Contents

Abstract

People who build successful businesses often experience asymmetrical wealth endowments, where most of their investments are tied to a particular asset related to their business. In the context of Texas, for example, the recent shale oil and gas development generated a great deal of wealth for many families in Texas, so it possibly has created substantial asymmetry in local investors' asset allocations. According to Modern Portfolio Theory [8], investors should diversify their portfolio into a mix of different assets in order to achieve an optimal portfolio, in which investors can either minimize the risk given a certain expected return or maximize the expected return given a certain risk. Expected return can be estimated from historical returns on the assets under consideration, while risk is measured by the covariance matrix of the historical returns. However, the tax costs associated with diversification of a dominant asset may mean the portfolio from classical Portfolio Theory is no longer optimal.

This thesis has two objectives. First, the classical portfolio optimization methods will be presented, and we will discuss how they can be modified to take tax costs into effect. Then, with the revised optimization tools, we will investigate different strategies for diversification of a portfolio initially consisting only of oil. To quantify the uncertainty in the forecasts, we will use GARCH models based on the historical data to simulate different plausible outcomes.

1. Introduction

The economic sector of Texas makes it possible that local investors have their major investment of the portfolio in the oil industry to take the advantage of the expected return.

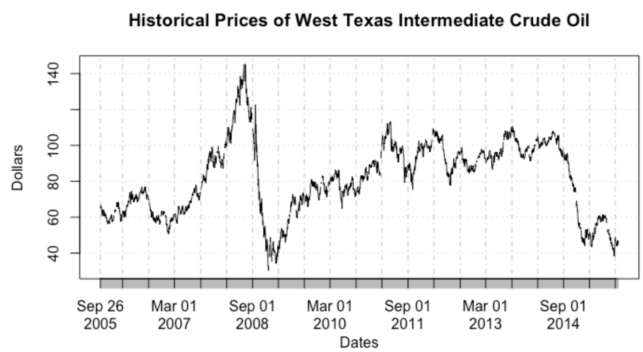

Figure 1. Historical Oil Prices in the Past 10 Years

However, with the occasional dramatic fall of oil prices in recent years as shown in Figure 1, a portfolio that started this period with oil as the dominant asset would be extremely suboptimal and suffer a major loss. In order to prevent further losses and rebuild the portfolio into an efficient one, it may be necessary to take strategies to get rid of the dominant asset and diversify the investments for a sustained profit.

The organization of the thesis is as follows. In Section I, we introduce the problem of optimal strategies for unbalanced portfolios, define basic concepts, and review what is known about the problem. In Section II, we provide details about the methods we used to investigate the problem. In Section III, we present our results and discuss their implications. Finally, in Section IV we provide conclusions and discuss possibilities for future work.

1.1. **Basic Concepts of Portfolio Theory.**

1.1.1. *Assets.* *Assets* are financial instruments that can be converted to cash. Examples of assets are cash, real estate, privately owned business or partnerships, and employer stock options [6]. In this thesis, we will study liquid assets such as stocks and commodities which are relatively easy to sell or buy. The value of an asset is its current price.

The goal of investing in assets is to make profit. A profit may arise from selling an asset at a higher price than was paid, or from dividends paid by the asset, or by short selling the asset (see subsection 1.1.6 below). These are collectively known as the *return* on the asset.

There is more than one way to define returns mathematically. The *simple gross return* is defined as

$$R_t = \frac{P_t}{P_{t-1}} - 1$$

where P_t is the price of an asset at time t.

Besides simple gross return, *log returns* are defined as

$$r_t = \log(1 + R_t)$$

where R_t is a simple gross return.

We will be using log returns for the modeling in the following section because log returns are stationary: that is, for the time frames under investigation, the mean and standard deviation of the time series are constant.

Along with returns, there comes the risk of loss, and usually the higher the potential returns are, the higher risks they have. Risk is measured by the standard deviation of the return of an asset, since standard deviation indicates the variability of return. Portfolio theory also includes the notion of a risk-free asset, defined as an asset whose standard deviation of the return is 0.

1.1.2. *Portfolios.* A *portfolio* is a combination of different financial assets such as different stocks. Each asset in a portfolio is associated with a *weight* ω_i, which is defined by the ratio of the current value of the asset to the total value of a portfolio. The sum of weights is thus 1.

In the thesis, we will be managing a portfolio with multiple assets. The expected return of a portfolio is defined as the sum of weighted expected return of each asset. Suppose we have N risky assets, the i_{th} risky asset is X_i, and X_i has expected return μ_i and weight ω_i. Then the *expected return of a portfolio* is

$$\sum_{i=1}^{N} \omega_i \mu_i = \omega^t \mu$$

where ω_i and μ_i is the weight and return of asset X_i.

The *risk of a portfolio* is defined by the variance of the return on the portfolio, which is

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \omega_i \omega_j \Omega_{i,j} = \omega^t \Omega \omega$$

where Ω is the covariance matrix of assets in a portfolio and $\Omega_{i,j} = Cov(X_i, X_j)$

1.1.3. *Portfolio Optimization.* There are two possible objectives for portfolio optimization: maximizing return or minimizing risk. Unfortunately, risk and return are not independent, and so we instead look for conditional optimization: finding a portfolio that either has a higher expected return than any other portfolio with the same level risk, or a portfolio that has the smallest risk of any portfolio with the same expected return [8]. From a financial advisor's perspective, the task is to maximize return for a given client's risk tolerance. A portfolio that satisfies either criteria will be called an *optimal portfolio*.

Due to the change of expected returns and risks of stocks, it is necessary to recalculate the optimal weights of stocks in a portfolio regularly. This recalculation is called *rebalancing*.

Suppose we find the optimal portfolio for each possible expected return and plot them with their corresponding risk. Then the points $(risk, return)$ form an increasing concave curve. The point on this locus that reaches the minimum value of the risk is called the *minimum variance portfolio*. All the points on the curve that have higher returns than the minimum variance portfolio are called the *efficient frontier*, and each portfolio on the efficient portfolio is optimal in either of the senses above, and is called an efficient portfolio. An example of an efficient frontier may be seen in Figure 2. Each dot is an example of a portfolio with one single asset.

1.1.4. *Sharpe Ratio.* In the section above, there are infinitely many optimal portfolios along efficient frontier. Is there a "preferred" optimal portfolio? In 1966, William F. Sharpe [12] introduced what is known as the Sharpe ratio, which is defined as the average return earned in excess of the risk-free rate per unit of volatility or total risk.

$$SharpeRatio = \frac{Portfolio\,Return - Risk\,free\,Return}{Risk\,of\,Portfolio}$$

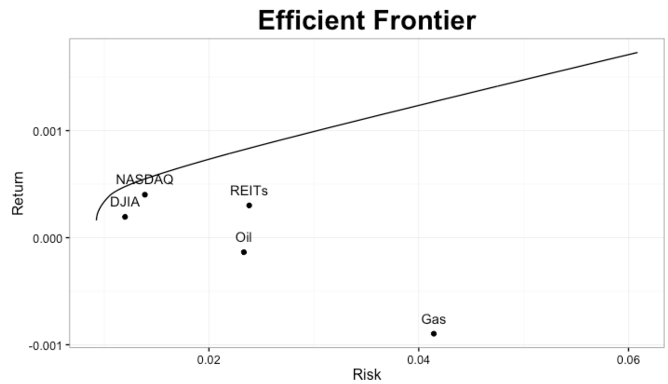

FIGURE 2. Efficient Frontier and Single Asset Portfolios

Suppose there is a risk-free asset, shown in Figure 3 as the point in the lower left hand corner. If we connect the risk-free point with a point on the efficient frontier, the slope of the resulting line represents the Sharpe ratio for the portfolio at that point. Therefore, the Sharpe ratio can be thought of as the ratio of the difference of a portfolio return and risk-free return to risk. A line with a large slope implies higher expected return to a given risk level, so larger Sharpe ratio gives a better portfolio result. As shown in Figure 3, the largest Sharpe Ratio can be represented by the tangent of the bottom angle.

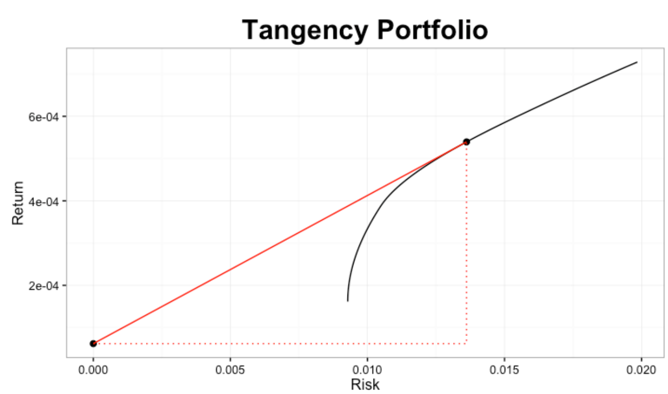

FIGURE 3. Sharpe Ratio

Hence, the portfolio with the largest Sharpe ratio is our target portfolio because it has the highest reward-to-risk ratio, and it is called the *tangency portfolio*.

1.1.5. *Taxes.* Almost everything people own and use for personal or investment purposes is a capital asset. When people sell a capital asset, the difference between the adjusted basis in the asset and the amount realized from the sale is a *capital gain or a capital loss*, and capital gains are taxable.

In reality, paying taxes is a significant cost to investors, and one of the reasons is that a pretax algorithm may not apply in an after-tax environment [7]. For example, if a profit is made upon rebalancing, then capital gain taxes lead to a reduction of the amount available for reinvesting, and that will cause a failure to reach theoretical optimal weights. In other words, the loss on tax will interfere with the optimization process. Therefore, our optimization process will incorporate tax costs when rebalancing the portfolio.

1.1.6. *Short Selling.* Short selling is a way to potentially profit when investors anticipate a stock price will fall. An investor borrows shares of a stock from a broker and sells them. At a later point in time, the investor buys the stock and return the shares to the lender. The difference between the initial sales price and the later purchase price will be a profit if the price has gone down. In the term of portfolio optimization, allowing short selling increases the flexibility of the investor to take advantage of changes in asset prices. Assets in a portfolio that are sold short are represented by negative weights, $\omega_i < 0$, since they represent debts the investor owes the broker.

1.2. **Mathematical Tools.**

1.2.1. *Linear Programming.* Linear programming has a linear objective function and linear constraints. As stated in the previous section, the expected return of a portfolio $\omega^t\mu$ is a linear function. In the process of portfolio optimization, we can apply linear programming to find the portfolio of the highest return. Therefore, the objective function we maximize is

$$f(\omega) = \omega^t\mu$$

subject to

$$\omega_b \leq \omega_i \leq 1$$

for each asset $X_i, i = 1, 2, \ldots, N$, with $\mu_i = \mathrm{E}(X_i)$ and for some borrowing limit $\omega_b < 0$.

In our algorithm below, linear programming will be used to locate the optimal portfolio with the highest return, which defines the upper boundary of the efficient frontier.

1.2.2. *Quadratic Programming.* Quadratic programming is used to minimize a quadratic objective function subject to linear constraints. As shown before, the variance or the risk of a portfolio $\omega^t\Omega\omega$ is a quadratic function, and if we need to minimize the risk to reach an optimal portfolio given a target return, then quadratic programming can be applied to solve the problem of portfolio optimization. The general form of quadratic objective function to be minimized is

$$f(w) = cw + \frac{1}{2}w^TQw$$

where c is an N-dimensional row vector describing the coefficients of the linear terms in the objective function, and Q is an symmetric matrix describing the coefficients of the quadratic terms. Linear inequality constraints are

$$A\omega \leq b$$

where A is an $m \times N$ matrix and b is an $m \times 1$ vector where m is the number of inequality constraints. Equality constraints are

$$A_{eq}\omega = b_{eq}$$

where A_{eq} is an $n \times N$ matrix and b_{eq} is an $n \times 1$ vector where n is the number of inequality constraints.

In the case of basic portfolio optimization, we can set $c = 0$, $w_i = \omega_i$, and $Q = \Omega$, then the objective function reduces to (one half of) the risk of the portfolio,

$$f(\omega) = \frac{1}{2}\omega^t \Omega \omega$$

There are two equality constraints, one is that the weights sum to 1 and the other that the portfolio return is a specified μ_p, so the equality constraints become

$$1^T \omega = 1$$

and

$$\mu^t \omega = \mu_p$$

The inequality constraints are similar to the part of linear programming, which are

$$\omega_b \leq \omega_i \leq 1$$

for some borrowing limit $\omega_b < 0$.

The formula above can be used for very basic cases, in which investors have complete freedom to pick portfolios. However, in the thesis, we are discussing the issue of asymmetrical wealth, therefore, we will have to modify the constraints of quadratic programming in order to model the dominant asset and some of the strategies that might be used to manage it.

There are many methods can be used to solve quadratic programming problems. Here is an outline using Lagrange multipliers. Suppose the weights of optimal portfolio are $\omega_{\mu p}$. To find $\omega_{\mu p}$, then the Lagrangian function is

$$L(\omega, \delta_1, \delta_2) = \omega^t \Omega \omega + \delta_1(\mu_p - \omega_{\mu p}^t \mu) + \delta_2(1 - \omega^t \mathbf{1})$$

where δ_1 and δ_2 are Lagrange multipliers. Then we just need to solve

$$\frac{\partial}{\partial \omega} L(\omega, \delta_1, \delta_2) = 2\Omega\omega_{\mu p} - \delta_1 \mu - \delta_2 \mathbf{1} = 0$$

1.2.3. *Time Series Models.* When studying the data of stock market, time series models are widely used because the data is a sequence of observations taken over time. Regular time series models like ARMA models (mixed Autoregressive and Moving Average) are not satisfactory for modeling the stock market due to their assumption of constant volatility. In other words, if we use ARMA models, risks of stocks are assumed constant, while in reality they are not.

Therefore, Engle [4] developed Generalized Autoregressive Conditional Heteroscedasticity (GARCH) time series models to permit the modeling of randomly varying risks. Let a_t be the random innovation in the standard deviation σ_t at time t. The GARCH (p, q) model is

$$a_t = \epsilon_t \sigma_t$$

where ϵ_t is an independent $\mathcal{N}(0,1)$ random value,

$$\sigma_t = \sqrt{\alpha_0 + \sum_{i=1}^{q} \alpha_i a_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2}$$

[4]. α and β are GARCH parameters. Parameter α controls the risk fluctuation, while β controls how the current risk correlates with the previous value.The conditional standard deviation can exhibit more persistent periods of high or low volatility because past values of σ_t process are fed back into the present value.

While the GARCH formula above is stated in full generality for lags p and q, in most cases, a GARCH (1,1) process is sufficient for simulation of real assets [10].

There are two major reasons to use a time series model such as GARCH(1,1) model. Firstly, the parameter values can give insight about characteristics of the asset's behavior over time, and secondly, we will be able to fit the model based on historical data and use the model to simulate new time series. However, a simple GARCH (1, 1) model is only applicable to one dimensional data such as the time series of one particular stock. In the management of our asymmetrical portfolio, we have more than one asset. Therefore, we need a multi-dimensional GARCH models for the portfolio. Bollerslev developed an n-dimensional GARCH model that comprises n-univariate GARCH processes related to one another with a constant conditional correlation matrix [2]. It is called the Constant Conditional Correlation GARCH or CCC-GARCH model, and we will use the diagonal specification. The CCC-GARCH defines portfolio risk as

$$H_t = D_t R D_t = (\rho_{ij} \sigma_{i,t} \sigma_{j,t})$$

where $D_t = diag(\sigma_{1,t} ... \sigma_{N,t})$. $\sigma_{i,t}^2$ can be defined as the variance at time t from a univariate GARCH model, and $R = (\rho_{ij})$ is a symmetric positive definite matrix with ρ_{ij} being the constant conditional correlation between assets X_i and X_j. The original CCC-GARCH model has a GARCH(1,1) specification for each conditional variance in D_t [1]. CCC-GARCH models can be fit starting from individual assets' GARCH(1,1) models.

1.3. **Asymmetrical Wealth Endowment.** Asymmetrical wealth endowments is defined as "an over allocation in an asset or asset class leads to a sub-optimal portfolio allocation" [6]. According to Modern Portfolio Theory [8], such over allocation of a particular asset generates a relatively risky portfolio. For example, Bill Gates used to be the largest shareholder of Microsoft, which means the majority of his wealth was tied up in Microsoft stock. Therefore, for people in such a position, practical strategies that can be used to help investors reduce the weight of their dominant liquid assets play a significant role when rebalancing portfolios.

Friday et al. [6] provides an overview of what asymmetrical endowment is and causes of it. In the discussion of possible rebalancing strategies, many were either of limited use due to specialized circumstances, or involved exotic option strategies that would not be attractive to many investors. Thus, we seek to provide a strategy that ordinary investors can follow without sophisticated financial knowledge.

2. Methodology

2.1. **Description of the Portfolio Optimization Algorithm.** Below is a more detailed description of the algorithm we developed to optimize a portfolio. The code is available as an appendix. All coding was performed in `R` [11].

Given time series for assets under consideration, we use mean log returns to define the expected return vector μ. There are two choices for defining a covariance matrix. One choice is to use the covariance matrix of the historical time series. On the other hand, a CCC-GARCH model can be fit to simulate a different plausible covariance matrix. A univariate GARCH model is fit for each asset, and the resulting parameters serve as initial values for a CCC-GARCH model to simulate a covariance matrix for portfolio optimization. The univariate GARCH models are fit using `R`'s `fGarch` package ([14]), and the CCC-GARCH model is fit using `R`'s `ccgarch` package ([9]).

Next, we use linear programming to find the portfolio with the highest feasible return. There is a highest feasible return because we imposed limitations on the amount the investor could borrow via short sales, as well as limits on the dominant asset in certain scenarios. As noted in the Introduction, the objective function for portfolio returns is linear so we can use linear programming. At the same time, we also apply quadratic programming to find the minimum variance portfolio, as portfolio risks have a quadratic objective function. As stated before, our target portfolio is tangency portfolio, and tangency portfolios will lie in between these two portfolios. The linear programming was performed using `R`'s `boot` package ([3]), while the quadratic programming used `R`'s `quadprog` package ([13]).

Then, on the interval of returns from the minimum risk portfolio to the maximum return portfolio, we are able to define an efficient frontier. Given the bounds on possible returns, a standard one-dimensional optimization method is applied to find the maximum Sharpe ratio which leads to a tangency portfolio.

Lastly, we compute the value of the portfolio using the optimal weights, then deduct any capital gains tax. The matrix system used to adjust for taxation is

$$s_i p_i = \omega_i \sum s_i p_i$$

$$\sum s_i p_i = \text{value of portfolio after taxes}$$

where s_i is the shares after rebalancing, ω_i is the optimal weight, and p_i is the current price for asset i.

2.2. **Description of Experiments.** The algorithm described above can be used to conduct simulations to guide decision making about portfolio management. Below, we describe one such set of simulations as an example of what is possible using the algorithm.

2.2.1. *Details of the Experiment.* In the context of Texas, suppose that West Texas Intermediate Crude Oil is the only asset that an investor has in their portfolio. Given the risk of falling oil prices, the investor wants to shift some or all of their portfolio to other assets. For this experiment, the investor will have four other possible assets to choose among: Nasdaq 100 Index, Dow Jones Industrial Average, Wilshire US Real Estate Investment Trust REIT Index (REIT), and Henry Hub

Natural Gas. Because we had to estimate covariance matrices, we had to keep the number of alternative assets small in order to control estimation errors. We chose 3-month US Treasury bills as the risk-free asset. [5] Although most investors do not own actual oil or gas directly, and cannot "own" the Dow Jones Industrial Average, there are index funds whose value closely follows these items, and so we treat them as "liquid" assets.

The Nasdaq-100 Index includes 100 of the largest domestic and international non-financial companies listed on The Nasdaq Stock Market, the Dow Jones Industrial Average (DJIA) is a price-weighted average of 30 significant stocks traded on the New York Stock Exchange and the Nasdaq, and Wilshire US Real Estate Investment Trust (REIT) provide a real estate investment, which might have a relatively low correlation with the other part of the market, so the risk of the portfolio may be decreased. The three indexes are taken into account to the portfolio because of their well-known market characteristics: they have relatively stable expected returns with low risks that we can take advantages of. The remaining asset, Henry Hub Natural Gas, was chosen because the natural gas industry is closely related to the oil industry. If we have two assets that are significantly related, the risk of the portfolio increases, and we wanted to include that possibility as part of the experiment.

The basic logic is that an investor desires to rebalance the portfolio regularly. If the rebalancing period is too short, then investors may spend a lot of money on transactions costs. On the other hand, if it is too long, then the forecast errors will be large. As a result, we chose a rebalancing period of three months because we are trying to predict a reasonable length of the future data and decrease transaction costs and error on forecast, so rebalancing quarterly is compromise choice. In the experiment, we have the investor rebalance quarterly for a five year period. Since each rebalancing is based on a five year historical dataset, we need 10 years of data overall.

To restrict the amount of short selling permitted in the experiments, we chose a maximum borrowing limit of $\omega_b = -1$, that is, no single asset can be short sold more than the total value of the portfolio.

Therefore, the optimal weight of each asset in the tangency portfolio that we are supposed to have for the next quarter could be found. The investor would be able to determine the loss or profit of their hypothetical portfolio at the end of each quarter by comparing our simulation results with the real data of the next quarter. Therefore, the investor can keep track of the value of the portfolio and the weight of the dominant asset in the experiments. More significantly, the investor will be able to conclude what strategy would have been better by comparing the final wealth.

2.3. **General Views of Three Strategies an Investor Might Follow.** There are several methods to deal with an asymmetrical portfolio. Depending on complexities of methods and investors' demands, the final goals vary. Basically, we investigate three strategies to compare in order to achieve the highest final portfolio value.

2.3.1. *Hold Asymmetrical Portfolio.* When facing a falling price on a single asset portfolio, investors could choose to do nothing and keep holding the shares of it. Investors who apply this strategy may take such price falling as fluctuations in market and expect the price will go back to where it used to be or even higher, so there is no need to do anything. This is especially true when they have large unrealized profits, that is, the current value of the asset would result in a profit if they sold the asset. In this way, they can avoid paying capital gain tax to prevent further losses if they choose not to sell any of the shares and keep making profit if the price of the asset goes higher in the future.

2.3.2. *Reduce Dominant Asset to Optimal Levels Immediately.* In contrast to holding the single asset portfolio when facing a price fall, investors can choose to reduce their holdings immediately to the amount recommended by portfolio optimization. Even though investors potentially have to pay a large amount of capital gain taxes if they have made profits on their acquisition price, they may still choose to do so because they estimate the price of the asset will not recover for a long period of time. By doing so, investors no longer have problem of asymmetrical wealth portfolio, they can start to build a new efficient portfolio with the money they cash out from the previous portfolio and have additional flexibility to rebalance it regularly.

2.3.3. *Decrease the Weight Gradually.* In between the two extreme methods above, investors could choose to decrease the weight of the dominant asset step by step. More specifically, every time when investors rebalance the portfolio, they liquefy a certain proportion of the dominant asset and reinvest the money in the other assets until the dominant asset is reduced to its optimal level. To do so, we can customize an inequality constraint in the optimization process by setting the minimum weight of the dominant asset when rebalancing each time, for example,

$$\omega_d \geq \omega_{min}$$

where ω_d is the weight of the dominant asset and ω_{min} is the minimum weight we set when rebalancing.

The inequality constraint works in several ways. Firstly, the minimum weight we assign each time prevents it from overselling, so the investor can avoid potential loss on tax. Secondly, the assigned weight can serve as a flag to show when the dominant asset has come to its optimal weight if the final computed weight is greater than the assigned.

3. Results

We used the algorithm and ran 100 simulations on both strategies of direct selling and gradual selling, starting with 10000 shares on oil, and the initial value of the portfolio for both cases is \$813,400.

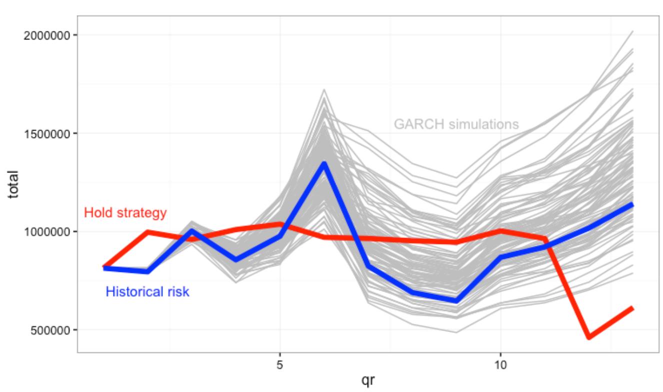

Figure 4. Gradual Selling

Figure 4 shows the result of selling the dominant asset gradually, in this case, selling 10% at most each time when rebalancing the portfolio. The red line represents the result of the holding strategy, and the blue line shows the result of using the historical covariance matrix instead of CCC-GARCH predicted covariance matrices. The minimum final wealth we got from the simulation for gradual selling is $730,784, the maximum is $1,925,896, and the mean is $1,244,182.

Figure 5 shows the results of selling the dominant asset directly to its optimal weight suggested by portfolio optimization without any other constraint. Similarly to Figure 4, the red line represents the result of the holding strategy, and the blue line shows the result of using the historical covariance matrix instead of CCC-GARCH predicted covariance matrices. The minimum final wealth we got from the simulation for gradual selling is $817,941, the maximum is $2,336,128, and the mean is $1,461,395.

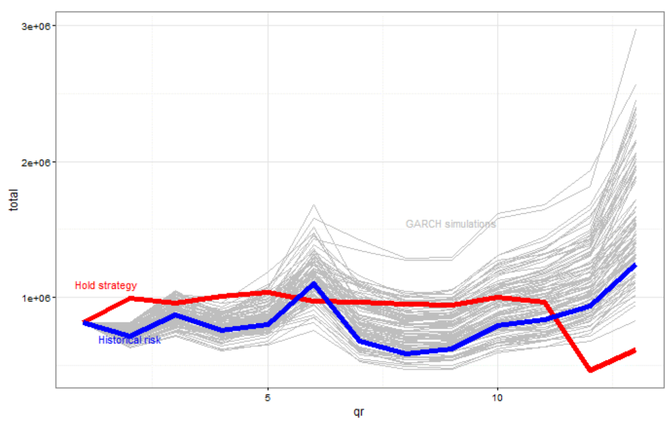

Figure 5. Immediate Selling

The results above are based on the time frame we selected, from September 2005 to September 2015, the five particular assets we chose, and the restrictions we put on the asset weights. The results of simulations are extremely sensitive to the selected time frame and assets. Given the time period, four assets, and a dominant asset and a risk-free asset we chose, based on the simulations, we conclude that selling the dominant asset directly to its optimal weight without any constraint may give investors higher return than gradual selling and holding strategies. However, different historical conditions, asset choices and constraints might well result in a substantially different outcome.

4. Conclusion and Discussion of Possible Future Work

As stated before, the major objective of the thesis is finding possible optimal strategies to solve the problem of reallocating asymmetrical wealth. Instead of giving a universal answer to such problem, we have developed an algorithm that help investors to explore the solutions based on different situations. Investors can use these tools to explore the likelihood of success of different management strategies under a variety of assumptions. In reality, different investors may have various financial situations such as different risk tolerance, so there is a need to change the constraints in the algorithm to make it applicable to particular investors.

There are still improvements that can be made in the algorithm. First of all, more advanced multivariate GARCH models can be applied in the forecast part to predict the covariance matrix of the next term. For example, Dynamic Conditional Correlation GARCH models, as known DCC-GARCH [1] make the conditional correlation matrix time-dependent instead of assuming it to be constant as CCC-GARCH does. It could possibly improve the quality of modeling to get more accurate forecasts. Secondly, a wider range of investment tools could be considered among the possible assets for a portfolio. We used five specific assets for the simulations, but in the financial market, there are more investment tools such as mutual funds can be used to reach higher returns or lower risks.

References

1. L. Bauwens, S. Laurent, and J. Rombouts. Multivariate GARCH models: a survey. *Journal of Applied Econometrics* **21** (2006): 79–109.
2. T. Bollerslev. Modeling the coherence in short-run nominal exchange rates: A multivariate generalized ARCH model. *Review of Economics and Statistics* **72** (1990), 498–505.
3. A. Canty and B. Ripley. `boot`: Bootstrap `R` (S-Plus) Functions. R package version 1.3-18 (2016).
4. R. F. Engle. Autoregressive Conditional Heteroscedasticity with Estimates of Variance of United Kingdom Inflation. *Econometrica* **50** (1982): 987–1007.
5. Federal Reserve Economic Data. `https://research.stlouisfed.org/fred2/` (downloaded September, 2015).
6. H. S. Friday et al. Tactical Asset Allocation for Investors with Asymmetrical Wealth Endowments. *Review of Business Research* **6** No. 5 (2006), 9–18.
7. S. Horan and A. Zaman. Tax-Adjusted Portfolio Optimization and Asset Location: Extensions and Synthesis. *Journal of Wealth Management* **11**:3 (2008).
8. H. M. Markowitz *Portfolio Selection: Efficient Diversification of Investments*, 2^{nd} ed. New York: Basil Blackwell, 1991, ISBN 978-1-55786-108-5.
9. T. Nakatani. `ccgarch`: R Package for Modeling Multivariate GARCH Models with Conditional Correlations, `R` package version 0.2.3. (2014)
10. R. F. Posedel. Properties and Estimation of GARCH(1,1) Model. *Metodološki zvezki*, **2**, No. 2, (2005), 243-257.
11. R Core Team. `R`: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. (2016)
12. W. F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance* **19** (1964): 425–442.
13. B. A. Turlach and A. Weingessel. `quadprog`: Functions to solve Quadratic Programming Problems. `R` package version 1.5-5. (2013)

14. D. Wuertz, Y. Chalabi, with contribution from M. Miklovic, C. Boudt, P. Chausse and others. `fGARCH`: Rmetrics - Autoregressive Conditional Heteroskedastic Modeling. R package version 3010.82, (2013).

Appendix: R code

See following pages.

```
library(boot)
library(ggplot2)
library(quadprog)
library(xts)
library(zoo)
library(timeSeries)
library(fGarch)
library(ccgarch)

## new lines
out.mx.big.gradual <- matrix(0, 1300, 7)
colnames(out.mx.big.gradual) <- c("DJ", "oil", "NASDAQ", "gas", "REIT", "qr", "sim.no")
out.mx.big.gradual <- data.frame(out.mx.big.gradual)
out.mx.big.gradual$qr <- rep(1:13, 100)
out.mx.big.gradual$sim.no <- sort(rep(1:100, 13))
m <- 1

#For the result of using historical covariance matrix
# out.mx.hist.risk <- matrix(0, 13, 7)
# colnames(out.mx.hist.risk) <- c("DJ", "oil", "NASDAQ", "gas", "REIT", "qr", "sim.no")
# out.mx.hist.risk <- data.frame(out.mx.hist.risk)
# out.mx.hist.risk$qr <- 1:13
# out.mx.hist.risk$sim.no <- factor(rep(1, 13))
# m <- 1

out.mx.big.gradual <- read.csv("outMxBigGradual.csv", row.names = 1)
m <- min((1:1300)[out.mx.big.gradual[,2]==0])
first.bb <- 1 + (m - 1) %/% 13

for(bb in first.bb:100) {
# for (bb in 1:1) { # for historical risk


#Start to build up the dataset.
dates <- DJIA_10y$DATE
#Convert factor to date
dates <- as.Date(dates)
#Delete the last day because we cannot compute the return of last day
dates <- head(dates,-1)

values_DJIA <- DJIA_10y$VALUE
#Convert factor to numeric
values_DJIA <- as.numeric(levels(values_DJIA))[values_DJIA]
return_DJIA <- diff(log(values_DJIA))

#Construct time series xts object
data <- as.xts(return_DJIA,order.by = as.Date(dates,"%d/%m/%Y"))
names(data)[1] <- "DJIA"

#input the data of oil, nasdaq index, and natural gas
values_oil <- DCOILWTICO_10y$VALUE
values_nasdaq <- NASDAQ100_10y$VALUE
values_gas <- DHHNGSP_10y$VALUE
values_reits <- WILLREITIND_10y$VALUE
values_bond <- TBond$VALUE

values_oil <- as.numeric(levels(values_oil))[values_oil]
values_nasdaq <- as.numeric(levels(values_nasdaq))[values_nasdaq]
values_gas <- as.numeric(levels(values_gas))[values_gas]
values_reits <- as.numeric(levels(values_reits))[values_reits]
values_bond <- as.numeric(levels(values_bond))[values_bond]


return_oil <- diff(log(values_oil))
return_nasdaq <- diff(log(values_nasdaq))
return_gas <- diff(log(values_gas))
return_reits <- diff(log(values_reits))
return_bond <- (1+values_bond/100)^(1/365)-1

#construct time series
data <- transform(data,oil = return_oil)
data <- transform(data,nasdaq = return_nasdaq)
data <- transform(data,gas = return_gas)
data <- transform(data,reits = return_reits)
data <- transform(data,bond = return_bond)


#convert it to object for fPortfolio
data_ts <- as.timeSeries(data)
```

```
#Remove NAs
data_port <- na.omit(data_ts)

#data_port has 2409 daily observations
#Use 1200(5 years) as starting base, increment is 93(3 months), 13 intervals of integers
tangentpoints <- c()
moneypoints <- c()

DJIA_money <- c()
DJIA_share <- c()
DJIA_buy_price <- c()
DJIA_profit <- c()

DJIA_money[1] <- 0
DJIA_share[1] <- 0
DJIA_buy_price[1] <- 0
#############################################################
oil_money <- c()
oil_share <- c()
oil_buy_price <- c()
oil_profit <- c()

oil_money[1] <- 0
oil_share[1] <- 0
oil_buy_price[1] <- 0
#############################################################
nasdaq_money <- c()
nasdaq_share <- c()
nasdaq_buy_price <- c()
nasdaq_profit <- c()

nasdaq_money[1] <- 0
nasdaq_share[1] <- 0
nasdaq_buy_price[1] <- 0
#############################################################
reits_money <- c()
reits_share <- c()
reits_buy_price <- c()
reits_profit <- c()

reits_money[1] <- 0
reits_share[1] <- 0
reits_buy_price[1] <- 0
#############################################################
gas_money <- c()
gas_share <- c()
gas_buy_price <- c()
gas_profit <- c()

gas_money[1] <- 0
gas_share[1] <- 0
gas_buy_price[1] <- 0
#############################################################
# bond_money <- c()
# bond_profit <- c()
#
# bond_money[1] <- 0
#############################################################


tax <- c()
tax[1] <- 0
DJIA_profit[1]=0
nasdaq_profit[1]=0
oil_profit[1]=0
gas_profit[1]=0
reits_profit[1]=0

#Initial oil price
initialOil <- 65.98
InitialOilshare <- 10000

basis_oil <- c()
basis_oil[1] <- 7
oil_money[1] <- 10000*81.34
oil_share[1] <- 10000
oil_buy_price[1] <- 81.34
## new line
out.mx.big.gradual[m, 1:5] <- c(0, oil_money[1], 0, 0, 0); m <- m+1

#For the result of using historical covariance matrix
```

```
# out.mx.hist.risk[m, 1:5] <- c(0, oil_money[1], 0, 0, 0); m <- m+1


basis_DJIA <- c()
basis_nasdaq <- c()
basis_gas <- c()
basis_reits <- c()




for(i in 2:13)
{


  j <- (1:nrow(DJIA_10y))[rownames(data_ts) == rownames(data_port[(1200 + (i-2)*93),0])]
  print(paste("Start of loop for i =", i))
  print(paste("Value to reallocate is ", DJIA_share[i-1]*values_DJIA[j] + oil_share[i-1]*values_oil[j] +
nasdaq_share[i-1]*values_nasdaq[j] + gas_share[i-1]*values_gas[j] + reits_share[i-1]*values_reits[j]))
  money <- DJIA_share[i-1]*values_DJIA[j] + oil_share[i-1]*values_oil[j] + nasdaq_share[i-
1]*values_nasdaq[j] + gas_share[i-1]*values_gas[j] + reits_share[i-1]*values_reits[j]

  ##Moving window
  all <- data_port[(1 + (i-2)*93):(1200 + (i-2)*93),]
  riskfree <- mean(all[,6])
  trainset <- data_port[(1 + (i-2)*93):(1200 + (i-2)*93),1:5]
#   Risk.mx <- cov(trainset)
  Return.v <- apply(trainset, 2, mean)

  #######################Multivariate Garch################################
  fitD <- garchFit(data = trainset$DJIA)
  fitO <- garchFit(data = trainset$oil)
  fitN <- garchFit(data = trainset$nasdaq)
  fitG <- garchFit(data = trainset$gas)
  fitR <- garchFit(data = trainset$reits)

  A <- diag(x =
c(as.numeric(fitD@fit$coef["alpha1"]),as.numeric(fitO@fit$coef["alpha1"]),as.numeric(fitN@fit$coef["alpha1"])

           nrow = 5, ncol = 5)

  B <- diag(x =
c(as.numeric(fitD@fit$coef["beta1"]),as.numeric(fitO@fit$coef["beta1"]),as.numeric(fitN@fit$coef["beta1"]),as

           nrow = 5, ncol = 5)
  R <- cov2cor(cov(trainset))
  a <- matrix(data =
c(as.numeric(fitD@fit$coef["omega"]),as.numeric(fitO@fit$coef["omega"]),as.numeric(fitN@fit$coef["omega"]),as

              nrow = 5, ncol = 1)

  estimation <- eccc.estimation(a,A,B,R,trainset,model = "diagonal")

  simulation <- eccc.sim(nobs =
100,estimation$para.mat$a,estimation$para.mat$A,estimation$para.mat$B,estimation$para.mat$R,model =
"diagonal")
  simulated_DJIA <- fitD@fit$par[1] + simulation$eps[,1]
  simulated_oil <- fitO@fit$par[1] + simulation$eps[,2]
  simulated_nasdaq <- fitN@fit$par[1] + simulation$eps[,3]
  simulated_gas <- fitG@fit$par[1] + simulation$eps[,4]
  simulated_reits <- fitR@fit$par[1] + simulation$eps[,5]

  forecast <- cbind(simulated_DJIA,simulated_oil,simulated_nasdaq,simulated_gas,simulated_reits)

  Risk.mx <- cov(forecast)


#   h: a matrix of the simulated conditional variances (T \times N)
#   eps: a matrix of the simulated time series with (E)CCC-GARCH process (T \times N)
#################################################################################################





####################################################################################
  A <- matrix(1,1,5)
  B <- matrix(data = c(0,1,0,0,0),1,5)
  D <- diag(5)
  Amat <- rbind(A,B,D,-D)
```

```
  f <- c(1,0.9 - (i-2)*0.1,rep(-1, 5),rep(-1, 5))
  #For immediate selloff we use the constraints
  # f <- c(1,rep(-1, 5),rep(-1, 5))

  sol <- solve.QP(Dmat=Risk.mx,dvec = rep(0,5), Amat=t(Amat), bvec=f, meq=1)


#   #Not allow short selling
#   D <- rbind(A,B,diag(5),-diag(5))
#   f <- c(1,0.9,rep(0,5),rep(-1,5))

return.at.min.risk <- sum(sol$solution*Return.v)


#Use linea programming to find the highest possible return, in selloff strategy remove the constraints of
the selling limit.
a <- c(Return.v[1], -Return.v[1], Return.v[2], Return.v[3], -Return.v[3],
       Return.v[4], -Return.v[4], Return.v[5], -Return.v[5])

A1 <- matrix(c(1, -1, 0, 0, 0, 0, 0, 0, 0,
               0, 0, 1, 0, 0, 0, 0, 0, 0,
               0, 0, 0, 1, -1, 0, 0, 0, 0,
               0, 0, 0, 0, 0, 1, -1, 0, 0,
               0, 0, 0, 0, 0, 0, 0, 1, -1), 5, 9, byrow=T)
b1 <- c(1, 1, 1, 1, 1)
A2 <- matrix(c(1, -1, 0, 0, 0, 0, 0, 0, 0,
               0, 0, 1, 0, 0, 0, 0, 0, 0,
               0, 0, 0, 1, -1, 0, 0, 0, 0,
               0, 0, 0, 0, 0, 1, -1, 0, 0,
               0, 0, 0, 0, 0, 0, 0, 1, -1), 5, 9, byrow=T)
b2 <- c(-1, 0.9 - (i-2)*0.1, -1, -1, -1)
A3 <- matrix(c(1, 1, 1, 1, 1, 1, 1, 1, 1), 1, 9, byrow=T)
b3 <- 1

bang<-simplex(a, A1, b1, A2, b2, A3, b3, maxi=T)

Return.seq <- seq(sum(sol$solution*Return.v), bang$value*0.9, length.out=250)
E <- matrix(data=Return.v, 1, 5)
Amat <- rbind(A,B,D,-D,E)
plot.mx <- matrix(0, length(Return.seq), 3)
colnames(plot.mx) <- c("Risk", "Return", "Sharpe.ratio")
plot.mx[1,] <- c(sqrt(sol$value), sum(sol$solution*Return.v),
                 (sum(sol$solution*Return.v)-riskfree)/sqrt(sol$value))

for (k in 1:length(Return.seq))
{
#  print(k)
  f <- c(1,0.9 - (i-2)*0.1,rep(-1, 5),rep(-1, 5), Return.seq[k])
  sol <- solve.QP(Dmat=Risk.mx,dvec = rep(0,5), Amat=t(Amat), bvec=f, meq=1)
  plot.mx[k,] <- c(sqrt(sol$value), sum(sol$solution*Return.v),
                   (sum(sol$solution*Return.v)-riskfree)/sqrt(sol$value))
}

plot.df <- data.frame(plot.mx)
Risk.free.df <- data.frame(Risk=0,Return=riskfree)

Sharpe.ratio.df <- data.frame(rbind(Risk.free.df,
                                    plot.df[(1:nrow(plot.df))
[plot.df$Sharpe.ratio==max(plot.df$Sharpe.ratio)],][c("Risk","Return")]))



ggplot(plot.df, aes(x=Risk, y=Return)) + geom_line() + theme_bw() +
  geom_point(data=Risk.free.df, aes(x=Risk, y=Return)) +
  geom_line(data=Sharpe.ratio.df, aes(x=Risk, y=Return), colour="red")


find.best.sharpe.ratio <- function(Return.param) {
  f <- c(1,0.9 - (i-2)*0.1,rep(-1, 5),rep(-1, 5), Return.param/10000)
  sol <- solve.QP(Dmat=Risk.mx,dvec = rep(0,5), Amat=t(Amat), bvec=f, meq=1)
  return((sum(sol$solution*Return.v)-riskfree)/sqrt(sol$value))
}

low.bd <- 0.9*return.at.min.risk+0.1*bang$value
high.bd <- 0.1*return.at.min.risk+0.9*bang$value
thud <- optimize(f=find.best.sharpe.ratio, interval=c(low.bd*10000, high.bd*10000), maximum=T)

best.Return <- thud$maximum/10000
f <- c(1,0.9 - (i-2)*0.1,rep(-1, 5),rep(-1, 5), best.Return)
sol <- solve.QP(Dmat=Risk.mx,dvec = rep(0,5), Amat=t(Amat), bvec=f, meq=1)
```

```
################################################################################
  DJIA_money[i]  <- money*sol$solution[1]
  oil_money[i]  <- money*sol$solution[2]
  nasdaq_money[i]  <- money*sol$solution[3]
  gas_money[i]  <- money*sol$solution[4]
  reits_money[i]  <- money*sol$solution[5]




  #Search assets prices

  j <- (1:nrow(DJIA_10y))[rownames(data_ts) == rownames(data_port[(1200 + (i-2)*93),0])]

  DJIA_buy_price[i] <- values_DJIA[j]
  oil_buy_price[i] <- values_oil[j]
  nasdaq_buy_price[i] <- values_nasdaq[j]
  gas_buy_price[i] <- values_gas[j]
  reits_buy_price[i] <- values_reits[j]

  DJIA_share[i] <- DJIA_money[i]/DJIA_buy_price[i]
  oil_share[i] <- oil_money[i]/oil_buy_price[i]
  nasdaq_share[i] <- nasdaq_money[i]/nasdaq_buy_price[i]
  gas_share[i] <- gas_money[i]/gas_buy_price[i]
  reits_share[i] <- reits_money[i]/reits_buy_price[i]


  ##########################################################



#Compute the profit of each asset for tax use.
################################################################################
    profit.f <- function(s_price, s_shares, e_price, e_shares)
    {
      if(e_shares < s_shares && s_shares > 0)
      {
        profit = (s_shares - max(c(0, e_shares)))*(e_price-s_price)
      }

      else if(e_shares > s_shares && s_shares < 0)
      {
        profit = (min(c(0, e_shares)) - s_shares)*(s_price - e_price)
      }

      else
      {
        profit = 0
      }

      profit

    }
  ################################################################################


#Tax Adjustments
  if (i == 2)
  {
    basis_DJIA[i-1] <- DJIA_buy_price[i-1]
    basis_nasdaq[i-1] <- nasdaq_buy_price[i-1]
    basis_gas[i-1] <- gas_buy_price[i-1]
    basis_reits[i-1] <- reits_buy_price[i-1]
  }

   DJIA_profit[i] <- profit.f(basis_DJIA[i-1],DJIA_share[i-1],DJIA_buy_price[i],DJIA_share[i])
   oil_profit[i] <- profit.f(basis_oil[i-1],oil_share[i-1],oil_buy_price[i],oil_share[i]) ##basis of oil 10%
of market price
   nasdaq_profit[i] <- profit.f(basis_nasdaq[i-1],nasdaq_share[i-1],nasdaq_buy_price[i],nasdaq_share[i])
   gas_profit[i] <- profit.f(basis_gas[i-1],gas_share[i-1],gas_buy_price[i],gas_share[i])
   reits_profit[i] <- profit.f(basis_reits[i-1],reits_share[i-1],reits_buy_price[i],reits_share[i])

   tax[i] <- (DJIA_profit[i] + nasdaq_profit[i] + gas_profit[i] + oil_profit[i] + reits_profit[i])*0.25

   taxA <- matrix(nrow = 6, ncol = 5)
```

```
   taxA[1,] = c(DJIA_buy_price[i]*(1-sol$solution[1]),-sol$solution[1]*oil_buy_price[i],
                -sol$solution[1]*nasdaq_buy_price[i], -sol$solution[1]*gas_buy_price[i],
                -sol$solution[1]*reits_buy_price[i])

   taxA[2,] = c(-sol$solution[2]*DJIA_buy_price[i],oil_buy_price[i]*(1-sol$solution[2]),
                -sol$solution[2]*nasdaq_buy_price[i], -sol$solution[2]*gas_buy_price[i],
                -sol$solution[2]*reits_buy_price[i])

   taxA[3,] = c(-sol$solution[3]*DJIA_buy_price[i],
                -sol$solution[3]*oil_buy_price[i],nasdaq_buy_price[i]*(1-sol$solution[3]),
                -sol$solution[3]*gas_buy_price[i],-sol$solution[3]*reits_buy_price[i])

   taxA[4,] = c(-sol$solution[4]*DJIA_buy_price[i],-sol$solution[4]*oil_buy_price[i],
                -sol$solution[4]*nasdaq_buy_price[i],gas_buy_price[i]*(1-sol$solution[4]),
                -sol$solution[4]*reits_buy_price[i])

   taxA[5,] = c(-sol$solution[5]*DJIA_buy_price[i],-sol$solution[5]*oil_buy_price[i],
                -sol$solution[5]*nasdaq_buy_price[i],-sol$solution[5]*gas_buy_price[i],
                reits_buy_price[i]*(1-sol$solution[5]))

   taxA[6,] = c(DJIA_buy_price[i],oil_buy_price[i],nasdaq_buy_price[i],gas_buy_price[i],reits_buy_price[i])

   afterTax <-
DJIA_buy_price[i]*DJIA_share[i]+oil_buy_price[i]*oil_share[i]+nasdaq_buy_price[i]*nasdaq_share[i]+
     gas_buy_price[i]*gas_share[i]+reits_buy_price[i]*reits_share[i] - tax[i]


   taxb = matrix(c(0,0,0,0,0,afterTax),nrow = 1, ncol = 6)


  endshares <- qr.solve(taxA,t(taxb))

  #Updates share with its real shares
  DJIA_share[i] <- endshares[1]
  oil_share[i] <- endshares[2]
  nasdaq_share[i] <- endshares[3]
  gas_share[i] <- endshares[4]
  reits_share[i] <- endshares[5]
 #Calculating basis

  if(abs(DJIA_share[i]) > abs(DJIA_share[i-1]) && prod(DJIA_share[i],DJIA_share[i-1]) > 0)
  {
    basis_DJIA[i] <- (DJIA_share[i-1]*DJIA_buy_price[i-1]+(DJIA_share[i]-DJIA_share[i-
1])*DJIA_buy_price[i])/(DJIA_share[i])
  }else if(prod(DJIA_share[i],DJIA_share[i-1]) <= 0)
  {
    basis_DJIA[i] <- DJIA_buy_price[i]
  }else
  {
    basis_DJIA[i] <- basis_DJIA[i-1]
  }

#################################
  if(abs(oil_share[i]) > abs(oil_share[i-1]) && prod(oil_share[i],oil_share[i-1]) > 0)
  {
    basis_oil[i] <- (oil_share[i-1]*oil_buy_price[i-1]+(oil_share[i]-oil_share[i-
1])*oil_buy_price[i])/(oil_share[i])
  }else if(prod(oil_share[i],oil_share[i-1]) <= 0)
  {
    basis_oil[i] <- oil_buy_price[i]
  }else
  {
    basis_oil[i] <- basis_oil[i-1]
  }
###############################################################################
  if(abs(nasdaq_share[i]) > abs(nasdaq_share[i-1]) && prod(nasdaq_share[i],nasdaq_share[i-1]) > 0)
  {
    basis_nasdaq[i] <- (nasdaq_share[i-1]*nasdaq_buy_price[i-1]+(nasdaq_share[i]-nasdaq_share[i-
1])*nasdaq_buy_price[i])/(nasdaq_share[i])
  }else if(prod(nasdaq_share[i],nasdaq_share[i-1]) <= 0)
  {
    basis_nasdaq[i] <- nasdaq_buy_price[i]
  }else
  {
    basis_nasdaq[i]<- basis_nasdaq[i-1]
  }
###############################################################################
  if(abs(gas_share[i]) > abs(gas_share[i-1]) && prod(gas_share[i],gas_share[i-1]) > 0)
  {
    basis_gas[i] <- (gas_share[i-1]*gas_buy_price[i-1]+(gas_share[i]-gas_share[i-
1])*gas_buy_price[i])/(gas_share[i])
```

```
  }else if(prod(gas_share[i],gas_share[i-1]) <= 0)
  {
    basis_gas[i] <- gas_buy_price[i]
  }else
  {
    basis_gas[i] <- basis_gas[i-1]
  }
################################################################################
  if(abs(reits_share[i]) > abs(reits_share[i-1]) && prod(reits_share[i],reits_share[i-1]) > 0)
  {
    basis_reits[i] <- (reits_share[i-1]*reits_buy_price[i-1]+(reits_share[i]-reits_share[i-
1])*reits_buy_price[i])/(reits_share[i])
  }else if(prod(reits_share[i],reits_share[i-1]) <= 0)
  {
    basis_reits[i] <- reits_buy_price[i]
  }else
  {
    basis_reits[i] <- basis_reits[i-1]
  }




##############################################################################################################



  #########################################################
  #search the asset price after 3 months
  k <- (1:nrow(DJIA_10y))[rownames(data_ts) == rownames(data_port[(1200 + (i-1)*93),0])]

  DJIA_sell_price <- values_DJIA[k]
  gas_sell_price <- values_gas[k]
  nasdaq_sell_price <- values_nasdaq[k]
  oil_sell_price <- values_oil[k]
  reits_sell_price <- values_reits[k]

  moneypoints[i] =
DJIA_sell_price*endshares[1]+oil_sell_price*endshares[2]+nasdaq_sell_price*endshares[3]+gas_sell_price*endsha


  money = moneypoints[i]

  col.names <- c("start.shares", "start.basis", "start.money", "start.weights",
                 "weights.this.qr", "end.shares", "profit", "end.money", "end.basis")
  debug.mx <- matrix(0,5,length(col.names))
  rownames(debug.mx) <- c("DJ", "oil", "NASDAQ", "gas", "reits")
  colnames(debug.mx) <- col.names
  debug.mx[,1] <- c(DJIA_share[i-1], oil_share[i-1], nasdaq_share[i-1],
                    gas_share[i-1], reits_share[i-1])
  debug.mx[,2] <- c(basis_DJIA[i-1], basis_oil[i-1], basis_nasdaq[i-1],
                    basis_gas[i-1], basis_reits[i-1])
  debug.mx[,3] <- c(DJIA_money[i-1], oil_money[i-1], nasdaq_money[i-1],
                    gas_money[i-1], reits_money[i-1])
  debug.mx[,4] <- debug.mx[,3]/sum(debug.mx[,3])
  debug.mx[,5] <- sol$solution
  debug.mx[,6] <- endshares
  debug.mx[,7] <- c(DJIA_profit[i], oil_profit[i], nasdaq_profit[i],
                    gas_profit[i], reits_profit[i])
  debug.mx[,8] <-
c(DJIA_buy_price[i]*endshares[1],oil_buy_price[i]*endshares[2],nasdaq_buy_price[i]*endshares[3],gas_buy_price

  debug.mx[,9] <- c(basis_DJIA[i], basis_oil[i], basis_nasdaq[i],
                    basis_gas[i], basis_reits[i])
  print(paste("At end of rebalancing for i =", i))
  print(debug.mx)
  print(paste("Value after rebalancing and taxes:", sum(debug.mx[,8])))
  ## new line
  out.mx.big.gradual[m,1:5] <- debug.mx[,8]; m <- m + 1
  ## historical risk
  ## out.mx.hist.risk[i, 1:5] <- debug.mx[,8]
}

  print(paste("we have finished iteration ", bb))
  write.csv(out.mx.big.gradual, "outMxBigGradual.csv")

}

plot(moneypoints,type = "l")
```

```
#Plot the wealth of each simulation period.
# Calculating the hold strategy for each asset
hold.strategy <- matrix(0, 13, 6)
colnames(hold.strategy) <- c("DJ", "oil", "NASDAQ", "gas", "REIT", "qr")
hold.strategy <- as.data.frame(hold.strategy)
hold.strategy$qr <- 1:13
for (i in 1:13) {
  j <- (1:nrow(DJIA_10y))[rownames(data_ts) == rownames(data_port[(1200 + (i-1)*93),0])]
  hold.strategy[i, "DJ"] <- values_DJIA[j]
  hold.strategy[i, "oil"] <- values_oil[j]
  hold.strategy[i, "NASDAQ"] <- values_nasdaq[j]
  hold.strategy[i, "gas"] <- values_gas[j]
  hold.strategy[i, "REIT"] <- values_reits[j]
}
for (j in c(1,3,4,5)) {
  hold.strategy[,j] <- hold.strategy[1,2]*10000*hold.strategy[,j]/hold.strategy[1,j]
}
hold.strategy[,2] <- hold.strategy[,2]*10000
hold.strategy$sim.no <- factor(rep(1, 13))

out.mx.big.gradual$total <- apply(out.mx.big.gradual[,1:5],1,sum)
#For the result of using historical covariance
# out.mx.hist.risk$total <- apply(out.mx.hist.risk[,1:5],1,sum)


ggplot(data=out.mx.big.gradual, aes(x=qr, y=total, group=as.factor(sim.no))) + geom_line(colour="grey") +
theme_bw() + geom_line(data=hold.strategy, aes(x=qr, y=oil, group=sim.no), colour="red", size=2) +
geom_line(data=out.mx.hist.risk, aes(x=qr, y=total, group=sim.no), colour="blue", size=2) + annotate("text",
label = "Hold strategy", x = 1.5, y = 1100000, size = 4, colour = "red") + annotate("text", label =
"Historical risk", x = 2, y = 700000, size = 4, colour = "blue") + annotate("text", label = "GARCH
simulations", x = 9, y = 1550000, size = 4, colour = "grey")
```