Advanced Analytics II Final Project

Sovik Kumar Nath

University of South Florida

Abstract

There is ongoing trend on application of data mining, computation and statistical tools for financial services and risk management. This paper discusses the application of data mining tools and R language for three different applications related to portfolio optimization and financial risk management.The applications are for portfolio optimization, asset pricing models and credit risk management. For the portfolio optimization, a portfolio of energy stocks were created and the optimal portfolio was determined using Markowitz mean-variance model. Quadratic programming and lagrange multipliers were applied to determine the optimal portfolio for Markowitz mean-variance model. The optimal portfolio were determined for three cases, equal weights of stock, no short sales i.e. all weights greater than zero and portfolio return equals that of a particular stocks' return. In the second analysis, capital asset pricing model (CAPM) was applied to one of the energy stocks (Exxon) and beta i.e. the sensitivity of the stock to market (SP500) was determined using linear regression model. For both the analyses, the time period was set between January 2010 and December 2014 . In the third analysis, credit risk management was studied for which credit default risk was analyzed using structural models and Merton's options based model. The firm value at maturity was determined applying Brownian motion and the results from this was used to calculate the price of risky debt using Black Scholes formula for pricing of European call option. In these three methods, data mining and R tools were used to compute the results.

Keywords: Portfolio optimization, Markowitz mean-variance model, quadratic programming beta, linear regression, credit defaults, Merton option pricing model, Black-Scholes options pricing formula

1. Background and Methods

1.1. ***Portfolio Optimization***

The concept of portfolio optimization was put forward by harry Markowitz. According to the portfolio theory, risk is measured by variance. Based on the correlation between the investment products, one can reduce the risk of the portfolio and get the desired return. The variance of the affine or convex combination of two random variables can be expressed as,

$$f(a) = Var(\alpha X + (1-\alpha) Y)$$
$$f(a) = \alpha^2 \sigma_x^2 + (1-\alpha)^2 \sigma_y^2 + 2\alpha(1-\alpha) Cov(X, Y)$$

X, Y: are random variables

σ_x, σ_y: are variances of X and Y

Mean Variance Model

The mean variance model (Markowitz 1952) is a similar model of higher dimensions. The terms which are associated with the mathematical formulation of the model are as follows,

- Asset X_i : random variable with finite variance
- Portfolio : the combination of assets, P $= \sum w_i X_i$ and $\sum w_i = w_i \vec{1} = 1$. If the combination is affine, there is no restriction on weights and if it is convex, then it is non negative.
- The optimization problem is thus the process of choosing best coefficients w_i such that the requirement of the investor is fulfilled, the portfolio has a minimum risk at a given level of return, μ or it has the maximum return at the given level of risk σ.

Let there be N random variables X_i $i = 1, ..., N$ with weights w_i $i = 1, ..., N$ having expected returns r $= (EX_1, EX_2, ..., EX_N)$. Then, the expected return of the portfolio is given by wr $= \sum w_i r_i$. The covariance matrix of the variables is given by,

$$Q = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1N} \\ \vdots & \ddots & \vdots \\ \sigma_{N1} & \cdots & \sigma_{NN} \end{bmatrix}$$

$$\sigma_{ii} = \sigma_i^2 \quad \text{and} \quad \sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$$

Thus the expected return and the expected variance of the portfolio is given by,

$$\mu = wr \quad \text{and} \quad \sigma_p^2 = w^T Q w$$

Based on Markovitz Theory, the investors problem is a constrained minimization problem, i.e., he seeks to minimize the risk for a given value of return,

$$\min \quad \sigma_p^2 = w^T Q w \quad \text{s.t.} \quad wr = \mu \quad \text{and} \quad w\vec{1} = 1 \;\; \ldots (1.1)$$

Lagrange Multipliers

The optimization problem is a quadratic problem. This can be solved using Lagrange Theorem.The Method of Lagrange multipliers (named after Joseph Louis Lagrange) is a strategy for finding the local maxima and minima of a function subject to equality constraints. For an optimization problem given by maximize f(a), s.t. $g_i(a) = c_i$.Both f and g are first partial derivatives. A set of new variables (λ_i) called a Lagrange multiplier are introduced. The Lagrange function (or Lagrangian) defined by,

$$L = f(a) + \sum \lambda_i \left(g_i(a) - c_i\right)$$

,

where the λ_i may be either added or subtracted. Then there exists λ_i such that the partial derivatives for the Lagrange function (L) are zero.

Using this, the quadratic function of the optimization problem leads to the problem of solving a linear system of equations. Thus the Lagrange function for portfolio optimization problem and the equivalent system of linear equations are as follows.

$$L\left(w, \lambda_1, \lambda_2\right) = \frac{1}{2} w^T Q w + \lambda_1 \left(wr - \mu\right) + \lambda_2 \left(w\vec{1} - 1\right)$$

$$\begin{bmatrix} Q & \vec{1} & \vec{r} \\ \vec{1} & 0 & 0 \\ \vec{r} & 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \mu \end{bmatrix} \ldots (1.2)$$

1.2. ***Asset Pricing Models***

Asset pricing models deal with the problem of absolute pricing (Cochrane 2005) and how to determine the value of assets with uncertain payments based on their risks. According to the concept of relative pricing, the riskiness of the underlying product is captured in its price and thus does not pay any additional role in the pricing of its derived instrument. The no arbitrage argument forces consistency in the prices of the derivatives and the underlying assets. Two types of assets are used to present the relationship between the asset return and the risk factor - Capital Asset Pricing Model (CAPM) and Arbitrage Pricing Theory (APT).

Capital Asset pricing Model (CAPM)

Capital Asset Pricing Model explains the pricing of assets based on economic considerations. CAPM tries to find answer to market conditions and is based on market equilibrium. According Sharpe (1964) and Lintner (1965), the existence of the market equilibrium is based on number of assumptions, tabulated below,

- Individual investors are risk takers
- Single Period Investment horizon
- Investments are limited to traded financial assets
- No taxes and no transaction costs
- Information is costless and available to all investors
- Investors are rational mean-variance optimizers
- Homogeneous expectation

In an environment where these assumptions are held and the investors are aware of assumptions, all the investors will held the same portfolio of risky assets which is the market portfolio. The market portfolio consists of all securities and the proportion of each security is its market value as a percentage of the total market value. The risk premium is a function of risk aversion of all market participants. The resulting equilibrium is a linear relationship between market risk premium and individual security's risk i.e.,

Individual security's risk = Function of Market Risk Premium

$$E(r_i) - r_f = \beta_i \left(E(r_m) - r_f\right) \ldots (2.1)$$

$E(r_i)$: expected return of a certain security

r_f: risk free return

$E(r_m)$: expected return of a market portfolio

β_i: security's covariance with market and variance of the market's returns

The risk in CAPM is measured by beta β_i, which is function of individual security's covariance with the market and the variance of market return, i.e.,

$$\beta_i = \frac{Cov_{i,m}}{Var_m} \ldots (2.2)$$

$Cov_{i,m}$: Covariance between security's return and the market return

Var_m: Variance of market return

There are a number of interpretation of beta.

- It shows the sensitivity of a stock's return to the return of the market portfolio
- A certain security's beta shows how much risk that security contributes to the market portfolio

Thus, CAPM states that the market gives a higher return only in case of a higher systematic risk (being a linear relationship). Equation 2.1 can be rearranged to obtain linear equation, termed as Security Market Line (SML),

$$E(r_i) = r_f + \beta_i \left(E(r_m) - r_f\right)$$

According to CAPM,

- Each security should be as the SML in equilibrium. Thus this equation holds for each security or portfolio even if it is not efficient.
- If this equation is not fulfilled, there is lack of equilibrium in the market. For example, if a security's return on the market is higher than it should be according to CAPM, then each investor needs to change the composition of his or her portfolio to decrease the security's return and thus fulfill the above equation. So, in case of CAPM, every investor needs to change the portfolio to get the fair price of a security and get equilibrium.

Beta Estimation

The sensitivity of a security towards a factor can be estimated from the past price movements. The log returns r_t of the stock and the market index and calculated as follows,

$$r_t = ln\frac{S_t}{S_{t-1}}$$

S_t: market price on day t of stock and market

r_t: log-returns of the stock and the market index

Risk premiums are determined by subtracting the risk-free daily log returns r_{ft}. LIBOR rates as used for risk-free rates,

$$r_{ft} = ln\left(1 + \frac{USDLIBOR}{36000} * (t+1) - t\}]\right)$$

$t, t-1$: refers to date difference in the number of days between the two closing values

Using r_{it}, r_{mt}, r_{ft}, the risk premiums are calculated as,

$$R_{it} = r_{it} - r_{ft}$$
$$R_{mt} = r_{mt} - r_{ft}$$

Method I

In order to estimate beta, one can use returns or risk premiums in the models as the parameters will differ (except for constant risk-free returns)

(Medvagyev-Szz 2010). This is especially true when using time series for estimation of beta. Using risk premiums, beta is estimated as,

$$\beta_i = \frac{Cov\left(R_{it}, R_{mt}\right)}{Var\left(R_{mt}\right)}$$

Method II

The second method of estimating beta is by using linear regression. Regression equation has the following form which is termed as Security Characteristic Line (SCL),

$$R_i = \alpha_i + \beta_i R_m + e_i$$

Ordinary least squared (OLS) estimation is used to determine the linear regression model. The intercept of the characteristic line is alpha (α). This is the part of the stock return unexplained by the market factor. The slope of the function shows the sensitivity of the stock towards the market factor which is beta (β).

1.3. ***Credit Risk Management***

Credit risk is defined as the the distribution of financial losses due to unexpected changes in quality of credit of a counterparty in a financial agreement (Giesecke 2004). According to literature, there are different types of credit risks such as default risk, downgrade risk and counterparty risk. Default risk is associated directly with the risk of non-performance of a claim or credit. Thus, default risk is one of the important components of managing credit risks. Measuring and predicting the probability of default or modelling the distribution of the financial loss due to default are important components which are considered by financial institutions as a part of credit risk management. There are different credit default models which give an overall picture of loss distributions and generating and pricing of a single debt instrument. Two such models are structural models and intensity models of which structural models are discussed here.

Structural Models

The options based model of Merton (Merton 1974) provides an introduction to structural model, an approach for credit default model. In this approach, the risky debt is evaluated as a contingent of the firm value. It is assumed that the firm value V, follows a geometric Brownian motion given by,

$$dV_t = \mu\, V_t + \sigma\, V_t\, dW_t$$

dV_t: the firm value at time t

μ: drift parameter

σ: volatility parameter (greater than zero)

dW: is differential of Wiener Process

V_0: initial asset value (greater than zero)

This model assumes a flat yield curve, with a constant interest rate r. The default state is the state when the value of the asset, V, falls below the liabilities (K) upon maturity of the debt (T). The firm value at maturity (V_T) is obtained from the integral,

$$dV_T = V_0 \exp\left(\int_0^T dlnV_t\right)$$

V_T: firm value at maturity

Using Ito's lemma, the differential of the logarithm of firm value ($dlnV_t$) can be derived as,

$$dlnV_t = \left(\mu - \frac{\sigma^2}{2}\right)\, dt\, + \sigma\, dW_t$$

Using a discrete approach by generating Gaussian distributed random variables, $\Delta W \sim \sqrt{\Delta t} N(0,1)$, the firm value at maturity can be obtained as,

$$V_T = \, V_0 \exp\left(\sum_{M}^{i=1}\left(\mu - \frac{\sigma^2}{2}\right)\, \Delta t\, + \sigma\, dW_t\right) \ldots (3.1)$$

Δt: one-period length of the elapsed time

Once the firm value has been obtained, the price of risky debt needs to be obtained. there are two methods for calculating the price of risky debt, using expected value of discounted payoff (Method I) and from Black-Scholes formula (Method II).

Method I

In this method the price of risky debt is calculated as the expected value of the discounted payoff of the risky debt at maturity as per the risk neurtal or martingale P measure, given by,

$$D = E^P \left(e^{-rT} \, min \, (V_T, \, k) \right)$$

r: risk free interest rate

k: face value of the debt

Method II

Based on the Black-Scholes pricing formula for European call options, the value of risky debt at time t=0 can be expressed as the difference between firm value (V) and equity value (Eq). The equity value is a European call option on V, which can be derived based on the pricing formula of with Black-Scholes (c^{BS}),

$$D = V - Eq$$
$$D = V - c^{BS} \, (V, \, K, \, r, \, \sigma, \, T)$$

r: risk free interest rate

k: face value of the debt

σ: volatility parameter (greater than zero)

T: time at maturity

There is difference be between the analytically and numerically computed prices. With increasing number of trajectories and decreasing the time period Δt, the simulated value converges to the theoretical price. The term structure of credit spreads (s(T)) on the risky debt depends on the maturity of debt (T). The credit spreads can be calculated by,

$$s(T) = \frac{1}{T} \, ln \, \frac{K}{D} - r$$

2. Results and Analysis

2.1. ***Portfolio Optimization***

The results for this section have been shown in the titled document in the back, Result 1: Portfolio optimization.

2.2. ***Asset Pricing Models***

The results for this section have been shown in the titled document in the back, Result 2: Asset Pricing Models.

2.3. ***Credit Risk Management***

The results for this section have been shown in the titled document in the back, Result 3: Credit Risk Management.

3. Conclusions

This paper discusses the application of data mining tools and R language for three different applications related to portfolio optimization and financial risk management.The applications are for portfolio optimization, asset pricing models and credit risk management. For the portfolio optimization, a portfolio of energy stocks were created and the optimal portfolio was determined using Markowitz mean-variance model. Quadratic programming and lagrange multipliers were applied to determine the optimal portfolio for Markowitz mean-variance model. The optimal portfolio were determined for three cases, equal weights of stock, no short sales i.e. all weights greater than zero and portfolio return equals that of a particular stocks' return. In the second analysis, capital asset pricing model (CAPM) was applied to one of the energy stocks (Exxon) and beta i.e. the sensitivity of the stock to market (SP500) was determined using linear regression model. For both the analyses, the time period was set between January 2010 and December 2014 . In the third analysis, credit risk management was studied for which credit default risk was analyzed using structural models and Merton's options based model. The firm value at maturity was determined applying Brownian motion and the results from this was used to calculate the price of risky debt using Black Scholes formula for pricing of European call option.

Appendix A. References

- H. M. Markowitz (1991), Foundations of Portfolio Theory, The Journal of Finance, Volume 46, Issue 2, pages 469477

- W. F. Sharpe (1964), Capital Asset Prices: A Theory of Market Equilibrium under Conditions Of Risk, Journal of Finance, American Finance Association 19, No. 3, 425-442, 09

- D. Wuertz and M. Hanf (2010), Portfolio Optimization with R/Rmetrics (Rmetrics Association and Finance Online) Available at www.rmetrics.org.

- J. H. Cochrane (2005), Asset pricing, Princeton University Press, New Jersey

- J. Lintner (1965), Security Prices, Risk, and Maximal Gains from Diversifiation, Journal of Finance 20, No. 4, 587-615.

- J. Lintner (1965), The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budget, Review of Economics and Statistics 47, No. 1, 13-37

- K. Giesecke (2004), Credit Risk Modeling and Valuation: An Introduction. Available at SSRN: http://ssrn.com/abstract=479323

- R. Merton (1974), On the Pricing of Corporate Debt: The Risk Structure of Interest Rates, Journal of Finance. 29, 449-470.

- G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning with Applications in R. Available at: http://www-bcf.usc.edu/ gareth/ISL/

- T. Hastie, R. Tibshirani, J. Friedman (2009), The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Available at: http://statweb.stanford.edu/ tibs/ElemStatLearn/

Result 1: Portfolio optimization

```
rm(list=ls ())
options(digits=4, width=70)
source("portfolio_noshorts.r")

library(Quandl)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
Quandl.auth("BJsCQQubXk9YGp8gLohB")

# Model Settings

Set_start_date <-'2010-01-01'
Set_end_date <- '2014-12-01'
#Stock Data Type ----> Energy Stocks
Get_Stock <- 'SNAT31/ES1'


# Downloading and Cleaning Data

#Stocktype Data
StockType <- Quandl(Get_Stock, start_date = Set_start_date, end_date = Set_end_date, header=T)
str(StockType)
```

```
## 'data.frame':    1244 obs. of  8 variables:
##  $ Date: Date, format: "2014-12-01" ...
##  $ XOM : num  92.3 90.5 94.5 94.8 95.7 ...
##  $ CVX : num  112 109 115 116 118 ...
##  $ COP : num  67.8 66.1 70.8 71.7 73.3 ...
##  $ RDS : num  39.8 39.3 41.6 41.6 42 ...
##  $ BP  : num  56.2 55.6 59.7 59.7 60.4 ...
##  $ TOT : num  39.2 39.2 42 42.4 42.4 ...
##  $ E   : num  19.5 19.1 21.8 22.2 22.6 ...
```

```
# Return Calculation
assets <- StockType[, -1]
return <- log(tail(assets, -1) / head(assets, -1))
head(return)
```

```
##         XOM       CVX       COP       RDS        BP        TOT
## 2 -0.019794 -0.025931 -0.025405 -0.012887 -0.009660 -0.0007652
## 3  0.042596  0.055734  0.069568  0.056126  0.071279  0.0692480
```

```
## 4  0.003170  0.008994  0.012626 -0.000481  0.000000  0.0092429
## 5  0.009869  0.012322  0.022061  0.011243  0.011484  0.0011788
## 6  0.011323  0.008384  0.004219  0.008998 -0.007474  0.0047015
## 7 -0.010279 -0.010768 -0.005856 -0.015442 -0.020380 -0.0324147
##          E
## 2 -0.02072
## 3  0.13307
## 4  0.01726
## 5  0.01785
## 6  0.00000
## 7 -0.01425
```

```
# Stock Summary
#
asset.names <- names(assets)
N = ncol(assets)
er <- c(colMeans(return))
names(er) <- asset.names
covmat <- matrix(cov(return), nrow=N, ncol=N)
dimnames(covmat) = list(asset.names, asset.names)

N
```

```
## [1] 7
```

```
er
```

```
##        XOM        CVX        COP        RDS         BP        TOT
## -0.0002440 -0.0002996 -0.0002276  0.0003019  0.0001055  0.0002051
##          E
##  0.0001966
```

```
covmat
```

```
##           XOM       CVX       COP       RDS        BP       TOT
## XOM 0.0001284 0.0001175 0.0001158 0.0001200 0.0001343 0.0001454
## CVX 0.0001175 0.0001578 0.0001322 0.0001286 0.0001494 0.0001577
## COP 0.0001158 0.0001322 0.0002256 0.0001375 0.0001519 0.0001601
## RDS 0.0001200 0.0001286 0.0001375 0.0003164 0.0001701 0.0001820
## BP  0.0001343 0.0001494 0.0001519 0.0001701 0.0002626 0.0002559
## TOT 0.0001454 0.0001577 0.0001601 0.0001820 0.0002559 0.0003343
## E   0.0001419 0.0001590 0.0001675 0.0001767 0.0002212 0.0002370
##             E
## XOM 0.0001419
## CVX 0.0001590
## COP 0.0001675
## RDS 0.0001767
## BP  0.0002212
## TOT 0.0002370
## E   0.0003199
```

```r
r.free = 0.00001
target.ticker <- "RDS"
```

```r
# compute equally weighted portfolio
ew = rep(1,N)/N
equalWeight.portfolio = getPortfolio(er=er,cov.mat=covmat,weights=ew)
class(equalWeight.portfolio)
```

```
## [1] "portfolio"
```

```r
names(equalWeight.portfolio)
```

```
## [1] "call"    "er"      "sd"      "weights"
```

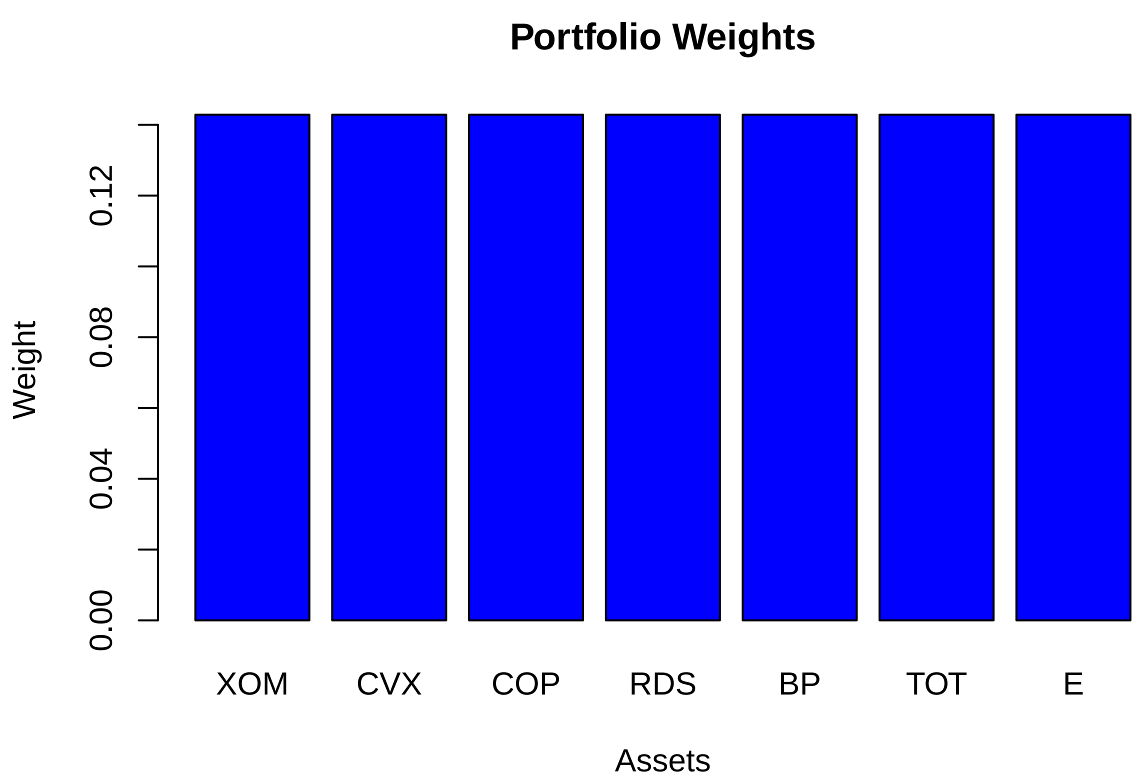
```r
equalWeight.portfolio
```

```
## Call:
## getPortfolio(er = er, cov.mat = covmat, weights = ew)
##
## Portfolio expected return:     5.415e-06
## Portfolio standard deviation:  0.01315
## Portfolio weights:
##    XOM    CVX    COP    RDS     BP    TOT      E
## 0.1429 0.1429 0.1429 0.1429 0.1429 0.1429 0.1429
```

```r
summary(equalWeight.portfolio)
```

```
## Call:
## getPortfolio(er = er, cov.mat = covmat, weights = ew)
##
## Portfolio expected return:     5.415e-06
## Portfolio standard deviation:  0.01315
## Portfolio weights:
##    XOM    CVX    COP    RDS     BP    TOT      E
## 0.1429 0.1429 0.1429 0.1429 0.1429 0.1429 0.1429
```

```r
plot(equalWeight.portfolio, col="blue")
```




```r
# compute global minimum variance portfolio
gmin.port = globalMin.portfolio(er, covmat)
attributes(gmin.port)
```

```
## $names
## [1] "call"    "er"      "sd"      "weights"
##
## $class
## [1] "portfolio"
```

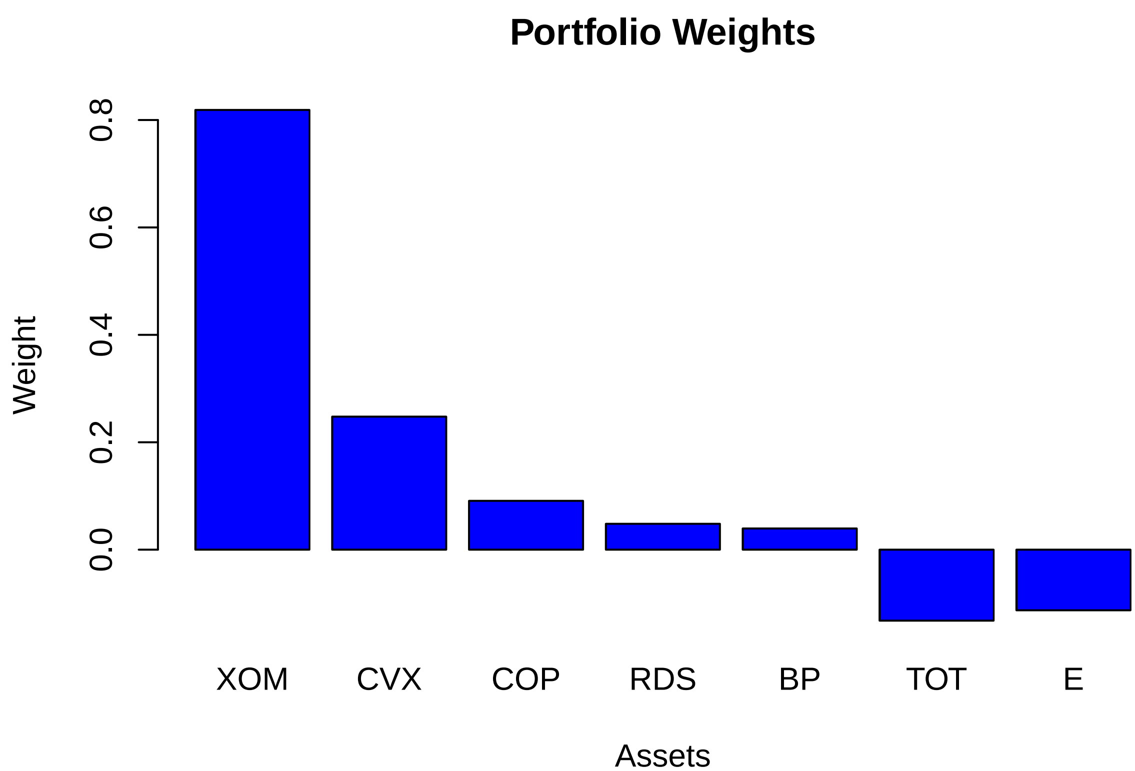
```r
print(gmin.port)
```

```
## Call:
## globalMin.portfolio(er = er, cov.mat = covmat)
##
## Portfolio expected return:     -0.0003253
## Portfolio standard deviation:  0.01098
## Portfolio weights:
##     XOM     CVX     COP     RDS      BP     TOT       E
##  0.8188  0.2477  0.0909  0.0482  0.0395 -0.1322 -0.1129
```

```r
summary(gmin.port, risk.free=r.free)
```

```
## Call:
## globalMin.portfolio(er = er, cov.mat = covmat)
##
## Portfolio expected return:     -0.0003253
## Portfolio standard deviation:  0.01098
## Portfolio Sharpe Ratio:        -0.03053
## Portfolio weights:
##     XOM     CVX     COP     RDS      BP     TOT       E
##  0.8188  0.2477  0.0909  0.0482  0.0395 -0.1322 -0.1129
```

```r
plot(gmin.port, col="blue")
```




```r
# compute global minimum variance portfolio with no short sales
gmin.port.ns = globalMin.portfolio(er, covmat, shorts=FALSE)
attributes(gmin.port.ns)
```

```
## $names
## [1] "call"    "er"      "sd"      "weights"
##
## $class
## [1] "portfolio"
```

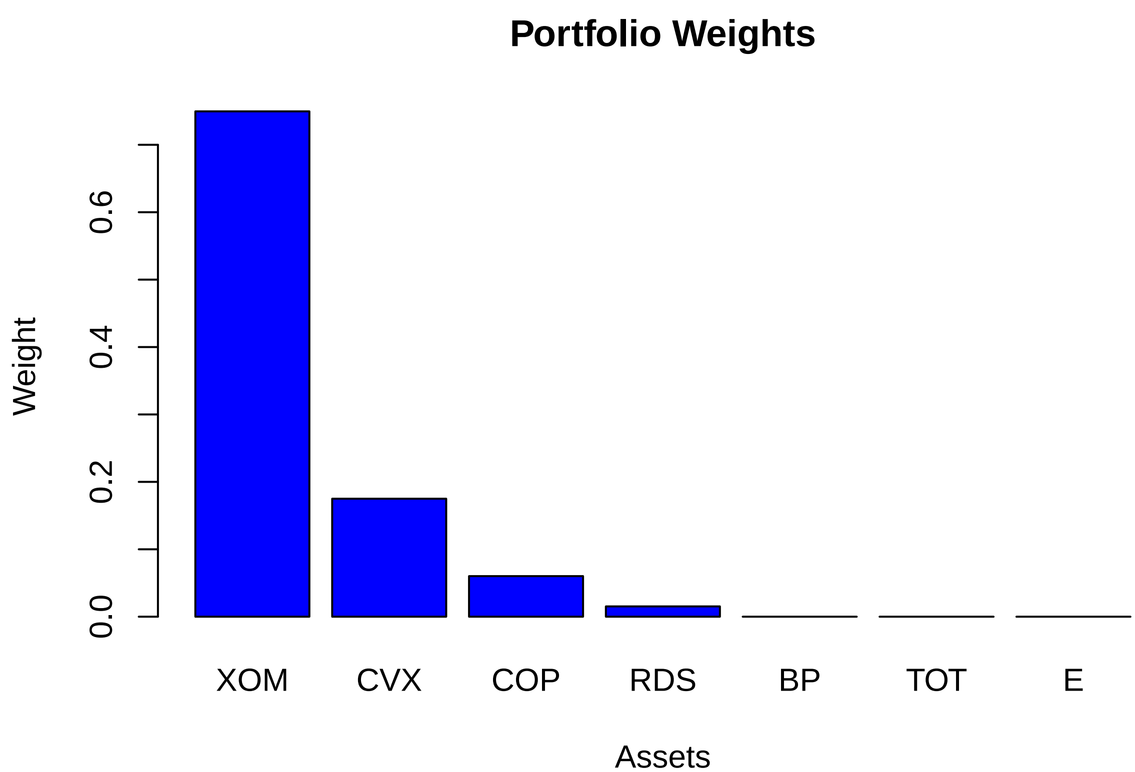
```r
print(gmin.port.ns)
```

```
## Call:
## globalMin.portfolio(er = er, cov.mat = covmat, shorts = FALSE)
##
## Portfolio expected return:     -0.0002444
## Portfolio standard deviation:  0.01121
## Portfolio weights:
##    XOM    CVX    COP    RDS     BP    TOT      E
## 0.7496 0.1750 0.0602 0.0153 0.0000 0.0000 0.0000
```

```r
summary(gmin.port.ns, risk.free=r.free)
```

```
## Call:
## globalMin.portfolio(er = er, cov.mat = covmat, shorts = FALSE)
##
## Portfolio expected return:     -0.0002444
## Portfolio standard deviation:  0.01121
## Portfolio Sharpe Ratio:        -0.0227
## Portfolio weights:
##    XOM    CVX    COP    RDS     BP    TOT      E
## 0.7496 0.1750 0.0602 0.0153 0.0000 0.0000 0.0000
```

```r
plot(gmin.port.ns, col="blue")
```



Results 2: Asset Pricing Models

```r
rm(list=ls ())
library(Quandl)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
Quandl.auth("BJsCQQubXk9YGp8gLohB")

# Model Settings

Set_start_date <-'2010-01-01'
Set_end_date <- '2014-12-01'
Get_Stock <- 'GOOG/NYSE_XOM'#Stock Data ----> Exxon
Get_Market <- 'YAHOO/INDEX_GSPC'#Market Index Data ----> S&P 500
Get_Riskfree <- 'FED/RILSPDEPM01_N_B'#Risk Free Data ----> USD LIBOR Rate
```

```r
# Downloading and Cleaning Data

#Stock Data
Stock <- Quandl(Get_Stock, start_date = Set_start_date, end_date = Set_end_date)
str(Stock)
```

```
## 'data.frame':    1244 obs. of  6 variables:
##  $ Date  : Date, format: "2014-12-01" "2014-11-28" ...
##  $ Open  : num  90.4 91.5 94.6 96 96.3 ...
##  $ High  : num  92.9 91.8 95.1 96 96.8 ...
##  $ Low   : num  90.3 90.1 94.4 94.4 95.2 ...
##  $ Close : num  92.3 90.5 94.5 94.8 95.7 ...
##  $ Volume: num  27584224 19556690 9822941 13747362 10172405 ...
```

```r
Stock <- Stock$Close

#Market Index Data
Market <- Quandl(Get_Market, start_date = Set_start_date, end_date = Set_end_date)
Market <- Market$'Adjusted Close'

#Risk Free Data
Riskfree <- Quandl(Get_Riskfree, start_date = Set_start_date, end_date = Set_end_date)
Riskfree <- Riskfree$Value

#Check if vectors are same length
sapply(list(Stock, Market, Riskfree), length)
```

```
## [1] 1244 1237 1259
```

```r
#Clean data to have same length
Stock <- Quandl(Get_Stock, start_date = Set_start_date, end_date = Set_end_date)
Market <- Quandl(Get_Market, start_date = Set_start_date, end_date = Set_end_date)
Riskfree <- Quandl(Get_Riskfree, start_date = Set_start_date, end_date = Set_end_date)

cdates <- Reduce(intersect, list(Stock$Date, Market$Date, Riskfree$Date))

Stock <- Stock[Stock$Date %in% cdates, 'Close']
Market <- Market[Market$Date %in% cdates, 'Adjusted Close']
Riskfree <- Riskfree[Riskfree$Date %in% cdates, 'Value']
```

```r
# Risk-Return Calculations

#Stock Return Function
logreturn <- function(x) log(tail(x, -1) / head(x, -1))

#Riskfree Reurn
rft <- log(1 + head(Riskfree, -1)/36000 * diff(cdates))
str(rft)
```

```
##  num [1:1231] -1.42e-05 -9.44e-06 -4.72e-06 -4.72e-06 -1.42e-05 ...
```

```r
#Risk Premium function
riskpremium <- function(x) logreturn(x) - rft
```

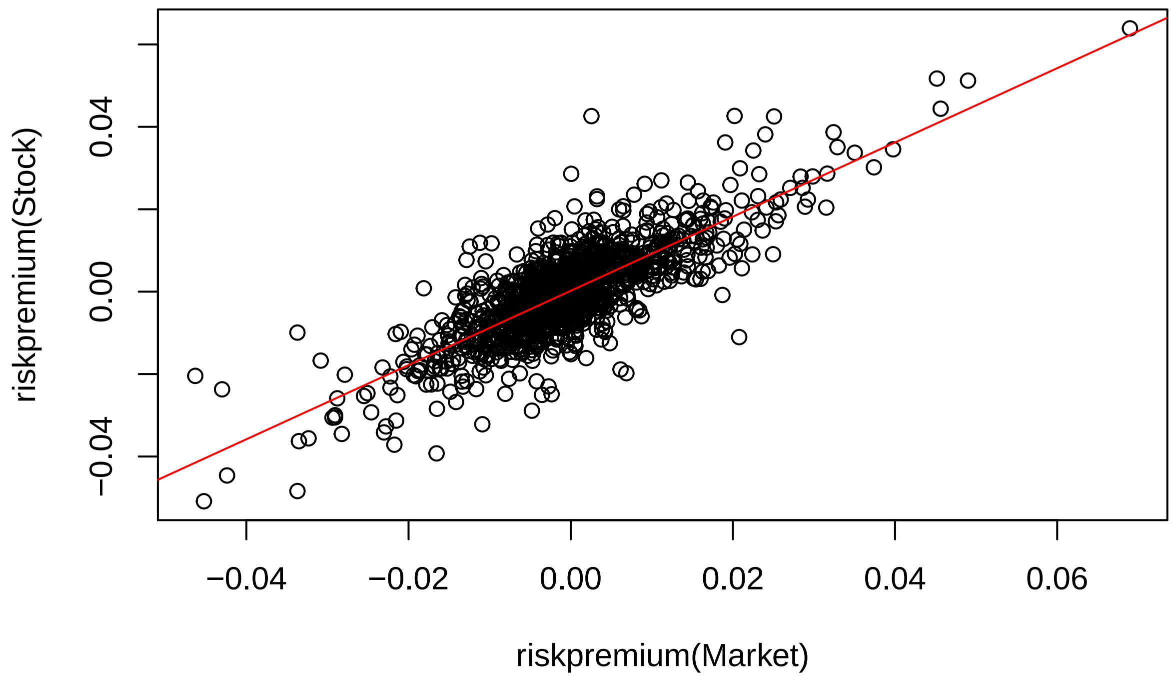
```r
# Beta Estimations

#Simple estimations based on Covariance and Variance of Stock and Market Returns
Beta <- cov(riskpremium(Stock), riskpremium(Market)) / var(riskpremium(Market))
Beta
```

```
## [1] 0.9004304
```

```r
#Beta estimation from Linear Regression
(fit <- lm(riskpremium(Stock) ~ riskpremium(Market)))
```

```
##
## Call:
## lm(formula = riskpremium(Stock) ~ riskpremium(Market))
##
## Coefficients:
##         (Intercept)  riskpremium(Market)
##            0.000201             0.900430
```

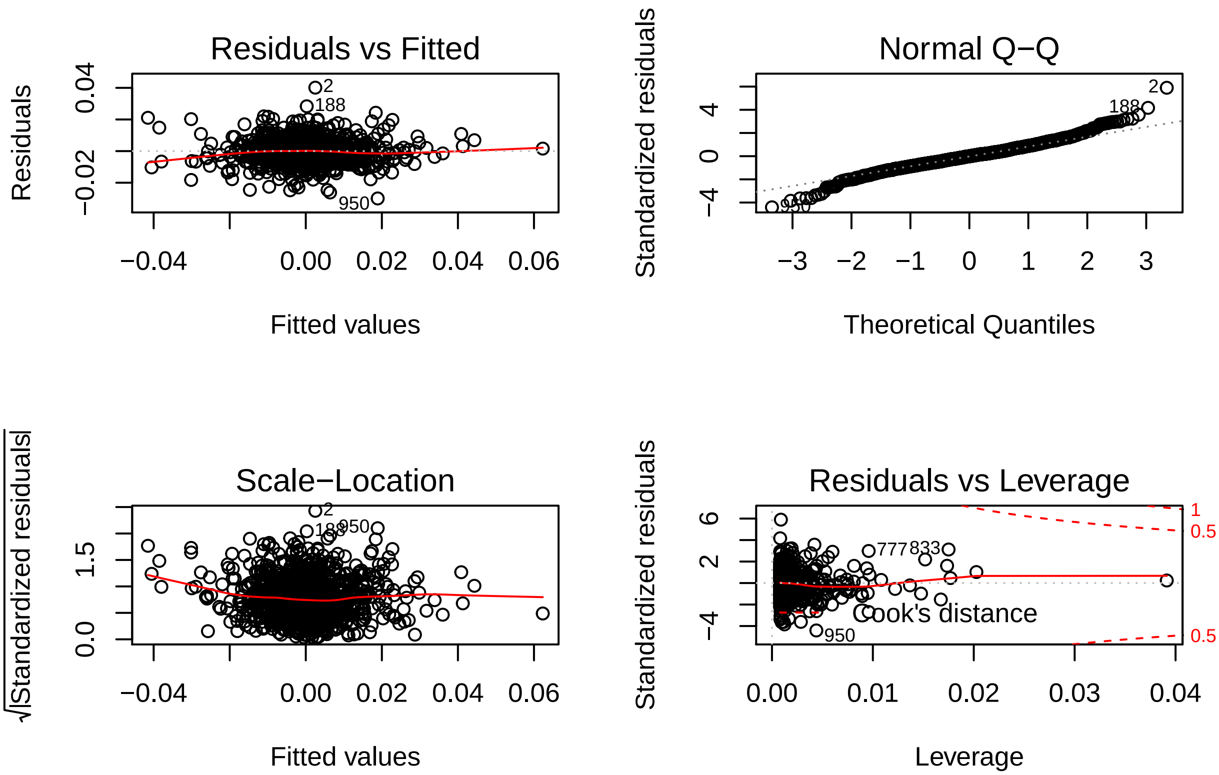
```r
plot(riskpremium(Market), riskpremium(Stock))
abline(fit, col = 'red')
```




```
summary (fit)
```

```
##
## Call:
## lm(formula = riskpremium(Stock) ~ riskpremium(Market))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.029937 -0.004053 -0.000049  0.003721  0.040104
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.0002010  0.0001941   1.036      0.3
## riskpremium(Market) 0.9004304  0.0191725  46.965   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006802 on 1229 degrees of freedom
## Multiple R-squared:  0.6422, Adjusted R-squared:  0.6419
## F-statistic:  2206 on 1 and 1229 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(fit)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

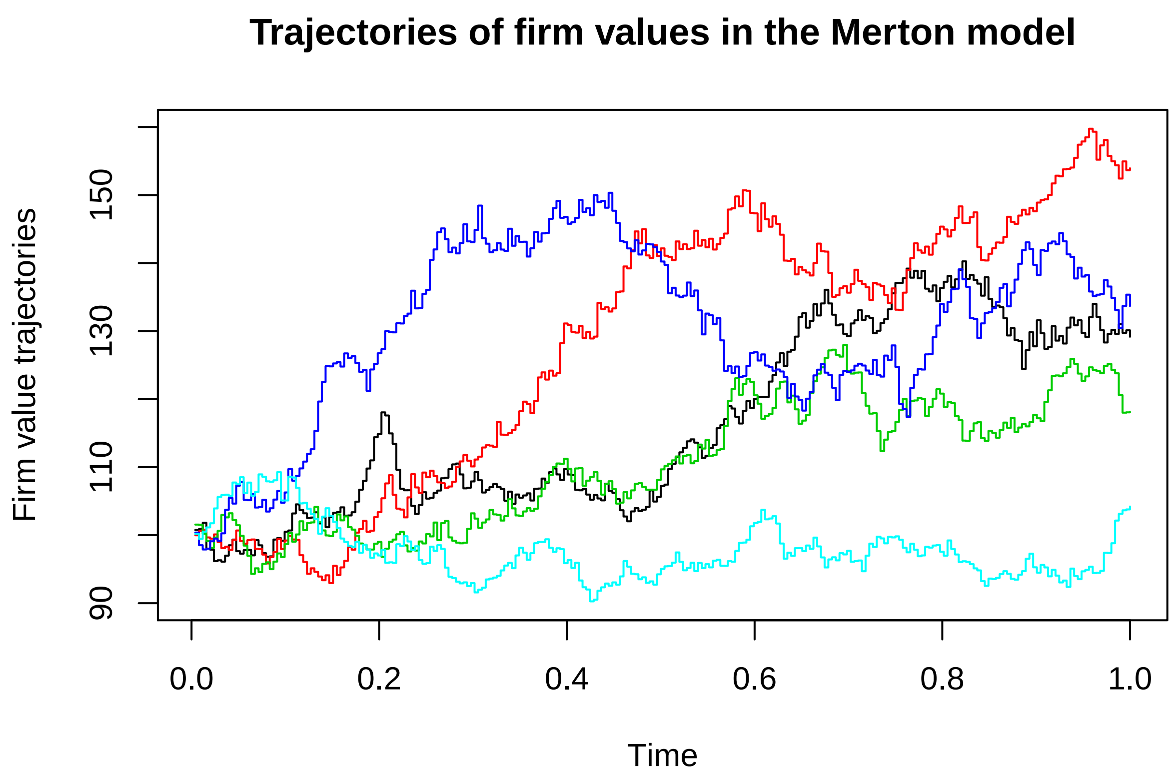
Results 3: Credit Risk Management

```r
rm(list=ls ())
library(fOptions)
```

```
## Loading required package: timeDate
## Loading required package: timeSeries
## Loading required package: fBasics
##
##
## Rmetrics Package fBasics
## Analysing Markets and calculating Basic Statistics
## Copyright (C) 2005-2014 Rmetrics Association Zurich
## Educational Software for Financial Engineering and Computational Science
## Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.
## https://www.rmetrics.org --- Mail to: info@rmetrics.org
```

```r
# Model and Parameter Settings
#Parameter Settings

#Initial asset value, drift, and volatility parameter settings
V0 <- 100; nu <- 0.1; sigma <- 0.2
#Length of ???t and the end of the time periods (Time)
dt <- 1 / 252; Time <- 1
#Number of time periods:
M <- Time / dt
#Number of generated trajectories:
n <- 10000
#Risk-free interest rate and face value of the debt
r <- 0.05; K <- 80
#Price of Risky Debt
Use_BS <- 0 #[Use Black scholes = 1, Use Discounted Payoff = 0]
st <- 0.1; et <- 10; len <- 0.1 #(Maturity time settings)
```

```r
######### Credit Risk Management - Default Models
# Merton Model

set.seed(117)

#Derivation of dlnVt using Ito's lemma
val <- rnorm(n*M, mean = (nu - sigma^2 / 2) * dt, sd = sigma * dt^0.5)
dlnV <- matrix(val, M, n)

#Firm Value at Maturity
V <- V0 * exp(apply(dlnV, 2, cumsum))

#Plot of Trajectories of Firm Values
matplot(x=seq(0 + dt, Time, dt), y=V[, 1:5], type='s', lty=1,
        xlab='Time', ylab='Firm value trajectories',
        main='Trajectories of firm values in the Merton model')
```




```
#Price of Risky Debt
if (Use_BS == 0){
  # Based on expected value of discounted payoff
  D <- exp(-r * Time) * mean((pmin(V[M, ], K)))
  #Price value of Risky Debt
  D
}
```

```
## [1] 75.73553
```

```
if (Use_BS == 1){
  # Based on BLack-Scholes Pricing Formula using Package
  Time <- seq(st, et, len)
  D <- V0 - GBSOption(TypeFlag ="c", S=V0, X=K, Time=Time, r=r, b=r, sigma=sigma)@price
  creditspreads <- (log (K/D))/Time-r
  matplot(x=Time, y=creditspreads, type='l',
          xlab='Maturity', ylab='Credit spreads',
          main='Term structure of credit spreads in the Merton model')
}
```