INVESTMENT PORTFOLIO ANALYSIS

Financial Economics I (FIN30150)

Dr. Conall O'Sullivan

University College Dublin

Aaron Davy – 13311866

Alison Kelly – 13314261

Alison Smith – 13364011

Luke Killoran – 13434418

Table of Contents

Introduction

Initial Set-Up

The ensuing report aims to critically examine and analyse our chosen portfolio of ten industries, using a combination of high level yet in depth market analysis techniques such as the Capital Asset Pricing Model (CAPM), the Fama-French (FF) Three Factor Model, and the Cahart Four Factor Model. Throughout our testing we are determining the adequacy of both models in deducing the future portfolio returns by comparing the predicted expected returns with the actual realised returns at given time points. It also considers the validity of the long-time persistent Betting Against Beta (BAB) anomaly and the premier market anomaly – Momentum. We used both Excel and the statistical language R to carry out our analysis. The combination of the two allowed us to vary our analytical approach depending on the research requirements.

We used historical monthly industry returns data from the Professor Kenneth French data sets, as well as historical monthly market returns on the U.S Fama-French three research factors. We used the most recent 240 months of the industry data which ranged from October 1996 to September 2016. For all sections in the project, the initial 120-month in-sample period ran from October 1996 to September 2006, while the out-of-sample period ran from October 2006 to September 2016. Thoughout the project, we assumed that inflation was negligent.

Industry Selection

The project required us to select ten industries from the 49 Professor Kenneth French portfolio file. During our selection we looked at the average firm size and book-to-market ratio as we sought a well-diversified portfolio. The main threat associated with selecting any portfolio is to have an over-exposure to risk, based on an individual's risk appetite. Diversification is a technique used to reduce risk by allocating investments amongst various industries. It aims to maximise returns by investing in different industries which would each act differently to the same financial event. While there is always the inevitable systematic risk present, under-diversification of your portfolio leaves you vulnerable to avoidable non-systematic risk. By curtailing risk, we aim to reduce the chances of heavy losses. By choosing complementary industries we aim to minimise our exposure to risk while also attempting to yield high returns. We have diversified our portfolio by selecting growth, value, cyclical and defensive industries. Our selected industries are as follows (where Small and Large indicate a small and large average firm size respectively – pertaining to the market capitalization);

Industry	Selection Purpose
Drugs	Defensive/Growth/Small
Soda	Growth/Large
Agriculture	Cyclical/Large
Toys	Cyclical/Value/Small
Steel	Cyclical/Value
Software	Growth/Small
Fun	Cyclical/Value
Oil	Defensive/Large
Food	Defensive/Large
Utilities	Defensive

Value stocks are ones that exhibit a high book-to-market ratio, and growth stocks are ones with a low book-to-market-ratio. Small and large stocks are categorized as such because they have a particularly low (high) market capitalization which is the total market value of a company's shares. Cyclical stocks are those that tend to be highly correlated with the overall larger economic cycle, for example, when the economy is performing well, the positive growth is often reflected in cyclical stocks. This is in stark contrast to defensive stocks which operate near-independently of this.

Mean-Variance Portfolios

Covariance Matrix

The 10x10 covariance matrix at the 120th month, denoted *V*, measures how the mean of each industry varies with another. The covariance between industry i and industry j was calculated by computing $\sigma_{ij} = \frac{1}{120}\sum_{t=1}^{120}(r_{i,t} - \bar{r}_i)\,(r_{j,t} - \bar{r}_j)\; for\; i, j = 1,2,\ldots,10$, where $\bar{r}_i$ is the arithmetic mean return of 120 months for industry i, and $r_{i,t}$ is the return on industry i and month t, where $t = 0$ is the start of the first month in our in-sample data. When $i = j$, $\sigma_{ij} = \sigma_{ii} = \sigma_i^2$, which is the variance of returns of industry i. The covariance matrix is provided in Figure 1. The majority of our industries have positive covariance suggesting the industries tend to perform well (or poorly) when others yield fruitful (or sparse) earnings. In order to calculate the Global Minimum Variance Portfolio (GMVP), the inverse of the covariance matrix is required. Thus, the covariance matrix was inverted in Excel using the "MINVERSE()" function.

Beta

The beta of an industry is the normalised version of the covariance of an asset with the market. It is a measure of specific systematic risk, a risk which effects all investments and cannot be avoided by diversification. We calculated the beta of each industry in two ways: firstly we regressed the excess earnings of the industry over the risk-free rate against the market risk premium for the 120 months of in-sample data. Thus, we could fit a sum of least squares line with a forced intercept of zero; the security market line as predicted by CAPM. The slope of this line is our initial estimate for the beta of that industry. (The reason we forced the intercept (alpha) to 0, is because when dealing with expected returns, CAPM states that the return of a stock depends only on its systematic risk.) Secondly we calculated the covariance of that industry with the market using the formula detailed above, as well as evaluating the variance of the returns of the market (which we computed by adding the market risk premium to the risk-free rate at the end of each month) and dividing these, in accordance with the following formula: $\beta_i = \frac{\sigma_{im}}{\sigma_m^2}$. We did this for the total of the 120 months of in-sample data. Finally we obtained a pooled estimate for beta by determining the mean of these two estimates. The Betas are detailed in Figure 2. Our analysis resulted in a range of beta values for our chosen industries. A Beta value close to 0 indicates the industry is weakly correlated with market returns, where as a Beta value close to 1 implies the industry tends to move in line with the market.

Expected Future Returns

Under the assumptions of CAPM, the expected future monthly return of an industry i is given by the following formula: $\bar{r}_i(t+1) = r_f(t) + \beta_i(\bar{r}_m - \bar{r}_f)$, where $r_f(t)$ is the 1-month US Treasury bill rate (assumed to be risk-free), and $(\bar{r}_m - \bar{r}_f)$ is the long-term (30 year) monthly average for the market risk premium, calculated to be 0.6375%. The expected returns of our selected industries are shown in Figure 3. The industries with higher beta values also had higher expected returns, which is consistent with CAPM predictions.

The Portfolios

The Global Minimum Variance Portfolio of a group of risky assets is the portfolio that is well-diversified among those assets to achieve the lowest possible lever of risk on the efficient frontier, based on the assets' covariance with the market. The formula for the weights of the GMVP can be derived from the Markowitz Model using Lagrange multipliers, resulting in the following: $w_{gmvp} = \frac{1}{C}V^{-1}\mathbf{1}$, where $C = \mathbf{1}^t V^{-1}\mathbf{1}$. The Tangent Portfolio of a group of risky assets is the portfolio that maximises the return-to-variability ratio of the pooled assets. The weights of the TP are given by $w_{tangent} = \frac{1}{D}V^{-1}(\bar{r} - \mathbf{1}r_f)$, where $D = (\bar{r} - \mathbf{1}r_f)^{\mathbf{t}}V^{-1}(\bar{r} - \mathbf{1}r_f)$. The weights vector of the GMVP and TP are exhibited in Figure 4. For the market portfolio, we used a naïvely diversified portfolio in all 49 industries, instead of using, for example, the S&P 500 or the Dow Jones. We thought that this would allow us to draw more level comparisons to our other portfolios – given that they were based on the same data. The naive-weighted portfolio is the portfolio with $\frac{1}{10}$ invested in each of our chosen industries. The risk-free portfolio is formed by investing $1 in 1-month US Treasury bills; which for the purposes of this project is assumed to be risk-free.

Portfolio Performance

In-Sample Performance

We calculated the expected return and standard deviation estimates for the first month of the out-of-sample data for the GMVP and the Tangent Portfolio using the equations $\bar{r}_{GMVP} = w_{GMVP}{}^{t}\bar{r}$ and $\sigma_{GMVP} = w_{GMVP}{}^{t}Vw_{GMVP}$ respectively, for the GMVP, and similarly for the TP – to yield: $\bar{r}_{GMVP}$ = 0.6554381% monthly, $\bar{r}_{tangent}$ = 1.041268% monthly, and σ_{GMVP} = 3.131986, $\sigma_{tangent}$ = 5.022915, which is as expected – lower standard deviation of returns for the GMVP and higher expected returns for the TP. (Note: $\bar{r}$ refers to the 10x1 vector containing the expected returns for the 121st month of the data and V refers to the 10x10 covariance matrix of the previous 120 months of returns data from our selected industries.

The plot of the Mean-Variance Frontier and Efficient Line is displayed in Figure 5. We swept out the curve using the Two-Fund Theorem which derived the equations $\bar{r_p} = \propto_p \cdot \bar{r}_{GMVP} + (1 - \propto_p) \cdot \bar{r}_{TP}$ and $\sigma_p = \propto_p{}^2 \cdot \sigma_{GMVP}{}^2 + \left(1 - \propto_p\right)^2 \cdot \sigma_{TP}{}^2 + 2 \cdot \propto_p \cdot (1 - \propto_p) \cdot \sigma_{GMVP,TP}$. We varied alpha over the interval [-5,5] and stored the results. To attain the Efficient Line, we utilised the One-Fund Theorem to supply the equation $\bar{r_p} = r_f + \sigma_p \sqrt{(\bar{r} - 1r_f)^t V^{-1} (\bar{r} - 1r_f)}$, and solved for σ_p in the interval [0,8]. At the outset of each month, we recalculated our estimates of beta, expected returns, and the covariance matrix. Using these updated estimates we rebalanced the weighting scheme in each of our portfolios.

Out-Of-Sample Performance

The plot of cumulative returns of each portfolio (the global minimum-variance portfolio, the tangent portfolio, the naive-weighted portfolio, the market portfolio, and the risk-free portfolio) are shown in Figure 6, along with monthly returns of each portfolio in Figure 7, 8, and 9. The new Efficient Frontier and Capital Allocation Line (as of the 240th month of data) are exhibited in Figure 10. If we turn our attention to the plot of cumulative returns, we see that the GMVP out-performed all the other portfolios managing a 174.85% return on the initial stake, and the TP accomplished the second best returns with a 149.08% cumulative return. This should be expected since naive-diversifying one’s portfolio as is done in the naive-weighted and market portfolios is often not the optimal strategy in achieving minimal variance or maximal returns because it doesn’t consider the differing expected values and variances of the industries. The reason why the GMVP’s returns bettered that of the TP is probably due to the fact that the riskier of our chosen industries did not capture the kind of fruitful returns that would compensate the investor for bearing such a risk. The naive-weighted 10 industry portfolio narrowly edged the better returns in comparison to the market portfolio. This explanation for this result is simply that our chosen industries performed more favourably on average in comparison with the other 39 industries.

Assessment using CAPM, FF 3 Factor Model and Cahart’s 4 Factor Model

When looking to CAPM to evaluate out-of-sample performance, we encountered a similar story. The GMVP exceeded CAPM’s predictions by approximately 0.53% per month. Moreover, the GMVP was the only portfolio that rejected the null hypothesis (which stated that the portfolio’s returns did not outperform CAPM’s prediction) at the 1% level of the significance. The other portfolios failed to reject H_0 at the 10% level of significance. Also, unsurprisingly, given the focus on achieving the minimum variance possible, the GMVP returned by far the lowest beta (at 0.558) which was nearly half that of its competing portfolios. The results of the CAPM linear regression are provided in Figure 11.

The adjusted-R-squared of the market risk premium for the returns of the GMVP was only 0.615, so we expanded our assessment of the returns to the Fama-French 3-Factor Model, and Cahart 4-Factor Model, in the hope of unveiling the reasons why the GMVP earned the best returns. This allowed to examine whether the portfolios yielded more favourable results than predicted by the "Small-Minus-Big" effect (which is that stocks with a small market capitilization generally outperform stocks with a large market cap. relative to CAPM), the "High-Minus-Low" effect (which is that value stocks; those with a high book-to-market ratio, generally outperform growth stocks; those with a low book-to-market ratio), and finally the "Momentum" effect (which is that stocks tend to maintain current results in the short-term). These effects were accounted for by utilising the monthly SMB, HML, and MOM factors supplied on the Fama and French homepage. They derived these factors by recording the monthly returns on two (monthly-rebalancing) well-diversified portfolios, one that goes long on small market cap. stocks, and the other that shorts large market. cap stocks, for example, and similarly for the HML and MOM factors.

The results of the regression using the FF 3-Factor Model, and the Cahart 4-Factor Model are exhibited in Figure 12, and 13. Again, the GMVP is the only one that significantly exceeded the each model's predictions at the 1% level of significance, on average by 0.47%. Conversely, the other portfolios failed to surpass both model's predictions at the 10% level of significance. Furthermore, we can see that the returns of the GMVP decreased as the SMB effect strengthens at the 1% level of significance. This is because the GMVP tended to be weighted in favour of industries with a larger average firm size. However, we can see that the returns of the GMVP climb as the MOM effect grows at the 1% level of significance, by an average of 0.13% per percentage increase in monthly returns of the MOM portfolio. Whereas the returns of the Tangent Portfolio, the naive-weighted portfolio, and the market portfolio shrank as the MOM effect heightened at the 5% level of significance. Perhaps this effect is a confounding factor in hindsight – given that the GMVP, the naïve-weighted portfolio, and the market portfolio were constructed without much regards for the short term performance of its constituents. Or perhaps, this is even more evidence that the riskier stocks fared poorly considering their relative risk, where their volatile nature meant that they would not maintain their current returns – especially after positive ones. Moreover, given the moderate exposure to risk that is associated with naïve-weighted portfolios, it easy to see how this could majorly impact the yield of naïve-weighted portfolio and the market portfolio. Other reasons why the TP and the naïve-weighted portfolio did not match the GMVP's returns could be because they were significantly negatively influenced by the HML effect (at the 1% level of significance) with 0.23% and 0.15% drop in monthly returns per percentage increase in the monthly returns of the HML portfolio. This could be due to the fact that the tangent portfolio and the naïve-weighted portfolio was more heavily weighted in the direction of low market cap. industries – since those with low market cap. had high expected returns, and were more prevalent among our selected industries respectively.

Financial Statistics

For further and more thorough analysis, we produced some financial measures that may better depict the differences between the four portfolios. These statistics are displayed in Figure 14. Unsurprisingly, GMVP exhibits the lowest standard deviation of returns. Yet, simultaneously boasts the highest arithmetic and geometric mean returns. The GMVP and TP's returns are more negatively skewed than that of the other portfolios. This means that they have a higher probability of achieving larger returns based on past performance. The market portfolio is subject to high kurtosis in its returns, more than twice that of the GMVP. This indicates a more substantial probability of earning large positive or large negative returns ("higher peaks, heavier tails"), which can be viewed as a measure of risk. Both the Tangent Portfolio and the naïve-weighted portfolio seem to be strongly positively correlated with the market, considering high values of the Pearson and Spearman's correlation coefficients. This means that the portfolios display a strong increasing linear relationship with the market, and their returns increase monotonically with that of the market. Thus explaining why their returns are so similar.

The 1-month non-parametric Value-At-Risk (with confidence level = 95%) was calculated by finding the ((1-0.95)*120+1)th = 6th largest monthly loss of each of the portfolios. Hence, there is a 5%

probability that the GMVP portfolio bears a monthly loss of greater than 4.98%. This is much less than the other portfolios which are subject to a monthly loss greater than 8% with 5% probability. This speaks to the amount of excess risk the other portfolios are exposed to in comparison to the GMVP. The 1-month non-parametric Expected Tail Loss (with confidence level = 95%) was evaluated by computing the sum of the 5 largest monthly losses and multiplying this figure by a weighting constant equal to $\frac{1}{5}$, and this found that if the GMVP absorbed a monthly return of less than -4.98% (= 95% VaR), the expected loss would be approximately -7.28%, much better than that of the others which are between -10.9% and -12%. Parametric estimations of VaR and ETL tend to better account for the tails of the distribution by more precisely estimating the shape of the distribution. The most commonly assumed parametric distribution that returns follow is the Normal distribution, and thus VaR at confidence level 95% would be equal to $-(\bar{r}_p - \sigma_p \cdot Z_{0.95})$ where $\bar{r}_p$ is the historical mean of the portfolio, σ_p is the standard deviation of returns and $Z_{0.95}$ is the 95th quantile of positive returns. However, this estimate for VaR can be significantly inaccurate when dealing with returns that do not adhere to a Normal distribution. Thus, what is widely cited in literature as “modified” VaR was invented by the Zangari (1996), Favre and Galeano(2002) and takes the higher moments of non-normal distributions (skewness, kurtosis) into account through the use of a Cornish Fisher expansion to provide a more robust estimate of VaR. Hence to calculate the 1-month “modified” VaR (with CL = 95%), we computed $-(\bar{r}_p - \sigma_p \cdot Z_{0.95}{}')$ where $Z_{0.95}{}' = Z_{0.95} + [({Z_{0.95}}^2 - 1) \cdot Skew(r)]/6 + [({Z_{0.95}}^3 - 3Z_{0.95}) \cdot Kurtosis(r)]/24 - [(2{Z_{0.95}}^3 - 5Z_{0.95}) \cdot Skew(r)^2]/36$. The “modified” VaR reinforced the analysis of the non-parametric VaR. The “modified” ETL was assessed by finding the expected value of the loss given it was in the $Z_{0.95}{}'$ quantile, and once again it rather reconfirmed the abnormity of the difference in risk borne by the GMVP and other portfolios. The semi-deviation and downside deviation indicate the deviation of returns below the mean and some minimum acceptable return respectively, which provide helpful measures of risk. We found that the standard deviation and downside deviation below the long run risk-free rate, and the long run market risk premium were all much lower for the GMVP than all the other portfolios, which is consistent with the analysis above.

We then evaluated Jensen’s index, or alpha, which can be used to determine roughly the compensation deserved by a fund manager, and found that only the GMVP’s alpha was positive, with the precise implication being that the GMVP surpassed the CAPM’s predictions by 0.176%, where the TP, the naïve-weighted portfolio, and the market portfolio all fell short by 0.124%, 0.143%, and 0.188% respectively. The GMVP also had nearly twice the Sharpe Ratio as the other three at around 0.15%, compared to approximately 0.09%. This means that the GMVP attained a higher risk-adjusted return, and thus was more efficient than the other three. In fact, the GMVP is even more efficient than any of the ‘efficient’ portfolios along the Capital Allocation Line (which has a slope of 0.13%). But the other three portfolios could be subject to unsystematic risk, and therefore, it may be more prudent to adopt Treynor’s measure (which substitutes the standard deviation of returns for the systematic risk embedded in the returns – estimated by the portfolio’s beta) to assess risk-adjusted return. Due to the GMVP’s low beta, it is again no wonder that its risk-adjusted return dwarfs that of the other three; clocking in with a 0.95% return per unit of systematic risk; twice that of its competitors. The Sortino’s Ratio adopts the downside deviation as its assessment of risk, and as a result, we also evaluated the discrepancy between risk-adjusted returns using the metric as well. All measurements of risk-adjusted return have produced the exact same ranking of portfolios – the GMVP, the TP, the naïve-weighted, followed by the market portfolio. This is entirely consistent with our review thusfar.

Stochastic Dominance

We then sorted each portfolio and industry’s monthly returns over the 120-month out-of-sample data, and for each 0.1 interval between -33 and 40, we evaluated the empirical cumulative distribution function of each by simply counting the number of returns worse than that point and dividing this value by 120. We also found the area under the each curve for every 0.1 interval between -33 and 40 by evaluating the ECDF at small intervals below that point and summing these figures. None of the portfolios first order stochastically dominated each other, since none of them were completely below (to the right) the another for the whole interval [-33,40]. Although it appeared that the GMVP may second order dominate the others since its ECDF was below the others at lower returns, and above at higher returns. This can be seen in Figure 15. Thus, we plotted a graph of the area under the ECDF curve of each portfolio minus that of the GMVP, which is displayed in Figure 17. Since the area under the ECDF curve of each portfolio is greater than that of the GMVP at all points, this means that the GMVP second order stochastically dominates the other portfolios. In other words, if each portfolio was to behave as past performance indicated, then any investor with a non-decreasing and concave utility function would prefer the GMVP if they cared only to maximise their utility. We followed the same method to examine whether the GMVP or TP first or second order dominated any of the individual industries. Figure 18-29 graphically detail our findings which were that neither the GMVP nor the Tangent Portfolio first order stochastically dominated any of the individual industries. Although, the GMVP second order stochastically dominated the Steel, Oil and Utilities industries, whereas the TP second order stochastically dominated the Steel, Oil and Toys industries. Also, rather remarkably, the food industry second order stochastically dominated the Tangent Portfolio.

Betting against Beta

Introduction

Betting against Beta, an anomaly which has been prevalent ever since Fischer Black explained its effects to the management of Wells Fargo in 1970 in the hope of persuading them to weight the first index fund (which he helped set up) more in favour of low beta assets and less in favour of high betas assets, as the story goes[1]. Of course, management of the firm refused, and unfortunately Black would never live long enough to test the proceeds of this remarkable and resilient anomaly, first documented by himself, Jensen and Scholes[2]. This anomaly not only goes against traditional financial economics theories but also promises considerable risk-adjusted capital appreciation if executed correctly, and applies across all asset classes. It capitalises on the inefficiencies within the Capital Asset Pricing Model, the manifestation of which can be explained by the flatness of the Security Market Line for U.S. stocks relative to the CAPM (Fama actually found this to be a negative slope in 1972)[3]. There are many schools of thought as to the reasoning behind such a phenomenon.

Some believe the principle that all invest in the portfolio with the highest expected return per unit risk, which tended to be high-beta assets. While others propose that investors are often limited by leverage or margin constraints, and thus consequently cannot re-weight their portfolio with the risk-free rate to attain a higher expected return along the Efficient Line (Capital Allocation Line). This forces them to overweight risky securities, and the sheer demand for these securities inflates their prices and conversely the lack of demand for safe securities causes their price to decrease. This was the reason put forth by Frazzini and Pedersen in their 2013 paper "Betting against Beta[4]". Others opine that, in fact, many investors are drawn to possibility of earning large returns in a short-time frame, and this generates what is termed "lottery demand" for high-beta stocks, which drives their prices higher, deflating their risk-adjusted return. This was the explanation provided Bali, Brown, Murray and Tang in their 2014 paper "Betting against Beta or Demand for Lottery[5]."

A very basic initial way of capturing the extent of this phenomenon is to plot excess returns of each industry against those of the market, and evaluate the slope of the SML line, which is what we did in Figure 30 and 31, using 120 months and 240 months of returns. As is displayed the slope of the SML line relative to CAPM seems much shallower than what we would have previously expected based solely on our inexperienced intuition. However, we need a more robust way of evaluating whether this is true.

Method of Testing

In order to test this effect, we thought the best way would be to construct a portfolio based on the principle of going long on so-called 'low' beta stocks and shorting 'high' beta stocks. However, in

[1] https://www.youtube.com/watch?v=aHE0queiTuA
[2] http://pages.stern.nyu.edu/~lpederse/papers/BettingAgainstBeta.pdf
[3] https://www.youtube.com/watch?v=4QvF6YdyWAo
[4] http://www.econ.yale.edu/~af227/pdf/Betting%20Against%20Beta%20-%20Frazzini%20and%20Pedersen.pdf
[5] http://business.gwu.edu/wp-content/uploads/2014/12/Finance_Seminars_12.02.2014_LotteryDemand.pdf

order to eradicate the effects of the other well-known anomalies such as the "Small Minus Big" and "High Minus Low" effects, and thus create a more thorough test, we ranked industries based on their mean average firm size and sum of book equity / market equity over the 6 months previous to October 1996 – the start of our out-of-sample period. We then selected industries that ranked in the top 10 and bottom 10 of each ratio, so long as they were not ranked highly or lowly in the other ratio. We also ensured that the collection of industries included ones classed as cyclical and defensive, to allow us to better hedge risk. The results of which yielded 4 portfolios each containing 8 industries: a large portfolio, a small portfolio, a value portfolio and a growth portfolio. We found an industry's beta in the same way as in the portfolio performance section – yielding an equally- pooled estimate of the CAPM linear regression estimate and the matrix derivation estimate, based on the returns since the first month of our in-sample data.

At the start of every month we ranked the betas of each portfolio, and split them into two portfolios, one containing the 4 higher betas, the other the 4 lower. We weighted each industry within each portfolio based on its rank r_i, where the weight of industry i in the long portfolio is given by $w_i = \frac{1}{2} - \frac{r_i}{\sum_1^4 r_i}$, and the weight of industry j in the short portfolio is given by $w_j = \frac{r_j}{\sum_1^4 r_j}$, so as to afford a higher weighting to the lower beta stocks in the long portfolio and the higher beta stocks in the short portfolio – thus, magnifying the effects of the betting against beta anomaly. Notice that the weights of both portfolios sum to 1. The amount invested in the each long portfolio, and short portfolio was $1/\beta_p^L$ and $-1/\beta_p^S$ respectively, where β_p^L is the beta of the long portfolio and β_p^S is the beta of short portfolio. Note that $\beta_p^L = \sum_{i=1}^4 \beta_i w_i$ and similarly for the short portfolio. This strategy was adopted in order to ensure the portfolios each containing its category's long and short portfolio all boasted a beta of 0 since: $\beta_p^T = \beta_p^L w_L + \beta_p^S w_S = \beta_p^L \left(\frac{1}{\beta_p^L}\right) + \beta_p^S \left(-\frac{1}{\beta_p^S}\right) = 1 - 1 = 0$. Now we also impose the constraint that the monetary amount we invest in each of the 4 portfolios is equal to 0. This means that we also borrow $1/\beta_p^L$ and invest $1/\beta_p^S$ at the risk-free rate of return. We now have fabricated 4 portfolios which have a zero portfolio beta and a zero expected return. Therefore, our total BAB portfolio has a zero portfolio beta and a zero expected return. (In the interest of minimising the effect of both of the other aforementioned anomalies, we invest $\frac{1}{4}$ in each of the portfolios.) The actual return on the one of the four portfolios at time t+1 is given by $r_{t+1}^{BAB} = \left(\frac{1}{\beta_p^L}\right)\left(r_{t+1}^L - r_f\right) - \left(\frac{1}{\beta_p^S}\right)\left(r_{t+1}^S - r_f\right)$, where r_{t+1}^L, r_{t+1}^S, is the return on the long and short portfolios at time t+1 respectively, and r_f is the 1-month US Treasury bill rate (assumed to be risk-free). This is the case since whatever we invest in our long portfolio must be borrowed at the risk-free rate of return, and whatever amount we short, the proceeds of which is invested in the risk-free asset.

The Results

The cumulative returns of this portfolio over the 120-month out-of-sample data are provided in Figure 32, along with that of the market portfolio from before. As we can see, the BAB portfolio returns were quite underwhelming in comparison to that of the market portfolio, yielding a return on investment of 63.02% compared to the 129.56% earned by the market. (Although it is important to note that the portfolio achieved these results with theoretically no systematic risk and an expected rate of return of 0, which is promising.) However, the breakdown of the four sub-portfolios' cumulative returns, which is provided in Figure 33, details the reason for these under-par returns – which is that the value portfolio performed exceptionally poorly, and since the BAB portfolio is a linear combination of these 4 lines, this exerts major downward pressure of the cumulative returns of the total portfolio. Furthermore, we monitored the beta of the long portfolio and the short portfolio in each category, and as is displayed in Figures 35-38, the discrepancy between the beta of the long and short portfolio of the value industries is negligent, even negative at one point. Therefore, although the value stocks BAB portfolio earned largely negative returns, this cannot be attributed to the fact that this portfolio longed 'high' betas and shorted 'low' betas – because in simple terms the 'high' betas weren't much higher than the 'low' betas. Understandably, when selecting 4 high beta industries and 4 low beta industries that are quite similar to begin with, it is quite probable that there isn't a substantial difference between all the beta values. Thus, we admitted that with only 49 stocks to choose from, it was not possible to select them in such a way so as to allow for the effect of the other anomalies whilst ensuring each portfolio had a large in discrepancy in its long portfolio betas and short portfolio betas.

New Method of Testing

As a result we were disappointed that were not able to display the extent of the strength of the betting against beta phenomenon, so we decided to retest the BAB factor using a different approach which was to build a portfolio that would focus solely on magnifying this effect, disregarding any other constraints. Thus, at the start of each month we would calculate the beta for every industry and rank them. Then we would select the 10 highest and lowest betas for which to short and long. The weighting scheme within each portfolio was the same as before where the weight of industry i in the long portfolio is given by $w_i = \frac{1}{5} - \frac{r_i}{\sum_1^{10} r_i}$, and the weight of industry j in the short portfolio is given by $w_j = \frac{r_j}{\sum_1^{10} r_j}$. Notice that the sum of these weights in both cases is 1. Also, as before the amount invested in the long portfolio and short portfolio was $1/\beta_p^L$ and $-1/\beta_p^S$ respectively, in order to achieve a portfolio with theoretically zero systematic risk, and as well as that we ensure a zero expected return on the portfolio investing 0 funds in aggregate.

The Results of New Method

The cumulative returns on our new BAB portfolio is quite impressive managing to yield a total return on investment of 269.87%. It is clear that when we dash the constraints impinged by the HML and SMB factors our new BAB portfolio dwarfs the returns of all the other portfolios examined thusfar, with soaring earnings more than twice that of the market – and this with no systematic risk and a zero expected rate of return. Given that the beta of the long and short portfolios were relatively stable, fluctuating around 0.5 and 1.5 respectively, we can be sure that the effect of the betting against beta phenomenon is amplified here. Thus, the powerful performance of our new BAB portfolio would seem to strongly indicate the presence of an anomaly. Although, we should also be quick to point out that disregarding the effects of the other anomalies can lead oneself exposed to confounding factors and spurious regression.

Financial Statistics

As can be seen in its arithmetic and geometric mean returns, displayed in Figure 48, the T10&B10 portfolio vastly surpassed the other portfolios based on return on investment. It was the only portfolio to earn significantly excess returns that CAPM predictions at the 1% level of significance, and the only one with a positive Jensen's alpha (equal to 0.46%). Although, it was subject to higher variability in returns with a large kurtosis (equal to 6.48), a beta of 0.67, a 1-month VaR (with a CL of 95%) equal to 9.5% with a subsequent 1-month 95% CL ETL of 21.15%. (The "modified" estimate here seems appropriate given the large kurtosis indicating non-normality.) However, it attained the largest risk-adjusted returns whether you measure this by total risk (Sharpe Ratio – of which it was more efficient than Capital Market Line), systematic risk (Treynor's measure), deviation below the mean return, or some minimum acceptable return (Sortino Ratio), or unsystematic risk (given by the Appraisal ratio – which indicates that the T10&B10 BAB portfolio could do even better by diversification). In conclusion, our research strongly advises investors to invest in the T10&B10 BAB portfolio rather than the LSVG BAB portfolio or even the market portfolio – assuming past performance is characteristic of future yields, and that our sample data is representative of the population of industry returns. Note that we could have regressed the returns of the BAB portfolios against the market risk premium, the SMB factor, and the HML to investigate these whether returns really were abnormal and unexplicable by the other factors (by examing whether the residuals were random, through QQ-plots, and Kolgomorov Smirnoff tests) – thus indicating the presence of an anomaly.

The TED Spread

Now that we strongly suspect that the betting against beta anomaly is prevalent, we would like to investigate the rationale behind its existence. Firstly, we wanted to explore the leverage constraint explanation so we adopted the same approach as Frazzini and Pedersen, and nominated the TED spread as our empirical proxy for the funding conditions within the market. Note that we could have used the US Treasury bill rate or the Financial Sector Leverage = Financial Sector Total Assets / Market Value of Equity instead.

The TED spread can be utilised as a gauge of credit risk and is defined as the difference between the three-month Eurodollar LIBOR (London Interbank Offered Rate: which emulates the credit ratings of corporate borrowers) and the three-month US Treasuries rate (considered risk free - r_f). An increasing TED spread is indicative of an increasing default rate which tightens pre-existing leverage constraints on investors. During times of narrowing funding liquidity constraints, the BAB factor realises negative returns as the dispersion of betas is significantly lower. In simple terms, a high TED spread is correlated with a simultaneous low BAB return. This can be seen in Figures 42 and 43, where we plotted the returns minus risk free rate of both our BAB portfolios (the first being the spilt portfolio with regards to small, large, growth and value and the second being the top 10/ bottom 10 portfolio) against the inverse of the TED Spread. The inverse of the TED spread slightly mimics that of the troughs and valleys of the returns of BAB portfolios. Moreover, we found the (Pearson) correlation of the TED spread with the returns of each of the BAB portfolios to be -0.3847243 and -0.3430903 respectively, which is encouraging.

For a more conclusive test, we regressed the returns of both BAB portfolios on firstly the TED spread and secondly the TED spread and the market risk premium, the results of which are provided in Figures 44 and 45. It is clear that the TED spread explains some of the variation in BAB returns at the 10% level of significance. However, most of the variation is explained by the Market Risk Premium, which is significant at the 0.01% level of significance. This can be seen in the adjusted-R-squared (which takes into account the number of parameters in the model). It is only roughly 0.125 (taking averages) when only regressing on the TED spread but jumps to approximately 0.4 after including the market risk premium. In conclusion, Frazzini and Pedersen found quite conclusive evidence that the TED spread is highly correlated with negative BAB returns, but our analysis cannot culminate with the same conclusion – only to acknowledge that there is some possibly weak negative correlation between the two. Perhaps this is because the BAB portfolios we constructed were established using bundles of stocks from the same industry, and therefore the betting against beta effect is slightly nullified. For instance, low beta industries could in theory contain a small number of stocks with extremely large betas, which are inefficient, and thus even though the weighted average beta is low, the betting against beta effect is reduced.

Lottery Demand

While the BAB factor seems, in practice, to produce abnormal returns of roughly 6.6% per annum and be one of the most persistent anomalies in empirical asset pricing research, Bali, Brown, Murray and Tang felt that funding liquidity sensitivity failed to explain BAB returns which lead them to analyse alternative motives. Their theory is that the BAB factor is in fact driven by demand for lottery-like assets, which is considered a behavioural phenomenon. Included in their paper is that when beta-sorted portfolios are constructed to be neutral to lottery demand, the beta anomaly is no longer detected.

Lottery investors, by definition, are those that operate to achieve a high probability of large capital gains as rapidly as possible, and this leads them to invest in high beta stocks. Due to the sheer demand on high beta stocks, their future returns decline because of price pressure. To proxy for this lottery-demand Bali, Brown, Murray and Tang formed their own variable, FMAX, which was the mean of the top 5 daily returns of a stock in a calendar month. The rationale was that stocks with abnormally large daily returns in a month are subject to lottery demand price pressure. We would have loved to test this but since we were dealing with large bundles of stocks from the same industry, measuring lottery demand for an industry felt like a futile process. This is because the information of these abnormal returns are lost when taking the average return of many stocks. Rather we attempted to unearth the market lottery demand. We did this using the MAX variable which was calculated for every month, and is defined is defined as the mean of the highest five daily industry returns over the past month. The logic was similar in that if the market experienced large abnormal returns one month this would lead to an in-flux of lottery investors the next month – thus we lagged the MAX factor by one month. We measured the (Pearson) correlation between the lagged MAX factor and the returns on the two BAB portfolios and found these to be -0.2269459 and -0.1346779 respectively. This is what we would have expected since an increase in the MAX factor leads to a growth in lottery demand which distorts the SML even more, providing a platform for further profits from our BAB portfolios.

Like the TED spread, we needed a more conclusive test, so we regressed the returns of both BAB portfolios on firstly the lagged MAX factor and secondly the lagged MAX factor and the market risk premium, the results of which are provided in Figures 46 and 47. Although significant at the 1% level of significance when it is the only variable in the model, the MAX variable explains less than 5% of the variability in BAB returns. Moreover, when the market risk premium is included in the model, the MAX factor becomes insignificant at the 50% level of significance for the T10&B10 BAB portfolio and at the 5% level of significance for the LSVG BAB portfolio. In conclusion, Bali, Brown, Murray and Tang found quite conclusive evidence that the FMAX variable is highly correlated with negative BAB returns, but as with the TED spread, our analysis cannot culminate with the same conclusion – only to acknowledge that there is some possibly weak negative correlation between the two. Perhaps we could conduct a better study of this if we could test the lottery demand for each stock as a predictor for its future performance, and build a zero systematic risk, zero investment portfolio to based on this principle.

The Momentum Anomaly

Introduction

The term momentum arises from the days of Isaac Newton, where he explained it to be the quantity of motion an object has. In finance, the term has come to mean the tendency for rising stocks to rise further and for falling stocks to continue to fall, a phenomenon that contradicts the "weak form of efficient markets" set out by Fama in 1960[6]. Dubbed the "Premier Anomaly" by Fama, if one invested so as to profit off this anomaly ever since the inception of the US market, they would earn a larger return than by any other anomaly, or the market itself[7]. It is no surprise that it has been one of the most well-known and prevalent anomalies ever since Levy recognised that "superior profits can be achieved by investing in securities which, historically have been relatively strong in price movement", in 1967[8]. Whereas, in 1993, Jegadesh and Titman found that constructing a portfolio which longs 'winners' and shorts 'losers' in the previous 3-12 months generated significantly abnormal risk-adjusted returns in the US from 1965 to 1989[9]. They also updated their earlier study in 2001 to find that momentum effect was still persistent.

Although, rather remarkably, the exact cause of this phenomenon is shrouded in doubt – with nothing alone being able to fully describe it. Causes explaining momentum range from overconfident traders, the bandwagon effect, data-mining, macroeconomic variables, under-reaction to good news – coined "anchoring", model misspecifications etc, many of which revolve around behavioural inefficiencies. However for every explanation declared, a counter-one is proposed. For instance, some believe the reason to be that overconfident traders can earn greater profits by out bidding normal, rational investors which caused momentum to occur by inflating the price to exceed its fundamental and correct value. Others opine that momentum is created by incessant over-reaction by overly confident traders attributing too much value to their own knowledge rather than public knowledge and thus induces a deviation from a stock's equilibrium. While others deem its cause to be that investors

[6] http://efinance.org.cn/cn/fm/Efficient%20Capital%20Markets%20A%20Review%20of%20Theory%20and%20Empirical%20Work.pdf
[7] https://www.youtube.com/watch?v=lf1u4dsHavs
[8] http://www.kentdaniel.net/papers/unpublished/djk_2012_9.pdf
[9] http://www.bauer.uh.edu/rsusmel/phd/jegadeesh-titman93.pdf

contribute to herd mentality (the bandwagon effect)[10]. Given that investors are unwilling to underperform their peers they can sometimes tend to imitate their actions them instead. This is evident when short term investors use recent stock performance as a trading signal and long term investors look to recent performance to confirm their beliefs. When these investors collide, persistent price deviations can occur. Whatever one believes, there is no debate as to the existence of momentum.

We initially plotted and regressed the excess returns of all industries (over the risk-free rate), at the 120th and 240th months of our data, against our proxy for an industry's previous 3-month momentum, which was the geometric average of its previous 3 monthly returns. The results of which are plotted in Figures 49 and 50, which show the sum of least squares line to have quite a significant slope, indicating the possible presence of the momentum anomaly. However, more substantial tests were required.

The Runs Test

We set out to investigate and examine the effect of momentum. Firstly, we wished to evaluate whether the Market Efficiency Hypothesis detailed by Fama in 1960 (particularly the weak form) was evident within our industry data. If so, then it would be nigh on impossible to earn significant consistent abnormal profits from investing based on the momentum principle. Thus, we wanted to test whether the time series of returns for the last 240 months before October 2016 from each of the 49 industries followed a random walk. Hence, we conducted a Runs test for randomness, which compares the runs of the series to the expected number of runs in a random series of equal size. We defined a run as a series of returns that were consecutive returns greater than or less than the arithmetic mean of the returns. The test had the added bonus of not requiring the returns to be normally distributed. So, H_0: Series is a Random walk and H_A: Series is not a Random walk.

All industries except "Telcm" (0.042969) and "R|Est" (0.000401) admitted p-values lower than 0.05. Thus, at the 5% level of significance, we failed to reject the null hypothesis that all the other 47 industries had returns that followed a random walk. This was discouraging since it demonstrated that perhaps the momentum effect wasn't very prevalent in our data.

The Augmented Dickey-Fuller Test

We continued our line of testing with the augmented Dickey-Fuller test which allowed us to test whether our time series of returns had a unit root or a trend term. The rationale was that if we found the industry returns to be trend stationary this may show the momentum effect. We chose to conduct the ADF test on a run of 90 days of consecutive returns (18 times) for each industry, with all return periods within the last 240 months. Here H_0: Series is non-stationary and H_A: Series is stationary. We found that each industry exhibited at least one (usually considerably many) 90 day stretches of returns that were non-stationary with quite a few p-values lower than 0.01, and thus, rejecting the null hypothesis at the 1% level of significance.

The KPSS Test (Kwiatkowski–Phillips–Schmidt–Shin)

Following the results of the ADF tests, it was imperative to discover whether the root of the non-stationarity embedded in the industry returns was due to a continuing trend, and that is why we adopted the approach of the KPSS test. Here H_0: X_t is trend stationary H_A: X_t is a unit root process. Similarly to the ADF test we ran a test of 90 days of consecutive returns (18 times) for each industry with all time-horizons in the last 240 months. For less than a quarter of industries we found a run of 90 days that had a p-value lower than 0.05, but these were rare. In fact, we never encountered a p-value upwards of 0.1 which strongly indicates that the cause for non-stationarity may not be due to a trend – rejecting the null hypothesis at the 10% level of significance. Therefore there isn't much proof that momentum is existent within our data. This although unpromising, is consistent with our analysis thusfar.

[10] http://www.stockopedia.com/content/what-really-causes-price-momentum-69873/

Autocorrelation Plots

Autocorrelation measures the strength of the correlation between consecutive (and lagged consecutive) values of a time series. Hence, if autocorrelation is strongly positive for some lag k, this could be interpreted as momentum, since this means that the value of every k^{th} return is usually maintained in the next k^{th} return, whether positive or negative – the very definition of momentum. We calculated the autocorrelation of all 49 industries at various monthly lags, and plotted the resulting ACF plots, three of which are displayed in Figures 51, 52, and 53. "RlEst" showed the most compelling correlation within the time series in that we had a (large relative to the others) significant 1st lag, yet, in truth, this was only around 0.2. Most other autocorrelation plots didn't seem to have any significant lags with most lags less than 0.1 such as "Food" and "Util", and so, this hinted at minimal correlation between past and future returns – not very indicative of the momentum effect.

Momentum Portfolio

We longed for a more succinct method of testing this phenomenon, and we had discovered that many had succeeded in the past by assembling their own momentum portfolio, as per the (Barberis and Shleifer - 2003) hypothesis that style-based momentum strategies are more profitable than traditional ones, for instance. The 'style investment' for this portfolio was similar to the process carried out in the betting against beta section, where this involved placing funds in broad categories of assets such as Large, Small, Growth and Value industries. Here, we also had 8 industries in each, and split them in half – one long portfolio, one short. Moreover, we maintained the constraints that 0 funds must be invested, and each resulting sub-portfolio (consisting of one long and short) would bear 0 systematic risk – so that the final portfolio achieved by investing $\frac{1}{4}$ in each sub-portfolio boasted 0 systematic risk and an expected return of 0. The only difference was that instead of ranking the industries by their respective betas, we instead computed the geometric average of their previous 3 monthly returns, and allocated the industries with the highest values of this momentum proxy to the long portfolio; and the lowest to the short portfolio. Furthermore, it is important to note that we placed an equal weighting on these assets (as opposed to affording higher and lower weights to those with larger and smaller values of momentum). This was because none of our research particularly indicated that stocks performing extremely well (or poorly) would continue to yield extremely fruitful (or abysmal) earnings, only that they would remain producing satisfactory (or unsatisfactory) results. Thus, for clarity, we selected a monthly rebalancing momentum portfolio with a 3 month time-horizon: a momentum portfolio with a formation period of 3 months and a holding period of 1 month.

The cumulative results of the portfolio, compared with that of the market portfolio, are presented in Figure 59. Although under-performing the market portfolio, our momentum portfolio managed this with 0 funds invested and 0 systematic risk. Observing the plot of the breakdown of sub-portfolio returns provided in Figure 60, we see that the value portfolio vastly surpassed the earnings of its peers. This provided evidence that perhaps the strict limitations imposed by longing and shorting 4 industries of each category (considering we only have 49 in total to choose from), inhibited the possible earnings of our momentum portfolio. Thus, we sought to dash these restrictions and amplify the momentum effect.

New Momentum Portfolio

Again, similar to the betting against beta section, we devised a new portfolio that would rank all 49 industries by their 3-month momentum, long the top 10 and short the bottom 10. (Note we maintained an equal weighting scheme as above.) The cumulative returns of this portfolio are displayed in Figure 66. As we can see, the portfolio suffered a massive shortfall around the 30th month of our out-of-sample data, and consequently never recovered, earning below the risk-free rate for a period of 50 months. As can be viewed in Figure 68, the weighted average momentum of the long portfolio far exceeded that of the short portfolio at all times – if our proxy for momentum was appropriate then the fact that the portfolio's returns were dwarfed by that of the market demonstrates how weak the momentum effect appears to be within our data. Perhaps this is because the momentum effect is slightly muted when bundling a large number of stocks, and recording their average as the industry return. This way stocks performng well can be overshadowed by those performing poorly and it not possible given our dataset to seperatethe two.

Other Momentum Portfolios

It was quite conceivable that the momentum effect was strong and influential within the industry data, but we just adopted the wrong formation or holding period. Accordingly, we tested both of the above portfolios with formation periods of 12, 6 and 1 month. For a holding period of 3 months, we also executed a test of the above portfolios with formation periods equal to 12, 6 and 3 months. Ergo, we had uncovered the possible returns for 14 different portfolio constructions to test the momentum anomaly, and all failed to improve upon the market portfolio. Each plot of cumulative results is arranged in Figures 53-58 and 71-76. Once more, the momentum effect signals its weakness in affecting the returns of the 49 industries. Perhaps, on reflection, we could have examined daily returns, and weekly portfolio returns. (We initially deviated from this idea because we thought that even if there was a significant momentum effect, in practice it would be mitigated by the consummate transaction costs associated with a daily or weekly rebalancing portfolio.)

Residual Momentum

Instead of remaining stubborn that our momentum proxy was relevant and appropriate, we opted for a completely different approach – residual momentum. Here we constructed a Large, Small, Value, Growth (LSVG) split momentum portfolio along with the Top 10 & Bottom 10 momentum portfolio, both with formation period equal to 3 months and holding period equal to 1 month. However, in this case, the rating agent was that of an industry's performance relative to CAPM.
At the outset of each month, for each industry, we regressed that industry's excess monthly returns on the market risk premium using the data from each month since the 1st month of our in-sample data. We evaluated the sum of least squares line, to get an estimate for alpha (the intercept), which measures how well the industry outperforms CAPM. With the intention of making the ranking mechanism more robust we divided the estimate for alpha by its standard error – giving us the t statistic of including alpha in the model.

The results using this metric continued the theme of under-par returns from portfolios constructed on the basis of the momentum principle. The cumulative returns of both portfolios are supplied in Figures 77 and 88. The LSVG portfolio's returns were overshadowed by that of the risk-free portfolio, while the T10&B10 portfolio's returns barely edged the better earnings in opposition with the risk-free portfolio. In either case, the momentum effect seems to be feeble within our dataset.

Financial Statistics and Stochastic Dominance

We picked the best performing momentum portfolios to compare with the market portfolio, in order to assess the extent of which the market's returns exceeded those of the momentum portfolios. These were the LSVG and T10&B10 portfolios with a formation period of 3 months, and a holding period of 1 month. The financial statistics figures are provided in Figure 92. As can be seen in its arithmetic and geometric mean returns, the market portfolio vastly surpassed the momentum portfolios based on return on investment. However, rather surprisingly, it was the only portfolio to earn significantly less returns that CAPM predictions, demonstrated by a negative Jensen's alpha. Perhaps this was due to the large amount of risk the naïve-weighted market portfolio was exposed to, relative to the momentum portfolios. This is evident in its substantial 1-month VaR and ETL values (at the 95% confidence level) of 8.3% and 12.1% (taking averages of the non-parametric and "modified" estimates). Although, it attained the largest risk-adjusted returns whether you measure this by total risk (Sharpe Ratio = 0.08), systematic risk (Treynor's measure = 0.42), or deviation below the mean return, or some minimum acceptable return (Sortino Ratio approximately = 0.11 whether using long run average risk-free rate of market risk premium). In fact, we plotted the empirical cumulative distribution function of each portfolio's returns (which is displayed in Figure 69), and subsequently graphed the area under the ECDF of the LSVG and the market portfolios minus that of the T10&B10 portfolio (which is shown in Figure 70). This demonstrates that the T10&B10 portfolio second order stochastically dominates the LSVG portfolio, and thus, if past performance was indicative of future earnings, then any investor with a non-decreasing and concave utility function would invest in the unconstrained T10&B10 rather than the category split LSVG portfolio. However, this seems beside the point, which is that our research overwhelmingly instructs investors to invest in the market portfolio rather than any portfolio constructed from monthly industry data with its premier and sole focus to profit from the momentum anomaly – assuming past performance is characteristic of future yields, and that our sample data is representative of the population of industry returns.

Appendix A: Graphs & Figures

Figure 1

The ith row and column refer the industries: Drugs, Soda, Agric, Toys, Steel, Softw, Fun, Oil, Food, and Util respectively for i = 1, 2, . . ., 10.

$$V = \begin{bmatrix} 24.58 & 11.03 & 1.91 & 3.86 & 11.04 & 17.95 & 11.19 & 5.27 & 8.85 & 6.44 \\ 11.03 & 72.50 & 2.35 & 13.16 & 14.16 & 6.47 & 15.29 & 9.40 & 14.81 & 10.42 \\ 1.91 & 2.35 & 34.22 & 12.90 & 13.32 & 13.70 & 9.21 & 9.89 & 5.14 & 4.79 \\ 3.86 & 13.16 & 12.90 & 39.88 & 28.35 & 19.40 & 22.42 & 12.65 & 7.82 & 2.90 \\ 11.04 & 14.16 & 13.32 & 28.35 & 81.18 & 53.23 & 37.82 & 26.50 & 4.40 & 8.56 \\ 17.95 & 6.47 & 13.70 & 19.40 & 53.23 & 90.07 & 38.09 & 13.64 & 1.68 & -0.74 \\ 11.19 & 15.29 & 9.21 & 22.42 & 37.82 & 38.09 & 45.74 & 10.78 & 7.62 & 4.93 \\ 5.27 & 9.40 & 9.89 & 12.65 & 26.50 & 13.64 & 10.78 & 32.57 & 7.47 & 13.54 \\ 8.85 & 14.81 & 5.14 & 7.82 & 4.40 & 1.68 & 7.62 & 7.47 & 18.07 & 7.92 \\ 6.44 & 10.42 & 4.79 & 2.90 & 8.56 & -0.74 & 4.93 & 13.54 & 7.92 & 20.34 \end{bmatrix}$$

Figure 2

Figures for each industry's beta at the end of the 120-month in-sample:

Drugs	Soda	Agric	Toys	Steel	Softw	Fun	Oil	Food	Util
0.4973	0.1818	0.2783	0.3550	0.4113	0.4110	0.5042	0.3626	0.3735	0.2598

Figure 3

Figures for each industry's expected return for the 1st month of the out-of-sample data:

Drugs	Soda	Agric	Toys	Steel	Softw	Fun	Oil	Food	Util
0.7270	0.525 9	0.587 5	0.636 3	0.672 2	0.672 0	0.731 4	0.6412	0.648 1	0.5757

Figure 4

$$w_{gmvp} = \begin{bmatrix} 0.197815 \\ -0.00777 \\ 0.149869 \\ 0.116198 \\ -0.0613 \\ 0.042157 \\ 0.020696 \\ 0.037762 \\ 0.222337 \\ 0.282231 \end{bmatrix} \qquad w_{tangent} = \begin{bmatrix} 0.066388 \\ 0.151628 \\ 0.062007 \\ -0.01145 \\ 0.097796 \\ 0.133833 \\ 0.193857 \\ 0.000811 \\ 0.043652 \\ 0.261475 \end{bmatrix}$$

Figure 5

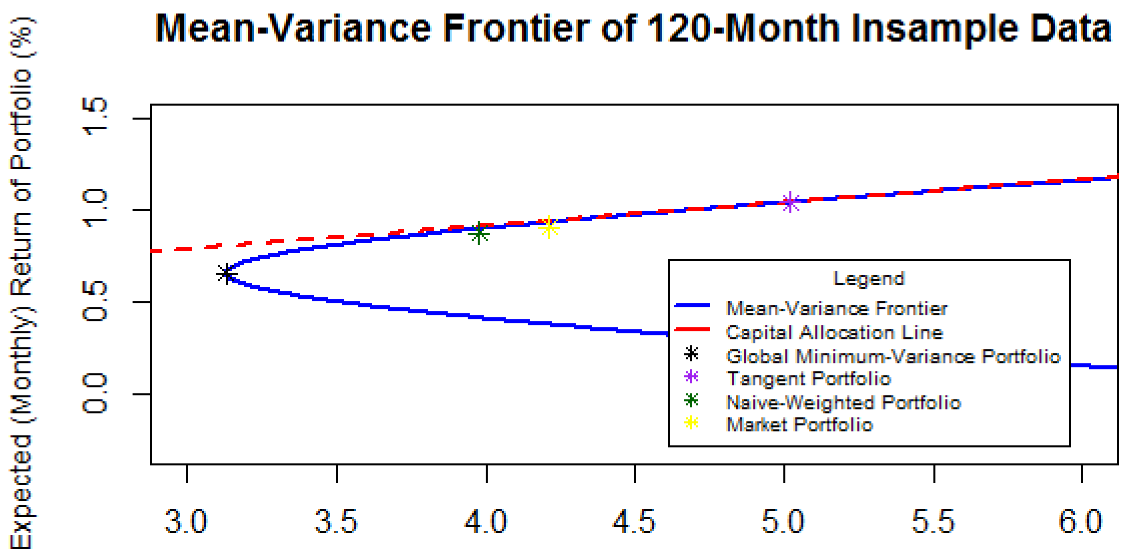

Figure 6

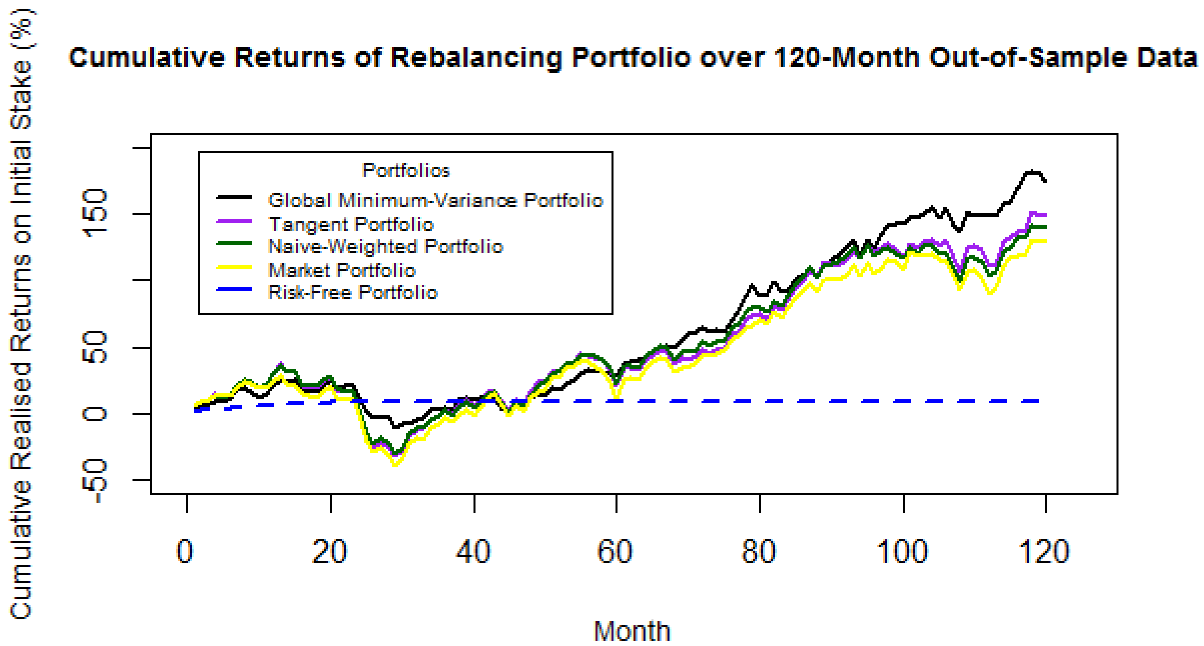

Figure 7

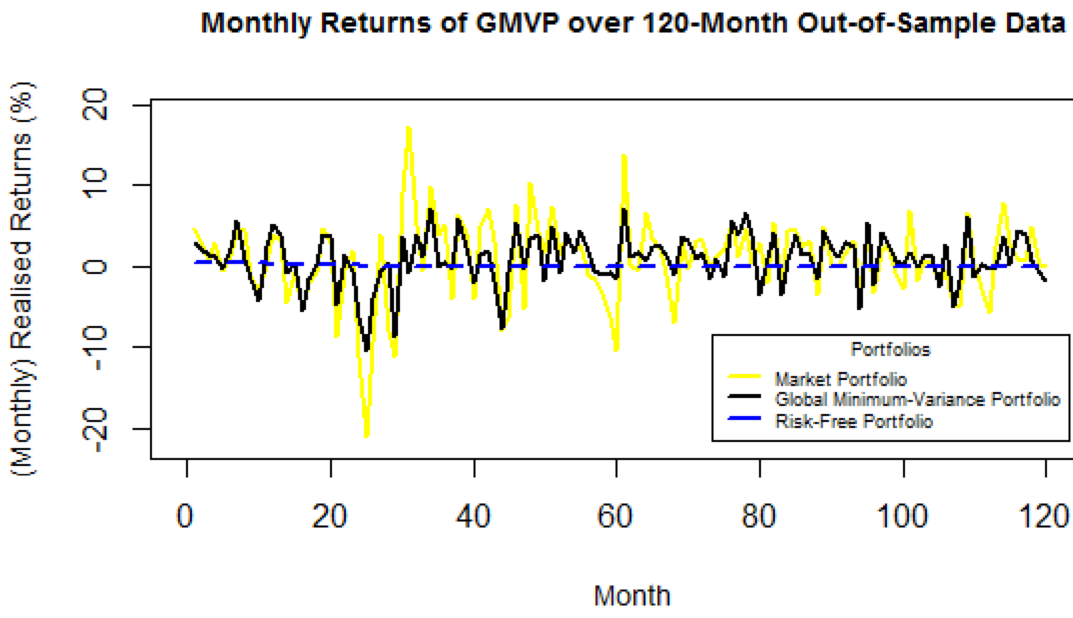

Figure 8

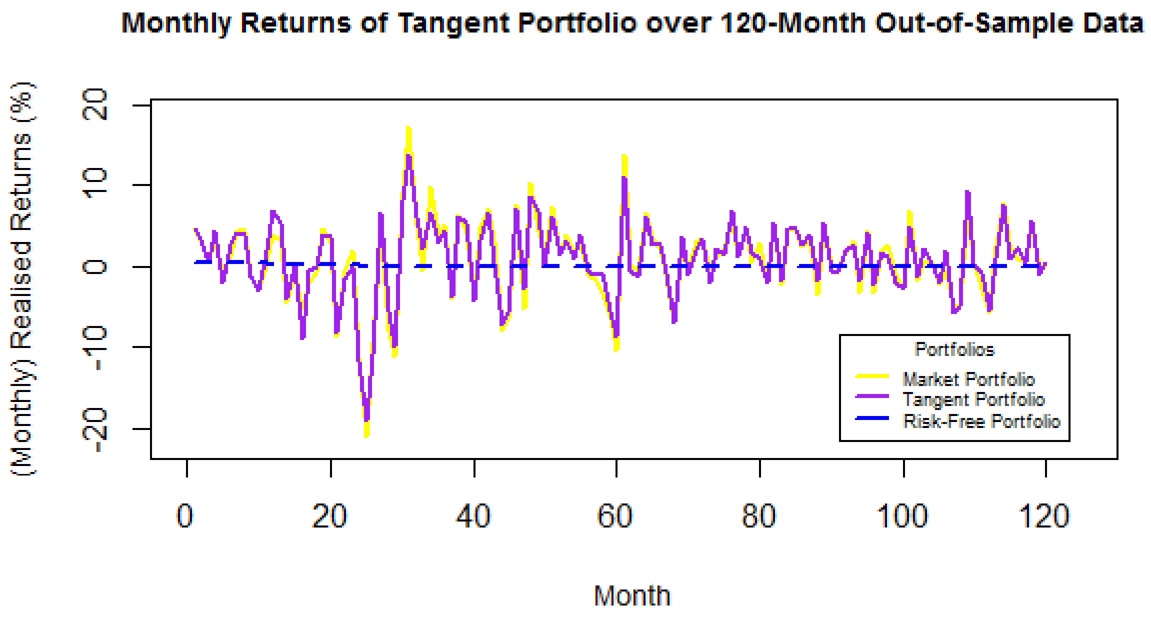

Figure 9

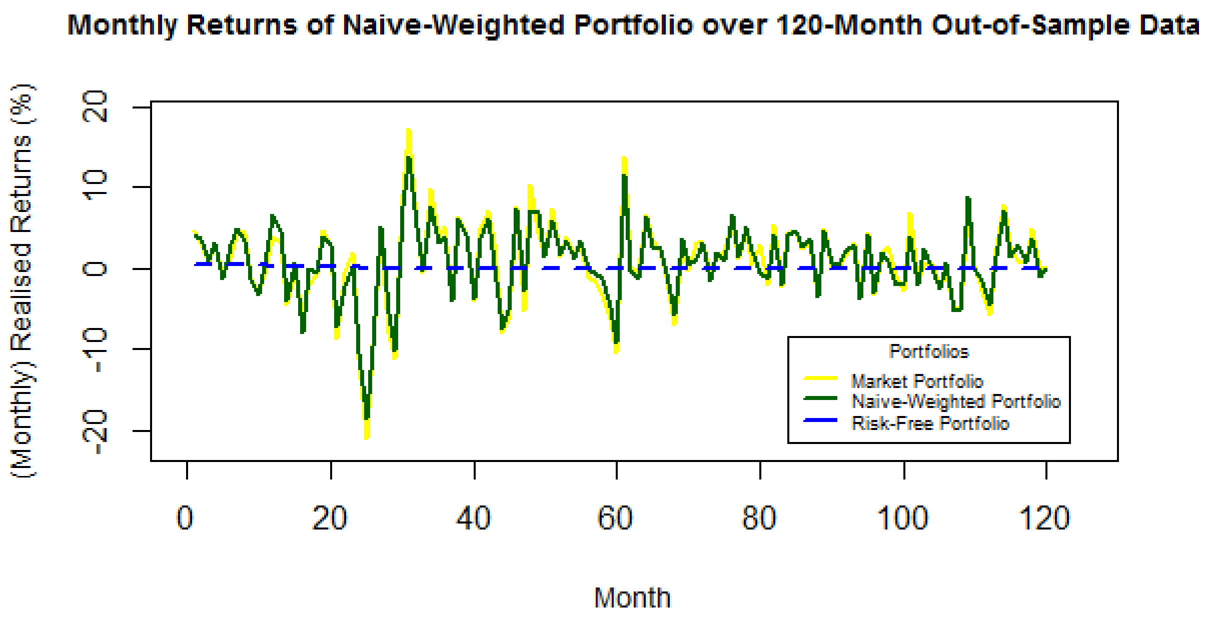

Figure 10

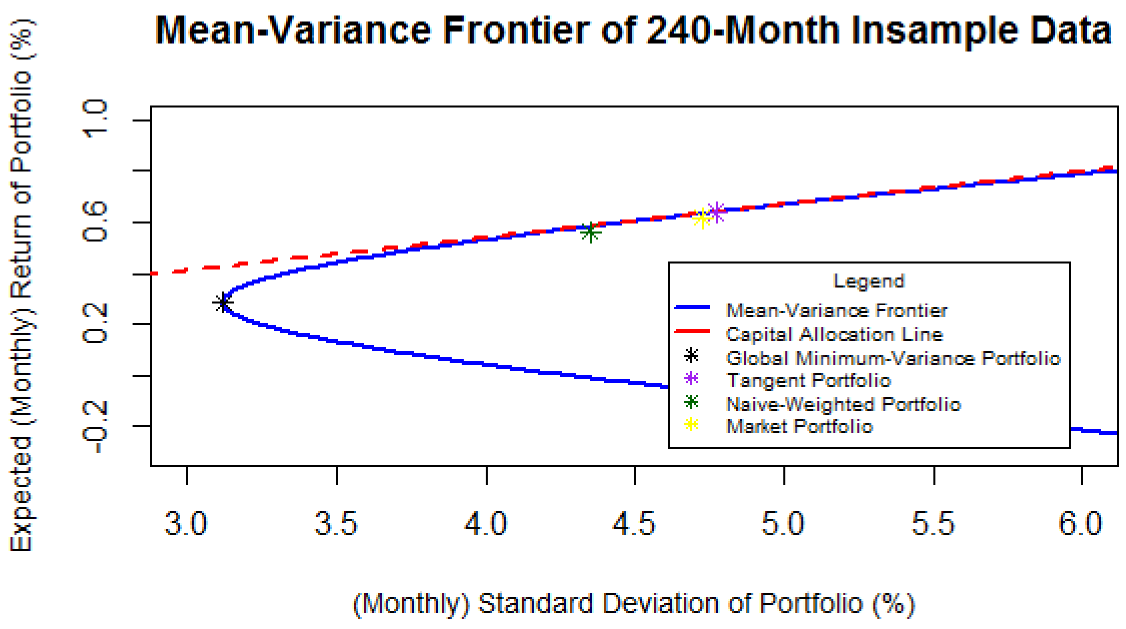

Figure 11`

	CAPM			
	Alpha Estimate	Standard Error	T-Test Statistic	P-value
GMVP	0.52915	0.18342	2.885	0.00466
TP	0.20123	0.12375	1.626	0.107
Naïve	0.18678	0.121	1.544	0.125
Market	0.09618	0.0996	0.966	0.336
	Beta Estimate	Standard Error	T-Test Statistic	P-value
GMVP	0.558	0.04036	13.824	< 10^(-15)
TP	1.04856	0.02723	38.504	< 10^(-15)
Naïve	1.00111	0.02663	37.595	< 10^(-15)
Market	1.12721	0.02192	51.426	< 10^(-15)
	F-Test Statistic	P-value	Adjusted R-Squared	
GMVP	191.1	< 10^(-15)	0.615	
TP	1483	< 10^(-15)	0.9257	
Naïve	1413	< 10^(-15)	0.9223	
Market	2645	< 10^(-15)	0.9569	

Figure 12

	Fama-French 3 Factor Model			
	Alpha Estimate	Standard Error	T-Test Statistic	P-value
GMVP	0.47315	0.17016	2.781	0.006334
TP	0.1443	0.11793	1.224	0.223561
Naïve	0.15367	0.12038	1.277	0.204
Market	0.10799	0.09121	1.184	0.239
	Beta Estimate	Standard Error	T-Test Statistic	P-value
GMVP	0.64808	0.04161	15.575	< 10^(-15)
TP	1.07151	0.02884	37.158	< 10^(-15)
Naïve	1.0176	0.02944	34.569	< 10^(-15)
Market	1.07954	0.0223	48.403	< 10^(-15)
	s (SMB) Estimate	Standard Error	T-Test Statistic	P-value
GMVP	-0.28944	0.08212	-3.525	0.000608
TP	0.07017	0.05691	1.233	0.220095
Naïve	0.02413	0.0581	0.415	0.679
Market	0.21327	0.04402	4.845	0.00000395
	h (HML) Estimate	Standard Error	T-Test Statistic	P-value
GMVP	-0.21087	0.06799	-3.101	0.002419
TP	-0.18146	0.04712	-3.851	0.000193
Naïve	-0.10703	0.0481	-2.225	0.028
Market	0.05825	0.03644	1.598	0.113
	F-Test Statistic	P-value	Adjusted R-Squared	
GMVP	82.87	< 10^(-15)	0.6736	
TP	557.8	< 10^(-15)	0.9335	
Naïve	485	< 10^(-15)	0.9242	
Market	1076	< 10^(-15)	0.9644	

Figure 13

	Cahart 4 Factor Model			
	Alpha Estimate	Standard Error	T-Test Statistic	P-value
GMVP	0.46566	0.16276	2.861	0.005018
TP	0.14813	0.11542	1.283	0.2019
Naïve	0.15737	0.11813	1.332	0.18546
Market	0.11372	0.08246	1.379	0.171
	Beta Estimate	Standard Error	T-Test Statistic	P-value
GMVP	0.68284	0.04106	16.629	< 10^(-15)
TP	1.05377	0.02912	36.19	< 10^(-15)
Naïve	1.00042	0.0298	33.567	< 10^(-15)
Market	1.05294	0.0208	50.613	< 10^(-15)
	s (SMB) Estimate	Standard Error	T-Test Statistic	P-value
GMVP	-0.29345	0.07855	-3.736	0.000293
TP	0.07221	0.0557	1.296	0.1974
Naïve	0.02611	0.05701	0.458	0.64783
Market	0.21633	0.0398	5.436	0.00000031
	h (HML) Estimate	Standard Error	T-Test Statistic	P-value
GMVP	-0.11883	0.07033	-1.69	0.093798
TP	-0.22844	0.04987	-4.581	0.0000118
Naïve	-0.15251	0.05104	-2.988	0.00344
Market	-0.01217	0.03563	-0.342	0.733
	m (MOM) Estimate	Standard Error	T-Test Statistic	P-value
GMVP	0.12695	0.03694	3.437	0.000821
TP	-0.06481	0.02619	-2.474	0.0148
Naïve	-0.06272	0.02681	-2.339	0.02104
Market	-0.09714	0.01872	-5.19	0.000000915
	F-Test Statistic	P-value	Adjusted R-Squared	
GMVP	70.9	< 10^(-15)	0.7014	
TP	438.3	< 10^(-15)	0.9363	
Naïve	379.1	< 10^(-15)	0.9271	
Market	994.4	< 10^(-15)	0.9709	

Figure 14

Note: All figures are corresponding to monthly percentage returns unless otherwise stated.

Financial Ratio	GMVP	Tangent	Naive	Market
Yearly Arithmetic Mean	11.20669	11.05526	10.46134	10.3398
Yearly Geometric Mean	10.52931	9.446315	8.999625	8.55684
Monthly Arithmetic Mean	0.889099	0.8776431	0.832575	0.8233248
Monthly Geometric Mean	0.8377438	0.7550359	0.7207035	0.6865436
Standard Deviation	3.205486	4.921195	4.706972	5.203945
Skewness	-0.7160866	-0.7539904	-0.6967746	-0.5794711
Kurtosis	1.009339	1.75371	1.910514	2.45112
Covariance with the Market	12.2488	24.74138	23.78609	27.08104
Pearson Correlation with the Market	0.734289	0.9660969	0.971066	1
Spearman Correlation with the Market	0.706306	0.9575873	0.9620113	1
Beta	0.5134138	0.9842259	0.9426375	-
Alpha from CAPM with intercept	0.5291456	0.2012346	0.1867754	0.09617862
Standard Error of Alpha	0.1834181	0.1237473	0.1210047	0.099604

P-value of Alpha	0.004655766	0.1065804	0.1253779	0.3362142
1 month 95% CL VaR (Non-Parametric)	4.979064	8.327416	7.5757	8.010867
1 month 95% CL VaR (“Modified”)	4.915437	8.007778	7.582297	8.265246
1 month 95% CL ETL (Non-Parametric)	7.280846	11.50956	10.90833	11.93544
1 month 95% CL ETL (“Modified”)	7.141542	12.38509	11.98636	13.73124
Semi-Standard Deviation	2.43978	3.758468	3.581241	3.918679
Downside Deviation (MAR = long run risk-free rate)	2.20549	3.529102	3.372968	3.715851
Downside Deviation (MAR = long run market risk premium)	2.304561	3.629342	3.473772	3.816268
Jensen’s Alpha	0.1758172	-0.123734	-0.1433537	-0.1877047
Sharpe Ratio	0.152857	0.09723754	0.09208826	0.08151637
Treynor’s Measure	0.9543588	0.4861942	0.4598341	0.4242067
Sortino’s Ratio (MAR = long run risk-free rate)	0.2221642	0.1355939	0.128509	0.1141614
Sortino’s Ratio (MAR = long run market risk premium)	0.2126136	0.131849	0.1247799	0.1111575
Appraisal Ratio	0.08871016	-0.09258018	-0.1096935	-

Figure 15

x refers to the monthly rate of return

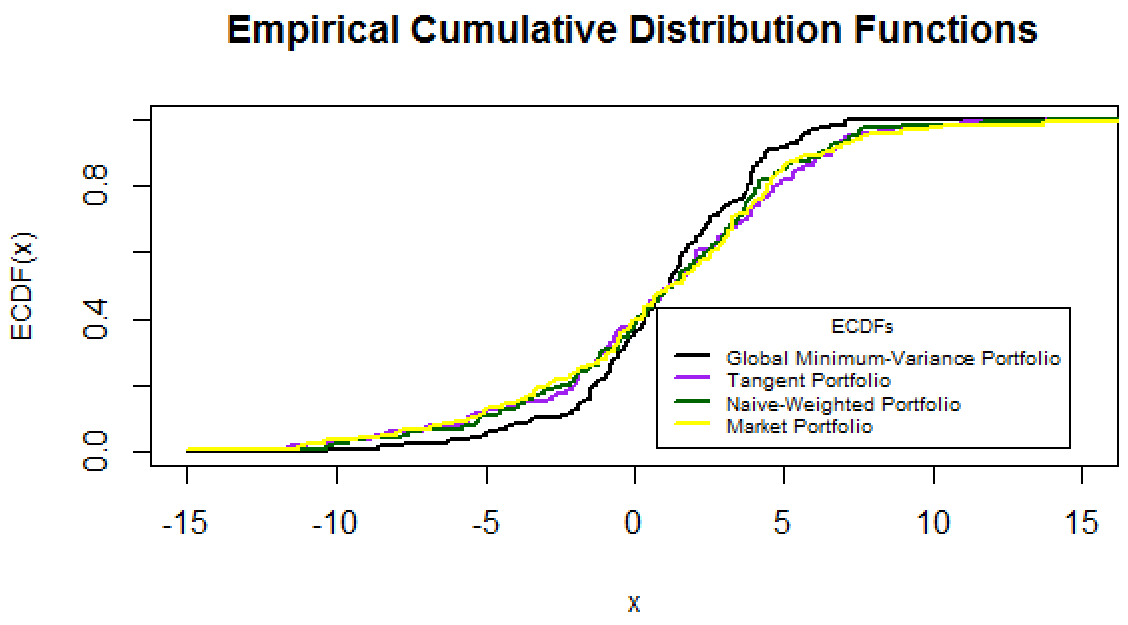

Figure 16

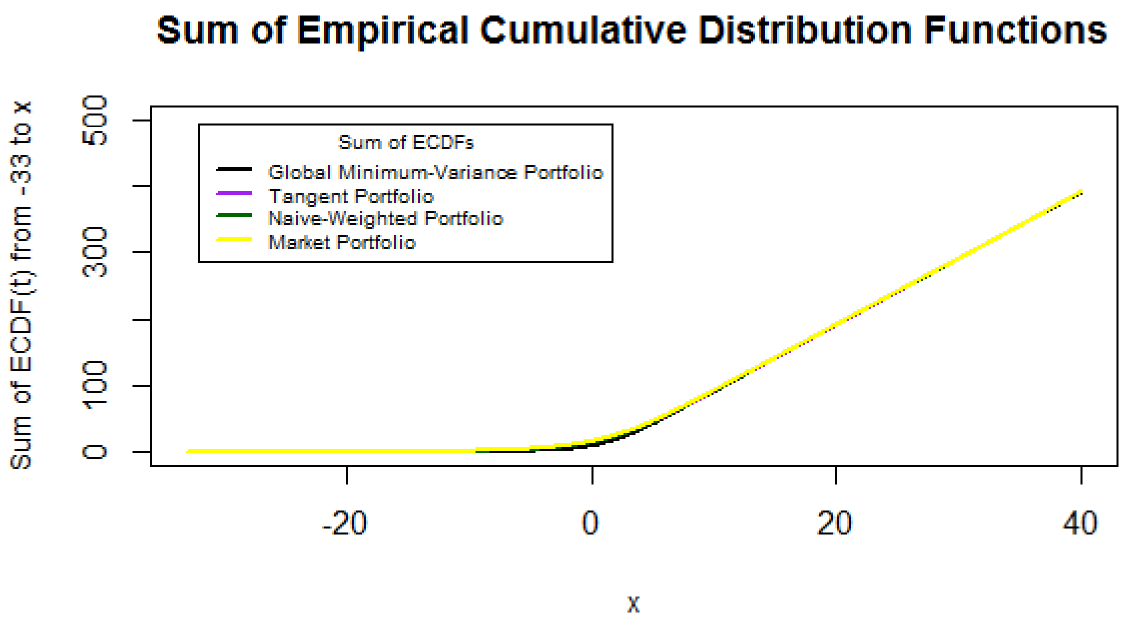

Figure 17

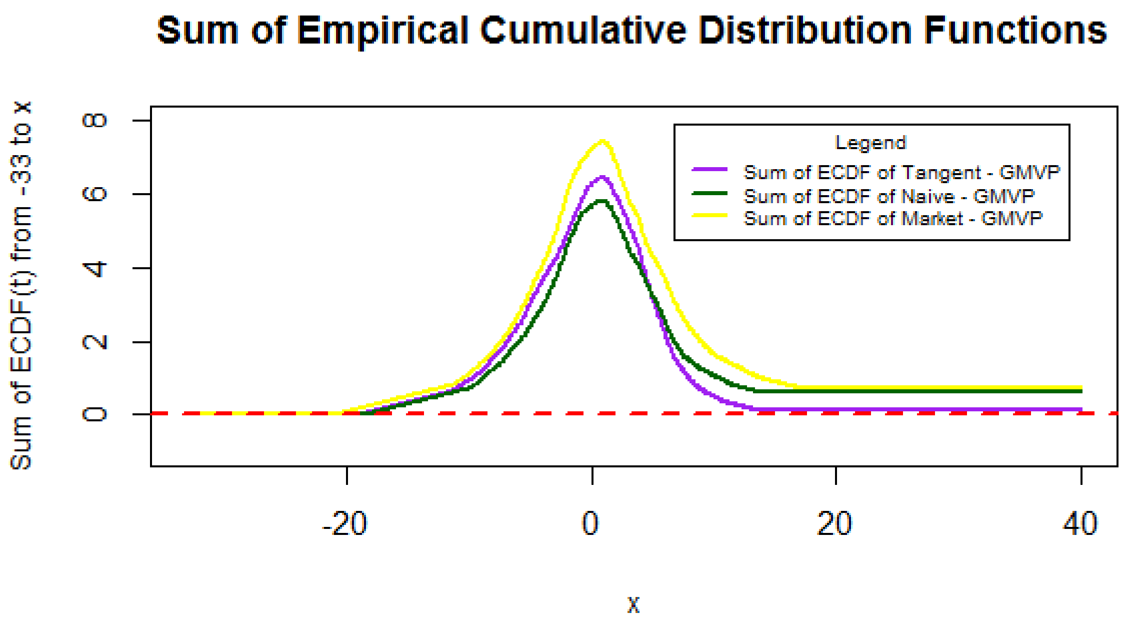

Figure 18

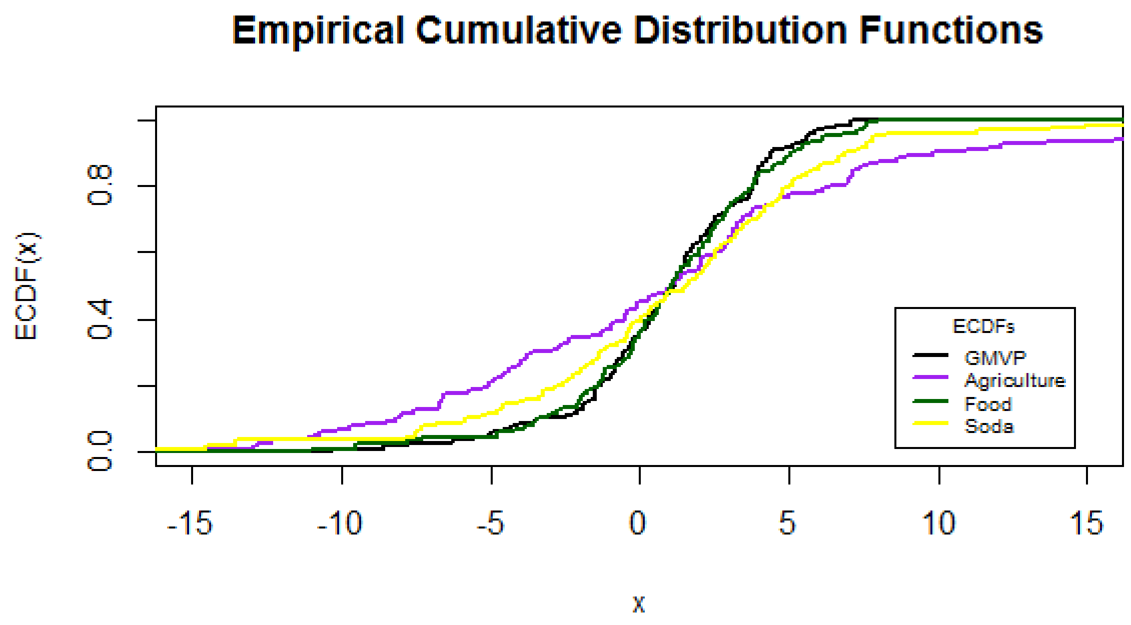

Figure 19

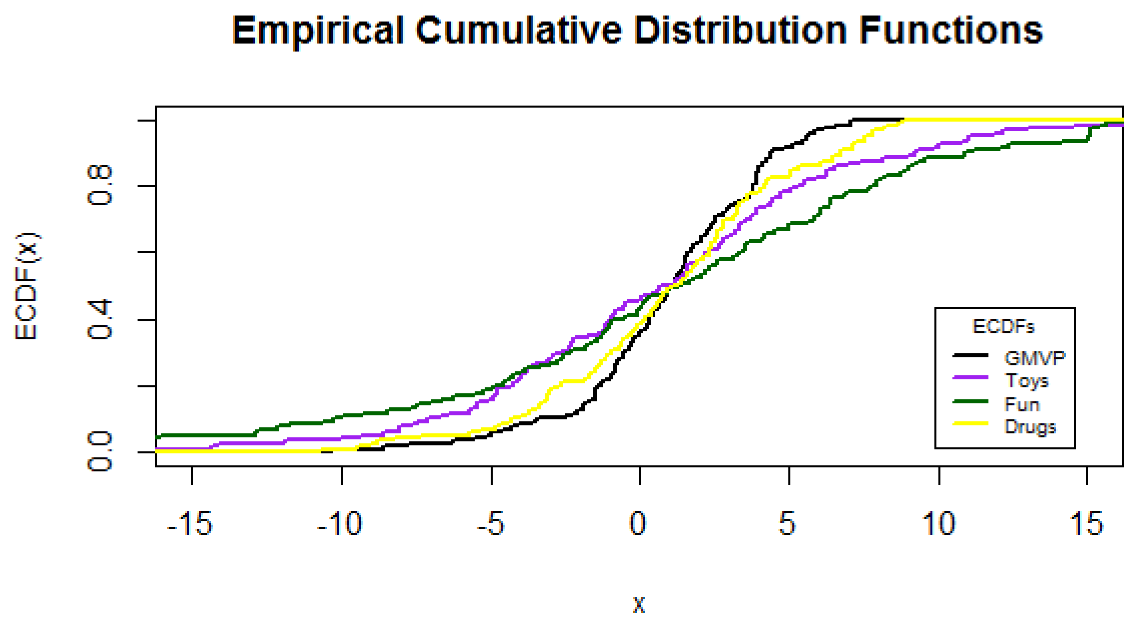

Figure 20

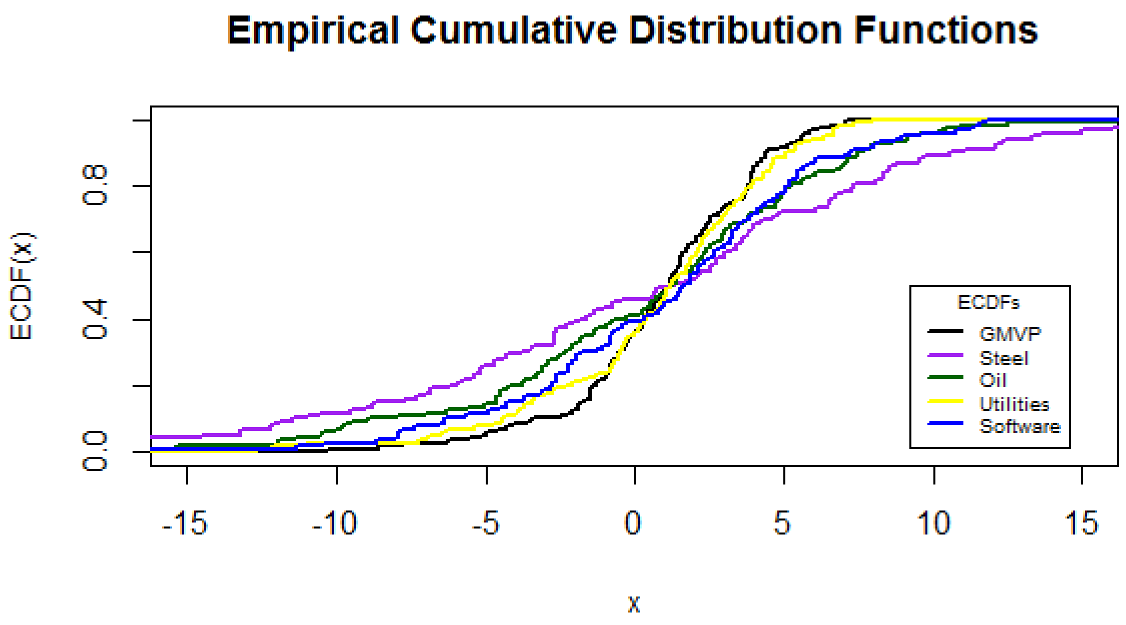

Figure 21

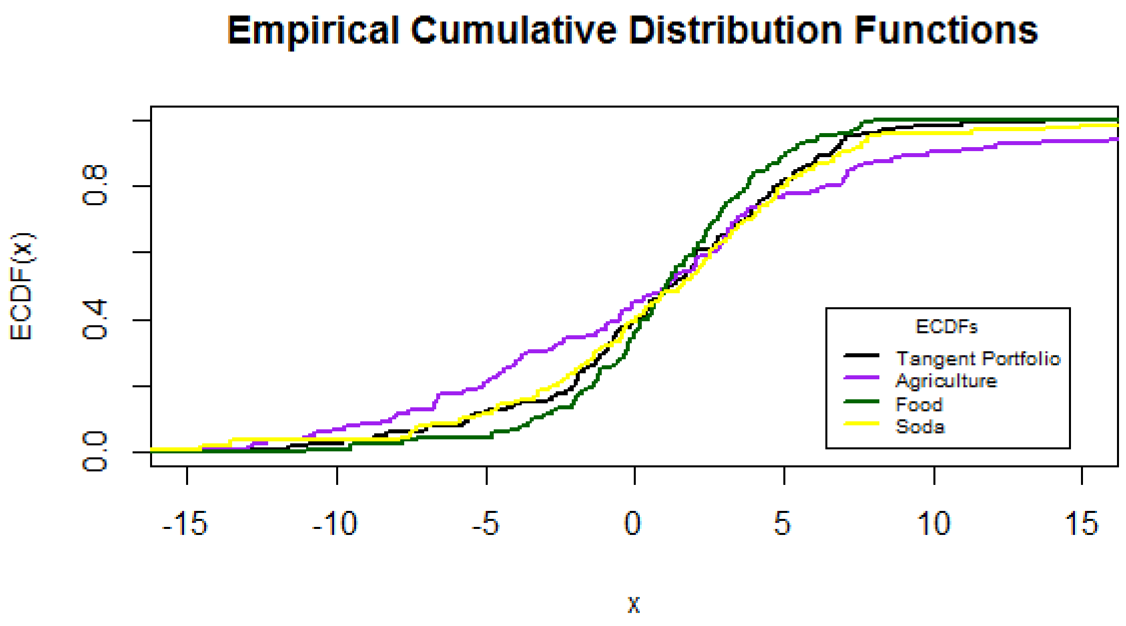

Figure 22

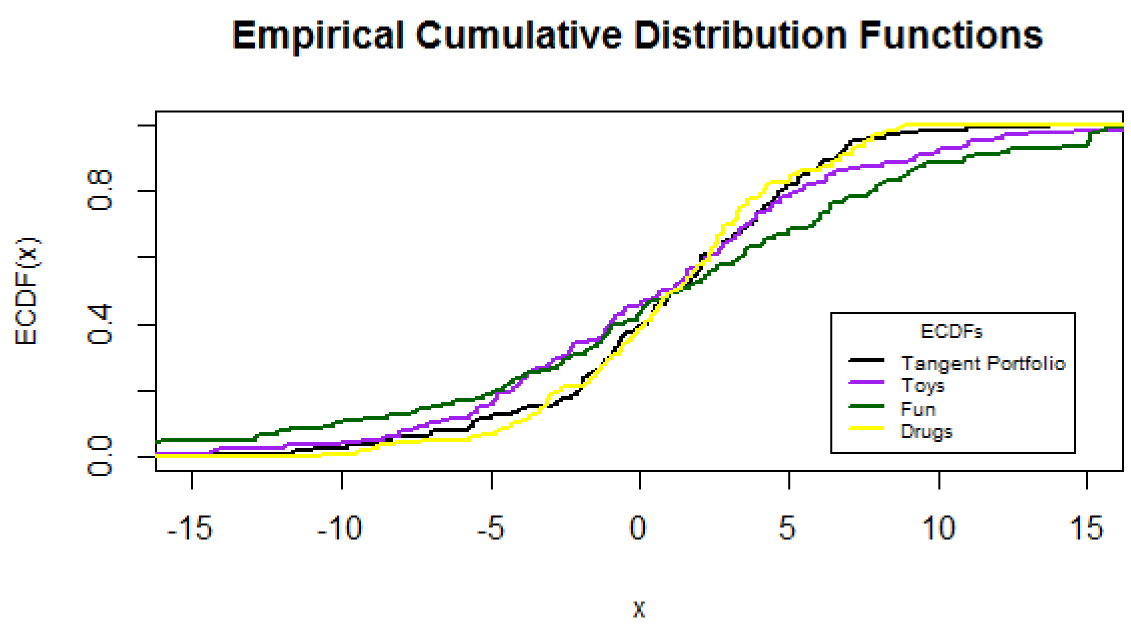

Figure 23

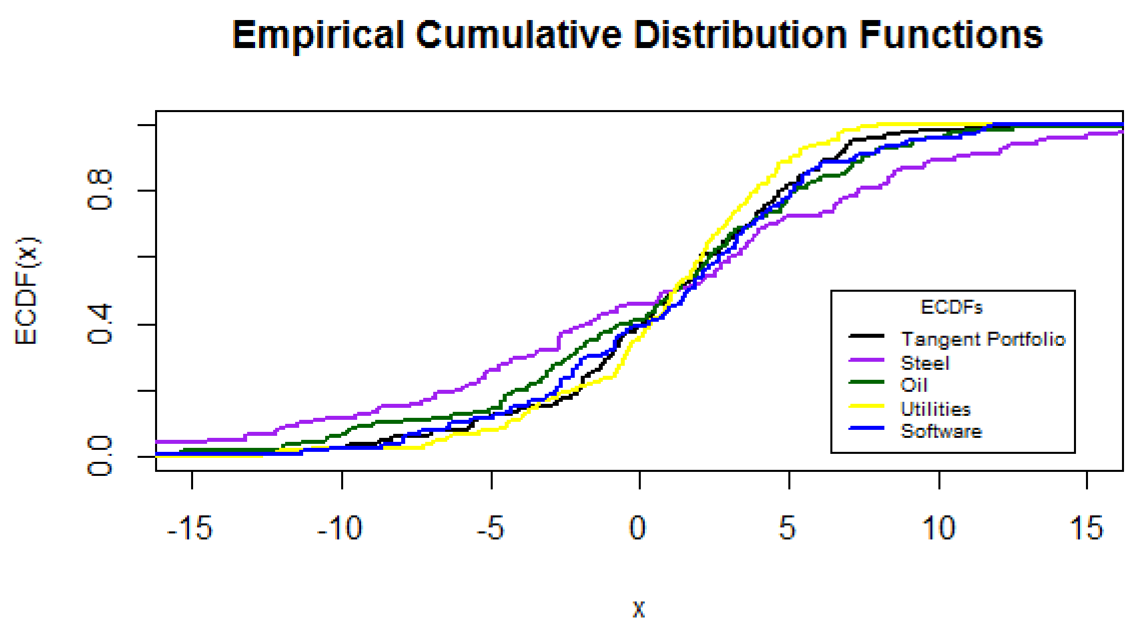

Figure 24

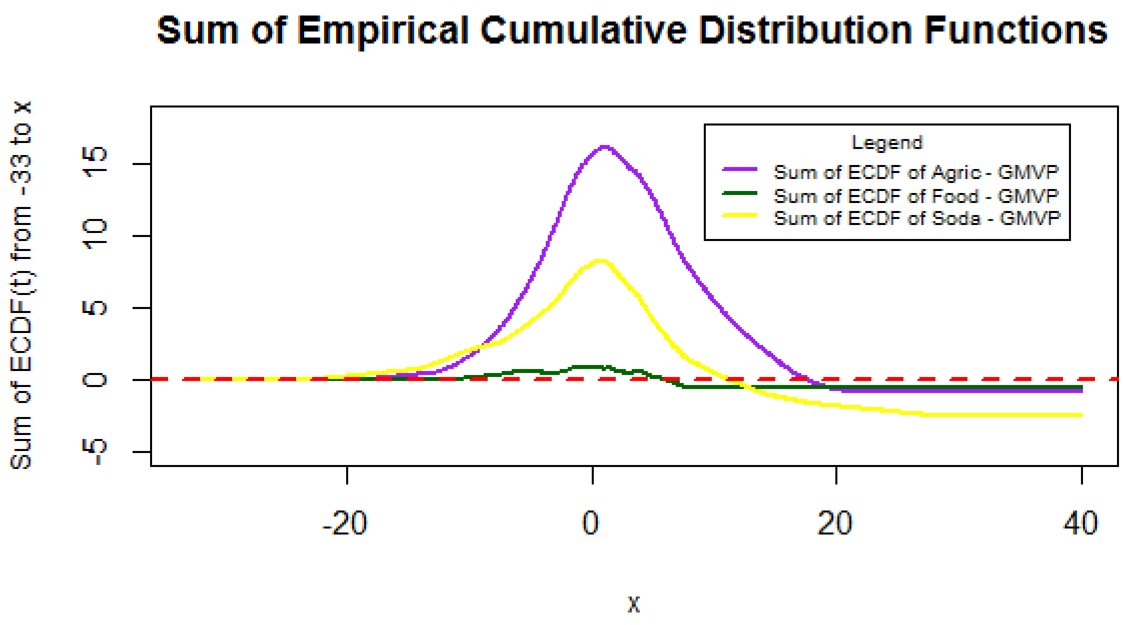

Figure 25

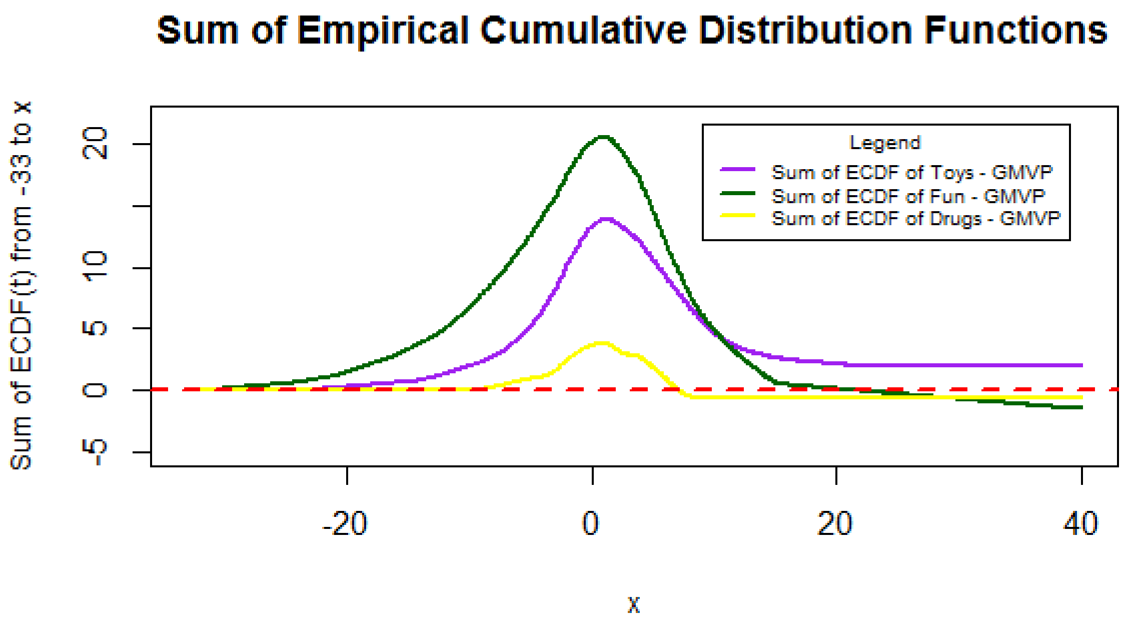

Figure 26

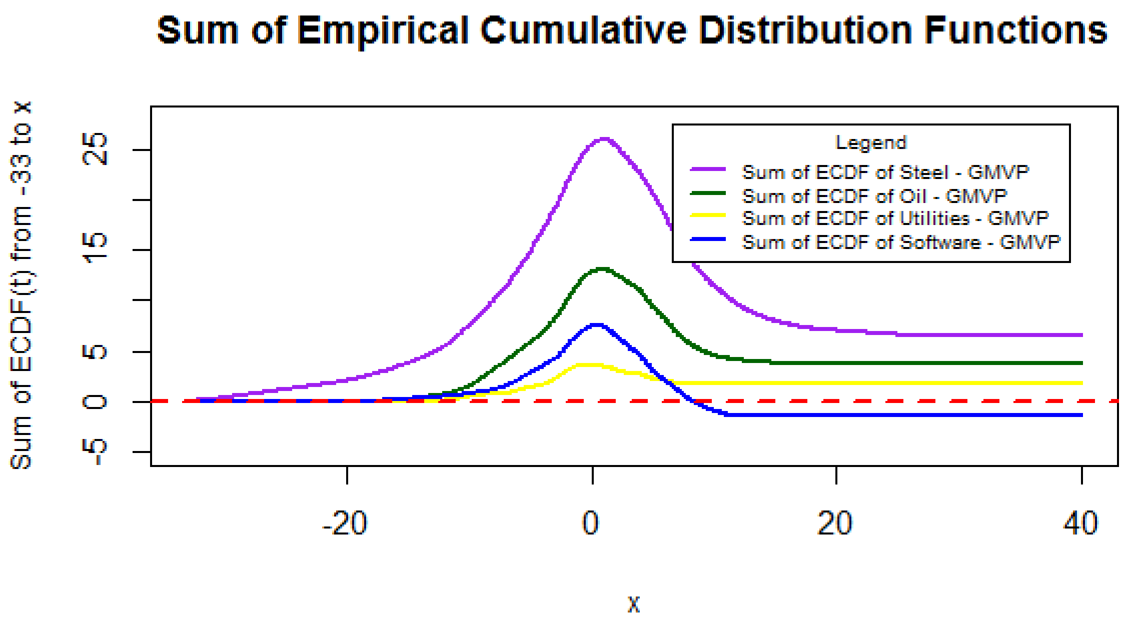

Figure 27

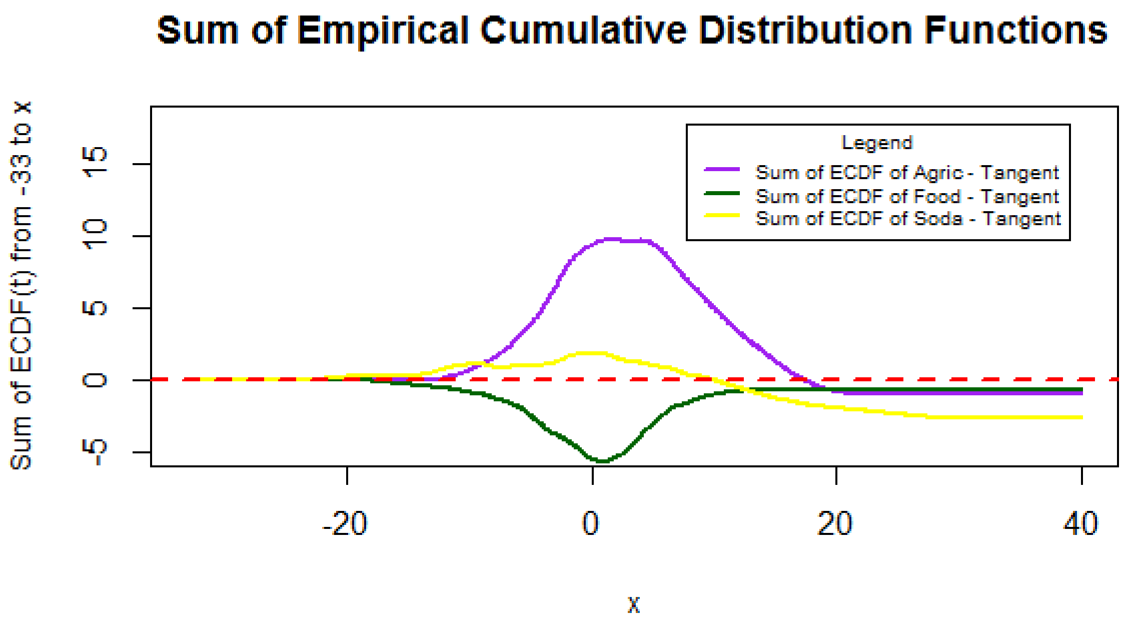

Figure 28

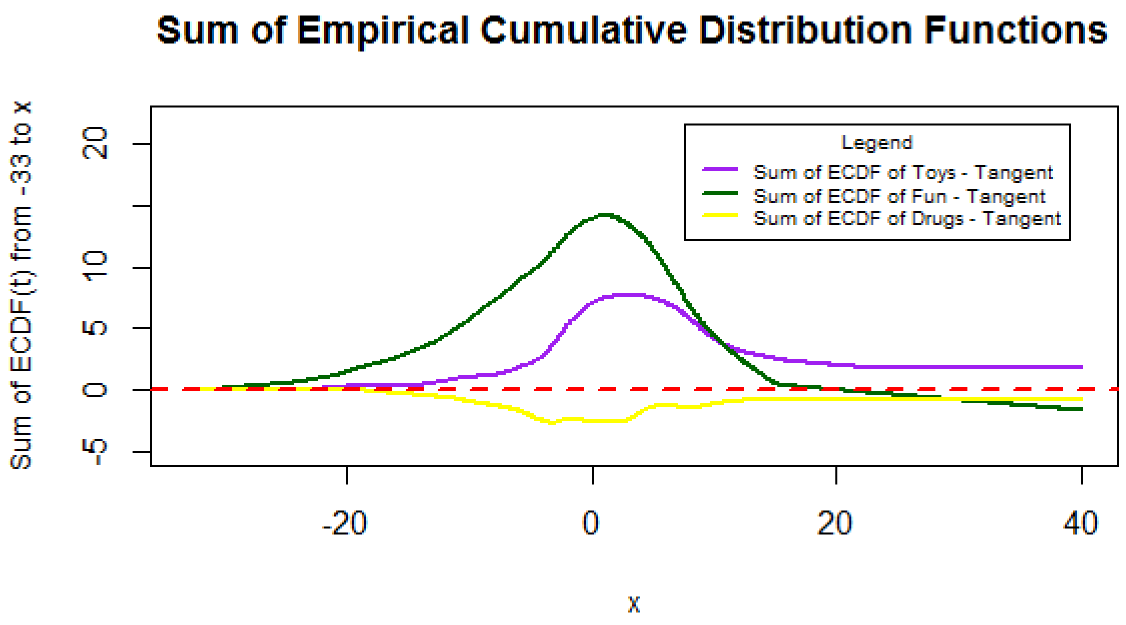

Figure 29

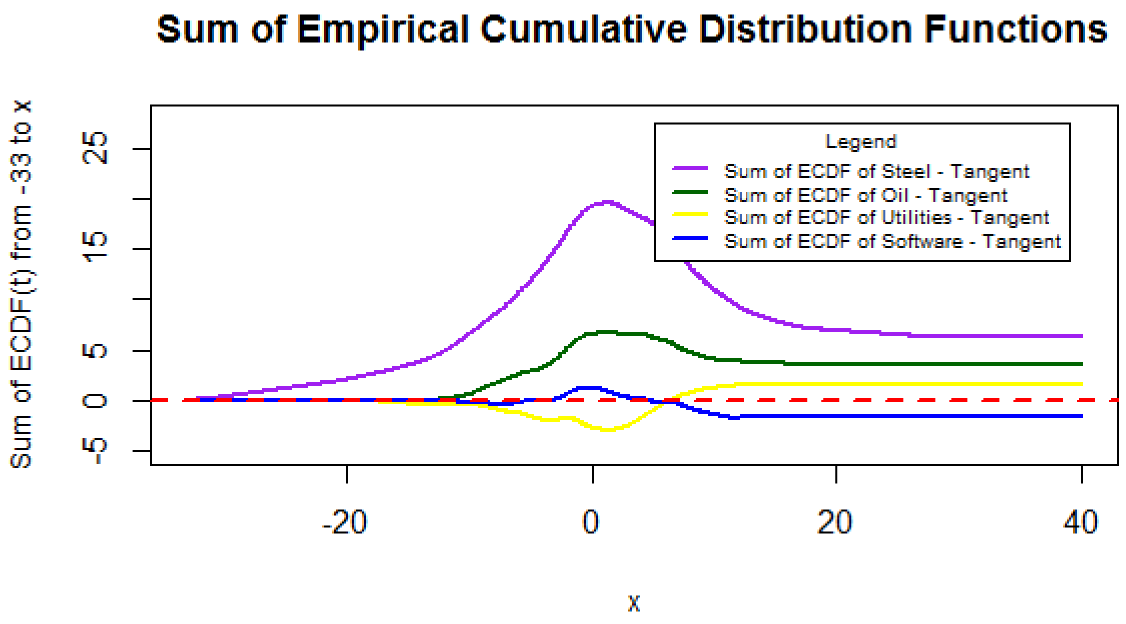

Figure 30

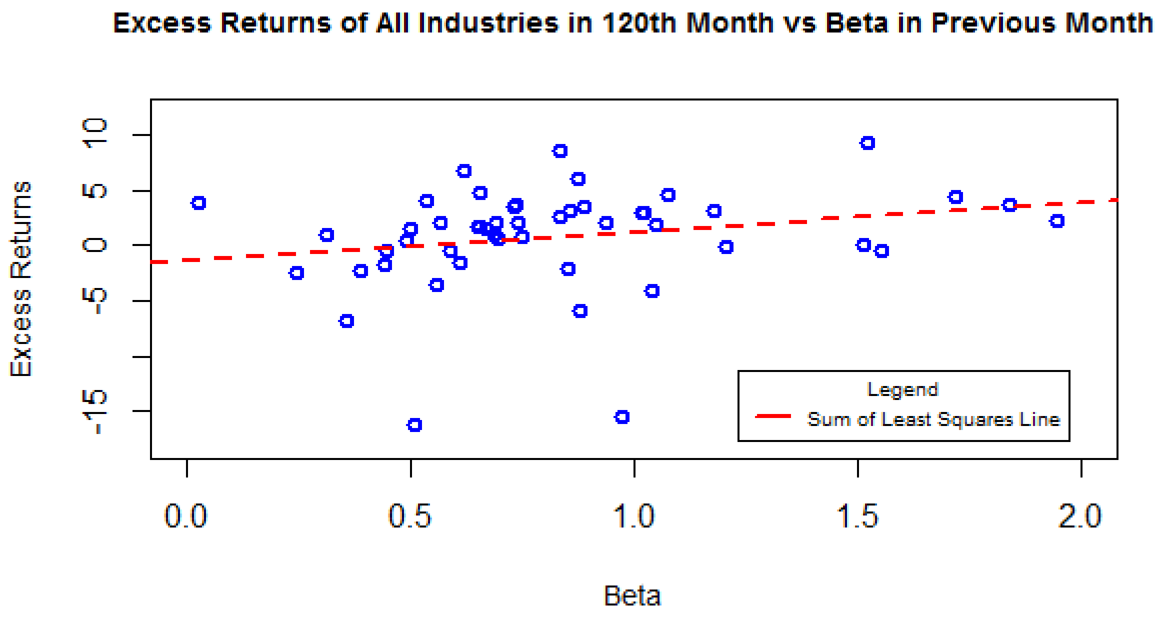

Figure 31

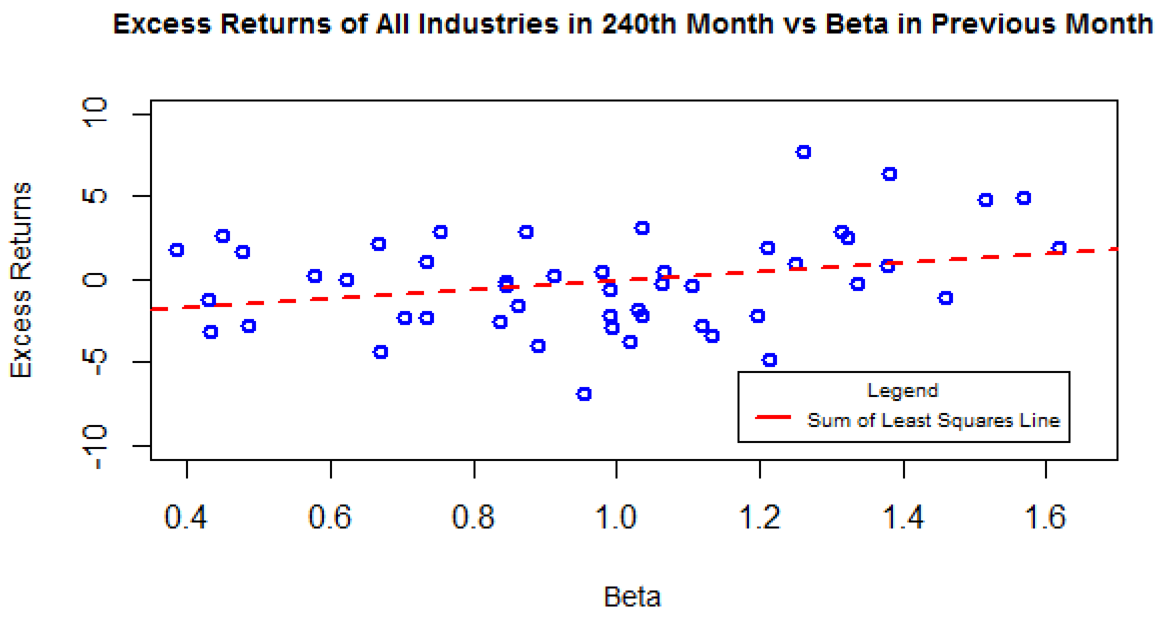

Figure 32

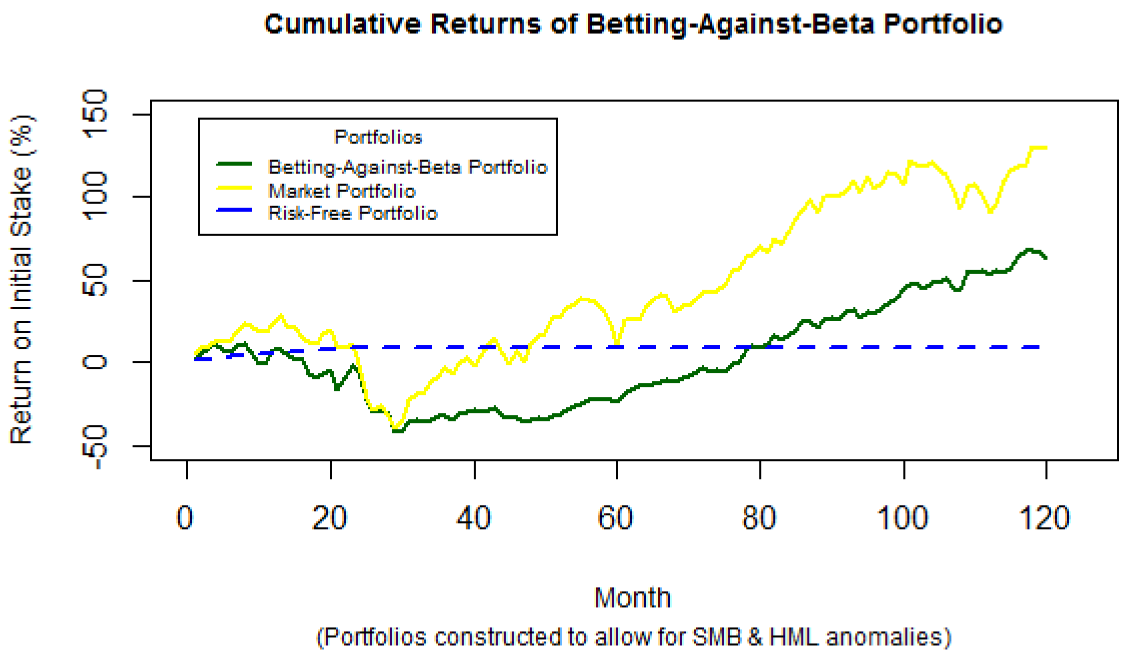

Figure 33

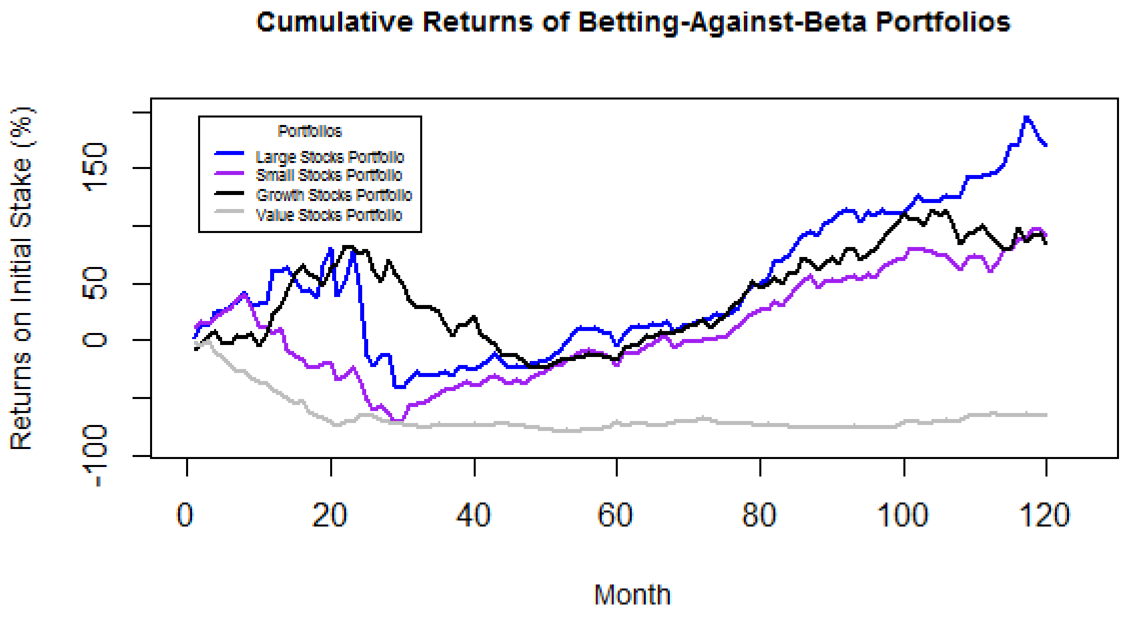

Figure 34

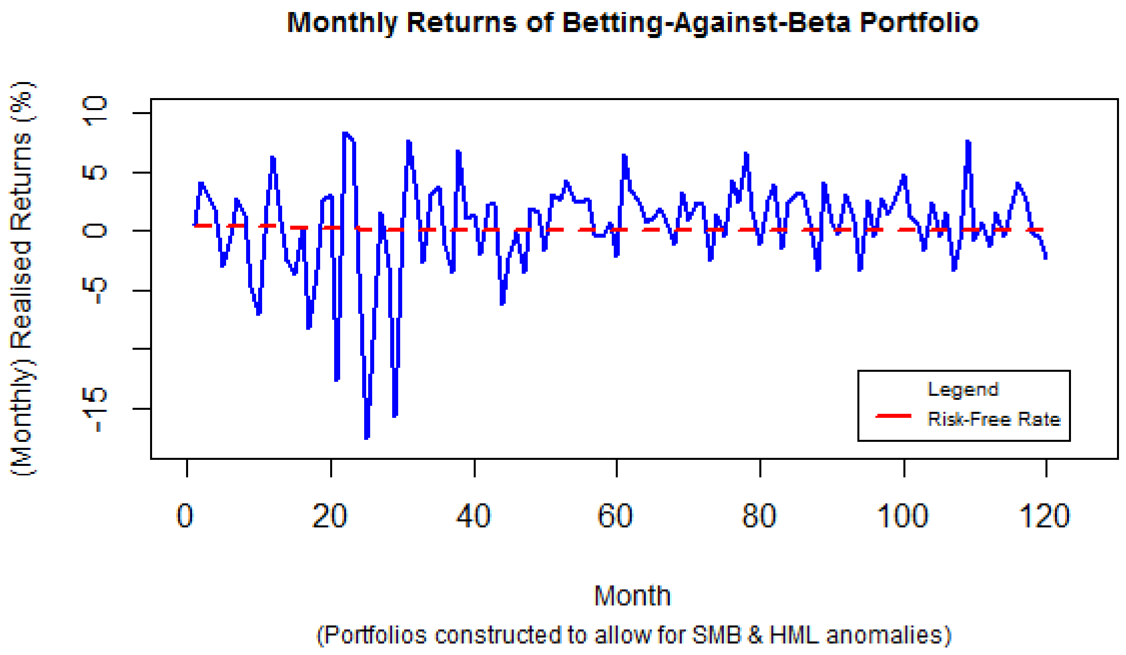

Figure 35

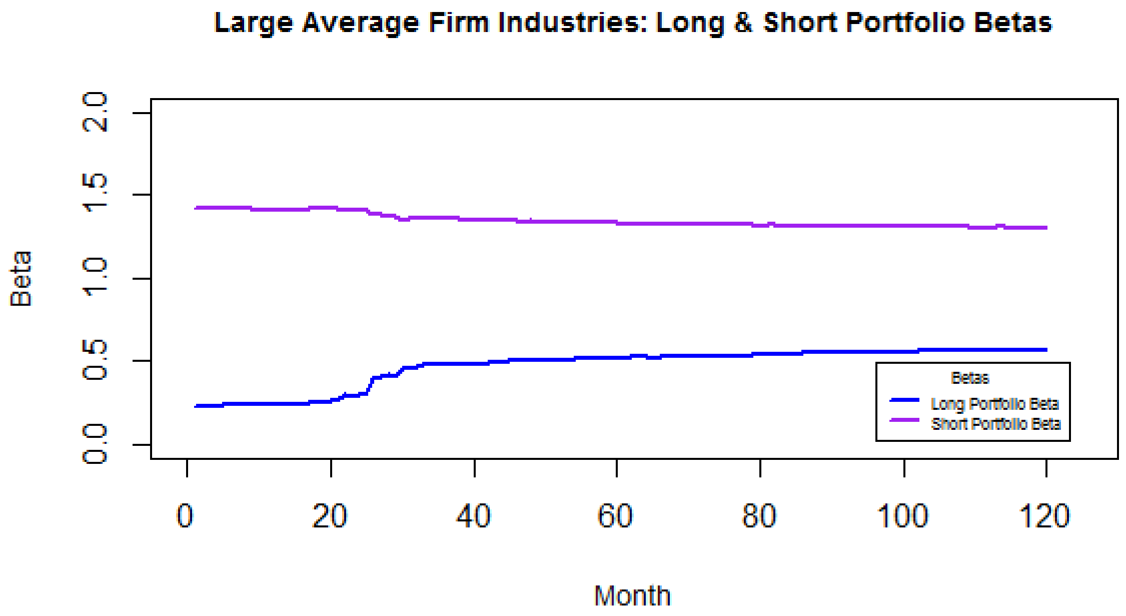

Figure 36

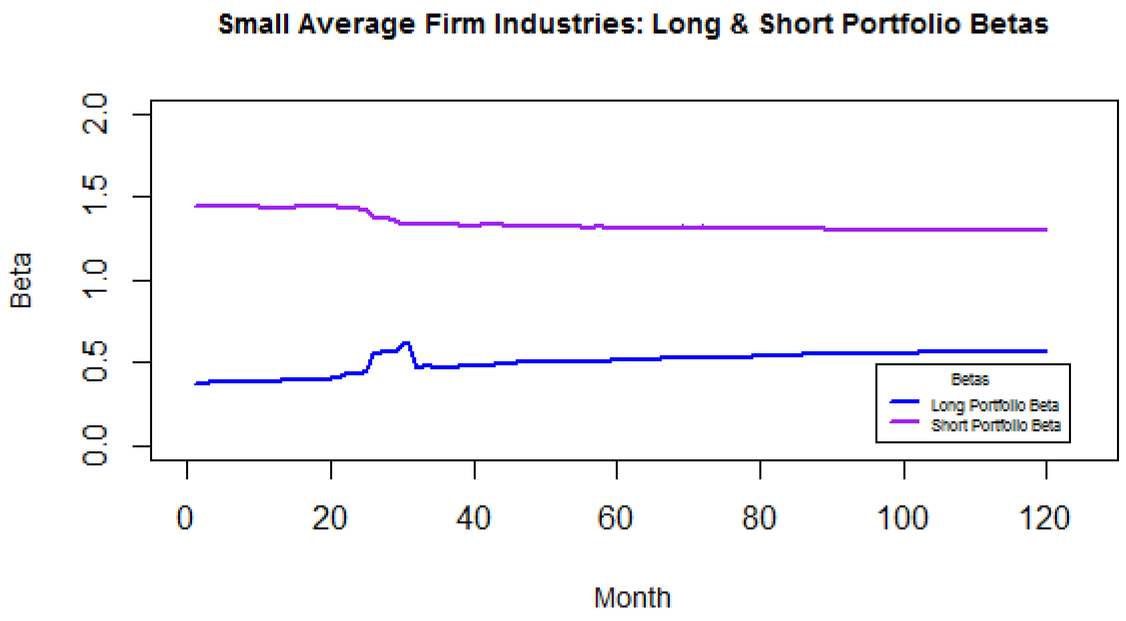

Figure 37

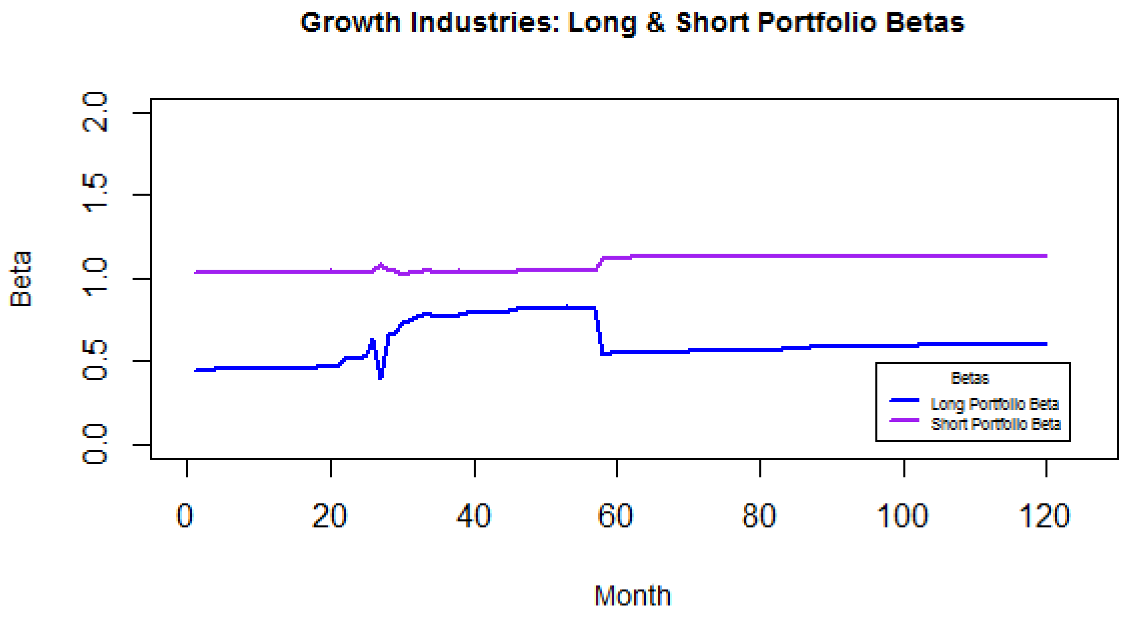

Figure 38

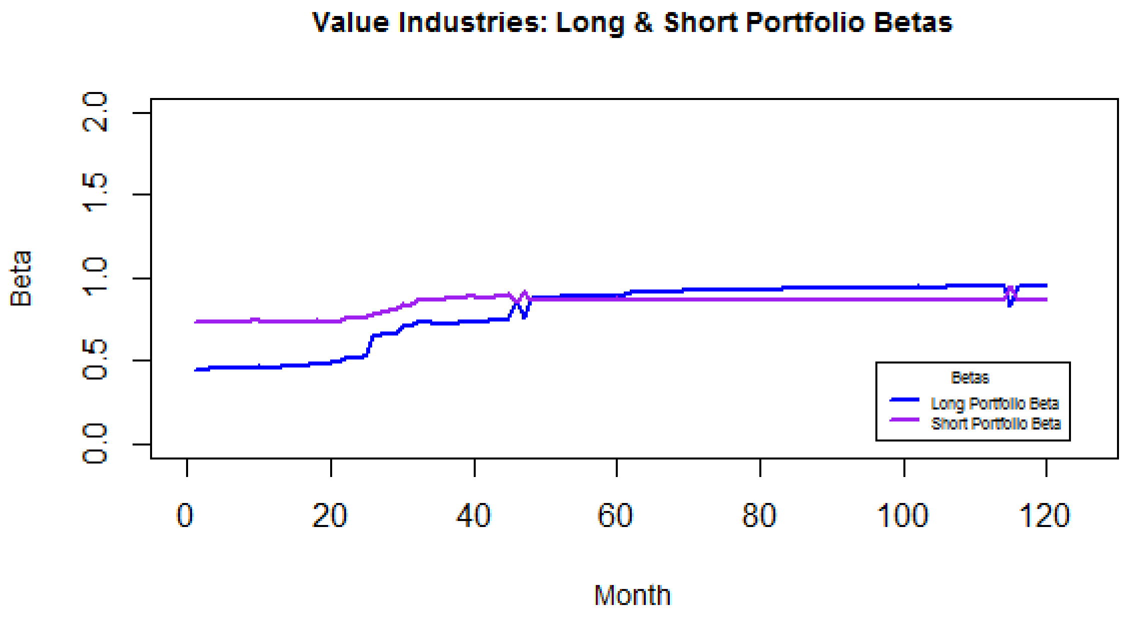

Figure 39

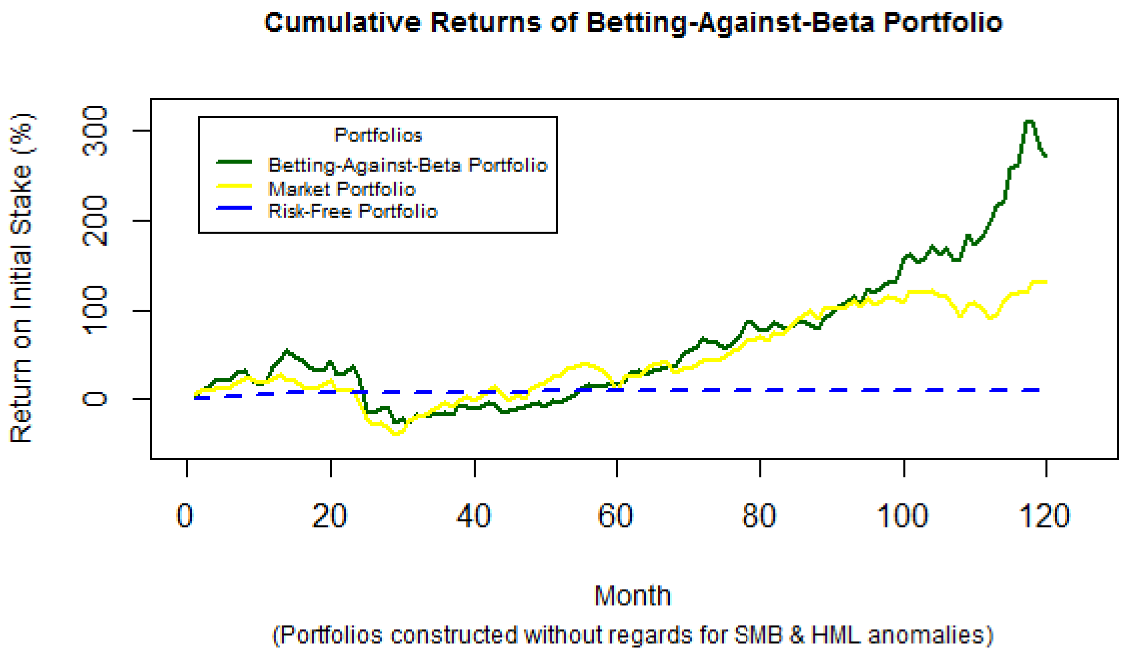

Figure 40

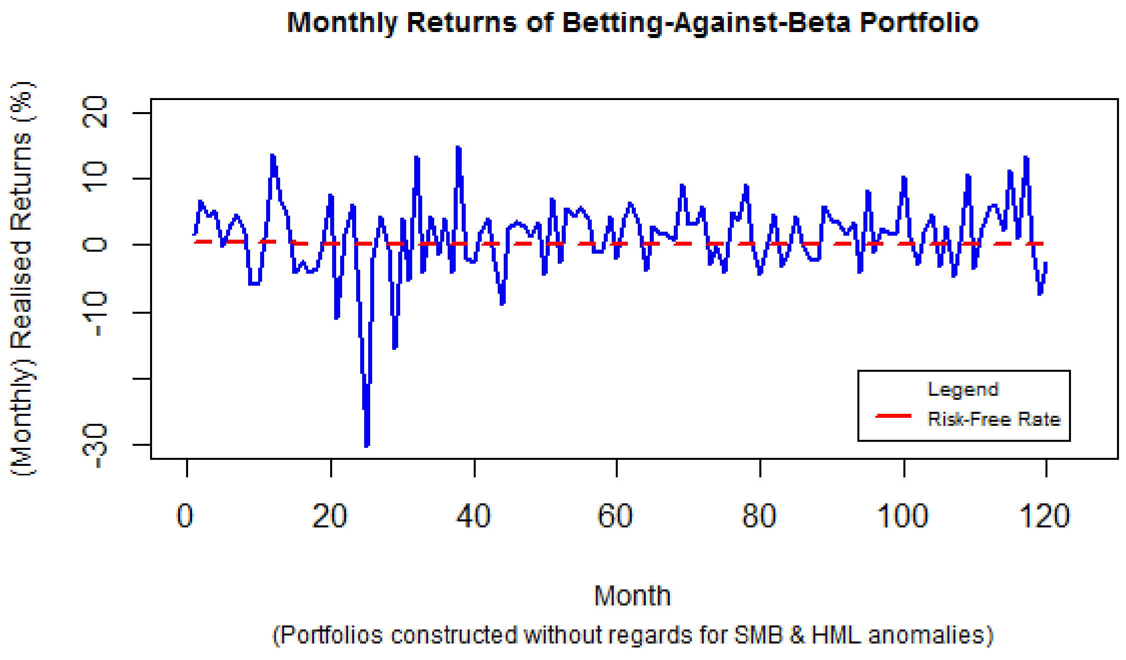

Figure 41

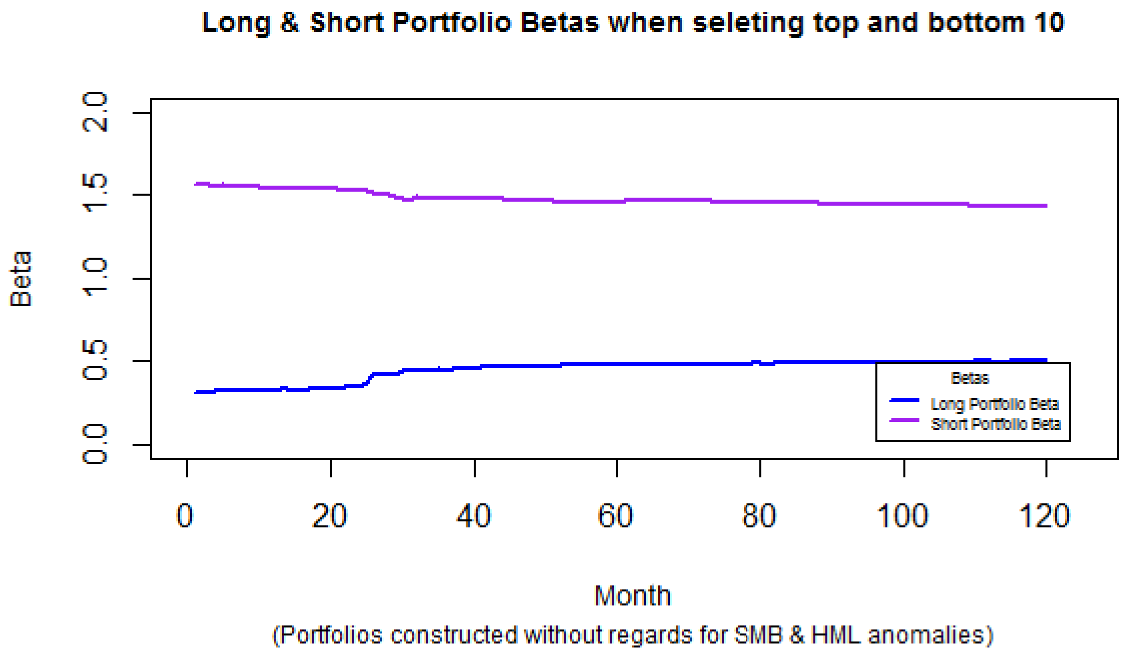

Figure 42

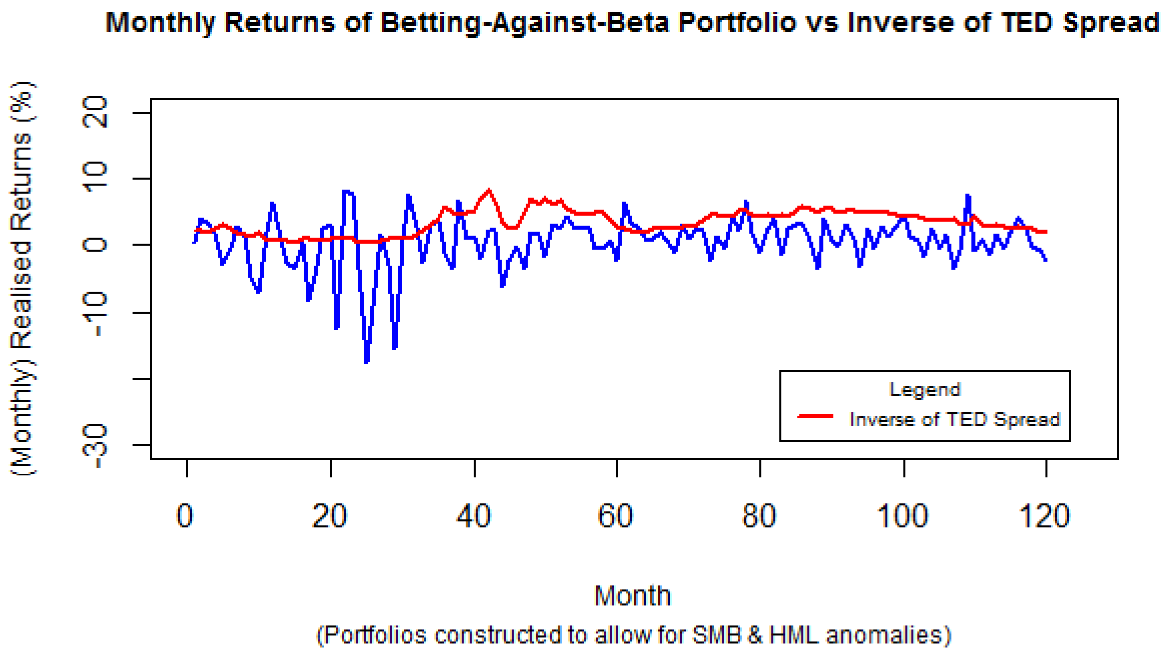

Figure 43

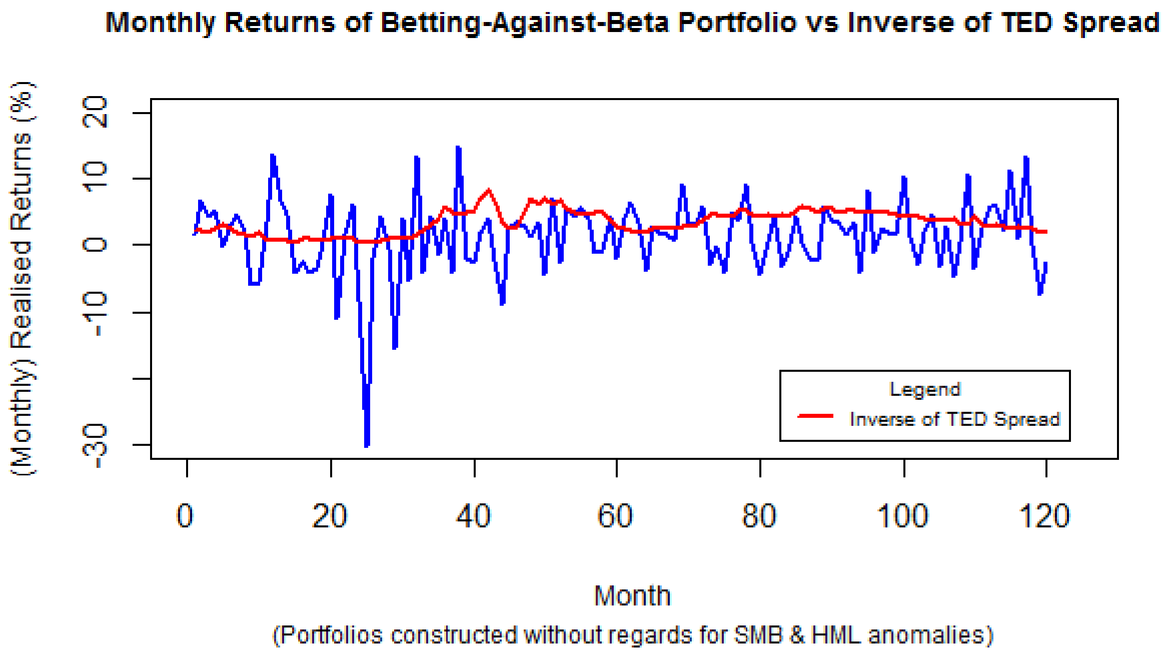

Figure 44

	TED Spread			
	Intercept Estimate	Standard Error	T-Test Statistic	P-value
LSVG	1.9627	0.4738	4.142	<10^(-4)
T10 & B10	3.18	0.697	4.563	<10^(-4)
	Sensitivity to TED Spread Estimate	Standard Error	T-Test Statistic	P-value
LSVG	-3.1159	0.6654	-4.683	<10^(-5)
T10 & B10	-3.995	0.979	-4.081	<10^(-4)
	F-Test Statistic	P-value	Adjusted R-Squared	
LSVG	21.93	<10^(-5)	0.1496	
T10 & B10	16.65	<10^(-4)	0.1163	

Figure 45

	Market Risk Premium & TED Spread			
	Intercept Estimate	Standard Error	T-Test Statistic	P-value
LSVG	0.5790	0.3814	1.518	0.1317
T10 & B10	1.7044	0.66	2.583	0.0110
	Sensitivity to TED Spread Estimate	Standard Error	T-Test Statistic	P-value
LSVG	-1.0985	0.5385	-2.040	0.0436
T10 & B10	-1.8434	0.9319	-1.978	0.0503
	Sensitivity to Market Risk Premium Estimate	Standard Error	T-Test Statistic	P-value
LSVG	0.5951	0.0612	9.723	10^(-15)
T10 & B10	0.6347	0.1059	5.993	10^(-7)
	F-Test Statistic	P-value	Adjusted R-Squared	
LSVG	66.92	<10^(-15)	0.5256	
T10 & B10	28.75	<10^(-10)	0.318	

Figure 46

	MAX			
	Intercept Estimate	Standard Error	T-Test Statistic	P-value
LSVG	2.1330	0.7830	2.724	0.00742
T10 & B10	2.6685	1.1487	2.323	0.0219
	Sensitivity to MAX Estimate	Standard Error	T-Test Statistic	P-value
LSVG	-0.2925	0.1185	-2.468	0.01501
T10 & B10	-0.2505	0.1730	-1.441	0.1524
	F-Test Statistic	P-value	Adjusted R-Squared	
LSVG	6.092	0.01501	0.04103	
T10 & B10	2.075	0.1524	0.008954	

Figure 47

	Market Risk Premium & MAX			
	Intercept Estimate	Standard Error	T-Test Statistic	P-value
LSVG	0.92089	0.56353	1.634	0.1049
T10 & B10	1.30532	0.98621	1.324	0.188
	Sensitivity to MAX Estimate	Standard Error	T-Test Statistic	P-value
LSVG	-0.15480	0.08459	-1.83	0.0698
T10 & B10	-0.09561	0.14803	-0.646	0.52
	Sensitivity to Market Risk Premium Estimate	Standard Error	T-Test Statistic	P-value
LSVG	0.62761	0.05731	10.952	<10^(-15)
T10 & B10	0.70578	0.10029	7.037	<10^(-9)
	F-Test Statistic	P-value	Adjusted R-Squared	
LSVG	66.09	<10^(-15)	0.5224	
T10 & B10	26.63	<10^(-9)	0.2977	

Figure 48

Note: All figures are corresponding to monthly percentage returns unless otherwise stated.

Financial Ratio	LSVG BAB	Top 10 & Bottom 10 BAB	Market
Yearly Arithmetic Mean	5.961563	16.28905	10.3398
Yearly Geometric Mean	4.903872	13.8603	8.55684
Monthly Arithmetic Mean	0.483718	1.265513	0.8233248
Monthly Geometric Mean	0.399749	1.087555	0.6865436
Standard Deviation	4.027605	5.831692	5.203945
Skewness	-1.59089	-1.30488	-0.5794711
Kurtosis	4.916034	6.480806	2.45112
Covariance with the Market	14.71091	16.3572	27.08104
Pearson Correlation with the Market	0.701876	0.5389914	1
Spearman Correlation with the Market	0.60316	0.4656087	1
Beta	0.593041	0.6710255	-
Alpha from CAPM with intercept	0.070438	0.8056508	0.09617862
Standard Error of Alpha	0.25943	0.4502841	0.099604
P-value of Alpha	0.786474	0.07614711	0.3362142
1 month 95% CL VaR (Non-Parametric)	6.229783	6.007248	8.010867
1 month 95% CL VaR (“Modified”)	7.338637	9.495528	8.265246
1 month 95% CL ETL Non-Parametric	11.71039	13.60491	11.93544
1 month 95% CL ETL (“Modified”)	13.05365	21.14774	13.73124
Semi-Standard Deviation	3.283713	4.513477	3.918679

Downside Deviation (MAR = long run risk-free rate)	3.247739	4.131456	3.715851
Downside Deviation (MAR = long run market risk premium)	3.339058	4.221638	3.816268
Jensen’s Alpha	-0.27829	0.455787	-0.1877047
Sharpe Ratio	0.021005	0.1485667	0.08151637
Treynor’s Measure	0.142655	1.291151	0.4242067
Sortino’s Ratio (MAR = long run risk-free rate)	0.026049	0.209707	0.1141614
Sortino’s Ratio (MAR = long run market risk premium)	0.025336	0.2052273	0.1111575
Appraisal Ratio	-0.09909	0.0938907	-

Figure 49

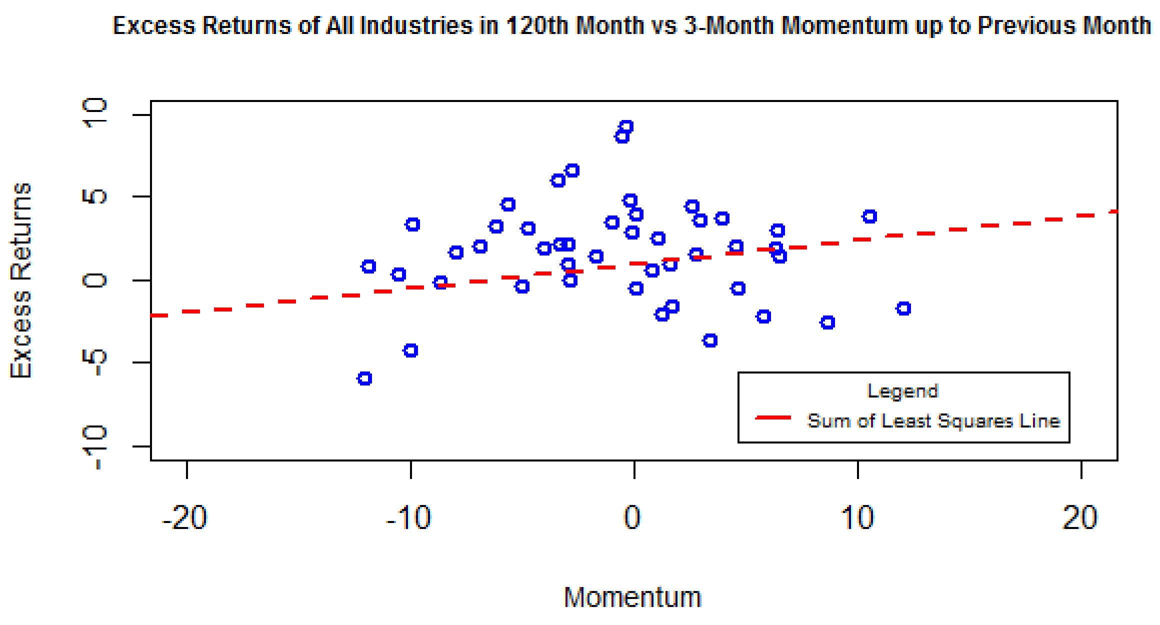

Figure 50

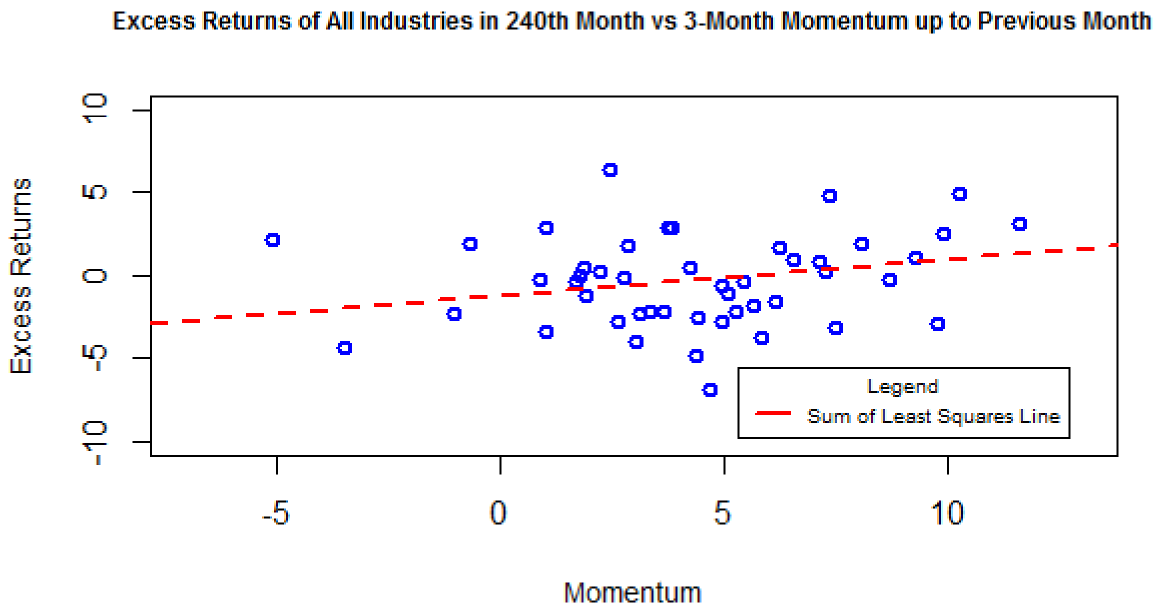

Figure 51

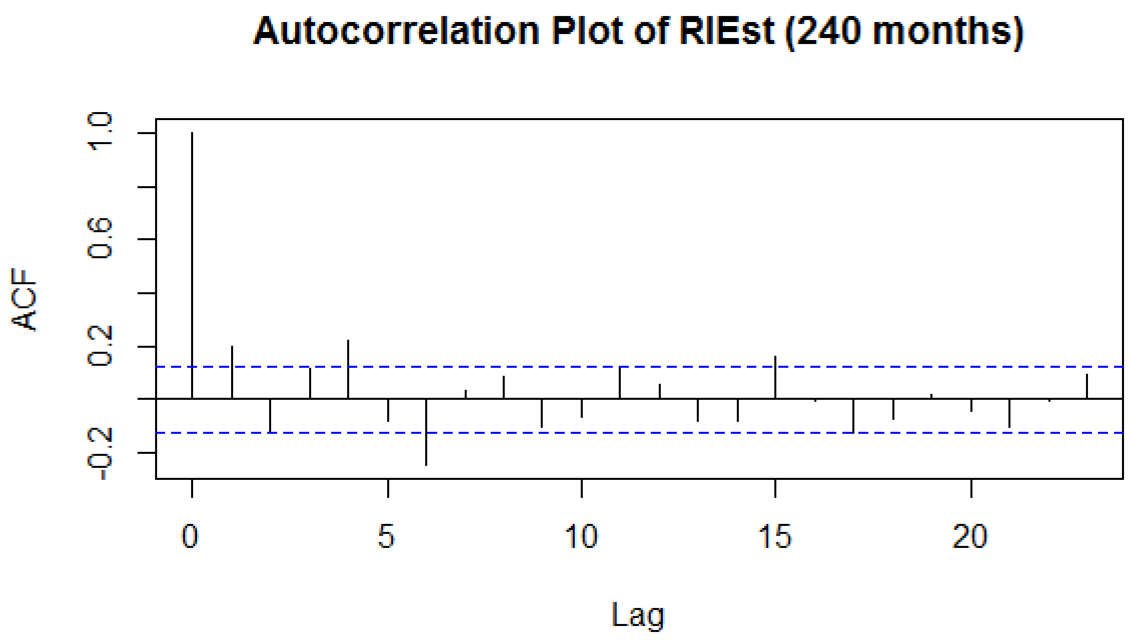

Figure 52

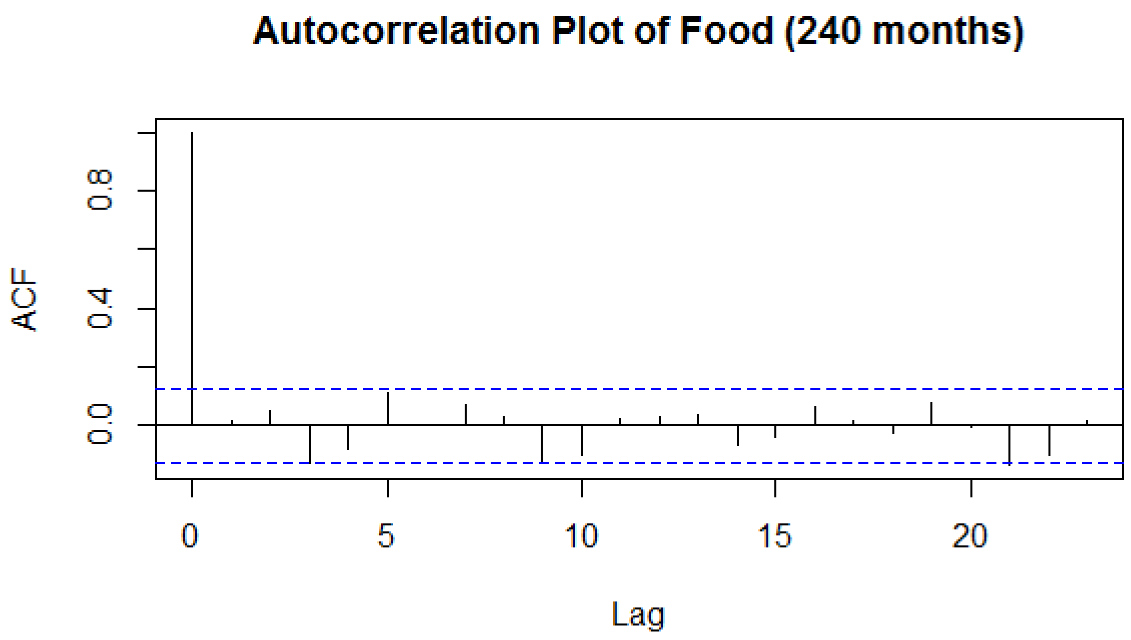

Figure 53

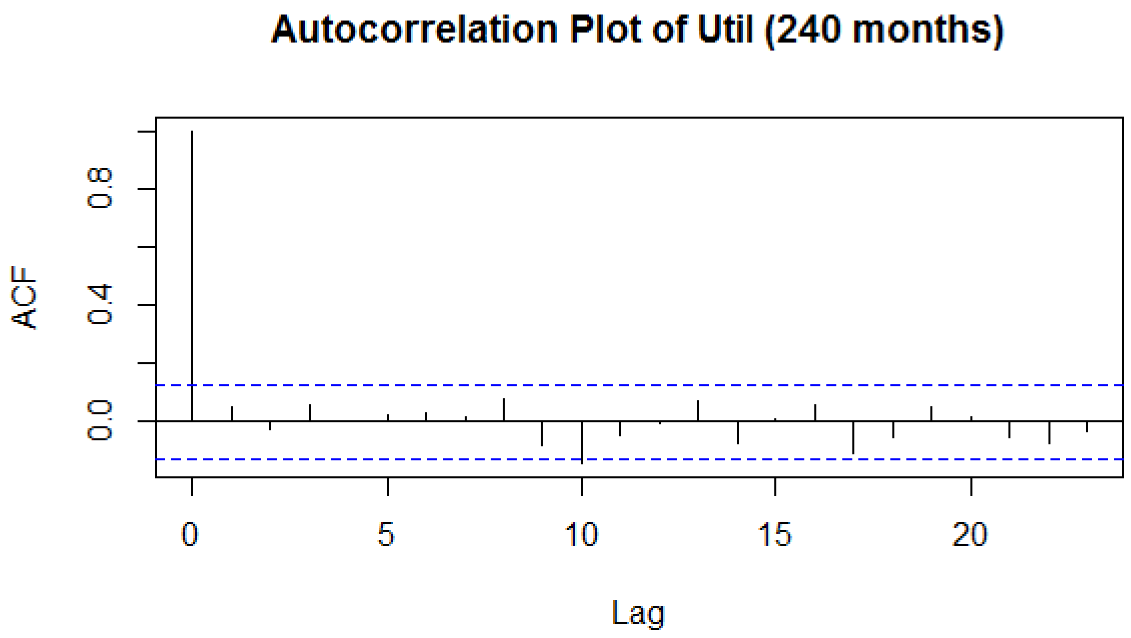

Figure 53

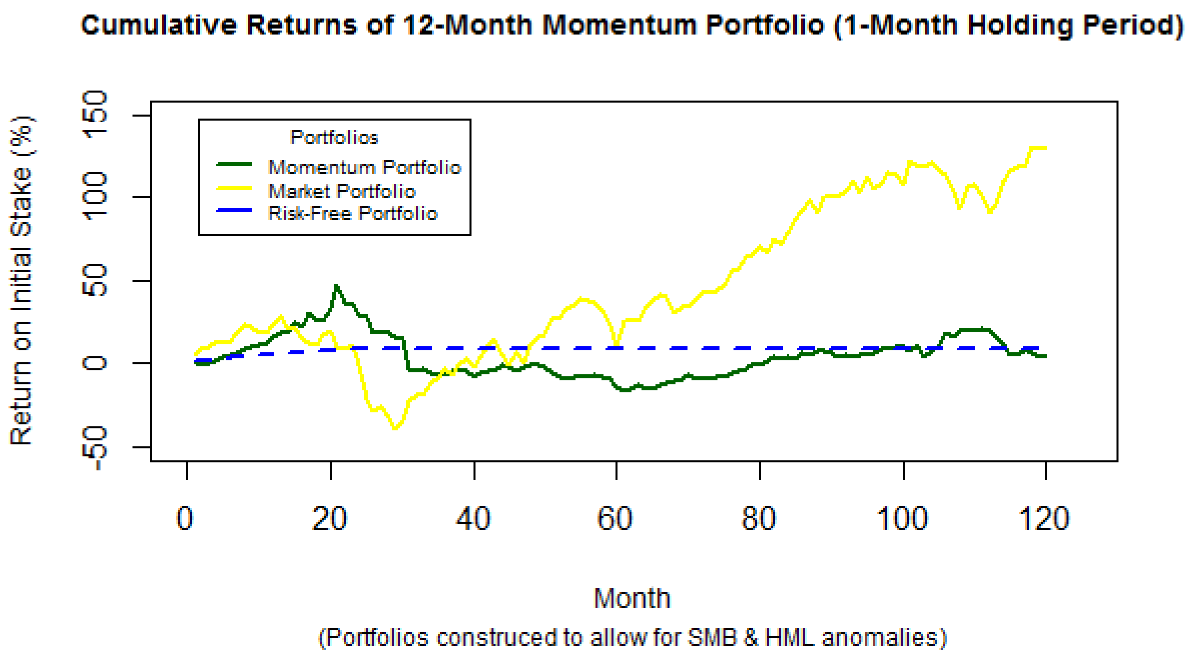

Figure 54

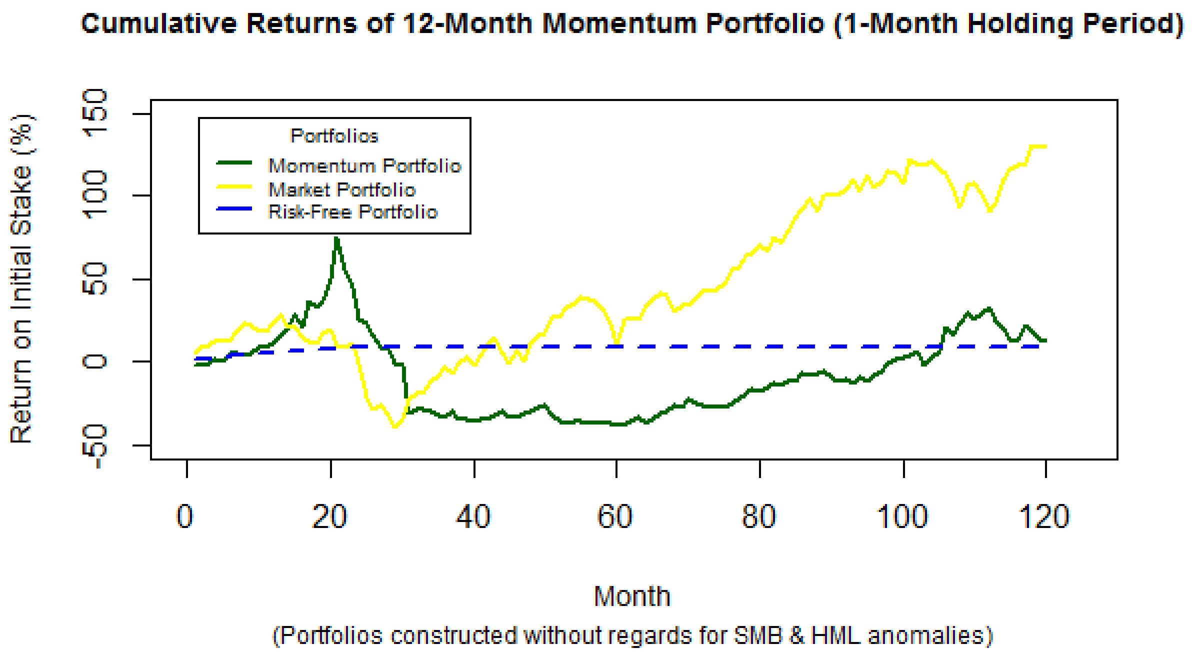

Figure 55

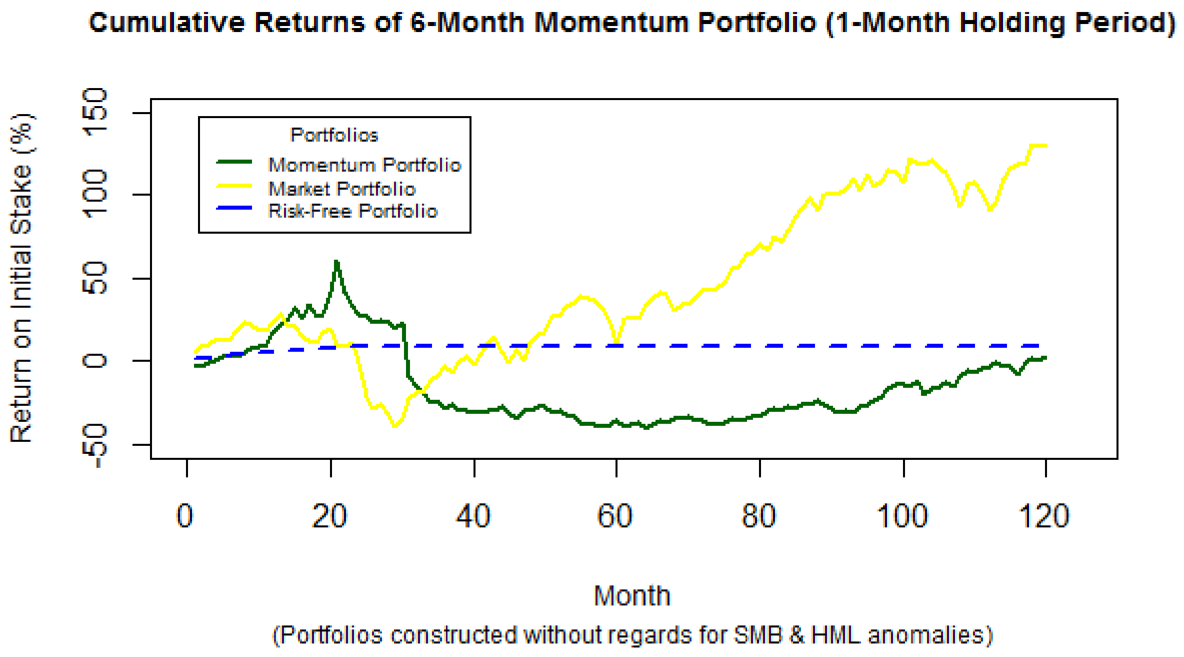

Figure 56

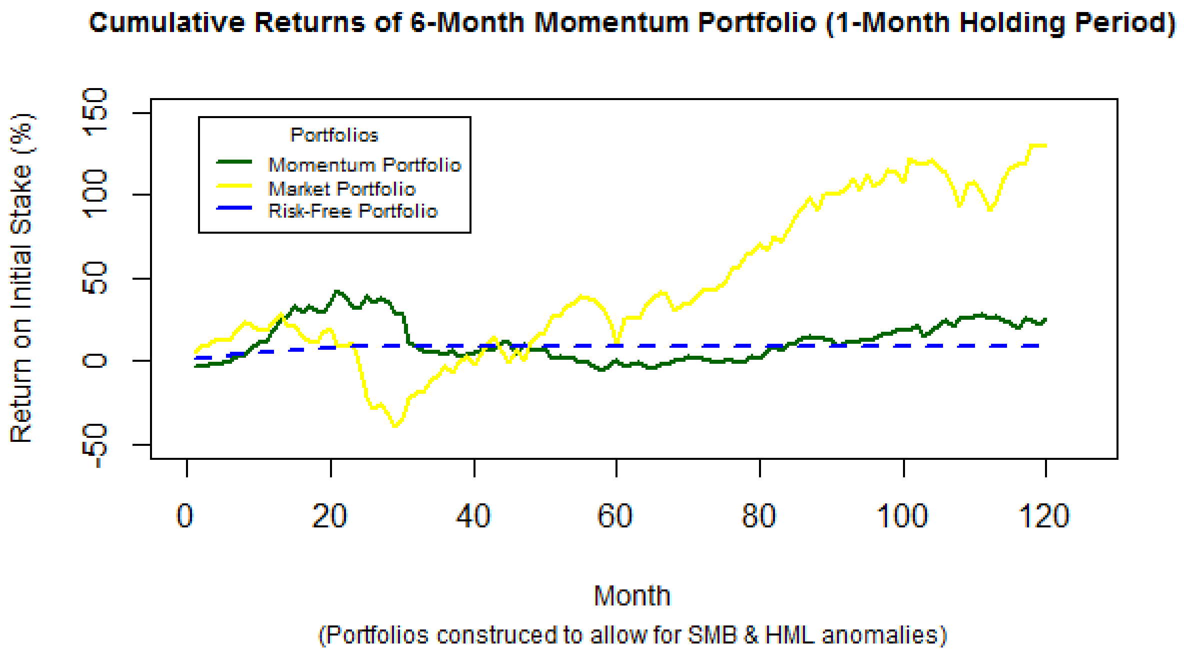

Figure 57

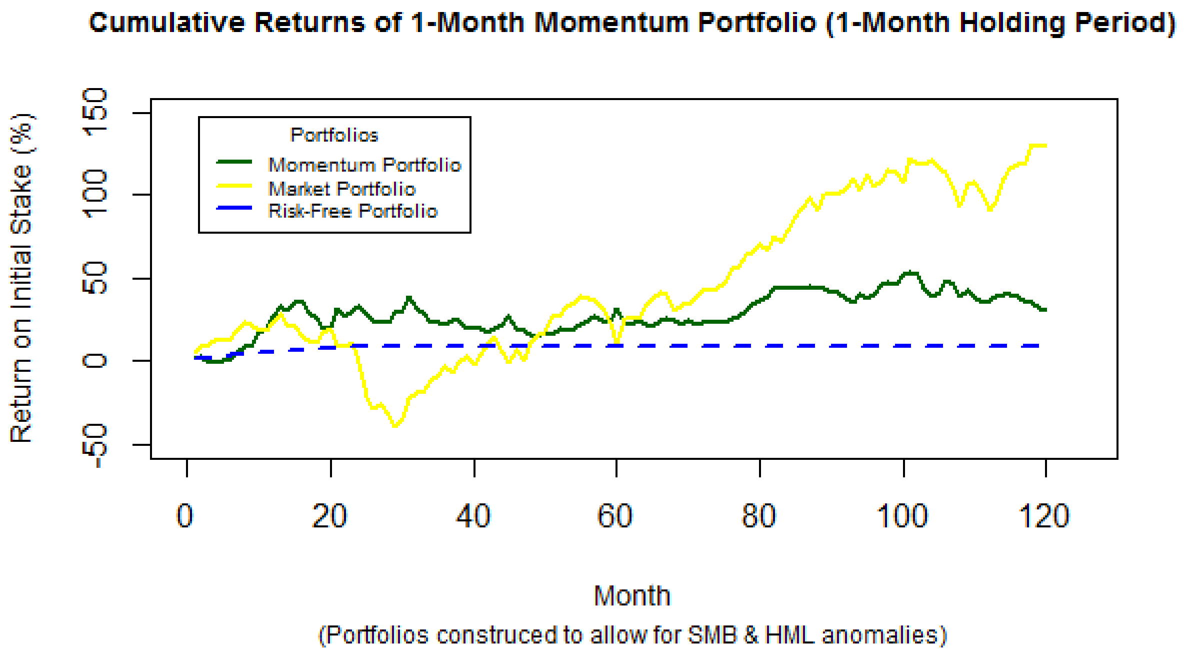

Figure 58

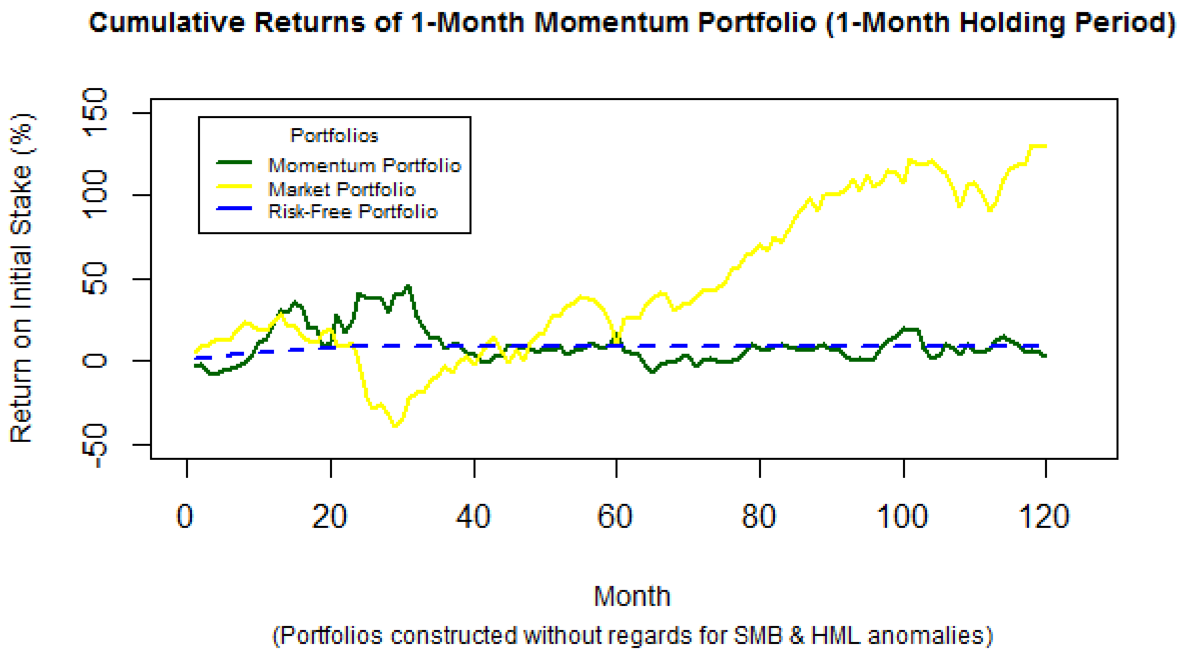

Figure 59

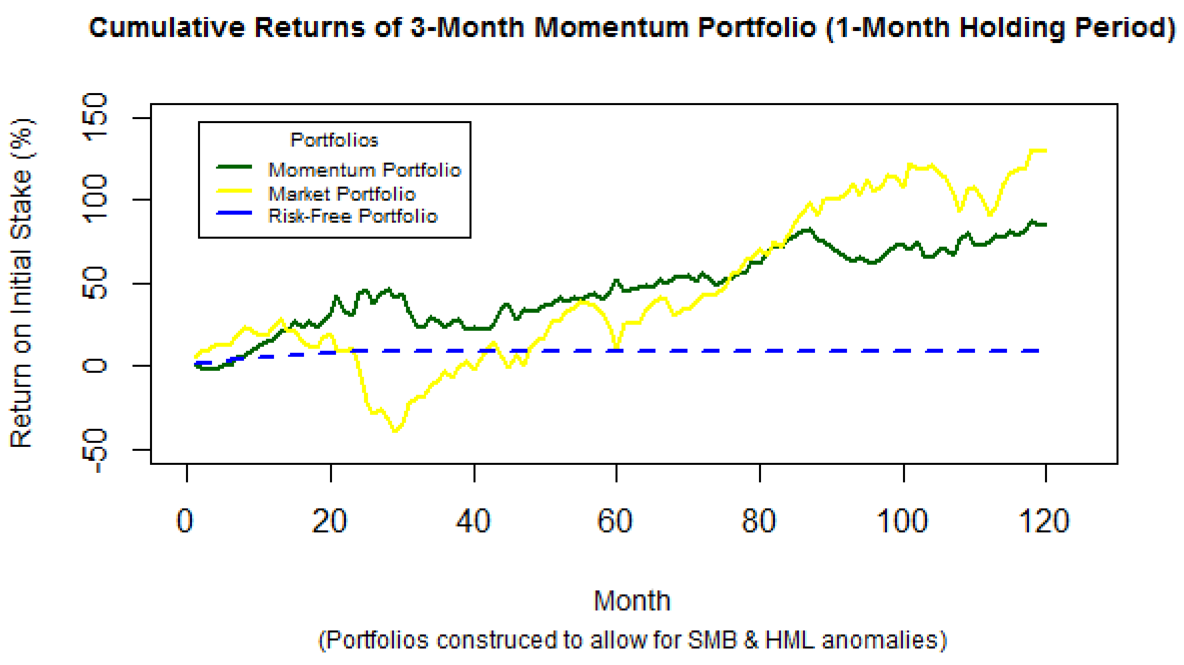

Figure 60

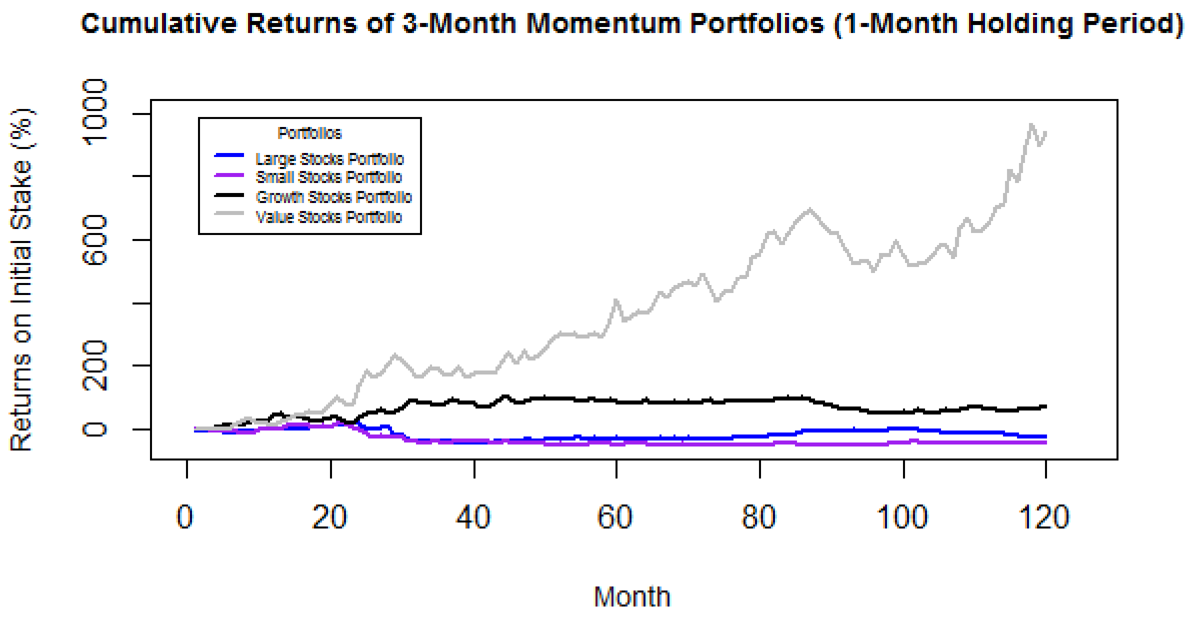

Figure 61

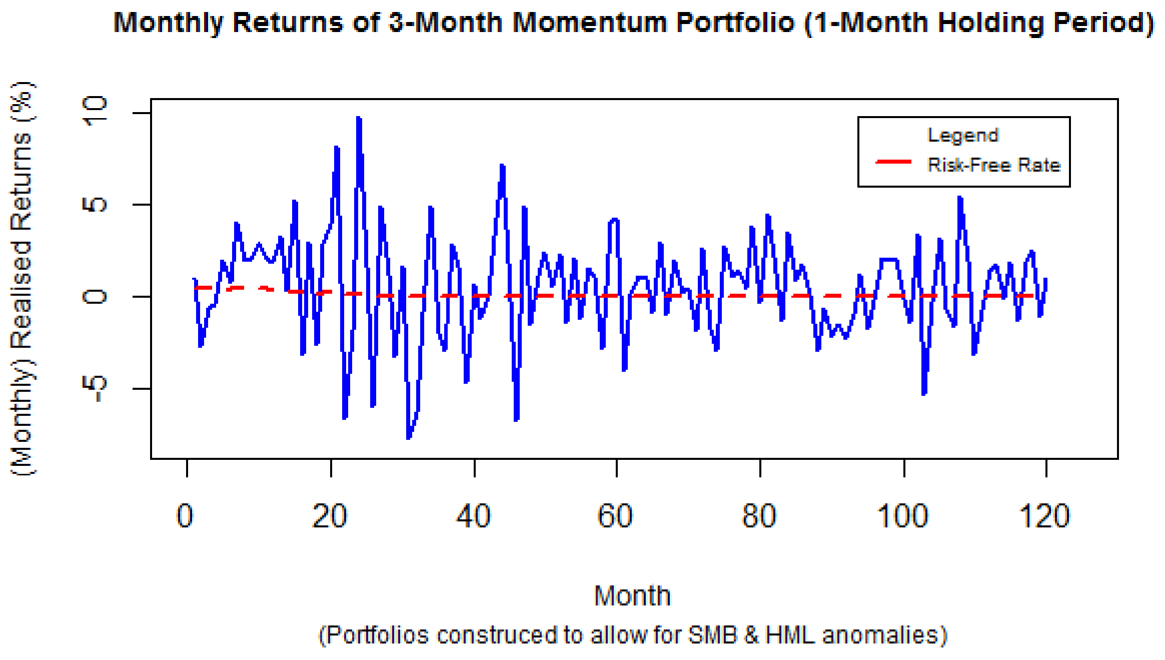

Figure 62

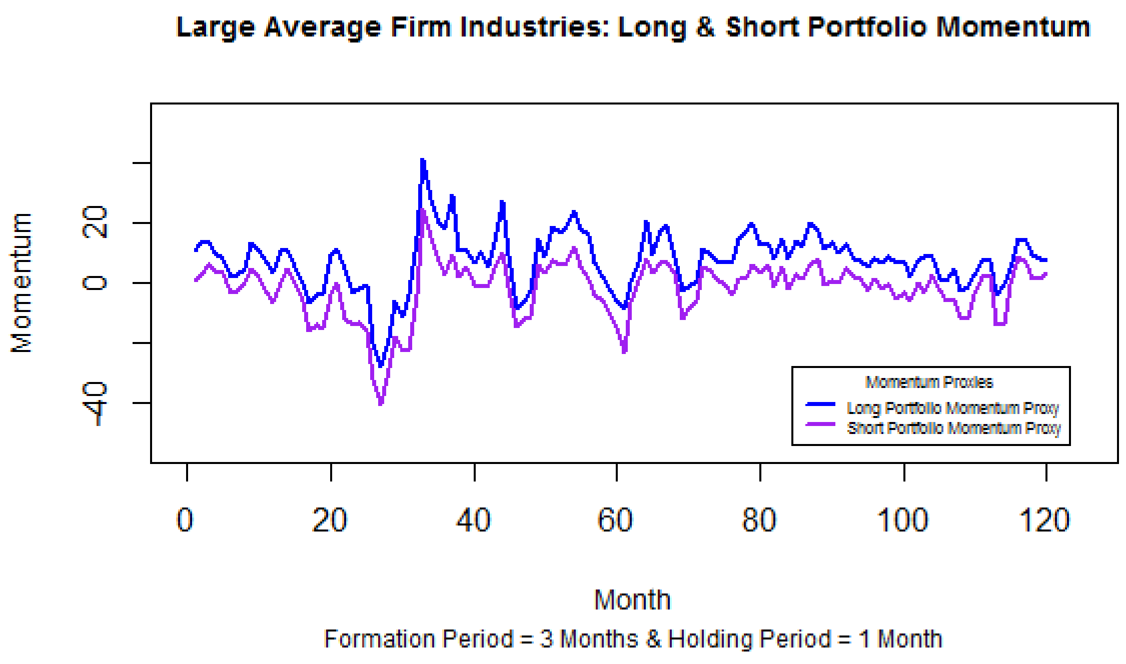

Figure 63

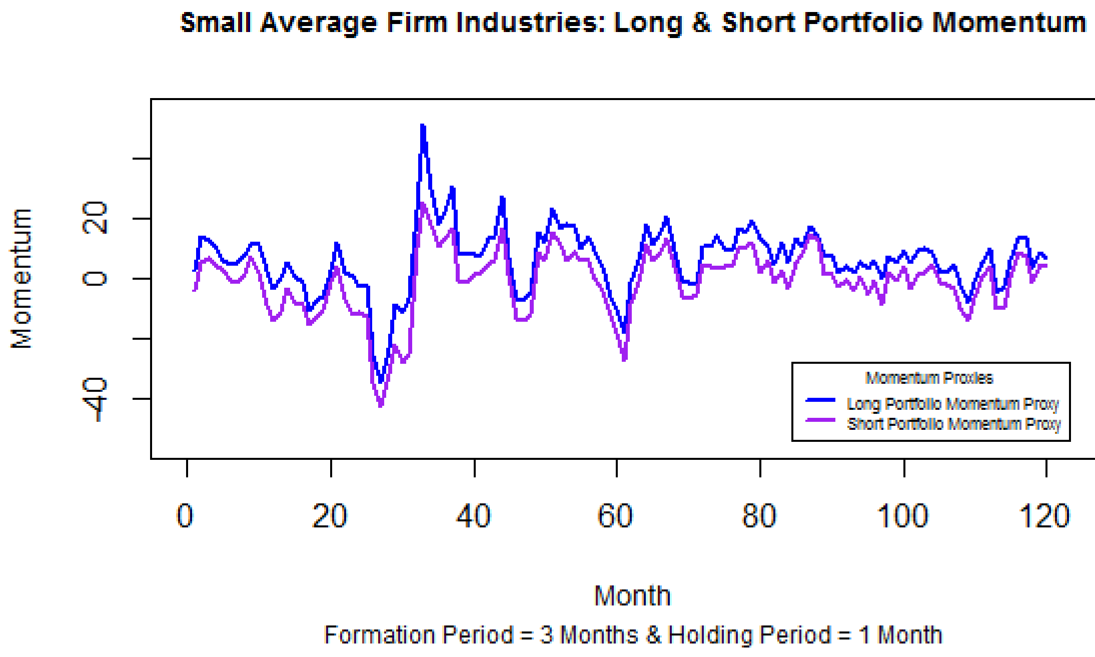

Figure 64

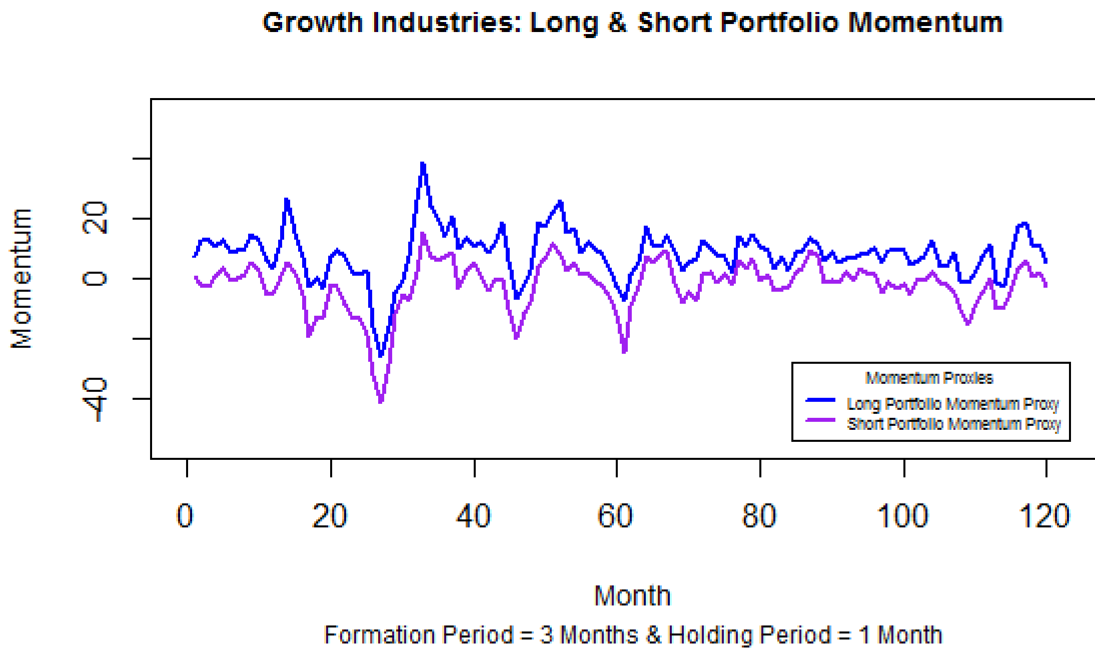

Figure 65

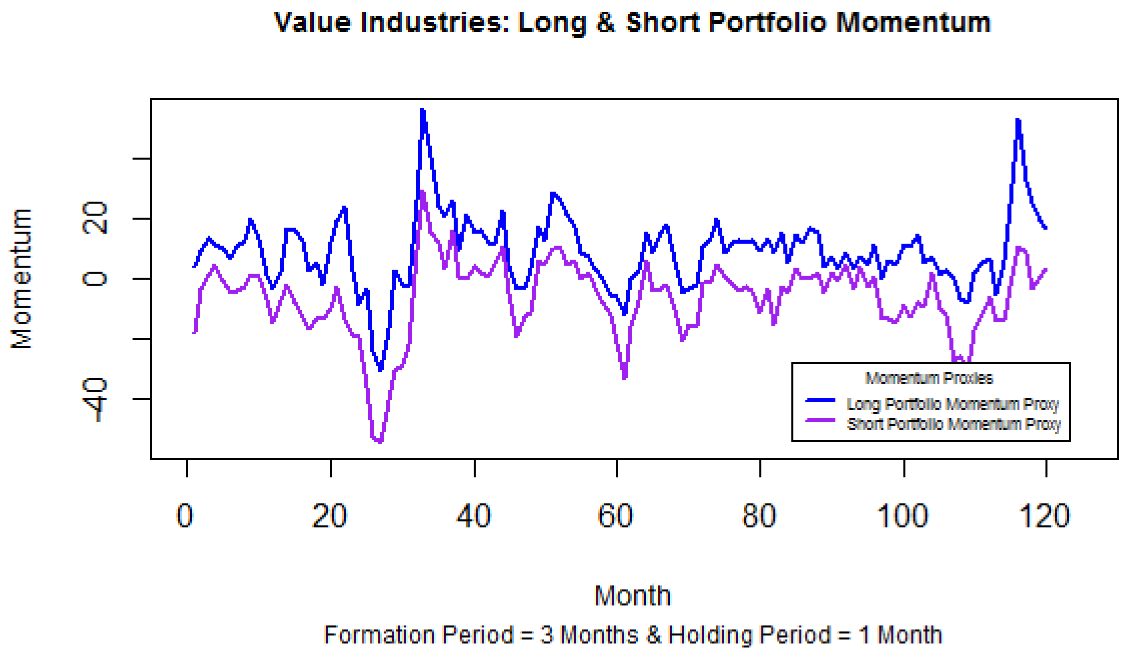

Figure 66

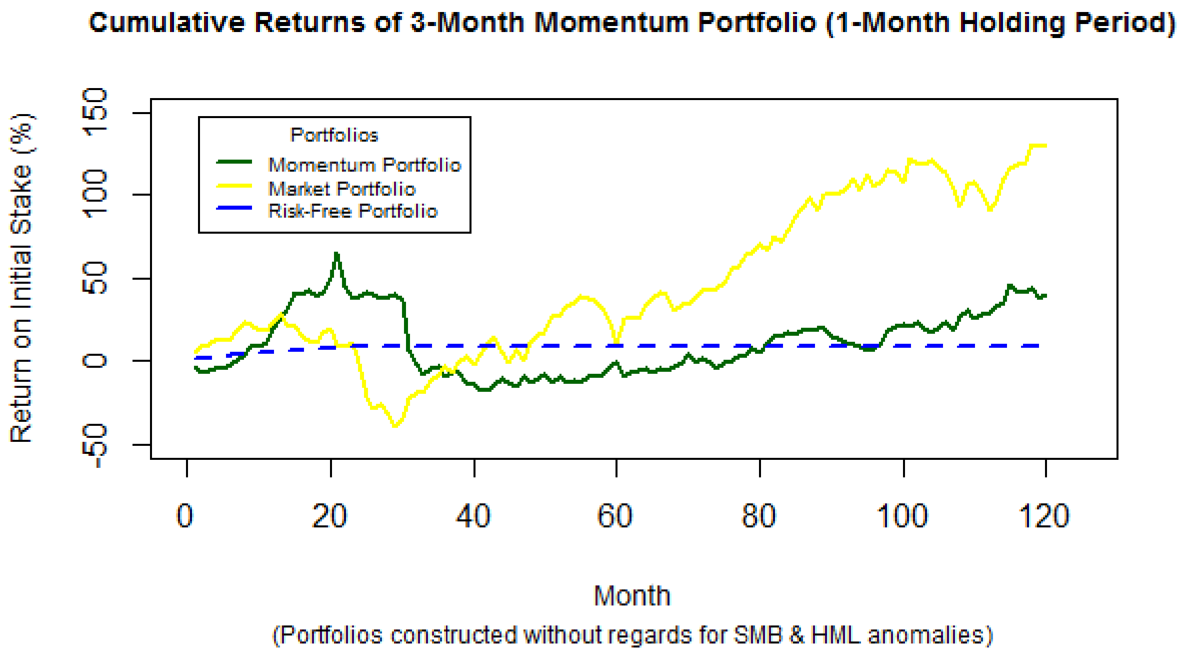

Figure 67

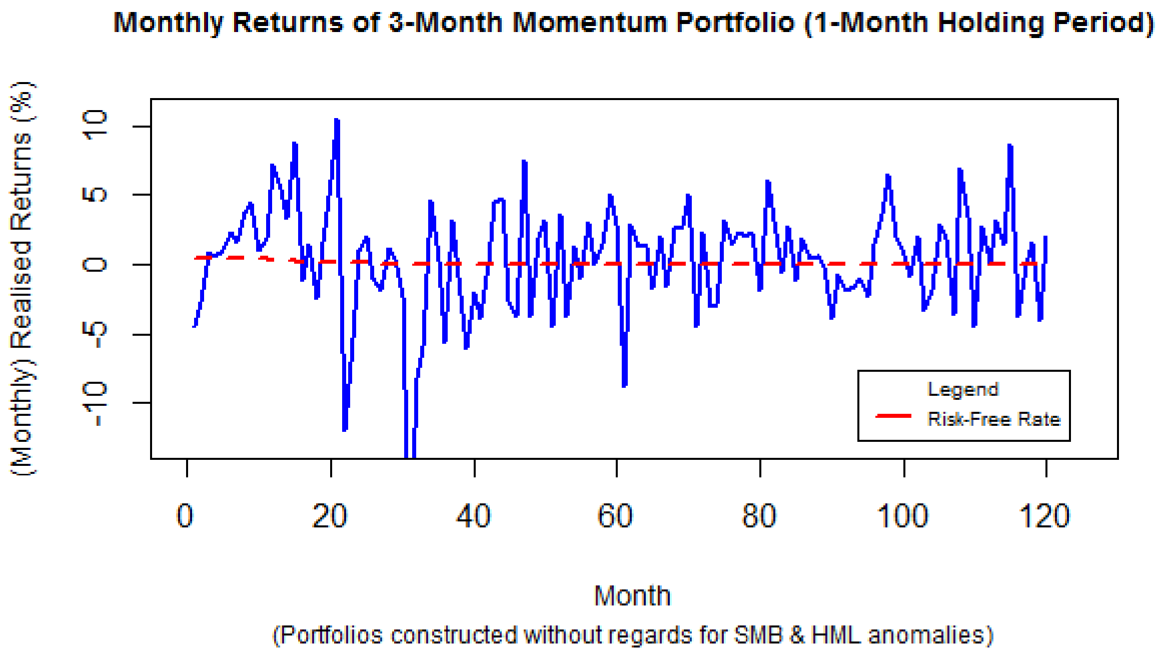

Figure 68

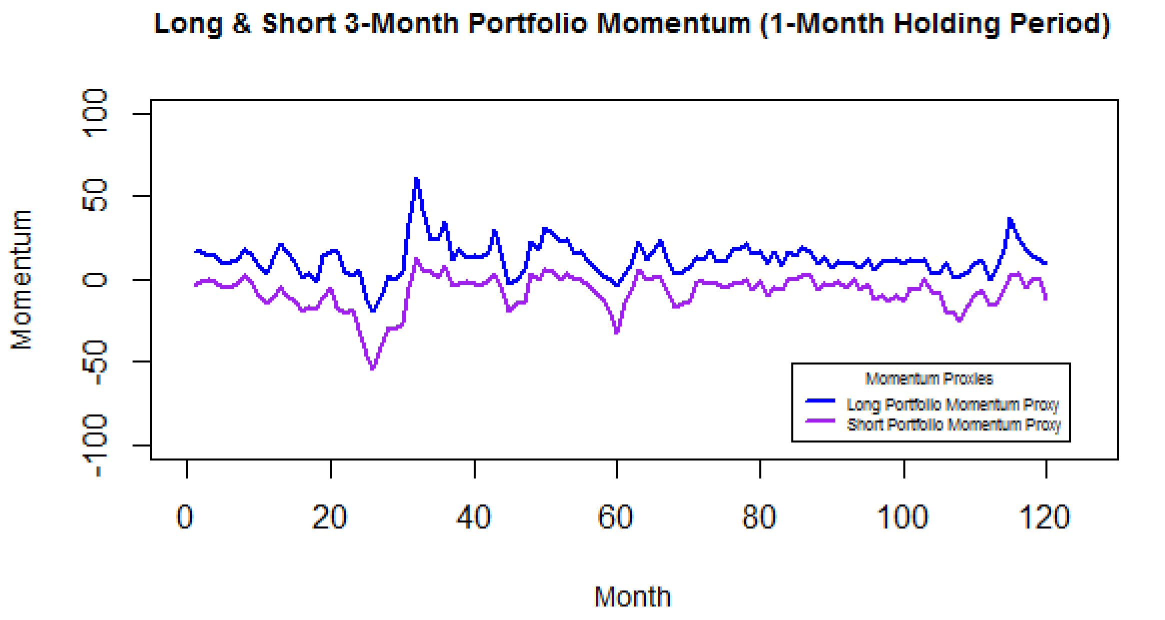

Figure 69

Note: here x denotes rate of return.

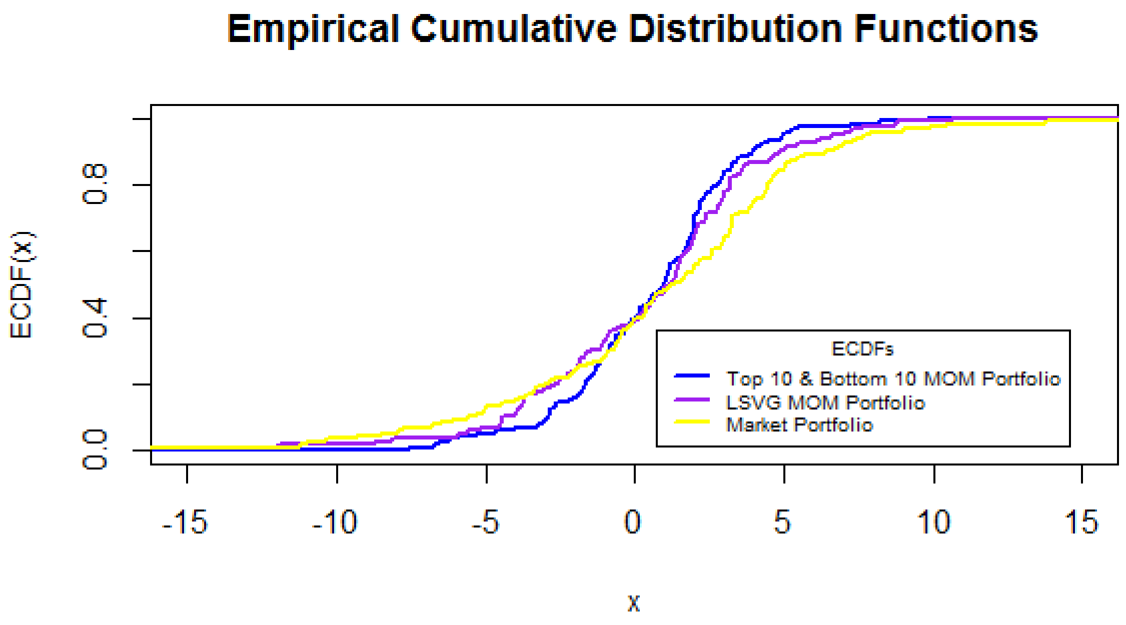

Figure 70

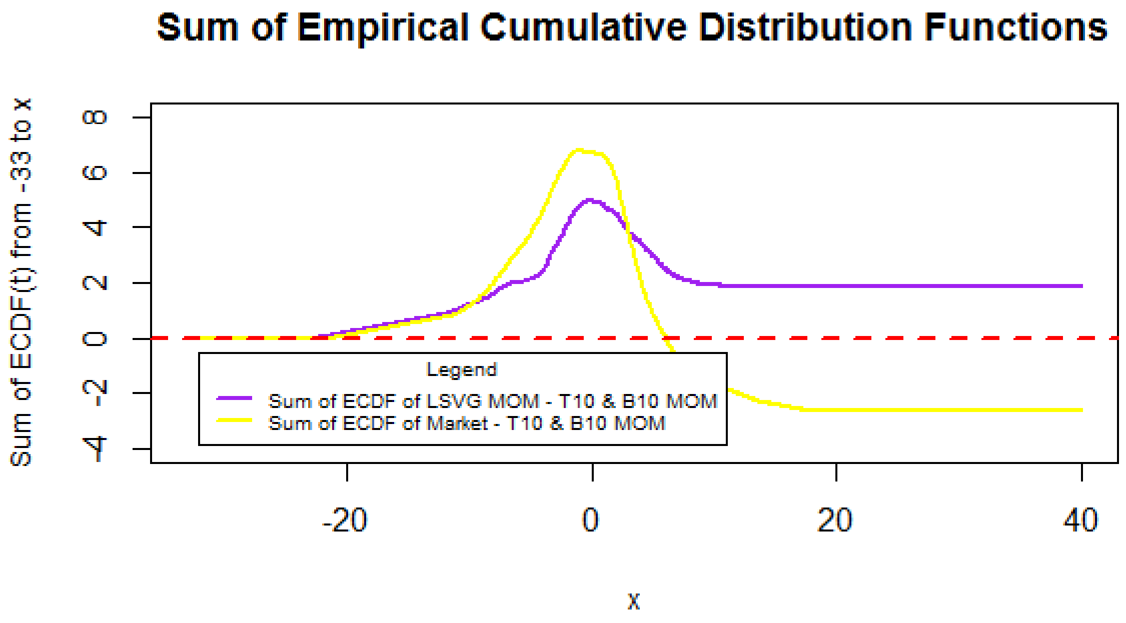

Figure 71

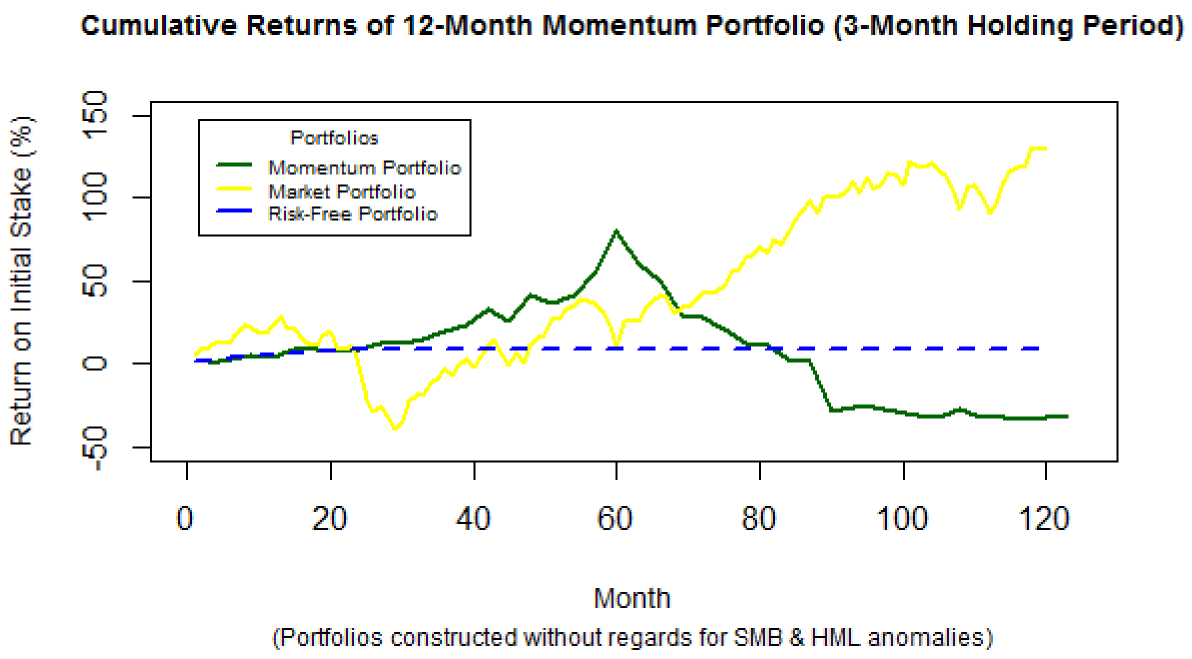

Figure 72

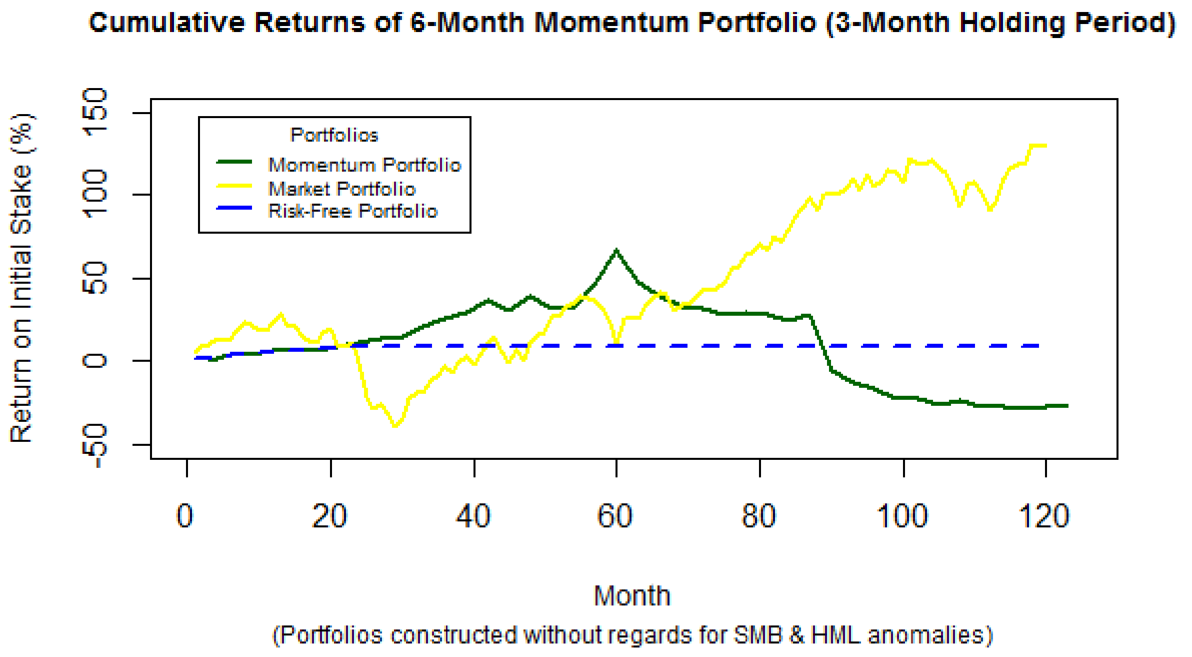

Figure 73

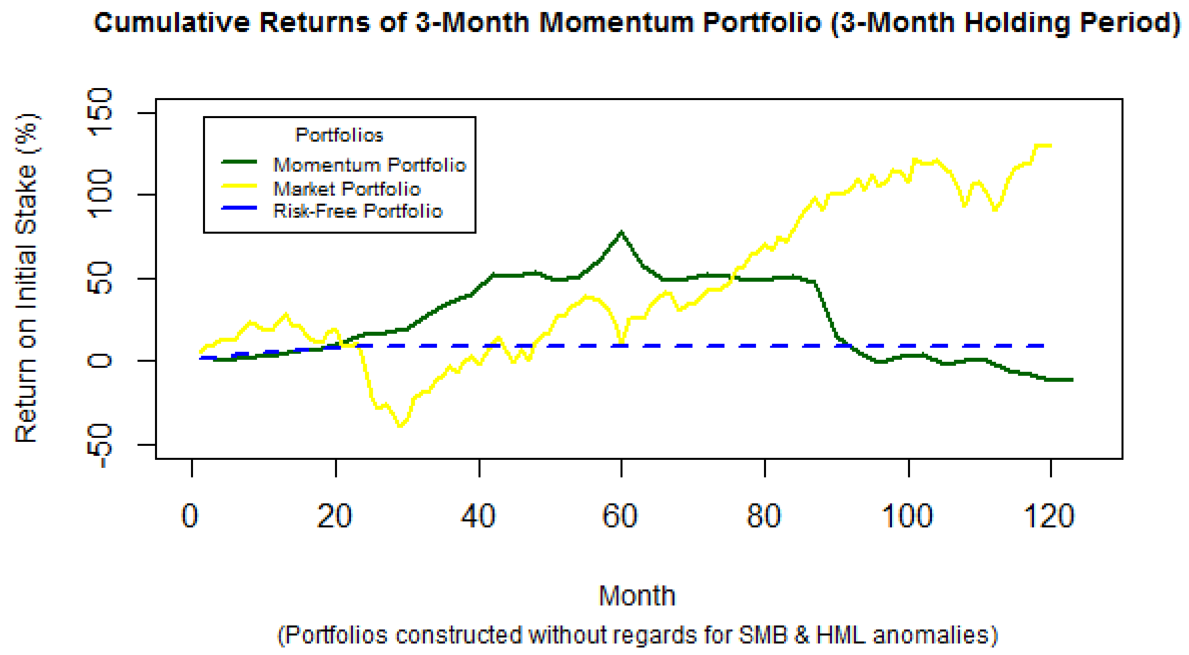

Figure 74

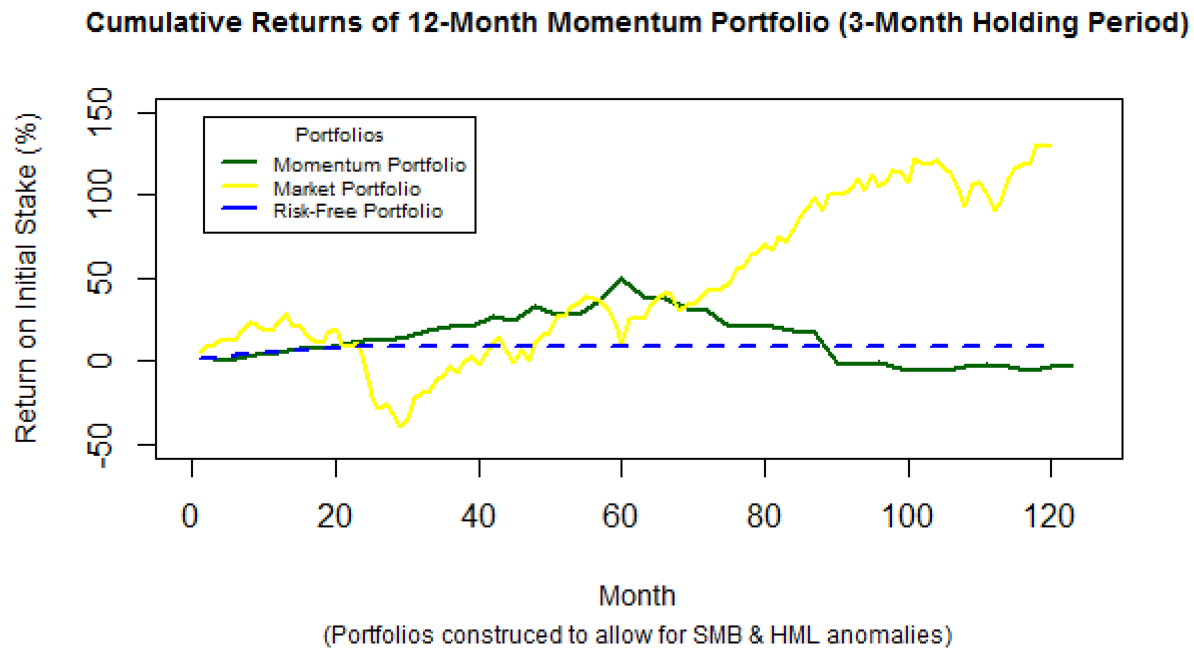

Figure 75

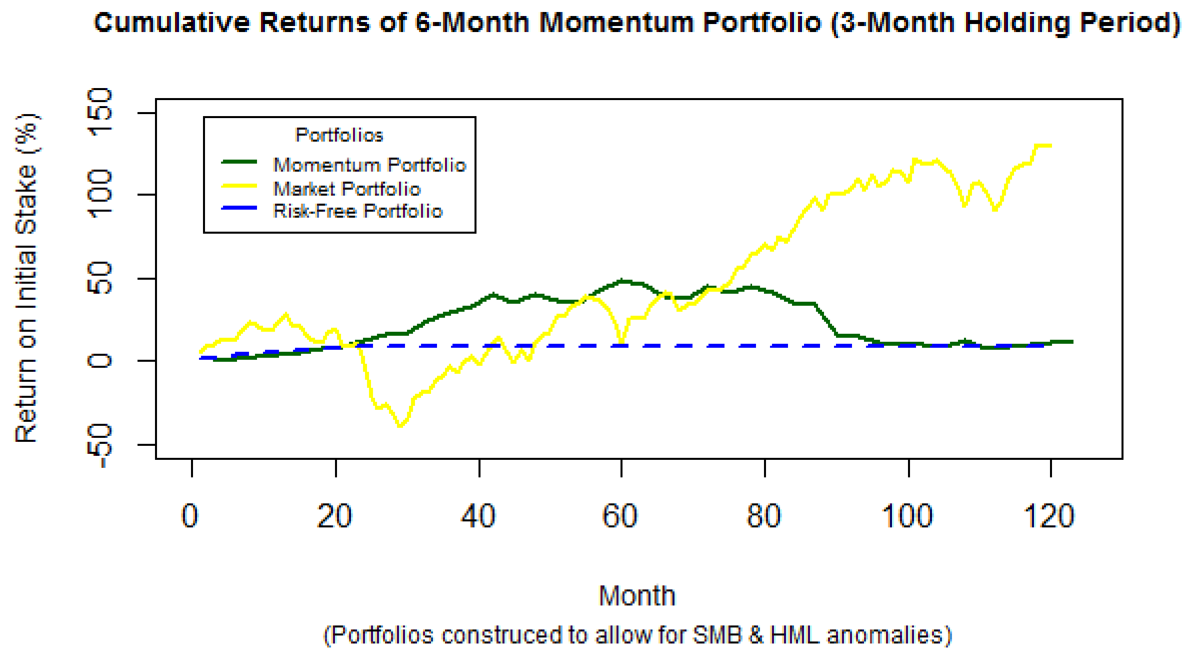

Figure 76

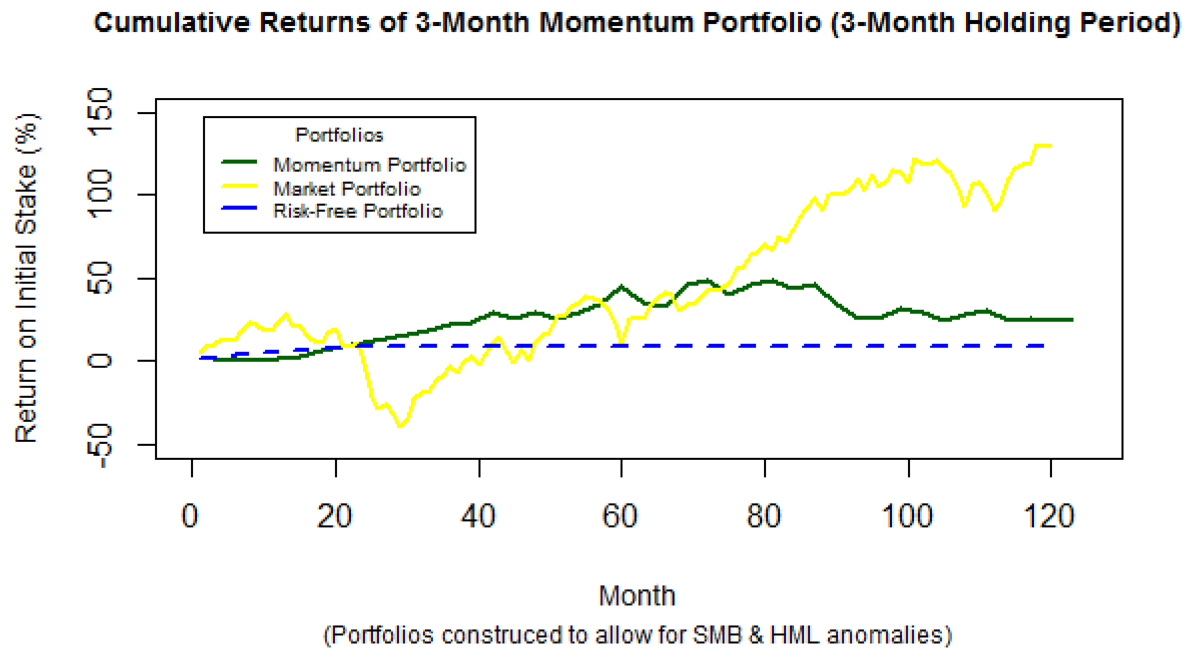

Figure 77

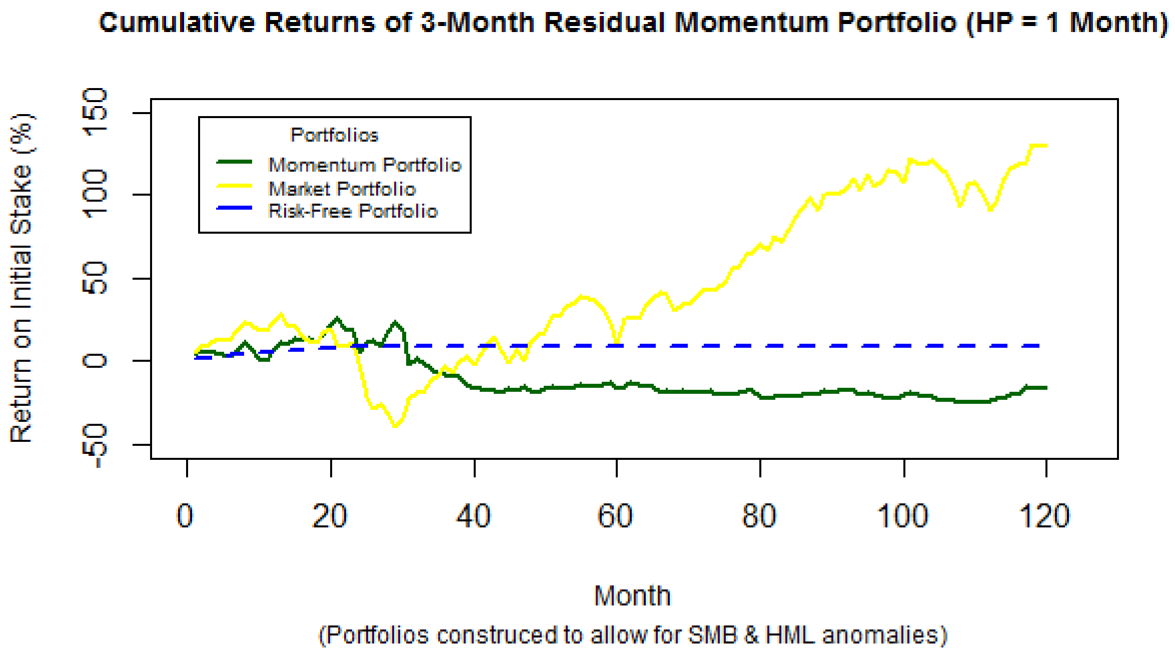

Figure 78

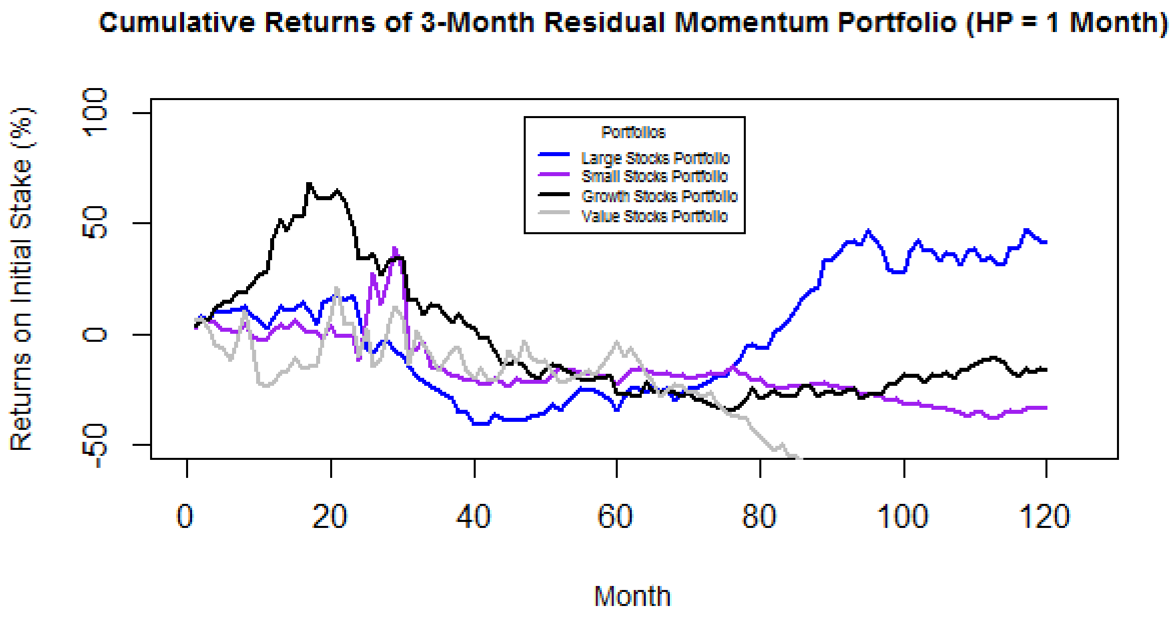

Figure 79

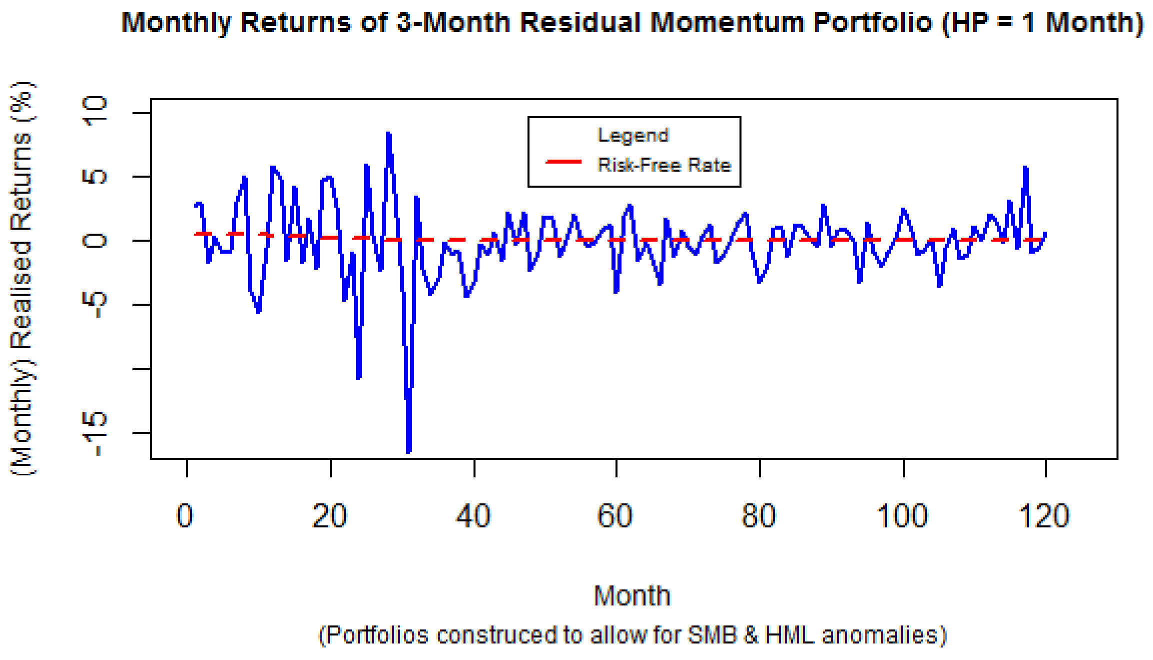

Figure 80

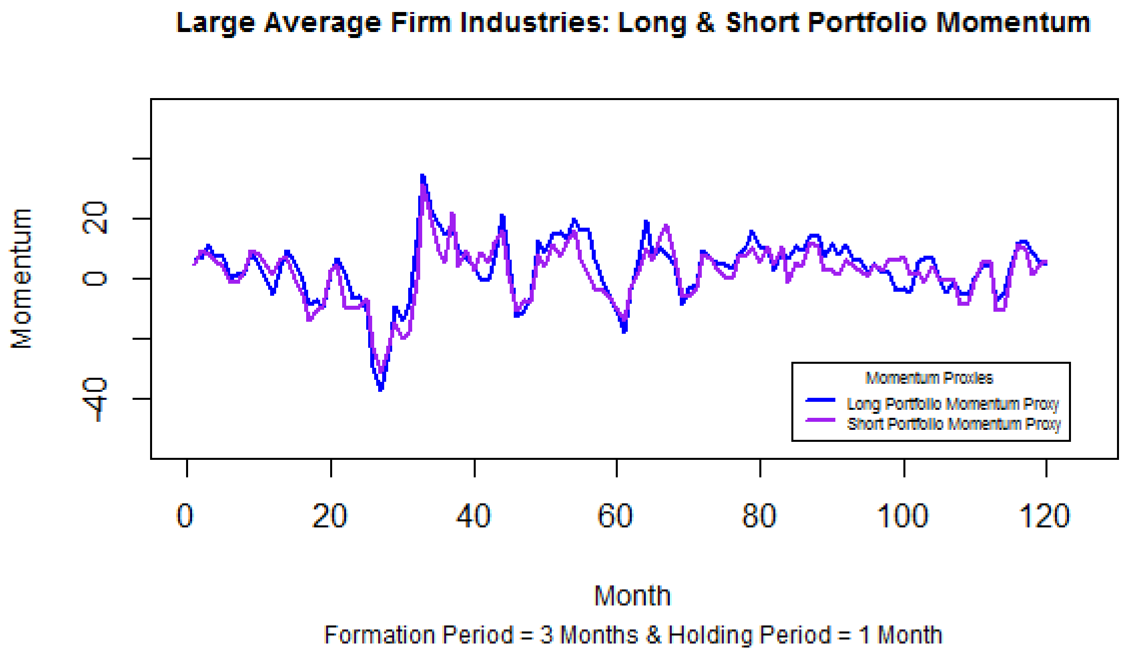

Figure 81

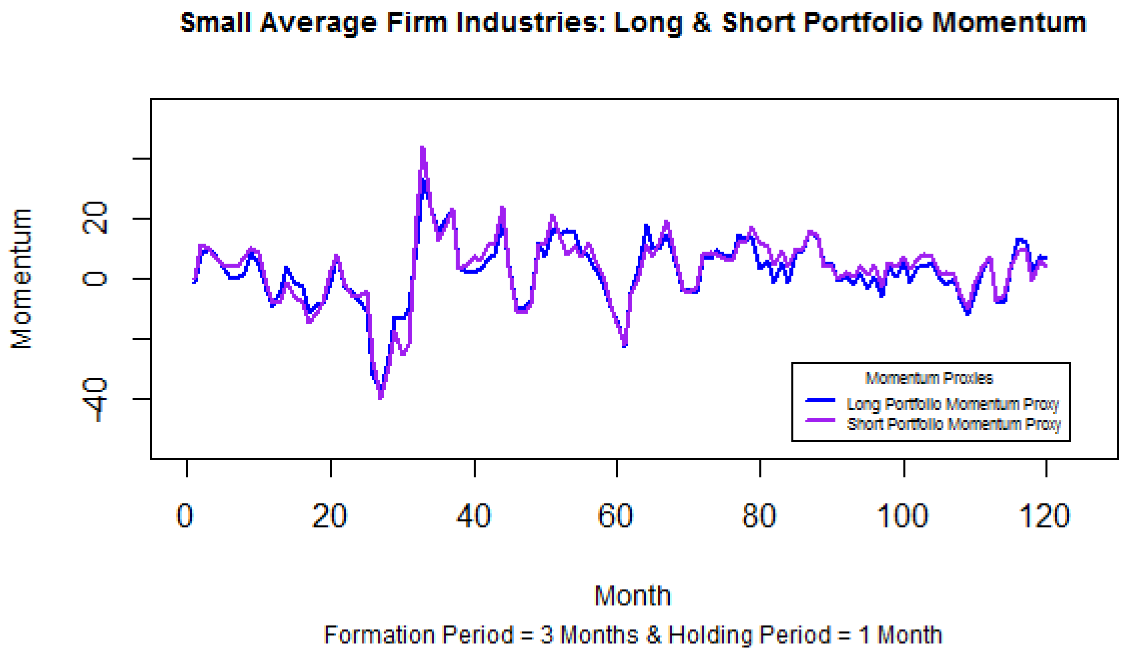

Figure 82

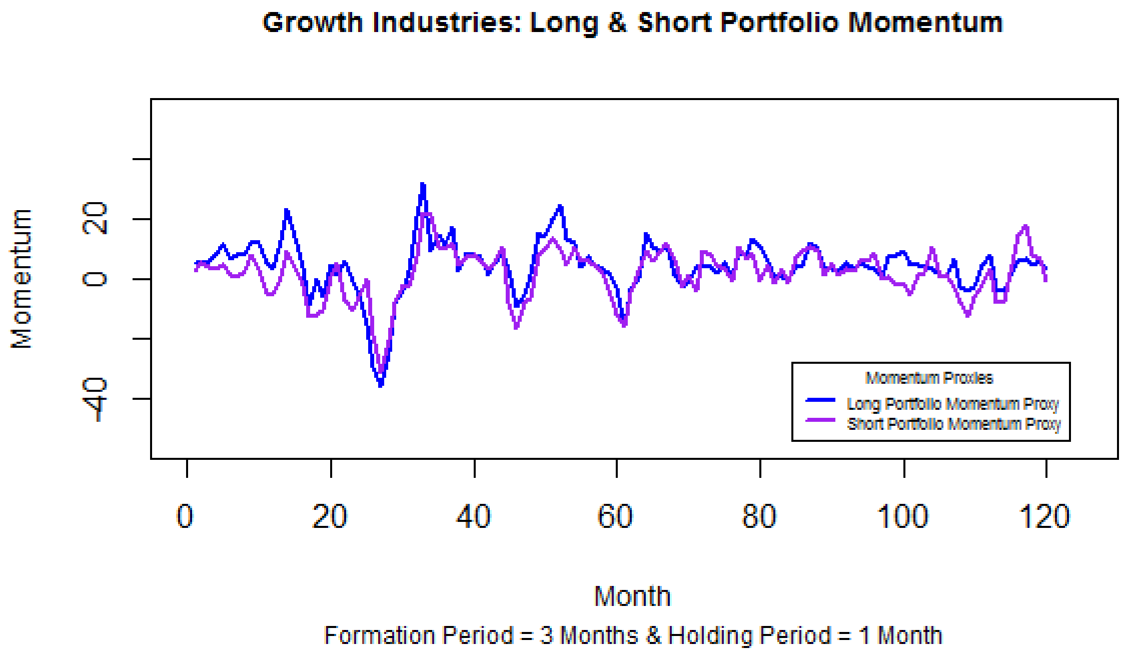

Figure 83

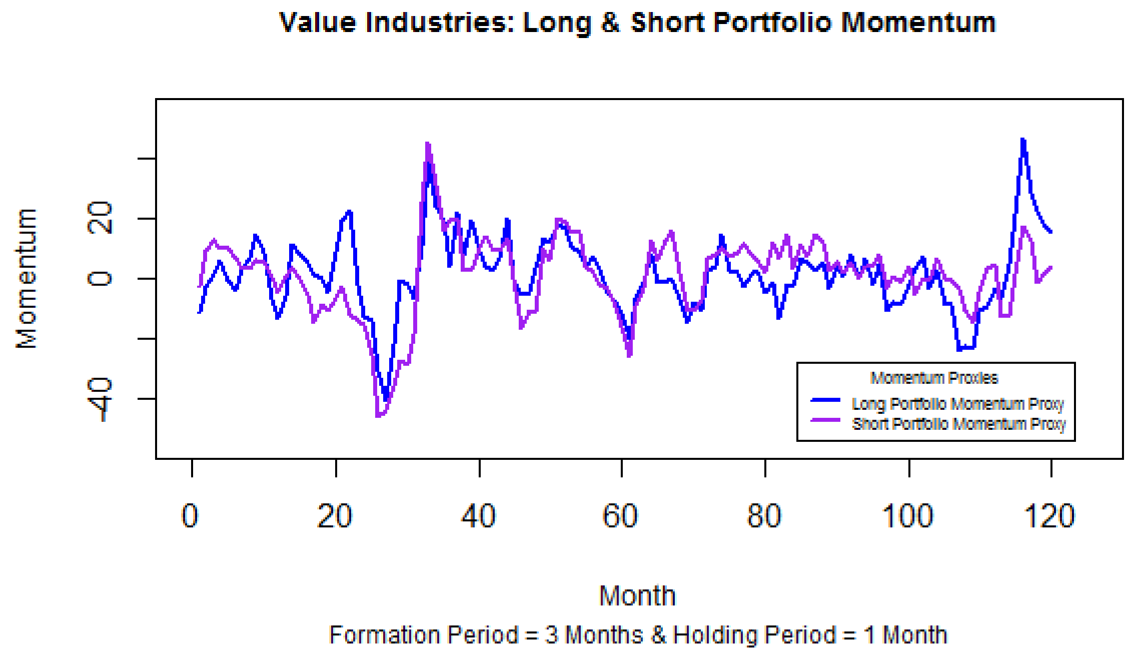

Figure 84

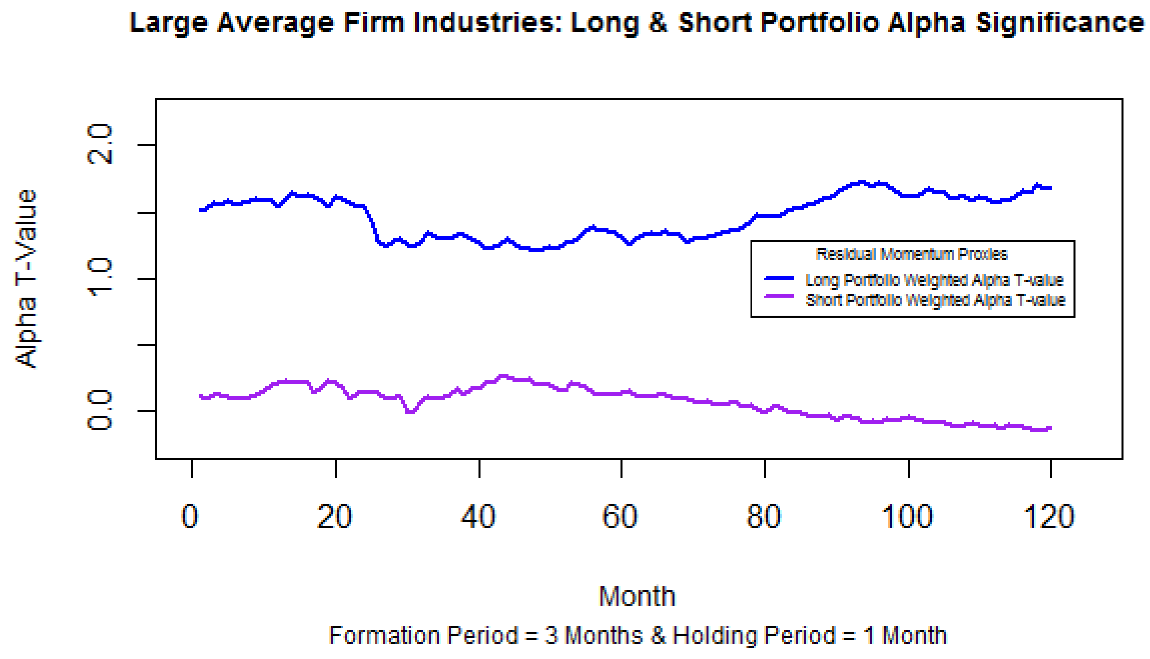

Figure 85

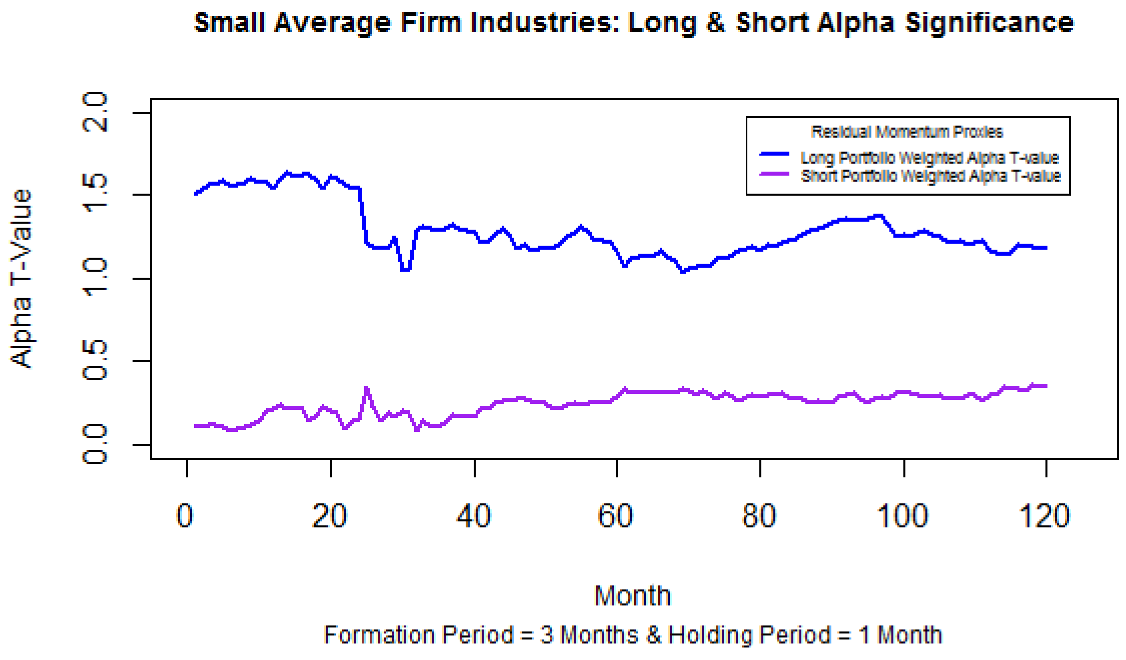

Figure 86

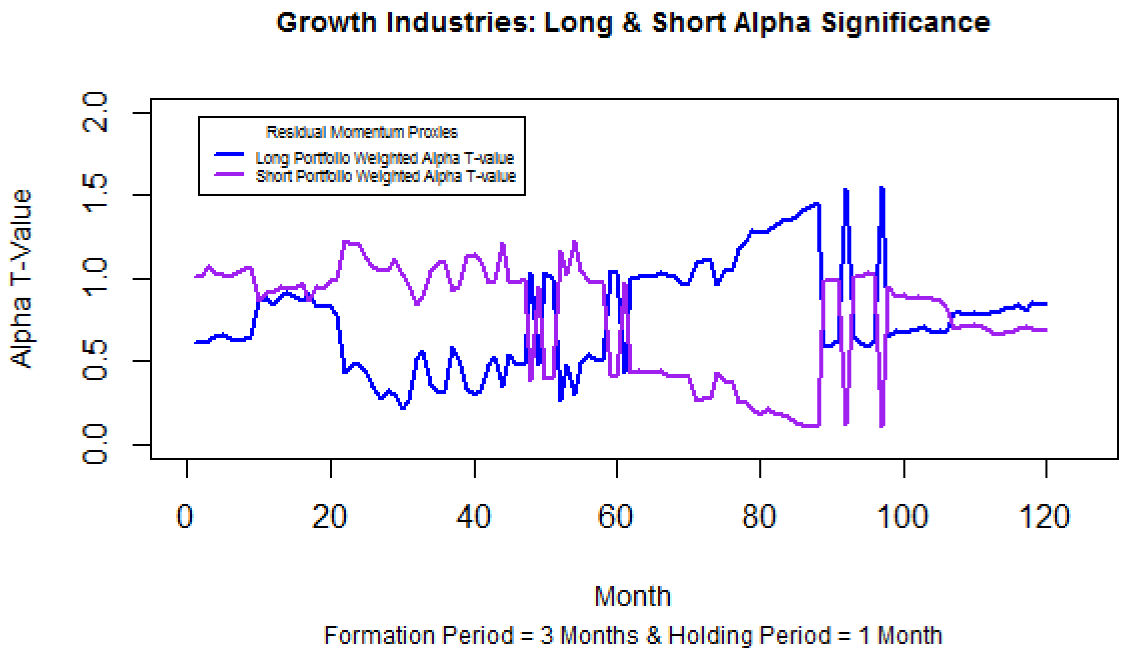

Figure 87

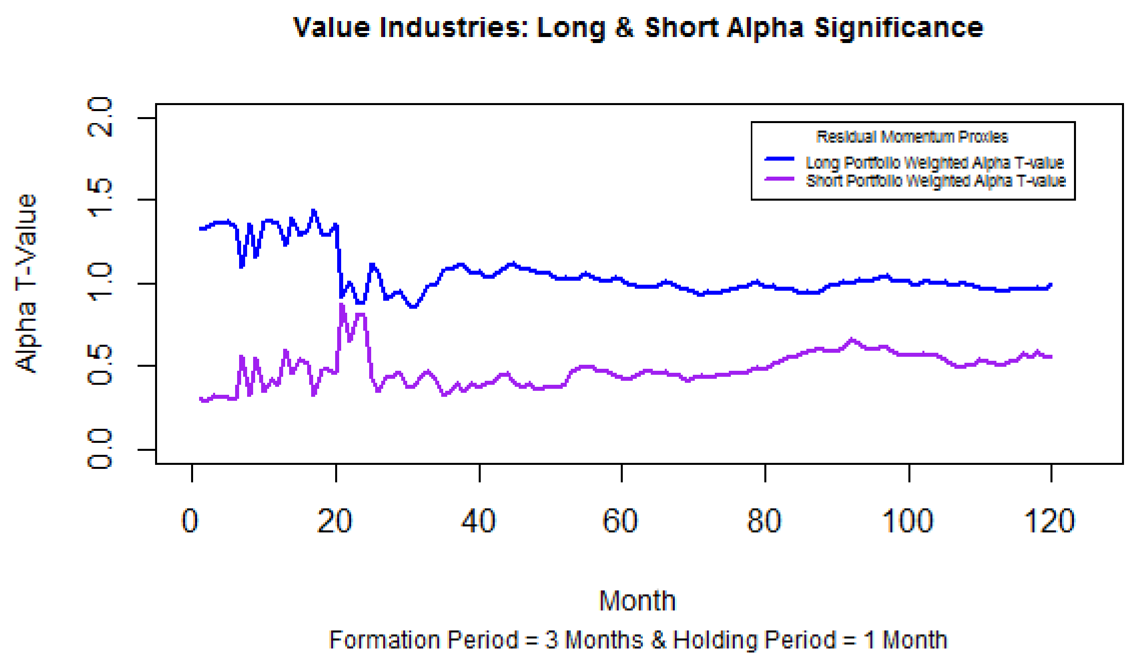

Figure 88

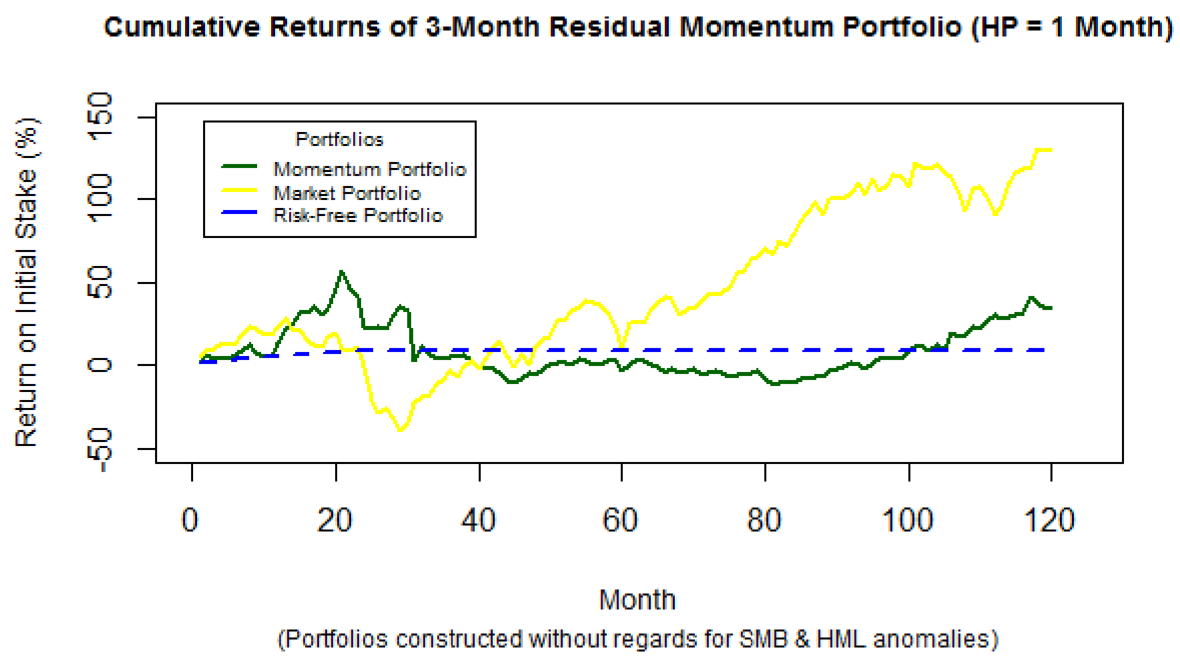

Figure 89

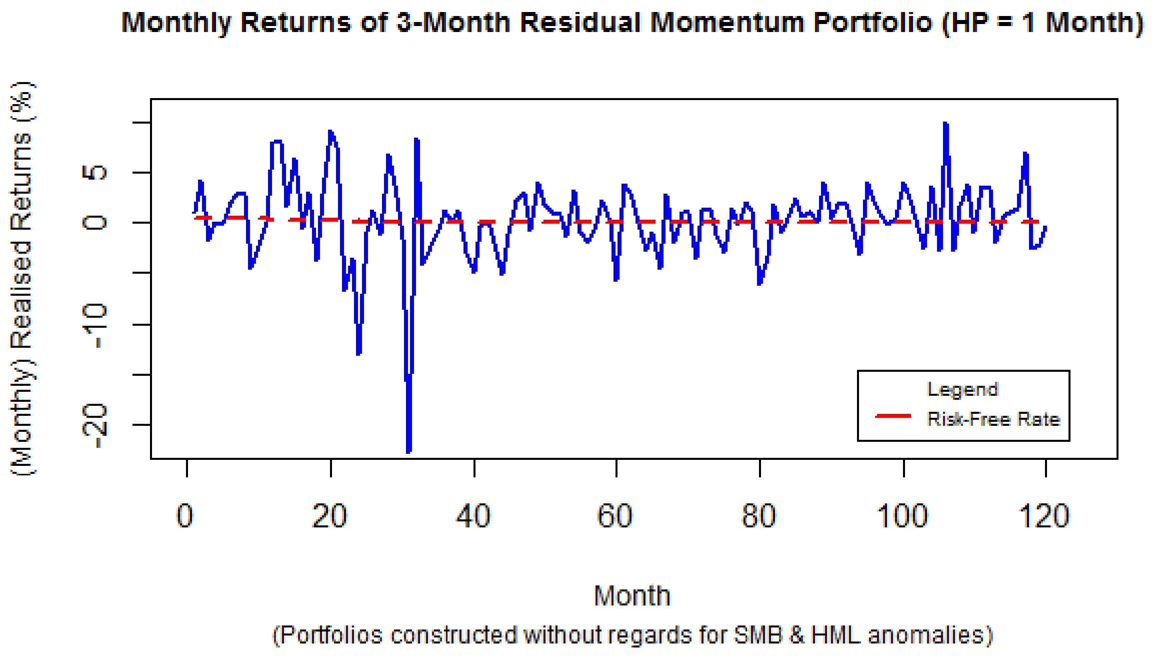

Figure 90

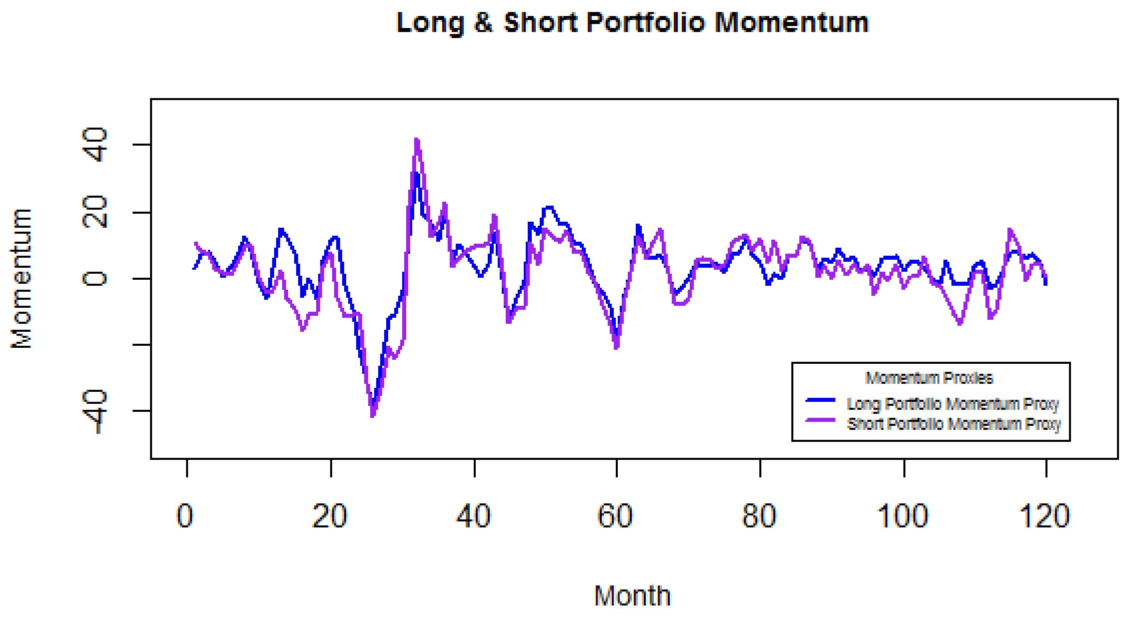

Figure 91

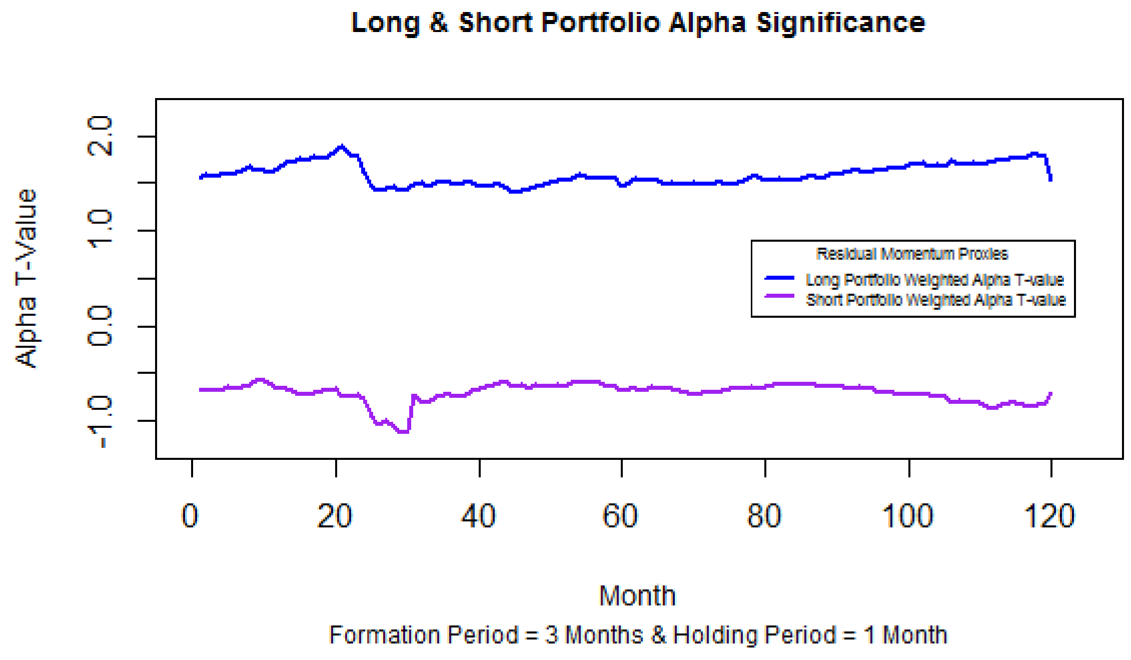

Figure 92

Note: All figures provided are in monthly percentage terms, unless otherwise stated.

Financial Ratio	LSVG MOM	Top 10 & Bottom 10 MOM	Market
Yearly Arithmetic Mean	6.874603	4.5176	10.3398
Yearly Geometric Mean	6.324467	6.324467	8.55684
Monthly Arithmetic Mean	0.5555879	0.3688895	0.8233248
Monthly Geometric Mean	0.5123517	0.5123517	0.6865436
Standard Deviation	2.95493	4.266925	5.203945
Skewness	-0.1429247	-1.343106	-0.5794711
Kurtosis	0.8009484	5.898942	2.45112
Covariance with the Market	-4.102318	-7.656894	27.08104
Pearson Correlation with the Market	-0.2667776	-0.34483	1
Spearman Correlation with the Market	-0.1578096	-0.2218418	1
Beta	-0.1475418	-0.2760597	-
Alpha from CAPM with intercept	0.6615542	0.5538225	0.09617862
Standard Error of Alpha	0.2648911	0.3765273	0.099604
P-value of Alpha	0.01388706	0.1439881	0.3362142
1 month 95% CL VaR (Non-Parametric)	4.660135	5.681141	8.010867
1 month 95% CL VaR (“Modified”)	4.355404	7.592723	8.265246
1 month 95% CL ETL Non-Parametric	6.428112	10.54753	11.93544
1 month 95% CL ETL (“Modified”)	6.120346	15.46477	13.73124
Semi-Standard Deviation	2.14327	3.343599	3.918679
Downside Deviation (MAR = long run risk-free rate)	2.061044	3.357626	3.715851
Downside Deviation (MAR = long run market risk premium)	2.173383	3.457975	3.816268
Jensen’s Alpha	0.2467523	0.1386954	-0.1877047

Sharpe Ratio	0.05295209	-0.007084416	0.08151637
Treynor's Measure	-1.060511	0.1095005	0.4242067
Sortino's Ratio (MAR = long run risk-free rate)	0.07591773	-0.00900299	0.1141614
Sortino's Ratio (MAR = long run market risk premium)	0.07199362	-0.008741725	0.1111575
Appraisal Ratio	0.08664645	0.03422481	-

Appendix B: Snippets of R-Code

Note: The intention here is to provide a brief look into how the workings outputted the results. As such there are many omissions.

Snippets of R-code used in this Project

```
# R Code for Financial Economics I Project

# Set working directory
setwd("C:\\Users\\Luke\\Documents\\FE1 Project\\")

# Input data
industrydata <- read.table("49_Industry_Portfolios.txt", header=TRUE)

# Define industry names
all_industries <- c(colnames(industrydata))
all_industries <- all_industries[-1]

# Read in FamaFrench data and merge
famafrench <- read.table("F-F_Research_Data_Factors.txt", header=TRUE)
alldata <- merge(industrydata, famafrench, by='Month')

# Calculate long run averages of rm-rf and rf
market.risk.premium <- function(no_months){
 rmrf <- mean(famafrench[0:843+no_months,'Mkt.RF'])
 rmrf
}

# Define ri - rf columns
all_industries_minus_rf <- paste(all_industries,'-RF', sep="")
alldata[, all_industries_minus_rf] <- alldata[, all_industries] - alldata[, 'RF']
alldata[,'Mkt'] <- alldata[,'Mkt.RF'] + alldata[,'RF']

# Choose Industries
industries <- c('Agric', 'Food', 'Soda', 'Toys', 'Fun','Drugs','Steel','Oil','Util','Softw')
industries_minus_rf <- paste(industries,'-RF', sep="")

# Keep final 240 months
alldata <- alldata[alldata$Month >= 199610 & alldata$Month <= 201609, ]

risk.free.rate <- function(no_months){
 rmrf <- alldata[no_months,'RF']
}

# Momentum
mom <- read.table("F-F_Momentum_Factor.txt", header=TRUE)
alldata <- merge(alldata, mom, by='Month')

# Find our chosen industries' Betas using linear regression
linear.regression.betas <- function(no_months){
 insample <- alldata[0:no_months,c(industries_minus_rf,'Mkt.RF')]
 Betas <- data.frame(matrix(nrow=10,ncol=1))
 for (i in 1:10){
  assign(paste("lin_mod",i,sep=""),
       lm(insample[,industries_minus_rf[i]] ~ 0+insample[,'Mkt.RF']))
  # 0 + forces intercept to be zero
```

```
    Betas[i,1] = eval(parse(text=paste("lin_mod",i,"$coefficients[[",1,"]]",sep="")))
    # coeffcient 1 is slope since intercept was removed
    i = i+1
  }
  Betas
}



# Find our chosen industries' Betas using cov(ri,rm)/var(rm)
industry_market.cov <- function(no_months){
  C = data.frame(matrix(nrow=10,ncol=1))
  insample = alldata[0:no_months,c(industries,'Mkt')]
  mean_rm = mean(insample[,'Mkt'])

  for (i in 1:10){
    temp_sum = 0
    mean_ri = mean(insample[,industries[i]])
    for (j in 1:no_months){
      temp_sum = temp_sum + (insample[j,industries[i]]-mean_ri)*
        (insample[j,'Mkt']- mean_rm)
      j = j+1
    }
    C[i,1] = temp_sum/(no_months-1)
    i = i+1
  }
  C
}

# Find covariance matrix V of chosen industries
covariance.all <- function(no_months){
  V = data.frame(matrix(nrow=10,ncol=10))
  means = data.frame(matrix(nrow=10,ncol=1))
  insample = alldata[0:no_months,all_industries]

  for (g in 1:49){
    means[g,1] = mean(insample[,g])
    g = g+1
  }

  for (i in 1:49){
    for (k in 1:49){
      temp_sum = 0
      for (j in 1:no_months){
        temp_sum = temp_sum + (insample[j,i]-means[i,1])*(insample[j,k]- means[k,1])
        j = j+1
      }
      V[i,k] = temp_sum/(no_months-1)
      k = k+1
    }
    i = i+1
  }
  V
}

# Define expected return of chosen industries = r-bar at time t+1
expected.return.chosen <- function(Betas,no_months){
  rbar = data.frame(matrix(nrow=10,ncol=1))
  rm_rf = market.risk.premium(no_months)
  rf = risk.free.rate(no_months)
  for (i in 1:10){
    rbar[i,1] = rf + Betas[i,1]*rm_rf
  }
  rbar
}

# Define unit vector
unitvector <- matrix(rep(1, 10), ncol=1)

# Calculate GMVP weights at month t+1
calculate.gmvp.weights <- function(V){
  # A = V^-1 * 1
  # B = 1^t * V^-1 * 1
  A <- solve(V) %*% unitvector
  B <- t(unitvector) %*% solve(V) %*% unitvector
  W <- A/B[1,1]
  W
}

# Check weights sum to 1

# Calculate Tangent weights
calculate.tangent.weights <- function(V,rbar,no_months){
  rf = alldata[no_months,'RF']
  rbar_minus_1rf = rbar-unitvector%*%rf
  A <<- solve(V)%*%as.matrix(rbar_minus_1rf)
  B <<- t(unitvector)%*%solve(V)%*%as.matrix(rbar_minus_1rf)
  W = A/B[1,1]
  W
}

# Calculate Betas function
calculate.betas.pooled.estimate <-  function(no_months){
  Betas_1 = linear.regression.betas(no_months)
  sigma_im = industry_market.cov(no_months)
  sigma_mm = var(alldata[0:no_months,'Mkt'])
  Betas_2 = sigma_im/sigma_mm
  Betas = 0.5*(Betas_1+Betas_2)
  Betas
}


# Define unit vector for 49 industries
unitvector_all <- matrix(rep(1, 49), ncol=1)

# Find Efficient Frontier after 240 months
no_months = 120

All_Betas_120 = calculate.allbetas.pooled.estimate(no_months)
rbar_all_120 = as.matrix(expected.return.all(All_Betas_120,no_months))
w_market_120 = as.matrix(unitvector_all*(1/49))
rbar_market_120 = t(w_market_120)%*%rbar_all_120
V_all_120 = as.matrix(covariance.all(120))
sigma_market_120 = sqrt(t(w_market_120)%*%V_all_120%*%w_market_120)

Betas_120 = as.matrix(calculate.betas.pooled.estimate(no_months))
V_120 = as.matrix(covariance(no_months))
rbar_120 = as.matrix(expected.return.chosen(Betas_120,no_months))
w_gmvp_120 = as.matrix(calculate.gmvp.weights(V_120))
w_tangent_120 = as.matrix(calculate.tangent.weights(V_120,rbar_120,no_months))
w_naive_120 = data.frame(matrix(nrow=10,ncol=1))
w_naive_120 = as.matrix(unitvector*0.1)
rbar_gmvp_120 = t(w_gmvp_120)%*%rbar_120
rbar_tangent_120 = t(w_tangent_120)%*%rbar_120
rbar_naive_120 = t(w_naive_120)%*%rbar_120
sigma_gmvp_120 = sqrt(t(w_gmvp_120)%*%V_120%*%w_gmvp_120)
sigma_tangent_120 = sqrt(t(w_tangent_120)%*%V_120%*%w_tangent_120)
sigma_naive_120 = sqrt(t(w_naive_120)%*%V_120%*%w_naive_120)
cov_gmvp_tangent_120 = t(w_gmvp_120)%*%V_120%*%w_tangent_120
corr_gmvp_tangent_120 = cov_gmvp_tangent_120/(sigma_gmvp_120*sigma_tangent_120)

# Using Two Fund Theorem
alpha = data.frame(matrix(seq(-5,5,by=0.01),ncol=1))
mv_frontier_rbar_120 = data.frame(matrix(nrow=1001,ncol=1))
mv_frontier_sigma_120 = data.frame(matrix(nrow=1001,ncol=1))
for (i in 1:1001){
  mv_frontier_rbar_120[i,1] = alpha[i,1]*rbar_gmvp_120+(1-alpha[i,1])*rbar_tangent_120
  mv_frontier_sigma_120[i,1] = sqrt((alpha[i,1]^2)*sigma_gmvp_120^2+
                                  ((1-alpha[i,1])^2)*sigma_tangent_120^2+2*alpha[i,1]*(1-alpha[i,1])*cov_gmvp_tangent_120)
}

# Using One Fund Theorem
```

```
rf_120 = alldata[no_months,'RF']
K = data.frame(matrix(nrow=10,ncol=1))
K = rbar_120- unitvector*rf_120
H = t(K)%*%solve(V_120)%*%K
cal_rbar_120 = data.frame(matrix(nrow=801,ncol=1))
cal_sigma_120 = data.frame(matrix(seq(0,8,by=0.01),ncol=1))
for (i in 1:801){
  cal_rbar_120[i,1] = sqrt(H)*cal_sigma_120[i,1]+rf_120
}

# Monthly Plot of Efficient Frontier after 120 months
t = 1:1001
plot(x = mv_frontier_sigma_120[t,1], y = mv_frontier_rbar_120[t,1],
     main='Mean-Variance Frontier of 120-Month Insample Data',
     xlab='(Monthly) Standard Deviation of Portfolio (%)', ylab='Expected (Monthly) Return of Portfolio (%)',
     type='l',xlim=c(3,6),ylim=c(-0.3,1.5),lwd=2,col='blue',cex.lab=0.9)
lines(x=cal_sigma_120[t,1],y=cal_rbar_120[t,1],lwd=2,lty=2,col="red")
points(x=sigma_gmvp_120,y=rbar_gmvp_120,col='black',pch=8)
points(x=sigma_tangent_120,y=rbar_tangent_120,col='purple',pch=8)
points(x=sigma_naive_120,y=rbar_naive_120,col='darkgreen',pch=8)
points(x=sigma_market_120,y=rbar_market_120,col='yellow',pch=8)
labels <- c("Mean-Variance Frontier","Capital Allocation Line","Global Minimum-Variance Portfolio",
          "Tangent Portfolio","Naive-Weighted Portfolio","Market Portfolio")
legend("bottomright", inset=.05, title="Legend",labels,cex=0.6,lwd=c(2,2,NA,NA,NA,NA),lty=c(1,1,NA,NA,NA,NA),
       pch=c(NA,NA,8,8,8,8), col=c("blue","red","black","purple","darkgreen","yellow"))

results <- data.frame(matrix(nrow=121, ncol=23))
colnames(results) <- c('Month','rbar_gmvp', 'rbar_tangent', 'rbar_naive',
                    'r_gmvp', 'r_tangent', 'r_naive','sigma_gmvp', 'sigma_tangent',
                    'sigma_naive', 'cov_gmvp_tangent',
                    'cumr_gmvp','cumr_tangent','cumr_naive'
                    ,'r_market','cumr_market','r_rf','cumr_rf'
                    ,'Mkt.RF','RF','SMB','HML','MOM')
results[1,'r_gmvp']=1
results[1,'r_tangent']=1
results[1,'r_naive']=1
results[1,'r_market']=1
results[1,'r_rf']=1
results[1,'cumr_gmvp']=1
results[1,'cumr_tangent']=1
results[1,'cumr_naive']=1
results[1,'cumr_market']=1
results[1,'cumr_rf']=1

# everything but actual and cumulative returns prospective
# and Mkt.RF, RF, HML and SMB
# ie rbar(2) is for 121st month (based on data up to and includ. 120th month)
# and r(2) is the return of 121st month
# Mkt.RF(2) is for MKt.RF of 121st month

for(i in 1:120){
  no_months = i+120
  Betas = as.matrix(calculate.betas.pooled.estimate(no_months-1))
  V = as.matrix(covariance(no_months-1))
  rbar = as.matrix(expected.return.chosen(Betas,no_months-1))
  w_gmvp = as.matrix(calculate.gmvp.weights(V))
  w_tangent = as.matrix(calculate.tangent.weights(V,rbar,no_months-1))
  w_naive = unitvector/10
  w_market = unitvector_all/49
  array_entry = no_months - 119

  results[array_entry,'Mkt.RF'] = alldata[no_months,'Mkt.RF']
  results[array_entry,'RF'] = alldata[no_months,'RF']
  results[array_entry,'SMB'] = alldata[no_months,'SMB']
  results[array_entry,'HML'] = alldata[no_months,'HML']
  results[array_entry,'MOM'] = alldata[no_months,'Mom']

  results[array_entry,'r_gmvp'] = as.matrix(alldata[no_months,industries])%*%w_gmvp
  results[array_entry,'r_tangent'] = as.matrix(alldata[no_months,industries])%*%w_tangent
  results[array_entry,'r_naive'] = as.matrix(alldata[no_months,industries])%*%w_naive
  results[array_entry,'r_market'] = as.matrix(alldata[no_months,all_industries])%*%w_market
  results[array_entry,'r_rf'] = alldata[no_months,'RF']
  results[array_entry,'rbar_gmvp'] = t(w_gmvp)%*%rbar
  results[array_entry,'rbar_tangent'] = t(w_tangent)%*%rbar
  results[array_entry,'rbar_naive'] = t(w_naive)%*%rbar
  results[array_entry,'sigma_gmvp'] = sqrt(t(w_gmvp)%*%V%*%w_gmvp)
  results[array_entry,'sigma_tangent'] = sqrt(t(w_tangent)%*%V%*%w_tangent)
  results[array_entry,'sigma_naive'] = sqrt(t(w_naive)%*%V%*%w_naive)
  results[array_entry,'cov_gmvp_tangent'] = t(w_gmvp)%*%V%*%w_tangent
  results[array_entry,'cumr_gmvp'] = ((1+results[array_entry,'r_gmvp']/100)*(1+results[array_entry-1,'cumr_gmvp']/100)-1)*100
  results[array_entry,'cumr_tangent'] = ((1+results[array_entry,'r_tangent']/100)*(1+results[array_entry-1,'cumr_tangent']/100)-1)*100
  results[array_entry,'cumr_naive'] = ((1+results[array_entry,'r_naive']/100)*(1+results[array_entry-1,'cumr_naive']/100)-1)*100
  results[array_entry,'cumr_market'] = ((1+results[array_entry,'r_market']/100)*(1+results[array_entry-1,'cumr_market']/100)-1)*100
  if (results[array_entry,'r_rf'] == 0) {
    results[array_entry,'cumr_rf'] = results[array_entry-1,'cumr_rf']
  }
  else {
    results[array_entry,'cumr_rf'] = ((1+results[array_entry,'r_rf']/100)*(1+results[array_entry-1,'cumr_rf']/100)-1)*100
  }
  i = i+1
}

# Evaluate performance of each portfolio using CAPM
CAPM_gmvp = lm(results[2:121,'r_gmvp']~results[2:121,'Mkt.RF'])
CAPM_tangent = lm(results[2:121,'r_tangent']~results[2:121,'Mkt.RF'])
CAPM_naive = lm(results[2:121,'r_naive']~results[2:121,'Mkt.RF'])
CAPM_market = lm(results[2:121,'r_market']~results[2:121,'Mkt.RF'])
CAPM_rf = lm(results[2:121,'r_rf']~results[2:121,'Mkt.RF'])

# Evaluate performance of each portfolio using Fama-French Model
FF_gmvp = lm(results[2:121,'r_gmvp']~results[2:121,'Mkt.RF']+results[2:121,'SMB']+results[2:121,'HML'])
FF_tangent = lm(results[2:121,'r_tangent']~results[2:121,'Mkt.RF']+results[2:121,'SMB']+results[2:121,'HML'])
FF_naive = lm(results[2:121,'r_naive']~results[2:121,'Mkt.RF']+results[2:121,'SMB']+results[2:121,'HML'])
FF_market = lm(results[2:121,'r_market']~results[2:121,'Mkt.RF']+results[2:121,'SMB']+results[2:121,'HML'])
FF_rf = lm(results[2:121,'r_rf']~results[2:121,'Mkt.RF']+results[2:121,'SMB']+results[2:121,'HML'])

# Evaluate performance of each portfolio using Cahart Model
Cahart_gmvp = lm(results[2:121,'r_gmvp']~results[2:121,'Mkt.RF']+results[2:121,'SMB']+results[2:121,'HML']+results[2:121,'MOM'])
Cahart_tangent = lm(results[2:121,'r_tangent']~results[2:121,'Mkt.RF']+results[2:121,'SMB']+results[2:121,'HML']+results[2:121,'MOM'])
Cahart_naive = lm(results[2:121,'r_naive']~results[2:121,'Mkt.RF']+results[2:121,'SMB']+results[2:121,'HML']+results[2:121,'MOM'])
Cahart_market = lm(results[2:121,'r_market']~results[2:121,'Mkt.RF']+results[2:121,'SMB']+results[2:121,'HML']+results[2:121,'MOM'])
Cahart_rf = lm(results[2:121,'r_rf']~results[2:121,'Mkt.RF']+results[2:121,'SMB']+results[2:121,'HML']+results[2:121,'MOM'])

summary(CAPM_gmvp)
t = 1:120
plot(y=results[t+1,'r_gmvp'],x=results[t+1,'Mkt.RF'],
     main='Plot of GMVP Returns vs Capital Asset Pricing Model',
     xlab='Market Risk Premium', ylab='Excess Returns of Portfolio over Risk-Free Rate',
     xlim=c(-10,10),ylim=c(-10,8),lwd=1,col='blue',cex.lab=0.9)
legend("bottomright", inset=.05, title="Legend","CAPM",cex=0.6,lwd=2,lty=1, col="red")
abline(CAPM_gmvp,col="red",lwd=2,lty=2)

CAPM_gmvp_residuals = resid(CAPM_gmvp)
CAPM_gmvp_fitted_values = fitted(CAPM_gmvp)
CAPM_gmvp_stdresid = rstandard(CAPM_gmvp)
plot(y=CAPM_gmvp_residuals,x=CAPM_gmvp_fitted_values,
     main="Residuals vs Fitted Values of GMVP Returns using CAPM",
     xlab='Fitted Values', ylab='Residuals',
     xlim=c(-5,7),ylim=c(-7,5),lwd=1,col='blue',cex.lab=0.9)
abline(0,0,col="red",lwd=2,lty=2)
qqnorm(CAPM_gmvp_stdresid,main="QQ-Plot of Residuals of GMVP Returns using CAPM",
       xlab='Theoretical Quartiles', ylab='Empirical Quartiles (Standardised Residuals)',
       xlim=c(-2.5,2.5),ylim=c(-3.9,3),lwd=1,col='blue',cex.lab=0.9)
qqline(CAPM_gmvp_stdresid,lwd=2,lty=1,col='red')

ks.test(CAPM_gmvp_residuals,pnorm,mean=0,sd=1)

################################################################################
# Betting Against Beta
# Choose Industries
large_industries = c('Other','Aero','Oil','Hardw','Fin','Food','Guns','Ships')
# large average firm
```

```
small_industries = c('Toys','Txtls','Rubbr','LabEq','ElcEq','BusSv','Whlsl','Cnstr')
# small average firm
growth_industries = c('MedEq','Drugs','Softw','Hlth','Mines','Meals','Agric','Soda')
# growth = low book-to-market ratio
value_industries = c('Autos','Steel','Banks','Trans','Gold','Coal','Books','Boxes')
# value = high book-to-market ratio

large_industries_minus_rf <- paste(large_industries,'-RF', sep="")
small_industries_minus_rf <- paste(small_industries,'-RF', sep="")
growth_industries_minus_rf <- paste(growth_industries,'-RF', sep="")
value_industries_minus_rf <- paste(value_industries,'-RF', sep="")


babresults <- data.frame(matrix(nrow=121, ncol=19))
colnames(babresults) <- c('Month','lreturnp', 'sreturnp', 'greturnp',
                         'vreturnp', 'rp', 'cumr_l','cumr_s', 'cumr_g',
                         'cumr_v', 'cum_rp','beta_ll','beta_ls',
                         'beta_sl','beta_ss','beta_gl','beta_gs',
                         'beta_vl','beta_vs')
babresults[1,'lreturnp']=1
babresults[1,'sreturnp']=1
babresults[1,'greturnp']=1
babresults[1,'vreturnp']=1
babresults[1,'rp']=1
babresults[1,'cumr_l']=1
babresults[1,'cumr_s']=1
babresults[1,'cumr_g']=1
babresults[1,'cumr_v']=1
babresults[1,'cum_rp']=1

# r(2) = return on 121st month
# rf(2) = risk free return on 120th month

for(i in 1:120){
  no_months = i+120
  rf = risk.free.rate(no_months-1)

  lbetas = calculate.betas.pooled.estimate.large(no_months-1)
  sbetas = calculate.betas.pooled.estimate.small(no_months-1)
  gbetas = calculate.betas.pooled.estimate.growth(no_months-1)
  vbetas = calculate.betas.pooled.estimate.value(no_months-1)
  lreturns = alldata[no_months,large_industries]
  sreturns = alldata[no_months,small_industries]
  greturns = alldata[no_months,growth_industries]
  vreturns = alldata[no_months,value_industries]

  ordered_lbetas = lbetas[order(lbetas),1]
  ordered_sbetas = lbetas[order(sbetas),1]
  ordered_gbetas = lbetas[order(gbetas),1]
  ordered_vbetas = lbetas[order(vbetas),1]
  ordered_lreturns = lreturns[order(lbetas)]
  ordered_sreturns = sreturns[order(sbetas)]
  ordered_greturns = greturns[order(gbetas)]
  ordered_vreturns = vreturns[order(vbetas)]

  llbetas = as.matrix(ordered_lbetas[1:4]) # long the lowest betas
  lsbetas = as.matrix(ordered_lbetas[5:8])
  slbetas = as.matrix(ordered_sbetas[1:4])
  ssbetas = as.matrix(ordered_sbetas[5:8])
  glbetas = as.matrix(ordered_gbetas[1:4])
  gsbetas = as.matrix(ordered_gbetas[5:8])
  vlbetas = as.matrix(ordered_vbetas[1:4])
  vsbetas = as.matrix(ordered_vbetas[5:8])
  llreturns = as.matrix(ordered_lreturns[1:4]) # returns for lowest betas
  lsreturns = as.matrix(ordered_lreturns[5:8])
  slreturns = as.matrix(ordered_sreturns[1:4])
  ssreturns = as.matrix(ordered_sreturns[5:8])
  glreturns = as.matrix(ordered_greturns[1:4])
  gsreturns = as.matrix(ordered_greturns[5:8])
  vlreturns = as.matrix(ordered_vreturns[1:4])
  vsreturns = as.matrix(ordered_vreturns[5:8])
  for (i in 1:4){
    llweights[i,1] = 0.5-llbetas[i,1]/sum(llbetas[,1])
    slweights[i,1] = 0.5-slbetas[i,1]/sum(slbetas[,1])
    glweights[i,1] = 0.5-glbetas[i,1]/sum(glbetas[,1])
    vlweights[i,1] = 0.5-vlbetas[i,1]/sum(vlbetas[,1])
    lsweights[i,1] = lsbetas[i,1]/sum(lsbetas[,1])
    ssweights[i,1] = ssbetas[i,1]/sum(ssbetas[,1])
    gsweights[i,1] = gsbetas[i,1]/sum(gsbetas[,1])
    vsweights[i,1] = vsbetas[i,1]/sum(vsbetas[,1])
  }

  # check sum to 1

  llbetap = t(llweights)%*%llbetas
  lsbetap = t(lsweights)%*%lsbetas
  slbetap = t(slweights)%*%slbetas
  ssbetap = t(ssweights)%*%ssbetas
  glbetap = t(glweights)%*%glbetas
  gsbetap = t(gsweights)%*%gsbetas
  vlbetap = t(vlweights)%*%vlbetas
  vsbetap = t(vsweights)%*%vsbetas
  llreturnp = as.matrix(llreturns)%*%as.matrix(llweights)
  lsreturnp = as.matrix(lsreturns)%*%as.matrix(lsweights)
  slreturnp = as.matrix(slreturns)%*%as.matrix(slweights)
  ssreturnp = as.matrix(ssreturns)%*%as.matrix(ssweights)
  glreturnp = as.matrix(glreturns)%*%as.matrix(glweights)
  gsreturnp = as.matrix(gsreturns)%*%as.matrix(gsweights)
  vlreturnp = as.matrix(vlreturns)%*%as.matrix(vlweights)
  vsreturnp = as.matrix(vsreturns)%*%as.matrix(vsweights)


  lreturnp = (1/llbetap)*(llreturnp - rf)+(-1/lsbetap)*(lsreturnp - rf)
  sreturnp = (1/slbetap)*(slreturnp - rf)+(-1/ssbetap)*(ssreturnp - rf)
  greturnp = (1/glbetap)*(glreturnp - rf)+(-1/gsbetap)*(gsreturnp - rf)
  vreturnp = (1/vlbetap)*(vlreturnp - rf)+(-1/vsbetap)*(vsreturnp - rf)

  rp = 0.25*(lreturnp+sreturnp+greturnp+vreturnp)
  array_entry = no_months - 119

  babresults[array_entry,'lreturnp']=lreturnp
  babresults[array_entry,'sreturnp']=sreturnp
  babresults[array_entry,'greturnp']=greturnp
  babresults[array_entry,'vreturnp']=vreturnp
  babresults[array_entry,'rp']=rp
  babresults[array_entry,'beta_ll']= llbetap
  babresults[array_entry,'beta_ls']= lsbetap
  babresults[array_entry,'beta_sl']= slbetap
  babresults[array_entry,'beta_ss']= ssbetap
  babresults[array_entry,'beta_gl']= glbetap
  babresults[array_entry,'beta_gs']= gsbetap
  babresults[array_entry,'beta_vl']= vlbetap
  babresults[array_entry,'beta_vs']= vsbetap
  babresults[array_entry,'cumr_l']=((1+babresults[array_entry,'lreturnp']/100)*(1+babresults[array_entry-1,'cumr_l']/100)-1)*100
  babresults[array_entry,'cumr_s']=((1+babresults[array_entry,'sreturnp']/100)*(1+babresults[array_entry-1,'cumr_s']/100)-1)*100
  babresults[array_entry,'cumr_g']=((1+babresults[array_entry,'greturnp']/100)*(1+babresults[array_entry-1,'cumr_g']/100)-1)*100
  babresults[array_entry,'cumr_v']=((1+babresults[array_entry,'vreturnp']/100)*(1+babresults[array_entry-1,'cumr_v']/100)-1)*100
  babresults[array_entry,'cum_rp']=((1+babresults[array_entry,'rp']/100)*(1+babresults[array_entry-1,'cum_rp']/100)-1)*100
  i = i+1
}

#####################################################
# Betting Against Beta: Ignore other anomalies
lbetas = data.frame(matrix(nrow=10,ncol=1))
sbetas = data.frame(matrix(nrow=10,ncol=1))
lweights = data.frame(matrix(nrow=10,ncol=1))
sweights = data.frame(matrix(nrow=10,ncol=1))
lreturns = data.frame(matrix(nrow=10,ncol=1))
sreturns = data.frame(matrix(nrow=10,ncol=1))
```

```
babresults2 <- data.frame(matrix(nrow=121, ncol=7))
colnames(babresults2) <- c('Month','lrp', 'srp', 'rp','cum_rp','beta_l','beta_h')
babresults2[1,'lrp']=1
babresults2[1,'srp']=1
babresults2[1,'rp']=1
babresults2[1,'cum_rp']=1

# r(2) = return on 121st month
# rf(2) = risk free return on 120th month

for(i in 1:120){
  no_months = i+120
  rf = risk.free.rate(no_months-1)

  all_betas = calculate.allbetas.pooled.estimate(no_months-1)
  returns = alldata[no_months,all_industries]
  ordered_betas = all_betas[order(all_betas),1]
  ordered_returns = returns[order(all_betas)]

  lbetas = as.matrix(ordered_betas[1:10]) # long the lowest betas
  sbetas = as.matrix(ordered_betas[40:49])
  lreturns = as.matrix(ordered_returns[1:10]) # returns for lowest betas
  sreturns = as.matrix(ordered_returns[40:49])

  for (i in 1:10){
    lweights[i,1] = 0.2-lbetas[i,1]/sum(lbetas[,1])
    sweights[i,1] = lbetas[i,1]/sum(lbetas[,1])
  }

  # check sum to 1

  lbetap = t(lweights)%*%lbetas
  sbetap = t(sweights)%*%sbetas
  lrp = as.matrix(lreturns)%*%as.matrix(lweights)
  srp = as.matrix(sreturns)%*%as.matrix(sweights)

  rp = (1/lbetap)*(lrp - rf)+(-1/sbetap)*(srp - rf)

  array_entry = no_months - 119

  babresults2[array_entry,'beta_l']=lbetap
  babresults2[array_entry,'beta_s']=sbetap
  babresults2[array_entry,'lrp']=lrp
  babresults2[array_entry,'srp']=srp
  babresults2[array_entry,'rp']=rp
  babresults2[array_entry,'cum_rp']=((1+babresults2[array_entry,'rp']/100)*(1+babresults2[array_entry-1,'cum_rp']/100)-1)*100
  i = i+1
}

# Input TED Spread
tedspread <- read.table("TED Spread.txt", header=TRUE)

babreturns = babresults[2:121,'rp'] - results[2:121,'RF']
bab2returns = babresults2[2:121,'rp'] - results[2:121,'RF']

lin_model_bab_mrp_ted = lm(babreturns~results[2:121,'Mkt.RF']+tedspread[121:240,'TED'])
lin_model_bab_ted = lm(babreturns~tedspread[121:240,'TED'])
lin_model_bab2_mrp_ted = lm(bab2returns~results[2:121,'Mkt.RF']+tedspread[121:240,'TED'])
lin_model_bab2_ted = lm(bab2returns~tedspread[121:240,'TED'])

# Input Daily Industry Returns
industrydaily <- read.table("49_Industry_Portfolios_Daily.txt", header=TRUE)

# Convert to date
industrydaily$Date <- as.Date(industrydaily$Date,origin="1960-10-01")

#  Get months
industrydaily$Month <- months(industrydaily$Date)

#  Get years
industrydaily$Year <- format(industrydaily$Date,format="%y")
```

```
# Keep final 240 months
industrydaily <- industrydaily[industrydaily$Year >= 96 | industrydaily$Year <= 16, ]

years = unique(industrydaily$Year)
months = unique(industrydaily$Month)
MAX_matrix = data.frame(matrix(nrow=21,ncol=12))
period = 12
for (i in 1:21){
  if (i==21){ period = 9}
  for(j in 1:period){
    data1 = industrydaily[industrydaily$Year == years[i],]
    data2 = data1[data1$Month == months[j],]
    n = nrow(data2)
    allreturns = as.vector(as.matrix(data2[,all_industries]))
    MAX_matrix[i,j] = mean(tail(sort(allreturns,decreasing=FALSE),5))
    j = j+1
  }
  i = i+1
}
MAX_vector = as.vector(t(MAX_matrix))
MAX = MAX_vector[9:248] # lagged 1 month

lin_model_bab_mrp_max = lm(babreturns~results[2:121,'Mkt.RF']+MAX[121:240])
lin_model_bab_max = lm(babreturns~MAX[121:240])
lin_model_bab2_mrp_max = lm(bab2returns~results[2:121,'Mkt.RF']+MAX[121:240])
lin_model_bab2_max = lm(bab2returns~MAX[121:240])

################################################################################
# Momentum Portfolio (12 months)

momentum <- function(no_months,industr,period){
  geometric_returns = data.frame(matrix(nrow=8,ncol=1))
  for (i in 1:8){
    temp_sum = 1
    for(j in 0:period){
      temp_sum = (1+alldata[no_months-j,industr[i]]/100)*temp_sum
      j = j+1
    }
    geometric_returns[i,1] = (temp_sum - 1)*100
    i = i+1
  }
  geometric_returns
}

momresults_12m <- data.frame(matrix(nrow=121, ncol=19))
colnames(momresults_12m) <- c('Month','lreturnp', 'sreturnp', 'greturnp',
                              'vreturnp', 'rp', 'cumr_l','cumr_s', 'cumr_g',
                              'cumr_v', 'cum_rp','mom_ll','mom_ls',
                              'mom_sl','mom_ss','mom_gl','mom_gs',
                              'mom_vl','mom_vs')
momresults_12m[1,'lreturnp']=1
momresults_12m[1,'sreturnp']=1
momresults_12m[1,'greturnp']=1
momresults_12m[1,'vreturnp']=1
momresults_12m[1,'rp']=1
momresults_12m[1,'cumr_l']=1
momresults_12m[1,'cumr_s']=1
momresults_12m[1,'cumr_g']=1
momresults_12m[1,'cumr_v']=1
momresults_12m[1,'cum_rp']=1

# r(2) = return on 121st month
# rf(2) = risk free return on 120th month

for(i in 1:120){
  no_months = i+120
  rf = risk.free.rate(no_months-1)

  lbetas = calculate.betas.pooled.estimate.large(no_months-1)
  sbetas = calculate.betas.pooled.estimate.small(no_months-1)
```

```
gbetas = calculate.betas.pooled.estimate.growth(no_months-1)
vbetas = calculate.betas.pooled.estimate.value(no_months-1)
lreturns = alldata[no_months,large_industries]
sreturns = alldata[no_months,small_industries]
greturns = alldata[no_months,growth_industries]
vreturns = alldata[no_months,value_industries]
lmomentum = momentum(no_months-1,large_industries_minus_rf,11)
smomentum = momentum(no_months-1,small_industries_minus_rf,11)
gmomentum = momentum(no_months-1,growth_industries_minus_rf,11)
vmomentum = momentum(no_months-1,value_industries_minus_rf,11)

ordered_lbetas = lbetas[order(lmomentum),1]
ordered_sbetas = lbetas[order(smomentum),1]
ordered_gbetas = lbetas[order(gmomentum),1]
ordered_vbetas = lbetas[order(vmomentum),1]
ordered_lreturns = lreturns[order(lmomentum)]
ordered_sreturns = sreturns[order(smomentum)]
ordered_greturns = greturns[order(gmomentum)]
ordered_vreturns = vreturns[order(vmomentum)]
ordered_lmomentum = lmomentum[order(lmomentum),1]
ordered_smomentum = smomentum[order(smomentum),1]
ordered_gmomentum = gmomentum[order(gmomentum),1]
ordered_vmomentum = vmomentum[order(vmomentum),1]

llmomentum = as.matrix(ordered_lmomentum[5:8]) # long the highest momentum
lsmomentum = as.matrix(ordered_lmomentum[1:4])
slmomentum = as.matrix(ordered_smomentum[5:8])
ssmomentum = as.matrix(ordered_smomentum[1:4])
glmomentum = as.matrix(ordered_gmomentum[5:8])
gsmomentum = as.matrix(ordered_gmomentum[1:4])
vlmomentum = as.matrix(ordered_vmomentum[5:8])
vsmomentum = as.matrix(ordered_vmomentum[1:4])
llbetas = as.matrix(ordered_lbetas[5:8]) # long the highest momentum
lsbetas = as.matrix(ordered_lbetas[1:4])
slbetas = as.matrix(ordered_sbetas[5:8])
ssbetas = as.matrix(ordered_sbetas[1:4])
glbetas = as.matrix(ordered_gbetas[5:8])
gsbetas = as.matrix(ordered_gbetas[1:4])
vlbetas = as.matrix(ordered_vbetas[5:8])
vsbetas = as.matrix(ordered_vbetas[1:4])
llreturns = as.matrix(ordered_lreturns[5:8]) # returns for highest momentum
lsreturns = as.matrix(ordered_lreturns[1:4])
slreturns = as.matrix(ordered_sreturns[5:8])
ssreturns = as.matrix(ordered_sreturns[1:4])
glreturns = as.matrix(ordered_greturns[5:8])
gsreturns = as.matrix(ordered_greturns[1:4])
vlreturns = as.matrix(ordered_vreturns[5:8])
vsreturns = as.matrix(ordered_vreturns[1:4])

for (i in 1:4){
  llweights[i,1] = 0.25 # llmomentum[i,1]/sum(llmomentum[,1])
  slweights[i,1] = 0.25 # slmomentum[i,1]/sum(slmomentum[,1])
  glweights[i,1] = 0.25 # glmomentum[i,1]/sum(glmomentum[,1])
  vlweights[i,1] = 0.25 # vlmomentum[i,1]/sum(vlmomentum[,1])
  lsweights[i,1] = 0.25 # 0.5-lsmomentum[i,1]/sum(lsmomentum[,1])
  ssweights[i,1] = 0.25 # 0.5-ssmomentum[i,1]/sum(ssmomentum[,1])
  gsweights[i,1] = 0.25 # 0.5-gsmomentum[i,1]/sum(gsmomentum[,1])
  vsweights[i,1] = 0.25 # 0.5-vsmomentum[i,1]/sum(vsmomentum[,1])
}

# check sum to 1

llbetap = t(llweights)%*%llbetas
lsbetap = t(lsweights)%*%lsbetas
slbetap = t(slweights)%*%slbetas
ssbetap = t(ssweights)%*%ssbetas
glbetap = t(glweights)%*%glbetas
gsbetap = t(gsweights)%*%gsbetas
vlbetap = t(vlweights)%*%vlbetas
vsbetap = t(vsweights)%*%vsbetas
llreturnp = as.matrix(llreturns)%*%as.matrix(llweights)
```

```
  lsreturnp = as.matrix(lsreturns)%*%as.matrix(lsweights)
  slreturnp = as.matrix(slreturns)%*%as.matrix(slweights)
  ssreturnp = as.matrix(ssreturns)%*%as.matrix(ssweights)
  glreturnp = as.matrix(glreturns)%*%as.matrix(glweights)
  gsreturnp = as.matrix(gsreturns)%*%as.matrix(gsweights)
  vlreturnp = as.matrix(vlreturns)%*%as.matrix(vlweights)
  vsreturnp = as.matrix(vsreturns)%*%as.matrix(vsweights)
  llmomentump = t(llweights)%*%llmomentum
  lsmomentump = t(lsweights)%*%lsmomentum
  slmomentump = t(slweights)%*%slmomentum
  ssmomentump = t(ssweights)%*%ssmomentum
  glmomentump = t(glweights)%*%glmomentum
  gsmomentump = t(gsweights)%*%gsmomentum
  vlmomentump = t(vlweights)%*%vlmomentum
  vsmomentump = t(vsweights)%*%vsmomentum

  lreturnp = (1/llbetap)*(llreturnp - rf)+(-1/lsbetap)*(lsreturnp - rf)
  sreturnp = (1/slbetap)*(slreturnp - rf)+(-1/ssbetap)*(ssreturnp - rf)
  greturnp = (1/glbetap)*(glreturnp - rf)+(-1/gsbetap)*(gsreturnp - rf)
  vreturnp = (1/vlbetap)*(vlreturnp - rf)+(-1/vsbetap)*(vsreturnp - rf)

  rp = 0.25*(lreturnp+sreturnp+greturnp+vreturnp)
  array_entry = no_months - 119

  momresults_12m[array_entry,'mom_ll']= llmomentump
  momresults_12m[array_entry,'mom_ls']= lsmomentump
  momresults_12m[array_entry,'mom_sl']= slmomentump
  momresults_12m[array_entry,'mom_ss']= ssmomentump
  momresults_12m[array_entry,'mom_gl']= glmomentump
  momresults_12m[array_entry,'mom_gs']= gsmomentump
  momresults_12m[array_entry,'mom_vl']= vlmomentump
  momresults_12m[array_entry,'mom_vs']= vsmomentump
  momresults_12m[array_entry,'lreturnp']=lreturnp
  momresults_12m[array_entry,'sreturnp']=sreturnp
  momresults_12m[array_entry,'greturnp']=greturnp
  momresults_12m[array_entry,'vreturnp']=vreturnp
  momresults_12m[array_entry,'rp']=rp
  momresults_12m[array_entry,'cumr_l']=((1+momresults_12m[array_entry,'lreturnp']/100)*(1+momresults_12m[array_entry-
1,'cumr_l']/100)-1)*100
  momresults_12m[array_entry,'cumr_s']=((1+momresults_12m[array_entry,'sreturnp']/100)*(1+momresults_12m[array_entry-
1,'cumr_s']/100)-1)*100
  momresults_12m[array_entry,'cumr_g']=((1+momresults_12m[array_entry,'greturnp']/100)*(1+momresults_12m[array_entry-
1,'cumr_g']/100)-1)*100
  momresults_12m[array_entry,'cumr_v']=((1+momresults_12m[array_entry,'vreturnp']/100)*(1+momresults_12m[array_entry-
1,'cumr_v']/100)-1)*100
  momresults_12m[array_entry,'cum_rp']=((1+momresults_12m[array_entry,'rp']/100)*(1+momresults_12m[array_entry-
1,'cum_rp']/100)-1)*100
  i = i+1
}

# KPSS Tests
k = data.frame(matrix(nrow=170,ncol=49))
for (i in 1:49){
  for (j in 270:2160){
    entry = j/270
    position = j+90
    assign(paste("k_",i,"_",j,sep=""),kpss.test(industrydaily[j:position,
                                                  all_industries[i]], null="Trend"))
    k[entry,i]=eval(parse(text=paste("k_",i,"_",j,"$p.value",sep="")))
    j = j+270
  }
  i = i+1
}

# ADF Tests
d = data.frame(matrix(nrow=170,ncol=49))
for (i in 45:49){
  for (j in 270:2160){
    entry = j/270
    position = j+90
    assign(paste("d_",i,"_",j,sep=""),adf.test(industrydaily[j:position,
                                        all_industries[i]]))
    d[entry,i]=eval(parse(text=paste("d_",i,"_",j,"$p.value",sep="")))
    j = j+270
  }
  i = i+1
}

# ACF Plots
for (i in 1:49){
  acf(alldata[1:240,all_industries[i]],type="correlation"
      ,main= paste("Autocorrelation Plot of ",all_industries[i]," (240 months)",sep=""))
}

momresults_12m_h3 <- data.frame(matrix(nrow=41, ncol=19))
colnames(momresults_12m_h3) <- c('Month','lreturnp', 'sreturnp', 'greturnp',
                              'vreturnp', 'rp', 'cumr_l','cumr_s', 'cumr_g',
                              'cumr_v', 'cum_rp','mom_ll','mom_ls',
                              'mom_sl','mom_ss','mom_gl','mom_gs',
                              'mom_vl','mom_vs')
momresults_12m_h3[1,'lreturnp']=1
momresults_12m_h3[1,'sreturnp']=1
momresults_12m_h3[1,'greturnp']=1
momresults_12m_h3[1,'vreturnp']=1
momresults_12m_h3[1,'rp']=1
momresults_12m_h3[1,'cumr_l']=1
momresults_12m_h3[1,'cumr_s']=1
momresults_12m_h3[1,'cumr_g']=1
momresults_12m_h3[1,'cumr_v']=1
momresults_12m_h3[1,'cum_rp']=1

# r(2) = return on 121st month
# rf(2) = risk free return on 120th month

j = 2
for(i in 3:120){
  no_months = i+120
  rf = risk.free.rate(no_months-1)

  lbetas = calculate.betas.pooled.estimate.large(no_months-1)
  sbetas = calculate.betas.pooled.estimate.small(no_months-1)
  gbetas = calculate.betas.pooled.estimate.growth(no_months-1)
  vbetas = calculate.betas.pooled.estimate.value(no_months-1)
  lreturns = alldata[no_months,large_industries]
  sreturns = alldata[no_months,small_industries]
  greturns = alldata[no_months,growth_industries]
  vreturns = alldata[no_months,value_industries]
  lmomentum = momentum(no_months-1,large_industries_minus_rf,11)
  smomentum = momentum(no_months-1,small_industries_minus_rf,11)
  gmomentum = momentum(no_months-1,growth_industries_minus_rf,11)
  vmomentum = momentum(no_months-1,value_industries_minus_rf,11)

  ordered_lbetas = lbetas[order(lmomentum),1]
  ordered_sbetas = lbetas[order(smomentum),1]
  ordered_gbetas = lbetas[order(gmomentum),1]
  ordered_vbetas = lbetas[order(vmomentum),1]
  ordered_lreturns = lreturns[order(lmomentum)]
  ordered_sreturns = sreturns[order(smomentum)]
  ordered_greturns = greturns[order(gmomentum)]
  ordered_vreturns = vreturns[order(vmomentum)]
  ordered_lmomentum = lmomentum[order(lmomentum),1]
  ordered_smomentum = smomentum[order(smomentum),1]
  ordered_gmomentum = gmomentum[order(gmomentum),1]
  ordered_vmomentum = vmomentum[order(vmomentum),1]

  llmomentum = as.matrix(ordered_lmomentum[5:8]) # long the highest momentum
  lsmomentum = as.matrix(ordered_lmomentum[1:4])
  slmomentum = as.matrix(ordered_smomentum[5:8])
  ssmomentum = as.matrix(ordered_smomentum[1:4])
  glmomentum = as.matrix(ordered_gmomentum[5:8])
  gsmomentum = as.matrix(ordered_gmomentum[1:4])
  vlmomentum = as.matrix(ordered_vmomentum[5:8])
```

```
vsmomentum = as.matrix(ordered_vmomentum[1:4])
llbetas = as.matrix(ordered_lbetas[5:8]) # long the highest momentum
lsbetas = as.matrix(ordered_lbetas[1:4])
slbetas = as.matrix(ordered_sbetas[5:8])
ssbetas = as.matrix(ordered_sbetas[1:4])
glbetas = as.matrix(ordered_gbetas[5:8])
gsbetas = as.matrix(ordered_gbetas[1:4])
vlbetas = as.matrix(ordered_vbetas[5:8])
vsbetas = as.matrix(ordered_vbetas[1:4])
llreturns = as.matrix(ordered_lreturns[5:8]) # returns for highest momentum
lsreturns = as.matrix(ordered_lreturns[1:4])
slreturns = as.matrix(ordered_sreturns[5:8])
ssreturns = as.matrix(ordered_sreturns[1:4])
glreturns = as.matrix(ordered_greturns[5:8])
gsreturns = as.matrix(ordered_greturns[1:4])
vlreturns = as.matrix(ordered_vreturns[5:8])
vsreturns = as.matrix(ordered_vreturns[1:4])

for (i in 1:4){
  llweights[i,1] = 0.25 # llmomentum[i,1]/sum(llmomentum[,1])
  slweights[i,1] = 0.25 # slmomentum[i,1]/sum(slmomentum[,1])
  glweights[i,1] = 0.25 # glmomentum[i,1]/sum(glmomentum[,1])
  vlweights[i,1] = 0.25 # vlmomentum[i,1]/sum(vlmomentum[,1])
  lsweights[i,1] = 0.25 # 0.5-lsmomentum[i,1]/sum(lsmomentum[,1])
  ssweights[i,1] = 0.25 # 0.5-ssmomentum[i,1]/sum(ssmomentum[,1])
  gsweights[i,1] = 0.25 # 0.5-gsmomentum[i,1]/sum(gsmomentum[,1])
  vsweights[i,1] = 0.25 # 0.5-vsmomentum[i,1]/sum(vsmomentum[,1])
}

# check sum to 1

llbetap = t(llweights)%*%llbetas
lsbetap = t(lsweights)%*%lsbetas
slbetap = t(slweights)%*%slbetas
ssbetap = t(ssweights)%*%ssbetas
glbetap = t(glweights)%*%glbetas
gsbetap = t(gsweights)%*%gsbetas
vlbetap = t(vlweights)%*%vlbetas
vsbetap = t(vsweights)%*%vsbetas
llreturnp = as.matrix(llreturns)%*%as.matrix(llweights)
lsreturnp = as.matrix(lsreturns)%*%as.matrix(lsweights)
slreturnp = as.matrix(slreturns)%*%as.matrix(slweights)
ssreturnp = as.matrix(ssreturns)%*%as.matrix(ssweights)
glreturnp = as.matrix(glreturns)%*%as.matrix(glweights)
gsreturnp = as.matrix(gsreturns)%*%as.matrix(gsweights)
vlreturnp = as.matrix(vlreturns)%*%as.matrix(vlweights)
vsreturnp = as.matrix(vsreturns)%*%as.matrix(vsweights)
llmomentump = t(llweights)%*%llmomentum
lsmomentump = t(lsweights)%*%lsmomentum
slmomentump = t(slweights)%*%slmomentum
ssmomentump = t(ssweights)%*%ssmomentum
glmomentump = t(glweights)%*%glmomentum
gsmomentump = t(gsweights)%*%gsmomentum
vlmomentump = t(vlweights)%*%vlmomentum
vsmomentump = t(vsweights)%*%vsmomentum

lreturnp = (1/llbetap)*(llreturnp - rf)+(-1/lsbetap)*(lsreturnp - rf)
sreturnp = (1/slbetap)*(slreturnp - rf)+(-1/ssbetap)*(ssreturnp - rf)
greturnp = (1/glbetap)*(glreturnp - rf)+(-1/gsbetap)*(gsreturnp - rf)
vreturnp = (1/vlbetap)*(vlreturnp - rf)+(-1/vsbetap)*(vsreturnp - rf)

rp = 0.25*(lreturnp+sreturnp+greturnp+vreturnp)

momresults_12m_h3[j,'mom_ll']= llmomentump
momresults_12m_h3[j,'mom_ls']= lsmomentump
momresults_12m_h3[j,'mom_sl']= slmomentump
momresults_12m_h3[j,'mom_ss']= ssmomentump
momresults_12m_h3[j,'mom_gl']= glmomentump
momresults_12m_h3[j,'mom_gs']= gsmomentump
momresults_12m_h3[j,'mom_vl']= vlmomentump
momresults_12m_h3[j,'mom_vs']= vsmomentump
```

```
  momresults_12m_h3[j,'lreturnp']=lreturnp
  momresults_12m_h3[j,'sreturnp']=sreturnp
  momresults_12m_h3[j,'greturnp']=greturnp
  momresults_12m_h3[j,'vreturnp']=vreturnp
  momresults_12m_h3[j,'rp']=rp
  momresults_12m_h3[j,'cumr_l']=((1+momresults_12m_h3[j,'lreturnp']/100)*(1+momresults_12m_h3[j-1,'cumr_l']/100)-1)*100
  momresults_12m_h3[j,'cumr_s']=((1+momresults_12m_h3[j,'sreturnp']/100)*(1+momresults_12m_h3[j-1,'cumr_s']/100)-1)*100
  momresults_12m_h3[j,'cumr_g']=((1+momresults_12m_h3[j,'greturnp']/100)*(1+momresults_12m_h3[j-1,'cumr_g']/100)-1)*100
  momresults_12m_h3[j,'cumr_v']=((1+momresults_12m_h3[j,'vreturnp']/100)*(1+momresults_12m_h3[j-1,'cumr_v']/100)-1)*100
  momresults_12m_h3[j,'cum_rp']=((1+momresults_12m_h3[j,'rp']/100)*(1+momresults_12m_h3[j-1,'cum_rp']/100)-1)*100
  j = j+1
  i = i+3
}

# Calculate all alpha significance levels
calculate.alpha.estimate.over.stderr <-  function(no_months){
  Alpha_tvalues <- data.frame(matrix(nrow=49,ncol=1))
  insample <- alldata[0:no_months,c(all_industries_minus_rf,'Mkt.RF')]
  for (i in 1:49){
    lin_mod_all = lm(insample[,all_industries_minus_rf[i]] ~ insample[,'Mkt.RF'])
    Alpha_tvalues[i,1] = coef(summary(lin_mod_all))[1,"t value"]
    i = i+1
  }
  Alpha_tvalues
}

# Find our chosen industries' alphas
calculate.alpha.estimate.over.stderr.eight <-  function(no_months,industr_minus_rf){
  Alpha_tvalues <- data.frame(matrix(nrow=8,ncol=1))
  insample <- alldata[0:no_months,c(industr_minus_rf,'Mkt.RF')]
  for (i in 1:8){
    lin_mod = lm(insample[,industr_minus_rf[i]] ~ insample[,'Mkt.RF'])
    Alpha_tvalues[i,1] = coef(summary(lin_mod))[1,"t value"]
    i = i+1
  }
  Alpha_tvalues
}

##############################################################################
# Residual Momentum Portfolio (3 months), Holding Period = 1month

momresults_3m_alpha <- data.frame(matrix(nrow=121, ncol=27))
colnames(momresults_3m_alpha) <- c('Month','lreturnp', 'sreturnp', 'greturnp',
                                   'vreturnp', 'rp', 'cumr_l','cumr_s', 'cumr_g',
                                   'cumr_v', 'cum_rp','mom_ll','mom_ls',
                                   'mom_sl','mom_ss','mom_gl','mom_gs',
                                   'mom_vl','mom_vs','alp_ll','alp_ls',
                                   'alp_sl','alp_ss','alp_gl','alp_gs',
                                   'alp_vl','alp_vs')
momresults_3m_alpha[1,'lreturnp']=1
momresults_3m_alpha[1,'sreturnp']=1
momresults_3m_alpha[1,'greturnp']=1
momresults_3m_alpha[1,'vreturnp']=1
momresults_3m_alpha[1,'rp']=1
momresults_3m_alpha[1,'cumr_l']=1
momresults_3m_alpha[1,'cumr_s']=1
momresults_3m_alpha[1,'cumr_g']=1
momresults_3m_alpha[1,'cumr_v']=1
momresults_3m_alpha[1,'cum_rp']=1

# r(2) = return on 121st month
# rf(2) = risk free return on 120th month

for(i in 1:120){
  no_months = i+120
  rf = risk.free.rate(no_months-1)

  lalphas = calculate.alpha.estimate.over.stderr.eight(no_months-1,large_industries_minus_rf)
  salphas = calculate.alpha.estimate.over.stderr.eight(no_months-1,small_industries_minus_rf)
  galphas = calculate.alpha.estimate.over.stderr.eight(no_months-1,growth_industries_minus_rf)
  valphas = calculate.alpha.estimate.over.stderr.eight(no_months-1,value_industries_minus_rf)
```

```
lbetas = calculate.betas.pooled.estimate.large(no_months-1)
sbetas = calculate.betas.pooled.estimate.small(no_months-1)
gbetas = calculate.betas.pooled.estimate.growth(no_months-1)
vbetas = calculate.betas.pooled.estimate.value(no_months-1)
lreturns = alldata[no_months,large_industries]
sreturns = alldata[no_months,small_industries]
greturns = alldata[no_months,growth_industries]
vreturns = alldata[no_months,value_industries]
lmomentum = momentum(no_months-1,large_industries_minus_rf,2)
smomentum = momentum(no_months-1,small_industries_minus_rf,2)
gmomentum = momentum(no_months-1,growth_industries_minus_rf,2)
vmomentum = momentum(no_months-1,value_industries_minus_rf,2)

ordered_lalphas = lalphas[order(lalphas),1]
ordered_salphas = lalphas[order(salphas),1]
ordered_galphas = lalphas[order(galphas),1]
ordered_valphas = lalphas[order(valphas),1]
ordered_lbetas = lbetas[order(lalphas),1]
ordered_sbetas = lbetas[order(salphas),1]
ordered_gbetas = lbetas[order(galphas),1]
ordered_vbetas = lbetas[order(valphas),1]
ordered_lreturns = lreturns[order(lalphas)]
ordered_sreturns = sreturns[order(salphas)]
ordered_greturns = greturns[order(galphas)]
ordered_vreturns = vreturns[order(valphas)]
ordered_lmomentum = lmomentum[order(lalphas),1]
ordered_smomentum = smomentum[order(salphas),1]
ordered_gmomentum = gmomentum[order(galphas),1]
ordered_vmomentum = vmomentum[order(valphas),1]

llalphas = as.matrix(ordered_lalphas[5:8]) # long the highest alphas
lsalphas = as.matrix(ordered_lalphas[1:4])
slalphas = as.matrix(ordered_salphas[5:8])
ssalphas = as.matrix(ordered_salphas[1:4])
glalphas = as.matrix(ordered_galphas[5:8])
gsalphas = as.matrix(ordered_galphas[1:4])
vlalphas = as.matrix(ordered_valphas[5:8])
vsalphas = as.matrix(ordered_valphas[1:4])
llmomentum = as.matrix(ordered_lmomentum[5:8]) # long the highest resid momentum
lsmomentum = as.matrix(ordered_lmomentum[1:4])
slmomentum = as.matrix(ordered_smomentum[5:8])
ssmomentum = as.matrix(ordered_smomentum[1:4])
glmomentum = as.matrix(ordered_gmomentum[5:8])
gsmomentum = as.matrix(ordered_gmomentum[1:4])
vlmomentum = as.matrix(ordered_vmomentum[5:8])
vsmomentum = as.matrix(ordered_vmomentum[1:4])
llbetas = as.matrix(ordered_lbetas[5:8]) # long the highest momentum
lsbetas = as.matrix(ordered_lbetas[1:4])
slbetas = as.matrix(ordered_sbetas[5:8])
ssbetas = as.matrix(ordered_sbetas[1:4])
glbetas = as.matrix(ordered_gbetas[5:8])
gsbetas = as.matrix(ordered_gbetas[1:4])
vlbetas = as.matrix(ordered_vbetas[5:8])
vsbetas = as.matrix(ordered_vbetas[1:4])
llreturns = as.matrix(ordered_lreturns[5:8]) # returns for highest momentum
lsreturns = as.matrix(ordered_lreturns[1:4])
slreturns = as.matrix(ordered_sreturns[5:8])
ssreturns = as.matrix(ordered_sreturns[1:4])
glreturns = as.matrix(ordered_greturns[5:8])
gsreturns = as.matrix(ordered_greturns[1:4])
vlreturns = as.matrix(ordered_vreturns[5:8])
vsreturns = as.matrix(ordered_vreturns[1:4])

for (i in 1:4){
  llweights[i,1] = 0.25 # llmomentum[i,1]/sum(llmomentum[,1])
  slweights[i,1] = 0.25 # slmomentum[i,1]/sum(slmomentum[,1])
  glweights[i,1] = 0.25 # glmomentum[i,1]/sum(glmomentum[,1])
  vlweights[i,1] = 0.25 # vlmomentum[i,1]/sum(vlmomentum[,1])
  lsweights[i,1] = 0.25 # 0.5-lsmomentum[i,1]/sum(lsmomentum[,1])
  ssweights[i,1] = 0.25 # 0.5-ssmomentum[i,1]/sum(ssmomentum[,1])
  gsweights[i,1] = 0.25 # 0.5-gsmomentum[i,1]/sum(gsmomentum[,1])
```

```
   vsweights[i,1] = 0.25 # 0.5-vsmomentum[i,1]/sum(vsmomentum[,1])
  }

  # check sum to 1

  llbetap = t(llweights)%*%llbetas
  lsbetap = t(lsweights)%*%lsbetas
  slbetap = t(slweights)%*%slbetas
  ssbetap = t(ssweights)%*%ssbetas
  glbetap = t(glweights)%*%glbetas
  gsbetap = t(gsweights)%*%gsbetas
  vlbetap = t(vlweights)%*%vlbetas
  vsbetap = t(vsweights)%*%vsbetas
  llreturnp = as.matrix(llreturns)%*%as.matrix(llweights)
  lsreturnp = as.matrix(lsreturns)%*%as.matrix(lsweights)
  slreturnp = as.matrix(slreturns)%*%as.matrix(slweights)
  ssreturnp = as.matrix(ssreturns)%*%as.matrix(ssweights)
  glreturnp = as.matrix(glreturns)%*%as.matrix(glweights)
  gsreturnp = as.matrix(gsreturns)%*%as.matrix(gsweights)
  vlreturnp = as.matrix(vlreturns)%*%as.matrix(vlweights)
  vsreturnp = as.matrix(vsreturns)%*%as.matrix(vsweights)
  llmomentump = t(llweights)%*%llmomentum
  lsmomentump = t(lsweights)%*%lsmomentum
  slmomentump = t(slweights)%*%slmomentum
  ssmomentump = t(ssweights)%*%ssmomentum
  glmomentump = t(glweights)%*%glmomentum
  gsmomentump = t(gsweights)%*%gsmomentum
  vlmomentump = t(vlweights)%*%vlmomentum
  vsmomentump = t(vsweights)%*%vsmomentum
  llalphasp = t(llweights)%*%llalphas
  lsalphasp = t(lsweights)%*%lsalphas
  slalphasp = t(slweights)%*%slalphas
  ssalphasp = t(ssweights)%*%ssalphas
  glalphasp = t(glweights)%*%glalphas
  gsalphasp = t(gsweights)%*%gsalphas
  vlalphasp = t(vlweights)%*%vlalphas
  vsalphasp = t(vsweights)%*%vsalphas

  lreturnp = (1/llbetap)*(llreturnp - rf)+(-1/lsbetap)*(lsreturnp - rf)
  sreturnp = (1/slbetap)*(slreturnp - rf)+(-1/ssbetap)*(ssreturnp - rf)
  greturnp = (1/glbetap)*(glreturnp - rf)+(-1/gsbetap)*(gsreturnp - rf)
  vreturnp = (1/vlbetap)*(vlreturnp - rf)+(-1/vsbetap)*(vsreturnp - rf)

  rp = 0.25*(lreturnp+sreturnp+greturnp+vreturnp)
  array_entry = no_months - 119

  momresults_3m_alpha[array_entry,'alp_ll']= llalphasp
  momresults_3m_alpha[array_entry,'alp_ls']= lsalphasp
  momresults_3m_alpha[array_entry,'alp_sl']= slalphasp
  momresults_3m_alpha[array_entry,'alp_ss']= ssalphasp
  momresults_3m_alpha[array_entry,'alp_gl']= glalphasp
  momresults_3m_alpha[array_entry,'alp_gs']= gsalphasp
  momresults_3m_alpha[array_entry,'alp_vl']= vlalphasp
  momresults_3m_alpha[array_entry,'alp_vs']= vsalphasp
  momresults_3m_alpha[array_entry,'mom_ll']= llmomentump
  momresults_3m_alpha[array_entry,'mom_ls']= lsmomentump
  momresults_3m_alpha[array_entry,'mom_sl']= slmomentump
  momresults_3m_alpha[array_entry,'mom_ss']= ssmomentump
  momresults_3m_alpha[array_entry,'mom_gl']= glmomentump
  momresults_3m_alpha[array_entry,'mom_gs']= gsmomentump
  momresults_3m_alpha[array_entry,'mom_vl']= vlmomentump
  momresults_3m_alpha[array_entry,'mom_vs']= vsmomentump
  momresults_3m_alpha[array_entry,'lreturnp']=lreturnp
  momresults_3m_alpha[array_entry,'sreturnp']=sreturnp
  momresults_3m_alpha[array_entry,'greturnp']=greturnp
  momresults_3m_alpha[array_entry,'vreturnp']=vreturnp
  momresults_3m_alpha[array_entry,'rp']=rp

momresults_3m_alpha[array_entry,'cumr_l']=((1+momresults_3m_alpha[array_entry,'lreturnp']/100)*(1+momresults_3m_alpha[array_e
ntry-1,'cumr_l']/100)-1)*100
```

```
momresults_3m_alpha[array_entry,'cumr_s']=((1+momresults_3m_alpha[array_entry,'sreturnp']/100)*(1+momresults_3m_alpha[array_e
ntry-1,'cumr_s']/100)-1)*100

momresults_3m_alpha[array_entry,'cumr_g']=((1+momresults_3m_alpha[array_entry,'greturnp']/100)*(1+momresults_3m_alpha[array_
entry-1,'cumr_g']/100)-1)*100

momresults_3m_alpha[array_entry,'cumr_v']=((1+momresults_3m_alpha[array_entry,'vreturnp']/100)*(1+momresults_3m_alpha[array_
entry-1,'cumr_v']/100)-1)*100
  momresults_3m_alpha[array_entry,'cum_rp']=((1+momresults_3m_alpha[array_entry,'rp']/100)*(1+momresults_3m_alpha[array_entry-
1,'cum_rp']/100)-1)*100
  i = i+1
}

# Stochastic Domainance
ecdf_gmvp = ecdf(results[2:121,'r_gmvp'])
ecdf_tangent = ecdf(results[2:121,'r_tangent'])
ecdf_naive = ecdf(results[2:121,'r_naive'])
ecdf_market = ecdf(results[2:121,'r_market'])

for (i in 1:10){
  assign(paste('ecdf_',industries[i],sep=""),ecdf(alldata[121:240,industries[i]]))
  i = i+1
}

sum.ecdf.gmvp <- function(x){
  s = seq(from=-33,to=x,by=0.1)
  integral_ecdf = sum(ecdf_gmvp(s))
  integral_ecdf
}

sum.ecdf.tangent <- function(x){
  s = seq(from=-33,to=x,by=0.1)
  integral_ecdf = sum(ecdf_tangent(s))
  integral_ecdf
}

sum.ecdf.naive <- function(x){
  s = seq(from=-33,to=x,by=0.1)
  integral_ecdf = sum(ecdf_naive(s))
  integral_ecdf
}

sum.ecdf.market <- function(x){
  s = seq(from=-33,to=x,by=0.1)
  integral_ecdf = sum(ecdf_market(s))
  integral_ecdf
}

# Plot of ECDFs
t = -1500:2000/100
plot(x=t,y=ecdf_gmvp(t),
     main='Empirical Cumulative Distribution Functions',
     xlab='x', ylab='ECDF(x)',
     type='l',xlim=c(-15,15),ylim=c(0,1),lwd=2,col='black',cex.lab=0.9)
lines(x=t,y=ecdf_tangent(t),lwd=2,lty=1,col="purple")
lines(x=t,y=ecdf_naive(t),lwd=2,lty=1,col="darkgreen")
lines(x=t,y=ecdf_market(t),lwd=2,lty=1,col="yellow")
labels <- c("Global Minimum-Variance Portfolio",
          "Tangent Portfolio","Naive-Weighted Portfolio","Market Portfolio")
legend("bottomright", inset=.05, title="ECDFs",labels,cex=0.6,
       lwd=c(2,2,2,2),col=c("black","purple","darkgreen","yellow"))

# Sums minus each other
t = -330:400/10
position = 1 + 10*(t+33)
plot(x=t,y=sum_ecdf_tangent[position,1]-sum_ecdf_gmvp[position,1],
     main='Sum of Empirical Cumulative Distribution Functions',
     xlab='x', ylab='Sum of ECDF(t) from -15 to x',
     type='l',xlim=c(-33,40),ylim=c(-1,8),lwd=2,lty=1,col='purple',cex.lab=0.9)
lines(x=t,y=sum_ecdf_naive[position,1]-sum_ecdf_gmvp[position,1],lwd=2,lty=1,col="darkgreen")
lines(x=t,y=sum_ecdf_market[position,1]-sum_ecdf_gmvp[position,1],lwd=2,lty=1,col="yellow")
abline(0,0,lwd=2,lty=2,col="red")
labels <- c("Sum of ECDF of Tangent - GMVP","Sum of ECDF of Naive - GMVP","Sum of ECDF of Market - GMVP")
legend("topright", inset=.05, title="Legend",labels,cex=0.6,
      lwd=c(2,2,2),col=c("purple","darkgreen","yellow"))


# Analyse Performance of Portfolios
# > install.packages("PerformanceAnalytics")
long_avg_mrp = market.risk.premium(240)
long_avg_rf = mean(famafrench[483:1083,'RF'])

arithmetic_mean_gmvp_yearly = ((1+mean(results[2:121,'r_gmvp'])/100)^12-1)*100
temp_sum = 1
for (i in 1:120){
  temp_sum = temp_sum*((1+results[1+i,'r_gmvp']/100))
}
geometric_mean_gmvp_yearly = (temp_sum^(1/10)-1)*100
arithmetic_mean_gmvp_monthly = mean(results[2:121,'r_gmvp'])
geometric_mean_gmvp_monthly = (temp_sum^(1/120)-1)*100
sd_gmvp = sqrt(var(results[2:121,'r_gmvp']))
skewness_gmvp = skewness(results[2:121,'r_gmvp'])
kurtosis_gmvp = kurtosis(results[2:121,'r_gmvp'])
cov_gmvp_market = cov(results[2:121,'r_gmvp'],results[2:121,'r_market'])
corr_gmvp_market_pearson = cor(results[2:121,'r_gmvp'],results[2:121,'r_market'],method="pearson")
corr_gmvp_market_spearman = cor(results[2:121,'r_gmvp'],results[2:121,'r_market'],method="spearman")
var_market = var(results[2:121,'r_market'])
lin_mod_gmvp_no_alpha = lm(results[2:121,'r_gmvp']~0+alldata[121:240,'Mkt.RF'])
beta_gmvp_1 = cov_gmvp_market/var_market
beta_gmvp_2 = lin_mod_gmvp_no_alpha$coefficients[[1]]
beta_gmvp = 0.5*(beta_gmvp_1+beta_gmvp_2)
lin_mod_gmvp = lm(results[2:121,'r_gmvp']~alldata[121:240,'Mkt.RF'])
alpha_linmodel_gmvp = coef(summary(lin_mod_gmvp))[1,"Estimate"]
stderr_alpha_gmvp = coef(summary(lin_mod_gmvp))[1,"Std. Error"]
pvalue_alpha_gmvp = coef(summary(lin_mod_gmvp))[1,"Pr(>|t|)"]
VaR_gmvp_1m_0.95CL_nonpar = VaR(results[2:121,'r_gmvp']/100,
                          p=0.95,method="historical",invert=FALSE)*100
VaR_gmvp_1m_0.95CL_mod = VaR(results[2:121,'r_gmvp']/100,
                          p=0.95,method="modified",invert=FALSE)*100
ETL_gmvp_1m_0.95CL_nonpar = ES(results[2:121,'r_gmvp']/100,
                          p=0.95,method="historical",invert=FALSE)*100
ETL_gmvp_1m_0.95CL_mod = ES(results[2:121,'r_gmvp']/100,
                         p=0.95,method="modified",invert=FALSE)*100
semisigma_gmvp = SemiDeviation(results[2:121,'r_gmvp'])
downsidedeviation_gmvp_rf = DownsideDeviation(results[2:121,'r_gmvp'],long_avg_rf)
downsidedeviation_gmvp_mrp = DownsideDeviation(results[2:121,'r_gmvp'],long_avg_mrp)
jensen_alpha_gmvp = arithmetic_mean_gmvp_monthly - long_avg_rf - beta_gmvp*(long_avg_mrp)
sharpe_gmvp = (arithmetic_mean_gmvp_monthly - long_avg_rf)/sd_gmvp

t = 1:120
plot(x = alldata[120+t,'Mkt.RF'], y = results[t+1,'r_gmvp']-alldata[t+120,'RF'],
    main='Sharpe Line of GMVP Portfolio',
    xlab='Market Risk Premium', ylab='Excess Returns of GMVP Portfolio'
    ,xlim=c(-2,7),ylim=c(-3,5),lwd=2,col='blue',cex.sub=0.75,cex.main=0.9,cex.lab=0.9)
abline(lin_mod_gmvp_no_alpha,col="red",lwd=2,lty=1)
abline(a=long_avg_rf,b=sharpe_gmvp,col="darkgreen",lwd=2,lty=2)
labels <- c("CAPM","Sharpe Line")
legend("bottomright", inset=.05, title="Legend",labels,cex=0.6,lwd=c(2), col=c("red","darkgreen"))

treynor_gmvp = (arithmetic_mean_gmvp_monthly - long_avg_rf)/beta_gmvp
sortino_gmvp_rf = (arithmetic_mean_gmvp_monthly - long_avg_rf)/downsidedeviation_gmvp_rf
sortino_gmvp_mrp = (arithmetic_mean_gmvp_monthly - long_avg_rf)/downsidedeviation_gmvp_mrp
lin_gmvp = lm(I(results[2:121,'r_gmvp']-alldata[121:240,'RF']-jensen_alpha_gmvp)
         ~0+alldata[121:240,'Mkt.RF'])
```

Appendix C: References

https://www.youtube.com/watch?v=aHE0queiTuA
[Accessed November 3rd]

http://pages.stern.nyu.edu/~lpederse/papers/BettingAgainstBeta.pdf
[Accessed November 3rd]

https://www.youtube.com/watch?v=4QvF6YdyWAo
[Accessed November 4th]

http://www.econ.yale.edu/~af227/pdf/Betting%20Against%20Beta%20-%20Frazzini%20and%20Pedersen.pdf
[Accessed November 4th]

http://business.gwu.edu/wp-content/uploads/2014/12/Finance_Seminars_12.02.2014_LotteryDemand.pdf
[Accessed November 5th]

http://efinance.org.cn/cn/fm/Efficient%20Capital%20Markets%20A%20Review%20of%20Theory%20and%20Empirical%20Work.pdf
[Accessed November 7th]

https://www.youtube.com/watch?v=lf1u4dsHavs
[Accessed November 2nd]

http://www.kentdaniel.net/papers/unpublished/djk_2012_9.pdf
[Accessed November 1st]

http://www.bauer.uh.edu/rsusmel/phd/jegadeesh-titman93.pdf
[Accessed November 3rd]

http://www.stockopedia.com/content/what-really-causes-price-momentum-69873/
[Accessed November 7th]