

The University of Akron
IdeaExchange@UAkron

Honors Research Projects

The Dr. Gary B. and Pamela S. Williams Honors
College

Spring 2016


Diversification and Market Neutral Portfolios in S&P500

Alan S. Agnew

University of Akron, asa67@zips.uakron.edu

Please take a moment to share how this work helps you [through this survey](#). Your feedback will be important as we plan further development of our repository.

Follow this and additional works at: http://ideaexchange.uakron.edu/honors_research_projects

 Part of the [Applied Statistics Commons](#), and the [Finance and Financial Management Commons](#)

Recommended Citation

Agnew, Alan S., "Diversification and Market Neutral Portfolios in S&P500" (2016). *Honors Research Projects*. 378.
http://ideaexchange.uakron.edu/honors_research_projects/378

This Honors Research Project is brought to you for free and open access by The Dr. Gary B. and Pamela S. Williams Honors College at IdeaExchange@UAkron, the institutional repository of The University of Akron in Akron, Ohio, USA. It has been accepted for inclusion in Honors Research Projects by an authorized administrator of IdeaExchange@UAkron. For more information, please contact mjon@uakron.edu, uapress@uakron.edu.

Diversification and Market Neutral Portfolios in S&P500

BY ALAN AGNEW
THE UNIVERSITY OF AKRON

Table of Contents

1. Introduction	2
2. The Two Stock Case.....	2
3. Selecting Benchmark Stocks	3
Selecting Weights for the Stocks	5
Find the Suitable Portfolio	6
Proper Portfolios to Use	7
4. Comparisons to Overall Market and Other Techniques	7
Minimum Variance Portfolio Performance.....	7
Sharpe Ratio Portfolio.....	8
Equal (Constant) Weight Portfolio.....	9
5. Construction of Market Neutral Portfolio.....	10
Correlation	10
Alpha and Beta.....	12
Market Neutrality with Minimum Variance Portfolio.....	14
Stocks Being Market Neutral?.....	15
Conclusion.....	16
Appendix	17
More Stocks? Issues with Weight Creation Discussion	17
Specific Functions in Code	19
Rate	19
StandardStockDiv.....	19
SelectionStats	20
2BasicStocks.....	20
GridSelection.....	20
AgScaledUniform	21
GetWeightedPortfolios	22
BestPorts.....	23
SimulateHolding.....	23
GetProfolioRate	23
QuarterCor	24
Quarter Alpha and Beta	24
MarNeutral.....	24

1. Introduction

Our goal is to investigate strategies to deal with the risks associated with holding asset in the stock market. We first deal with risk of holding a specific stock, by the use of diversification. Later, we'll attempt to deal with the market risk, which is the risk of entire market going up and down. Data used in this project comes from daily adjusted closing price of stocks listed in the S&P500 index ranging from January 3rd, 2000 to December 31st, 2015 and the data is processed using statistical software R.

Sections 2 through 4 of this paper demonstrate diversification and how to lower stock specific risk. Section 2 shows a case with two stocks in a portfolio. Moreover, the ideal portion of your budget allocated to each stock in the portfolio will be discussed.

Section 3 scales up the discussion to twenty benchmark stocks in a portfolio. Creating thousands of potential portfolios was more difficult to compute, but it was necessary for Section 4. This part is the evaluation of how much money the portfolios create compared to just using the overall S&P500.

Section 5 discusses calculating a stock's beta, alpha, from Sharpe's single index model and correlation with the S&P500, then calculate the same measures for the portfolio overall. By using those measures, we'll attempt to make our portfolio neutral to the market. That is, on average, the portfolio's value will change independent of the market in crisis situations. This part will deal with the market risk component and mix in with our diversification from above to also deal with stock specific risk.

2. The Two Stock Case

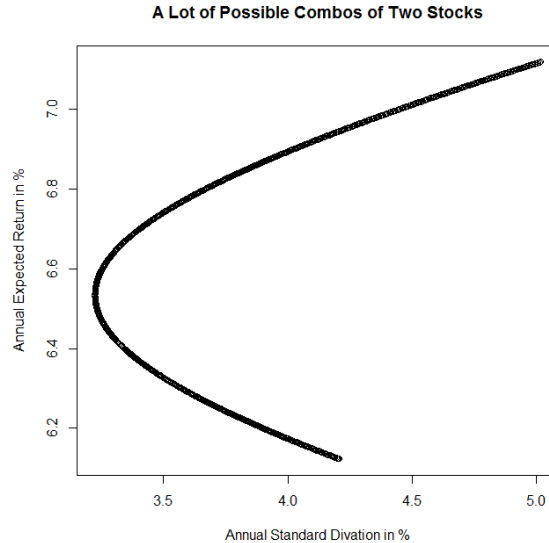
This section shows an example by using the first two stocks in a simple portfolio. We will select IBM as our first stock and Microsoft as our second.

Here is a table of annualized average returns and standard deviations of those returns for our two stocks when they are not paired up.

	Average Yearly Gain	Standard Deviation of Returns
IBM	6.12%	4.21%
Microsoft	7.12%	5.02%

The ideal portfolio will minimize the standard deviation of a combination of these stocks in a portfolio.

We will use uniformly distributed weights on zero to one that add to 1. We will create 10,000 such weights and create 10,000 portfolios with the weights to show how every possibility works for an investor. Notice how standard deviation and expected return of the portfolio show up soon. This code is under 2BasicStocks in Appendix, and the graph is below.



The most risk averse investor should choose the portfolio closest to 3.2% standard deviation. Thankfully, we have an easy command to find which amount should be invested in each stock by finding the minimum variance portfolio and its weights.

The table below shows the portions to give to each stock to achieve the minimum variance, and we also see the expected return and standard deviation of this portfolio in another table.

	IBM	Microsoft
Weights in Portfolio	0.59	0.41

Thus, investing 59% of your budget into IBM, and 41% in Microsoft results in this portfolio statistic below. That's the least risky portfolio we can make with these two stocks.

Annual Expected Return	Annual Standard Deviation of Return
6.53%	3.22%

It remains to be seen how much we can improve by adding 18 more stocks to our portfolio. Maybe this will further reduce standard deviation.

To do all of this, we loaded quantmod to process this data from Yahoo Finance, then we had to find daily changes of every stock in value. The Rate code in the Appendix shows this.

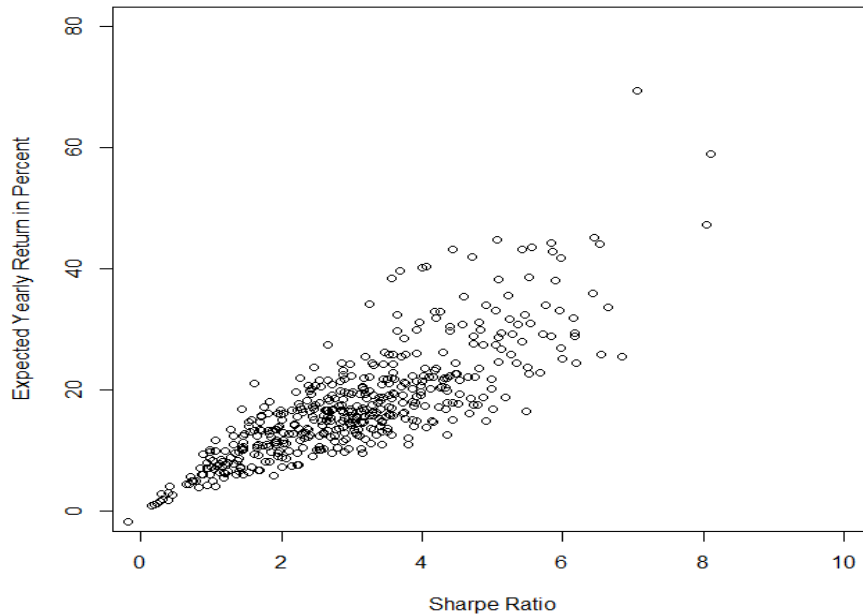
Then we had to find the standard deviations and expected return for all time in StandardStockDiv of Appendix. Following that, we select stocks and create basic statistics about our stocks for all time. That code is in SelectionStats of Appendix.

3. Selecting Benchmark Stocks

We can make a portfolio of however many stocks we want, but we also need to consider if it's a good idea to invest in a stock. If you invest in a stock with low expected returns and high

variance of actual returns, then you could be better off with other investments that have low returns and less risk.

To make such a comparison, we will take the expected return of the S&P 500 stocks and divide that by the standard deviation of those returns on a **yearly** basis. These values will be the x-axis of the following graph (Sharpe Ratio, where risk free rate is 0%), and the y-axis will feature the average returns of the stocks (for example, 10 indicates an average of 10% increase per 250 trading days, which is a whole year). The Sharpe Ratio is defined as (Expected return – risk free rate) / (Standard deviation of returns).



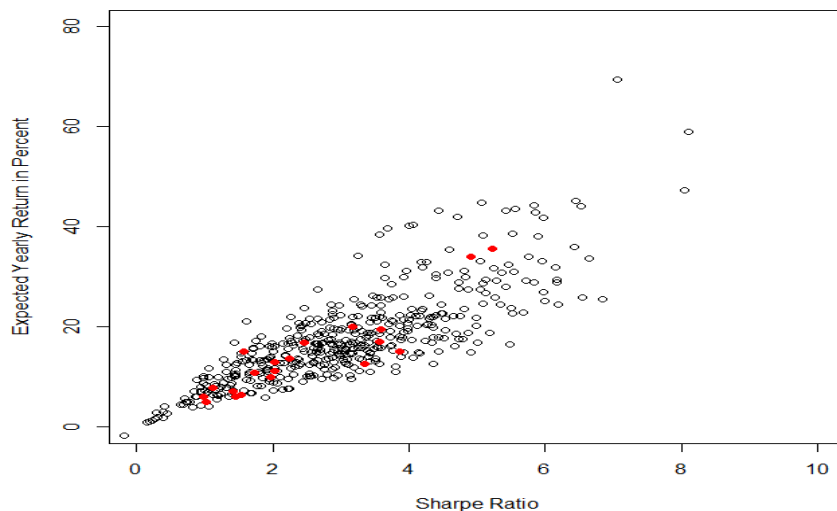
For any expected daily return, we would want the highest Sharpe Ratio.

Of course, these are historical return, and entire sectors of the stock market can be hit harder than others in the future. Thus, we'll support our portfolio's diversification by investing in many sectors of the economy. Here is a listing of what we chose for our examples that shows the sector, company, the average **yearly** return on its stock, and the standard deviation of those returns.

Sector	Company	E(X)	SD(X)
Tech	IBM	6.12%	4.21%
Tech	Apple	35.69%	6.83%
Tech	Microsoft	7.11%	5.01%
Special IT	Cerner Med Tech	34.12%	6.94%
Special IT	Waters Corporation	20.12%	6.34%
Energy	NiSource	15.11%	3.91%
Energy	Newfield Exploration CO	16.95%	6.92%
Telecommunications	AT&T	6.50%	4.24%
Telecommunications	Comcast	11.16%	5.51%
Food	McDonalds	12.73%	3.79%

Energy	CMS Energy Corp	9.94%	5.03%
Banking and Investing	Goldman Sacs	13.76%	6.16%
Banking and Investing	Bank of New York Mellon Corp	10.81%	6.28%
Banking and Investing	Vornado Realty Trust	19.58%	5.48%
Manufacturing	Masco	12.94%	6.36%
Manufacturing	General Electric	5.02%	4.95%
Manufacturing	Ford	7.87%	7.02%
Retail	Office Depot	15.11%	9.58%
Retail	The Gap	6.19%	6.36%
Retail	CVS	17.05%	4.78%

Let's highlight where those stocks lie in our earlier graph:



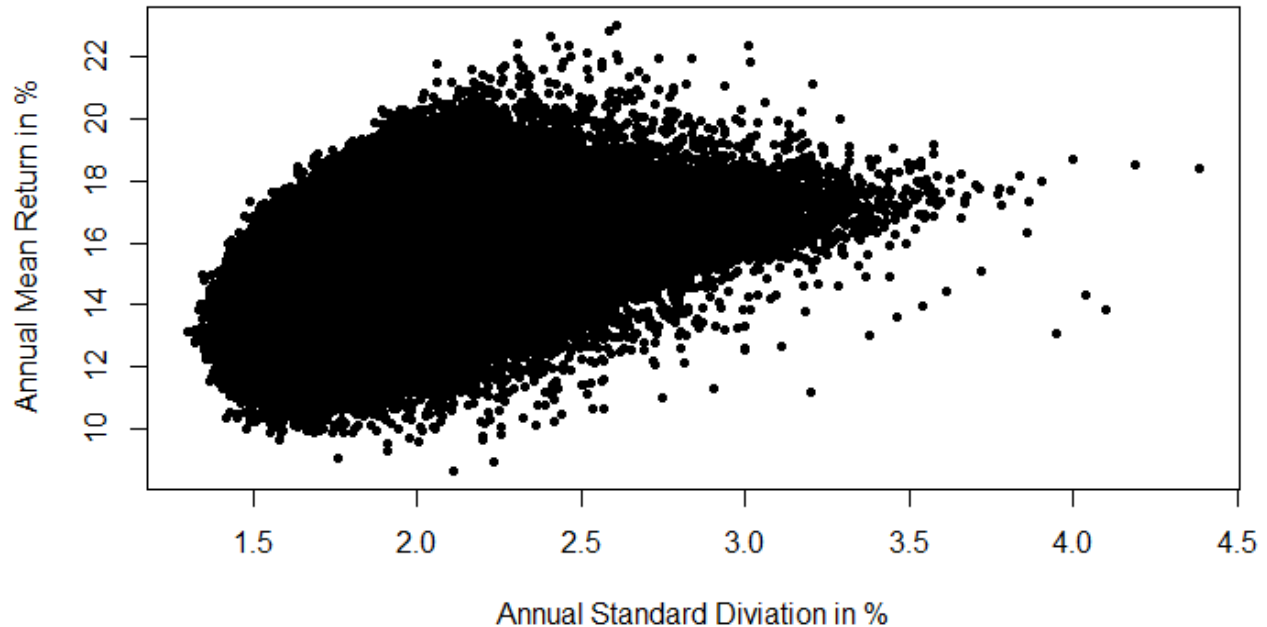
It seems that we could do better, but we will see how our portfolio optimizing will work in order to create the best weights for each of the stocks in our benchmark.

Selecting Weights for the Stocks

What percent of your investing budget should go into each of the stocks? We will take an approach of generating many sets of weights, then compare the performance for each set of weights assuming they are held constant for all 15 years. Those percentages will be the weights from 0 to 1, and we will show 100,000 possible portfolios that are created from the `AgScaledUniform` function that creates the weights. Also, `GetWeightedPortfolios` is a new function in our appendix that assigns all of the portfolios with these weights while giving the expected return and the standard deviation of each portfolio.

With a plot command, we create this dot plot to show all of the portfolios performances under the assumption that you invested into each portfolio in Year 2000 with a specific proportion of investing budget in each stock and never took any money out or in.

A lot of possible Combos for Twenty Stocks



[Find the Suitable Portfolio](#)

Now let's sort by the portfolios' standard deviation of returns.
`SortedPortfolio<-(PortfolioWeighted[order(PortfolioWeighted[,2]),])`

Then we can detect the minimum variance portfolio by looking at the first row of this sorted matrix. It's decent. We get about 13.16% expected return each year with only a 1.3% standard deviation on those returns.

ExpectedReturn	StandardDiv	WeightIndex
13.16%	1.3%	81284
13.13%	1.31%	32656
12.79%	1.31%	38507

Then we look at what portion of our budget should go into each stock with the following. Note that the abbreviations stand for the stock. AAPL means Apple, for example, on the S&P 500, and 0.047 means you should invest 4.766% of your budget into that stock to follow the minimum variance portfolio. Here is a table of stocks and weights:

IBM	MSFT	AAPL	CERN	WAT	T	CMCSA	MCD	MAS	GE
0.115	0.040	0.048	0.038	0.044	0.105	0.036	0.061	0.070	0.061
F	ODP	GPS	CVS	NI	NFX	CMS	GS	BK	VNO
0.026	0.005	0.029	0.072	0.041	0.054	0.038	0.028	0.059	0.031

Proper Portfolios to Use

We can also evaluate by finding the highest expected return of a portfolio given a standard deviation. We can do this by taking the Sharpe Ratio of all portfolios, assuming a risk free return of 0%, as in BestPorts in Appendix, and sorting the portfolios by that ratio.

This matrix gives you the most expected return per standard deviation. Keep in mind that our minimum variance portfolio has a Sharpe Ratio (of yearly returns) of about 10.118. The six highest Sharpe Ratios are below in descending order.

If you seek a portfolio that has returns greater than risk, then the one at the bottom suits you. Its returns are expected to be 11.63 times greater than the standard deviation of the returns on a yearly basis.

ExpectedReturn	StandardDiv	WeightIndex	Sharpe Ratio
17.18	1.52	80446.00	11.27
18.43	1.63	35875.00	11.29
16.77	1.49	94181.00	11.29
17.63	1.55	48341.00	11.35
16.88	1.47	28676.00	11.48
17.32	1.49	26328.00	11.63

Subsetting the matrix to cases where expected return is between, for example, 17 and 18% is possible, and the highest Sharpe Ratio among those would then be the best portfolio because you get the most return with the least risk.

Another way would be multiplying Expected Return by 10, rounding it, then dividing by 10, then ordering the matrix by expected return and standard deviation. That would make ordering the matrix easy, and you'd see many portfolios with, say, 16.5% average return, and a sorted list of the standard deviations. All you do then is select the one with the lowest standard deviation for the desired expected return of 16.5%.

In the end, given an amount of risk you're willing to take, you can fully realize the returns that your risk ought to earn.

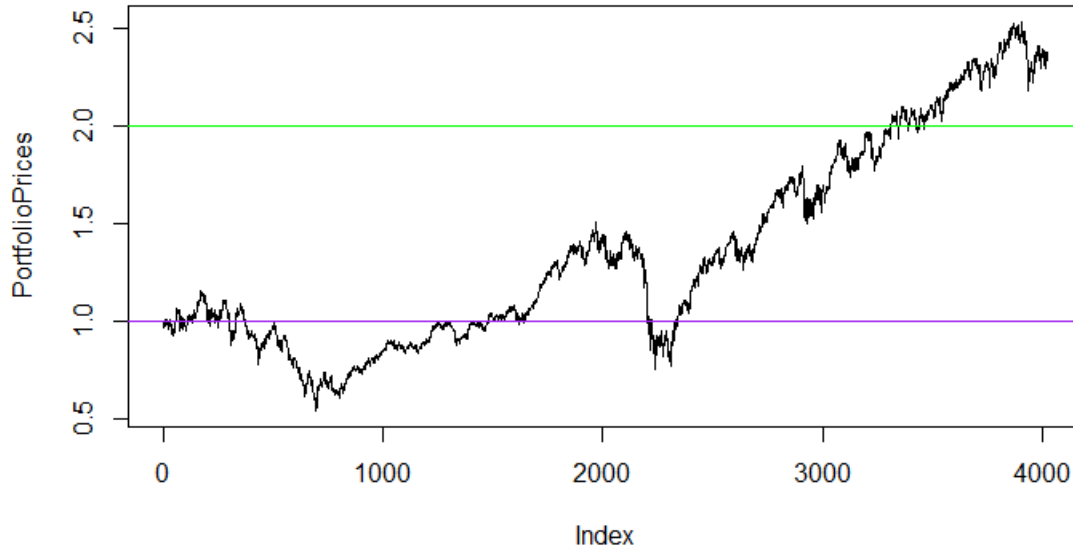
4. Comparisons to Overall Market and Other Techniques

To see how a given portfolio's value moves over time, we have a function called SimulateHolding in the Appendix that will let us track how a portfolio preforms given a combination of weights and a selection of stocks.

There will be three portfolios compared to the overall market. One is the minimum variance, the other is Sharpe Ratio max, and the last is just investing 1/20 in all 20 stocks. This will show how different diversification options perform along the ups and downs of the market.

Minimum Variance Portfolio Performance

Suppose we start with one dollar in the minimum variance portfolio in 2000, then let it grow for each of the trading days until December 2015. We would get this chart that shows the performance of that dollar:



Let's compare it to the overall market index of the S&P500
D.ad[, 1]/110.33

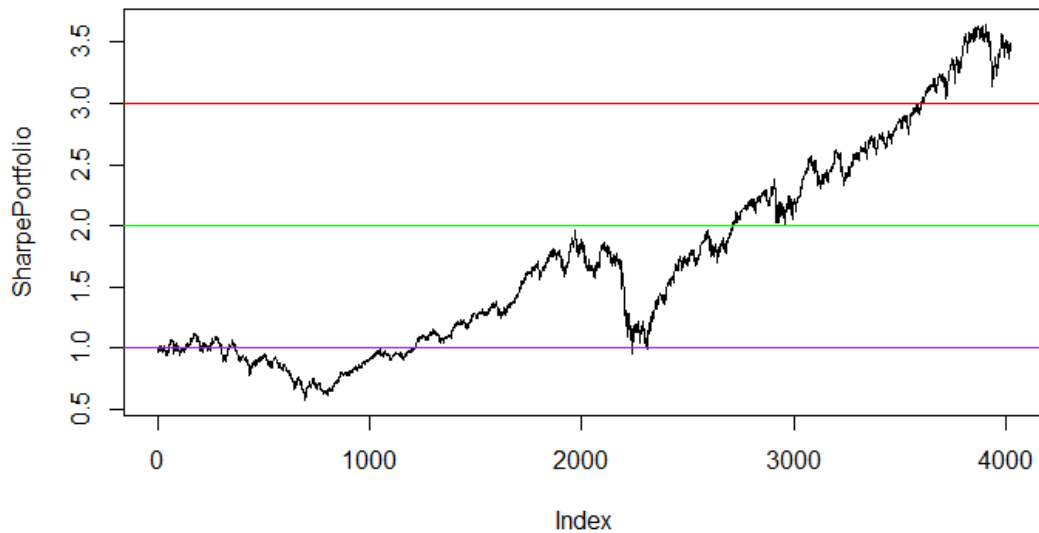


As we can see, our minimum variance portfolio has turned a dollar into about two and a half dollars, where the market only turned that one dollar into \$1.8. That's good news, and it suggests that our minimum variance portfolio of twenty stocks can easily beat the market over time.

Sharpe Ratio Portfolio

Let's try the same with the portfolio that has the highest Sharpe Ratio:

Maximum Sharpe Ratio Portfolio Performance

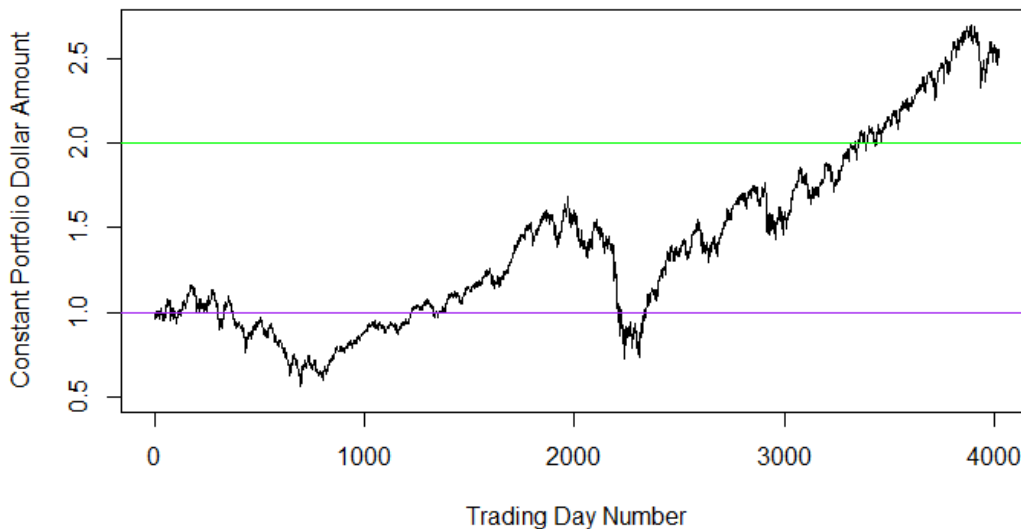


That isn't too shabby of a performance for 20 stocks. A dollar invested in 2000 grew to over three dollars by 2015's end.

Equal (Constant) Weight Portfolio

How about the simplest case that we can think of where you invest a twentieth of your budget into each of the twenty stocks? The performance of this method is shown below:

Constant Portfolio Growth of \$1



In this case, the simple method of giving 5% of your budget to each of the stocks resulted in a graph very alike to the minimum variance portfolio. In fact, notice how close their standard deviations and average daily returns are in the next table.

We see that the minimum variance portfolio has a lower standard deviation and average rate increase on value daily.

Daily EqualPortfolio Std Div	Daily MinVariancePortfolio Std Div
1.39%	1.31%
Average EqualPortfolio Incre	Average MinVariancePortfolio Increase
0.033%	0.030%

Perhaps the constant portfolio did well because of good stock selection. Had we selected worse stocks into the portfolio, it may have been a bad idea to hold onto any of it. At the same time, the returns on the simple portfolio we saw are not the best compared to the maximum Sharpe Ratio portfolio or other portfolios we could make by seeking a desired return for the lowest standard deviation.

While the constant portfolio is hardly much better than the minimum variance portfolio, simply running a weight selection program like we did is still worth the time if you truly want the lowest risk. After all, it will take less than an hour, and we were doing this simulation on stocks being held for 15 years.

Otherwise, if you want more returns and like some risk, the Sharpe Portfolio is an example of one ideal situation, but many more portfolios can be crafted given a range of risk the investor is willing to take on.

Note that these were all buy and hold strategies for the long term. Diversifying enabled us to overcome risk held by individual stocks wildly moving, but we need to address the market risk next that all the portfolios deal with. Beating market performance with effective diversification is only the start.

5. Construction of Market Neutral Portfolio

Our methods in the last section have dealt with the price movements of individual stocks. However, they have not addressed the movements of the market that take the portfolio down quickly. After all, we have dealt with stock specific risk, but portfolios still need to perform better against sudden market drops.

Correlation

Perhaps correlation between the stocks in the portfolio and with the overall market plays a role. Below is a table of correlations regarding the natural log of the returns of one stock to another's. To do this, we took the rates of all daily changes, then we took the natural log of those changes, then we asked for every possible correlation of the stocks to each other and the market.

Notice that IBM is in the first column, and S&P500 is in the last column.

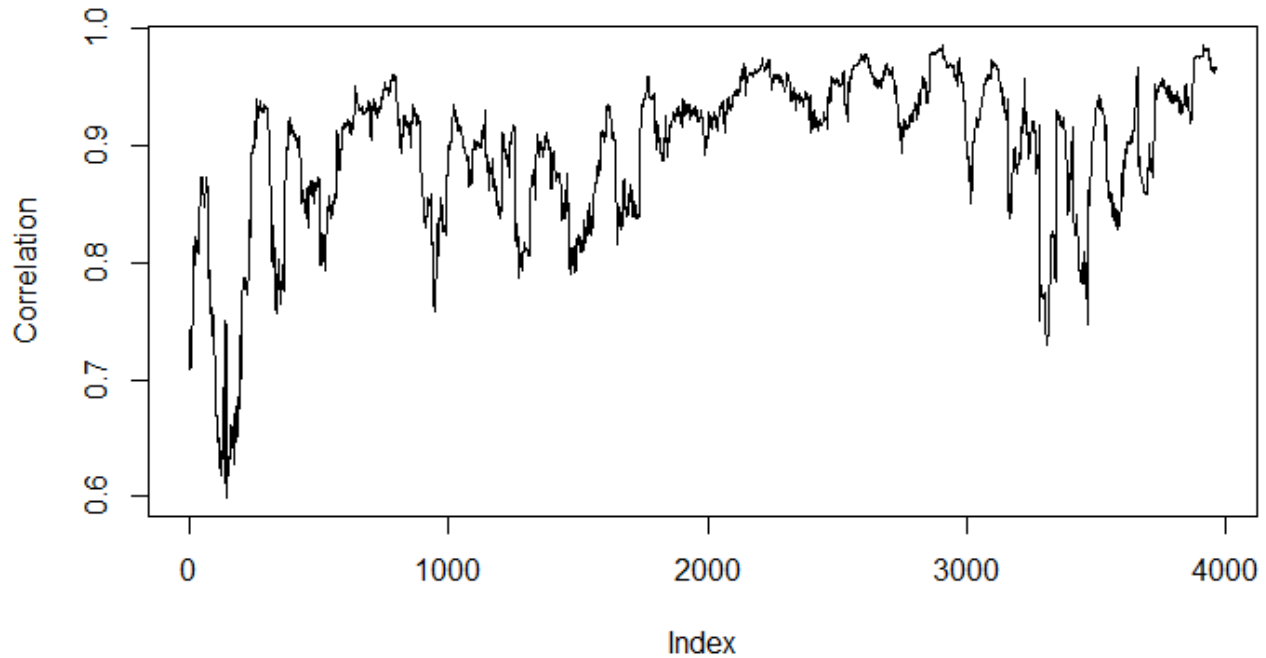
Alan Agnew Honors Report

IBM	MSFT	AAPL	CERN	WAT	T	CMCSA	MCD	MAS	GE	F	ODP	GPS	CVS	NI	NFX	CMS	GS	BK	VNO	SPY
1.00	0.50	0.39	0.30	0.27	0.38	0.41	0.30	0.35	0.49	0.33	0.31	0.28	0.26	0.34	0.27	0.30	0.46	0.42	0.34	0.64
0.50	1.00	0.40	0.31	0.27	0.37	0.42	0.27	0.35	0.45	0.34	0.29	0.32	0.26	0.31	0.28	0.23	0.47	0.42	0.35	0.65
0.39	0.40	1.00	0.24	0.25	0.25	0.28	0.21	0.25	0.35	0.26	0.24	0.22	0.20	0.19	0.21	0.17	0.35	0.32	0.26	0.49
0.30	0.31	0.24	1.00	0.19	0.22	0.26	0.17	0.25	0.31	0.22	0.22	0.25	0.18	0.16	0.20	0.16	0.30	0.26	0.24	0.41
0.27	0.27	0.25	0.19	1.00	0.21	0.26	0.19	0.25	0.32	0.21	0.24	0.23	0.23	0.24	0.24	0.19	0.33	0.30	0.24	0.44
0.38	0.37	0.25	0.22	0.21	1.00	0.39	0.29	0.33	0.40	0.32	0.28	0.25	0.28	0.35	0.25	0.33	0.40	0.43	0.36	0.58
0.41	0.42	0.28	0.26	0.26	0.39	1.00	0.30	0.37	0.47	0.34	0.32	0.32	0.32	0.39	0.30	0.33	0.45	0.47	0.40	0.63
0.30	0.27	0.21	0.17	0.19	0.29	0.30	1.00	0.30	0.36	0.27	0.26	0.27	0.29	0.29	0.22	0.24	0.29	0.35	0.32	0.47
0.35	0.35	0.25	0.25	0.25	0.33	0.37	0.30	1.00	0.50	0.42	0.44	0.35	0.32	0.34	0.34	0.28	0.46	0.47	0.50	0.60
0.49	0.45	0.35	0.31	0.32	0.40	0.47	0.36	0.50	1.00	0.44	0.41	0.39	0.33	0.37	0.34	0.32	0.54	0.57	0.53	0.73
0.33	0.34	0.26	0.22	0.21	0.32	0.34	0.27	0.42	0.44	1.00	0.34	0.29	0.23	0.30	0.27	0.27	0.40	0.41	0.36	0.53
0.31	0.29	0.24	0.22	0.24	0.28	0.32	0.26	0.44	0.41	0.34	1.00	0.36	0.30	0.27	0.30	0.21	0.40	0.39	0.44	0.51
0.28	0.32	0.22	0.25	0.23	0.25	0.32	0.27	0.35	0.39	0.29	0.36	1.00	0.28	0.24	0.21	0.20	0.36	0.38	0.33	0.49
0.26	0.26	0.20	0.18	0.23	0.28	0.32	0.29	0.32	0.33	0.23	0.30	0.28	1.00	0.29	0.22	0.22	0.26	0.36	0.31	0.45
0.34	0.31	0.19	0.16	0.24	0.35	0.39	0.29	0.34	0.37	0.30	0.27	0.24	0.29	1.00	0.36	0.47	0.34	0.38	0.40	0.53
0.27	0.28	0.21	0.20	0.24	0.25	0.30	0.22	0.34	0.34	0.27	0.30	0.21	0.22	0.36	1.00	0.27	0.35	0.33	0.37	0.51
0.30	0.23	0.17	0.16	0.19	0.33	0.33	0.24	0.28	0.32	0.27	0.21	0.20	0.22	0.47	0.27	1.00	0.29	0.32	0.31	0.45
0.46	0.47	0.35	0.30	0.33	0.40	0.45	0.29	0.46	0.54	0.40	0.40	0.36	0.26	0.34	0.35	0.29	1.00	0.63	0.48	0.71
0.42	0.42	0.32	0.26	0.30	0.43	0.47	0.35	0.47	0.57	0.41	0.39	0.38	0.36	0.38	0.33	0.32	0.63	1.00	0.56	0.71
0.34	0.35	0.26	0.24	0.24	0.36	0.40	0.32	0.50	0.53	0.36	0.44	0.33	0.31	0.40	0.37	0.31	0.48	0.56	1.00	0.62
0.64	0.65	0.49	0.41	0.44	0.58	0.63	0.47	0.60	0.73	0.53	0.51	0.49	0.45	0.53	0.51	0.45	0.71	0.71	0.62	1.00

This tells us that everything is correlated to each other over a 15 year period. We need more specific information like a correlation based on return rates in the past quarter of a year to make better judgements.

Let's try this by finding the correlation of our minimum variance portfolio based on the previous sixty trading days. This is a quarter running correlation, and it may show spikes of correlation during downturns in the market. Code for this one is in QuarterCor of Appendix.

Quarter Running Correlation of Portfolio to Market



It shows that the portfolio generally correlates with the market no matter what the trouble or good of the day is. We'll need a way to measure how much the portfolio responds to market movement, and we'll do that with beta next.

Alpha and Beta

We'll use the Alpha and Beta code from the Appendix to calculate these quarter running estimates in single index model. By using these estimates, we hope to obtain market neutrality later on.

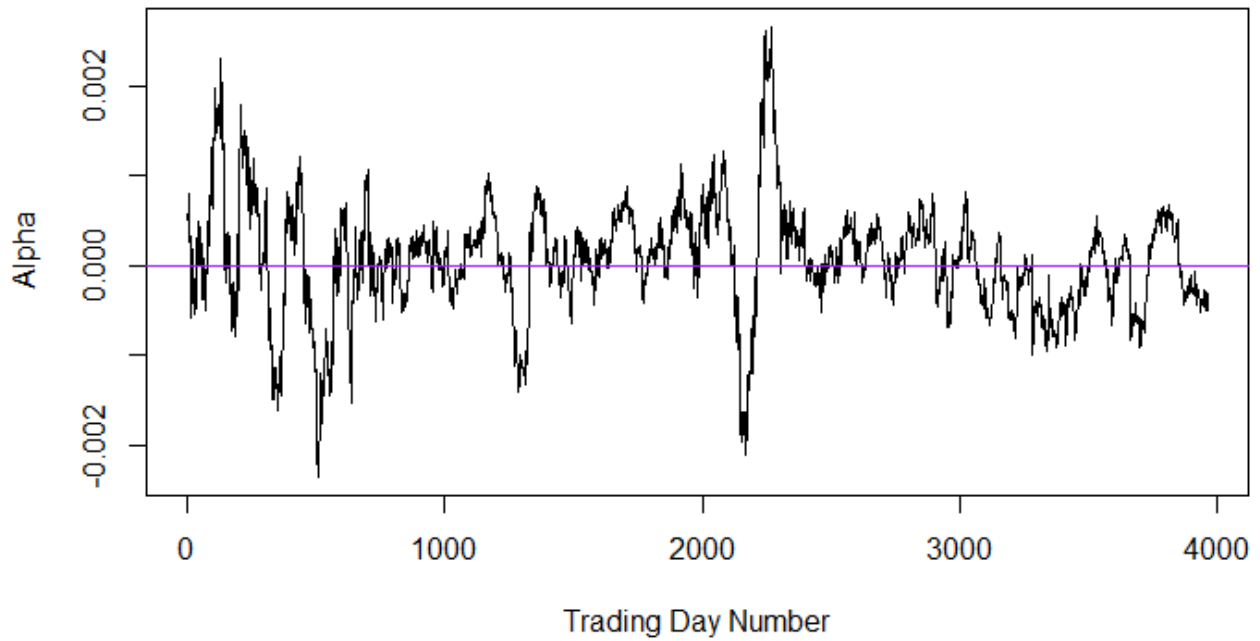
Alpha and beta are like coefficients in linear regression estimates of the return of the portfolio based on returns in the overall market. For example, alpha is the intercept, which may suggest the portfolio has a 10% increase higher than the market return with the portfolio based on the past sixty trading days. Beta measures how much return the portfolio gets when the market increases. For example, if beta is 1.1 and the market increases 1%, then beta would predict that the portfolio increases 1.1%

Keep in mind that the beta works if the market goes down, too. Thus, a 1% decrease in market will imply a 1.1% decrease in portfolio if they share that 1.1 beta.

Next two charts show alpha and beta of minimum variance portfolio to the S&P500 index.

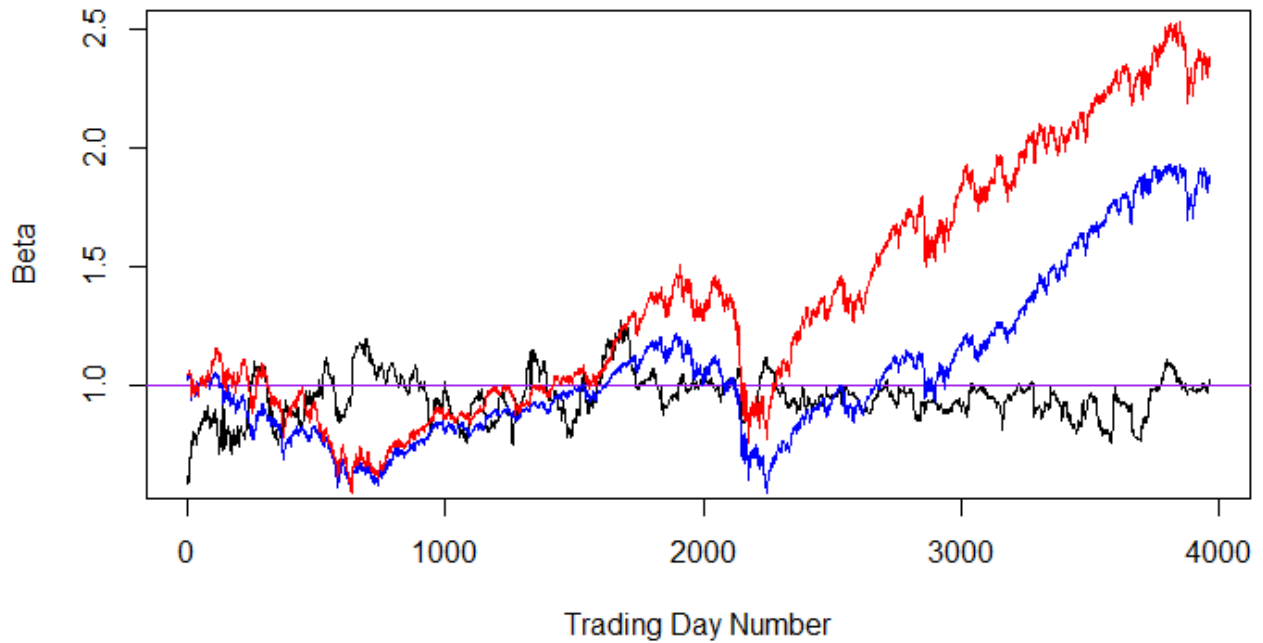
For this chart of alpha, keep in mind that this is tracking the daily trading day rates. As such, they will be rather small since we usually see a stock move maybe .1% or less over a single day.

Quarter Running Estimate of Profolio's Alpha



It spikes up and down all the time. It's rather close to zero, so the beta is more important, perhaps. However, to display how beta relates, we will plot the portfolio's value in red, and the stock market's overall value with blue.

Quarter Running Estimate of Portfolio's Beta



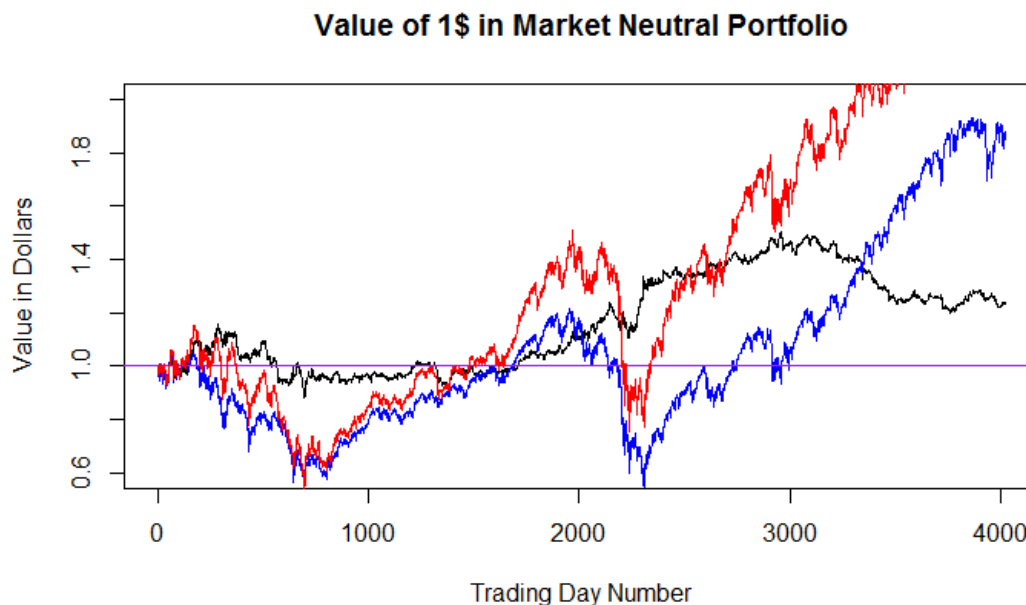
That chart shows that the fluctuations in beta have generally helped the portfolio rise up and up, but the beta also harms the portfolio during downturns. Let's see what can help against that.

Market Neutrality with Minimum Variance Portfolio

We will create a market neutral portfolio using the code MarNeutral in Appendix. This works by using the quarterly estimates of the beta of the portfolio to the market, then shorting the market index by that beta. For example, if beta is 0.9 based on today's quarterly estimate, and the market goes down 0.1%, then you gain 0.09% because of how you shorted the market. However, since the portfolio is correlated with market, the portfolio will likely go down 0.09% that day, too, but you end up breaking even.

That's the theory, so let's see the actual performance of this method.

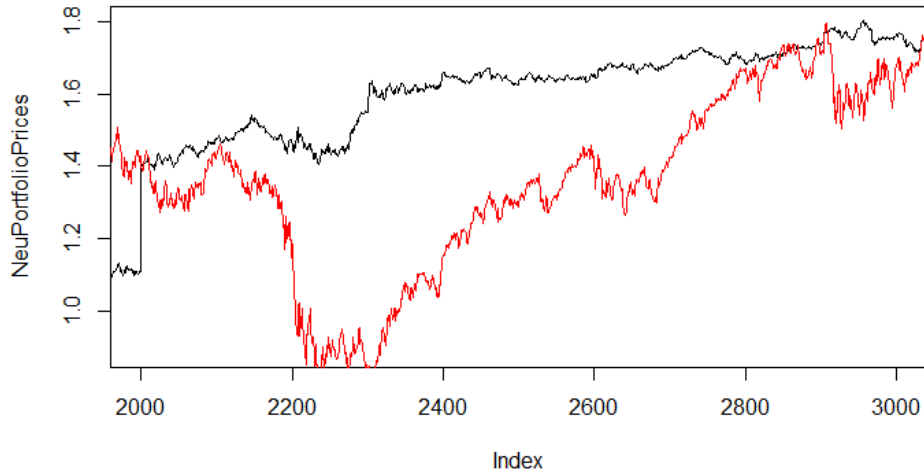
Blue line is the overall stock market value, red is the minimum variance portfolio, and the black is the market neutral portfolio.



Turns out that the market neutral portfolio did better than the stock index and minimum variance portfolio for some time around 2008. Market neutral portfolios don't give excellent returns though, so they should be used when expecting a downturn. Engaging the market neutral portfolio at, say, trading day 2000 would have a better chart as we see below.

The red line displays the minimum variance portfolio gaining up, but the black displays the value of the market neutral portfolio assuming you started using it on trading day 2000. You didn't need to predict the crash, you just needed to expect it in a long time to save a lot of value. If you bought back into the minimum variance portfolio, then the gains will be even better.

Let's show that behavior with the next chart. Assume that you bought into the market neutral portfolio many days before the 2008 crash and got off the ride of your old minimum variance portfolio.



Not too bad, but it would be even better if the investor went off of the market neutral portfolio once the crash ended. That way, the future returns would flow into the investor's pockets.

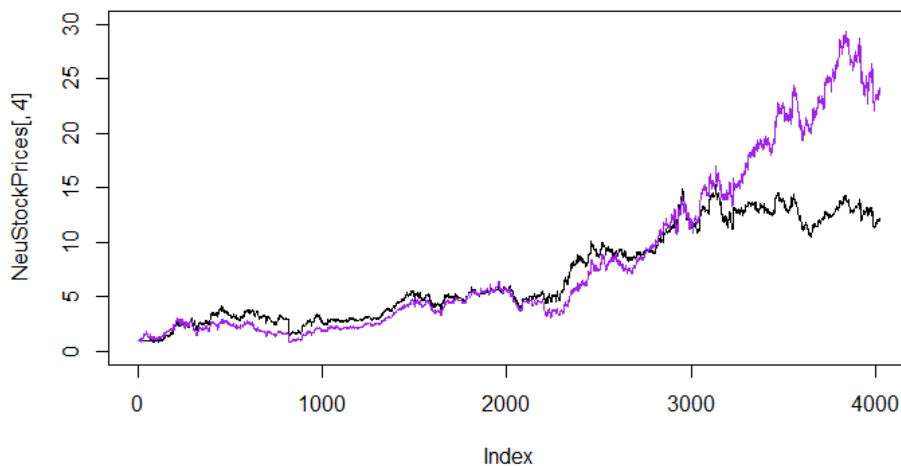
Stocks Being Market Neutral?

Market neutrality does not seem to be as effective when applied on individual stocks. This might be due to stock specific risk overwhelming the technique, but we'll see how different it is when you apply it to individual stocks instead of a portfolio.

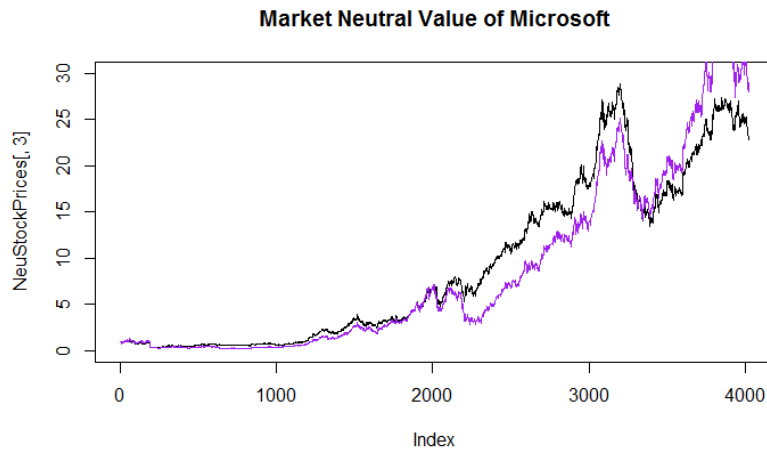
Let's take the beta of each of our portfolio's stocks to the market, then short the market with those betas like with the portfolio earlier. New values of the stocks over time are calculated and this code is under StocksNeutral in Appendix.

Of course, market neutrality dampens returns and losses, but we'll notice that it doesn't do that much damage. The purple line on the next chart shows Cerner Corp's stock value on its own if you invest one dollar into it. Now watch the market neutral value of that stock in action with the black line:

Market Neutral Value of Cerner



Microsoft has similar oddities, but we notice more clearly that the market neutral method beat holding Microsoft alone during the 2008 issues:



The reason that the market neutral method is actually giving a good return may have more to do with how inaccurate the estimate of beta is for a single stock. Perhaps it is easier to estimate beta for a portfolio because of the amount of stocks that reduce stock specific risk.

We don't know if the market neutral method displays much use for stocks, but Microsoft's above chart indicates it may still help absorb shock from market wide disruptions.

Conclusion

This project has demonstrated possibility that diversification can achieve very high profit, beating the market by wide margin. Diversification significantly reduces the risks in specific stocks, but it does not address the market risk of drops across all stocks. A well-diversified portfolio has possibility of enjoying the strategy of market neutrality, making a profit when the whole market is going down, but it requires the investor to know when to use such a strategy. It doesn't have to be the day before a crash, it could be 100 trading days, but it would nonetheless absorb the shock of the fall. This would allow the investor stay afloat during the financial storm, and wait for the market to rise again.

Further questions to address include evaluating more strategies of when to buy into market neutral and when to get back to the normal market to maximize returns. Also, we may see if this has any improvement over simply selling the stocks and going into less risky assets like treasuries while waiting for a market crash.

Finally, going into other markets like real estate, commodities, and bonds may dampen the stock market risk of your portfolio. However, it may be possible that other markets share the same risk of crashing as the stock market does. Further investigation may find further diversification techniques handy to dip into these markets.

Appendix

More Stocks? Issues with Weight Creation Discussion

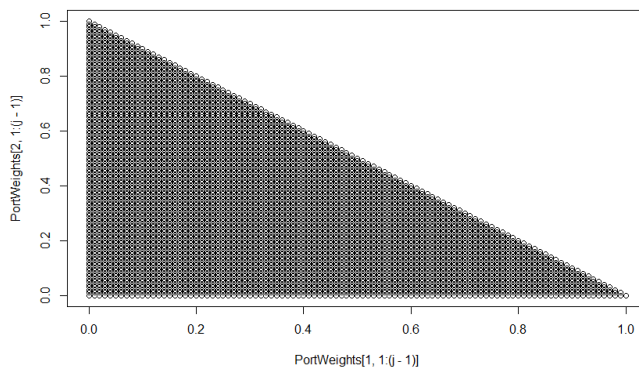
The same procedure does not scale up with three or more weights. We could write a program that generates the values of two uniform random numbers between 0 and 1, then generates them again if their sum is greater than 1. (The third value would be 1 minus their sum, and that would be the third weight.) However, this is a terrible way to scale up.

To get any meaningful diversification with, say, twenty stocks, you would need the first 19 random numbers, on average, to be less than $1/19$. The chances of that are extraordinarily low, and millions of attempts in a computer algorithm failed to produce even one combination of the twenty weights that satisfies $X_1+X_2+\dots+X_{20}=1.0$.

Therefore, we attempted to select every possible combination discretely and not randomly. The process works as such and the code is listed in the Appendix as GridSelection:

- Define spacing. We used 0.01 so that X_i could be 0, 0.01, and so on to 1 for i in 1 to our number of desired weights (call it n).
- Set Guess to 1, j to 1, max size of matrix, and X_1 to -0.99.
- Begin While loop that stops if we have more combinations than the max size, or if our final weight, say, $X_{(n-1)}$ is 1.01.
 - If Guess is 1, increase X_1 by 0.01
 - Else, increase $X(\text{Guess})$ by 0.01 and set X_i where $i < \text{Guess}$ to 0.
 - If any X_i is over 1, increase $X_{(i+1)}$ by 0.01 and set X_i to 0.
 - Assign $X_1, X_2, \dots,$ and $X_{(n-1)}$ to our matrix of weights.
 - If sum of those weights is greater than 1, set j to $j-1$, set Guess to Guess+1. Also, if Guess $>$ $(n-1)$, set Guess to 1.
 - Else, set X_n to the matrix as $1-(X_1+X_2+\dots+X_{(n-1)})$, and set Guess to 1.
 - Set j to $j+1$.
- If j hits max size, print out that max size was reached.

At the end, we can plot something like this:

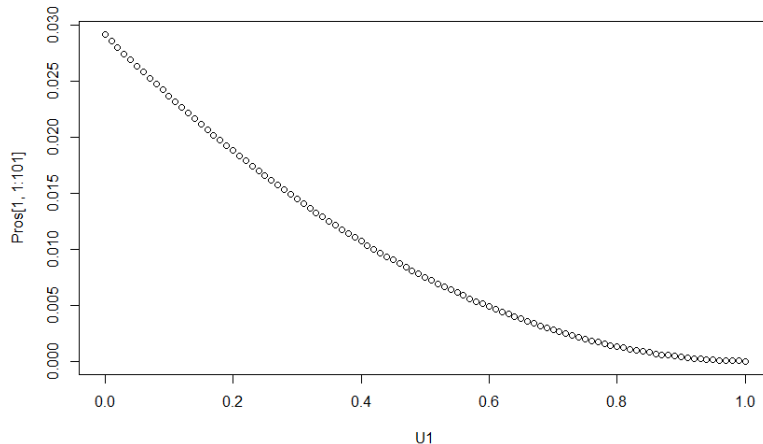


Notice how uniform X_1 is to X_2 . X_1 behaves like this to $X_3, X_4,$ and so on.

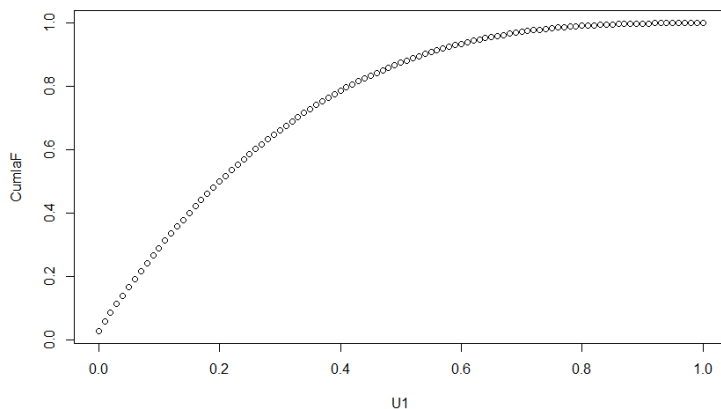
However, this program feels an insane about of memory usage. When we use 4 weights in a portfolio, it consumes 305 MBs. At 5 weights, it goes up to 3GBs, and we can see this is way out of our hardware capabilities.

Still, we can learn what individual weights should look like. After all, if we know the distribution of the weights of a portfolio from this method, we can calculate weights like when we did it for just two weights earlier. We will scale up to more than two weights this time.

After calculating how frequently a weight of a value (like 0.01) shows up, we can plot the density to show how likely it is to see .5 or .9 for X1 like below:



Seems that a value of .9 for X1 has a chance below 0.5% of happening. We calculated a density curve, so how about a cumulative curve:



It turns out this represents the curve of $-(x-1)^2+1$. Therefore, using inverse transformation, the distribution of our weights is $1-(1-x)^{1/2}$.

In fact, if there are 3 or more weights to calculate, then, for U where U is uniformly distributed between zero and one, the distribution of any weight is given by

$$1-(1-U)^{1/[\text{Number of weights} - 2]}$$

To implement this into a function that generates the weights for stocks, we use the code called AgScaledUniform from Appendix.

Now we need to use the weights that this code generates in order to plot returns

Specific Functions in Code

Rate

```
setwd("C://Users//Alan//Documents//2016_05_AlanAgnew-H")
library(quantmod) #Not used in code yet, good to get used to though
WorkD<- as.data.frame(D.ad, drop=FALSE)
```

```
nr<-nrow(D.ad)
nc<-ncol(D.ad)
#start with finding rates. Should be useful.
#dif<-matrix(NA,nr-1,nc)
rate<-matrix(NA,nr-1,nc)
```

```
#Danger, high computation load here in this loop.
#May not need to use this anymore thanks to our attaching.
```

```
computeRates<- function(){
#dif<-matrix(NA,nr-1,nc)
rate<-matrix(NA,nr-1,nc)
for(i in 1:nc){
  for(j in 1:(nr-1)){
    #dif[j,i]<-WorkD[j+1,i]-WorkD[j,i]
    rate[j,i]<-WorkD[j+1,i]/WorkD[j,i]
  }
}
colnames(rate)<-colnames(D.ad)
return(rate)
}
rate<-computeRates()
```

StandardStockDiv

```
MeanVal<-matrix(NA,1,nc)
StockSD<-matrix(NA,1,nc)
UnTradedDays<-matrix(NA,1,nc)
MeanRateChange<-matrix(NA,1,nc)
SDRateChange<-matrix(NA,1,nc)
```

```
for(i in 1:nc){
  UnTradedDays[i]<-sum(is.na(WorkD[,i]))
  MeanVal[i]<- mean(WorkD[,i], na.rm=TRUE)
  StockSD[i]<- sd(WorkD[,i], na.rm = TRUE)
  SDRateChange[i]<-sd(rate[,i], na.rm=TRUE)
  MeanRateChange[i]<-mean(rate[,i], na.rm=TRUE)
}
```

```
BindedAvg<-cbind(t(MeanRateChange),t(SDRateChange))
```

Alan Agnew Honors Report

SelectionStats

```
SelectedStocks <-  
c(230,301,4,89,456,416,101,282,279,196,175,330,205,125,317,316,105,206,63,452)  
#Record how many stocks you have  
NumWeights<-length(SelectedStocks)  
  
SDtoMeanPortfolio<-  
matrix(c(MeanRateChange[SelectedStocks],SDRateChange[SelectedStocks]),nrow=length(SelectedStocks),ncol=2)  
  
#There are 250 trading days a year  
NonTradingDays<-106  
PortfolioAnnualized<-matrix(0,length(SelectedStocks),2)  
  
for(i in 1:(length(SelectedStocks))){  
  PortfolioAnnualized[i,1]<-(SDtoMeanPortfolio[i,1]^(356-NonTradingDays)-1)*100  
  PortfolioAnnualized[i,2]<-SDtoMeanPortfolio[i,2]*(356-NonTradingDays)  
}
```

2BasicStocks

```
NumWeights<-10000  
SeqWeights<-seq(0,1,length.out=NumWeights)  
#Create matrix then chart  
PortfolioWeighted<-matrix(0,NumWeights,2)  
for(i in 1:NumWeights){  
  PortfolioWeighted[i,1]<-SeqWeights[i]*PortfolioAnnualized[1,1]+(1-  
SeqWeights[i])*PortfolioAnnualized[2,1]  
  PortfolioWeighted[i,2]<- ((SeqWeights[i]*PortfolioAnnualized[1,2])^2+((1-  
SeqWeights[i])*PortfolioAnnualized[2,2])^2)^.5  
#Note that we assumed no covariance!  
}  
plot(PortfolioWeighted[,2],PortfolioWeighted[,1],main="A lot of possible Combos for Two  
Stocks",xlab="Annual Standard Deviation in %",ylab="Annual Mean Return in %")
```

GridSelection

```
Weights<-4  
MaxSize<-10000000  
Byit<-0.01  
  
U1<-seq(0,1,by=Byit)  
AllSeq<-matrix(0,length(U1),Weights)  
  
for(i in 1:Weights){  
  AllSeq[,i]<-U1  
}  
  
PortWeights<-matrix(0,Weights,MaxSize)  
  
IndexSeq<-matrix(1,(Weights-1),1)
```

```
IndexSeq[1]<-0
```

```
j<-1
```

```
Guess<-1
```

```
while(j < MaxSize && IndexSeq[Weights-1]<length(U1)){
```

```
  if(Guess==1){
```

```
    IndexSeq[Guess]<-IndexSeq[Guess]+1
```

```
  }
```

```
  else{
```

```
    for(i in 1:(Guess-1)){
```

```
      IndexSeq[i]<-1
```

```
    }
```

```
    IndexSeq[Guess]<-IndexSeq[Guess]+1
```

```
    #The guess may be wrong, but our summation after we obtain the next set of
```

```
    #Weights will show if our guess was right, and it likely is.
```

```
  }
```

```
  for(i in 1:(Weights-2)){
```

```
    if(IndexSeq[i]>length(U1)){
```

```
      IndexSeq[i+1]<-IndexSeq[i+1]+1
```

```
      IndexSeq[i]<-1
```

```
    }
```

```
  }
```

```
  for(i in 1:(Weights-1)){
```

```
    PortWeights[i,j]<-AllSeq[IndexSeq[i],i]
```

```
  }
```

```
  if(sum(PortWeights[,j]) > 1){
```

```
    j<-j-1
```

```
    Guess<-Guess+1
```

```
    if(Guess>(Weights-1)){
```

```
      Guess<-1
```

```
    }
```

```
  }
```

```
  else{
```

```
    PortWeights[Weights,j]<-1-sum(PortWeights[,j])
```

```
    Guess<-1
```

```
  }
```

```
  j<-j+1
```

```
}
```

```
if(j>=MaxSize){
```

```
  cat("Max size of",MaxSize, "reached.")
```

```
}
```

```
j-1
```

```
AgScaledUniform
```

```
AgScaledUniform<-function(sim,Weights){
```

```
  Selection<-matrix(0,Weights,sim)
```

```
  X<-matrix(0,Weights,1)
```

```
  Y<-matrix(0,Weights,1)
```

```
  for(k in 1:sim){
```

```
    D<-runif((Weights-1),0,1)
```

Alan Agnew Honors Report

```
for(i in 1:length(D)){
  Y[i]<-1-(1-D[i])^(1/Weights)
}
while(sum(Y)>1){
  D<-runif((Weights-1),0,1)
  for(i in 1:length(D)){
    Y[i]<-1-(1-D[i])^(1/Weights)
  }
}
for(i in 1:(Weights-1)){
  X[i,1]<-Y[i,1]
}
X[Weights]<-(1-sum(Y))
for(j in 1:Weights){
  Selection[j,k]<-X[j]
}
}
return(Selection)
}
```

GetWeightedPortfolios

```
GetWeightedPortfolios<-function(sim,NumWeights,PortfolioAnnualized,NWeights){
PortfolioWeighted<-matrix(0,sim,3)
colnames(PortfolioWeighted) <- c("ExpectedReturn","StandardDiv", "WeightIndex")
for(j in 1:sim){
  for(i in 1:NumWeights){
    #First, mean returns are in column one
    PortfolioWeighted[j,1]<-
PortfolioWeighted[j,1]+PortfolioAnnualized[i,1]*NWeights[i,j]
    PortfolioWeighted[j,2]<-
PortfolioWeighted[j,2]+(PortfolioAnnualized[i,2]*NWeights[i,j])^2

    #Note that we assumed no covariance!
    #Mark what sectorArea we use in each. Good for after ordering and sorting
    #To trace the source.
    PortfolioWeighted[j,3]<-j
  }
}
#You need to take the square root to get actual SD!
for(i in 1:sim){
  PortfolioWeighted[i,2]<-(PortfolioWeighted[i,2])^0.5
}
return(PortfolioWeighted)
}
PortfolioWeighted<-GetWeightedPortfolios(sim,NumWeights,PortfolioAnnualized,NWeights)
```


Alan Agnew Honors Report

BestPorts

```
BestPorts<-matrix(0,sim,4)
BestPorts[,1:3]<-SortedPortfolio
for(i in 1:sim){
  BestPorts[i,4]<-BestPorts[i,1]/BestPorts[i,2]
}
colnames(BestPorts)<-c("ExpectedReturn","StandardDiv","WeightIndex","Sharpe Ratio")
BestPorts<-BestPorts[order(BestPorts[,4]),]
```

SimulateHolding

```
SimulateHolding<-function(WIndex,NWeights, SelectedStocks) {
PortfolioPrices<-matrix(0,nr,1)
#This k refers to an index for weight
#FUNCTION DESIGNED FOR ONLY ONE WEIGHT INDEX
for(k in WIndex){
  for(i in 1:nr){
    for(j in 1:length(SelectedStocks)){
      PortfolioPrices[i,1]<-
PortfolioPrices[i,1]+D.ad[i,SelectedStocks[j]]*NWeights[j,k]
    }
  }
}
#Must alter this line for more than one simulation
PortfolioPrices<-PortfolioPrices/PortfolioPrices[1,1]
return(PortfolioPrices)
}
PortfolioPrices<-SimulateHolding(SortedPortfolio[1,3],NWeights,SelectedStocks)
```

GetProfolioRate

```
GetProfolioRate<-function(PortfolioPrices) {
ProfolioRate<-matrix(NA,nr-1,1)

for(i in 1){
  for(j in 1:(nr-1)){
    ProfolioRate[j,i]<-PortfolioPrices[j+1,i]/PortfolioPrices[j,i]
  }
}
return(ProfolioRate)
}
#Using constant weights of 1/20
ProfolioRate<-GetProfolioRate(PortfolioPrices)
```

QuarterCor

```
QuarterCor<- matrix(NA, nr-61, 1)
#Use correct def of sample corr for Pearson.
for(i in 61:(nr-1)){
  var1<- var(logRate[(i-60):i,1])
  var2<- var(LogProfolioRate[(i-60):i,1])
  varD<- var(logRate[(i-60):i,1]-LogProfolioRate[(i-60):i,1])
  QuarterCor[(i-60),1]<- (var1+var2-varD)/(2*sqrt(var1)*sqrt(var2))
}
```

Quarter Alpha and Beta

```
QuarterBeta<- matrix(NA, nr-61, 1)
i<-0
for(i in 61:(nr-1)){
  QuarterBeta[(i-60),1]<-cov(rate[(i-60):i,1], ProfolioRate[(i-60):i,1])/var(rate[(i-60):i,1])
}
```

```
QuarterAlpha<- matrix(NA, nr-61, 1)
i<-0
for(i in 61:(nr-1)){
  QuarterAlpha[(i-60),1]<- mean(ProfolioRate[(i-60):i,1]) - QuarterBeta[(i-60),1]*(mean(rate[(i-60):i,1]))
}
```

MarNeutral

```
MarNeutral<-matrix(NA,nr-61,1)
for(i in 1:(nr-61)){
  MarNeutral[i,1]<- ProfolioRate[i+60,1]-QuarterBeta[i,1]*rate[i+60,1]
}
NeuPortfolioPrices<-matrix(NA,nr,1)
NeuPortfolioPrices[1:61,1]<-1

for(i in 62:nrow(NeuPortfolioPrices)){
  NeuPortfolioPrices[i,1]<-NeuPortfolioPrices[i-1,1]*(1+MarNeutral[i-61,1])
}
```

StocksNeutral

```
QuarterBetaBasket<- matrix(NA, nr-61, 20)
i<-0
for(j in 1:20){
  for(i in 61:(nr-1)){
    QuarterBetaBasket[(i-60),j]<-cov(rate[(i-60):i,1], rate[(i-60):i,SelectedStocks[j]])/var(rate[(i-60):i,1])
  }
}
```

Alan Agnew Honors Report

```
}  
}  
StockNeutral<-matrix(NA,nr-61,20)  
for(j in 1:20){  
  for(i in 1:(nr-61)){  
    StockNeutral[i,j]<- rate[i+60,SelectedStocks[j]]-QuarterBetaBasket[i,j]*rate[i+60,1]  
  }  
}  
NeuStockPrices<-matrix(NA,nr,20)  
NeuStockPrices[1:61,]<-1  
for(j in 1:20){  
  for(i in 62:nrow(NeuStockPrices)){  
    NeuStockPrices[i,j]<-NeuStockPrices[i-1,j]*(1+StockNeutral[i-61,j])  
  }  
}
```