Modern Portfolio Theory Applied

Matthew B. Mykytiak

Michael T. Soukup

Northwestern University School of Professional Studies

Masters of Science in Data Science 460, Prof: Dr. Joe Wilck

Abstract

Harry Markowitz published his seminal work on Modern Portfolio Theory in 1952 laying the theoretical groundwork for constructing efficient portfolios. Efficient portfolios are those that leverage diversification to minimize portfolio variance while maximizing investment returns. Building upon Markowitz's initial work, we sought to validate Markowitz's theories with empirical findings leading to heuristics for how to adequately diversify a portfolio taking portfolio size, industry breadth, individual security limits, and investing strategy into consideration. To carry out our study we collected data on 416 stocks traded on US markets using the AlphaVantage REST API from 1/1/2006 -12/30/2020. Once portfolios were constructed varying our four factors under study investment resources were allocated using Linear Programming techniques, expected returns calculated from the Capital Asset Pricing Model (CAPM), and our portfolio constraints. Our experiment tracked the annualized rates of return of 1,049 portfolios over a 5-year period. Portfolio size was found to be the most important factor influencing portfolio returns and variance. The optimal portfolio size was found to be 44 stocks. Individual security limits, industry sector limits, and investing strategy trailed portfolio size in relative importance. 92.5% of our portfolios outperformed the Dow Jones Industrial Average (DJIA) and S&P 500 suggesting our resource allocation strategy is another factor contributing significantly to long-term market performance.

Keywords

Diversification, Portfolio, Risk, Linear Programming, Markowitz

Introduction

Financial markets are ingrained in our society and impact individual livelihoods. At the extreme end of the spectrum, thousands spend their entire career trying to make investments that outperform average financial market returns. On the other end of the spectrum, millions rely on market prosperity and investment returns from their retirement accounts to provide the resources necessary to financially support their friends, family, and themselves. For the masses, these retirement accounts are portfolios of stock securities, and the long-term performance of these portfolios greatly impact the quality of life for millions of Americans.

Because of the far-reaching impact portfolio performance has on nearly every American, we sought to determine the optimal way to diversify a portfolio and allocate investment resources in order to maximize returns over time.

Building off Harry Markowitz's Nobel Prize winning work (Markowitz, 1952), the primary objective of our study is to determine the optimal level of portfolio diversification needed in order to maximize returns in the stock market. Through diversification an investor reduces the variance of the portfolio and ensures less risk is absorbed and therefore returns are more predictable. An under-diversified portfolio is one which is highly volatile that can soar during times of market prosperity but also plummet during times of market or sector turmoil. Expected returns are subject to excessive risk and uncertainty, which is undesirable for an investor. On the other hand, an over-diversified portfolio is one which has asset allocation that is too broad and diluted. This type of investment helps protect an investor from down-side losses, but also does not allow the investor to gain alpha or beat the market. Over-diversified portfolios revert investors back to the mean not allowing for significant wealth accumulation. Meanwhile, a properly diversified portfolio is one which exposes an investor to enough of the market to minimize losses while enabling one to capture significant gains as select companies or sectors boom. Since all the portfolios we studied herein utilize some degree of diversification, we want to focus primarily on maximizing market returns and treat controlling portfolio variance as a secondary objective.

Another objective of our work was to determine if the impacts of diversification were arbitrary or if a strategic investment approach unlocked optimal returns. For instance, there are several attributes associated with individual securities that provide a numerical indication of their suspected market performance. Beta and expected market return are examples of these numerical indicators. We sought to determine if selecting portfolios of stocks based on these numerical indicators would help to strategically build a portfolio and tailor returns to an investor's goals.

Our hypothesis was that a strategically diversified portfolio constructed of stocks that have high expected rates of return will outperform the market and other portfolios over time. This portfolio will contain 33 stocks and have broad limits on asset allocation for both individual stocks and sector investments. We believed this portfolio would enable maximum returns as it would contain the appropriate number of stocks to achieve optimal volume diversification. By having broad limits on asset allocation, we could leverage expected rates of return and allocate funds to more optimistic stocks or sectors by our Linear Programming approach to achieve maximum results.

Literature Review

In 1952, Harry Markowitz published his paper titled *Portfolio Selection* in The Journal of Finance where Markowitz went on to introduce the world to his Modern Portfolio Theory (Markowitz, 1952). In his work, Markowitz refuted the claim that investors should aim to maximize expected returns when selecting investments but should rather seek to optimize portfolio selection by maximizing expected returns while minimizing portfolio variance. In doing so, investors will be seeking efficient portfolios that achieve the highest possible return given the amount of variance the portfolio contains. The need to take portfolio variance into account is because markets have imperfections and if the maximum return were solely sought after, a diversified portfolio would never be preferable to a non-diversified portfolio. Markowitz bluntly states that diversification is both observed and sensible; a rule of behavior which does not imply the superiority of diversification must be rejected. Thus, Markowitz dubbed the expected returns – variance of returns (E-V) rule in order to outline how one can go about selecting efficient portfolios and move an individual from speculation to investing (Markowitz, 1952).

Not only does Markowitz's E-V hypothesis imply diversification, it also implies the right kind of diversification and for the right reasons. The adequacy of diversification does not depend solely on the number of different securities held but also in the breadth of industry sectors that the portfolio holds. The reason this is important is that it is more likely for firms within the same industry to do poorly at the same time than for firms in dissimilar industries, which increases the covariance between the two firms ultimately increasing the portfolio's volatility. A good portfolio is a balanced whole, providing investors with protections and opportunities with respect to a wide range of contingencies (Markowitz, 1952).

While *Portfolio Selection* provides a solid mathematical and theoretical framework for selecting portfolios, it fails to demonstrate the effectiveness of the E-V rule with actual stock portfolios. We seek to do that in our work by demonstrating the benefit diversification has on portfolio performance over time. Markowitz's paper also failed to derive results analytically for the n-security case. Instead of deriving analytical results for the n-security case, we sought to empirically determine the optimal number of securities which harnesses the impact of diversification. Further our work sought to validate Markowitz's theories by determining heuristics for how to adequately diversify a portfolio taking portfolio size, industry breadth, and individual security limits into consideration.

Markowitz also failed to describe how expected security returns could be determined and assumed they were known. To make up for this omission our study leveraged years of market data to generate numerical indicators for each security allowing us to utilize the Capital Asset Pricing Model (CAPM) developed by William F. Sharpe to estimate expected security returns (Sharpe, 1964). By creating hundreds of portfolios from historical data and analyzing their performance over time, our work provides a descriptive framework for investors when it comes to selecting efficient portfolios moving Markowitz's work from theory to application.

Another shortcoming of Markowitz's work was a robust approach for determining the allocation of portfolio investments. Markowitz did propose the Critical Line Algorithm (CLA) in attempt to answer this question, however the CLA had considerable application flaws. For instance, CLA optimization yields significantly different portfolio allocations with small deviations in forecasted security returns. CLA produces sensitive results primarily due to inversions of large covariance matrices which are computationally demanding. Other researchers have proposed alternative algorithms to address the question of portfolio allocation such as the Hierarchical Risk Parity algorithm (HRP) (Vyas, n.d.). HRP was designed to remedy the numerical difficulties encountered when inverting sizeable covariance matrices. HRP has three distinct phases: hierarchical tree clustering, matrix seriation, and recursive bisection. Yet HRP is not without its own pitfalls such as naively bisecting all assets into two equal sized groups threatening to separate highly correlated assets or poor performance with highly correlated groups of assets (Nicolas Fermin Cota, 2019). In our work, we took a different approach to determining suitable asset allocation by leveraging Linear Programming (LP) techniques. The use of Linear Programming to determine asset allocation is not novel but we believe an effective and intuitive way to accomplish the same goal.

Methodology

The overall objective of this study was to determine the best strategy for allocating investment resources to maximize returns from the stock market for long term holdings. To do this we constructed thousands of stock portfolios with varying investment strategies and levels of diversification. Linear Programming (LP) techniques were utilized to determine the optimal investment allocation into each security contained within the respective portfolio. This required implementing various stock and sector constraints while utilizing the Capital Asset Pricing Model (CAPM) to determine expected market returns for each stock which served as our objective function coefficients. The objective of our LP model was to maximize annualized returns. The portfolios and their investment allocations were evaluated by determining their annualized returns over a five-year period and compared to the Dow Jones Industrial Average (DJIA) and S&P 500 as controls. Our model assumes no short selling of securities, dividends were reinvested, the portfolios holdings are not adjusted over the life of the holding period and stock, sector, size, and budgetary constraints must be upheld.

Computational Experiment

In order to execute our experimental design, 15 years' worth of stock market data was compiled via the AlphaVantage REST API (https://www.alphavantage.co/query). This API allowed us to pull historic daily stock market data in US markets. The data extracted from AlphaVantage was sub-setted from 1/1/2006 to 12/31/2015 and the daily stock returns, return deviations, security correlation coefficients, beta, and expected rates of return using CAPM (Equation 1) were determined ("Capital Asset Pricing Model (CAPM)", 2021). For this analysis 416 of the S&P 500 stocks were analyzed over the given time period. 84 stocks were dropped from analysis due to limited data history. Additionally, each stock was labelled according to the industrial sector it belongs to.

Equation 1.

$$ER_i = R_f + \beta_i(ER_m - R_f)$$

Once this data was compiled, we were able to build portfolios according to various metrics and strategies. To evaluate the impact of various levels of volume diversification, seven different sizes of portfolios were considered in this study. The seven sizes considered were portfolios made up of 11, 22, 33, 44, 55, 66, and 77 stocks.

Four different investing strategies were also considered. The first strategy was to build a portfolio which capitalized on long term market prosperity. This portfolio consisted of the stocks from each of the eleven market sectors with the highest beta values. The second strategy considered building a risk-adverse portfolio. This portfolio contained the stocks from each sector with the lowest beta values. The third strategy considered was to build a portfolio of stocks with the best performance outlook as revealed by the expected rate of return calculated by CAPM. The final investment strategy considered was a random investment strategy where portfolios were created of N random stocks. Due to the random nature in selecting this final investing strategy, five different random portfolios were constructed for a total of eight portfolios all together.

Once the securities for the respective portfolios were identified, Linear Programming techniques were used to allocate investment resources. In setting up this Linear Programming problem, the number of decision variables was dictated by the size of the portfolio (Equation 2.). Each decision variable indicated the amount of money invested in each stock. We assumed a budget of $1,000,000. The expected market return for each stock was determined by CAPM. These expected rates of returns served as the objective function coefficients for our Linear Programming model (Equation 3.). Our LP problem sought to maximize our investment returns while adhering to the portfolio constraints.

Equation 2.
$$X_i = Money[USD]\ invested\ in\ each\ stock$$

Equation 3.
$$MAX : \sum_{i=1}^{N} ER_i X_i$$

For the Linear Programming problem, we considered two levels of constraints: sector limits and stock limits. These constraints enforced our desire to have diversified portfolios while allowing the opportunity to learn how to best allocate investment resources to maximize returns. Six levels of sector limits were considered: 0.04 - 0.16, 0.05- 0.15, 0.06- 0.14, 0.07-0.13, 0.08 - 0.12, and 0.09-0.10 (Equation 4 & 5). Additionally, six levels of stock limits were considered: 0.01-0.12, 0.015-0.105, 0.02-0.09, 0.025-0.075, 0.03-0.06, and 0.035-0.045 (Equation 6 & 7). The stock and sector constraints dictate what percentage of the overall portfolio can be made up of a single sector or a single stock. For instance, the stock constraints of 0.01 - 0.12 indicate a single stock must make up at least 1% of the total portfolio resources but cannot account for more than 12% of the total portfolio resources. Similar logic can be applied for the sector constraint limits. Budgetary constraints (Equation 8.) and non-negativity constraints (Equation 9.) were also enforced in our LP model. In total, 2,016 portfolios were generated for this study. However, only 1,049 were evaluated as 967 were dropped from the study due to stock, sector, or size constraint infringements.

Equation 4.
$$\sum_{j=1}^{N=11} X_j \geq Sector\ Lower\ Limit * Budget$$

Equation 5.
$$\sum_{j=1}^{N=11} X_j \leq Sector\,Upper\,Limit * Budget$$

Equation 6.
$$X_i \geq Stock\,Lower\,Limit * Budget$$

Equation 7.
$$X_i \leq Stock\,Upper\,Limit * Budget$$

Equation 8.
$$\sum_{i=1}^{N} X_i = Budget$$

Equation 9.
$$X_i \geq 0, for\,all\,i$$

The portfolios were evaluated by determining their performance from 1/1/2016 - 12/31/2020. Portfolio performance was determined by annualized returns over the five-year period. After all our portfolios were built, allocated, and assessed, the portfolios were aggregated by the varying experimental variables we considered in this study. The Dow Jones Industrial Average (DJIA) and the S&P 500 were used as controls in our experiment to represent overall market performance.

Results and Discussion

In the first part of analysis, we will review the impacts each factor had on portfolio returns in isolation. Once it is understood how each factor impacts portfolio performance on its own, we will consider the combinatorial effect the factors have when applied together. Then we will compare the performance of our portfolios to our controls seeking understanding for any differences. We will close our discussion talking how our work can be advanced for future study and investigation.

Considering the impact of the size of the portfolio on the annualized rates of returns, the data initially suggests that the preferred portfolio size to select in order to maximize investment returns is either 22 or 11. This is because these portfolios yielded the highest maximum annualized rate of return and highest interquartile range, respectively. These results can be seen in Figure 1. We do not suggest this investment strategy because these portfolio sizes experienced greater variance in results as indicated by high standard deviations. Therefore, the efficiency of these portfolios is less compared to alternative sizes.

The results for portfolios containing 11 and 22 stocks are not validated by any meaningful trends as shown in Figure A5. Rather our results indicated that there was no clear trend regarding the average or median expected return with size of the portfolio. For all seven portfolio sizes considered, the mean and median annualized returns randomly fluctuated between 15.7% - 21.9%. The trend and impact of volume diversification can be seen by the reduction in standard deviation for the annualized returns as portfolio size increased. This finding confirms that as the portfolio size increases, the portfolio variance decreases supporting Markowitz's ideas.

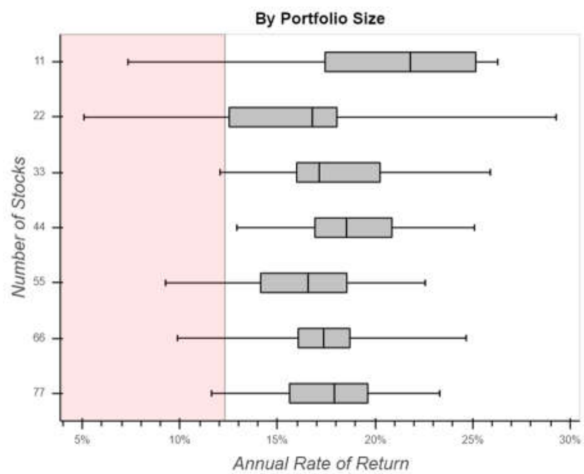

Figure 1. Boxplot of annualized rates of returns for the portfolios under study aggregated by the number of stocks in the portfolio. Red shaded area represents the annualized return of the Dow Jones Industrial Average at 12.28% for reference.

Another important consideration when investing is minimizing losses. In our figure, this can be interpreted as achieving the highest minimum rate of return. There is a positive correlation between portfolio size and minimum rate of return displaying that diversification will help ensure losses are avoided.

Taking these facts together, we believe that that ideal portfolio size contains 44 stocks. The reason being because this portfolio size has led to the highest interquartile range second only to N = 11. Portfolios made up of 44 stocks lead to the highest minimum annualized rate of return and a standard deviation 36% lower compared to portfolios made up of 11 stocks. This portfolio size ensures an investor will minimize their downside while seeking a highly efficient portfolio seeking high returns.

We also considered varying the stock allocation constraints dictating how much of our portfolio can be made up of a single stock and the results are shown in Figure 2. Our data indicated a positive correlation associated with the narrowing of these constraint limits and the mean and median annualized return for the portfolio as seen in Figure A6. This suggests that narrow asset allocation ranges should be adopted when implementing the investment strategy for a portfolio. This finding supports the pursuit of diversification on the stock level.

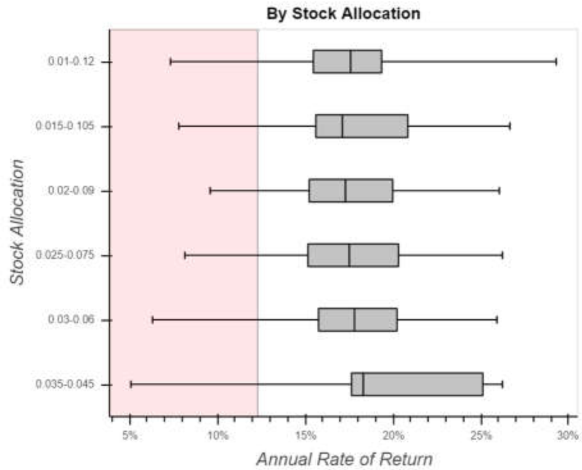

Figure 2. Boxplot of annualized rates of returns for the portfolios under study aggregated by their portfolio stock allocation limits. Red shaded area represents the annualized return of the Dow Jones Industrial Average at 12.28% for reference.

Our data indicated a strong linear correlation between the width of sector allocation and annualized returns as shown in Figure A7. This finding was inverse to our findings regarding width of stock constraints. As the sector constraints widened, the mean, median, min, and max annualized returns all trended upwards. These strong linear correlations indicate sector diversification should not be equitable. Rather, some sectors should be heavily weighted in the portfolio compared to others. On the basis of asset allocation, stocks should be weighted equitably while sectors should allow the freedom to disproportionately weight industries with high expected returns. This further emphasizes the importance of diversification indicating it is beneficial bet on industries just not individual stocks as industry performance is less volatile compared to individual securities.

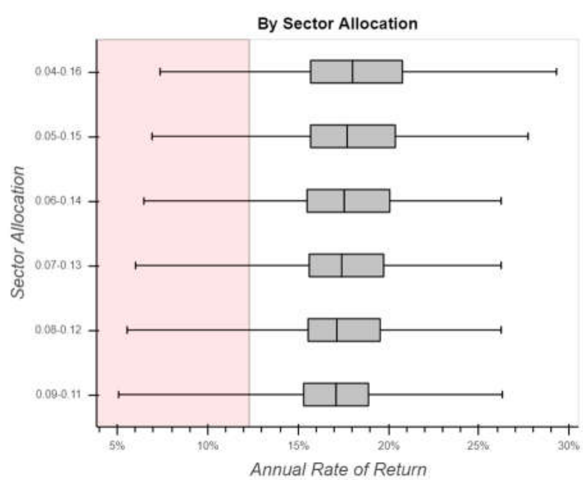

Figure 3. Boxplot of annualized rates of returns for the portfolios under study aggregated by their portfolio sector allocation limits. Red shaded area represents the annualized return of the Dow Jones Industrial Average at 12.28% for reference.

Lastly, we considered the impact of a strategic portfolio creation shown in Figure 4. The portfolio generated by seeking the most risk (highest betas) resulted in the highest expected rates or return. The portfolio generated by seeking to minimize risk (lowest beta) resulted in the lowest downside. The portfolio seeking to maximize returns (highest CAPM) minimized portfolio volatility. The results here are counter-intuitive and indicate that selecting stocks based on CAPM values better dictates market volatility compared to beta, a known measure of market volatility. We recommend re-conducting this research in the future in order to verify this finding. If confirmed, it will be important to determine the mechanism for why it is true.

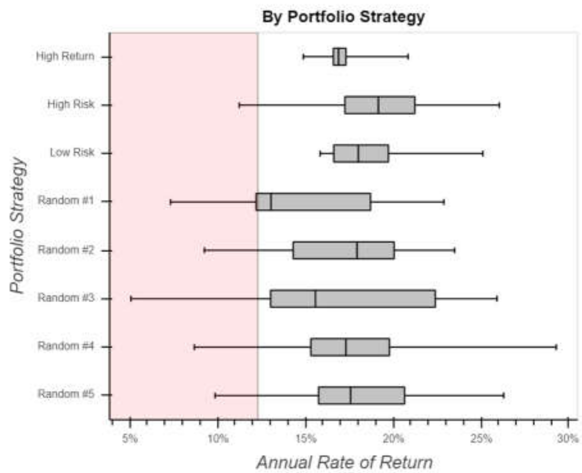

Figure 4. Boxplot of annualized rates of returns for the portfolios under study aggregated by their portfolio strategy. Red shaded area represents the annualized return of the Dow Jones Industrial Average at 12.28% for reference.

Nonetheless, both the High and Low Risk portfolios outperformed the High Return portfolio refuting our hypothesis. The takeaway here is that over time, beta is a superior numerical indicator compared to CAPM for projecting market returns. Our experiment revealed that strategic portfolio creation outperformed randomized portfolios. Regardless of the strategy, our data showed that a strategic approach to portfolio creation led to more predictable expected returns indicated by lower standard deviations and smaller interquartile ranges relative to randomly created portfolios.

In order to understand the combinatorial effects of the four factors on portfolio results, we grouped our data by portfolio size, width of limits on asset allocation for individual stocks, width of limits on asset allocation for market sectors, and portfolio strategy. Then we determined which combination of factors both outperformed our controls and expectations. Of the 1,049 portfolios considered only 112, or approximately 10%, fit this description.

Of the portfolios that beat the DJIA, S&P 500, and outperformed expectations, only two portfolio sizes were represented: 33 and 44. When N = 33, the sector ranges were limited to the three most narrow constraints and the stock ranges had to be 0.025-0.075. On the other hand, when N = 44 there was considerable flexibility regarding stock and sector constraints. For N = 44 the stock constraints represented the wider end of the available options while the sector constraints varied the whole gamut. For all combinations of portfolio factors that matched our filter criteria, all investment strategies (except for High Return) were represented indicating strategy had a minimal relative impact on portfolio performance. In total, 91 of the 143 N = 44 portfolios exceeded expectations while only 21 of the 239 N = 33 portfolios achieved the same feat. These results are summarized in Table 1.

Number of Stocks in Portfolio	Stock Allocation Range	Sector Allocation Range	Annual Rate of Return
33	0.025-0.075	0.07-0.13	0.1810
33	0.025-0.075	0.08-0.12	0.1784
33	0.025-0.075	0.09-0.11	0.1795
44	0.01-0.12	0.07-0.13	0.1957
44	0.01-0.12	0.08-0.12	0.1927
44	0.015-0.105	0.04-0.16	0.1981
44	0.015-0.105	0.05-0.15	0.1954
44	0.015-0.105	0.06-0.14	0.1916
44	0.015-0.105	0.07-0.13	0.1906
44	0.015-0.105	0.08-0.12	0.1880
44	0.015-0.105	0.09-0.11	0.1829
44	0.02-0.09	0.04-0.16	0.1941
44	0.02-0.09	0.05-0.15	0.1903
44	0.02-0.09	0.06-0.14	0.1872
44	0.02-0.09	0.07-0.13	0.1847
44	0.02-0.09	0.08-0.12	0.1869

Table 1. Table summarizing the combinatorial effects for portfolios that performed better than the Dow Jones Industrial Average (DJIA), the S&P 500, and expectations. Displayed are the annualized rates of returns for these portfolios and their factor levels.

The combinatorial impacts of the factors conclude that the optimal portfolio size is 44 stocks. The N = 44 portfolio size allows for the greatest flexibility in stock and sector allocation limits. While this conclusion is clear from our study, we suggest considering a lower stock size resolution in future studies to confirm this finding. Nonetheless, a portfolio with 44 individual stocks will give investors the best chance to maximize their returns. This statement is solidified as all portfolios with 44 stocks that beat the controls and expected returns had higher annual rates of return compared to their 33 stock counter parts.

One interesting outcome of this combinatorial approach was that it appears to refute our individual factor findings. For instance, the N = 33 stock portfolios that passed our selection criteria leveraged sector allocations on the narrow end of the spectrum. Similarly, the highest performing N = 44 portfolios utilized stock allocation ranges on the wide end of the spectrum. This indicates that stock and sector weighting allocations interact, inverting the impacts of these factors in isolation. Our high performing portfolios exclusively had stock constraints that favored the wider limits when analyzed in combination to the sector limits which is opposite of what we found when studying individual factor impacts.

The combinatorial analysis demonstrates the relative importance of each factor based on the flexibility each factor was granted given our high performing portfolios. For instance, the more levels the factor could take on, the less important that factor is when constructing high performing portfolios. This takeaway illuminates the relative importance of each factor when building a portfolio: Size > Stock constraints > sector constraints > strategy.

We found that our portfolios routinely and significantly outperformed the controls. Only 7.5%, or 79 out of 1,049 portfolios failed to outperform either the DJIA or S&P 500. The DJIA experienced an annual rate of return of 12.28% from 2016 – 2020 while the S&P 500 had an annualized rate of return of 13.29% over the same time period. Our portfolios averaged an annualized rate of return of 17.73% from 2016-2020 outperforming the S&P 500 and DJIA by 33.3% and 44.3%, respectively. It is important to consider what led to these differences as the DJIA and S&P 500 are not just controls for our study, but common benchmarks for market performance.

The DJIA is an index consisting of 30 stocks aiming to track the health of the American Economy ("Dow Jones Industrial Average (DJIA) Definition", 2021). The DJIA lacks sector diversification by only representing 8 of the 11 industrial sectors with its holdings ("Markets - Dow 30", 2021). Meanwhile, the S&P 500 is an index made up of 505 stocks across all eleven industry sectors ("List of S&P 500 companies - Wikipedia", 2021). The sector weighting is not truly equitable, but each sector is represented to a considerable degree by count of stock listings within the index. While neither the DJIA or S&P 500 is optimized based on our findings for portfolio size nor sector weighting, it appears another factor is at play as neither of the factors we have considered thus far seem likely to contribute to the high-performance gains captured by our constructed portfolios.

The DJIA is a price-weighted index ("Dow Jones Industrial Average (DJIA) Definition", 2021). This means that the amount of each stock held in the index is based on the relative dollar value for each share of stock. The S&P 500 is a capitalization weighted index meaning companies with higher market capitalization will occupy higher relative holdings within the index compared to low market capitalization companies ("List of S&P 500 companies - Wikipedia", 2021). The portfolios we generated took an alternative approach and allocated resources using Linear Programming. Our objective function utilized CAPM expected rates of return as coefficients seeking to maximize our portfolio gain. We believe this resource allocation strategy is what explains the significant performance increase of our portfolios compared to the DJIA and S&P 500. This is a contentious point and should be a subject of further study.

Future work should explore the impact of resource allocation methodology on portfolio performance. This future work should control the stocks that make up each portfolio to isolate the impact resource allocation has on portfolio performance. Our findings suggest that a Linear Programming approach to asset allocation will lead to superior performance gains as compared to alternative options such as price-weighted portfolios or market capitalization weighted portfolios. However, our methodology only considered one approach for Linear Programming. We suggest considering alternative objective function coefficients in place of the expected returns calculated by the CAPM. Examples of alternative objective function coefficients are beta, price to earnings ratios, or expected returns calculated from Arbitrage Pricing Theory (APT) or the Conditional Capital Asset Pricing Model (CCAPM) ("CAPM vs. Arbitrage Pricing Theory: What's the Difference?", 2021). Once the best asset allocation strategy is discovered and validated, a logical next step will be to determine the optimal asset re-allocation frequency utilizing the tactic for the life of the portfolio.

Our ultimate goal is to maximize investment returns for the life of the portfolio. While the work conducted within is a good start at highlighting important diversification factors to consider when building a portfolio, it only analyzed portfolio performance over a 5-year period. Considering most retirement accounts accrue over 30-40 years, it will be necessary to validate our findings over longer time periods.

Conclusion

Our findings confirm the importance of diversification both when attempting to maximize expected rates of return and minimize portfolio volatility. Our work provides empirical evidence to support Markowitz's Nobel Prize winning theories first published in 1952. We found that a portfolio built with 44 individual stocks will give an investor the best chance to maximize their returns while simultaneously protecting their downside. While in isolation it was found that individual stock constraints should be minimized while sector constraints maximized, a combinatorial analysis of the factors under study indicated these factors interact inverting those findings. The combinatorial analysis revealed the relative importance of the factors we considered were portfolio size, stock constraints, sector constraints, and strategy from most to least important. When compared to our controls, the portfolios we constructed significantly outperformed those benchmarks. This suggested that the asset allocation method may also have a considerable impact in determining portfolio performance and should be a subject to undertake in future studies.

References

Capital Asset Pricing Model (CAPM). (2021). Retrieved 25 May 2021, from https://www.investopedia.com/terms/c/capm.asp

CAPM vs. Arbitrage Pricing Theory: What's the Difference?. (2021). Retrieved 26 May 2021, from https://www.investopedia.com/articles/markets/080916/capm-vs-arbitrage-pricing-theory-how-they-differ.asp#:~:text=The%20arbitrage%20pricing%20theory%20is,users%20to%20quantify%20multiple%20factors.

Dow Jones Industrial Average (DJIA) Definition. (2021). Retrieved 25 May 2021, from https://www.investopedia.com/terms/d/djia.asp

List of S&P 500 companies - Wikipedia. (2021). Retrieved 25 May 2021, from https://en.wikipedia.org/wiki/List_of_S%26P_500_companies

Markets - Dow 30. (2021). Retrieved 25 May 2021, from https://www.cnbc.com/dow-30/

Markowitz, H. (1952). Portfolio Selection. *The Journal Of Finance*, *7*(1), 77-91.

Markowitz, H. (1959). *Portfolio Selection: Efficient Diversification of Investments*. New York: John Wiley & Sons, Inc.

Nicolas Fermin Cota, R. (2019). Shortfalls of the Hierarchical Risk Parity. Retrieved 24 May 2021, from https://www.linkedin.com/pulse/shortfalls-hierarchical-risk-parity-rafael-nicolas-fermin-cota/

Sharpe, W. (1964). Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk. *The Journal Of Finance*, *14*(3), 425-442.

Vyas, A. The Hierarchical Risk Parity Algorithm: An Introduction - Hudson & Thames. Retrieved 24 May 2021, from https://hudsonthames.org/an-introduction-to-the-hierarchical-risk-parity-algorithm/#:~:text=Hierarchical%20Risk%20Parity%20uses%20single,and%20columns%20for%20stocks%20and%20.

Appendix A

Additional Tables and Figures

Number of Stocks	Count	Mean	Std	Min	25%	50%	75%	Max
11	164	0.2095	0.0436	0.0733	0.1744	0.2181	0.2517	0.2630
22	263	0.1568	0.0396	0.0507	0.1252	0.1678	0.1804	0.2929
33	239	0.1781	0.0317	0.1205	0.1598	0.1714	0.2026	0.2592
44	143	0.1884	0.0279	0.1292	0.1692	0.1854	0.2087	0.2510
55	96	0.1635	0.0319	0.0925	0.1413	0.1656	0.1856	0.2258
66	96	0.1737	0.0310	0.0986	0.1607	0.1736	0.1871	0.2468
77	48	0.1769	0.0270	0.1161	0.1562	0.1791	0.1963	0.2332

Table A1. Table of annualized rates of returns for the portfolios under study aggregated by the number of stocks in the portfolio.

Stock Allocation	Count	Mean	Std	Min	25%	50%	75%	Max
0.01-0.12	318	0.1746	0.0360	0.0733	0.1546	0.1758	0.1935	0.2929
0.015-0.105	270	0.1775	0.0356	0.0780	0.1561	0.1711	0.2084	0.2665
0.02-0.09	167	0.1765	0.0363	0.0959	0.1523	0.1729	0.1998	0.2605
0.025-0.075	120	0.1760	0.0393	0.0815	0.1516	0.1751	0.2031	0.2623
0.03-0.06	120	0.1783	0.0460	0.0631	0.1576	0.1780	0.2023	0.2592
0.035-0.045	54	0.1944	0.0594	0.0507	0.1764	0.1830	0.2511	0.2623

Table A2. Table of annualized rates of returns for the portfolios under study aggregated by their portfolio stock allocation limits.

Sector Allocation	Count	Mean	Std	Min	25%	50%	75%	Max
0.04-0.16	179	0.1810	0.0408	0.0737	0.1569	0.1801	0.2077	0.2929
0.05-0.15	177	0.1795	0.0394	0.0692	0.1569	0.1771	0.2038	0.2772
0.06-0.14	177	0.1778	0.0389	0.0647	0.1550	0.1755	0.2007	0.2623
0.07-0.13	176	0.1769	0.0381	0.0601	0.1562	0.1741	0.1973	0.2623
0.08-0.12	172	0.1752	0.0383	0.0555	0.1556	0.1714	0.1954	0.2623
0.09-0.11	168	0.1728	0.0397	0.0507	0.1531	0.1709	0.1889	0.2630

Table A3. Table of annualized rates of returns for the portfolios under study aggregated by their portfolio sector allocation limits.

Strategy	Count	Mean	Std	Min	25%	50%	75%	Max
High Return	120	0.1710	0.0110	0.1489	0.1660	0.1689	0.1732	0.2086
High Risk	150	0.1916	0.0337	0.1124	0.1726	0.1916	0.2124	0.2605
Low Risk	150	0.1875	0.0251	0.1585	0.1662	0.1801	0.1973	0.2511
Random #1	111	0.1507	0.0388	0.0733	0.1220	0.1305	0.1872	0.2289
Random #2	138	0.1719	0.0348	0.0925	0.1431	0.1794	0.2006	0.2350
Random #3	122	0.1675	0.0570	0.0507	0.1303	0.1558	0.2241	0.2592
Random #4	110	0.1804	0.0375	0.0868	0.1532	0.1731	0.1979	0.2929
Random #5	148	0.1881	0.0444	0.0986	0.1576	0.1758	0.2065	0.2630

Table A4. Table of annualized rates of returns for the portfolios under study aggregated by their portfolio strategy.

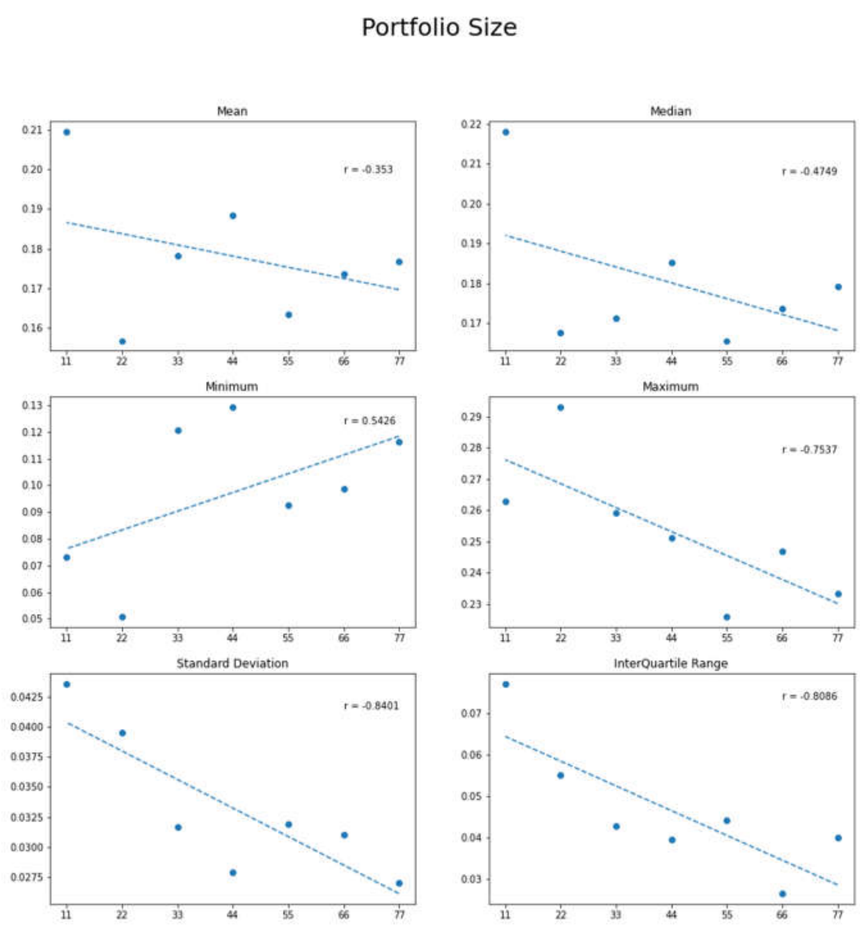

Figure A5. Correlation plots of portfolio stock size vs. the aggregated means, medians, minimums, maximums, standard deviations, and Interquartile Ranges.

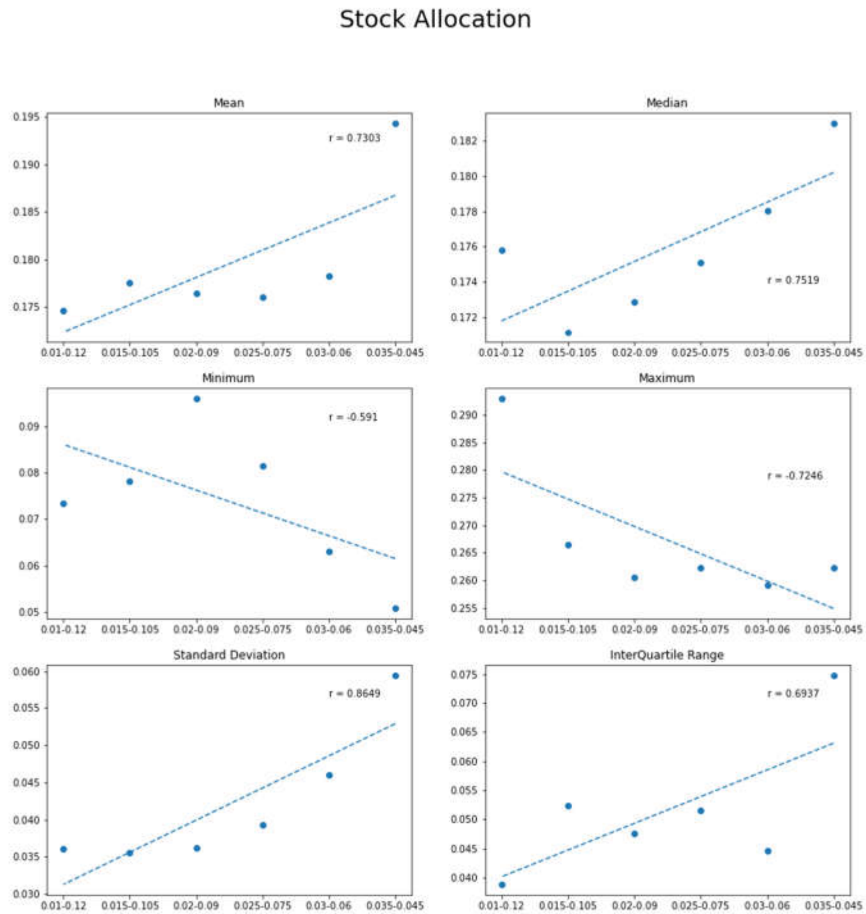

Figure A6. Correlation plots of portfolio stock allocation vs. the aggregated means, medians, minimums, maximums, standard deviations, and Interquartile Ranges.

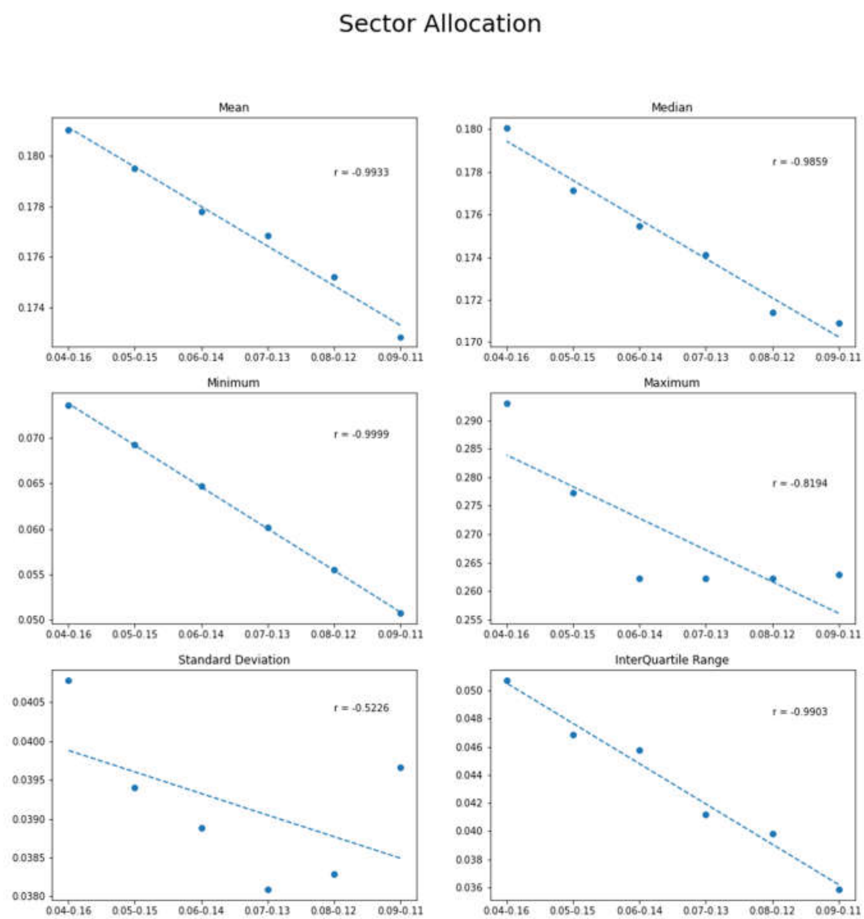

Figure A7. Correlation plots of portfolio sector allocation vs. the aggregated means, medians, minimums, maximums, standard deviations, and Interquartile Ranges.

Appendix B

Python code for accessing stock market data via AlphaVantage REST API, security analysis, asset allocation via Linear Programming, portfolio performance tracking and metrics tracking.

```
import requests
import pandas as pd
import json
import time
import numpy as np
import pulp
import warnings
import random
warnings.filterwarnings("ignore")

def get_data(ticker):
    url = 'https://www.alphavantage.co/query'
    params = {'function':'TIME_SERIES_DAILY_ADJUSTED'
              , 'outputsize':'full'
              , 'datatype':'json'
              , 'apikey':'KJBT6V8VSE91UVDZ'}
    params['symbol'] = ticker
    req = requests.get(url, params=params).json()
    df_ticker = pd.DataFrame()
    for k,v in req.items():
        if k == 'Time Series (Daily)':
            for k, v in v.items():
                df_row = pd.json_normalize(v)
                df_row.insert(0, 'ticker', ticker)
                df_row.insert(0, 'date', k)
                df_ticker = df_ticker.append(df_row)
    df_ticker.rename(columns={'1. open':'open'
                                   , '2. high':'high'
                                   , '3. low':'low'
                                   , '4. close':'close'
                                   , '5. adjusted close':'adjusted_close'
                                   , '6. volume':'volume'
                                   , '7. dividend amount':'dividend'
                                   , '8. split coefficient':'split'}, inplace= True)
    df_ticker = df_ticker.astype({'date':'datetime64'
                                  , 'open':'float'
                                  , 'high':'float'
                                  ,'low':'float'
                                  ,'close':'float'
                                  ,'adjusted_close':'float'
                                  , 'volume':'int64'
                                  , 'dividend':'float'
                                  ,'split':'float'})
    df_ticker['previous_adjusted_close'] = df_ticker['adjusted_close'].shift(periods=-1, fill_value = 0)
    df_ticker['daily_return'] = (df_ticker['adjusted_close']-df_ticker['previous_adjusted_close'])/df_tick
er['previous_adjusted_close']
    return df_ticker


def stock_returns(start_date, end_date, risk_free_rate, filename):
    # Read Historical Stock Data for individual stocks, the Dow Jones, and Sector Groupings
    df_all = pd.read_csv('./inputs/daily_stock_data.csv', usecols = ['date', 'ticker', 'daily_return', 'ad
justed_close'])
    df_all = df_all.astype({'date':'datetime64'})
    df_djia = pd.read_csv('./inputs/daily_index_djia.csv', usecols = ['date', 'daily_return'])
    df_djia = df_djia.astype({'date':'datetime64'})
    df_sp500 = pd.read_csv('./inputs/daily_index_sp500.csv', usecols = ['date', 'daily_return'])
    df_sp500 = df_djia.astype({'date':'datetime64'})

    df_sectors = pd.read_csv('./inputs/sectors_sp500.csv')
    df_merged = df_all.merge(df_djia, how = 'inner', on = 'date')
```

```
    df_merged.rename(columns = {'daily_return_x':'daily_return','daily_return_y':'dow_jones_daily_return'}
, inplace = True)
    df_merged = df_merged.merge(df_sp500, how = 'inner', on = 'date')
    df_merged.rename(columns = {'daily_return_x':'daily_return','daily_return_y':'sp500_daily_return'}, in
place = True)
    df_historical = df_merged[(df_merged['date']>='{0}'.format(start_date)) & (df_merged['date']<='{0}'.fo
rmat(end_date))]

    # Standard Deviation of Stock and Market for time period
    df_std = df_historical.groupby('ticker').agg({'daily_return':['std'], 'dow_jones_daily_return':['std']
})
    df_std.reset_index(inplace = True, drop = False)
    df_std.columns = df_std.columns.droplevel(1)

    # Correlation between Stock and Market for time period
    df_corr = df_historical.groupby('ticker')[['daily_return', 'dow_jones_daily_return']].corr()
    df_corr.reset_index(inplace = True)
    df_corr = df_corr[df_corr['daily_return'] == 1]
    df_corr.drop(columns = ['level_1', 'daily_return'], inplace = True)
    df_corr.rename(columns = {'dow_jones_daily_return':'correlation'}, inplace = True)

    # Calculation of Betas
    df_beta = df_std.merge(df_corr, how = 'inner', on = 'ticker')
    df_beta['beta'] = (df_beta['daily_return']/df_beta['dow_jones_daily_return']) * df_beta['correlation']
    df_beta.rename(columns = {'daily_return':'stock_stddev', 'dow_jones_daily_return':'market_stddev'}, in
place=True)
    df_beta = df_beta.merge(df_sectors, how = 'inner', left_on = 'ticker', right_on = 'Symbol')
    df_beta.drop(columns = ['Symbol'], inplace = True)
    df_beta.rename(columns = {'Name':'company_name', 'Sector':'company_sector'}, inplace = True)
    df_beta.dropna(axis=0, inplace = True)
    df_beta.reset_index(drop = True,  inplace = True)

    # Calculation of Annual Return
    df_return = df_historical[['date','ticker','adjusted_close']]
    df_return_dates =  df_return.groupby('ticker').agg({'date':['min', 'max']})
    df_return_dates.columns = df_return_dates.columns.droplevel(0)
    df_return_dates.reset_index(drop = False, inplace = True)
    df_return_dates = df_return_dates.melt(id_vars=['ticker'])
    df_return_dates.rename(columns={'value':'date'}, inplace = True)
    df_return = df_return.merge(df_return_dates, how='inner', on   = ['date', 'ticker'])
    df_return = df_return.pivot(index = 'ticker', columns = 'variable', values = ['adjusted_close', 'date'
])
    df_return['adjusted_close_chg'] = df_return[('adjusted_close', 'max')]/df_return[('adjusted_close', 'm
in')]
    df_return['date_chg'] = df_return[(              'date', 'max')] - df_return[(              'date', 'm
in')]
    df_return['date_chg'] = df_return['date_chg'].dt.days.astype('int16')/365
    df_return['aror'] = (df_return['adjusted_close_chg']**(1/df_return['date_chg']))-1
    df_return.reset_index(inplace = True, drop = False)
    df_return = df_return[['ticker', 'aror']]
    df_return.columns = df_return.columns.droplevel(1)

    # Calculate Expected Rate of Return
    df_final = df_beta.merge(df_return, how='inner', on='ticker')
    df_final = df_final[['ticker', 'company_name', 'company_sector', 'beta', 'aror']]
    df_final['eror'] = risk_free_rate + (df_final['beta']*(df_final['aror'] - risk_free_rate))
    df_final.to_csv('./portfolios/{0}'.format(filename), index = False)
    return df_final

def generate_portfolio(df, n, portfolio_type, sort_by = None, sort_by_order = None, group_by = None):
    portfolio_type = str(portfolio_type)
    if portfolio_type.startswith('Random') == True:
        df = df.iloc[random.sample(range(0, len(df)-1), 11 * n)]
        #df = df.iloc[set(np.random.randint(0, len(df)-1, size = 11 * n).tolist())]
        df.drop(columns = ['company_name', 'row_num'], inplace = True)
        df['portfolio_type'] = portfolio_type
```

```
        df = df[['portfolio_type', 'ticker', 'company_sector', 'beta', 'aror', 'eror']]
        df.reset_index(inplace = True, drop = True)
    else:
        if sort_by_order == 'Ascending':
            ascending = True
        else:
            ascending = False
        if type(sort_by) != list():
            sort_by = [sort_by]
        if type(group_by) != list():
            group_by = [group_by]
        df['row_num'] = df.sort_values(sort_by, ascending = ascending).groupby(group_by).cumcount() + 1
        df = df[df['row_num'] <= n]
        df['portfolio_type'] = portfolio_type
        df = df[['portfolio_type', 'ticker', 'company_sector', 'beta', 'aror', 'eror']]
        df.reset_index(inplace = True, drop = True)
    return df


def optimize_portfolio(df, budget, stock_lower_limit, stock_upper_limit, sector_lower_limit, sector_upper_
limit):
    names = df['ticker']
    # Create a LP Maximization problem
    Lp_prob = pulp.LpProblem('Problem', pulp.LpMaximize)
    # Create Variables
    list_of_vars = [pulp.LpVariable(name, stock_lower_limit * budget, stock_upper_limit * budget) for name
in names]
    # Objective Function
    Lp_prob += pulp.lpSum([df.eror[i]*list_of_vars[i] for i in range(len(df))])
    # Constraints:
    Lp_prob += pulp.lpSum(list_of_vars[i] for i in range(len(df))) == budget
    sectors = df['company_sector'].unique()
    for sector in sectors:
        comp_in_sector = df[df['company_sector'] == sector]
        Lp_prob += pulp.lpSum(list_of_vars[index] for index, row in comp_in_sector.iterrows()) >= budget *
sector_lower_limit
        Lp_prob += pulp.lpSum(list_of_vars[index] for index, row in comp_in_sector.iterrows()) <= budget *
sector_upper_limit
    #print(Lp_prob)
    status = Lp_prob.solve()
    # Generate Allocation Dataframe to join to original portfolio dataframe
    df_allocation = pd.DataFrame(columns = ['ticker', 'allocation'])
    for v in Lp_prob.variables():
        df_var_allocation = pd.DataFrame(data = {'ticker': [v.name], 'allocation': [v.varValue]})
        df_allocation = df_allocation.append(df_var_allocation)
    df_portfolio = df.merge(df_allocation, how = 'inner', on = 'ticker')
    df_portfolio['allocation_pct'] = df_portfolio['allocation']/budget
    df_portfolio['stock_ll'] = stock_lower_limit
    df_portfolio['stock_ul'] = stock_upper_limit
    df_portfolio['sector_ll'] = sector_lower_limit
    df_portfolio['sector_ul'] = sector_upper_limit
    return df_portfolio


def performance_portfolio(df):
    df_prices = pd.read_csv('./inputs/daily_stock_data.csv', usecols = ['ticker', 'date', 'adjusted_close'
])
    df_prices = df_prices[(df_prices['date']>'2015-12-31') & (df_prices['date']<='2020-12-31')]
    # Beginning Prices
    df_prices_dt_beg = df_prices.groupby('ticker').min()
    df_prices_dt_beg.reset_index(inplace = True, drop = False)
    df_prices_dt_beg.drop(columns = 'adjusted_close', inplace = True)
    df_prices_dt_beg = df_prices.merge(df_prices_dt_beg, how = 'inner', on = ['ticker', 'date'])
    # Ending Prices
    df_prices_dt_end = df_prices.groupby('ticker').max()
    df_prices_dt_end.reset_index(inplace = True, drop = False)
    df_prices_dt_end.drop(columns = 'adjusted_close', inplace = True)
    df_prices_dt_end = df_prices.merge(df_prices_dt_end, how = 'inner', on = ['ticker', 'date'])
    df = df.merge(df_prices_dt_beg, how = 'inner', on = 'ticker')
    df['shares'] = (df['allocation']/df['adjusted_close'])
```

```
    df.drop(columns = 'adjusted_close', inplace = True)
    df = df.merge(df_prices_dt_end, how = 'inner', on = 'ticker')
    df.rename(columns = {'date_x': 'beg_date', 'date_y': 'end_date'}, inplace = True)
    df['end_value'] = df['adjusted_close'] * df['shares']
    return df

df_sectors = pd.read_csv('sectors_sp500.csv')
df_all = pd.read_csv('daily_stock_data.csv')
df_merged = df_sectors.merge(df_all, how='outer', left_on='Symbol', right_on='ticker')
df_merged = df_merged[df_merged['date'].isnull() == True]
df_remaining = df_merged[['Symbol', 'Name', 'Sector']]


i = 0
df_all = pd.DataFrame()


for index, row in df_sectors.iterrows():
    i += 1
    ticker = row['Symbol']
    try:
        df_current = get_data(ticker)
        df_all = df_all.append(df_current)
    except:
        pass
    if i % 20 == 0:
        df_all.to_csv('daily_stock_data.csv')
df_all.to_csv('daily_stock_data.csv')

df_metrics = stock_returns('2006-01-01', '2015-12-31', 0.0188, '10yr_historical_metrics.csv')

df_simulation = pd.DataFrame()
budget = 1000000
stock_lower_limit = 0.01
stock_upper_limit = 0.12
sector_lower_limit = 0.04
sector_upper_limit = 0.16


for i in range(1, 8, 1):
    df_high_eror = generate_portfolio(df_metrics, i, 'High Return', 'eror', 'Descending', 'company_sector'
)
    df_high_risk = generate_portfolio(df_metrics, i, 'High Risk', 'beta', 'Descending', 'company_sector')
    df_low_risk = generate_portfolio(df_metrics, i, 'Low Risk', 'beta', 'Ascending', 'company_sector')
    df_rand1 = generate_portfolio(df_metrics, i, 'Random #1')
    df_rand2 = generate_portfolio(df_metrics, i, 'Random #2')
    df_rand3 = generate_portfolio(df_metrics, i, 'Random #3')
    df_rand4 = generate_portfolio(df_metrics, i, 'Random #4')
    df_rand5 = generate_portfolio(df_metrics, i, 'Random #5')
    for j in np.arange(0, 0.10, 0.01):
        sector_lower = sector_lower_limit + j
        sector_upper = sector_upper_limit - j
        if round(sector_lower,2) < round(sector_upper,2):
            for k in np.arange(0, 0.04, 0.005):
                stock_lower = stock_lower_limit + k
                stock_upper = stock_upper_limit - (3 * k)
                if round(stock_lower,3) < round(stock_upper,3):
                    df_high_eror_result = performance_portfolio(optimize_portfolio(df_high_eror, budget, s
tock_lower, stock_upper, sector_lower, sector_upper))
                    df_high_eror_result['n'] = i * 11
                    df_simulation = df_simulation.append(df_high_eror_result)
                    df_high_risk_result = performance_portfolio(optimize_portfolio(df_high_risk, budget, s
tock_lower, stock_upper, sector_lower, sector_upper))
                    df_high_risk_result['n'] = i * 11
                    df_simulation = df_simulation.append(df_high_risk_result)
                    df_low_risk_result = performance_portfolio(optimize_portfolio(df_low_risk, budget, sto
ck_lower, stock_upper, sector_lower, sector_upper))
                    df_low_risk_result['n'] = i * 11
                    df_simulation = df_simulation.append(df_low_risk_result)
                    df_rand1_result = performance_portfolio(optimize_portfolio(df_rand1, budget, stock_low
er, stock_upper, sector_lower, sector_upper))
```

```
                    df_rand1_result['n'] = i * 11
                    df_simulation = df_simulation.append(df_rand1_result)
                    df_rand2_result = performance_portfolio(optimize_portfolio(df_rand2, budget, stock_low
er, stock_upper, sector_lower, sector_upper))
                    df_rand2_result['n'] = i * 11
                    df_simulation = df_simulation.append(df_rand2_result)
                    df_rand3_result = performance_portfolio(optimize_portfolio(df_rand3, budget, stock_low
er, stock_upper, sector_lower, sector_upper))
                    df_rand3_result['n'] = i * 11
                    df_simulation = df_simulation.append(df_rand3_result)
                    df_rand4_result = performance_portfolio(optimize_portfolio(df_rand4, budget, stock_low
er, stock_upper, sector_lower, sector_upper))
                    df_rand4_result['n'] = i * 11
                    df_simulation = df_simulation.append(df_rand4_result)
                    df_rand5_result = performance_portfolio(optimize_portfolio(df_rand5, budget, stock_low
er, stock_upper, sector_lower, sector_upper))
                    df_rand5_result['n'] = i * 11
                    df_simulation = df_simulation.append(df_rand5_result)
        df_simulation.to_csv('simulation.csv', index = False)

td = pd.read_csv('simulation.csv')
```