Final Project: Modern Portfolio Risk Management

Authors: Advay Koranne (ak845), Jack Young (jry28), Blaze Ezlakowski (be96)

Contents

1 Annotated Checklist

1. How does our project relate to the linear algebra part of CS 3220?
Answer:
Our project relies heavily on linear algebra, particularly with the usage of matrices and matrix algebra. Most of our data is stored in vectors and matrices; for example, the asset weights are stored in a vector where the ith entry corresponds to the proportional weight coefficient of the ith stock. Another obvious linear algebra structure is our covariance matrix, which simplifies the structure of many necessary multiplications for calculating portfolio standard deviation and variance; matrix algebra makes such tasks very smooth. A few examples of variables for which we use vectors and matrices to streamline computation are the following:
   (a) Σ: the co-variance matrix of asset returns
   (b) $\vec{w}$: portfolio weights
   (c) $\vec{R}$: expected returns
   (d) q: is our risk tolerance where 0 is the minimal risk and ∞ is the maximum risk

   We will make calculations using these vectors and matrices to find the efficient frontier of asset weights.

2. How does our project relate to the statistics part of CS 3220?
Answer:
The two most valuable statistical tools for our financial analysis are variance and co-variance. In the field of finance the variance of a portfolio represents an approximation of a portfolio's risk, and the co-variance of two assets describes how one asset will respond to a change in the price of the other asset. The variance of a portfolio is a value generated from the variances and proportional weights of the individual assets that comprise the portfolio. The co-variances must be calculated iteratively by multiplying the differences between the specific weekly returns of each asset and their mean weekly returns. A high individual asset variance corresponds to a large standard deviation of returns. We choose to model the returns of a stock over a certain time period as

normally distributed around a mean vector with standard deviation represented by variance. The mean vector is represented by the expected return of the stock over the examined time scale.

3. How does our project relate to the optimization part of CS 3220?
Answer:
For a given risk tolerance there should be an optimal portfolio that maximizes potential return at that risk level. Since the risk level of a portfolio is represented by the portfolio variance, and the portfolio variance is lengthy to compute, instead of sweeping along the dimension of risk to draw the Efficient Frontier curve we allow freedom along the dimension of "risk tolerance". This risk tolerance value is represented by a scalar value denoted $q \in [0, \infty)$. If we were to be maximally intolerant of risk, we would say $q = 0$, and these risk preferences would declare: "I don't care how high returns are, I want as little risk as possible". The most efficient portfolio for these risk preferences would be the leftmost point on the "Markowitz Bullet", and as we allow for higher risks in exchange for higher returns, and we sweep q from $0 \rightarrow \infty$, we draw the upper half of the Markowitz Bullet, declaring "I don't care how risky it is, I want the maximum returns possible!". We attempt to find the Efficient Frontier by minimizing the following expression:
$$w^T \Sigma w - q * R^T w$$
Where the left term can be thought of as 'risk' and the right term as 'q * returns', so when q is 0, the returns term drops out and we are only minimizing risk.

4. What is the application or the problem that we are solving/exploring?
Answer:
Modern portfolio theory is a widely used framework for maximizing the expected returns of a portfolio. The applications of this are fairly obvious, if we are unknowingly exposing ourselves to unnecessary financial risk we ought to know about it, and perhaps we ought to try to minimize it.

5. What is the goal and how are we evaluating this goal?
Answer:
The goal is to be able to backtest our algorithm and evaluate it's performance. We will take data from multiple years back and test our algorithm on the most recent past month to evaluate it in a similar environment to current market conditions while remaining capable of evaluating it quickly. We will be able to see how an optimized portfolio compares to common market indicators like the S&P.

6. What is the theory, relevant to our application/problem, that we will discuss?
Answer:
A fundamental theoretical assumption we make is the efficient market hypothesis, which assumes that assets are accurately priced. Another assumption is that the variance in periodic returns of an asset consists of an accurate measurement of risk, when in reality risk is borne of many other sources, and may be unforeseeable. Modern portfolio theory is also at root a technical analysis tool, so there is no consideration of the fundamental value of the underlying assets in relation to reality, they are simply assumed to be probabilistic event generators. For this reason, MPT is more commonly used for collections of ETF's than individual stocks, but we would like to evaluate the performance of an MPT based algorithm on many different types of assets.

7. What have we implemented?
Answer:
We have implemented a way to find the best distribution of money to maximize returns given a set of stocks for a given risk tolerance. Using a gradient descent / and brute force algorithm, we have a way to visually see which portfolio arrangements maximize returns.

8. What are the sources of our data? (Or state that you have created synthetic data.)
Answer:

We will be using yfinance which uses Yahoo's public API to provide stock data.

2 Introduction

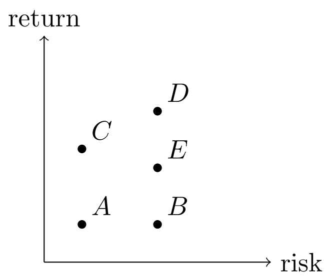
The topic we have decided to research is portfolio risk management in the field of quantitative finance. The primary motivation behind modern portfolio theory (MPT) is that given a portfolio of stocks and a risk tolerance, how do you maximize returns while staying close to your risk. Modern portfolio theory falls into the broader category of Quantitative analysis which leverage mathematical and statistical models for financial investments and managements.

The field of finance has changed drastically over the years — and the implementation of mathematical models, statistics, probability, and computer science has drastically improved the ability to make returns based on things that humans were never able to see before. "A study in 2019 showed that around 92% of trading in the Forex market was performed by trading algorithms rather than humans." "A report in 2009 stated that high frequency trading firms made up between 60% and 73% of all US equity trading volume." [1] Many of the top hedge funds which trade millions of dollars per day and there trading strategies rely primarily on probability, statistics, and computer science. The theory was invented by Harry Markowitz in 1952 and he was awarded a Nobel prize.

3 Application and Goal

Let's say you have a portfolio of 10 stocks from the Standard Poor's index (SP 500) and you want to find the optimal way to invest your money in the stocks you hold. How should you distribute your money into investments in each stock? Let's say you have the following stocks in your portfolio (denoted with their ticker symbol): ['AAPL', 'MSFT', 'AMZN', 'FB', 'GOOGL', 'GOOG', 'TSLA', 'NVDA', 'JPM', 'NFLX']. An intuitive solution would be to hold an equal dollar value in each stock — we have 10 different stocks in our portfolio, so the value to invest in each stock should make up 10% of the total dollar value of the portfolio. If you invest $100, you would own $10 worth of AAPL, FB etc... But what if it were possible to change the way you distributed your investments such that you achieve a higher return on investment? By only moving money between the stocks in your portfolio, how high could your returns get?

Let's say we are given a choice of many portfolios (A, B, C, D, E) with expected return and implied risk showcased by the location of the portfolio on the graph below.



Which of the portfolios should you choose to invest your money in? MPT assumes that you are risk averse, which means that if you have two portfolios with equal expected return, you would prefer the less risky one. For example you would always choose portfolio A over portfolio B, because they have the same expected return, and B is riskier. Similarly, no matter how tolerant of risk you are, you should always choose portfolio A over C, and portfolio E over B or D, because risk levels are shared within these groups of portfolios. Among portfolios with the same risk level, the best choice is the one with the highest returns. Because of this, we can conclude that A and E exist on the "Efficient Frontier" for this group of portfolios. We cannot definitively conclude which of A and E is the 'better' portfolio, as risk preferences vary across individuals, however we do know that none of the other portfolios would make for a better choice.

[1] https://en.wikipedia.org/wiki/Algorithmic_trading

In theory, the number of possible portfolios is infinite, as we can assign any proportional weight to every asset in the portfolio, so for example we could buy 0.01$ of Amazon and sell 0.01$ of Facebook to slightly shift our portfolio to a new expected return and risk level. In reality, we are constrained by the amount of money we have to invest, and the fact that shares are usually traded whole, though fractional trading is supported by some brokerage firms. Despite these constraints, we still have access to the actual efficient frontier, and we have created the possibility of working within your constraints to construct portfolios that approximate points on the efficient frontier. Our goal is to find the MPT for various groups of assets using statistics, probability, and optimization.

4 Theory

The first and simplest equation we will be implementing and using is the expected return of a portfolio[2]. We define μ_i as the expected value of the ith asset, and σ_i as the variance of the ith asset. So in order to calculate the expected return of a portfolio we can do the following calculation:

$$\sum_{i=1}^{N} w_i \mu_i = \begin{bmatrix} w_0 \ldots & w_1 \end{bmatrix} \cdot \begin{bmatrix} \mu_1 \ldots \\ \mu_i \end{bmatrix}$$

Furthermore we discussed in the introduction that portfolio variance can tell us the risk of a set of assets. We can calculate the variance of a portfolio via the following equation:

$$\begin{aligned} w^T \cdot \Sigma \cdot w &= \\ \begin{bmatrix} w_0 \ w_1 \end{bmatrix} \cdot \begin{bmatrix} \sigma_A^2 & \sigma_{AB} \\ \sigma_{AB} & \sigma_B^2 \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \cdot \end{bmatrix} &= \begin{bmatrix} w_0 & w_1 \end{bmatrix} \cdot \begin{bmatrix} \sigma_A^2 w_0 + \sigma_{AB} w_1 \\ \sigma_{AB} w_0 + \sigma_B^2 w_1 \end{bmatrix} \\ &= w_0(\sigma_A^2 w_0 + \sigma_{AB} w_1) + w_1(\sigma_{AB} w_0 + \sigma_B^2 w_1) \\ &= (\sigma_A^2 w_0^2 + \sigma_{AB} w_1 w_0) + (\sigma_{AB} w_0 w_1 + \sigma_B^2 w_1^2) \\ &= (\sigma_A^2 w_0^2 + \sigma_{AB} w_1 w_0) + (\sigma_{AB} w_0 w_1 + \sigma_B^2 w_1^2) \\ &= \sigma_A^2 w_0^2 + 2\sigma_{AB} w_1 w_0 + \sigma_B^2 w_1^2 \\ &= \sigma_A^2 w_0^2 + 2\sigma_A \sigma_B w_1 w_0 + \sigma_B^2 w_1^2 \ \blacksquare \end{aligned}$$

5 Application and implementation

In order to implement the "efficient frontier" we first got all the data needed from yfinance for our portfolio of 10 stocks. We wrote the the following functions to help us compute the important metrics we needed:

1. portfolio_returns(weights): which returns the expected portfolio return based on the weights and the means of the stocks.
2. portfolio_std(weights): returns the portfolio standard deviation based on the weights and the co-variance of the stocks.

5.1 Code and Output:

Using yfinance we are able download stock data from a start date to a end date. The following data table below shows the co-variances of the stocks:

AAPL	AMZN	FB	MSFT
-0.009770	-0.012213	-0.005305	-0.012530
0.007937	0.014776	0.018658	0.002582
-0.004714	0.002089	0.002161	-0.009160
0.015958	-0.007839	0.010087	0.015803

[2]Referenced Cornell Quant Fund notes

5.2 Experiment

Our experiment then consisted of running 3000 trials where we generated 10 random weights which summed to 1 calculated the return of the portfolio based on these weights and then also calculated the standard deviation. We then plotted the standard deviation (risk) vs. the returns.

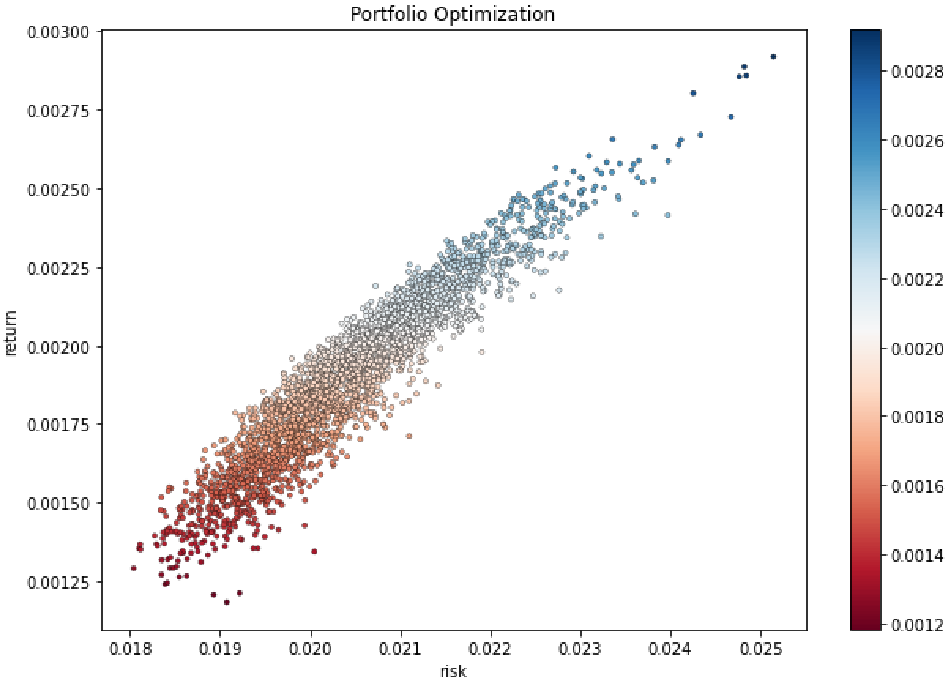
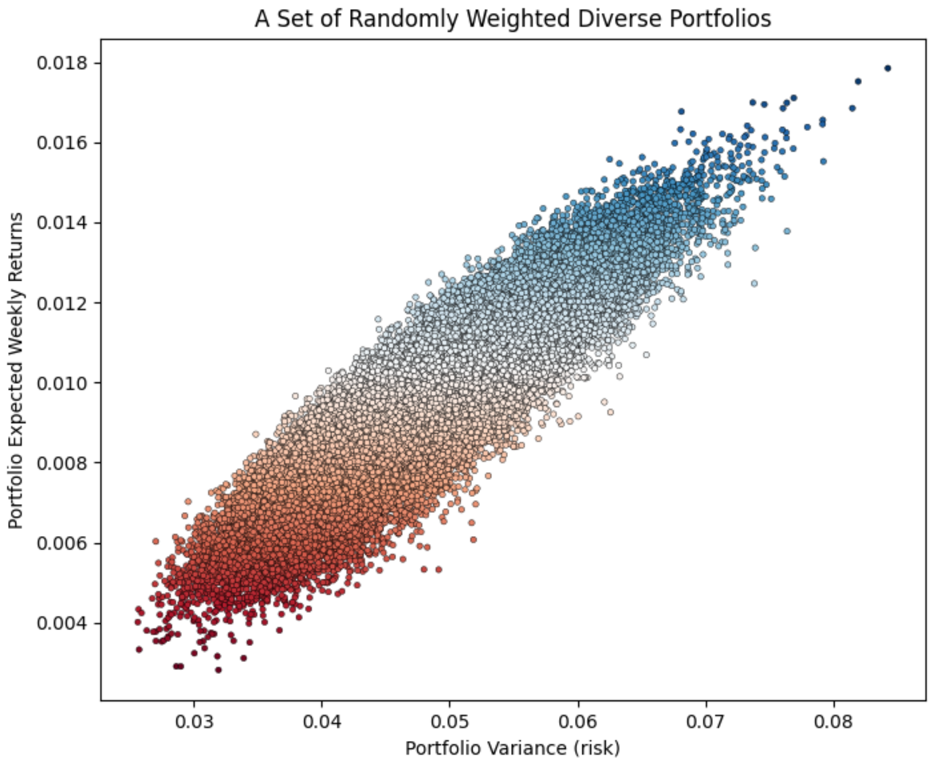


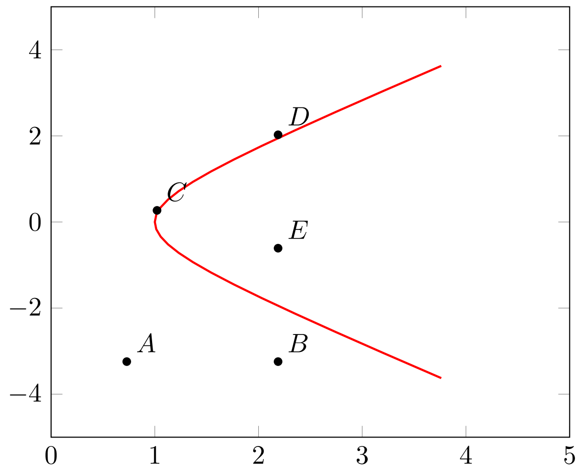
Image made using numpy (normalized)

This trial ran for 100000 points, with a large and diverse portfolio consisting of ETFs, precious metals, retail and tech stocks, commodity indices, crypto-currencies, and real-estate investment trusts.



6 Efficient Frontier

As mentioned in the introduction the goal of MPT is to find the maximum returns for a given risk. We showed that in the graph above it does not make sense to choose the portfolio which gives the highest return for the same risk. Hence our goal is to find the boundary curve around the graph above i.e we want to draw the following line:



For a rough example — we would want the line in red to be our efficient frontier because that would allows us to maximize the returns for a given risk. We can find the "efficient frontier" in red by minimizing the following equation according to our risk which we denote $q \in [0, \infty)$:

$$\underbrace{w^T \Sigma w}_{a} - \underbrace{q * R^T w}_{b} \tag{1}$$

Where (a) is the variance of the portfolio return and (b) is the expected return of the portfolio. We want to minimize for $\vec{w}$ because that will tell us what our portfolio should be made of.

7 Algorithm

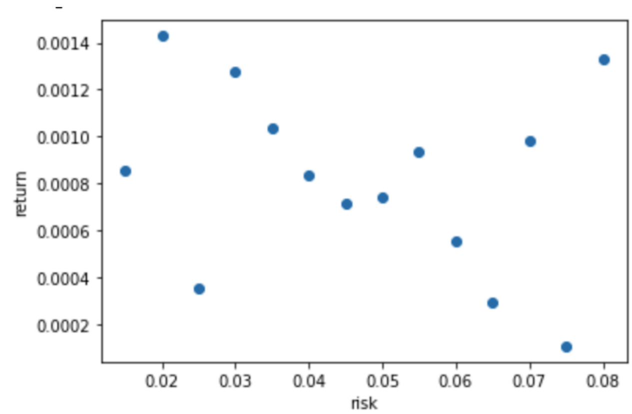
As mentioned above the goal of the "efficient frontier" is to find the correct weightage of a portfolio which satisfies an investors risk tolerance while maximizing the returns. According to the graph our goal is to find the red line. As mentioned above in the Efficient Frontier section our goal is to minimize the equation shown above.

7.1 Brute force

The first method we tried was a simple brute force approach. It consisted of the following steps:

1. iterate over multiple values of q (the risk) - this is the x-axis in our plot.
2. for each value of q generate 1,000,000 different weights randomly.
3. keep track of the minimum value of the equation shown above
4. return the risk, expected return, and the weights.

If we plot the output we get the following graph:

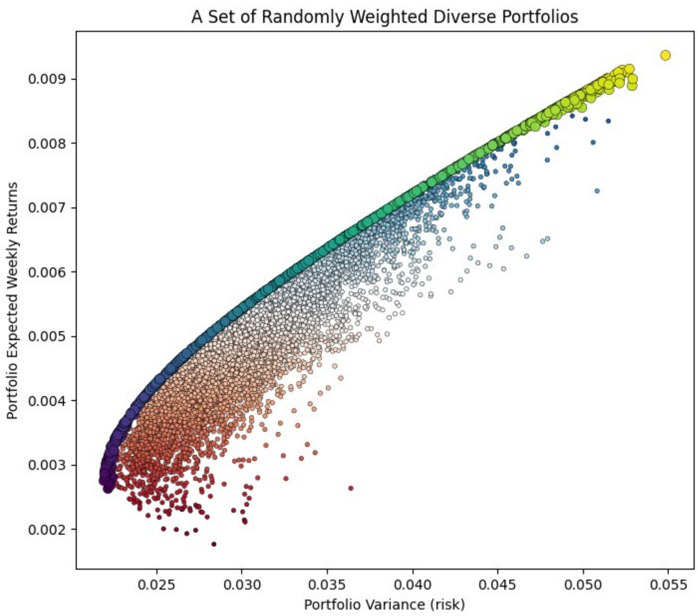


Though this graph does not look like the efficient frontier, it still shows the important aspects needed to find it. The graph above shows the frontier parameterized on q. Furthermore, "the point on the frontier at which the inverse of the slope of the frontier would be q if portfolio return variance instead of standard deviation were plotted horizontally" [3]. We can see from the graph above that there is a linear relationship forming which would support the hypothesis of the inverse slope of the frontier equation. Additionally, one would assume that if were were to making the number of trials of guessing random weights higher (i.e 1 billion etc...) it would continue getting more accurate.

Furthermore we have added the output of the code which provides the ideal weightages for a given q where $q \in (0.01, 0.08)$:

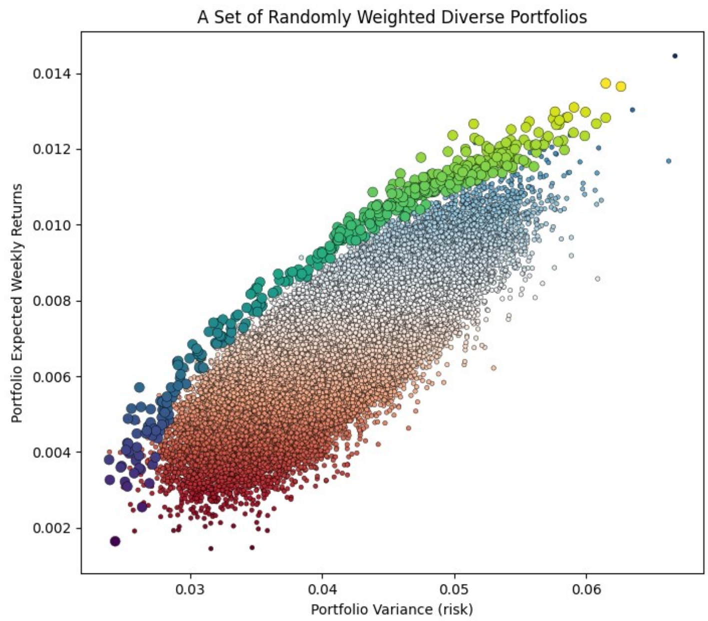
```
q: 0.015 weights: [0.00140536 0.59833109 0.39894887 0.00131468]
q: 0.02 weights: [0.00306813 0.58608827 0.40942191 0.00142168]
....
q: 0.05499999999999999 weights: [0.00170241 0.56464497 0.43189987 0.00175276]
q: 0.05999999999999999 weights: [4.61788868e-03 5.73128916e-01 4.22134461e-01 1.18734604e-04]
q: 0.06499999999999999 weights: [0.00145667 0.5469789  0.44974007 0.00182436]
q: 0.06999999999999999 weights: [0.00347356 0.54261839 0.45202804 0.00188   ]
q: 0.075 weights: [0.00095143 0.55202789 0.44625867 0.00076201]
```

This plot shows the "Brute Force" method of calculating the efficient frontier for a 5 stock portfolio as q sweeps from 0 to 2.



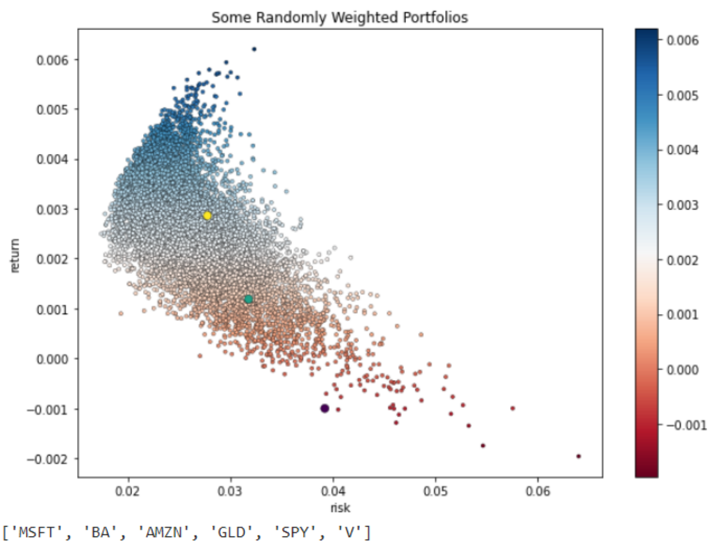
[3] https://en.wikipedia.org/wiki/Modern_portfolio_theory

This plot shows the "Brute Force" method of calculating the efficient frontier for a very diverse 27 stock portfolio as q sweeps from 0 to 1, obviously this efficient frontier is less clearly defined, as the number of iterations per q value needs to be kept around the order of 10^4 to keep runtime below a few minutes.



7.2 Gradient Descent

This plot shows colored points representing 3 guesses made by our gradient descent algorithm with q values very close together. If our gradient descent was working properly, these points would bunch up together somewhere on the leftmost side of the cloud.



['MSFT', 'BA', 'AMZN', 'GLD', 'SPY', 'V']

N.B: Additionally, it is said that this problem is very easily solved using Lagrange multipliers. [4].

8 Back testing our Strategy

Backtesting is a critical part in quantitative finance. Since there is a lot of data available about past historical data you should. test your model to see how it will perform. The strategy we might employ to backtest our model included picking a start date for when we would want to start collecting data

[4]https://en.wikipedia.org/wiki/Modern_portfolio_theory

about our stocks and an end date (01/01/2020-present). We would then run our optimisation algorithm for our stocks and our dates. Using this we got the ideal portfolio weights during this time period. We would then use those weights and test them for a set period of time (maybe for a week). This would tell us whether or not our model would actually work on market data. One would obviously not want to invest in individual stocks if they could get a hire return using a 10 year treasure bond which has 0% associated risk.

9 Conclusion

Overall we were able to successfully gather data using yfinance and store it an efficient manner using matrices. We were able to successfully show the important of an "efficient frontier" by modeling the risk and return associated with different weightages of stocks. We were also able to somewhat optimize a portfolio using our gradient descent algorithm, but ultimately our portfolio was not fully optimized. As described above, if our portfolio was fully optimized, we would expect each of our 'optimally' weighed portfolios to fall on the edge of the "efficient frontier". Although one of the points above does seem to fall on it, the rest of the points seem more clustered towards the middle, away from the edge.

Possible explanations for this are that our algorithm did not run with enough steps (because it was becoming very time-intensive), and perhaps with more steps the portfolios could have been further optimized and all fallen within the efficient frontier. Another distinct possibility as to why we could not fully optimize the portfolio was perhaps an error with our derivative function. Since the equation we were trying to minimize was composed of multiple vectors and matrices, it was much more difficult to figure out how to compute an accurate derivative for the equation, compared to what we did in class with homework 6. Although we explained the workings of our derivative function above, it remains a possibility that the computed derivative was wrong, and therefore threw the rest of the gradient descent algorithm off.

Nevertheless, our foray into Modern Portfolio Theory was fruitful, as we learned a lot about optimizing a portfolio, and were able to utilize the concepts of linear algebra, statistics, and optimization in order to implement our own gradient descent algorithm to optimize a portfolio.

References

[1] efficient_frontier/efficient _frontier_implementation.ipynb at master · tthustla/efficient_frontier. https://github.com/tthustla/efficient_frontier/blob/master/Efficient%20_Frontier_implementation.ipynb. (Accessed on 12/13/2021).

[2] How to construct an efficient portfolio using the modern portfolio theory in python? — by saad ahmed — towards data science. https://towardsdatascience.com/how-to-construct-an-efficient-portfolio-using-the-modern-portfolio-theory-in-python-5c5ba2b0cff4 (Accessed on 12/13/2021).

[3] Modern portfolio theory - wikipedia. https://en.wikipedia.org/wiki/Modern_portfolio_theory. (Accessed on 12/13/2021).

[4] Modern portfolio theory in python: Efficient frontier and minimum-variance portfolio - youtube. https://www.youtube.com/watch?v=mJTrQfzr0R4. (Accessed on 12/13/2021).

[5] Modern portfolio theory (mpt) definition. https://www.investopedia.com/terms/m/modernportfoliotheory.asp. (Accessed on 12/13/2021).