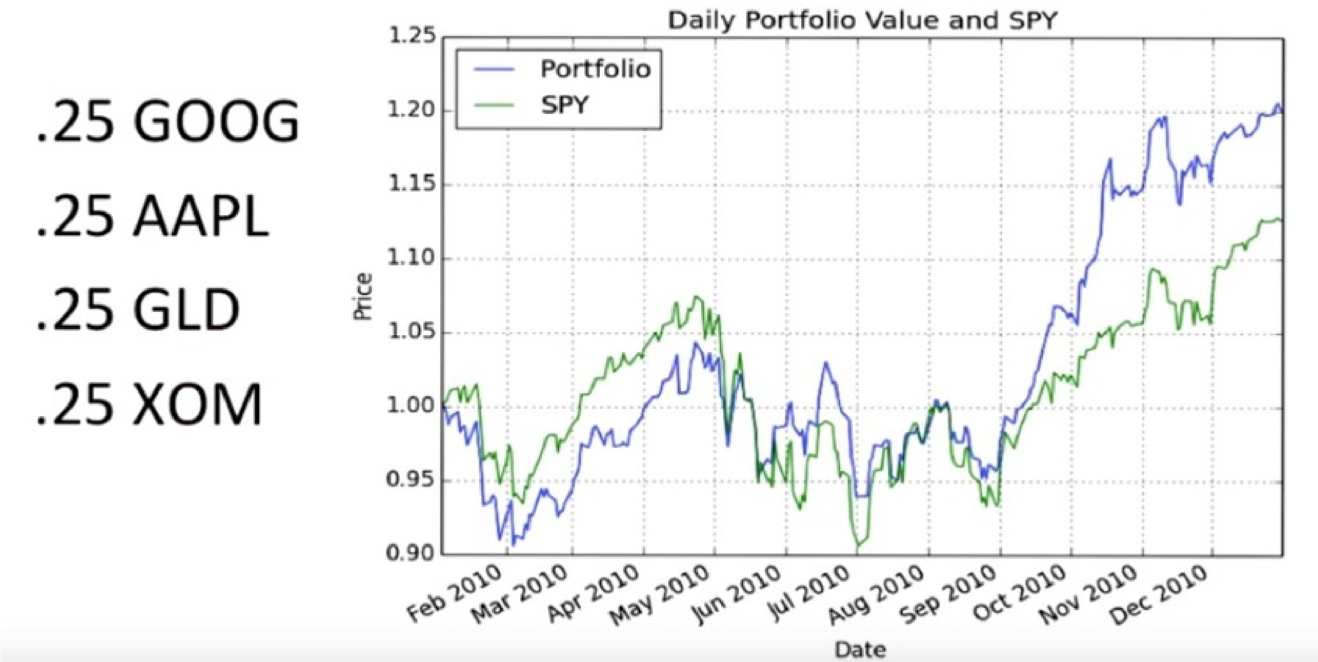
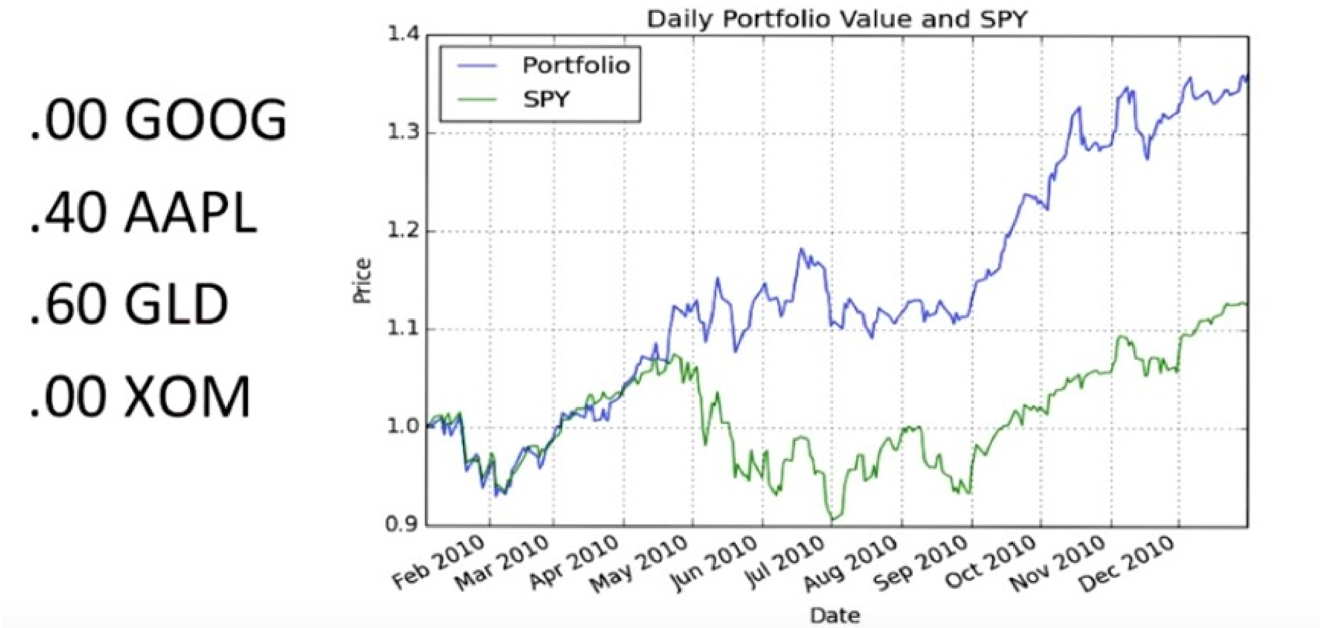
1. Your final project in this mini course is to create a portfolio optimizer. That might sound like a tough problem, but you've already got the tools to do it. In this short lesson, we're going to lay the groundwork to help you know what portfolio optimization is and how to build one. First, what is portfolio optimization? Given a set of assets and a time period, find an allocation of funds to assets that maximizes performance. What is performance? We could choose from a number of metrics, including cumulative return, volatility or risk, and risk adjusted return, which is Sharpe ratio [LAUGH]. You can use any criteria you like, but for this assignment, we're going to focus on Sharpe ratio. Now, let me show you how to use a minimizer to optimize a portfolio.

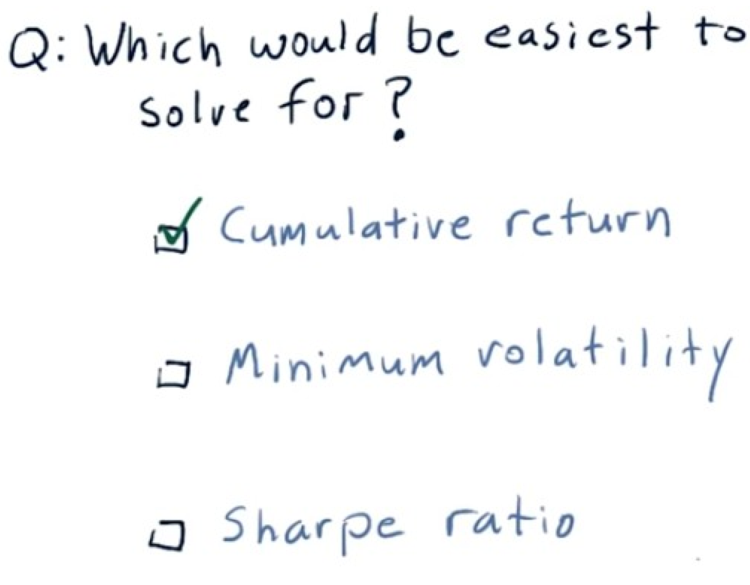


2. As an example consider this portfolio where we have Google, Apple, Gold, and Exxon from the beginning of 2010 to the end of 2010. This is what the performance of that portfolio would look like in the blue line, if we had an equal allocation to each asset. So, we gave 25% to each of those stocks at the beginning, and then we observed at the end, the total performance. And we include a chart here for S&P 500 for comparison. Now, this is an un-optimized portfolio.

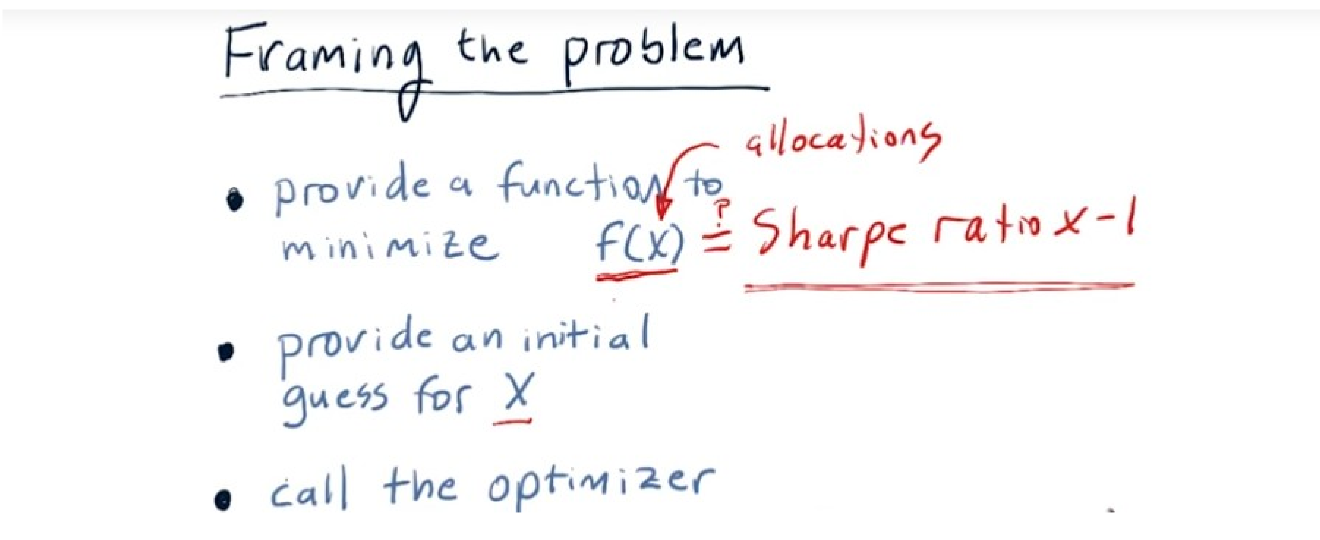


Here's what it would look like if we optimized for sharp ratio. Look at that. Look how much higher performance we get compared to our benchmark S&P 500. And observe over here the allocations to the assets. 60% goes to Gold, 40% to Apple and none to Google or Exxon. Now, one thing to emphasize here is, this is looking back in time, so we're looking back historically. What happened in 2010 and observing at the beginning of 2010, if we knew what we knew now, how should we best allocate? So, there's a question as to how much this can help our portfolio going forward. And I'll give you that answer ahead of time. Indeed, when we optimize for sharp ratio, hold that portfolio, rebalance, in other words re-optimize that sharp ratio and continue that month by month, we very often, most frequently see that that improves the performance of the portfolio versus just an equal allocation. But I do want to observe that this is computing back in time, so, yes, of course we can find a better portfolio, but indeed it helps going forward as well.

3. Suppose I were to give you the following problem. Take these four stocks and over a particular time period, I want you to find the optimal allocation to those four stocks. I might ask you to find the optimal allocation that maximizes cumulative return, minimizes volatility, or maximizes Sharpe ratio. Which of these optimizers would be the easiest to write and why?

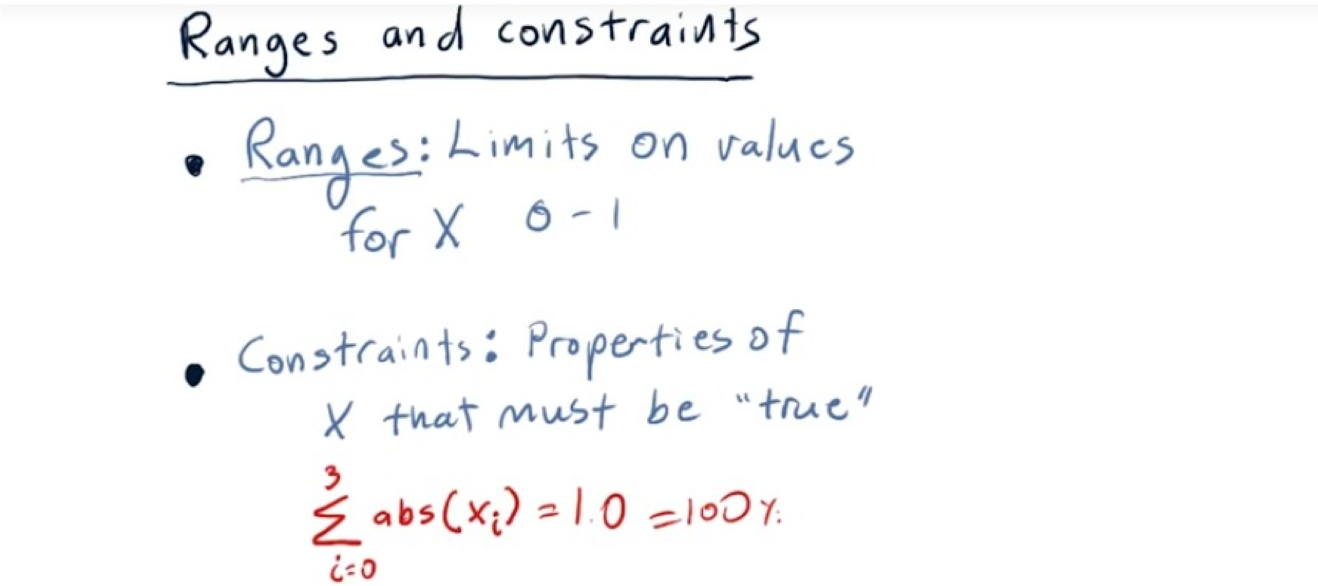


4. The answer is, it would be easiest to write an optimizer that optimizes for cumulative return. Because all you have to do is find the single stock that maximized return over that time period. If you're going to optimize for minimum volatility or Sharpe ratio, you actually have to evaluate various combinations of those stocks. In this case the allocation would just be 100% to the highest returning stock.



5. We're going to focus here on optimizing for Sharpe ratio. So solving that problem is not trivial, but it's not too hard either. What we need to do is frame the portfolio optimization problem as a minimization problem, and then we can solve it using the tools you have already. As you recall, in order to use a optimizer that minimizes, we have to do three things. First, we have to provide a function to minimize f(x) that takes in x, two, an initial guess for that x, and three, call the optimizer and let it run. Now, in this case, x are the allocations that we're looking for. And we want the optimizer to try different allocations in order to discover the best set of allocations that optimizes this function. Well, what is that function exactly? We said just a moment ago that we want to optimize for Sharpe ratio. So, is this just a Sharpe ratio? Well not exactly, because what the minimizer will do, in this case, is try to find the smallest Sharpe ratio. So it'll find allocations that minimize that. And we want, of

course, the largest Sharpe ratio, because larger Sharpe ratios are better. That's easy to fix. All we do is multiply this by -1. So, all that we want our optimizer to do is optimize for a negative Sharpe ratio. And that'll find the best allocation or the best value for x to solve this problem. And remember, x can have multiple dimensions, so each dimension of x here is an allocation to each of the stocks. So, if we're trying to solve for a portfolio of four stocks, x will have four dimensions, and the value for each of those dimensions is the percentage of funds to allocate to each of those stocks.



6. There's two more things you need to know about before you start optimizing portfolios. One of them will help your optimization run faster. And the other is essential for you to get the right answer. We'll start with the faster thing first. The first thing you can do is you can tell the optimizer that it should only look at certain ranges for X. In other words, for this problem, for each of the various allocations, it's only worth looking at values between 0 and 1. In other words, the value of 2 would indicate 200% of your fund is in a particular asset and that's not possible. It's only feasible to have 0% to 100% or 0 to 1 in each of these assets. So, you can tell the optimizer only focus on values between 0 and 1 for each of the dimensions of X. And if you do that, the optimizer can run much more quickly because it knows not to look at other values of X outside those bounds. It limits the search area significantly. Another thing you can do with the optimizers in numbpie are provide constraints. Constraints are properties of the values of X that must be true. As an example we want the sum of our allocations to add up to one. So let's say for example we're for portfolio of four holdings. So we have X0, X1, X2, X3. We want the sum of the absolute values of those to be equal to 1.0. In other words, our total allocation should add up to 100%. The optimizers we use in this class have the ability for you to express that and that guarantees that at the end when it finds out the values for X, you end up with a total of 100% allocated to the various assets. Now we're going to show you how to do these two things in the assignment text itself, because the syntax is a little bit tricky and we want to convey to you exactly how to do that. So we look forward to seeing you solve this problem and good luck on your final project.