MA574 Portfolio Valuation and Risk Management

Fall 2017

Professor Marcel Blais

Final Course Report

Khasan Dymov

Alexander Shoop

Contents

1 Abstract

The purpose of this project is to implement the mathematics and techniques learned in the WPI course MA 574: *Portfolio Valuation and Risk Management* to a virtual portfolio of assets. In particular, the topics focused on in this report are Markowitz Portfolio Optimization (also known as Markowitz Portfolio Theory, or MPT), the Capital Asset Pricing Model (CAPM), and Factor Modeling.

2 Introduction

2.1 Background

The inspiration for this project came from Harry Markowitz's Modern Portfolio Theory (MPT) covered in our class textbook *Statistics & Finance - An Introduction* by David Ruppert. This theory, which has served as a foundation for economists and investment advisors for over six decades, was published in the *Journal of Finance* in 1955 [1]. Interestingly enough, when Harry Markowitz presented what would become MPT while defending his dissertation at the University of Chicago School of Economics, economist Milton Friedman argued that portfolio theory was not economics, and thus, Markowitz should not be awarded his Ph.D [1]. Fortunately for Markowitz, the committee approved his work and he was awarded his Ph.D. In 1990, nearly 40 years after publication, Markowitz received the Nobel Prize in Economics for his work, indicating its incredible value in modern economics.

The central idea of Markowitz's Modern Portfolio Theory is actually relatively simple: for an investor constructing a portfolio, the ultimate goal (if we assume investors are rational...) is to maximize expected returns and minimize risk. Mathematically, this translates to maximizing a portfolio's mean return and minimizing the portfolio's return variance. The mathematics in MPT boil down to a non-linear optimization problem, where an investor seeks to find the optimal weights for n risky assets in their portfolio that satisfy their requirements for risk and return.

Our goal was to apply Markowitz's MPT to our own portfolio and analyze the results. From there, we would critique our portfolio performance through various portfolio metrics.

But we also took it further. William Sharpe, a student of Harry Markowitz, built upon Markowitz's MPT and developed his own model, known as the Capital Asset Pricing Model, or CAPM for short [2]. This model "describes the relationship between systematic risk and expected return for assets, particularly stocks" [2]. We will apply this model to our portfolio and data and analyze the results. Additionally, we will perform various factor modeling on our portfolio, to compare to William Sharpe's Capital Asset Pricing Model. In particular, we will focus on the French & Fama Three Factor Model, a model that "expands on the capital asset pricing model (CAPM) by adding size and value factors to the market risk

factor in CAPM" [3]. We will even make adjustments to the Fama and French Model with the addition of several new factors.

2.2 Methodology

As discussed in our prior report, Markowitz Portfolio Optimization - *Phase 1*, we selected 20 stocks based mostly on personal interest and familiarity. Below are the stocks we selected, with their respective tickers:

1. Amazon Inc - AMZN
2. AutoZone Inc - AZO
3. The Boeing Company - BA
4. Cigna Corporation - CI
5. Costco Wholesale Corp. - COST
6. Corbus Pharmaceuticals - CRBP
7. Tableau Software Inc - DATA
8. Ford Motor Company - F
9. Alphabet Inc - GOOG
10. Hubspot Inc - HUBS
11. JPMorgan Chase & Co - JPM
12. Manulife Financial Corp - MFC
13. 3M Company - MMM
14. Vail Resorts Inc - MTN
15. Netflix Inc - NFLX
16. NIKE Inc - NKE
17. Starbucks Corp - SBUX
18. iShares 20+ Year Treasury Bond ETF - TLT
19. Vanguard High Dividend Yield Fund - VYM
20. Waste Management Inc - WM

However, for *Phase 2* of this project, with implementation in Interactive Brokers, we also needed an asset to serve as our "risk-free" asset. As discussed in MA 574, a "risk-free" asset is an asset with some return and zero risk or volatility. This is the purely mathematical definition - in reality, there is no such thing, as some level of risk exists in *every* investment. However, United States government bonds (Treasuries) are typically regarded as the closest proxy to a risk-free asset. As such, we explored different options for U.S. government bonds.

The image below, sourced from TreasuryDirect.gov, shows a snapshot of U.S. Notes of varying maturity and yield that were available at the time we were shopping. For the purposes of our portfolio, and given the equities in our portfolio, we felt that a five year note would be appropriate. As such, we selected the following bond (indicated in red in the image):

1. CUSIP: 9128283C2
2. Yield: 2.000%
3. Maturity: 10/31/2022

Note: The following table displays data for the 20 most recently auctioned securities that have not yet matured, for each security type. If you would like to see data for additional securities, please use our Auction Query.

Bills | CMBs | Notes | Bonds | TIPS | FRNs

Security Term	CUSIP	Reopening	Issue Date	Maturity Date	High Yield	Interest Rate
7-Year	9128283D0	No	10/31/2017	10/31/2024	2.280%	2.250%
5-Year	9128283C2	No	10/31/2017	10/31/2022	2.058%	2.000%
2-Year	912828F62	Yes	10/31/2017	10/31/2019	1.596%	1.500%
3-Year	9128282Z2	No	10/16/2017	10/15/2020	1.657%	1.625%
10-Year	9128282R0	Yes	10/16/2017	08/15/2027	2.346%	2.250%
7-Year	9128282Y5	No	10/02/2017	09/30/2024	2.130%	2.125%
5-Year	9128282W9	No	10/02/2017	09/30/2022	1.911%	1.875%
2-Year	9128282X7	No	10/02/2017	09/30/2019	1.462%	1.375%
10-Year	9128282R0	Yes	09/15/2017	08/15/2027	2.180%	2.250%
3-Year	9128282V1	No	09/15/2017	09/15/2020	1.433%	1.375%
7-Year	9128282U3	No	08/31/2017	08/31/2024	1.941%	1.875%
2-Year	9128282T6	No	08/31/2017	08/31/2019	1.345%	1.250%
5-Year	9128282S8	No	08/31/2017	08/31/2022	1.742%	1.625%
10-Year	9128282R0	No	08/15/2017	08/15/2027	2.250%	2.250%
3-Year	9128282Q2	No	08/15/2017	08/15/2020	1.520%	1.500%
7-Year	9128282N9	No	07/31/2017	07/31/2024	2.126%	2.125%
5-Year	9128282P4	No	07/31/2017	07/31/2022	1.884%	1.875%
2-Year	9128282K5	No	07/31/2017	07/31/2019	1.395%	1.375%
10-Year	912828X88	Yes	07/17/2017	05/15/2027	2.325%	2.375%
3-Year	9128282J8	No	07/17/2017	07/15/2020	1.573%	1.500%

Figure 1: US Treasury Note Selection, sourced from Treasurydirect.gov [4]

Once we selected all of our assets, we constructed a spreadsheet (in Google Sheets) to keep track of our portfolio. This spreadsheet, *MarketPortfolio.xlsx* can be found in the *.zip*-file associated with this project report. In this spreadsheet, we tracked our check-in times, portfolio value (net liquidation value), excess liquidity, equity value, bond value, equity weight, bond weight, prices, sizes of positions, total position value, and individual equity weights. We also kept track of the Markowitz Optimization outputs computed by our Python script *Portfolio Rebalancer.py* (this script can be found in Appendix A), most notably, the expected return and risk of the tangency portfolio, as well as the individual asset weights. Additionally, we tracked the specific adjustments necessary to each asset position to satisfy the tangency portfolio weights.

Our script also computed and plotted the Markowitz Efficient Frontier, tangency portfolio, and Capital Market Line. We saved the plots for the initial portfolio setup, as well as the plots for every weekly rebalance. Below are the plots for the initial portfolio setup (computed based on a 2-year lookback):

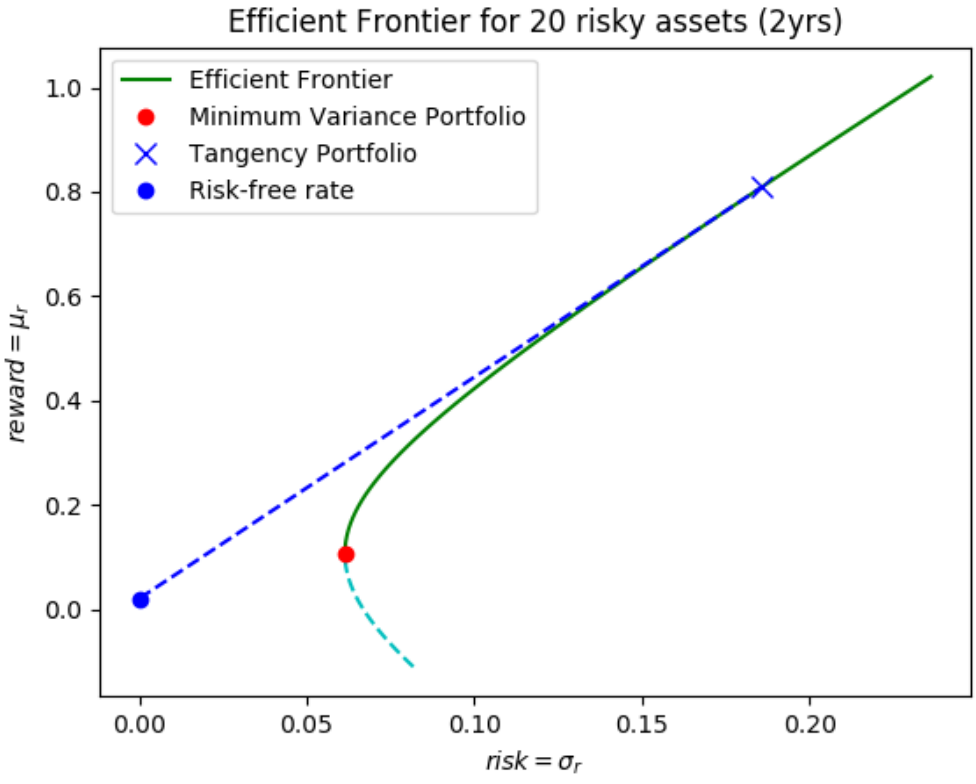

Figure 2

The tangency portfolio, indicated by the "X," had the following computed annual risk and return:

$$\sigma_r = 0.1857112219,\ \mu_r = 0.8089575445$$

This level of annual return is *extremely* high for an equity portfolio, but the corresponding level of volatility, approximately 18.6%, seems appropriate. Our bond, indicated by the blue dot, sits at the point (0,0.02), based on the risk-free assumption and the yield of 2.00%. Our initial portfolio then is equally split between the bond and the equity portfolio, with ≈ $500,000 invested in each. This then moves our overall portfolio down the Capital Market Line (indicated by the blue dashes) exactly equidistant between the bond and the tangency portfolio. Our overall portfolio then, has initial parameters:

$$\sigma_p = 0.09285561093,\ \mu_p = 0.4044787723$$

We also looked at the weights for the initial tangency portfolio - partially because we needed to know them to form our positions in Interactive Brokers, and partially because we were surprised at the values for risk and reward for the tangency portfolio. The bar chart below shows the initial tangency portfolio weights:

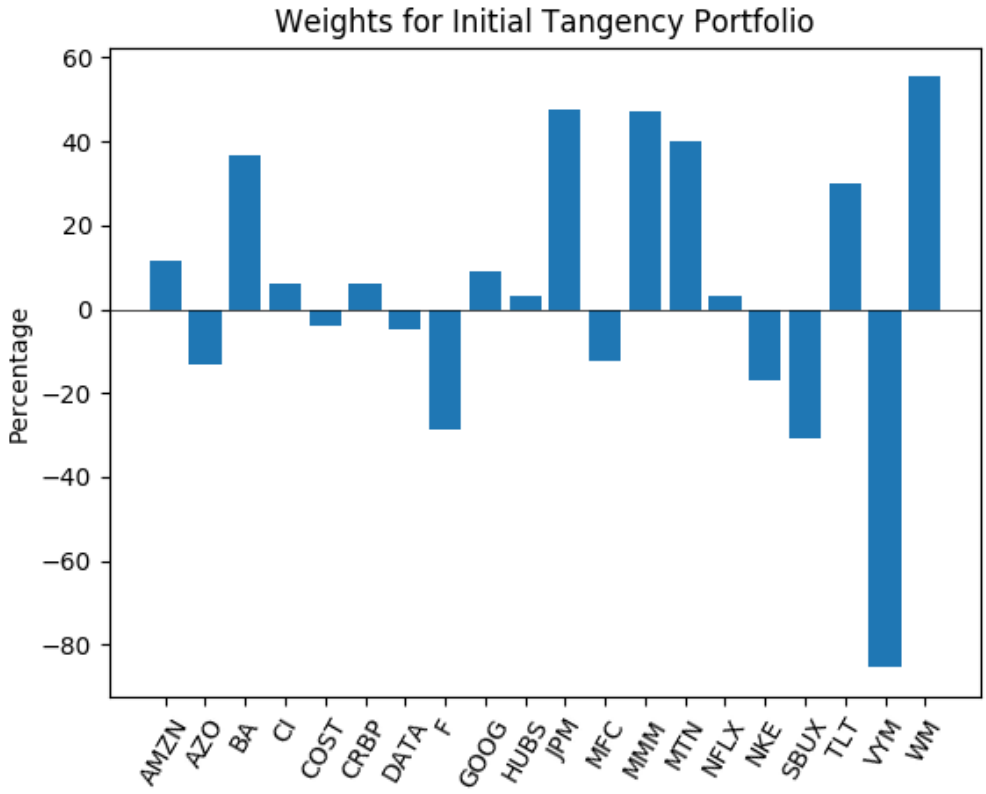

Figure 3

The results are rather interesting. First of all, the extremely high expected return now

makes more sense. Given that the Markowitz optimization is unconstrained, the routine decided to build a fair amount of leverage into the portfolio - as can be seen by some of the very large positions (the sum of the weights is in fact 1, so the results are valid). The most curious position is the extremely large short position in VYM, the Vanguard High Dividend Yield fund, at -85.44%. This fund has very high correlation with the S&P500, so it is interesting that the Markowitz routine chose to essentially short the market in order to place large bets on other assets.

Out of curiosity, we compared these asset weights to the weights for the minimum variance portfolio for these 20 assets, based on the same 2-year lookback. Here we see that the majority of the positions have positive weight, and most are fairly low - with the exception of the positions in TLT and VYM. This comparison does shed some light on the situation - it appears that the Markowitz routine believes a large long position in VYM is useful for reducing portfolio volatility. Thus, forming a large short position in VYM in order to place bets on other assets is the safest approach as the volatility itself is not very volatile. It is interesting however that this was not also the case for TLT - perhaps there is more going on that is immediately evident.

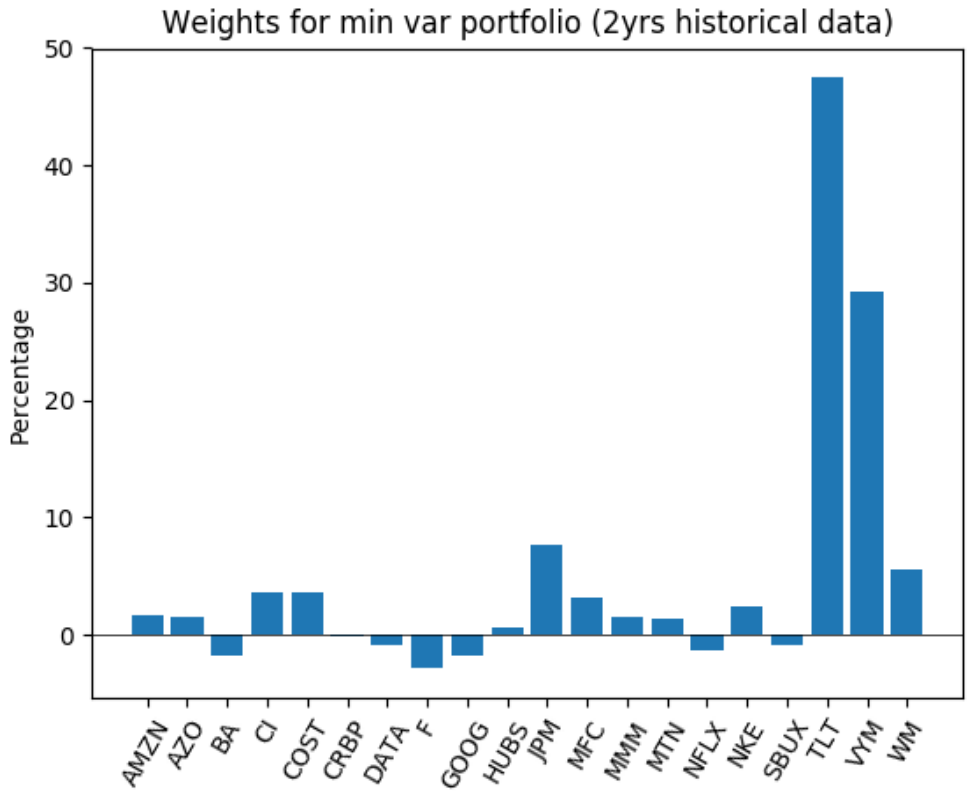

Figure 4

After the initial portfolio setup, rebalancing was done on a weekly basis and the portfolio was tracked in our MarketPortfolio spreadsheet.

The portfolio was closed on December 4th, 2017. Once this was completed, all of the analysis, CAPM modeling, and Factor modeling began.

3 Portfolio Management

3.1 General Process

After our initial portfolio setup, we developed a routine for managing the portfolio with regard to the project guidelines. After every Friday market close, and before market open on Monday, we recorded all the necessary data and made any necessary adjustments. First, all positions were recorded in our *MarketPortfolio.xlsx* file - we tracked the closing price, the size, the net value of the position, and the resulting asset weights in the portfolio. We also recorded the portfolio net liquidation value as well as the excess liquidity, a habit we picked up from the liquidity and margin project from earlier in the course. Once this was complete, we ran our Markowitz optimization script (with a rolled-forward-by-1-week adjusted 2-year window), which can be found in Appendix A: Portfolio Rebalancer.py. This gave us the new weights for the tangency portfolio, as well as the expected return and volatility of the tangency portfolio. From this output, we were able to calculate the necessary adjustments to each position. Lastly, we recorded the adjustment to each position in the sheet, and placed the orders in Interactive Brokers by market-open on Monday.

We found that rebalancing the portfolio every week became very tedious, as we had to take the new weights computed for the tangency portfolio and manually calculate the adjustment to the position in each of our securities (except for the bond position). This was very laborous and had lots of room for human error, so we wrote a small function that computed the adjustments for us (note: the function PortfolioRebalancer is just one part of the script file PortfolioRebalancer.py):

```
1
2  def PortfolioRebalancer(currwts, newwts, currcashval, lastcloseprices, tickers
   ↪ ):
3      #function takes as input current stock weights in portfolio,
4      #new calculated tangency portfolio weights, the current cash value
5      #of the portfolio, latest closing prices for all the stocks, and tickers
6
7      newcashvalue = np.array(np.divide(np.multiply(newwts, currcashval),
   ↪ currwts))
8      cashvalchange = np.array(np.subtract(newcashvalue, currcashval))
9      sharechange = np.array(np.divide(cashvalchange, lastcloseprices))
10
11
12     for x in range(0,20):
13         print("Adjust position in " + tickers[x] + " by " + str(sharechange[x
   ↪ ]) + " shares.")
```

Figure 2 below is a small sample of the output of this function.

```
Adjust position in AMZN by [ 6.19736942] shares.
Adjust position in AZO by [ 3.88447357] shares.
Adjust position in BA by [ 32.37483384] shares.
Adjust position in CI by [ 42.51440625] shares.
```

Figure 5

And so our weekly rebalancing process was significantly streamlined - we simply had to update our Excel sheet, where we were keeping track of our portfolio adjustments, after market close on Friday, adjust the date range in our Python script, and then based on that output, place the orders in Interactive Brokers. As a matter of fact, it is even possible to automate this last step of order placement through the IB Excel API; however, we decided to not implement this due to time constraints (ironic...). Lastly, it should be noted that it is not possible in Interactive Brokers paper trading accounts to trade fractional shares - as such, we adjusted our positions by truncating the values produced by our rebalancing calculations (e.g. A computed adjustment of 6.19736 shares resulted in a transaction of 6 shares).

Our full trade history, sourced from Interactive Brokers' built-in reporting capabilities[5], can be found in Appendix D.

3.2 Portfolio Analysis

3.2.1 General Performance Overview

The screenshot below, sourced from Interactive Brokers, shows our final portfolio value, as well as the realized profit and loss for each asset, after we completely closed our portfolio. The final portfolio value of $988,083 indicates a loss from $1,000,000 of approximately 1.19%. The individual asset returns seemed to be a mix between profits and losses. Our largest loss came from VYM, which as noted before, had a very large short position in the beginning. This fact remained constant throughout the lifetime of the portfolio - the Markowitz optimization routine always wanted to massively short VYM. It is also worth noting that our bond position lost $2,281.

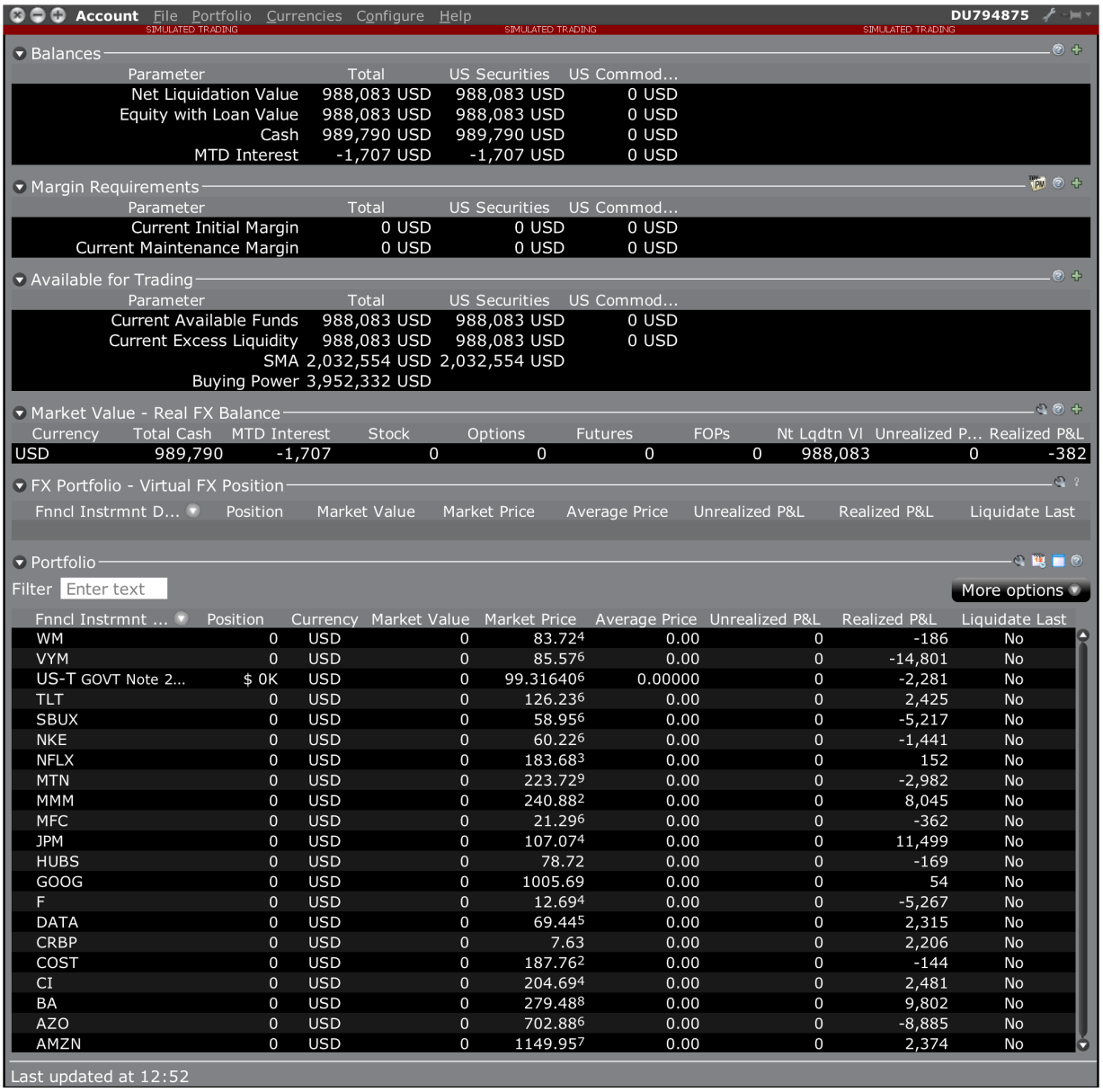

Figure 6: Final Portfolio Statistics & Realized P&L from Interactive Brokers

The plot below shows all of the efficient frontiers, our realized portfolios (mix of stocks and bond), and the capital market lines we computed throughout the five weeks of portfolio management. Opacity indicates the time - less opaque plot elements are further in the past. Based on this, we see that our realized portfolio (the red dot), consistently trended left and downwards from the capital market line each week. In fact, the overall portfolio expected return for our last week of management is nearly 50% down from the initial expected return. We believe this is due to the fact that as the stock portion of the portfolio lost more money than the bond, the resulting weight of the bond value in the portfolio dragged the return down the CML towards the point (0,0.02). It is worth mentioning that, as expected, none

of the red dots sit directly on their respective CMLs. This is because as prices fluctuated during the trading week, the individual asset weights in the portfolio changed, turning the portfolio "inefficient" based upon the definition in the Markowitz model.

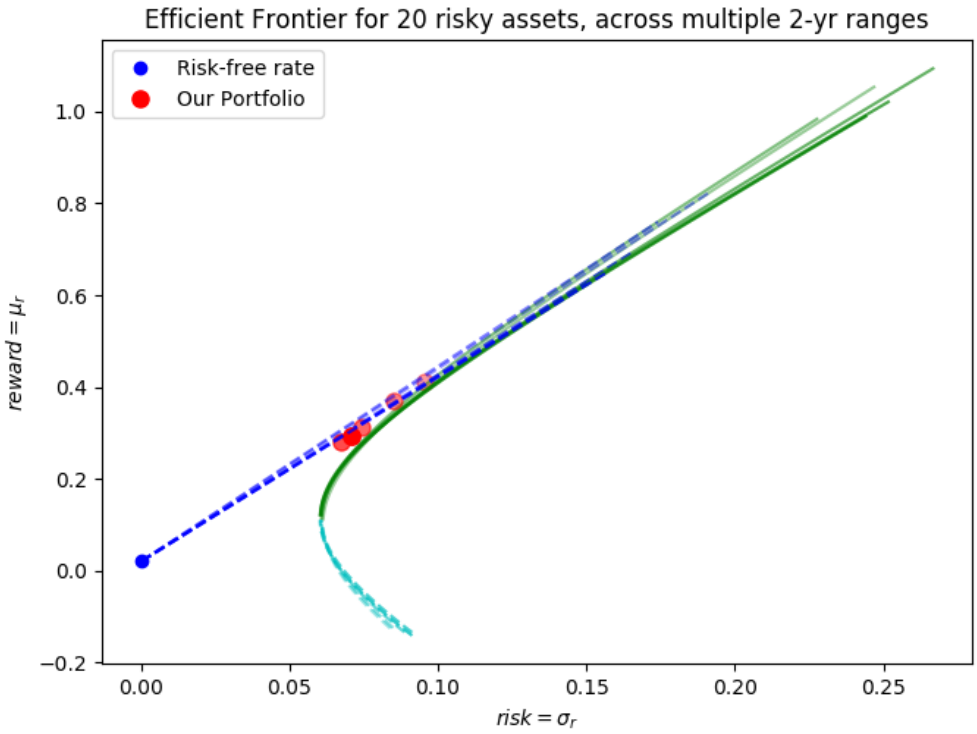

Figure 7

The two plots below indicate the weekly portfolio value and weekly portfolio return.

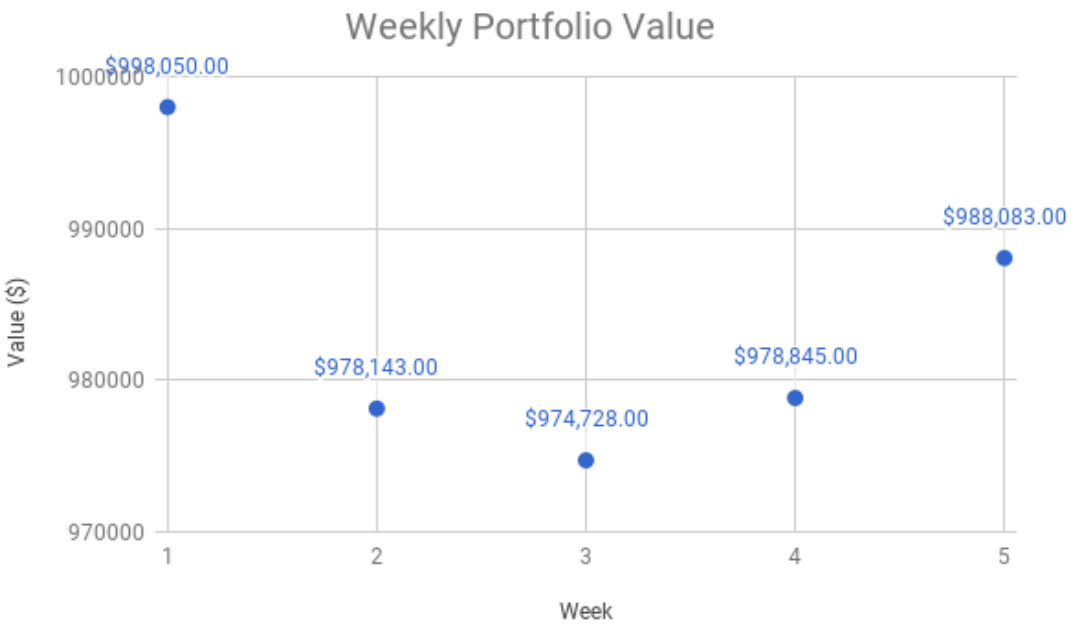

Figure 8

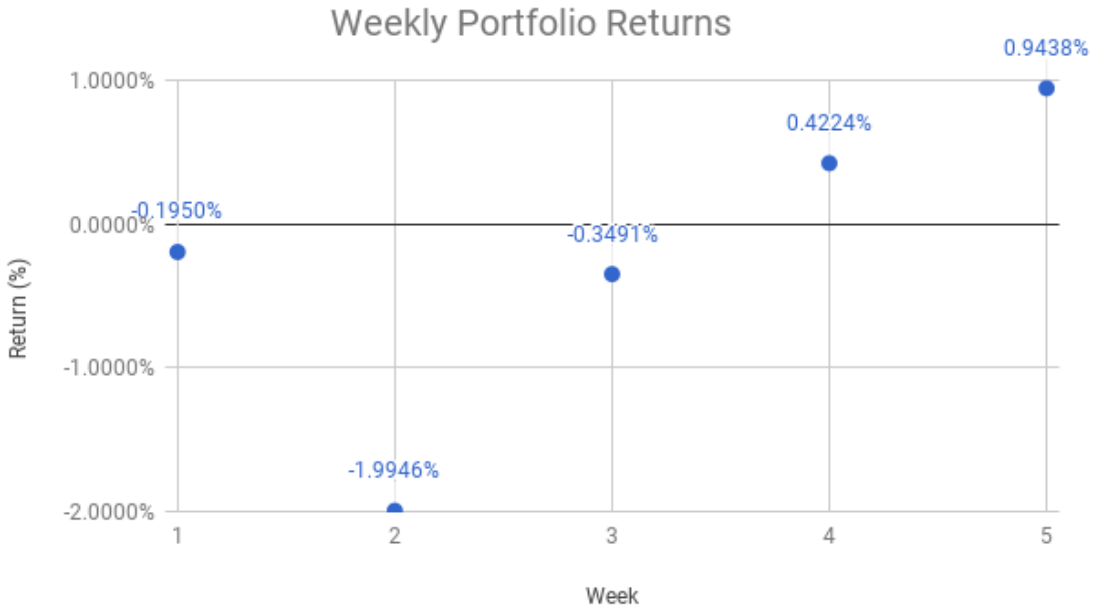

Figure 9

We see from these two figures that our portfolio lost money during the first three weeks, but began to climb substantially after week 3.

3.2.2 Returns

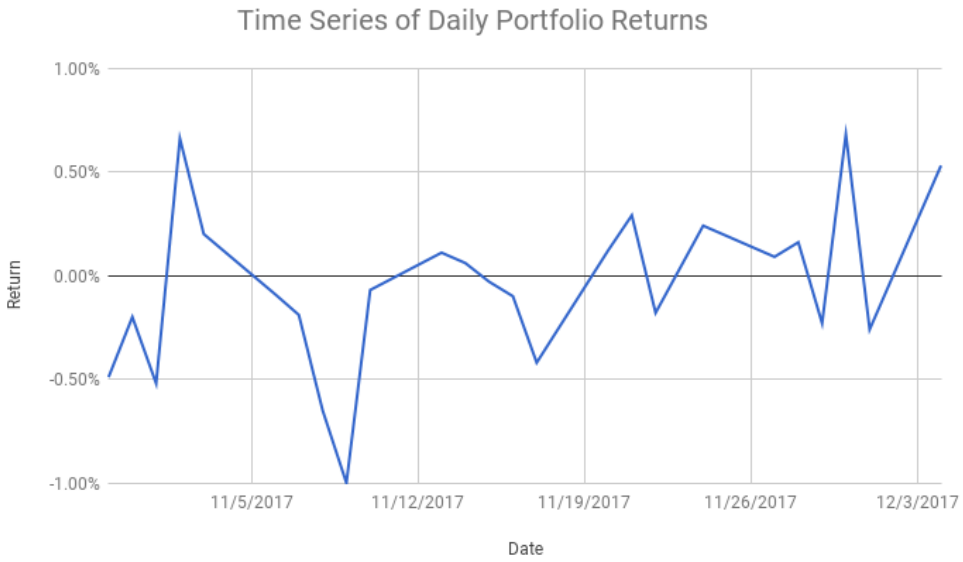

Figure 10

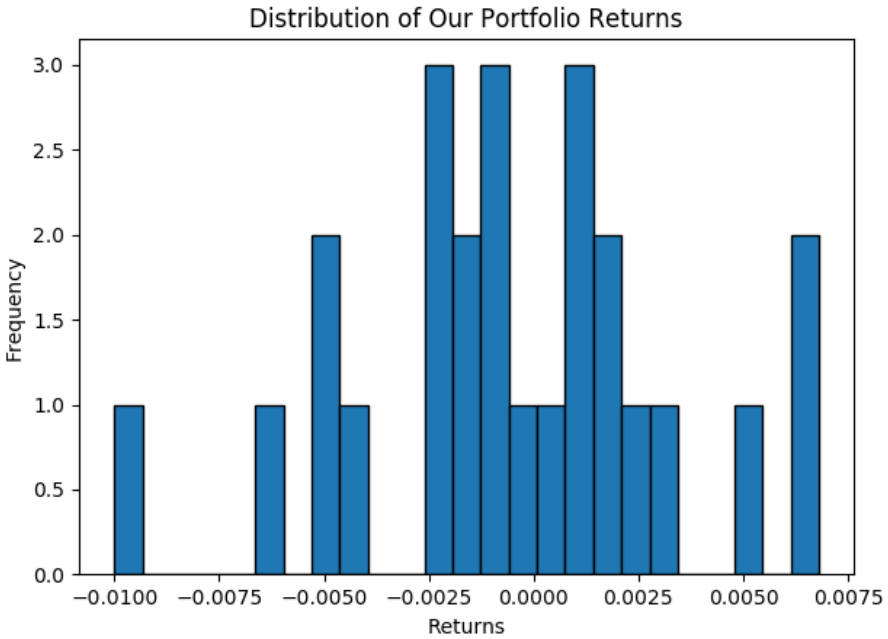

Figure 11: Daily Raw Returns (not percent returns)

The following table provides several summary statistics about our portfolio:

Portfolio Returns Summary	
Days	25
Mean Daily Return	-0.05%
Daily Return SD	0.39%
Annualized Return	-12.11%
Annualized SD	6.25%

Figure 12

Based on our 25 days of data, the annualized return is -12.11% and the annualized volatility is 6.25%. Although this result is disappointing, 25 trading days in general are not sufficient enough of a good sample for making any sort of long-term projections or statements about a portfolio.

3.2.3 Sharpe Ratio

The Sharpe Ratio is one of the most famous measures used in portfolio management. A consequence of William Sharpe's work upon Modern Portfolio Theory, the Sharpe Ratio is "the average return earned in excess of the risk-free rate per unit of volatility or total risk" [6]. The formula for calculating the Sharpe Ratio, as shown in MA 574: Portfolio Valuation & Risk Management, is:

$$RS = \frac{\mu_p - \mu_f}{\sigma_p} \tag{1}$$

From Section 3.2.2, our parameters for μ_p and σ_p are -12.11% and 6.25% respectively. The value for μ_f is the yield on our bond, 2%. With these values, our computed Sharpe Ratio then, is

$$RS = -2.256915737 \tag{2}$$

3.2.4 Treynor Ratio

$$RT = \frac{\mu_p - \mu_f}{\beta_p} \quad , \quad \beta_p = \frac{\sigma_{pM}}{\sigma_M^2} \tag{3}$$

As for the Treynor Ratio, our parameters for μ_p and μ_f are -12.11% and 2%, respectively. For our portfolio β_p, we utilized our CAPM script and computed a value of 1.050789, indicating that our portfolio is slightly more volatile than the market. With these values, our computed Treynor Ratio is:

$$RT = -0.1342375092 \tag{4}$$

3.2.5 Information Ratio - S&P500

The Information Ratio seeks to make a comparison between our portfolio and some other benchmark. For our purposes here, we utilized the S&P500 as our benchmark. As defined in MA 574, the Information Ratio RI can be calculated as such:

$$RI = \frac{\mu_p - \mu_B}{\sigma_{R_p - R_B}} \quad , \quad B = Benchmark \tag{5}$$

As before, our value of μ_p is -12.11%, the annualized return of our portfolio based on daily returns from October 30^{th}, 2017 to December 4^{th}, 2017. In order to remain consistent in the use of this formula, we utilized the daily return data for the S&P500 index from this date range - data obtained from Yahoo Finance [7]. From this data, we calculated the annualized return on the S&P500, and the annualized standard deviation of the difference of returns between our portfolio and the S&P500; $\mu_B = 0.2527948234$ and $\sigma_{R_p - R_B} = 0.06583540257$ respectively. With these values, we calculated our Information Ratio to be:

$$RI = -5.678557537 \tag{6}$$

Based on the heuristic rule provided in lecture that an RI > 1 is an indicator of success, our calculated value shows that we are quite far from success. However, our result may simply be a consequence of the amount of data used.

3.2.6 Sortino Ratio

$$\text{Sortino Ratio} = \frac{\mu_p - r_o}{\sigma_{r_o\text{-}}} \tag{7}$$

where r_o is a target or minimum accepted return, and $\sigma_{r_o\text{-}}$ is the sample downside semistandard deviation. For our minimum accepted return, we used 2%, the yield on our U.S. Note. As before, μ_p is -12.11%. In order to compute the downside semistandard deviation, we computed the daily return equivalent to our minimum required annual return, and then compared the resulting value of 0.000079213646 (0.00792113646%) to all of our daily portfolio returns. The daily returns above this value were removed, and the daily standard deviation of the remaining values was computed, and then annualized. The result was $\sigma_{r_o\text{-}} = 4.297855969\%$, which gave a Sortino Ratio of:

$$\text{Sortino Ratio} = -3.281992208 \tag{8}$$

As before, as this ratio is negative, it indicates that our portfolio has performed poorly. The data and calculations for the Sharpe Ratio, Treynor Ratio, Information Ratio and Sortino Ratio can be found in the Excel file *PortfolioReturns.xlsx*, located in the *.zip*-file associated with this project report.

3.2.7 Maximum Drawdown

Definition 3.1. *Maximum Drawdown:* The maximum drawdown is the max equity retractment over a holding period; usually quoted as a percentage.

Definition 3.2. The maximum drawdown over [0,T] is

$$MDD(T) = \max_{0 \le t \le T} [M(t) - V(t)] \tag{9}$$

where M(t) indicates the "high watermark" at time t, and V(t) indicates the portfolio value at time t.

The figure below, provided during a MA 574 lecture by Professor Marcel Blais, illustrates this concept very well:

Figure 13: Maximum Drawdown Concept. *Source: Professor Marcel Blais*

We will evaluate our maximum drawdown on a weekly basis. The following plot (from before) shows our weekly portfolio values:

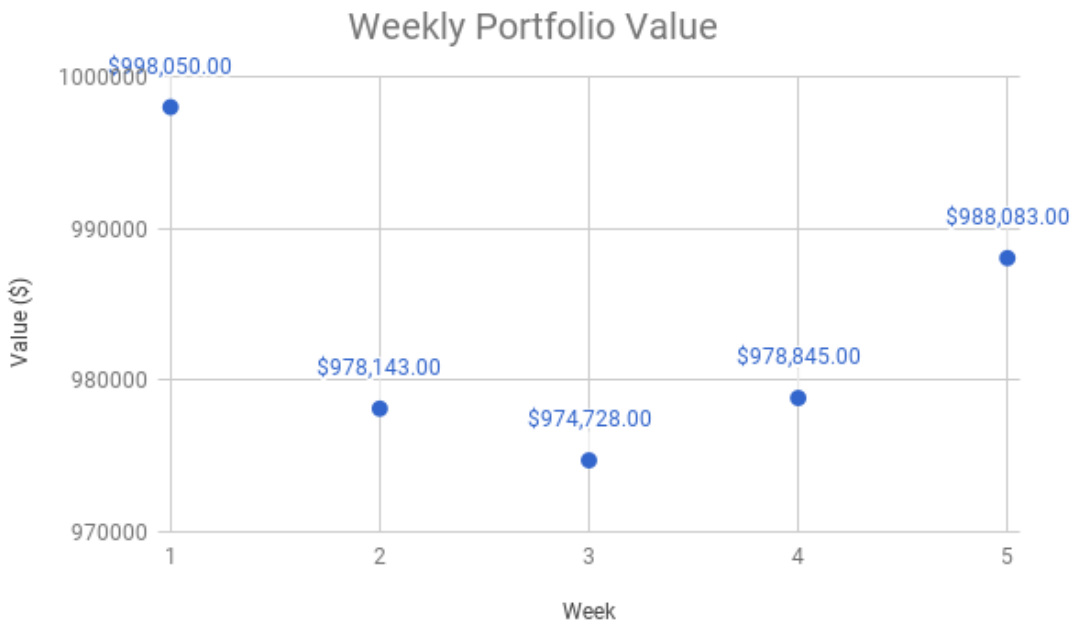

Figure 14: Time series of our weekly portfolio value

Based on this graphic, we see that our lowest portfolio value occured in Week 3 - \$974,728.00. Our "high watermark" is the highest value our portfolio achieved, which is rather unforunately our initial portfolio value of \$1,000,000. As such, our maximum drawdown is

$$MDD(T) = \$1,000,000 - \$974,728.00 = \$25,272.00 \tag{10}$$

which is equivalent to a 2.53% drop in portfolio value.

3.2.8 Portfolio Alpha & Beta

From the traditional CAPM model that we constructed (discussed in later sections), we computed the following portfolio alpha and beta:

Table 1: Portfolio Alpha and Beta

Alpha	Beta
-0.012	1.050789

The result for alpha is consistent with our overall level of return, and the value for beta is logical.

3.2.9 Comparison to Industry Benchmarks

Based on the performance of our portfolio over the 25 trading days, it is not particularly valuable to compare it to any standard industry benchmarks as the result will always be the same - our loss of -12.11% is indicative of poor performance. However, given a much longer time span, it may be interesting to compare the performance of the portfolio to various benchmarks such as the S&P500, the Vanguard Total Stock Market Index (VTSMX), and the Russell 2000. Benchmarking is a widely debated topic, and the methods and approaches used can often vary investor-to-investor.

3.2.10 Margin Call Analysis

Throughout the lifetime of our portfolio, we did not receive any margin calls. Our net liquidation value remained high enough at all times to satisfy the margin requirements within Interactive Brokers. As such, no adjustments needed to be made.

3.2.11 Leverage Analysis

Beyond computing ratios, we analyzed the leverage of our portfolio throughout its lifetime. Leverage can be a useful and informative metric about a portfolio's risk and margin requirements. It can also be utilized for "knowing the amount of debt held by a company in evaluating whether it can pay its debts off" [8]. For the purposes of this project, we considered the leverage ratio L to be:

$$L = \frac{\text{Portfolio Market Value}}{\text{Net Liquidation Value}} \tag{11}$$

We computed the leverage ratio of our portfolio on a daily basis throughout its lifespan, utilizing the values provided through the Interactive Brokers portfolio management features. The plot below shows the daily leverage ratio of our portfolio:

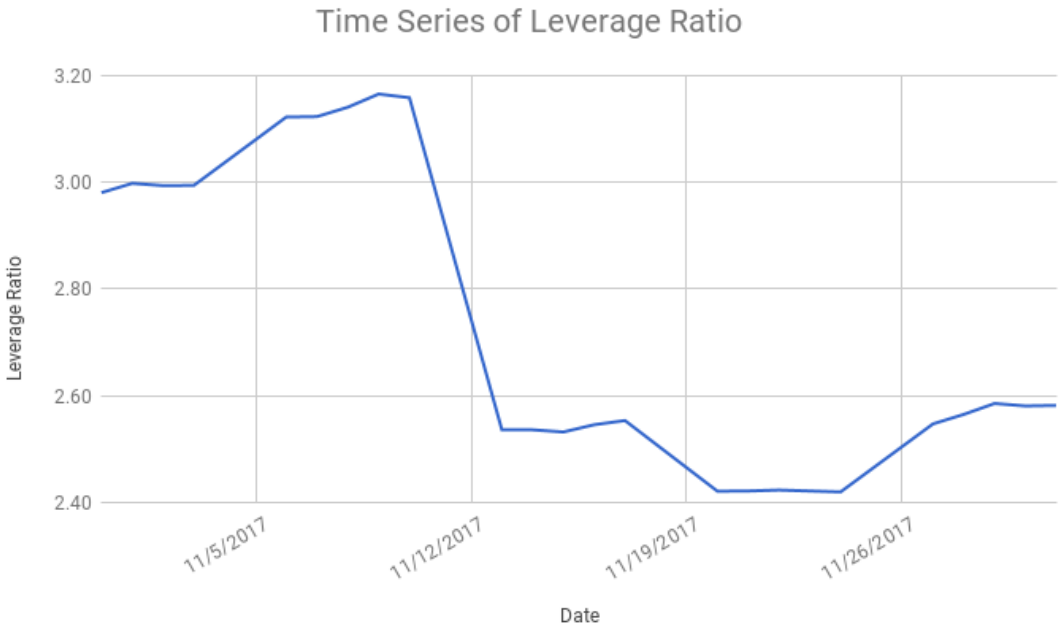

Figure 15: Time series of daily portfolio leverage ratio

This plot indicates that our portfolio is fairly highly leveraged, with a high of approximately 3.2x and a low of approximately 2.4x. If we compare this plot to our initial tangency portfolio weights...

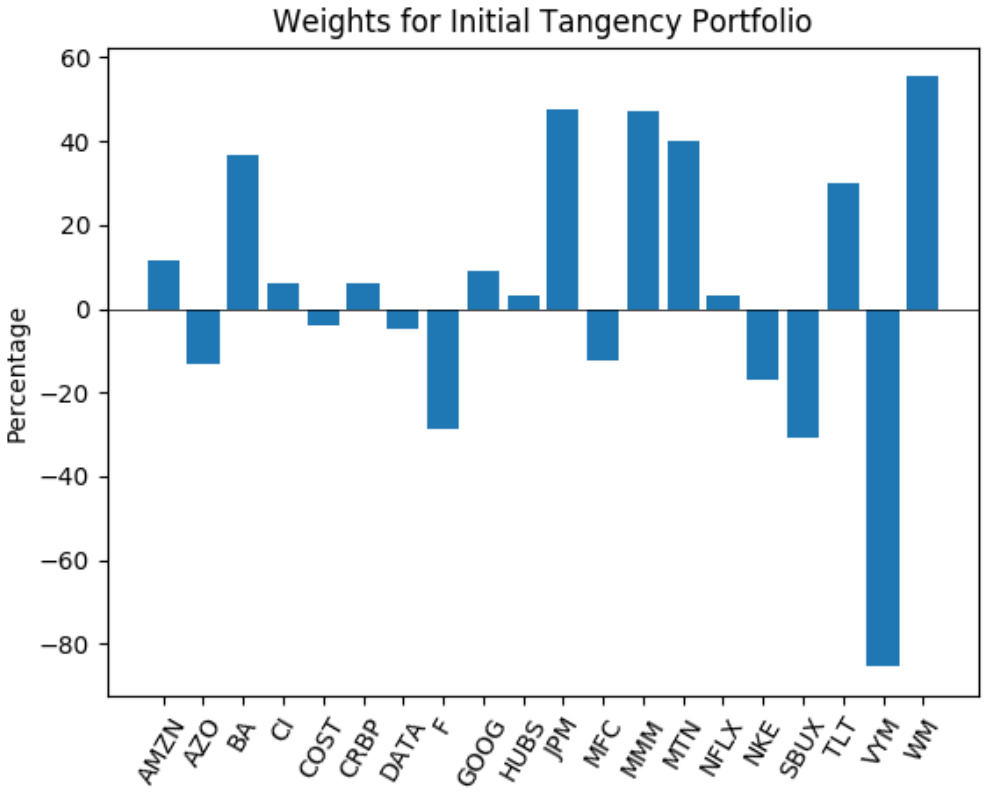

Figure 16

...we see that these results are logical. Although the weights in our assets did change from week to week, there was still nonetheless a high amount of leverage in our portfolio due to some very large long and short positions every week. Such a high leverage ratio makes our portfolio more risky and raises our margin requirements significantly. Although we never received any margin calls from Interactive Brokers (likely due to our large bond position serving as a hedge to our equity positions), such a high leverage ratio should be a cause for concern.

If we were to do this project again, we would like to form our portfolios using constrained Markowitz optimizations that significantly limit the size of short positions. This would help bring the "Portfolio Market Value" at any point much closer to the portfolio Net Liquidation Value, bringing the leverage ratio much closer to 1. This in turn would reduce our margin requirements and may also help reduce our losses as a result of small but adverse movements in any of our assets. From Section 3.2.1, we see that our position in VYM, which was consistently a *very* large short position, lost us a disproportionally large amount of money relative to all of our other positions. Reducing the portfolio leverage is one possible step to mitigate this issue in the future.

4 Capital Asset Pricing Model

The capital asset pricing model (aka. CAPM) is one of the most well-known theoretical models in the financial analysis world that is best used for "passive" investing known as indexing[9]. We aimed to calculate alternative β coefficients for the CAPM. For this project, we used the same 20 risky assets as in our Markowitz Portfolio Optimization project. We calculated the β coefficients from the traditional approach using historical returns as well as the alternative methods, such as least trimmed squares (LTS), constant β shrinkage estimation, and exponentially weighted moving average (EWMA). We decided to use a 5-year date-range (between 1/1/2012 and 1/1/2017) for the purposes of presenting these alternative β coefficients.

4.1 Traditional Approach

The traditional CAPM approach involves using the historical returns for each security and the risk-free asset (aka. S&P500 index for our case). We used the following simple asset returns formula to determine our asset returns:

$$R_{j,t} = \frac{P_{j,t}}{P_{j,t-1}} - 1 \tag{12}$$

Where $P_{j,t}$ is the monthly adjusted close price for the jth security. For our results and analysis, we primarily used the "excess" returns, which is the difference between the security/risk-free asset return and the risk-free rate μ_f ($\mu_f = 0.02$, from the previous section 2.2).

$$R^*_{j,t} = R_{j,t} - \mu_f, \quad R^*_{M,t} = R_{M,t} - \mu_f \tag{13}$$

After gathering our respective excess returns, we utilized the built-in Python ordinary least squares (OLS) estimator `statsmodels.api.OLS()` for the following CAPM regression model:

$$R^*_{j,t} = \beta_j(R^*_{M,t}) + \epsilon_{j,t} \tag{14}$$

The table above contains each of our assets and their respective CAPM (traditional method) β coefficients. Furthermore, our portfolio β was determined to be 1.050789.

At a glance, by definition from our MA 574 class, any β that is > 1 is aggressive. Thus, one can say for eg. that the securities JPM, HUBS, and CRBP are highly aggressive to the market. Therefore, since our portfolio $\beta = 1.050789$, our portfolio overall is indeed aggressive; this was anticipated especially considering our relatively volatile assets such as AMZN, HUBS, and NFLX.

It is interesting to note and point out the "negative" β coefficients in our results above, namely for NKE and TLT. By definition, when a stock has a negative β, this means it is inverse to the market. We looked at the segmented time analysis for NKE; as it turns out,

Table 2: Ordered Stock Betas

Stock	Beta
NKE	-0.56412
TLT	-0.456008
MTN	0.444818
CI	0.508103
AZO	0.60286
WM	0.70178
SBUX	0.774694
DATA	0.778643
VYM	0.90504
COST	0.913119
GOOG	0.979013
F	1.039445
MMM	1.04565
BA	1.08145
NFLX	1.268151
AMZN	1.399145
MFC	1.468536
JPM	1.505132
HUBS	2.162088
CRBP	2.442068

we can observe that at around July 2015, NKE returns not only jumped but some of the the peaks coincide with the troughs for the S&P 500 index returns, ie. opposite to the market.

Figure 17: NKE & S&P500 Returns of Last 5 Years. *Source: Yahoo Finance* [10]

4.2 Least Trimmed Squares

For the least trimmed squares (LTS) approach, this involved "trimming" the values that are smaller the α-quantile and the values that are greater than the $(1-\alpha)-$quantile. Essentially, the asset returns that are seen before the α-quantile and those after the $(1-\alpha)$-quantile are set to the value at the α-quantile or $(1-\alpha)$-quantile. This can also be called "feeding in the residuals." The list of estimated β coefficients can be found in the comparison table in section 4.5, but we chose VYM as our representative asset to demonstrate the original-to-trimmed returns distribution. We chose α to be 0.05. The original returns distribution as well as trimmed returns distribution for VYM is as follows:

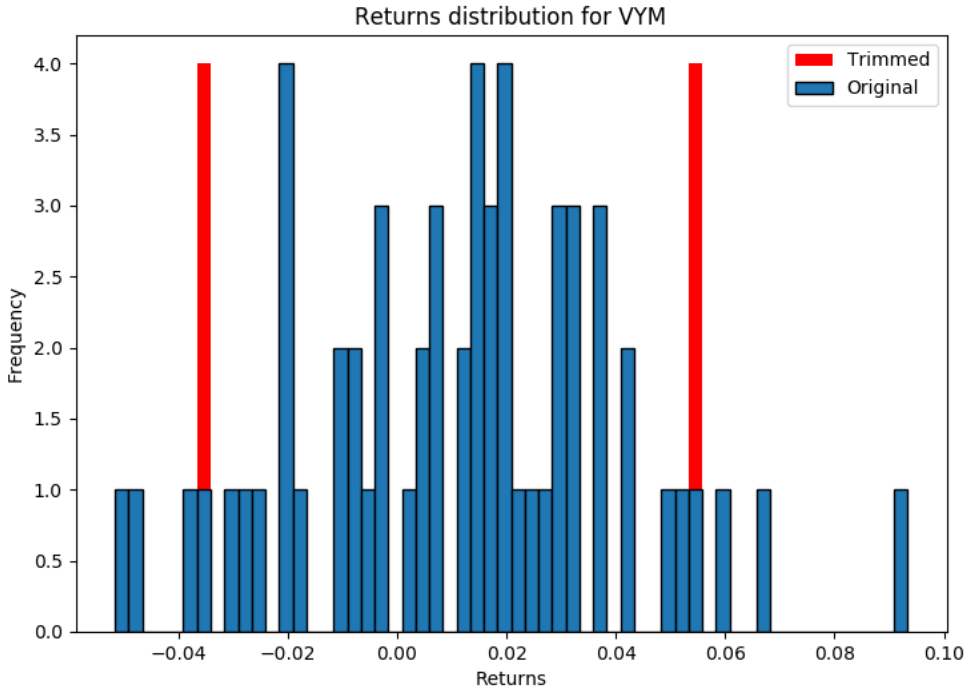

Figure 18: Returns Distribution (original & trimmed) of VYM

4.3 Constant β Shrinkage Estimation

This shrinkage approach is useful because our estimated $\hat{\beta}$ coefficients could be far from the true β's. This method involves using the previously estimated $\hat{\beta}_{i,OLS}$ coefficients from the traditional CAPM method, calculating the average $\bar{\beta}$, and then using a shrinkage parameter. We chose $\alpha_i = \frac{2}{3}$ because it is common in practice. We used the James-Stein estimator[11] formula, detailed below. Our results can be found in section 4.5:

$$\hat{\beta}_{i,JS} = \bar{\beta} + \alpha_i(\hat{\beta}_{i,OLS} - \bar{\beta}) \tag{15}$$

4.4 Exponentially Weighted Moving Average (EWMA)

This EWMA method is an extremely useful technique because it puts more emphasis (ie, weight) to more recent data points versus older data. One reason why older data may not be as reliable is due to mergers and acquisitions; there should be more leverage on newer data since it is more relevant to current time. EWMA method uses a smoothing parameter, namely $\lambda \in (0, 1)$. For demonstration purposes, we chose $\lambda = 0.94$ (because it is one of the standard values for monthly data) and $\lambda = 0.8$ (to have a value less than our first λ). The larger the λ, the more weight is put to the past. In fact, if $\lambda = 1$, then we just have the CAPM traditional model. We were able to calculate the new β coefficients via:

$$\hat{\beta}_{i,t} = \frac{\sigma_{i,M,t}}{\sigma^2_{M,t}} \tag{16}$$

Where the numerator is the ith asset covariance estimate and the denominator is the market variance estimate. Their iterative formulas are:

$$\sigma_{i,M,t} = (1-\lambda)R^*_{i,t-1}R^*_{M,t-1} + \lambda\sigma_{i,M,t-1}, \quad \sigma^2_{M,t} = (1-\lambda)R^2_{M,t} + \lambda\sigma^2_{M,t-1} \tag{17}$$

4.5 Comparison and Analysis

Finally, after accomplishing all the above alternative methods along with the traditional CAPM approach for calculating the various β coefficients, we have our final β comparison table. We performed a conditional color format, where large β's are colored green and small β's are colored red:

Stocks	Beta CAPM	Beta LTS	Shrinkage Beta	EWMA (Lambda = 0.94)	EWMA (Lambda = 0.8)
AMZN	1.399145	1.194054	1.249423	1.303481	0.780364
AZO	0.60286	0.569738	0.718567	0.645602	0.625685
BA	1.08145	0.926644	1.037627	1.005153	0.77169
CI	0.508103	0.51499	0.655395	0.856552	0.865074
COST	0.913119	0.839198	0.925406	1.103	1.151821
CRBP	2.442068	2.498855	1.944706	2.409043	2.647842
DATA	0.778643	0.81902	0.835755	-0.137976	-1.677926
F	1.039445	0.946539	1.009623	0.573273	0.232377
GOOG	0.979013	0.824539	0.969335	1.192994	1.233285
HUBS	2.162088	1.888545	1.758052	2.230947	2.211438
JPM	1.505132	1.215495	1.320081	2.107107	2.809843
MFC	1.468536	1.163366	1.295684	1.840141	2.283879
MMM	1.04565	0.951167	1.01376	1.104079	0.841187
MTN	0.444818	0.401589	0.613205	0.643012	0.182447
NFLX	1.268151	0.92305	1.162094	2.175395	1.479845
NKE	-0.56412	0.435616	-0.05942	0.186329	0.109066
SBUX	0.774694	0.706595	0.833122	1.00854	1.042099
TLT	-0.456008	-0.353081	0.012655	-0.641323	-0.974508
VYM	0.90504	0.795745	0.92002	0.891551	0.793794
WM	0.70178	0.684433	0.784514	0.728559	0.483231

Figure 19: All alternative β coefficients.

We can see, at least across the traditional, LTS, and shrinkage methods that the estimated β coefficients are relatively the same. We also see that CRBP is the most aggressive (with a $\beta > 2.0$) and that NKE and TLT are the most inverse to the market.
More importantly though, we see the differing β coefficients for the exponentially weighted

moving average method, especially for the DATA asset. Why might this be? The answer becomes clear when one looks at the time-series price figure for DATA:

Figure 20: 5-year time-series of adjusted close price for DATA [12].

The EWMA approach very much took into consideration the July 2015 - February 2016 extreme drop that happened with Tableau Software (DATA). Some reasons for this drop were because of the over-crowded data analytics competition, namely Microsoft, and its "own sales execution"[13]. Thankfully, EWMA (especially with $\lambda = 0.80$, thus more consideration into the past) took this into account and thus assigned a negative β coefficient.
If we were hired as financial analysts for a firm, we would choose the exponentially weighted moving average approach thanks to its usefulness when considering more relevant new data.

5 Factor Modeling

For the factor model portion of our project, we still used the same exact 20 risky assets from our Markowitz Portfolio Project and we used the S&P500 index as the market portfolio. We implemented the traditional CAPM method, the French-and-Fama model, and two

additional factors.

Our first factor (F_1) was the Momentum Factor for Global ex USA portfolio from Ken French's website. This factor is the overall stock return portfolio of major country stock returns MINUS USA's stocks. In more detail, "the global portfolios use global size breaks, but [we] use the momentum breakpoints for the four regions to allocate the stocks of these regions." [14] We wanted to choose this factor because it appeared to give the overall global stock return trend, but without USA's global portfolio in consideration. This way we could see if our assets followed the non-USA global portfolio trend.

Our second factor (F_2) was the RSTAMOM index retrieved from Bloomberg Terminal which measures the adjusted total monthly % change in "retail and food services sales in USA" [15]. This factor was specific to USA, and we so believed that the retail & food sales would have some significance to our 20 risky assets which involve consumer sales.

For this section of our project, we used a 10-month holding period, where any date past 1/1/2017 was considered "new." In other words, our historical analysis was between 3/1/2016 and 1/1/2017.

We first implemented the traditional CAPM historical returns approach to determine our β coefficients, expected asset returns, variance of asset returns, and covariance matrix between all 20 of our assets. We performed these steps for each of our factor models, including the extensions with our chosen F_1 and F_2 factors. The results for each model follow. For our visuals, we utilized a red-green color shading scheme - green indicates positive values, and the more positive the value, the more green it becomes. Red indicates negative values, and the more negative the value, the more red it becomes.

5.1 CAPM

stocks	Beta	Expected Asset Returns	Asset Returns Variance
AMZN	0.3882996761	0.05862408491	0.04653771858
AZO	1.368458362	0.1561202576	0.02605462627
BA	-0.3334384091	-0.01316704653	0.02570371607
CI	2.069167984	0.2258196924	0.04783724458
COST	1.726060103	0.1916908256	0.03148173099
CRBP	1.000302784	0.1194999018	1.019769336
DATA	2.381244777	0.2568619036	0.1012313381
F	1.100985359	0.1295147759	0.01197206546
GOOG	1.226849848	0.1420344894	0.03890423958
HUBS	3.930907599	0.4110065298	0.1794704381
JPM	1.529645798	0.1721535371	0.04238492542
MFC	1.688330885	0.1879379084	0.06504423972
MMM	1.275724638	0.1468960542	0.0139290697
MTN	-0.2050561288	-0.0003968888414	0.01862031354
NFLX	-2.75434536	-0.253974138	0.1826653003
NKE	0.8264901644	0.1022107981	0.02235244801
SBUX	1.36008683	0.1552875432	0.02751813557
TLT	-0.1938543273	0.0007173519343	0.02090183911
VYM	0.8717926501	0.1067170266	0.003691830407
WM	0.2226501202	0.04214695936	0.01544017372

Figure 21

	AMZN	AZO	BA	CI	COST	CRBP	DATA	F	GOOG	HUBS	JPM	MFC	MMM	MTN	NFLX	NKE	SBUX	TLT	VYM	WM
AMZN	0.00059	0.00206	-0.00050	0.00312	0.00260	0.00151	0.00359	0.00166	0.00185	0.00592	0.00230	0.00254	0.00192	-0.00031	-0.00415	0.00125	0.00205	-0.00029	0.00131	0.00034
AZO	0.00206	0.00727	-0.00177	0.01099	0.00917	0.00531	0.01265	0.00585	0.00651	0.02087	0.00812	0.00897	0.00677	-0.00109	-0.01463	0.00439	0.00722	-0.00103	0.00463	0.00118
BA	-0.00050	-0.00177	0.00043	-0.00268	-0.00223	-0.00129	-0.00308	-0.00142	-0.00159	-0.00509	-0.00198	-0.00218	-0.00165	0.00027	0.00356	-0.00107	-0.00176	0.00025	-0.00113	-0.00029
CI	0.00312	0.01099	-0.00268	0.01661	0.01386	0.00803	0.01912	0.00884	0.00985	0.03156	0.01228	0.01356	0.01024	-0.00165	-0.02212	0.00664	0.01092	-0.00156	0.00700	0.00179
COST	0.00260	0.00917	-0.00223	0.01386	0.01156	0.00670	0.01595	0.00737	0.00822	0.02633	0.01025	0.01131	0.00854	-0.00137	-0.01845	0.00554	0.00911	-0.00130	0.00584	0.00149
CRBP	0.00151	0.00531	-0.00129	0.00803	0.00670	0.00388	0.00924	0.00427	0.00476	0.01526	0.00594	0.00655	0.00495	-0.00080	-0.01069	0.00321	0.00528	-0.00075	0.00338	0.00086
DATA	0.00359	0.01265	-0.00308	0.01912	0.01595	0.00924	0.02200	0.01017	0.01134	0.03632	0.01413	0.01560	0.01179	-0.00189	-0.02545	0.00764	0.01257	-0.00179	0.00806	0.00206
F	0.00166	0.00585	-0.00142	0.00884	0.00737	0.00427	0.01017	0.00470	0.00524	0.01679	0.00654	0.00721	0.00545	-0.00088	-0.01177	0.00353	0.00581	-0.00083	0.00372	0.00095
GOOG	0.00185	0.00651	-0.00159	0.00985	0.00822	0.00476	0.01134	0.00524	0.00584	0.01871	0.00728	0.00804	0.00607	-0.00098	-0.01311	0.00393	0.00648	-0.00092	0.00415	0.00106
HUBS	0.00592	0.02087	-0.00509	0.03156	0.02633	0.01526	0.03632	0.01679	0.01871	0.05996	0.02333	0.02575	0.01946	-0.00313	-0.04201	0.01261	0.02075	-0.00296	0.01330	0.00340
JPM	0.00230	0.00812	-0.00198	0.01228	0.01025	0.00594	0.01413	0.00654	0.00728	0.02333	0.00908	0.01002	0.00757	-0.00122	-0.01635	0.00491	0.00807	-0.00115	0.00517	0.00132
MFC	0.00254	0.00897	-0.00218	0.01356	0.01131	0.00655	0.01560	0.00721	0.00804	0.02575	0.01002	0.01106	0.00836	-0.00134	-0.01805	0.00541	0.00891	-0.00127	0.00571	0.00146
MMM	0.00192	0.00677	-0.00165	0.01024	0.00854	0.00495	0.01179	0.00545	0.00607	0.01946	0.00757	0.00836	0.00632	-0.00102	-0.01364	0.00409	0.00673	-0.00096	0.00432	0.00110
MTN	-0.00031	-0.00109	0.00027	-0.00165	-0.00137	-0.00080	-0.00189	-0.00088	-0.00098	-0.00313	-0.00122	-0.00134	-0.00102	0.00016	0.00219	-0.00066	-0.00108	0.00015	-0.00069	-0.00018
NFLX	-0.00415	-0.01463	0.00356	-0.02212	-0.01845	-0.01069	-0.02545	-0.01177	-0.01311	-0.04201	-0.01635	-0.01805	-0.01364	0.00219	0.02944	-0.00883	-0.01454	0.00207	-0.00932	-0.00238
NKE	0.00125	0.00439	-0.00107	0.00664	0.00554	0.00321	0.00764	0.00353	0.00393	0.01261	0.00491	0.00541	0.00409	-0.00066	-0.00883	0.00265	0.00436	-0.00062	0.00280	0.00071
SBUX	0.00205	0.00722	-0.00176	0.01092	0.00911	0.00528	0.01257	0.00581	0.00648	0.02075	0.00807	0.00891	0.00673	-0.00108	-0.01454	0.00436	0.00718	-0.00102	0.00460	0.00118
TLT	-0.00029	-0.00103	0.00025	-0.00156	-0.00130	-0.00075	-0.00179	-0.00083	-0.00092	-0.00296	-0.00115	-0.00127	-0.00096	0.00015	0.00207	-0.00062	-0.00102	0.00015	-0.00066	-0.00017
VYM	0.00131	0.00463	-0.00113	0.00700	0.00584	0.00338	0.00806	0.00372	0.00415	0.01330	0.00517	0.00571	0.00432	-0.00069	-0.00932	0.00280	0.00460	-0.00066	0.00295	0.00075
WM	0.00034	0.00118	-0.00029	0.00179	0.00149	0.00086	0.00206	0.00095	0.00106	0.00340	0.00132	0.00146	0.00110	-0.00018	-0.00238	0.00071	0.00118	-0.00017	0.00075	0.00019

Figure 22

5.2 French & Fama

	Beta Mkt	Beta SMB	Beta HML	Expected Asset Returns	Asset Returns Variance
AMZN	0.1698462869	0.003781682778	-0.01098111501	0.01964142025	0.003878143215
AZO	0.8648806327	0.005091822115	-0.003230150577	0.09956736621	0.002171218856
BA	-0.3273569166	-0.00110307105	0.00638308311	-0.0001603163643	0.002141976339
CI	-0.697907128	0.02408340261	0.005976476186	-0.01289901187	0.003986437049
COST	3.044086261	-0.01160646563	-0.002024806957	0.2902523382	0.002623477583
CRBP	-10.17605313	0.1029467193	-0.01041176178	-0.8565684858	0.08498077804
DATA	2.756496105	-0.002248434483	-0.007008433458	0.262888406	0.00843594484
F	1.459949113	-0.004062747475	0.004940836689	0.1615210655	0.0009976721219
GOOG	3.021834997	-0.01521674464	-0.006349870969	0.2772113634	0.003242019965
HUBS	2.735683525	0.01272587459	-0.01156851131	0.2662324271	0.01495586984
JPM	1.438783981	-0.001481606583	0.01403720119	0.1781058969	0.003532077118
MFC	0.3269370546	0.009974198836	0.01435932074	0.0851244414	0.00542035331
MMM	0.7704877252	0.004727728006	-0.0009209782378	0.09454983679	0.001160755808
MTN	-0.3132525658	0.001190907799	-0.001284227724	-0.01047926426	0.001551692795
NFLX	2.163103286	-0.0454041571	0.005244714194	0.1906463654	0.01522210836
NKE	1.025820319	-0.002513744654	0.004313390774	0.1212476946	0.001862704001
SBUX	0.545487862	0.006966817672	0.00250915075	0.08167233311	0.002293177964
TLT	0.1056764182	-0.001157006237	-0.009478496528	0.01190242841	0.001741819926
VYM	0.9822043255	-0.001202984657	0.00123556728	0.1128532644	0.0003076525339
WM	0.5847124409	-0.003354531801	0.0004562680005	0.07241071988	0.001286681144

Figure 23

	AMZN	AZO	BA	CI	COST	CRBP	DATA	F	GOOG	HUBS	JPM	MFC	MMM	MTN	NFLX	NKE	SBUX	TLT	VYM	WM
AMZN	1.489E-03	6.022E-04	-8.772E-04	-4.798E-04	4.615E-04	1.659E-03	1.210E-03	-5.137E-04	9.574E-04	2.026E-03	-1.643E-03	-1.646E-03	2.884E-04	1.439E-04	-1.107E-03	-4.637E-04	-1.451E-04	1.211E-03	-5.389E-05	-3.728E-05
AZO	6.022E-04	7.808E-04	-3.843E-04	1.001E-03	6.927E-04	2.066E-03	1.271E-03	2.647E-04	5.790E-04	2.300E-03	1.678E-04	3.668E-04	6.546E-04	-3.126E-05	-1.957E-03	1.841E-04	5.993E-04	2.316E-04	3.279E-04	4.155E-05
BA	-8.772E-04	-3.843E-04	5.239E-04	2.902E-04	-3.678E-04	-6.767E-04	-8.003E-04	2.518E-04	-6.544E-04	-1.284E-03	9.103E-04	9.437E-04	-1.977E-04	-7.389E-05	5.980E-04	2.367E-04	6.180E-05	-7.100E-04	-1.764E-06	3.517E-06
CI	-4.798E-04	1.001E-03	2.902E-04	3.777E-03	8.181E-05	7.533E-03	7.663E-04	8.897E-04	-9.318E-04	2.517E-03	2.224E-03	3.233E-03	1.127E-03	-1.667E-04	-4.809E-03	7.653E-04	1.675E-03	-1.105E-03	6.074E-04	-6.372E-05
COST	4.615E-04	6.927E-04	-3.678E-04	8.181E-05	1.451E-03	-2.782E-03	1.653E-03	5.893E-04	1.452E-03	2.138E-03	4.204E-04	8.588E-05	5.740E-04	-1.127E-04	-8.674E-05	4.021E-04	4.217E-04	2.620E-04	4.870E-04	2.291E-04
CRBP	1.659E-03	2.066E-03	-6.767E-04	7.533E-03	-2.782E-03	3.104E-02	3.760E-04	-1.029E-03	-3.841E-03	5.461E-03	-4.780E-04	2.934E-03	1.840E-03	3.334E-04	-1.446E-02	-6.377E-04	2.396E-03	-4.114E-05	-1.195E-04	-9.332E-04
DATA	1.210E-03	1.271E-03	-8.003E-04	7.663E-04	1.653E-03	3.760E-04	2.474E-03	4.677E-04	1.692E-03	3.876E-03	-2.945E-05	-8.047E-05	1.003E-03	-6.081E-05	-2.004E-03	2.915E-04	7.399E-04	6.933E-04	5.786E-04	1.741E-04
F	-5.137E-04	2.647E-04	2.518E-04	8.897E-04	5.893E-04	-1.029E-03	4.677E-04	7.036E-04	1.941E-04	6.452E-04	1.352E-03	1.338E-03	3.655E-04	-1.550E-04	-3.327E-04	5.583E-04	5.728E-04	-5.976E-04	3.897E-04	1.310E-04
GOOG	9.574E-04	5.790E-04	-6.544E-04	-9.318E-04	1.452E-03	-3.841E-03	1.692E-03	1.941E-04	1.827E-03	1.966E-03	-5.888E-04	-1.131E-03	3.516E-04	-2.507E-05	7.175E-04	6.407E-05	-3.804E-05	8.390E-04	2.946E-04	2.158E-04
HUBS	2.026E-03	2.300E-03	-1.284E-03	2.517E-03	2.138E-03	5.461E-03	3.876E-03	6.452E-04	1.966E-03	6.852E-03	7.847E-05	5.539E-04	1.872E-03	-6.088E-05	-5.434E-03	4.205E-04	1.595E-03	9.541E-04	9.144E-04	1.313E-04
JPM	-1.643E-03	1.678E-04	9.103E-04	2.224E-03	4.204E-04	-4.780E-04	-2.945E-05	1.352E-03	-5.888E-04	7.847E-05	3.129E-03	3.350E-03	5.112E-04	-3.103E-04	-8.226E-04	1.122E-03	1.153E-03	-1.714E-03	6.317E-04	1.617E-04
MFC	-1.646E-03	3.668E-04	9.437E-04	3.233E-03	8.588E-05	2.934E-03	-8.047E-05	1.338E-03	-1.131E-03	5.539E-04	3.350E-03	3.974E-03	7.249E-04	-2.980E-04	-2.437E-03	1.138E-03	1.494E-03	-1.894E-03	6.526E-04	6.521E-05
MMM	2.884E-04	6.546E-04	-1.977E-04	1.127E-03	5.740E-04	1.840E-03	1.003E-03	3.655E-04	3.516E-04	1.872E-03	5.112E-04	7.249E-04	5.950E-04	-5.976E-05	-1.770E-03	2.771E-04	6.347E-04	-2.739E-05	3.350E-04	4.450E-05
MTN	1.439E-04	-3.126E-05	-7.389E-05	-1.667E-04	-1.127E-04	3.334E-04	-6.081E-05	-1.550E-04	-2.507E-05	-6.088E-05	-3.103E-04	-2.980E-04	-5.976E-05	3.524E-05	-3.223E-06	-1.239E-04	-1.101E-04	1.487E-04	-7.877E-05	-2.950E-05
NFLX	-1.107E-03	-1.957E-03	5.980E-04	-4.809E-03	-8.674E-05	-1.446E-02	-2.004E-03	-3.327E-04	7.175E-04	-5.434E-03	-8.226E-04	-2.437E-03	-1.770E-03	-3.223E-06	8.455E-03	-3.041E-04	-2.050E-03	8.711E-05	-5.903E-04	2.468E-04
NKE	-4.637E-04	1.841E-04	2.367E-04	7.653E-04	4.021E-04	-6.377E-04	2.915E-04	5.583E-04	6.407E-05	4.205E-04	1.122E-03	1.138E-03	2.771E-04	-1.239E-04	-3.041E-04	4.476E-04	4.642E-04	-5.253E-04	2.997E-04	9.490E-05
SBUX	-1.451E-04	5.993E-04	6.180E-05	1.675E-03	4.217E-04	2.396E-03	7.399E-04	5.728E-04	-3.804E-05	1.595E-03	1.153E-03	1.494E-03	6.347E-04	-1.101E-04	-2.050E-03	4.642E-04	8.481E-04	-4.582E-04	4.013E-04	3.848E-05
TLT	1.211E-03	2.316E-04	-7.100E-04	-1.105E-03	2.620E-04	-4.114E-05	6.933E-04	-5.976E-04	8.390E-04	9.541E-04	-1.714E-03	-1.894E-03	-2.739E-05	1.487E-04	8.711E-05	-5.253E-04	-4.582E-04	1.130E-03	-1.847E-04	-2.829E-05
VYM	-5.389E-05	3.279E-04	-1.764E-06	6.074E-04	4.870E-04	-1.195E-04	5.786E-04	3.897E-04	2.946E-04	9.144E-04	6.317E-04	6.526E-04	3.350E-04	-7.877E-05	-5.903E-04	2.997E-04	4.013E-04	-1.847E-04	2.661E-04	7.749E-05
WM	-3.728E-05	4.155E-05	3.517E-06	-6.372E-05	2.291E-04	-9.332E-04	1.741E-04	1.310E-04	2.158E-04	1.313E-04	1.617E-04	6.521E-05	4.450E-05	-2.950E-05	2.468E-04	9.490E-05	3.848E-05	-2.829E-05	7.749E-05	4.910E-05

Figure 24

5.3 CAPM Extension - F_1

stocks	Beta CAPM	Beta F1	Expected Asset Returns	Asset Returns Variance
AMZN	0.6041980458	0.004794948032	0.1455193597	0.04653771858
AZO	1.69913554	0.007344103094	-0.01616708259	0.02605462627
BA	-0.4811260308	-0.003280036214	0.3704289645	0.02570371607
CI	1.902845641	-0.003693900017	0.01995684419	0.04783724458
COST	1.93338817	0.004604607755	-0.03016989977	0.03148173099
CRBP	0.8160761117	-0.004091542326	0.1718122795	1.019769336
DATA	2.239402536	-0.003150214503	-0.02557768105	0.1012313381
F	0.8328017035	-0.005956166755	0.1814836967	0.01197206546
GOOG	1.288879963	0.001377644386	0.07132183059	0.03890423958
HUBS	3.936794718	0.0001307486859	-0.2305043542	0.1794704381
JPM	0.9677557394	-0.01247917543	0.204368138	0.04238492542
MFC	1.204216129	-0.01075184172	0.1582171849	0.06504423972
MMM	1.390171426	0.002541781117	0.05089268834	0.0139290697
MTN	-0.2422877951	-0.0008268886189	0.3128493742	0.01862031354
NFLX	-3.086230723	-0.007370936019	0.923058729	0.1826653003
NKE	0.621192884	-0.004559505437	0.2037722118	0.02235244801
SBUX	1.457163969	0.002156013671	0.04429724885	0.02751813557
TLT	0.1859262855	0.008434655179	0.1829483871	0.02090183911
VYM	0.8586476026	-0.0002919420814	0.1413248818	0.003691830407
WM	0.4577127676	0.005220572903	0.1639591037	0.01544017372

Figure 25

	AMZN	AZO	BA	CI	COST	CRBP	DATA	F	GOOG	HUBS	JPM	MFC	MMM	MTN	NFLX	NKE	SBUX	TLT	VYM	WM
AMZN	0.00034	0.00062	-0.00024	0.00003	0.00050	-0.00012	0.00011	-0.00022	0.00024	0.00050	-0.00057	-0.00044	0.00032	-0.00008	-0.00079	-0.00017	0.00030	0.00049	0.00009	0.00035
AZO	0.00062	0.00129	-0.00045	0.00057	0.00119	0.00006	0.00076	-0.00007	0.00067	0.00175	-0.00049	-0.00026	0.00080	-0.00017	-0.00191	-0.00006	0.00080	0.00070	0.00036	0.00059
BA	-0.00024	-0.00045	0.00017	-0.00007	-0.00038	0.00006	-0.00013	0.00013	-0.00019	-0.00043	0.00036	0.00027	-0.00024	0.00006	0.00060	0.00010	-0.00024	-0.00033	-0.00008	-0.00024
CI	0.00003	0.00057	-0.00007	0.00156	0.00094	0.00086	0.00174	0.00102	0.00076	0.00262	0.00161	0.00163	0.00073	-0.00010	-0.00150	0.00077	0.00081	-0.00053	0.00060	-0.00010
COST	0.00050	0.00119	-0.00038	0.00094	0.00123	0.00032	0.00114	0.00027	0.00077	0.00220	0.00012	0.00031	0.00086	-0.00016	-0.00197	0.00019	0.00089	0.00039	0.00047	0.00043
CRBP	-0.00012	0.00006	0.00006	0.00086	0.00032	0.00054	0.00093	0.00067	0.00033	0.00126	0.00114	0.00110	0.00028	-0.00002	-0.00051	0.00050	0.00033	-0.00050	0.00030	-0.00020
DATA	0.00011	0.00076	-0.00013	0.00174	0.00114	0.00093	0.00196	0.00109	0.00089	0.00302	0.00168	0.00173	0.00088	-0.00012	-0.00183	0.00082	0.00096	-0.00049	0.00068	-0.00004
F	-0.00022	-0.00007	0.00013	0.00102	0.00027	0.00067	0.00109	0.00084	0.00033	0.00139	0.00149	0.00141	0.00026	-0.00001	-0.00042	0.00064	0.00032	-0.00071	0.00033	-0.00032
GOOG	0.00024	0.00067	-0.00019	0.00076	0.00077	0.00033	0.00089	0.00033	0.00051	0.00156	0.00039	0.00048	0.00055	-0.00010	-0.00122	0.00025	0.00058	0.00007	0.00034	0.00018
HUBS	0.00050	0.00175	-0.00043	0.00262	0.00220	0.00126	0.00302	0.00139	0.00156	0.00500	0.00192	0.00213	0.00163	-0.00026	-0.00351	0.00104	0.00173	-0.00023	0.00111	0.00029
JPM	-0.00057	-0.00049	0.00036	0.00161	0.00012	0.00114	0.00168	0.00149	0.00039	0.00192	0.00273	0.00254	0.00023	0.00003	-0.00019	0.00113	0.00033	-0.00143	0.00048	-0.00072
MFC	-0.00044	-0.00026	0.00027	0.00163	0.00031	0.00110	0.00173	0.00141	0.00048	0.00213	0.00254	0.00239	0.00035	0.00000	-0.00050	0.00107	0.00045	-0.00126	0.00052	-0.00059
MMM	0.00032	0.00080	-0.00024	0.00073	0.00086	0.00028	0.00088	0.00026	0.00055	0.00163	0.00023	0.00035	0.00061	-0.00011	-0.00137	0.00019	0.00063	0.00019	0.00035	0.00026
MTN	-0.00008	-0.00017	0.00006	-0.00010	-0.00016	-0.00002	-0.00012	-0.00001	-0.00010	-0.00026	0.00003	0.00000	-0.00011	0.00002	0.00026	-0.00001	-0.00011	-0.00008	-0.00005	-0.00007
NFLX	-0.00079	-0.00191	0.00060	-0.00150	-0.00197	-0.00051	-0.00183	-0.00042	-0.00122	-0.00351	-0.00019	-0.00050	-0.00137	0.00026	0.00314	-0.00031	-0.00141	-0.00062	-0.00075	-0.00069
NKE	-0.00017	-0.00006	0.00010	0.00077	0.00019	0.00050	0.00082	0.00064	0.00025	0.00104	0.00113	0.00107	0.00019	-0.00001	-0.00031	0.00048	0.00024	-0.00054	0.00025	-0.00024
SBUX	0.00030	0.00080	-0.00024	0.00081	0.00089	0.00033	0.00096	0.00032	0.00058	0.00173	0.00033	0.00045	0.00063	-0.00011	-0.00141	0.00024	0.00066	0.00015	0.00038	0.00024
TLT	0.00049	0.00070	-0.00033	-0.00053	0.00039	-0.00050	-0.00049	-0.00071	0.00007	-0.00023	-0.00143	-0.00126	0.00019	-0.00008	-0.00062	-0.00054	0.00015	0.00092	-0.00009	0.00055
VYM	0.00009	0.00036	-0.00008	0.00060	0.00047	0.00030	0.00068	0.00033	0.00034	0.00111	0.00048	0.00052	0.00035	-0.00005	-0.00075	0.00025	0.00038	-0.00009	0.00025	0.00004
WM	0.00035	0.00059	-0.00024	-0.00010	0.00043	-0.00020	-0.00004	-0.00032	0.00018	0.00029	-0.00072	-0.00059	0.00026	-0.00007	-0.00069	-0.00024	0.00024	0.00055	0.00004	0.00036

Figure 26

5.4 CAPM Extension - F_2

	Beta CAPM	Beta F2	Expected Asset Returns	Asset Returns Variance
AMZN	0.3305450216	-0.006172527558	0.1795781391	0.04653771858
AZO	1.93336699	0.06037459852	0.3537649615	0.02605462627
BA	0.06346524892	0.04241907062	0.5567245241	0.02570371607
CI	2.366228608	0.03174834832	0.1093534313	0.04783724458
COST	2.151678911	0.04548800175	0.2221389514	0.03148173099
CRBP	-0.09472183146	-0.1170307341	-0.3003311068	1.019769336
DATA	1.979128148	-0.04297620683	-0.2091273025	0.1012313381
F	0.8528623078	-0.02651814623	-0.004806033824	0.01197206546
GOOG	0.4183805894	-0.08640513611	-0.2302232349	0.03890423958
HUBS	1.892007628	-0.2179073942	-0.6927601799	0.1794704381
JPM	1.256508277	-0.02919156717	-0.06753292737	0.04238492542
MFC	1.584313748	-0.01111682946	-0.01780969251	0.06504423972
MMM	1.7634506	0.05212570255	0.3262374105	0.0139290697
MTN	-0.5606015596	-0.0379989109	0.1211535775	0.01862031354
NFLX	-4.208390826	-0.1554010804	-0.02865146023	0.1826653003
NKE	0.8693950059	0.004585454099	0.1648707552	0.02235244801
SBUX	1.496951094	0.01462736556	0.1334750453	0.02751813557
TLT	0.09879124889	0.03127649019	0.4662531754	0.02090183911
VYM	0.954951969	0.00888765057	0.1780116897	0.003691830407
WM	0.7860072411	0.06020878118	0.5571928006	0.01544017372

Figure 27

	AMZN	AZO	BA	CI	COST	CRBP	DATA	F	GOOG	HUBS	JPM	MFC	MMM	MTN	NFLX	NKE	SBUX	TLT	VYM	WM
AMZN	5.38E-05	1.22E-04	-7.69E-05	2.34E-04	1.79E-04	2.22E-04	3.34E-04	1.60E-04	2.25E-04	6.73E-04	2.16E-04	2.21E-04	1.17E-04	5.60E-06	-2.18E-04	1.00E-04	1.59E-04	-5.01E-05	1.02E-04	-2.17E-05
AZO	1.22E-04	1.09E-03	1.95E-04	1.17E-03	1.13E-03	-5.02E-04	7.07E-04	2.73E-04	-1.54E-04	-1.86E-05	4.41E-04	6.57E-04	9.85E-04	-3.97E-04	-2.47E-03	4.03E-04	7.20E-04	1.67E-04	4.57E-04	5.84E-04
BA	-7.69E-05	1.95E-04	2.76E-04	-4.31E-05	7.17E-05	-7.71E-04	-5.00E-04	-2.69E-04	-6.22E-04	-1.66E-03	-3.30E-04	-2.45E-04	1.58E-04	-1.93E-04	-5.84E-04	-6.31E-05	-6.37E-05	1.98E-04	-4.36E-05	3.17E-04
CI	2.34E-04	1.17E-03	-4.31E-05	1.52E-03	1.35E-03	1.73E-04	1.41E-03	6.24E-04	4.54E-04	1.71E-03	9.00E-04	1.08E-03	1.07E-03	-2.98E-04	-2.50E-03	5.72E-04	9.72E-04	2.98E-06	6.21E-04	4.04E-04
COST	1.79E-04	1.13E-03	7.17E-05	1.35E-03	1.24E-03	-1.53E-04	1.07E-03	4.53E-04	1.60E-04	8.70E-04	6.76E-04	8.75E-04	1.03E-03	-3.45E-04	-2.48E-03	4.89E-04	8.48E-04	8.19E-05	5.41E-04	4.90E-04
CRBP	2.22E-04	-5.02E-04	-7.71E-04	1.73E-04	-1.53E-04	2.15E-03	1.44E-03	7.71E-04	1.75E-03	4.68E-03	9.51E-04	7.20E-04	-4.03E-04	5.28E-04	1.54E-03	1.96E-04	2.11E-04	-5.52E-04	1.43E-04	-8.70E-04
DATA	3.34E-04	7.07E-04	-5.00E-04	1.41E-03	1.07E-03	1.44E-03	2.08E-03	1.00E-03	1.44E-03	4.28E-03	1.35E-03	1.36E-03	6.83E-04	6.03E-05	-1.23E-03	6.10E-04	9.63E-04	-3.29E-04	6.20E-04	-1.74E-04
F	1.60E-04	2.73E-04	-2.69E-04	6.24E-04	4.53E-04	7.71E-04	1.00E-03	4.86E-04	7.43E-04	2.17E-03	6.48E-04	6.40E-04	2.69E-04	6.17E-05	-4.30E-04	2.78E-04	4.32E-04	-1.80E-04	2.79E-04	-1.34E-04
GOOG	2.25E-04	-1.54E-04	-6.22E-04	4.54E-04	1.60E-04	1.75E-03	1.44E-03	7.43E-04	1.48E-03	4.08E-03	9.44E-04	7.98E-04	-9.58E-05	3.57E-04	7.02E-04	2.75E-04	3.71E-04	-4.38E-04	2.43E-04	-6.07E-04
HUBS	6.73E-04	-1.86E-05	-1.66E-03	1.71E-03	8.70E-04	4.68E-03	4.28E-03	2.17E-03	4.08E-03	1.13E-02	2.79E-03	2.47E-03	1.04E-04	8.46E-04	1.02E-03	9.17E-04	1.30E-03	-1.16E-03	8.49E-04	-1.47E-03
JPM	2.16E-04	4.41E-04	-3.30E-04	9.00E-04	6.76E-04	9.51E-04	1.35E-03	6.48E-04	9.44E-04	2.79E-03	8.71E-04	8.78E-04	4.28E-04	4.68E-05	-7.56E-04	3.91E-04	6.16E-04	-2.18E-04	3.97E-04	-1.25E-04
MFC	2.21E-04	6.57E-04	-2.45E-04	1.08E-03	8.75E-04	7.20E-04	1.36E-03	6.40E-04	7.98E-04	2.47E-03	8.78E-04	9.38E-04	6.19E-04	-5.55E-05	-1.27E-03	4.44E-04	7.21E-04	-1.52E-04	4.63E-04	3.21E-05
MMM	1.17E-04	9.85E-04	1.58E-04	1.07E-03	1.03E-03	-4.03E-04	6.83E-04	2.69E-04	-9.58E-05	1.04E-04	4.28E-04	6.19E-04	8.89E-04	-3.49E-04	-2.22E-03	3.73E-04	6.63E-04	1.38E-04	4.22E-04	5.11E-04
MTN	5.60E-06	-3.97E-04	-1.93E-04	-2.98E-04	-3.45E-04	5.28E-04	6.03E-05	6.17E-05	3.57E-04	8.46E-04	4.68E-05	-5.55E-05	-3.49E-04	2.07E-04	9.72E-04	-7.81E-05	-1.64E-04	-1.46E-04	-1.03E-04	-3.20E-04
NFLX	-2.18E-04	-2.47E-03	-5.84E-04	-2.50E-03	-2.48E-03	1.54E-03	-1.23E-03	-4.30E-04	7.02E-04	1.02E-03	-7.56E-04	-1.27E-03	-2.22E-03	9.72E-04	5.68E-03	-8.31E-04	-1.52E-03	-4.77E-04	-9.61E-04	-1.45E-03
NKE	1.00E-04	4.03E-04	-6.31E-05	5.72E-04	4.89E-04	1.96E-04	6.10E-04	2.78E-04	2.75E-04	9.17E-04	3.91E-04	4.44E-04	3.73E-04	-7.81E-05	-8.31E-04	2.24E-04	3.72E-04	-3.26E-05	2.38E-04	9.64E-05
SBUX	1.59E-04	7.20E-04	-6.37E-05	9.72E-04	8.48E-04	2.11E-04	9.63E-04	4.32E-04	3.71E-04	1.30E-03	6.16E-04	7.21E-04	6.63E-04	-1.64E-04	-1.52E-03	3.72E-04	6.27E-04	-2.41E-05	4.01E-04	2.16E-04
TLT	-5.01E-05	1.67E-04	1.98E-04	2.98E-06	8.19E-05	-5.52E-04	-3.29E-04	-1.80E-04	-4.38E-04	-1.16E-03	-2.18E-04	-1.52E-04	1.38E-04	-1.46E-04	-4.77E-04	-3.26E-05	-2.41E-05	1.43E-04	-1.75E-05	2.38E-04
VYM	1.02E-04	4.57E-04	-4.36E-05	6.21E-04	5.41E-04	1.43E-04	6.20E-04	2.79E-04	2.43E-04	8.49E-04	3.97E-04	4.63E-04	4.22E-04	-1.03E-04	-9.61E-04	2.38E-04	4.01E-04	-1.75E-05	2.56E-04	1.34E-04
WM	-2.17E-05	5.84E-04	3.17E-04	4.04E-04	4.90E-04	-8.70E-04	-1.74E-04	-1.34E-04	-6.07E-04	-1.47E-03	-1.25E-04	3.21E-05	5.11E-04	-3.20E-04	-1.45E-03	9.64E-05	2.16E-04	2.38E-04	1.34E-04	5.00E-04

Figure 28

5.5 CAPM Extension - F_1 & F_2

stocks	Beta CAPM	Beta F1	Beta F2	Expected Asset Returns	Asset Returns Variance
AMZN	0.3702120396	0.006166307777	-0.0316064743	-0.004509516008	0.04653771858
AZO	1.970383962	0.005754353463	0.03663982484	0.1577074224	0.02605462627
BA	0.02334556979	-0.006236674761	0.06814325752	0.836662663	0.02570371607
CI	2.326493815	-0.006176843594	0.05722575177	0.3128829957	0.04783724458
COST	2.172292523	0.003204422257	0.03227083712	0.1196173603	0.03148173099
CRBP	-0.08699448248	0.001201230005	-0.1219854041	-0.323943453	1.019769336
DATA	1.969055891	-0.001565750064	-0.03651801415	-0.1734287787	0.1012313381
F	0.8152103306	-0.005853066142	-0.002376215931	0.1691782456	0.01197206546
GOOG	0.4585480677	0.006244105226	-0.1121599712	-0.3558060925	0.03890423958
HUBS	1.967110022	0.01167479932	-0.2660620168	-0.7861272671	0.1794704381
JPM	1.168657035	-0.01365663025	0.02713744185	0.3562088345	0.04238492542
MFC	1.503851716	-0.01250796461	0.04047431996	0.3828116194	0.06504423972
MMM	1.765645365	0.0003411800899	0.05071844937	0.3140459161	0.0139290697
MTN	-0.5541624819	0.001000965952	-0.04212755897	0.09074521266	0.01862031354
NFLX	-4.213313629	-0.0007652583965	-0.1522446468	-0.007807669351	0.1826653003
NKE	0.8321122207	-0.00579567461	0.02849066362	0.363552824	0.02235244801
SBUX	1.508870972	0.001852966047	0.006984503497	0.07721940551	0.02751813557
TLT	0.1542446998	0.008620336587	-0.004279500465	0.1608463567	0.02090183911
VYM	0.949643217	-0.0008252548405	0.01229154937	0.2048533527	0.003691830407
WM	0.8064426178	0.003176715297	0.04710589845	0.4301390251	0.01544017372

Figure 29

	AMZN	AZO	BA	CI	COST	CRBP	DATA	F	GOOG	HUBS	JPM	MFC	MMM	MTN	NFLX	NKE	SBUX	TLT	VYM	WM
AMZN	0.00045	0.00049	-0.00048	-0.00016	0.00039	0.00030	0.00023	-0.00022	0.00063	0.00142	-0.00066	-0.00058	0.00014	0.00007	-0.00027	-0.00027	0.00028	0.00050	0.00005	0.00018
AZO	0.00049	0.00144	-0.00018	0.00080	0.00132	-0.00043	0.00061	-0.00008	0.00022	0.00068	-0.00038	-0.00009	0.00101	-0.00034	-0.00252	0.00005	0.00083	0.00068	0.00041	0.00077
BA	-0.00048	-0.00018	0.00068	0.00036	-0.00014	-0.00085	-0.00040	0.00011	-0.00103	-0.00242	0.00056	0.00057	0.00014	-0.00026	-0.00053	0.00031	-0.00018	-0.00036	0.00001	0.00011
CI	-0.00016	0.00080	0.00036	0.00192	0.00114	0.00010	0.00151	0.00100	0.00005	0.00095	0.00178	0.00189	0.00105	-0.00036	-0.00245	0.00095	0.00085	-0.00055	0.00067	0.00020
COST	0.00039	0.00132	-0.00014	0.00114	0.00135	-0.00011	0.00102	0.00026	0.00037	0.00126	0.00022	0.00046	0.00104	-0.00031	-0.00251	0.00030	0.00091	0.00037	0.00051	0.00060
CRBP	0.00030	-0.00043	-0.00085	0.00010	-0.00011	0.00217	0.00142	0.00070	0.00183	0.00483	0.00078	0.00056	-0.00040	0.00054	0.00153	0.00012	0.00023	-0.00044	0.00013	-0.00083
DATA	0.00023	0.00061	-0.00040	0.00151	0.00102	0.00142	0.00211	0.00110	0.00134	0.00409	0.00157	0.00157	0.00068	0.00004	-0.00122	0.00070	0.00093	-0.00047	0.00063	-0.00023
F	-0.00022	-0.00008	0.00011	0.00100	0.00026	0.00070	0.00110	0.00084	0.00036	0.00146	0.00148	0.00140	0.00025	0.00000	-0.00038	0.00063	0.00032	-0.00071	0.00033	-0.00033
GOOG	0.00063	0.00022	-0.00103	0.00005	0.00037	0.00183	0.00134	0.00036	0.00189	0.00484	0.00005	-0.00002	-0.00007	0.00042	0.00065	-0.00010	0.00049	0.00012	0.00019	-0.00040
HUBS	0.00142	0.00068	-0.00242	0.00095	0.00126	0.00483	0.00409	0.00146	0.00484	0.01276	0.00113	0.00095	0.00015	0.00097	0.00093	0.00021	0.00153	-0.00011	0.00075	-0.00108
JPM	-0.00066	-0.00038	0.00056	0.00178	0.00022	0.00078	0.00157	0.00148	0.00005	0.00113	0.00282	0.00266	0.00038	-0.00010	-0.00065	0.00122	0.00035	-0.00145	0.00051	-0.00058
MFC	-0.00058	-0.00009	0.00057	0.00189	0.00046	0.00056	0.00157	0.00140	-0.00002	0.00095	0.00266	0.00257	0.00057	-0.00019	-0.00117	0.00120	0.00048	-0.00128	0.00057	-0.00038
MMM	0.00014	0.00101	0.00014	0.00105	0.00104	-0.00040	0.00068	0.00025	-0.00007	0.00015	0.00038	0.00057	0.00089	-0.00035	-0.00222	0.00035	0.00067	0.00017	0.00042	0.00052
MTN	0.00007	-0.00034	-0.00026	-0.00036	-0.00031	0.00054	0.00004	0.00000	0.00042	0.00097	-0.00010	-0.00019	-0.00035	0.00022	0.00096	-0.00014	-0.00015	-0.00006	-0.00011	-0.00029
NFLX	-0.00027	-0.00252	-0.00053	-0.00245	-0.00251	0.00153	-0.00122	-0.00038	0.00065	0.00093	-0.00065	-0.00117	-0.00222	0.00096	0.00569	-0.00079	-0.00153	-0.00055	-0.00095	-0.00147
NKE	-0.00027	0.00005	0.00031	0.00095	0.00030	0.00012	0.00070	0.00063	-0.00010	0.00021	0.00122	0.00120	0.00035	-0.00014	-0.00079	0.00057	0.00026	-0.00055	0.00029	-0.00010
SBUX	0.00028	0.00083	-0.00018	0.00085	0.00091	0.00023	0.00093	0.00032	0.00049	0.00153	0.00035	0.00048	0.00067	-0.00015	-0.00153	0.00026	0.00066	0.00014	0.00038	0.00028
TLT	0.00050	0.00068	-0.00036	-0.00055	0.00037	-0.00044	-0.00047	-0.00071	0.00012	-0.00011	-0.00145	-0.00128	0.00017	-0.00006	-0.00055	-0.00055	0.00014	0.00092	-0.00009	0.00052
VYM	0.00005	0.00041	0.00001	0.00067	0.00051	0.00013	0.00063	0.00033	0.00019	0.00075	0.00051	0.00057	0.00042	-0.00011	-0.00095	0.00029	0.00038	-0.00009	0.00026	0.00011
WM	0.00018	0.00077	0.00011	0.00020	0.00060	-0.00083	-0.00023	-0.00033	-0.00040	-0.00108	-0.00058	-0.00038	0.00052	-0.00029	-0.00147	-0.00010	0.00028	0.00052	0.00011	0.00061

Figure 30

5.6 French & Fama Extension - F_1

stocks	Beta Mkt	Beta SMB	Beta HML	Beta F1	Expected Asset Returns	Asset Returns Variance
AMZN	-0.4229291446	0.005989375257	-0.02087048401	-0.01173313429	-0.04066818794	0.04653771858
AZO	1.602208236	0.002345769364	0.00907080584	0.01459433595	0.1989725925	0.02605462627
BA	-0.05558566723	-0.002115237397	0.0109170871	0.005379319715	0.5367907189	0.02570371607
CI	-0.4867088076	0.02329682995	0.00949993214	0.004180365993	1.052359593	0.04783724458
COST	3.476040108	-0.01321520514	0.00518154938	0.008549903096	-0.2427628457	0.03148173099
CRBP	-11.84936462	0.1091786868	-0.03832788973	-0.03312078638	6.847716601	1.019769336
DATA	1.422920941	0.002718242107	-0.02925668497	-0.02639619602	-0.3446147996	0.1012313381
F	1.143375778	-0.002883723291	-0.0003406077763	-0.006266112342	0.09564204547	0.01197206546
GOOG	2.430897006	-0.01301589539	-0.01620858566	-0.01169676482	-0.4023096865	0.03890423958
HUBS	1.37987074	0.01777537146	-0.03418775667	-0.02683635764	-0.3025561205	0.1794704381
JPM	1.236526522	-0.0007283326656	0.01066290708	-0.004003394548	0.3925402055	0.04238492542
MFC	0.4502606301	0.009514900903	0.01641674803	0.002441012221	0.8615902605	0.06504423972
MMM	1.055633517	0.003665750423	0.003836155419	0.005644049486	0.2190641727	0.0139290697
MTN	-0.5923392237	0.002230319156	-0.005940275861	-0.005524117671	0.2652039903	0.01862031354
NFLX	1.583032871	-0.04324378232	-0.004432694957	-0.01148165683	-0.3749151298	0.1826653003
NKE	0.8490663443	-0.001855454179	0.001364575443	-0.003498589876	0.1752070052	0.02235244801
SBUX	1.192147701	0.004558441775	0.01329748209	0.01279969837	0.4297588629	0.02751813557
TLT	0.2390677236	-0.001653799737	-0.007253107669	0.002640288407	0.01553145895	0.02090183911
VYM	1.085003742	-0.00158584381	0.002950586684	0.002034766118	0.1492463018	0.003691830407
WM	1.428057594	-0.006495429098	0.01452593245	0.01669280035	0.2429832472	0.01544017372

Figure 31

	AMZN	AZO	BA	CI	COST	CRBP	DATA	F	GOOG	HUBS	JPM	MFC	MMM	MTN	NFLX	NKE	SBUX	TLT	VYM	WM
AMZN	0.00207	-0.00012	-0.00114	-0.00069	0.00004	0.00330	0.00252	-0.00020	0.00154	0.00335	-0.00145	-0.00177	0.00001	0.00042	-0.00054	-0.00029	-0.00078	0.00108	-0.00015	-0.00086
AZO	-0.00012	0.00168	-0.00005	0.00126	0.00122	0.00003	-0.00035	-0.00012	-0.00014	0.00065	-0.00008	0.00052	0.00100	-0.00037	-0.00266	-0.00003	0.00139	0.00039	0.00045	0.00107
BA	-0.00114	-0.00005	0.00065	0.00038	-0.00017	-0.00143	-0.00140	0.00011	-0.00092	-0.00189	0.00082	0.00100	-0.00007	-0.00020	0.00034	0.00016	0.00035	-0.00065	0.00004	0.00038
CI	-0.00069	0.00126	0.00038	0.00385	0.00023	0.00695	0.00030	0.00078	-0.00114	0.00204	0.00215	0.00328	0.00123	-0.00026	-0.00501	0.00070	0.00190	-0.00106	0.00064	0.00023
COST	0.00004	0.00122	-0.00017	0.00023	0.00176	-0.00398	0.00070	0.00036	0.00103	0.00117	0.00028	0.00017	0.00078	-0.00031	-0.00050	0.00028	0.00088	0.00036	0.00056	0.00083
CRBP	0.00330	0.00003	-0.00143	0.00695	-0.00398	0.03566	0.00406	-0.00015	-0.00221	0.00921	0.00008	0.00259	0.00105	0.00110	-0.01286	-0.00015	0.00061	-0.00041	-0.00040	-0.00326
DATA	0.00252	-0.00035	-0.00140	0.00030	0.00070	0.00406	0.00541	0.00116	0.00299	0.00686	0.00042	-0.00035	0.00037	0.00055	-0.00073	0.00068	-0.00068	0.00040	0.00035	-0.00168
F	-0.00020	-0.00012	0.00011	0.00078	0.00036	-0.00015	0.00116	0.00087	0.00050	0.00135	0.00146	0.00127	0.00022	-0.00001	-0.00003	0.00065	0.00023	-0.00067	0.00034	-0.00031
GOOG	0.00154	-0.00014	-0.00092	-0.00114	0.00103	-0.00221	0.00299	0.00050	0.00240	0.00329	-0.00039	-0.00125	0.00007	0.00025	0.00128	0.00024	-0.00067	0.00071	0.00019	-0.00061
HUBS	0.00335	0.00065	-0.00189	0.00204	0.00117	0.00921	0.00686	0.00135	0.00329	0.00989	0.00053	0.00028	0.00123	0.00056	-0.00414	0.00082	0.00015	0.00066	0.00068	-0.00176
JPM	-0.00145	-0.00008	0.00082	0.00215	0.00028	0.00008	0.00042	0.00146	-0.00039	0.00053	0.00320	0.00331	0.00042	-0.00022	-0.00063	0.00118	0.00094	-0.00176	0.00060	-0.00012
MFC	-0.00177	0.00052	0.00100	0.00328	0.00017	0.00259	-0.00035	0.00127	-0.00125	0.00028	0.00331	0.00400	0.00078	-0.00035	-0.00256	0.00110	0.00163	-0.00187	0.00067	0.00024
MMM	0.00001	0.00100	-0.00007	0.00123	0.00078	0.00105	0.00037	0.00022	0.00007	0.00123	0.00042	0.00078	0.00073	-0.00019	-0.00204	0.00019	0.00094	0.00004	0.00038	0.00044
MTN	0.00042	-0.00037	-0.00020	-0.00026	-0.00031	0.00110	0.00055	-0.00001	0.00025	0.00056	-0.00022	-0.00035	-0.00019	0.00016	0.00026	-0.00004	-0.00041	0.00009	-0.00013	-0.00042
NFLX	-0.00054	-0.00266	0.00034	-0.00501	-0.00050	-0.01286	-0.00073	-0.00003	0.00128	-0.00414	-0.00063	-0.00256	-0.00204	0.00026	0.00901	-0.00013	-0.00267	-0.00004	-0.00069	-0.00056
NKE	-0.00029	-0.00003	0.00016	0.00070	0.00028	-0.00015	0.00068	0.00065	0.00024	0.00082	0.00118	0.00110	0.00019	-0.00004	-0.00013	0.00050	0.00028	-0.00056	0.00027	-0.00015
SBUX	-0.00078	0.00139	0.00035	0.00190	0.00088	0.00061	-0.00068	0.00023	-0.00067	0.00015	0.00094	0.00163	0.00094	-0.00041	-0.00267	0.00028	0.00154	-0.00032	0.00051	0.00094
TLT	0.00108	0.00039	-0.00065	-0.00106	0.00036	-0.00041	0.00040	-0.00067	0.00071	0.00066	-0.00176	-0.00187	0.00004	0.00009	-0.00004	-0.00056	-0.00032	0.00116	-0.00016	0.00016
VYM	-0.00015	0.00045	0.00004	0.00064	0.00056	-0.00040	0.00035	0.00034	0.00019	0.00068	0.00060	0.00067	0.00038	-0.00013	-0.00069	0.00027	0.00051	-0.00016	0.00028	0.00022
WM	-0.00086	0.00107	0.00038	0.00023	0.00083	-0.00326	-0.00168	-0.00031	-0.00061	-0.00176	-0.00012	0.00024	0.00044	-0.00042	-0.00056	-0.00015	0.00094	0.00016	0.00022	0.00122

Figure 32

5.7 French & Fama Extension - F_2

stocks	Beta Mkt	Beta SMB	Beta HML	Beta F2	Expected Asset Returns	Asset Returns Variance
AMZN	0.09781947488	0.00385492371	-0.01098472688	-0.00684070992	-0.0109833483	0.04653771858
AZO	1.49035592	0.004455803509	-0.003198785402	0.05940419802	0.3908881232	0.02605462627
BA	0.1216954258	-0.001559692839	0.006405601354	0.04264851832	0.7463082011	0.02570371607
CI	-0.4232762438	0.02380414241	0.005990247867	0.02608292885	1.206675122	0.04783724458
COST	3.553187526	-0.01212414856	-0.001999277491	0.04835163433	-0.1274568477	0.03148173099
CRBP	-11.66197826	0.1044576919	-0.0104862752	-0.1411249856	4.734196954	1.019769336
DATA	2.309934608	-0.001794345526	-0.007030826796	-0.04241195152	-0.3661461173	0.1012313381
F	1.188935107	-0.003787165125	0.004927246381	-0.02573941763	0.02599970242	0.01197206546
GOOG	2.148582292	-0.01432877195	-0.006393661226	-0.08293673253	-0.5424274182	0.03890423958
HUBS	0.4068804571	0.0150939331	-0.01168529181	-0.2211768896	-0.6669799829	0.1794704381
JPM	1.13120759	-0.001168845534	0.01402177742	-0.0292119116	0.2757738472	0.04238492542
MFC	0.1819300501	0.01012165015	0.0143520492	-0.01377196666	0.753395662	0.06504423972
MMM	1.309279417	0.004179854176	-0.0008939599112	0.05117146753	0.4304400657	0.0139290697
MTN	-0.7169867742	0.001601447549	-0.001304473438	-0.03834445154	0.1280461509	0.01862031354
NFLX	0.6331877039	-0.04384845251	0.005167994824	-0.145302956	-0.6271510558	0.1826653003
NKE	1.079447627	-0.00256827593	0.004316079976	0.005093226347	0.2171865141	0.02235244801
SBUX	0.6822162189	0.006827784555	0.00251600715	0.01298570631	0.4123605218	0.02751813557
TLT	0.4407389633	-0.001497716761	-0.009461694433	0.03182239518	0.1283124377	0.02090183911
VYM	1.078699938	-0.001301106847	0.001240406163	0.009164621785	0.1773107809	0.003691830407
WM	1.228376993	-0.004009046279	0.0004885452967	0.06113171418	0.4363338422	0.01544017372

Figure 33

	AMZN	AZO	BA	CI	COST	CRBP	DATA	F	GOOG	HUBS	JPM	MFC	MMM	MTN	NFLX	NKE	SBUX	TLT	VYM	WM
AMZN	1.495E-03	5.480E-04	-9.160E-04	-5.036E-04	4.174E-04	1.788E-03	1.249E-03	-4.903E-04	1.033E-03	2.227E-03	-1.617E-03	-1.633E-03	2.418E-04	1.789E-04	-9.748E-04	-4.683E-04	-1.569E-04	1.182E-03	-6.225E-05	-9.303E-05
AZO	5.480E-04	1.251E-03	-4.660E-05	1.208E-03	1.076E-03	9.480E-04	9.356E-04	6.089E-05	-7.777E-05	5.484E-04	-6.354E-05	2.577E-04	1.060E-03	-3.349E-04	-3.108E-03	2.245E-04	7.021E-04	4.836E-04	4.004E-04	5.256E-04
BA	-9.160E-04	-4.660E-05	7.664E-04	4.385E-04	-9.288E-05	-1.479E-03	-1.041E-03	1.055E-04	-1.126E-03	-2.542E-03	7.443E-04	8.654E-04	9.321E-05	-2.919E-04	-2.281E-04	2.656E-04	1.356E-04	-5.291E-04	5.034E-05	3.511E-04
CI	-5.036E-04	1.208E-03	4.385E-04	3.868E-03	2.499E-04	7.042E-03	6.189E-04	8.002E-04	-1.220E-03	1.748E-03	2.123E-03	3.185E-03	1.305E-03	-3.000E-04	-5.314E-03	7.830E-04	1.720E-03	-9.941E-04	6.392E-04	1.488E-04
COST	4.174E-04	1.076E-03	-9.288E-05	2.499E-04	1.763E-03	-3.692E-03	1.380E-03	4.234E-04	9.175E-04	7.121E-04	2.321E-04	-2.882E-06	9.038E-04	-3.599E-04	-1.023E-03	4.350E-04	5.054E-04	4.671E-04	5.461E-04	6.231E-04
CRBP	1.788E-03	9.480E-04	-1.479E-03	7.042E-03	-3.692E-03	3.369E-02	1.174E-03	-5.449E-04	-2.281E-03	9.622E-03	7.153E-05	3.194E-03	8.774E-04	1.055E-03	-1.173E-02	-7.336E-04	2.151E-03	-6.398E-04	-2.919E-04	-2.083E-03
DATA	1.249E-03	9.356E-04	-1.041E-03	6.189E-04	1.380E-03	1.174E-03	2.714E-03	6.132E-04	2.161E-03	5.127E-03	1.357E-04	-2.605E-06	7.136E-04	1.560E-04	-1.183E-03	2.627E-04	6.665E-04	5.134E-04	5.268E-04	-1.715E-04
F	-4.903E-04	6.089E-05	1.055E-04	8.002E-04	4.234E-04	-5.449E-04	6.132E-04	7.919E-04	4.787E-04	1.404E-03	1.452E-03	1.386E-03	1.899E-04	-2.347E-05	1.659E-04	5.408E-04	5.282E-04	-7.068E-04	3.583E-04	-7.871E-05
GOOG	1.033E-03	-7.777E-05	-1.126E-03	-1.220E-03	9.175E-04	-2.281E-03	2.161E-03	4.787E-04	2.744E-03	4.411E-03	-2.658E-04	-9.792E-04	-2.142E-04	3.989E-04	2.324E-03	7.764E-06	-1.816E-04	4.872E-04	1.933E-04	-4.600E-04
HUBS	2.227E-03	5.484E-04	-2.542E-03	1.748E-03	7.121E-04	9.622E-03	5.127E-03	1.404E-03	4.411E-03	1.337E-02	9.397E-04	9.599E-04	3.629E-04	1.070E-03	-1.150E-03	2.704E-04	1.212E-03	1.589E-05	6.442E-04	-1.671E-03
JPM	-1.617E-03	-6.354E-05	7.443E-04	2.123E-03	2.321E-04	7.153E-05	1.357E-04	1.452E-03	-2.658E-04	9.397E-04	3.242E-03	3.404E-03	3.119E-04	-1.610E-04	-2.568E-04	1.102E-03	1.102E-03	-1.838E-03	5.960E-04	-7.635E-05
MFC	-1.633E-03	2.577E-04	8.654E-04	3.185E-03	-2.882E-06	3.194E-03	-2.605E-06	1.386E-03	-9.792E-04	9.599E-04	3.404E-03	4.000E-03	6.309E-04	-2.276E-04	-2.171E-03	1.129E-03	1.470E-03	-1.953E-03	6.357E-04	-4.701E-05
MMM	2.418E-04	1.060E-03	9.321E-05	1.305E-03	9.038E-04	8.774E-04	7.136E-04	1.899E-04	-2.142E-04	3.629E-04	3.119E-04	6.309E-04	9.441E-04	-3.213E-04	-2.762E-03	3.118E-04	7.233E-04	1.897E-04	3.975E-04	4.615E-04
MTN	1.789E-04	-3.349E-04	-2.919E-04	-3.000E-04	-3.599E-04	1.055E-03	1.560E-04	-2.347E-05	3.989E-04	1.070E-03	-1.610E-04	-2.276E-04	-3.213E-04	2.312E-04	7.395E-04	-1.500E-04	-1.765E-04	-1.396E-05	-1.256E-04	-3.420E-04
NFLX	-9.748E-04	-3.108E-03	-2.281E-04	-5.314E-03	-1.023E-03	-1.173E-02	-1.183E-03	1.659E-04	2.324E-03	-1.150E-03	-2.568E-04	-2.171E-03	-2.762E-03	7.395E-04	1.127E-02	-4.027E-04	-2.302E-03	-5.293E-04	-7.678E-04	-9.373E-04
NKE	-4.683E-04	2.245E-04	2.656E-04	7.830E-04	4.350E-04	-7.336E-04	2.627E-04	5.408E-04	7.764E-06	2.704E-04	1.102E-03	1.129E-03	3.118E-04	-1.500E-04	-4.027E-04	4.510E-04	4.730E-04	-5.037E-04	3.059E-04	1.364E-04
SBUX	-1.569E-04	7.021E-04	1.356E-04	1.720E-03	5.054E-04	2.151E-03	6.665E-04	5.282E-04	-1.816E-04	1.212E-03	1.102E-03	1.470E-03	7.233E-04	-1.765E-04	-2.302E-03	4.730E-04	8.706E-04	-4.031E-04	4.172E-04	1.443E-04
TLT	1.182E-03	4.836E-04	-5.291E-04	-9.941E-04	4.671E-04	-6.398E-04	5.134E-04	-7.068E-04	4.872E-04	1.589E-05	-1.838E-03	-1.953E-03	1.897E-04	-1.396E-05	-5.293E-04	-5.037E-04	-4.031E-04	1.265E-03	-1.458E-04	2.310E-04
VYM	-6.225E-05	4.004E-04	5.034E-05	6.392E-04	5.461E-04	-2.919E-04	5.268E-04	3.583E-04	1.933E-04	6.442E-04	5.960E-04	6.357E-04	3.975E-04	-1.256E-04	-7.678E-04	3.059E-04	4.172E-04	-1.458E-04	2.773E-04	1.522E-04
WM	-9.303E-05	5.256E-04	3.511E-04	1.488E-04	6.231E-04	-2.083E-03	-1.715E-04	-7.871E-05	-4.600E-04	-1.671E-03	-7.635E-05	-4.701E-05	4.615E-04	-3.420E-04	-9.373E-04	1.364E-04	1.443E-04	2.310E-04	1.522E-04	5.473E-04

Figure 34

5.8 French & Fama Extension - F_1 & F_2

stocks	Beta Mkt	Beta SMB	Beta HML	Beta F1	Beta F2	Expected Asset Returns	Asset Returns Variance
AMZN	-0.07110911267	0.007209026411	-0.03054339948	-0.02326505387	0.08874699566	0.260883525	0.04653771858
AZO	1.597456873	0.00232929786	0.00920143948	0.01475007561	-0.001198536586	0.1945857529	0.02605462627
BA	0.1191641454	-0.001509433885	0.00611252791	-0.0003486110533	0.04408083526	0.7524406206	0.02570371607
CI	-0.4109525436	0.02355945347	0.00741709469	0.001697235231	0.01910960214	1.170018302	0.04783724458
COST	3.588502506	-0.01282533282	0.002089516048	0.004863622591	0.02836876531	-0.1716179549	0.03148173099
CRBP	-11.89226047	0.1090299801	-0.0371485141	-0.03171475063	-0.01082052707	6.628540156	1.019769336
DATA	1.984591976	0.004665381735	-0.0446992293	-0.04480658944	0.1416821453	0.02494960178	0.1012313381
F	1.143424897	-0.002883553011	-0.0003419582542	-0.006267722363	0.00001239035496	0.09568387109	0.01197206546
GOOG	2.134253285	-0.01404426735	-0.008052683863	-0.001973408513	-0.07482870952	-0.5321699003	0.03890423958
HUBS	0.4365358246	0.01450512129	-0.008251772166	0.004084173861	-0.2379572852	-0.6823492155	0.1794704381
JPM	1.127974367	-0.001104649388	0.01364743248	-0.0004452835598	-0.02738240224	0.281650901	0.04238492542
MFC	0.2478325767	0.008813145498	0.02198229084	0.009076177419	-0.05106270237	0.5999247012	0.06504423972
MMM	1.293619931	0.004490775656	-0.002707026469	-0.002156643857	0.06003233839	0.4623066313	0.0139290697
MTN	-0.7254233775	0.001768957587	-0.002281269401	-0.001161899438	-0.0335706263	0.1417928067	0.01862031354
NFLX	0.7484508554	-0.04613701985	0.01851324489	0.01587418369	-0.2105242448	-0.6894903193	0.1826653003
NKE	1.014637894	-0.001281468863	-0.003187637222	-0.00892567652	0.0417656082	0.3346842413	0.02235244801
SBUX	0.8553207873	0.003390767122	0.0225581778	0.02384017511	-0.08496496477	0.1037416651	0.02751813557
TLT	0.4174539762	-0.001035389814	-0.01215764688	-0.003206837201	0.04499813128	0.1666349681	0.02090183911
VYM	1.091845996	-0.001562123833	0.002762466233	0.001810491449	0.001725966171	0.1553480396	0.003691830407
WM	1.364670415	-0.006715172826	0.01626869512	0.01877049855	-0.01598948654	0.1837268003	0.01544017372

Figure 35

	AMZN	AZO	BA	CI	COST	CRBP	DATA	F	GOOG	HUBS	JPM	MFC	MMM	MTN	NFLX	NKE	SBUX	TLT	VYM	WM
AMZN	0.00256	-0.00013	-0.00090	-0.00058	0.00020	0.00324	0.00330	-0.00020	0.00112	0.00204	-0.00160	-0.00205	0.00034	0.00023	-0.00170	-0.00006	-0.00125	0.00133	-0.00014	-0.00095
AZO	-0.00013	0.00168	-0.00006	0.00126	0.00122	0.00003	-0.00036	-0.00012	-0.00013	0.00067	-0.00008	0.00052	0.00100	-0.00037	-0.00265	-0.00003	0.00139	0.00039	0.00045	0.00107
BA	-0.00090	-0.00006	0.00077	0.00044	-0.00010	-0.00146	-0.00101	0.00011	-0.00112	-0.00254	0.00074	0.00086	0.00009	-0.00029	-0.00024	0.00027	0.00012	-0.00053	0.00005	0.00034
CI	-0.00058	0.00126	0.00044	0.00387	0.00027	0.00694	0.00047	0.00078	-0.00123	0.00176	0.00212	0.00322	0.00130	-0.00030	-0.00526	0.00075	0.00180	-0.00100	0.00065	0.00021
COST	0.00020	0.00122	-0.00010	0.00027	0.00181	-0.00399	0.00095	0.00036	0.00090	0.00075	0.00023	0.00008	0.00088	-0.00037	-0.00087	0.00035	0.00073	0.00044	0.00056	0.00080
CRBP	0.00324	0.00003	-0.00146	0.00694	-0.00399	0.03567	0.00397	-0.00015	-0.00216	0.00937	0.00010	0.00263	0.00101	0.00113	-0.01272	-0.00018	0.00067	-0.00044	-0.00040	-0.00325
DATA	0.00330	-0.00036	-0.00101	0.00047	0.00095	0.00397	0.00666	0.00116	0.00233	0.00477	0.00017	-0.00080	0.00090	0.00026	-0.00258	0.00105	-0.00143	0.00080	0.00037	-0.00182
F	-0.00020	-0.00012	0.00011	0.00078	0.00036	-0.00015	0.00116	0.00087	0.00050	0.00135	0.00146	0.00127	0.00022	-0.00001	-0.00003	0.00065	0.00023	-0.00067	0.00034	-0.00031
GOOG	0.00112	-0.00013	-0.00112	-0.00123	0.00090	-0.00216	0.00233	0.00050	0.00275	0.00440	-0.00026	-0.00101	-0.00021	0.00040	0.00226	0.00004	-0.00027	0.00050	0.00019	-0.00053
HUBS	0.00204	0.00067	-0.00254	0.00176	0.00075	0.00937	0.00477	0.00135	0.00440	0.01341	0.00094	0.00103	0.00035	0.00106	-0.00102	0.00020	0.00140	-0.00001	0.00066	-0.00152
JPM	-0.00160	-0.00008	0.00074	0.00212	0.00023	0.00010	0.00017	0.00146	-0.00026	0.00094	0.00324	0.00340	0.00031	-0.00016	-0.00027	0.00111	0.00108	-0.00184	0.00059	-0.00009
MFC	-0.00205	0.00052	0.00086	0.00322	0.00008	0.00263	-0.00080	0.00127	-0.00101	0.00103	0.00340	0.00416	0.00059	-0.00025	-0.00189	0.00097	0.00190	-0.00201	0.00067	0.00029
MMM	0.00034	0.00100	0.00009	0.00130	0.00088	0.00101	0.00090	0.00022	-0.00021	0.00035	0.00031	0.00059	0.00095	-0.00032	-0.00283	0.00035	0.00062	0.00020	0.00039	0.00038
MTN	0.00023	-0.00037	-0.00029	-0.00030	-0.00037	0.00113	0.00026	-0.00001	0.00040	0.00106	-0.00016	-0.00025	-0.00032	0.00023	0.00070	-0.00013	-0.00023	-0.00001	-0.00013	-0.00038
NFLX	-0.00170	-0.00265	-0.00024	-0.00526	-0.00087	-0.01272	-0.00258	-0.00003	0.00226	-0.00102	-0.00027	-0.00189	-0.00283	0.00070	0.01176	-0.00068	-0.00156	-0.00063	-0.00071	-0.00035
NKE	-0.00006	-0.00003	0.00027	0.00075	0.00035	-0.00018	0.00105	0.00065	0.00004	0.00020	0.00111	0.00097	0.00035	-0.00013	-0.00068	0.00061	0.00005	-0.00045	0.00027	-0.00019
SBUX	-0.00125	0.00139	0.00012	0.00180	0.00073	0.00067	-0.00143	0.00023	-0.00027	0.00140	0.00108	0.00190	0.00062	-0.00023	-0.00156	0.00005	0.00199	-0.00055	0.00050	0.00102
TLT	0.00133	0.00039	-0.00053	-0.00100	0.00044	-0.00044	0.00080	-0.00067	0.00050	-0.00001	-0.00184	-0.00201	0.00020	-0.00001	-0.00063	-0.00045	-0.00055	0.00129	-0.00016	0.00011
VYM	-0.00014	0.00045	0.00005	0.00065	0.00056	-0.00040	0.00037	0.00034	0.00019	0.00066	0.00059	0.00067	0.00039	-0.00013	-0.00071	0.00027	0.00050	-0.00016	0.00028	0.00022
WM	-0.00095	0.00107	0.00034	0.00021	0.00080	-0.00325	-0.00182	-0.00031	-0.00053	-0.00152	-0.00009	0.00029	0.00038	-0.00038	-0.00035	-0.00019	0.00102	0.00011	0.00022	0.00124

Figure 36

5.9 Recap

To compare the different factor models, we utilized the matrix norm approach provided in the project guidelines. This was implemented using the `numpy.linalg.norm` function from the numpy library in Python, which uses the Frobenius norm for unordered norms [16].

	CAPM + F1	CAPM + F2	CAPM + F1 + F2	FF + F1	FF + F2	FF + F1 + F2
CAPM + F1	-	0.01612839349	0.01580362711	0.04727571084	0.04526575934	0.04844141244
CAPM + F2	0.01612839349	-	0.008747349727	0.04640838613	0.04287255482	0.04610362816
CAPM + F1 + F2	0.01580362711	0.008747349727	-	0.04529557746	0.04159542112	0.04491693634
FF + F1	0.04727571084	0.04640838613	0.04529557746	-	0.01190094569	0.00975880674
FF + F2	0.04526575934	0.04287255482	0.04159542112	0.01190094569	-	0.01024218435
FF + F1 + F2	0.04844141244	0.04610362816	0.04491693634	0.00975880674	0.01024218435	-

Figure 37

5.10 Portfolio Optimization

Based on the results we have seen, the French-Fama models seemed to be unreasonable and there may be some error in our code implementation. Therefore, we chose the factor models CAPM & F1 and CAPM & F2 for demonstration purposes in order to continue our work in portfolio optimization.

For the purpose of backtesting, we utilized the portfolio backtesting capabilities on the website `Portfoliovisualizer.com`. This tool allows you to "construct one or more portfolios based on the selected mutual funds, ETFs and stocks to analyze and backtest portfolio returns, risk characteristics, standard deviation, annual returns and rolling returns. The results include a visualization of the portfolio growth chart and rolling returns, CAGR, standard deviation, Sharpe ratio, Sortino ratio, annual returns and inflation adjusted returns" [17].

Taking our chosen models of "CAPM & F1" and "CAPM & F2" (and the associated expected asset returns as well as covariance returns) and also using our previous work in our Markowitz Efficient Frontier project, we formed optimal portfolios and observed their respective weights for a target portfolio overall return of 10%; we decided on a target return of 10% because this is the average annual return for the S&P 500 index [18].

Our code that calculates the optimal weights takes in a vector of expected asset returns, covariance matrix of asset returns, and a target return value. Due to the nature of our "CAPM & F1" and "CAPM & F2" models, we got a weird assortment of expected asset returns; the results can be found below in Table 3. We believe it is because of our chosen factors and especially also because of how the covariance matrices of asset returns gets calculated.

Table 3: Factor Based Portfolio Weights

Stock	CAPM+F1 Weights	CAPM+F2 Weights
AMZN	-1.28360	-0.03588
AZO	-2.56679	-0.69166
BA	-2.56025	0.36103
CI	-0.76878	0.33432
COST	0.41914	-0.15293
CRBP	-1.69795	-0.03873
DATA	2.10227	-0.91175
F	-0.19794	-0.39646
GOOG	0.42695	1.20205
HUBS	-0.21813	0.05860
JPM	0.07241	-0.08883
MFC	-1.20424	0.49460
MMM	1.95534	-0.10846
MTN	0.51758	-0.65460
NFLX	2.01097	-0.22027
NKE	-1.75463	0.92387
SBUX	3.70834	0.15667
TLT	-2.66854	1.38948
VYM	1.82389	-0.42545
WM	2.88396	-0.19560

As we can see, the CAPM + F1 model has a very large amount of leverage in it, while the CAPM + F2 still has some large, but much more reasonable weights. We believe that the excessive leveraging may be a result of the covariance matrix estimates. We took these weights and plugged them into Portfolio Visualizer to complete the backtesting.

5.10.1 Backtesting Results

The following table provides a brief summary of the performance of our two portfolios, as well as several metrics used previously in this report:

Table 4: CAPM+F1 & CAPM+F2 Portfolio Performance Overview

Portfolio	Initial Balance	Final Balance	Stdev	Max.DD	Sharpe Ratio	Sortino Ratio
CAPM+F1	$500,000	$1,658,145	96.22%	-36.64%	1.96	5.6
CAPM+F2	$500,000	$416,926	40.22%	-34.59%	-0.37	-0.51

As we can see, there is a large difference in the performance of the two portfolios. The extremely high leveraged CAPM+F1 model had an extremely high return, while the CAPM+F2 model ended up losing money. This is reflected in the Sharpe and Sortino Ratios

for the two portfolios. The following plots show the growth of the two portfolios over the backtesting period, as well as the overall annualized returns:

Figure 38: Monthly Portfolio Value

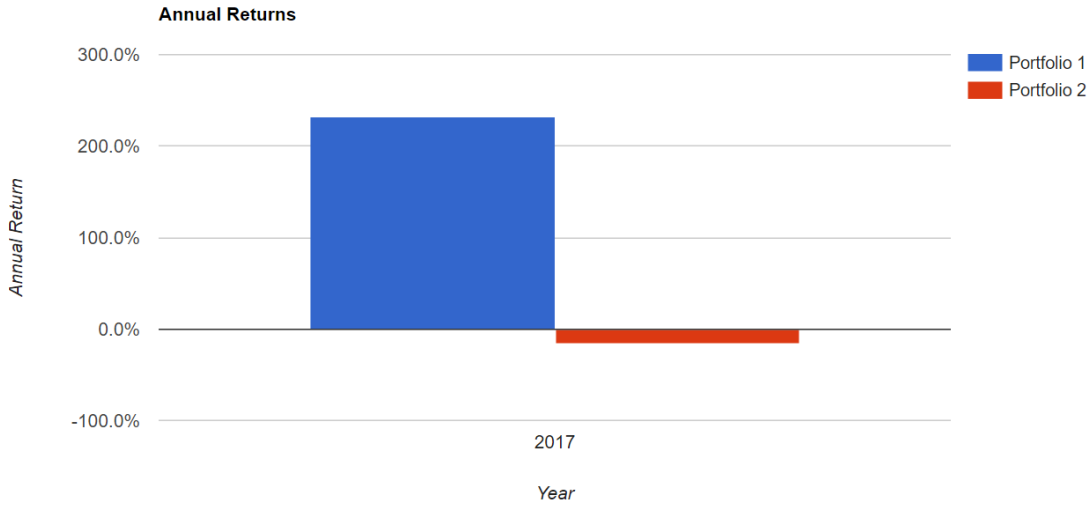

Figure 39: Annualized Portfolio Returns

The full results of our portfolio backtesting can be accessed here: `https://goo.gl/p2vHi9`, as well as in the Excel file *PV_Backtesting.csv* included in the *.zip*-file associated with this report. It is very interesting and we recommend taking a look.

To compare our performance and models, we looked at the actual annualized expected returns and covariance returns for our 20 risky assets, for a holding period between Jan

1, 2017 through Oct 31, 2017. We then utilized the same Frobenius norm built-in Python function to calculate the difference between covariances; all these results can be found below:

Table 5: Actual Expected Asset Returns (1/1/2017 - 10/31/2017)

Stock	Actual Expected Returns
AMZN	0.62346
AZO	-0.27282
BA	0.89932
CI	0.58733
COST	0.090401
CRBP	-0.03929
DATA	1.21599
F	0.05263
GOOG	0.38032
HUBS	1.18401
JPM	0.23655
MFC	0.20381
MMM	0.40478
MTN	0.58115
NFLX	0.75294
NKE	0.11041
SBUX	0.02053
TLT	0.07766
VYM	0.13281
WM	0.23527

Actual annualized covariance returns

	AMZN	AZO	BA	CI	COST	CRBP	DATA	F	GOOG	HUBS	JPM	MFC	MMM	MTN	NFLX	NKE	SBUX	TLT	VYM	WM
AMZN	0.0476669	-0.0053024	0.0055368	0.0079857	-0.0006210	0.0196055	0.0057371	-0.0032367	0.0229151	0.0281806	0.0026765	0.0058544	0.0039161	0.0082917	0.0247400	-0.0012577	0.0060232	-0.0009345	0.0030166	0.0032646
AZO	-0.0053024	0.0713333	0.0013940	0.0006698	0.0127008	-0.0008282	0.0015407	0.0115061	-0.0043348	-0.0037196	0.0005003	0.0051681	0.0000337	-0.0002541	-0.0056158	0.0100374	0.0079417	-0.0005453	0.0025222	0.0030527
BA	0.0055368	0.0013940	0.0325129	0.0037813	-0.0007846	0.0077578	0.0094674	0.0036812	0.0058293	0.0130640	0.0067357	0.0061069	0.0042632	0.0061336	0.0080162	-0.0038520	0.0046582	-0.0038224	0.0042278	0.0026138
CI	0.0079857	0.0006698	0.0037813	0.0224099	0.0011110	0.0003934	0.0088024	0.0046952	0.0064418	0.0130989	0.0054217	0.0057674	0.0026216	0.0039990	0.0040895	0.0037488	0.0008440	-0.0023357	0.0030248	0.0035592
COST	-0.0006210	0.0127008	-0.0007846	0.0011110	0.0356838	0.0034485	-0.0015728	0.0027368	0.0021965	-0.0006266	0.0023339	0.0016262	0.0005427	0.0012833	-0.0017178	0.0081469	0.0037525	-0.0002237	0.0024015	0.0026438
CRBP	0.0196055	-0.0008282	0.0077578	0.0003934	0.0034485	0.3431371	0.0258308	0.0213331	0.0261658	0.0615891	0.0129844	0.0285536	-0.0018788	0.0177044	0.0205636	0.0198687	0.0046893	-0.0046656	0.0094126	0.0010124
DATA	0.0057371	0.0015407	0.0094674	0.0088024	-0.0015728	0.0258308	0.0831104	0.0122560	0.0127746	0.0394883	0.0135585	0.0119618	0.0037317	0.0163108	0.0196363	0.0018316	0.0122003	-0.0023073	0.0063726	0.0043361
F	-0.0032367	0.0115061	0.0036812	0.0046952	0.0027368	0.0213331	0.0122560	0.0334117	0.0020068	0.0126661	0.0112663	0.0140860	0.0027775	0.0046304	0.0019670	0.0059637	0.0028418	-0.0051170	0.0057466	0.0039972
GOOG	0.0229151	-0.0043348	0.0058293	0.0064418	0.0021965	0.0261658	0.0127746	0.0020068	0.0247864	0.0287156	0.0054969	0.0072410	0.0054646	0.0085071	0.0218726	0.0006751	0.0033153	-0.0010115	0.0033609	0.0026855
HUBS	0.0281806	-0.0037196	0.0130640	0.0130989	-0.0006266	0.0615891	0.0394883	0.0126661	0.0287156	0.1348474	0.0116325	0.0206258	0.0072599	0.0253142	0.0351262	0.0105196	0.0090912	-0.0067674	0.0084688	0.0061825
JPM	0.0026765	0.0005003	0.0067357	0.0054217	0.0023339	0.0129844	0.0135585	0.0112663	0.0054969	0.0116325	0.0261630	0.0177420	0.0034443	0.0033905	0.0059414	0.0044654	0.0038092	-0.0088804	0.0060093	0.0034466
MFC	0.0058544	0.0051681	0.0061069	0.0057674	0.0016262	0.0285536	0.0119618	0.0140860	0.0072410	0.0206258	0.0177420	0.0350953	0.0014509	0.0051999	0.0088569	0.0057023	0.0047691	-0.0070731	0.0066543	0.0030952
MMM	0.0039161	0.0000337	0.0042632	0.0026216	0.0005427	-0.0018788	0.0037317	0.0027775	0.0054646	0.0072599	0.0034443	0.0014509	0.0167038	0.0024457	0.0054732	0.0025101	0.0018713	-0.0003913	0.0030783	0.0016815
MTN	0.0082917	-0.0002541	0.0061336	0.0039990	0.0012833	0.0177044	0.0163108	0.0046304	0.0085071	0.0253142	0.0033905	0.0051999	0.0024457	0.0290167	0.0127570	0.0019348	0.0027838	-0.0009498	0.0036475	0.0033045
NFLX	0.0247400	-0.0056158	0.0080162	0.0040895	-0.0017178	0.0205636	0.0196363	0.0019670	0.0218726	0.0351262	0.0058414	0.0088569	0.0054732	0.0127570	0.0825444	0.0048330	0.0036353	-0.0012960	0.0043898	0.0034635
NKE	-0.0012577	0.0100374	-0.0038520	0.0037488	0.0081469	0.0198687	0.0018316	0.0059637	0.0006751	0.0105196	0.0044654	0.0057023	0.0025101	0.0019348	0.0048330	0.0590087	0.0038553	-0.0022434	0.0028341	0.0035802
SBUX	0.0060232	0.0079417	0.0046582	0.0008440	0.0037525	0.0046893	0.0122003	0.0028418	0.0033153	0.0090912	0.0038092	0.0047691	0.0018713	0.0027838	0.0036353	0.0038553	0.0307970	-0.0008911	0.0033397	0.0043495
TLT	-0.0009345	-0.0005453	-0.0038224	-0.0023357	-0.0002237	-0.0046656	-0.0023073	-0.0051170	-0.0010115	-0.0067674	-0.0088804	-0.0070731	-0.0003913	-0.0009498	-0.0012960	-0.0022434	-0.0008911	0.0092664	-0.0020985	-0.0013612
VYM	0.0030166	0.0025222	0.0042278	0.0030248	0.0024015	0.0094126	0.0063726	0.0057466	0.0033609	0.0084688	0.0060093	0.0066543	0.0030783	0.0036475	0.0043898	0.0028341	0.0033397	-0.0020985	0.0039575	0.0030535
WM	0.0032646	0.0030527	0.0026138	0.0035592	0.0026438	0.0010124	0.0043361	0.0039972	0.0026855	0.0061825	0.0034466	0.0030952	0.0016815	0.0033045	0.0034835	0.0035802	0.0043495	-0.0013612	0.0030535	0.0123553

Figure 40

	Actual Cov	CAPM + F1	CAPM + F2
Actual Cov	-	0.4376514213	0.4233969401
CAPM + F1	0.4376514213	-	0.2054495368
CAPM + F2	0.4233969401	0.2054495368	-

Figure 41

Finally, after analyzing all of the above tables and comparisons, one thing we can immediately deduce is that our "CAPM & F1" chosen factor model's covariance of asset returns is more aligned with the actual 10-month holding-period covariance of returns.

Without a doubt, the worst performing portfolio was the "CAPM & F2" factor model while the best performing was the "CAPM & F1" model. We believe this is the case due to our chosen factors of "global ex US momentum factor" and "adjusted retail US sales." It is conceivable that because of the volatility for certain assets in our portfolio (namely, NFLX, HUBS, and CRBP) and because of the global international presence of some of our assets (namely, JPM, MFC, and AMZN) that our factor models gave the results that they did.

5.11 Quantile-Based Portfolio Back-testing

For this portion of the project, we first ranked our 20 risky assets in order from best to worst for each of our previously chosen factor models. How did we rank our assets? We ordered in decreasing order the "expected asset returns" and then looked at the increasing order of "variance asset returns." This means we consider both the reward and risk aspects of our 20 assets. We ended up with the following ranking:

Table 6: Asset Ranking (best-to-worst)

CAPM+F1	CAPM+F2
HUBS	WM
DATA	BA
CI	TLT
MFC	AZO
JPM	MMM
COST	COST
SBUX	AMZN
AZO	VYM
MMM	NKE
GOOG	SBUX
F	MTN
CRBP	CI
NKE	F
VYM	MFC
AMZN	NFLX
WM	JPM
MTN	DATA
BA	GOOG
TLT	CRBP
NFLX	HUBS

From this table then, we deduce the assets used in the quintile-based backtesting, by taking the five best and five worst assets from each model. The following table shows the assets used and their respective positions in our backtested models:

Table 7: Top and Bottom Quintile Assets for Each Model

	CAPM+F1	**CAPM+F2**
Position	Asset	Asset
Long	HUBS	WM
Long	DATA	BA
Long	CI	TLT
Long	MFC	AZO
Long	JPM	MMM
Short	WM	JPM
Short	MTN	DATA
Short	BA	GOOG
Short	TLT	CRBP
Short	NFLX	HUBS

Given that each of these portfolios has 10 assets and the project guidelines call for equal weights in each of the long and short positions, we will backtest each of these portfolios with 25% in each long asset and 5% in each short asset. This will allow us to invest 100% of our portfolio, and are not unreasonable figures for equity weights. Once more, we will utilize PortfolioVisualizer for backtesting. The backtesting date range is also the same as before: January 1, 2017 to October 31, 2017.

5.11.1 Backtesting Results

Table 8: CAPM+F1 & CAPM+F2 Quintile Backtesting Summary

Portfolio	Initial Bal.	Final Bal.	Stdev	Return	Max.DD	Sharpe Ratio	Sortino Ratio
CAPM+F1	$500,000	$754,189	12.70%	50.84%	-2.86%	3.59	10.91
CAPM+F2	$500,000	$622,149	10.74%	24.43%	-1.82%	2.23	10.04

This table has several interesting results. First of all, both portfolios performed very well, with the CAPM+F1 portfolio returning 50.84% annually, and the CAPM+F2 portfolio returning 24.43% annually. Both of these are admirable results. The CAPM+F1 portfolio is more volatile than the CAPM+F2 portfolio, as indicated by the higher standard deviation. Furthermore, the CAPM+F1 portfolio retracted significantly more than the CAPM+F2 portfolio, with a maximum drawdown of -2.86% versus -1.82%. Both the Sharpe and Sortino Ratios indicate that these were successful portfolios, but the Sortino Ratio tells us a bit more. Despite the fact that the return on the CAPM+F1 portfolio was double that of the CAPM+F2 portfolio, the Sortino Ratios are very similar. This indicates that the CAPM+F2 portfolio achieves its returns with more "efficiency" than the CAPM+F1 portfolio, as it has much less downside volatility. So for a more risk-averse approach, the CAPM+F2 portfolio is significantly better than the CAPM+F1 portfolio.

The plots below show the growth throughout the backtesting period of our two quintile CAPM+F1 portfolio and the CAPM+F2 portfolio, respectively.

Figure 42: CAPM+F1 Quintile Portfolio Growth

Figure 43: CAPM+F2 Quintile Portfolio Growth

The full results of our CAPM+F1 quintile backtesting can be found here: `https://goo.gl/PeUA2F` and in the Excel file *PV_Backtesting2.csv*. The full results of our CAPM+F2 quintile backtesting can be found here: `https://goo.gl/pvy2UH` and in the Excel file *PV_Backtesting3.csv*.

Lastly, if we compare these two models to the Markowitz optimized portfolios from the previous section, we see that both of our quintile models underperform the CAPM+F1 model,

but outperform the CAPM+F2 model. However, that said, the CAPM+F1 results from the prior section are rather extreme, both in terms of return and volatility. This is primarily due to the highly leveraged weighting scheme computed by the Markowitz optimization. As such, we would be very apprehensive about utilizing this model in the future as it has an extreme amount of downside potential. The CAPM+F2 model from before did lose money, so we are not particularly thrilled with it, considering the state of the stock market during the backtesting period. As such, if we had to choose a model, we would be fairly happy with either of the two quintile models.

If we were seeking to maximize returns, we could use the CAPM+F1 model, and if we wanted to be a bit more conservative, we could use the CAPM+F2 model. Overall, both of these models seemed more reasonable than the models in the prior section. We ultimately believe this to be due to the fact that we are using an unconstrained Markowitz optimization, and would be curious to see how the results change if we were to implement a constrained Markowitz routine to the model construction.

6 Conclusion

Overall, this set of projects was extremely educational and we are happy to have had the opportunity to get hands-on experience in this area. There are certainly areas we could improve in our implementation and analysis- however, we do believe that the fundamentals of our work are solid and substantial.

All related project materials, including code, plots, and data, can be found in the .zip-file associated with this project. For any questions, please contact the authors of this paper at akshoop@wpi.edu (Alexander Shoop) and kmdymov@wpi.edu (Khasan Dymov).

Appendix A Portfolio Rebalancer.py

```
1
2 #!/usr/bin/env python3
3 # -*- coding: utf-8 -*-
4 """
5 @author: Khasan Dymov and Alex Shoop
6 """
7
8 import numpy as np
9 import matplotlib.pyplot as plt
10 from pandas_datareader import data
11 from datetime import datetime
12
13 ##############################################################################################
14 #setting output np options
15 np.set_printoptions(threshold = np.inf)
16
17 #Import specified date range of prices data,
18 #and calculated returns (log returns)
19
20 #Specified tickers
21 tickers = ['AMZN','AZO','BA','CI','COST','CRBP','DATA','F','GOOG','HUBS',\
22            'JPM','MFC','MMM','MTN','NFLX','NKE','SBUX','TLT','VYM','WM']
23
24
25 #Specified date range
26 startDate = datetime(2015,11,24)
27 endDate = datetime(2017,11,24)
28
29 #Use pandas_reader.data.DataReader to load desired data
30 securityRawData = data.DataReader(tickers, 'yahoo', startDate, endDate)
31
32 #Getting adjusted close prices
33 close = securityRawData['Adj Close']
34 #fixing date ordering
35 close = close.iloc[::-1]
36 #getting log returns
37 dailyLogReturns = np.log(close).diff()
38 #removing the 1st row, of empty values for first date
39 dailyLogReturns = dailyLogReturns.drop(startDate)
40 #Pulling just the last day closing prices for rebalancing function
41 finalprices = np.matrix([close.iloc[::-1].iloc[0]]).T
42
43 #Gather expected returns, and covariance matrix
44 mu2yrsdaily = dailyLogReturns.mean()
45 var2yrsdaily = dailyLogReturns.var()
46 sd2yrsdaily = np.sqrt(var2yrsdaily)
47 cov2yrsdaily = dailyLogReturns.cov()
48
49 #Annualize parameter estimates
```

```
50
51 #2 year lookback:
52 muannual = np.sqrt(((1+mu2yrsdaily)**505))-1
53 varannual = np.sqrt(252)*var2yrsdaily
54 sdannual = 252*sd2yrsdaily
55 covannual = 252*cov2yrsdaily
56
57 # OPTIMAL PORTFOLIO FUNCTION
58 def optimalPortfolio(mu, omega, muP):
59     # We follow by example
60     # the process as explained in Statistics & Finance by David Ruppert
61     # Necessary coefficients for risk-efficient portfolio
62     n1 = np.matrix(np.ones_like(mu)).T
63     n1t = n1.T
64     invomega = np.linalg.inv(omega)
65     muR = np.matrix(mu).T
66     mut = muR.T
67
68     A = np.matrix.item(n1t*invomega*muR)
69     B = np.matrix.item(mut*invomega*muR)
70     C = np.matrix.item(n1t*invomega*n1)
71     D = B*C-A**2
72
73
74     # Fixed vectors for efficient weights calculation
75     g = (B*invomega*n1 - A*invomega*muR)/D
76     h = (C*invomega*muR - A*invomega*n1)/D
77
78     # The final formula:
79     wts = g + muP*h
80
81     # Our efficient weights
82     return wts
83
84 #Step 1: Find the minimum variance portfolio for our problem
85
86 n1 = np.matrix(np.ones_like(muannual)).T
87 n1t = n1.T
88 invomega = np.linalg.inv(covannual)
89 muR = np.matrix(muannual).T
90 mut = muR.T
91
92 A = np.matrix.item(n1t*invomega*muR)
93 B = np.matrix.item(mut*invomega*muR)
94 C = np.matrix.item(n1t*invomega*n1)
95 D = B*C-A**2
96
97 #Need to use specific g,h for minimum variance portfolio
98
99 gmin = (B*invomega*n1 - A*invomega*muR)/D
100 hmin = (C*invomega*muR - A*invomega*n1)/D
```

```
g1 = gmin.T*np.matrix(covannual)*gmin
h1 = hmin.T*np.matrix(covannual)*hmin
g1h1 = gmin.T*np.matrix(covannual)*hmin

mumin = np.matrix.item(- g1h1/h1)
sigmamin = np.matrix.item(np.sqrt(g1 - g1h1**2/h1))

#Weights associated with this minimum variance portfolio
wtsmin = np.array(gmin + mumin*hmin)
barX = np.arange(np.size(wtsmin))

#Step 2: Calculate the efficient frontier

muP = np.linspace(min(muannual), max(muannual), 100)
sigmaP = np.zeros(0)

for target in muP:
    wtsP = optimalPortfolio(muannual, covannual, target)
    sigmaP = np.append(sigmaP, np.sqrt(wtsP.T*np.matrix(covannual)*wtsP))

effindices = np.where(muP>mumin)
ineffindices = np.where(muP<mumin)

#Step 2.5: Calculate the tangency portfolio
muF = 0.02
omegabar = invomega*(muR - muF*n1)
omegaT = omegabar/(n1t*omegabar)
sigmaT = np.matrix.item(np.sqrt(omegaT.T*np.matrix(covannual)*omegaT))
muT = np.matrix.item(mut*omegaT)

#Step 3: Plot minimum variance portfolio and upper section of efficient
    ↪ frontier

plt.plot(sigmaP[effindices], muP[effindices], 'g', label = "Efficient Frontier
    ↪ ")
plt.plot(sigmaP[ineffindices], muP[ineffindices], 'c--')
plt.plot(sigmamin, mumin, 'ro', label = "Minimum Variance Portfolio")
plt.plot(sigmaT, muT, 'bx', markersize = 8, label = "Tangency Portfolio")
plt.plot([0,sigmaT],[muF,muT], 'b--')
plt.plot(0,muF,'bo',label = "Risk-free rate")
plt.title("Efficient Frontier for 20 risky assets (2yrs)")
plt.xlabel(r"$risk = \sigma_r$")
plt.ylabel(r"$reward = \mu_r$")
plt.legend()


#Step 4: Bar chart of weights

#plt.bar(barX, (wtsmin)*100)
#plt.axhline(y = 0, linewidth = 0.5, color = 'black')
#plt.xticks(barX, stocks, rotation=60)
```

```
#plt.ylabel("Percentage")
#plt.title("Weights for min var portfolio (2yrs)")

#Step 5: Bar chart of weights in tangency portfolio
wtstangent = optimalPortfolio(muannual, covannual, muT)

#plt.bar(barX, (wtstangent)*100)
#plt.axhline(y = 0, linewidth = 0.5, color = 'black')
#plt.xticks(barX, stocks, rotation=60)
#plt.ylabel("Percentage")
#plt.title("Weights for Initial Tangency Portfolio")

print("Expected return = " + str(muT))
print ("Portfolio std. dev = " + str(sigmaT))
print("The weights of the assets in the tangency portfolio are " + str(
    ↪ wtstangent))


#Rebalancing function

def PortfolioRebalancer(currwts, newwts, currcashval, lastcloseprices, tickers
    ↪ ):
    #function takes as input current stock weights in portfolio,
    #new calculated tangency portfolio weights, the current cash value
    #of the portfolio, latest closing prices for all the stocks, and tickers

    newcashvalue = np.array(np.divide(np.multiply(newwts, currcashval),
    ↪ currwts))
    cashvalchange = np.array(np.subtract(newcashvalue, currcashval))
    sharechange = np.array(np.divide(cashvalchange, lastcloseprices))


    for x in range(0,20):
        print("Adjust position in " + tickers[x] + " by " + str(sharechange[x
    ↪ ]) + " shares.")

#Current weights and cash value pasted from Excel file (may automate in future
    ↪ )
#MUST CONVERT TO MATRIX, THEN TRANSPOSE!
currwts = np.matrix([0.1209828446,
-0.1262697252,
0.2820707935,
0.109136186,
-0.004279442142,
0.05482424766,
-0.0438397217,
-0.2276068822,
-0.02123034674,
0.01040946642,
0.3374382686,
-0.01833587159,
```

```
0.3876116784,
0.3434809172,
-0.009318535294,
-0.03576921161,
-0.2293183649,
0.3503882697,
-0.8579834182,
0.5776088474]).T

currcashval = np.matrix([53370,
-55702.23824,
124431.8423,
48144.00144,
-1887.819945,
24185,
-19339.32,
-100405.8,
-9365.489865,
4592,
148856.48,
-8088.63,
170989.8237,
151522.1154,
-4110.75,
-15779.12,
-101160.7982,
154569.2037,
-378488.1663,
254804.5905]).T


PortfolioRebalancer(currwts, wtstangent, currcashval, finalprices, tickers)
```

Appendix B FactorModels_wFamaFrench.py

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
@author: Khasan Dymov and Alex Shoop
"""

import os
import datetime
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pandas_datareader as pdr
import statsmodels.api as sm
```

```
import fix_yahoo_finance as yf #yahoo finance data fix
##############################################################################
yf.pdr_override() # yahoo finance data fix

#Specified tickers
tickers = ["AMZN",'AZO','BA','CI','COST','CRBP','DATA','F','GOOG','HUBS',\
           'JPM','MFC','MMM','MTN','NFLX','NKE','SBUX','TLT','VYM','WM']

#Market portfolio is S&P 500
sp = "^GSPC"

#Date range
startDate = datetime.datetime(2010,1,1)
endDate = datetime.datetime(2017,1,1)

#secDF = pd.DataFrame()
#i = 0
#for tick in tickers:
#    secDF[tick] = pdr.get_data_yahoo(tick, startDate, endDate)['Adj Close']
#    print(i)
#    i += 1

#Fetch Yahoo Finance data
# NB: If the data reader below breaks, just run again.
#     Yahoo data reader can be wacky sometimes.
#Using only adj close data
print("Attempting to import monthly YF data.....")
secData = pdr.get_data_yahoo(tickers, startDate, endDate, interval = 'm')['Adj
    ↪ Close']
SP500Data = pdr.get_data_yahoo(sp, startDate, endDate, interval = 'm')['Adj
    ↪ Close']
print(".\n")
print(".\n")
print(".\n")
print("Successfully imported all YF data.")

#Simple monthly returns, for securities and market portfolio
returnsSecuDF = pd.DataFrame(index = secData.index)
returnsSecuDF = secData/secData.shift(1) - 1
returnsMarketDF = pd.DataFrame(index = SP500Data.index)
returnsMarketDF = SP500Data/SP500Data.shift(1) - 1

#Expected returns of jth security, E[R_j]
expReturnsSecu = returnsSecuDF.mean()
#Expected returns of market portfolio, E[R_M]
expReturnsMarket = returnsMarketDF.mean()
#Annualize the parameters
muSecu = expReturnsSecu*12
muMarket = expReturnsMarket*12

#risk-free rate (same as in Markowitz Proj)
```

```
muF = 0.02

#set up the final output dataframe
finalAssetsDF = pd.DataFrame(columns = secData.columns, \
                             index = ['Beta','Expected Asset Returns','Asset
    ↪ Returns Variance'])
#Beta calculation
for stock in finalAssetsDF:
    tempDF = pd.concat([returnsSecuDF[stock], returnsMarketDF], axis = 1)
    tempDF = tempDF.dropna()    # dropping any NaN row entries

    # Returns for stock security, and market portfolio
    R_sec = tempDF.iloc[:,0]
    R_market = tempDF.iloc[:,1]

    # Excess returns, for regression model usage
    R_excessSec = R_sec - muF  # excess returns for jth security
    R_excessMarket = R_market - muF  # excess returns for market portfolio

    #Beta estimate, via covariance formula between R_j and R_M
    covariR = np.cov(R_sec, R_market)
    beta_formula = covariR[0,1]/covariR[1,1]
    #alpha = expReturnsSecu.AMZN - beta_formula*(expReturnsMarket)
    #print("Beta estimate via covariance formula is: ")
    #print(beta_formula)

    #CAPM beta estimate, via OLS regression model
    # The formula we are estimating is:
    #    R*_j = beta (R*_M) + eps

    # Setup for OLS fit
    Y = R_excessSec
    X = R_excessMarket.rename('ExcessMarketReturn')
    X = sm.add_constant(X)
    modelFit = sm.OLS(Y, X).fit()
    #print("OLS results are: ")
    #print(modelFit.summary())

    #Output results
    betaCAPM = modelFit.params['ExcessMarketReturn']
    expAssetReturn = muF + betaCAPM*(muMarket - muF)
    resids = modelFit.resid
    varAssetReturn = betaCAPM**2 * returnsMarketDF.var() + resids.var()

    #print statement
    print("Beta from covariance formula is: ")
    print(beta_formula)
    print("Beta from OLS regression model fit is: ")
    print(betaCAPM)

    finalAssetsDF[stock] = [modelFit.params['ExcessMarketReturn'], \
```

```
                                                    expAssetReturn,\

                                                    varAssetReturn]
#Print results
finalAssetsDF


# below is for Fama-French
#os.chdir(os.path.dirname(os.path.dirname(__file__)))
#os.chdir('Data')
file = 'F-F_SML_HML.csv'
dataCSV = pd.read_csv(file)
FFdata = dataCSV[['SMB','HML']]
FFdata.index = secData.index    # setting correct indices
FFdata = FFdata[1:]     # removing 1st row for correct dimensions

#set up the final output dataframe
finalAssetsDFTWO = pd.DataFrame(columns = secData.columns, \
                                index = ['Beta Mkt','Beta SMB',\
                                         'Beta HML','Expected Asset Returns',\
                                         'Asset Returns Variance'])

#Beta calculation
for stock in finalAssetsDFTWO:
    tempDF = pd.concat([returnsSecuDF[stock], returnsMarketDF, FFdata], axis =
    ↪  1)
    tempDF = tempDF.dropna()    # dropping any NaN row entries

    # Returns for stock security, and market portfolio, and Fama French
    ↪ factors
    R_sec = tempDF.iloc[:,0]
    R_market = tempDF.iloc[:,1]
    FFnewdata = pd.concat([tempDF.iloc[:,2], tempDF.iloc[:,3]], axis = 1)

    # Excess returns, for regression model usage
    R_excessSec = R_sec - muF  # excess returns for jth security
    R_excessMarket = R_market - muF  # excess returns for market portfolio

    #Beta estimate, via OLS regression model
    # The formula we are estimating is:
    #    R*_j = beta1(R*_M) + beta2(SMB) + beta3(HML) + eps

    # Setup for OLS fit
    Y = R_excessSec
    X = pd.concat([R_excessMarket,FFnewdata],axis=1)
    X = X.rename(columns = {'Adj Close': 'ExcessMarketReturn'})
    factorsCOV = X.cov()
    X = sm.add_constant(X)
    modelFit = sm.OLS(Y, X).fit()
    #print("OLS results are: ")
    #print(modelFit.summary())
```

```
163
164     #Output results
165     betas = modelFit.params['ExcessMarketReturn'],\
166             modelFit.params['SMB'],\
167             modelFit.params['HML']
168     expAssetReturn = muF + betas[0]*(muMarket - muF) + \
169                          betas[1]*FFdata['SMB'].mean() + \
170                          betas[2]*FFdata['HML'].mean()
171     resids = modelFit.resid
172     betasVec = np.array([betas])
173     # diff formula to capm version
174     varAssetReturn = np.dot(np.dot(betasVec,factorsCOV), betasVec.T) + resids.
    ↪ var()
175
176     finalAssetsDFTWO[stock] = [betas[0],betas[1],betas[2],expAssetReturn,
    ↪ varAssetReturn.item()]
```

Appendix C FactorModels_F1F2.py

```
1
2  #!/usr/bin/env python3
3  # -*- coding: utf-8 -*-
4  """
5  @author: Khasan Dymov and Alex Shoop
6  """
7
8  import os
9  import datetime
10 import numpy as np
11 import pandas as pd
12 import matplotlib.pyplot as plt
13 import pandas_datareader as pdr
14 import statsmodels.api as sm
15 import fix_yahoo_finance as yf #yahoo finance data fix
16 ##############################################################################
17 yf.pdr_override() # yahoo finance data fix
18
19 #Specified tickers
20 tickers = ["AMZN",'AZO','BA','CI','COST','CRBP','DATA','F','GOOG','HUBS',\
21            'JPM','MFC','MMM','MTN','NFLX','NKE','SBUX','TLT','VYM','WM']
22
23 #Market portfolio is S&P 500
24 sp = "^GSPC"
25
26 #Date range
27 startDate = datetime.datetime(2016,3,1)
28 endDate = datetime.datetime(2017,1,1)
29
30 #Fetch Yahoo Finance data
```

```
# NB: If the data reader below breaks, just run again.
#     Yahoo data reader can be wacky sometimes.
#Using only adj close data
print("Attempting to import monthly YF data.....")
secData = pdr.get_data_yahoo(tickers, startDate, endDate, interval = 'm')['Adj Close']
secData = secData[::-1]
SP500Data = pdr.get_data_yahoo(sp, startDate, endDate, interval = 'm')['Adj Close']
print(".\n")
print(".\n")
print(".\n")
print("Successfully imported all YF data.")

#Importing F1 (Global ex US Momentum factor) and F2 (Bloomberg RSTAMOM index)
file1 = 'Global_ex_US_MOM_Factor_10mo.csv'
file2 = 'bloomberg_RSTAMOM_10mo.csv'
dataCSV = pd.read_csv(file1)
F1data = dataCSV[['MOM']]
F1data.index = secData.index    # setting correct indices
dataCSV = pd.read_csv(file2)
F2data = dataCSV[['RSTAMOM']]
F2data.index = secData.index    # setting correct indices


#Simple monthly returns, for securities and market portfolio
returnsSecuDF = pd.DataFrame(index = secData.index)
returnsSecuDF = secData/secData.shift(1) - 1
returnsMarketDF = pd.DataFrame(index = SP500Data.index)
returnsMarketDF = SP500Data/SP500Data.shift(1) - 1

#Expected returns of jth security, E[R_j]
expReturnsSecu = returnsSecuDF.mean()
#Expected returns of market portfolio, E[R_M]
expReturnsMarket = returnsMarketDF.mean()
#Annualize the parameters
#muSecu = expReturnsSecu*12
#muMarket = expReturnsMarket*12m
muMarket = (1+expReturnsMarket)**12 - 1

#risk-free rate (same as in Markowitz Proj)
muF = 0.02

#set up the final output dataframe
finalAssetsF1 = pd.DataFrame(columns = secData.columns, \
                             index = ['Beta CAPM','Beta F1','Expected Asset Returns','Asset Returns Variance'])
finalAssetsF2 = pd.DataFrame(columns = secData.columns, \
                             index = ['Beta CAPM','Beta F2','Expected Asset Returns','Asset Returns Variance'])
finalAssetsF1F2 = pd.DataFrame(columns = secData.columns, \
```

```
                                index = ['Beta CAPM','Beta F1','Beta F2','
    ↪ Expected Asset Returns','Asset Returns Variance'])

# dropping NaN rows
tempDF = pd.concat([returnsSecuDF, returnsMarketDF, F1data, F2data], axis = 1)
tempDF = tempDF.dropna()
# Returns for ALL securities, and market portfolio
# and F1 F2 factors
R_sec = tempDF.iloc[:,:20]
R_market = tempDF.iloc[:,20]
newF1 = tempDF.iloc[:,21]
newF2 = tempDF.iloc[:,22]
newF1F2 = pd.concat([tempDF.iloc[:,21], tempDF.iloc[:,22]], axis = 1)
# Excess returns, for regression model usage
R_excessSec = R_sec - muF  # excess returns for jth security
R_excessMarket = R_market - muF  # excess returns for market portfolio

# X variable setup, for OLS regression purposes
# Also, getting CovF matrix for each factor model
XF1 = pd.concat([R_excessMarket,newF1],axis=1)
XF1 = XF1.rename(columns = {'Adj Close': 'ExcessMarketReturn'})
factorsF1COV = XF1.cov()
XF1 = sm.add_constant(XF1)

XF2 = pd.concat([R_excessMarket,newF2],axis=1)
XF2 = XF2.rename(columns = {'Adj Close': 'ExcessMarketReturn'})
factorsF2COV = XF2.cov()
XF2 = sm.add_constant(XF2)

XF1F2 = pd.concat([R_excessMarket,newF1F2],axis=1)
XF1F2 = XF1F2.rename(columns = {'Adj Close': 'ExcessMarketReturn'})
factorsF1F2COV = XF1F2.cov()
XF1F2 = sm.add_constant(XF1F2)

# big beta matrix, for covariance calculation purposes
bigBetaF1 = np.zeros(0)
bigBetaF2 = np.zeros(0)
bigBetaF1F2 = np.zeros(0)

#Beta calculation FOR CAPM
for stock in finalAssetsF1:
    #CAPM beta estimate, via OLS regression model
    # The formula we are estimating is:
    #    R*_j = beta (R*_M) + eps

    # Setup for OLS fit
    Y = R_excessSec[stock]
    modelFitF1 = sm.OLS(Y, XF1).fit()
    modelFitF2 = sm.OLS(Y, XF2).fit()
    modelFitF1F2 = sm.OLS(Y, XF1F2).fit()

```

```
    #Output results
    betasF1CAPM = modelFitF1.params['ExcessMarketReturn'],modelFitF1.params['MOM']
    if stock == 'AMZN':
        bigBetaF1 = np.append(bigBetaF1, betasF1CAPM)
    else:
        bigBetaF1 = np.vstack([bigBetaF1, betasF1CAPM])

    # remember, need to annualize expected returns and variance returns
    #expAssetF1Return = muF + betasF1CAPM[0]*(muMarket - muF) + betasF1CAPM[1]*((1+F1data.mean())**12 - 1)
    expAssetF1Return = (1 + muF + betasF1CAPM[0]*(expReturnsMarket - muF) + betasF1CAPM[1]*(F1data.mean().item()))**12 - 1
    residsF1 = modelFitF1.resid
    betasF1Vec = np.array([betasF1CAPM])
    varAssetF1Return = (np.dot(np.dot(betasF1Vec,factorsF1COV), betasF1Vec.T) + residsF1.var())*12
    finalAssetsF1[stock] = [betasF1CAPM[0], betasF1CAPM[1], expAssetF1Return.item(),varAssetF1Return.item()]

    betasF2CAPM = modelFitF2.params['ExcessMarketReturn'],modelFitF2.params['RSTAMOM']
    if stock == 'AMZN':
        bigBetaF2 = np.append(bigBetaF2, betasF2CAPM)
    else:
        bigBetaF2 = np.vstack([bigBetaF2, betasF2CAPM])

    #expAssetF2Return = muF + betasF2CAPM[0]*(muMarket - muF) + betasF2CAPM[1]*((1+F2data.mean())**12 - 1)
    expAssetF2Return = (1 + muF + betasF2CAPM[0]*(expReturnsMarket - muF) + betasF2CAPM[1]*(F2data.mean().item()))**12 - 1
    residsF2 = modelFitF2.resid
    betasF2Vec = np.array([betasF2CAPM])
    varAssetF2Return = (np.dot(np.dot(betasF2Vec,factorsF2COV), betasF2Vec.T) + residsF2.var())*12
    finalAssetsF2[stock] = [betasF2CAPM[0], betasF2CAPM[1], expAssetF2Return.item(),varAssetF2Return.item()]

    betasF1F2CAPM = modelFitF1F2.params['ExcessMarketReturn'],modelFitF1F2.params['MOM'],modelFitF1F2.params['RSTAMOM']
    if stock == 'AMZN':
        bigBetaF1F2 = np.append(bigBetaF1F2, betasF1F2CAPM)
    else:
        bigBetaF1F2 = np.vstack([bigBetaF1F2, betasF1F2CAPM])

    #expAssetF1F2Return = muF + betasF1F2CAPM[0]*(muMarket - muF) + betasF1F2CAPM[1]*((1+F1data.mean().item())**12 - 1) + betasF1F2CAPM[2]*((1+F2data.mean().item())**12 - 1)
    expAssetF1F2Return = (1 + muF + betasF1F2CAPM[0]*(expReturnsMarket - muF) + betasF1F2CAPM[1]*(F1data.mean().item()) + betasF1F2CAPM[2]*(F2data.mean().item()))**12 - 1
```

```
164     residsF1F2 = modelFitF1F2.resid
165     betasF1F2Vec = np.array([betasF1F2CAPM])
166     varAssetF1F2Return = (np.dot(np.dot(betasF1F2Vec,factorsF1F2COV),
    ↪ betasF1F2Vec.T) + residsF1F2.var())*12
167     finalAssetsF1F2[stock] = [betasF1F2CAPM[0], betasF1F2CAPM[1],
    ↪ betasF1F2CAPM[2], expAssetF1F2Return,varAssetF1F2Return.item()]
168
169 # calculating covariance between asset returns
170 covAssetReturnsCAPM_F1 = np.empty((20,20))
171 covAssetReturnsCAPM_F1[:] = np.nan
172 covAssetReturnsCAPM_F2 = np.empty((20,20))
173 covAssetReturnsCAPM_F2[:] = np.nan
174 covAssetReturnsCAPM_F1F2 = np.empty((20,20))
175 covAssetReturnsCAPM_F1F2[:] = np.nan
176 for i in range(20):
177     for j in range(20):
178         covAssetReturnsCAPM_F1[i][j] = np.dot(np.dot(bigBetaF1[i],
    ↪ factorsF1COV), bigBetaF1[j])
179         covAssetReturnsCAPM_F2[i][j] = np.dot(np.dot(bigBetaF2[i],
    ↪ factorsF2COV), bigBetaF2[j])
180         covAssetReturnsCAPM_F1F2[i][j] = np.dot(np.dot(bigBetaF1F2[i],
    ↪ factorsF1F2COV), bigBetaF1F2[j])
181 covPD_CAPM_F1 = pd.DataFrame(covAssetReturnsCAPM_F1, index = secData.columns,
    ↪ columns=secData.columns)
182 covPD_CAPM_F2 = pd.DataFrame(covAssetReturnsCAPM_F2, index = secData.columns,
    ↪ columns=secData.columns)
183 covPD_CAPM_F1F2 = pd.DataFrame(covAssetReturnsCAPM_F1F2, index = secData.
    ↪ columns, columns=secData.columns)
184
185
186
187 ##########################################################################
188
189
190 # below is for Fama-French
191 file = 'F-F_SML_HML_10mo.csv'
192 dataCSV = pd.read_csv(file)
193 FFdata = dataCSV[['SMB','HML']]
194 FFdata.index = secData.index    # setting correct indices
195 FFdata = FFdata[1:]     # removing 1st row for correct dimensions
196
197 #set up the final output dataframe
198 finalAssetsFF_F1 = pd.DataFrame(columns = secData.columns, index = ['Beta Mkt'
    ↪ ,'Beta SMB','Beta HML','Beta F1','Expected Asset Returns','Asset Returns
    ↪  Variance'])
199
200 finalAssetsFF_F2 = pd.DataFrame(columns = secData.columns, \
201                                 index = ['Beta Mkt','Beta SMB',\
202                                          'Beta HML','Beta F2','Expected Asset
    ↪ Returns',\
203                                          'Asset Returns Variance'])
```

```
finalAssetsFF_F1F2 = pd.DataFrame(columns = secData.columns, \
                                  index = ['Beta Mkt','Beta SMB',\
                                           'Beta HML','Beta F1','Beta F2','Expected
    ↪  Asset Returns',\
                                           'Asset Returns Variance'])
# dropping NaN rows
tempDF = pd.concat([returnsSecuDF, returnsMarketDF, FFdata, F1data, F2data],
    ↪ axis = 1)
tempDF = tempDF.dropna()
# Returns for ALL securities, and market portfolio, and Fama-French factors
# and F1 F2 factors
R_sec = tempDF.iloc[:,:20]
R_market = tempDF.iloc[:,20]
FFnewdata = pd.concat([tempDF.iloc[:,21], tempDF.iloc[:,22]], axis = 1)
newF1 = tempDF.iloc[:,23]
newF2 = tempDF.iloc[:,24]
newF1F2 = pd.concat([tempDF.iloc[:,23], tempDF.iloc[:,24]], axis = 1)
# Excess returns, for regression model usage
R_excessSec = R_sec - muF  # excess returns for jth security
R_excessMarket = R_market - muF  # excess returns for market portfolio

# X variable setup, for OLS regression purposes
# Also, getting CovF matrix for each factor model
XF1 = pd.concat([R_excessMarket,FFnewdata,newF1],axis=1)
XF1 = XF1.rename(columns = {'Adj Close': 'ExcessMarketReturn'})
factorsFF_F1COV = XF1.cov()
XF1 = sm.add_constant(XF1)

XF2 = pd.concat([R_excessMarket,FFnewdata,newF2],axis=1)
XF2 = XF2.rename(columns = {'Adj Close': 'ExcessMarketReturn'})
factorsFF_F2COV = XF2.cov()
XF2 = sm.add_constant(XF2)

XF1F2 = pd.concat([R_excessMarket,FFnewdata,newF1,newF2],axis=1)
XF1F2 = XF1F2.rename(columns = {'Adj Close': 'ExcessMarketReturn'})
factorsFF_F1F2COV = XF1F2.cov()
XF1F2 = sm.add_constant(XF1F2)

# big beta matrix, for covariance calculation purposes
bigBetaFF_F1 = np.zeros(0)
bigBetaFF_F2 = np.zeros(0)
bigBetaFF_F1F2 = np.zeros(0)

#Beta calculation FOR FRENCH-FAMA
for stock in finalAssetsFF_F1:
    #Beta estimate, via OLS regression model
    # The formula we are estimating is:
    #    R*_j = beta1(R*_M) + beta2(SMB) + beta3(HML) + eps

    # Setup for OLS fit
    Y = R_excessSec[stock]
```

```
253     modelFitF1 = sm.OLS(Y, XF1).fit()
254     modelFitF2 = sm.OLS(Y, XF2).fit()
255     modelFitF1F2 = sm.OLS(Y, XF1F2).fit()
256 
257     #Output results
258     betasFF_F1 = modelFitF1.params['ExcessMarketReturn'],modelFitF1.params['
    ↪ SMB'],\
259                  modelFitF1.params['HML'],modelFitF1.params['MOM']
260     if stock == 'AMZN':
261         bigBetaFF_F1 = np.append(bigBetaFF_F1, betasFF_F1)
262     else:
263         bigBetaFF_F1 = np.vstack([bigBetaFF_F1, betasFF_F1])
264 
265     #expAssetReturnFF_F1 = muF + betasFF_F1[0]*(muMarket - muF) + \
266     #             betasFF_F1[1]*((1+FFdata['SMB'].mean())**12 - 1) + betasFF_F1
    ↪ [2]*((1+FFdata['HML'].mean())**12 - 1) + \
267     #             betasFF_F1[3]*((1+F1data.mean().item())**12 - 1)
268     expAssetReturnFF_F1 = (1 + muF + betasFF_F1[0]*(expReturnsMarket - muF) +
    ↪ \
269                  betasFF_F1[1]*(FFdata['SMB'].mean()) + betasFF_F1[2]*(FFdata['
    ↪ HML'].mean()) + \
270                  betasFF_F1[3]*(F1data.mean().item()))**12 - 1
271     residsF1 = modelFitF1.resid
272     betasF1Vec = np.array([betasFF_F1])
273     varAssetReturnFF_F1 = (np.dot(np.dot(betasF1Vec,factorsFF_F1COV),
    ↪ betasF1Vec.T) + residsF1.var())*12
274     finalAssetsFF_F1[stock] = [betasFF_F1[0],betasFF_F1[1],betasFF_F1[2],
    ↪ betasFF_F1[3],expAssetReturnFF_F1,varAssetReturnFF_F1.item()]
275 
276 
277     betasFF_F2 = modelFitF2.params['ExcessMarketReturn'],modelFitF2.params['
    ↪ SMB'],\
278                  modelFitF2.params['HML'],modelFitF2.params['RSTAMOM']
279     if stock == 'AMZN':
280         bigBetaFF_F2 = np.append(bigBetaFF_F2, betasFF_F2)
281     else:
282         bigBetaFF_F2 = np.vstack([bigBetaFF_F2, betasFF_F2])
283 
284     #expAssetReturnFF_F2 = muF + betasFF_F2[0]*(muMarket - muF) + \
285     #             betasFF_F2[1]*((1+FFdata['SMB'].mean())**12 - 1) + betasFF_F2
    ↪ [2]*((1+FFdata['HML'].mean())**12 - 1) + \
286     #             betasFF_F2[3]*((1+F2data.mean().item())**12 - 1)
287     expAssetReturnFF_F2 = (1 + muF + betasFF_F2[0]*(expReturnsMarket - muF) +
    ↪ \
288                  betasFF_F2[1]*(FFdata['SMB'].mean()) + betasFF_F2[2]*(FFdata['
    ↪ HML'].mean()) + \
289                  betasFF_F2[3]*(F2data.mean().item()))**12 - 1
290     residsF2 = modelFitF2.resid
291     betasF2Vec = np.array([betasFF_F2])
292     varAssetReturnFF_F2 = (np.dot(np.dot(betasF2Vec,factorsFF_F2COV),
    ↪ betasF2Vec.T) + residsF2.var())*12
```

```
293     finalAssetsFF_F2[stock] = [betasFF_F2[0],betasFF_F2[1],betasFF_F2[2],
    ↪ betasFF_F2[3],expAssetReturnFF_F2,varAssetReturnFF_F2.item()]
294
295     betasFF_F1F2 = modelFitF1F2.params['ExcessMarketReturn'],modelFitF1F2.
    ↪ params['SMB'],\
296                 modelFitF1F2.params['HML'],modelFitF1F2.params['MOM'],
    ↪ modelFitF1F2.params['RSTAMOM']
297     if stock == 'AMZN':
298         bigBetaFF_F1F2 = np.append(bigBetaFF_F1F2, betasFF_F1F2)
299     else:
300         bigBetaFF_F1F2 = np.vstack([bigBetaFF_F1F2, betasFF_F1F2])
301
302     #expAssetReturnFF_F1F2 = muF + betasFF_F1F2[0]*(muMarket - muF) + \
303     #            betasFF_F1F2[1]*((1+FFdata['SMB'].mean())**12 - 1) +
    ↪ betasFF_F1F2[2]*((1+FFdata['HML'].mean())**12 - 1) + \
304     #            betasFF_F1F2[3]*((1+F1data.mean().item())**12 - 1) +
    ↪ betasFF_F1F2[4]*((1+F2data.mean().item())**12 - 1)
305     expAssetReturnFF_F1F2 = (1 + muF + betasFF_F1F2[0]*(expReturnsMarket - muF
    ↪ ) + \
306                 betasFF_F1F2[1]*(FFdata['SMB'].mean()) + betasFF_F1F2[2]*(
    ↪ FFdata['HML'].mean()) + \
307                 betasFF_F1F2[3]*(F1data.mean().item()) + betasFF_F1F2[4]*(
    ↪ F2data.mean().item()))**12 - 1
308     residsF1F2 = modelFitF1F2.resid
309     betasF1F2Vec = np.array([betasFF_F1F2])
310     varAssetReturnFF_F1F2 = (np.dot(np.dot(betasF1F2Vec,factorsFF_F1F2COV),
    ↪ betasF1F2Vec.T) + residsF1F2.var())*12
311     finalAssetsFF_F1F2[stock] = [betasFF_F1F2[0],betasFF_F1F2[1],betasFF_F1F2
    ↪ [2],betasFF_F1F2[3],betasFF_F1F2[4],expAssetReturnFF_F1F2,
    ↪ varAssetReturnFF_F1F2.item()]
312
313 # calculating covariance between asset returns
314 covAssetReturnsFF_F1 = np.empty((20,20))
315 covAssetReturnsFF_F1[:] = np.nan
316 covAssetReturnsFF_F2 = np.empty((20,20))
317 covAssetReturnsFF_F2[:] = np.nan
318 covAssetReturnsFF_F1F2 = np.empty((20,20))
319 covAssetReturnsFF_F1F2[:] = np.nan
320 for i in range(20):
321     for j in range(20):
322         covAssetReturnsFF_F1[i][j] = np.dot(np.dot(bigBetaFF_F1[i],
    ↪ factorsFF_F1COV), bigBetaFF_F1[j])
323         covAssetReturnsFF_F2[i][j] = np.dot(np.dot(bigBetaFF_F2[i],
    ↪ factorsFF_F2COV), bigBetaFF_F2[j])
324         covAssetReturnsFF_F1F2[i][j] = np.dot(np.dot(bigBetaFF_F1F2[i],
    ↪ factorsFF_F1F2COV), bigBetaFF_F1F2[j])
325 covPD_FF_F1 = pd.DataFrame(covAssetReturnsFF_F1, index = secData.columns,
    ↪ columns=secData.columns)
326 covPD_FF_F2 = pd.DataFrame(covAssetReturnsFF_F2, index = secData.columns,
    ↪ columns=secData.columns)
```

```
covPD_FF_F1F2 = pd.DataFrame(covAssetReturnsFF_F1F2, index = secData.columns,
    ↪ columns=secData.columns)

# norm length covariance comparisons, etc
np.linalg.norm(covPD_CAPM_F1 - covPD_CAPM_F2)
```

Appendix D EF_Portfolio_multiWeek_plotting.py

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
@author: Khasan Dymov and Alex Shoop
"""

import numpy as np
import matplotlib.pyplot as plt
from pandas_datareader import data
import datetime
from time import sleep

##############################################################################
#setting output np options
np.set_printoptions(threshold = np.inf)

# OPTIMAL PORTFOLIO FUNCTION
def optimalPortfolio(mu, omega, muP):
    # We follow by example
    # the process as explained in Statistics & Finance by David Ruppert
    # Necessary coefficients for risk-efficient portfolio
    n1 = np.matrix(np.ones_like(mu)).T
    n1t = n1.T
    invomega = np.linalg.inv(omega)
    muR = np.matrix(mu).T
    mut = muR.T

    A = np.matrix.item(n1t*invomega*muR)
    B = np.matrix.item(mut*invomega*muR)
    C = np.matrix.item(n1t*invomega*n1)
    D = B*C-A**2


    # Fixed vectors for efficient weights calculation
    g = (B*invomega*n1 - A*invomega*muR)/D
    h = (C*invomega*muR - A*invomega*n1)/D

    # The final formula:
    wts = g + muP*h

```

```
42      # Our efficient weights
43      return wts
44
45  #Import specified date range of prices data,
46  #and calculated returns (log returns)
47
48  #Initial beginning date range
49  startDate = datetime.datetime(2015,10,27)
50  endDate = datetime.datetime(2017,10,27)
51
52  #Specified tickers
53  tickers = ['AMZN','AZO','BA','CI','COST','CRBP','DATA','F','GOOG','HUBS',\
54             'JPM','MFC','MMM','MTN','NFLX','NKE','SBUX','TLT','VYM','WM']
55
56  #Money weights (risk-free + securities) of our portfolio for plotting purposes
57  ourPortfolio = [500000/1000000, 487595/998050, 439878/978143, 437527/974728, \
58                  441136/978516]
59
60  # BIG WHILE LOOP FOR MULTIPLE WEEKS, IN ORDER TO DO BIG PLOT
61  count = 0
62  # change count to 6, for finale where historical data range should be:
63  # startdate = 2015, 12, 1
64  # enddate = 2017, 12, 1
65  while count < 5:
66      #Rolling forward the date range if we should be Oct 27 or later
67      if count != 0:
68          # add 1 more week
69          startDate += datetime.timedelta(7)
70          endDate += datetime.timedelta(7)
71
72      #Use pandas_reader.data.DataReader to load desired data
73
74      ## this could be more efficiently implemented
75
76      # if the data reader below breaks, just run again.
77      # Yahoo data reader can be wacky sometimes.
78
79      securityRawData = data.DataReader(tickers, 'yahoo', startDate, endDate)
80
81
82      #Getting adjusted close prices
83      close = securityRawData['Adj Close'].astype('float')
84      #fixing date ordering
85      close = close.iloc[::-1]
86      #getting log returns
87      dailyLogReturns = np.log(close).diff()
88      #removing the 1st row, of empty values for first date
89      dailyLogReturns = dailyLogReturns.drop(startDate)
90      #Pulling just the last day closing prices for rebalancing function
91      finalprices = np.matrix([close.iloc[::-1].iloc[0]]).T
92
```

```
    #Gather expected returns, and covariance matrix
    mu2yrsdaily = dailyLogReturns.mean()
    var2yrsdaily = dailyLogReturns.var()
    sd2yrsdaily = np.sqrt(var2yrsdaily)
    cov2yrsdaily = dailyLogReturns.cov()

    #Annualize parameter estimates

    #2 year lookback:
    muannual = np.sqrt(((1+mu2yrsdaily)**505))-1
    varannual = np.sqrt(252)*var2yrsdaily
    sdannual = 252*sd2yrsdaily
    covannual = 252*cov2yrsdaily


    #Step 1: Find the minimum variance portfolio for our problem

    n1 = np.matrix(np.ones_like(muannual)).T
    n1t = n1.T
    invomega = np.linalg.inv(covannual)
    muR = np.matrix(muannual).T
    mut = muR.T

    A = np.matrix.item(n1t*invomega*muR)
    B = np.matrix.item(mut*invomega*muR)
    C = np.matrix.item(n1t*invomega*n1)
    D = B*C-A**2

    #Need to use specific g,h for minimum variance portfolio

    gmin = (B*invomega*n1 - A*invomega*muR)/D
    hmin = (C*invomega*muR - A*invomega*n1)/D
    g1 = gmin.T*np.matrix(covannual)*gmin
    h1 = hmin.T*np.matrix(covannual)*hmin
    g1h1 = gmin.T*np.matrix(covannual)*hmin

    mumin = np.matrix.item(- g1h1/h1)
    sigmamin = np.matrix.item(np.sqrt(g1 - g1h1**2/h1))

    #Weights associated with this minimum variance portfolio
    wtsmin = np.array(gmin + mumin*hmin)
    barX = np.arange(np.size(wtsmin))

    #Step 2: Calculate the efficient frontier

    muP = np.linspace(min(muannual), max(muannual), 100)
    sigmaP = np.zeros(0)

    for target in muP:
        wtsP = optimalPortfolio(muannual, covannual, target)
        sigmaP = np.append(sigmaP, np.sqrt(wtsP.T*np.matrix(covannual)*wtsP))
```

```
144
145     effindices = np.where(muP>mumin)
146     ineffindices = np.where(muP<mumin)
147
148     #Step 2.5: Calculate the tangency portfolio
149     muF = 0.02
150     omegabar = invomega*(muR - muF*n1)
151     omegaT = omegabar/(n1t*omegabar)
152     sigmaT = np.matrix.item(np.sqrt(omegaT.T*np.matrix(covannual)*omegaT))
153     muT = np.matrix.item(mut*omegaT)
154
155     #Step 3: Plot minimum variance portfolio and upper section of efficient
    ↪ frontier
156
157     plt.plot(sigmaP[effindices], muP[effindices], 'g', alpha = 2/5 + (1/10)*
    ↪ count)
158     plt.plot(sigmaP[ineffindices], muP[ineffindices], 'c--', alpha = 2/5 +
    ↪ (1/10)*count)
159     #plt.plot(sigmaT, muT, 'rx', markersize = 8, alpha = 3/4 - (1/10)*count)
160     if count == 4:
161         plt.plot(sigmaT*ourPortfolio[-1], muT*ourPortfolio[-1], 'ro',
    ↪ markersize = 8, label = "Our Portfolio")
162     plt.plot(sigmaT*ourPortfolio[count], muT*ourPortfolio[count], 'ro',
    ↪ markersize = 8, alpha = 2/5 + (1/10)*count)
163     plt.plot([0,sigmaT],[muF,muT], 'b--', alpha = 2/5 + (1/10)*count)
164     # Just one plot function for the risk-free rate
165     if count == 0:
166         plt.plot(0,muF,'bo',label = "Risk-free rate")
167     # increment loop
168     count += 1
169
170 plt.title("Efficient Frontier for 20 risky assets, across multiple 2-yr ranges
    ↪ ")
171 plt.xlabel(r"$risk = \sigma_r$")
172 plt.ylabel(r"$reward = \mu_r$")
173 plt.legend()
174 plt.show()
175
176 #Step 4: Bar chart of weights
177
178 #plt.bar(barX, (wtsmin)*100)
179 #plt.axhline(y = 0, linewidth = 0.5, color = 'black')
180 #plt.xticks(barX, stocks, rotation=60)
181 #plt.ylabel("Percentage")
182 #plt.title("Weights for min var portfolio (2yrs)")
183
184 #Step 5: Bar chart of weights in tangency portfolio
185 wtstangent = optimalPortfolio(muannual, covannual, muT)
186
187 #plt.bar(barX, (wtstangent)*100)
188 #plt.axhline(y = 0, linewidth = 0.5, color = 'black')
```

```
#plt.xticks(barX, stocks, rotation=60)
#plt.ylabel("Percentage")
#plt.title("Weights for Initial Tangency Portfolio")

print("Expected return = " + str(muT))
print ("Portfolio std. dev = " + str(sigmaT))
print("The weights of the assets in the tangency portfolio are " + str(
    ↪ wtstangent))
```

Appendix E CAPM_altBeta.py

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
@author: Khasan Dymov and Alex Shoop
"""

import os
import datetime
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pandas_datareader as pdr
import statsmodels.api as sm
import fix_yahoo_finance as yf #yahoo finance data fix
################################################################################
yf.pdr_override() # yahoo finance data fix

#Specified tickers
tickers = ["AMZN",'AZO','BA','CI','COST','CRBP','DATA','F','GOOG','HUBS',\
           'JPM','MFC','MMM','MTN','NFLX','NKE','SBUX','TLT','VYM','WM']

#Market portfolio is S&P 500
sp = "^GSPC"

#Date range, 5 years
startDate = datetime.datetime(2012,1,1)
endDate = datetime.datetime(2017,1,1)

#Fetch Yahoo Finance data
# NB: If the data reader below breaks, just run again.
#     Yahoo data reader can be wacky sometimes.
#Using only adj close data
print("Attempting to import monthly YF data.....")
secData = pdr.get_data_yahoo(tickers, startDate, endDate, interval = 'm')['Adj
    ↪  Close']
#below is only necessary if date range is smaller, like 2 years.
#secData = secData[::-1]     # fixing ordering, so that it's oldest to newest
```

```
SP500Data = pdr.get_data_yahoo(sp, startDate, endDate, interval = 'm')['Adj Close']
print(".\n")
print(".\n")
print(".\n")
print("Successfully imported all YF data.")


#Simple monthly returns, for securities and market portfolio
returnsSecuDF = pd.DataFrame(index = secData.index)
returnsSecuDF = secData/secData.shift(1) - 1
returnsMarketDF = pd.DataFrame(index = SP500Data.index)
returnsMarketDF = SP500Data/SP500Data.shift(1) - 1

#Expected returns of jth security, E[R_j]
expReturnsSecu = returnsSecuDF.mean()
#Expected returns of market portfolio, E[R_M]
expReturnsMarket = returnsMarketDF.mean()
#Annualize the parameters
muSecu = expReturnsSecu*12
muMarket = expReturnsMarket*12

#risk-free rate (same as in Markowitz Proj)
muF = 0.02

#set up the final output dataframe
finalAssets_a = pd.DataFrame(columns = secData.columns, \
                             index = ['Beta CAPM'])

#a, Historical returns data Beta calculation
for stock in finalAssets_a:
    tempDF = pd.concat([returnsSecuDF[stock], returnsMarketDF], axis = 1)
    tempDF = tempDF.dropna()    # dropping any NaN row entries, ie not all historical data available for this security

    # Returns for stock security, and market portfolio
    R_sec = tempDF.iloc[:,0]
    R_market = tempDF.iloc[:,1]

    # Excess returns, for regression model usage
    R_excessSec = R_sec - muF  # excess returns for jth security
    R_excessMarket = R_market - muF  # excess returns for market portfolio

    #CAPM beta estimate, via OLS regression model
    # The formula we are estimating is:
    #    R*_j = beta (R*_M) + eps

    # Setup for OLS fit
    Y = R_excessSec
    Xhist = R_excessMarket
    Xhist = Xhist.rename('ExcessMarketReturn')
```

```
    Xhist = sm.add_constant(Xhist)
    modelFit_a = sm.OLS(Y, Xhist).fit()

    #Output results
    betas_a = modelFit_a.params['ExcessMarketReturn']
    finalAssets_a[stock] = [betas_a]

# print results
orderedBetas_a = finalAssets_a.transpose().sort_values(by='Beta CAPM')

portfolioReturns = [-0.00490, -0.00200, -0.00520, 0.00660, 0.00200, -0.00090,\
                    -0.00190, -0.00650, -0.01000, -0.00070, 0.00110, 0.00060,\
                    -0.00030, -0.00100, -0.00420, 0.00120, 0.00290, -0.00180,\
                    0.00240, 0.00090, 0.00160, -0.00230, 0.00680, -0.00260, 0.00530]
SP500portfolioRelated = [-0.003192, 0.000944, 0.001592, 0.000190, 0.003097,
    ↪ 0.001271,\
                         -0.000189, 0.001444, -0.003762, -0.000898, 0.000984,\
                         -0.002310, -0.005526, 0.008196, -0.002626, 0.001276,\
                         0.006541, -0.000750, 0.002056, -0.000384, 0.009848,\
                         -0.000369, 0.008191, -0.002024, -0.001052]

plt.hist(portfolioReturns, 25, edgecolor = 'black')
plt.xlabel("Returns")
plt.ylabel("Frequency")
plt.title("Distribution of Our Portfolio Returns")

tempDF = pd.concat([pd.DataFrame(portfolioReturns), pd.DataFrame(
    ↪ SP500portfolioRelated)], axis = 1)
# Returns for our portfolio, and market portfolio
R_port = tempDF.iloc[:,0].rename("Portfolio")
R_market = tempDF.iloc[:,1].rename("Market")
# Excess returns, for regression model usage
newR_excessPort = R_port - muF
R_excessMarket = R_market - muF
modelFitPortfolio = sm.OLS(newR_excessPort, R_excessMarket).fit()
portfolioBETA = modelFitPortfolio.params['Market']

########################################
# b, least trimmed squares

#set up the final output dataframe
finalAssets_b = pd.DataFrame(columns = secData.columns, \
                             index = ['Beta LTS'])
#a, Historical returns data Beta calculation
for stock in finalAssets_b:
    tempDF = pd.concat([returnsSecuDF[stock], returnsMarketDF], axis = 1)
    tempDF = tempDF.dropna()    # dropping any NaN row entries, ie not all
    ↪ historical data available for this security

    # Returns for stock security, and market portfolio
    R_sec = tempDF.iloc[:,0]
```

```
    R_market = tempDF.iloc[:,1]
    q_alpha = R_sec.quantile(0.05)
    q_alphaOpp = R_sec.quantile(1 - 0.05)
    # setting the alpha-quantiles to the actual values. Thus, trimming it.
    R_sec[R_sec < q_alpha] = q_alpha
    R_sec[R_sec > q_alphaOpp] = q_alphaOpp

    # Excess returns, for regression model usage
    R_excessSec = R_sec - muF  # excess returns for jth security
    R_excessMarket = R_market - muF  # excess returns for market portfolio

    # Setup for OLS fit
    Y = R_excessSec
    Xtrim = R_excessMarket
    Xtrim = Xtrim.rename('ExcessMarketReturn')
    Xtrim = sm.add_constant(Xtrim)
    modelFit_b = sm.OLS(Y, Xtrim).fit()

    #Output results
    betas_b = modelFit_b.params['ExcessMarketReturn']
    finalAssets_b[stock] = [betas_b]

# choose VYM as representative asset
newVYM = returnsSecuDF['VYM']
newVYM = newVYM.dropna()
q_alpha = newVYM.quantile(0.05)
q_alphaOpp = newVYM.quantile(1 - 0.05)
# setting the alpha-quantiles to the actual values. Thus, trimming it.
newVYM[newVYM < q_alpha] = q_alpha
newVYM[newVYM > q_alphaOpp] = q_alphaOpp

# new regression
R_newMarket = returnsMarketDF.iloc[1:]
# Excess returns, for regression model usage
R_excessSec = newVYM - muF  # excess returns
R_excessMarket = R_newMarket - muF  # excess returns for market portfolio
# Setup for OLS fit
Y = R_excessSec
Xtrim = R_excessMarket
Xtrim = Xtrim.rename('ExcessMarketReturn')
Xtrim = sm.add_constant(Xtrim)
modelFit_b = sm.OLS(Y, Xtrim).fit()

betas_b_VYM = modelFit_b.params['ExcessMarketReturn']

# histogram plotting
vymReturns = returnsSecuDF['VYM'].dropna()
binBounds1 = np.linspace(min(newVYM), max(newVYM), 59)
binBounds2 = np.linspace(min(vymReturns), max(vymReturns), 59)

```

```
plt.hist(newVYM, bins = binBounds2, label = 'Trimmed', color = 'red') # trimmed distribution
plt.hist(vymReturns, bins = binBounds2, label = 'Original', edgecolor = 'black') # original distribution
plt.title("Returns distribution for VYM")
plt.xlabel("Returns")
plt.ylabel("Frequency")
plt.legend()
plt.show()

########################################
# c, constant beta shrinkage, alpha_i = 2/3
betaBar = orderedBetas_a.mean().item()
alphaShrink = 2/3
# James-Stein estimator
betasJS = betaBar + alphaShrink*(orderedBetas_a - betaBar)

########################################
# d, exponentially weighted moving average
lamb1 = 0.94    # the standard for 'monthly data', according to Professor Blais
lamb2 = 0.8
# need to make sure returns are ordered from newest to oldest
newToOldSecReturns = returnsSecuDF[::-1]
newMarketReturns = returnsMarketDF[::-1]

# covariance estimate
def covExpWeighted(assetReturns, marketReturns, t, lam):
    if t == 0:
        return 0
    else:
        return (1 - lam)*assetReturns[t-1]*marketReturns[t-1] + lam*(covExpWeighted(assetReturns, marketReturns, t-1,lam))
# variance estimate
def varExpWeighted(marketReturns, t, lam):
    if t == 0:
        return 0
    else:
        return (1 - lam)*marketReturns[t]**2 + lam*varExpWeighted(marketReturns, t-1, lam)

#set up the final output dataframe
finalAssets_d = pd.DataFrame(columns = newToOldSecReturns.columns, \
                             index = ['EWMA Beta (Lambda=0.94)', 'EWMA Beta (Lambda=0.8)'])

# exponentially weighted avg beta calculation for each stock
for stock in finalAssets_d:
    tempDF = pd.concat([newToOldSecReturns[stock], newMarketReturns], axis = 1)
```

```
    tempDF = tempDF.dropna()     # dropping any NaN row entries, ie not all
    ↪ historical data available for this security

    # Returns for stock security, and market portfolio
    R_sec = tempDF.iloc[:,0]
    R_market = tempDF.iloc[:,1]

    # function values
    cov_lamb1 = covExpWeighted(R_sec, R_market, len(R_sec), lamb1)
    cov_lamb2 = covExpWeighted(R_sec, R_market, len(R_sec), lamb2)
    var_lamb1 = varExpWeighted(R_market, len(R_sec)-1, lamb1)
    var_lamb2 = varExpWeighted(R_market, len(R_sec)-1, lamb2)

    beta_lamb1 = cov_lamb1 / var_lamb1
    beta_lamb2 = cov_lamb2 / var_lamb2

    finalAssets_d[stock] = [beta_lamb1, beta_lamb2]

# print results
finalAssets_d
orderedBetas_d = finalAssets_d.transpose().sort_values(by='EWMA Beta (Lambda
    ↪ =0.94)')
```

Appendix F Full Trade History

Activity Statement

October 20, 2017 - December 5, 2017

Interactive Brokers LLC, Two Pickwick Plaza, Greenwich, CT 06830

Symbol	Date/Time	Exchange	Quantity	T. Price	C. Price	Proceeds	Comm/Fee	Basis	Realized P/L	MTM P/L	Code
Stocks											
USD											
AMZN	2017-10-30, 10:54:11	ISLAND	51	1,116.8900	1,110.8500	-56,961.39	-1.00	56,962.39	0.00	-308.04	O
AMZN	2017-11-06, 09:30:09	ISLAND	3	1,109.5000	1,120.6600	-3,328.50	-1.00	3,329.50	0.00	33.48	O
AMZN	2017-11-13, 09:31:26	ISLAND	-9	1,125.0000	1,129.1700	10,125.00	-1.23	-10,052.19	71.58	-37.53	C
AMZN	2017-11-27, 10:33:48	AMEX	6	1,208.6000	1,195.8300	-7,251.60	-1.00	7,252.60	0.00	-76.62	O
AMZN	2017-12-04, 09:30:10	ARCA	-51	1,173.9600	1,133.9500	59,871.96	-2.39	-57,492.30	2,377.27	2,040.51	C
Total AMZN			**0**			**2,455.47**	**-6.62**	**0.00**	**2,448.85**	**1,651.80**	
AZO	2017-10-30, 09:30:37	-	-119	582.7231933	590.1500	69,344.06	-2.62	-69,341.44	0.00	-883.79	O;P
AZO	2017-11-06, 09:30:27	BYX	-18	604.0100	607.0000	10,872.18	-1.25	-10,870.93	0.00	-53.82	O
AZO	2017-11-13, 09:32:06	NYSE	17	598.5000	595.2600	-10,174.50	-1.00	9,905.84	-269.66	-55.08	C
AZO	2017-11-20, 09:30:43	NYSE	32	629.9900	634.6500	-20,159.68	-1.00	18,646.29	-1,514.39	149.12	C
AZO	2017-11-27, 09:30:47	BYX	3	635.5100	637.9500	-1,906.53	-1.00	1,748.09	-159.44	7.32	C
AZO	2017-12-04, 09:30:38	BATS	85	690.0100	709.7700	-58,650.85	-1.00	49,912.16	-8,739.69	1,679.60	C
Total AZO			**0**			**-10,675.32**	**-7.87**	**0.00**	**-10,683.19**	**843.35**	
BA	2017-10-30, 09:30:31	-	717	255.7332636	259.2500	-183,360.75	-3.58	183,364.34	0.00	2,521.50	O;P
BA	2017-11-06, 09:30:32	-	-155	262.1800	264.0700	40,637.90	-1.96	-39,611.25	1,024.69	-292.95	C;P
BA	2017-11-13, 09:30:23	ISLAND	-11	261.1200	262.4200	2,872.32	-1.07	-2,811.05	60.20	-14.30	C
BA	2017-11-20, 09:30:29	BYX	-83	262.7500	264.6300	21,808.25	-1.51	-21,215.30	591.43	-156.04	C
BA	2017-11-27, 09:30:48	DRCTEDGE	32	265.7900	265.5800	-8,505.28	-1.00	8,506.28	0.00	-6.72	O
BA	2017-12-04, 09:31:12	NYSE	-500	276.0000	277.9700	138,000.00	-5.75	-128,233.01	9,761.24	-985.00	C
Total BA			**0**			**11,452.44**	**-14.87**	**0.00**	**11,437.57**	**1,066.49**	
CI	2017-10-30, 09:30:35	-	147	201.9000	197.7200	-29,679.30	-1.00	29,680.30	0.00	-614.46	O;P
CI	2017-11-06, 09:30:28	NYSE	148	201.9000	205.0000	-29,881.20	-1.00	29,882.20	0.00	458.80	O
CI	2017-11-13, 12:51:43	IEX	-7	197.9000	197.6100	1,385.30	-1.03	-1,413.37	-29.10	2.03	C
CI	2017-11-20, 09:30:32	IEX	-48	198.0500	199.4100	9,506.40	-1.23	-9,691.68	-186.51	-65.28	C
CI	2017-11-27, 10:34:05	ARCA	42	201.5100	199.9300	-8,463.42	-1.00	8,464.42	0.00	-66.36	O
CI	2017-12-04, 09:32:18	-	-282	210.6600	204.1100	59,406.12	-2.82	-56,921.87	2,481.43	1,847.10	C;P
Total CI			**0**			**2,273.90**	**-8.07**	**0.00**	**2,265.83**	**1,561.83**	
COST	2017-10-30, 09:30:09	-	-117	161.7383761	160.2300	18,923.39	-1.45	-18,921.94	0.00	176.48	O;P
COST	2017-11-06, 09:30:29	BATS	36	165.9200	165.0500	-5,973.12	-1.00	5,822.50	-151.62	-31.32	C
COST	2017-11-13, 09:30:33	-	146	170.9600	171.4600	-24,960.16	-1.00	13,099.44	-749.32	73.00	C;O;P
COST	2017-11-20, 09:30:30	-	-76	171.0400	172.2800	12,999.04	-1.31	-11,112.40	3.94	-94.24	C;O;P
COST	2017-11-27, 09:59:13	ISLAND	2	172.0000	172.6100	-344.00	-1.00	342.07	-2.93	1.22	C
COST	2017-12-04, 09:30:37	ISLAND	9	186.9300	189.5600	-1,682.37	-1.00	1,539.32	-144.05	23.67	C
Total COST			**0**			**-1,037.22**	**-6.76**	**-9,231.00**	**-1,043.98**	**148.81**	
CRBP	2017-10-30, 09:30:35	-	4,360	7.1901376	7.1500	-31,349.00	-21.80	31,370.80	0.00	-175.00	O;P
CRBP	2017-11-06, 09:30:03	-	343	7.1354227	6.9500	-2,447.45	-1.71	2,449.16	0.00	-63.60	O;P
CRBP	2017-11-13, 09:30:31	-	-661	7.2500	6.8500	4,792.25	-3.49	-4,762.50	26.25	264.40	C;P
CRBP	2017-11-20, 09:32:23	-	-587	7.0000	7.0000	4,109.00	-3.10	-4,229.34	-123.43	0.00	C;P
CRBP	2017-11-27, 09:30:49	EDGEA	80	7.1500	7.1500	-572.00	-1.00	573.00	0.00	0.00	O
CRBP	2017-12-04, 09:30:36	-	-3,535	7.819802	7.6000	27,643.00	-18.73	-25,401.12	2,223.14	777.00	C;P
Total CRBP			**0**			**2,175.80**	**-49.84**	**0.00**	**2,125.96**	**802.80**	
DATA	2017-10-30, 09:30:09	-	-317	79.0000	79.5300	25,043.00	-2.20	-25,040.80	0.00	-168.01	O;P
DATA	2017-11-06, 09:30:06	NYSE	38	74.5400	71.4000	-2,832.52	-1.00	3,001.55	168.03	-119.32	C
DATA	2017-11-13, 09:30:27	BYX	5	71.7500	71.2400	-358.75	-1.00	394.94	35.19	-2.55	C
DATA	2017-11-20, 09:30:32	NYSE	-2	70.0000	70.1000	140.00	-1.00	-139.00	0.00	-0.20	O
DATA	2017-11-27, 09:30:25	NYSE	-11	69.8100	70.3900	767.91	-1.02	-766.89	0.00	-6.38	O
DATA	2017-12-04, 09:30:58	NYSE	287	70.5000	68.7700	-20,233.50	-1.44	22,550.20	2,315.26	-496.51	C
Total DATA			**0**			**2,526.14**	**-7.66**	**0.00**	**2,518.48**	**-792.97**	
F	2017-10-30, 10:48:21	NYSE	-11,832	12.0400	12.1000	142,457.28	-63.86	-142,393.42	0.00	-709.92	O
F	2017-11-06, 09:30:29	NYSE	12	12.3300	12.3300	-147.96	-1.00	144.42	-4.54	0.00	C
F	2017-11-06, 09:30:29	NYSE	500	12.3300	12.3300	-6,165.00	-2.50	6,017.30	-150.20	0.00	C
F	2017-11-13, 09:32:17	-	2,491	12.0000	12.1600	-29,892.00	-12.46	29,978.20	73.74	398.56	C;P
F	2017-11-20, 09:30:31	-	531	12.0400	12.1300	-6,393.24	-2.66	6,390.37	-5.52	47.79	C;P
F	2017-11-27, 09:30:34	NYSE	-505	12.0900	12.1100	6,105.45	-2.73	-6,102.72	0.00	-10.10	O
F	2017-12-04, 09:30:35	-	8,803	12.6307986	12.6300	-111,188.92	-44.02	105,965.86	-5,267.08	-7.03	C;P
Total F			**0**			**-5,224.39**	**-129.21**	**0.00**	**-5,353.60**	**-280.70**	
GOOG	2017-10-30, 09:30:07	ARCA	44	1,014.4800	1,017.1100	-44,637.12	-1.00	44,638.12	0.00	115.72	O
GOOG	2017-11-06, 09:32:32	ISLAND	-44	1,030.0000	1,025.9000	45,320.00	-2.05	-44,638.12	679.83	180.40	C
GOOG	2017-11-06, 09:32:32	ISLAND	-3	1,030.0000	1,025.9000	3,090.00	-1.07	-3,088.93	0.00	12.30	O

GOOG	2017-11-20, 09:30:03	-	-11	1,020.0100	1,018.3800	11,220.11	-1.26	-11,228.06	-9.21	17.93	C;O;P
GOOG	2017-12-04, 09:30:10	ISLAND	9	1,013.9000	998.6800	-9,125.10	-1.00	9,179.88	53.78	-136.98	C
Total GOOG			**0**			**748.44**	**-7.38**	**-2,048.18**	**741.06**	**198.67**	
HUBS	2017-10-30, 09:30:05	-	185	86.1662162	85.4500	-15,940.75	-1.00	15,941.75	0.00	-132.50	O;P
HUBS	2017-11-06, 09:30:30	BYX	3	84.6500	83.4500	-253.95	-1.00	254.95	0.00	-3.60	O
HUBS	2017-11-13, 10:12:30	NYSE	-152	80.3000	79.9500	12,205.60	-1.30	-13,105.40	-901.10	53.20	C
HUBS	2017-11-20, 09:30:32	ARCA	20	80.4000	79.8000	-1,608.00	-1.00	1,609.00	0.00	-12.00	O
HUBS	2017-11-27, 09:30:42	NYSE	1	82.1000	80.5000	-82.10	-1.00	83.10	0.00	-1.60	O
HUBS	2017-12-04, 09:30:33	ISLAND	-57	81.1500	77.3500	4,625.55	-1.11	-4,783.40	-158.96	216.60	C
Total HUBS			**0**			**-1,053.65**	**-6.41**	**0.00**	**-1,060.06**	**120.10**	
JPM	2017-10-30, 09:30:15	-	2,339	101.2546815	101.4100	-236,834.70	-11.70	236,846.40	0.00	363.29	O;P
JPM	2017-11-06, 09:30:08	-	151	101.2602649	100.7800	-15,290.30	-1.00	15,291.30	0.00	-72.52	O;P
JPM	2017-11-13, 09:49:26	-	-1,028	97.4500	97.8600	100,178.60	-7.58	-104,067.70	-3,896.68	-421.48	C;P
JPM	2017-11-20, 09:30:10	NYSE	52	98.4800	99.0100	-5,120.96	-1.00	5,121.96	0.00	27.56	O
JPM	2017-11-27, 09:30:31	NYSE	100	98.1700	97.9300	-9,817.00	-1.00	9,818.00	0.00	-24.00	O
JPM	2017-12-04, 09:30:34	-	-1,614	108.1127385	106.9500	174,493.96	-12.29	-163,009.96	11,471.71	1,876.66	C;P
Total JPM			**0**			**7,609.60**	**-34.56**	**0.00**	**7,575.04**	**1,749.51**	
MFC	2017-10-30, 10:06:38	NYSE	-3,101	20.2500	20.2200	62,795.25	-17.32	-62,777.93	0.00	93.03	O
MFC	2017-11-06, 09:30:30	EDGEA	-42	20.7700	20.8400	872.34	-1.03	-871.31	0.00	-2.94	O
MFC	2017-11-13, 09:30:01	-	2,398	21.1350042	21.0700	-50,681.74	-11.99	48,546.10	-2,147.63	-155.88	C;P
MFC	2017-11-20, 09:31:54	BYX	364	20.9500	20.9000	-7,625.80	-1.82	7,368.97	-258.65	-18.20	C
MFC	2017-11-27, 09:30:31	NYSE	85	21.1700	21.0700	-1,799.45	-1.00	1,720.78	-79.67	-8.50	C
MFC	2017-12-04, 09:30:18	-	296	21.4700	21.1600	-6,355.12	-1.48	6,013.40	-343.20	-91.76	C;P
Total MFC			**0**			**-2,794.52**	**-34.64**	**0.00**	**-2,829.16**	**-184.25**	
MMM	2017-10-30, 09:30:49	-	1,004	233.8976096	231.0200	-234,833.20	-5.02	234,838.22	0.00	-2,889.12	O;P
MMM	2017-11-06, 09:30:47	-	126	232.5400	230.3100	-29,300.04	-1.00	29,301.04	0.00	-280.98	O;P
MMM	2017-11-13, 09:53:48	-	-336	227.4021429	228.2200	76,407.12	-3.48	-78,622.32	-2,218.68	-274.80	C;P
MMM	2017-11-20, 09:31:39	ISLAND	-55	229.8500	231.4900	12,641.75	-1.30	-12,869.72	-229.27	-90.20	C
MMM	2017-11-27, 09:30:34	NYSE	74	232.4700	234.0000	-17,202.78	-1.00	17,203.78	0.00	113.22	O
MMM	2017-12-04, 09:30:49	-	-813	243.5427798	239.2600	198,000.28	-8.74	-189,851.00	8,140.55	3,481.90	C;P
Total MMM			**0**			**5,713.13**	**-20.54**	**0.00**	**5,692.59**	**60.02**	
MTN	2017-10-30, 09:31:04	-	873	227.9819817	226.3700	-199,028.27	-4.36	199,032.64	0.00	-1,407.26	O;P
MTN	2017-11-06, 09:30:28	NYSE	-66	235.1100	236.7100	15,517.26	-1.37	-15,047.34	468.55	-105.60	C
MTN	2017-11-13, 11:30:20	NYSE	-84	232.0000	229.3300	19,488.00	-1.46	-19,150.66	335.88	224.28	C
MTN	2017-11-20, 09:30:21	ARCA	-70	229.9400	228.4900	16,095.80	-1.38	-15,958.70	135.72	101.50	C
MTN	2017-11-27, 09:35:57	NYSE	29	232.3700	227.3000	-6,738.73	-1.00	6,739.73	0.00	-147.03	O
MTN	2017-12-04, 09:30:49	-	-682	223.8120528	220.8600	152,639.82	-7.02	-155,615.66	-2,982.86	2,013.30	C;P
Total MTN			**0**			**-2,026.12**	**-16.59**	**0.00**	**-2,042.71**	**679.19**	
NFLX	2017-10-30, 09:33:19	ISLAND	74	199.1300	198.3700	-14,735.62	-1.00	14,736.62	0.00	-56.24	O
NFLX	2017-11-06, 09:30:08	ISLAND	11	200.0100	200.1300	-2,200.11	-1.00	2,201.11	0.00	1.32	O
NFLX	2017-11-13, 09:35:15	ISLAND	-64	191.9800	195.0800	12,286.72	-1.29	-12,745.18	-459.76	-198.40	C
NFLX	2017-11-20, 09:30:09	-	-42	193.2550	194.1000	8,116.71	-1.19	-8,250.91	-135.39	-35.49	C;O;P
NFLX	2017-11-27, 09:30:15	DRCTEDGE	-14	195.3900	195.0500	2,735.46	-1.06	-2,734.40	0.00	4.76	O
NFLX	2017-12-04, 09:30:12	ISLAND	35	189.7000	184.0400	-6,639.50	-1.00	6,792.76	152.26	-198.10	C
Total NFLX			**0**			**-436.34**	**-6.55**	**0.00**	**-442.89**	**-482.15**	
NKE	2017-10-30, 09:30:07	-	-1,531	55.7600	55.2700	85,368.56	-9.81	-85,358.75	0.00	750.19	O;P
NKE	2017-11-06, 09:30:05	-	526	55.3200	56.0400	-29,098.32	-2.63	29,326.39	225.44	378.72	C;P
NKE	2017-11-13, 09:30:16	-	285	56.0000	55.9100	-15,960.00	-1.42	15,889.77	-71.65	-25.65	C;P
NKE	2017-11-20, 09:30:27	-	454	58.6845815	59.2500	-26,642.80	-2.27	25,312.13	-1,332.94	256.70	C;P
NKE	2017-11-27, 09:30:16	-	-223	59.1700	59.6300	13,194.91	-1.45	-13,193.46	0.00	-102.58	O;P
NKE	2017-12-04, 09:30:08	-	489	60.2500	60.1000	-29,462.25	-2.45	28,023.92	-1,440.78	-73.35	C;P
Total NKE			**0**			**-2,599.90**	**-20.03**	**0.00**	**-2,619.93**	**1,184.03**	
SBUX	2017-10-30, 10:58:08	-	-2,787	54.7000	55.1700	152,448.90	-17.79	-152,431.11	0.00	-1,309.89	O;P
SBUX	2017-11-06, 09:30:04	ISLAND	-97	55.9900	56.5700	5,431.03	-1.14	-5,429.89	0.00	-56.26	O
SBUX	2017-11-13, 09:30:07	-	880	56.9000	56.6400	-50,072.00	-4.40	48,130.38	-1,946.02	-228.80	C;P
SBUX	2017-11-20, 09:30:01	-	223	56.8200	56.8100	-12,670.86	-1.12	12,196.68	-475.30	-2.23	C;P
SBUX	2017-11-27, 09:30:05	BYX	-120	56.6400	55.9100	6,796.80	-1.17	-6,795.63	0.00	87.60	O
SBUX	2017-12-04, 09:30:02	-	1,901	57.5957864	58.7600	-109,489.59	-9.50	104,329.57	-5,169.52	2,213.17	C;P
Total SBUX			**0**			**-7,555.72**	**-35.12**	**0.00**	**-7,590.84**	**703.59**	
TLT	2017-10-30, 10:56:56	-	1,214	123.6891763	124.4200	-150,158.66	-6.07	150,164.73	0.00	887.22	O;P
TLT	2017-11-06, 09:30:06	-	-103	125.9100	126.1100	12,968.73	-1.31	-12,740.07	227.35	-20.60	C;P
TLT	2017-11-14, 12:32:15	ISLAND	39	125.0700	125.2000	-4,877.73	-1.00	4,878.73	0.00	5.07	O
TLT	2017-11-20, 09:30:02	ISLAND	69	125.9300	126.3400	-8,689.17	-1.00	8,690.17	0.00	28.29	O
TLT	2017-11-27, 09:30:07	EDGEA	31	126.8800	126.6200	-3,933.28	-1.00	3,934.28	0.00	-8.06	O
TLT	2017-12-04, 09:30:16	-	-1,250	125.8900	126.6100	157,362.50	-10.03	-154,927.84	2,424.63	-900.00	C;P
Total TLT			**0**			**2,672.39**	**-20.42**	**0.00**	**2,651.97**	**-8.08**	
VYM	2017-10-30, 10:58:48	-	-4,400	82.7000	82.4600	363,880.00	-30.93	-363,849.07	0.00	1,056.00	O;P
VYM	2017-10-30, 11:00:59	-	-752	82.6800	82.4600	62,175.36	-5.29	-62,170.07	0.00	165.44	O;P
VYM	2017-11-06, 09:30:02	-	-1,052	82.8095057	82.6800	87,115.60	-7.40	-87,108.20	0.00	136.24	O;P
VYM	2017-11-13, 09:30:02	-	1,559	82.4376972	82.6200	-128,520.37	-7.80	128,918.34	390.18	284.21	C;P

VYM	2017-11-27, 09:30:26	-	-456	83.0000	83.0400	37,848.00	-3.21	-37,844.79	0.00	-18.24	O;P
VYM	2017-12-04, 09:30:10	-	5,015	85.681336	85.1900	-429,691.90	-25.08	414,942.20	-14,774.77	-2,464.05	C;P
Total VYM			**0**			**-14,290.89**	**-80.69**	**0.00**	**-14,371.58**	**-831.80**	
WM	2017-10-30, 09:30:07	-	3,340	82.5700599	81.8700	-275,784.00	-16.70	275,800.70	0.00	-2,338.20	O;P
WM	2017-11-06, 09:30:04	-	818	81.6893399	81.2400	-66,821.88	-4.09	66,825.97	0.00	-367.56	O;P
WM	2017-11-13, 09:30:08	-	-990	81.3876768	82.1400	80,573.80	-6.93	-81,676.95	-1,110.08	-744.80	C;P
WM	2017-11-20, 09:30:05	NYSE	2	80.9800	80.9200	-161.96	-1.00	162.96	0.00	-0.12	O
WM	2017-11-27, 09:30:22	BEX	-10	80.3900	80.8500	803.90	-1.02	-825.05	-22.17	-4.60	C
WM	2017-12-04, 09:30:17	-	-3,160	82.3493038	83.1500	260,223.80	-22.19	-260,287.63	-86.02	-2,530.20	C;P
Total WM			**0**			**-1,166.34**	**-51.93**	**0.00**	**-1,218.27**	**-5,985.48**	
Total						**-11,233.10**	**-565.76**	**-11,279.18**	**-11,798.86**	**2,204.76**	
Symbol	**Date/Time**	**Exchange**	**Quantity**	**T. Price**	**C. Price**	**Proceeds**	**Comm/Fee**	**Basis**	**Realized P/L**	**MTM P/L**	**Code**
Bonds											
USD											
T 2 10/31/22 0.0%	2017-10-30, 10:55:32	BARCBONDG	500,000	99.95703	100.015625	-499,785.15	-100.00	499,885.15	0.00	292.98	O
T 2 10/31/22 0.0%	2017-12-04, 08:00:22	BARCBONDG	-500,000	99.34766	99.367188	496,738.30	-100.75	-499,885.15	-3,247.60	-97.64	C
Total T 2 10/31/22			**0**			**-3,046.85**	**-200.75**	**0.00**	**-3,247.60**	**195.34**	
Total						**-3,046.85**	**-200.75**	**0.00**	**-3,247.60**	**195.34**	

References

[1] Alan Levine. Harry markowitz, father of modern portfolio theory still diversified. http://post.nyssa.org/nyssa-news/2011/12/harry-markowitz-father-of-modern-portfolio-theory-still-diversified.html, Dec 2011.

[2] Capital asset pricing model. https://www.investopedia.com/terms/c/capm.asp.

[3] Fama and french three factor model. https://www.investopedia.com/terms/f/famaandfrenchthreefactormodel.asp.

[4] Announcements, data & results. https://www.treasurydirect.gov/instit/annceresult/annceresult.htm.

[5] Activity statements. https://gdcdyn.interactivebrokers.com/Universal/servlet/AccountAccess.AuthenticateSSO?action=RM_VIEW_ACTIVITY&clt=1&mid=001.

[6] Sharpe ratio. https://www.investopedia.com/terms/s/sharperatio.asp.

[7] S&p 500 (). https://finance.yahoo.com/quote/%5EGSPC?p=^GSPC.

[8] Leverage ratio. https://www.investopedia.com/terms/l/leverageratio.asp.

[9] David Ruppert. *Statistics and finance: an introduction*. Springer, 2009.

[10] Yahoo finance nke. https://goo.gl/YBLJNb.

[11] W. James and Charles Stein. Estimation with quadratic loss. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 361–379. University of California Press, 1961.

[12] Yahoo finance data. https://goo.gl/yvjdc4.

[13] Why tableau stock dropped. http://fortune.com/2016/03/23/tableau-software-stock-drop/.

[14] Description of momentum factors for developed markets. http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/f-f_developed_mom.html.

[15] Bloomberg terminal computer. https://www.bloomberg.com/professional/solution/bloomberg-terminal/.

[16] numpy.linalg.norm. https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.linalg.norm.html, journal=SciPy.

[17] Backtest portfolio asset allocation. https://www.portfoliovisualizer.com/backtest-portfolio, journal=.

[18] Average annual return of s&p 500 index. https://www.investopedia.com/ask/answers/042415/what-average-annual-return-sp-500.asp, journal=.