STATS 240: Final Project

Julie Zhu (lijingz), Johnny Weger (jdweger), Francis Choi (kchoi92), Xianming Li (xmli)

Due: December 6, 2016

```
# Libraries used
library(tidyverse)
library(lubridate)
library(purrr)
library(stringr)
library(MASS)
library(fGarch)
library(quadprog)
```

We first read in the data of of 6 Fama-French portfolios that include stocks traded on NYSE, AMEX, and NASDAQ from January 1992 to January 2012.

```
df <- read.table("FF6Portfolios.txt", skip = 3, header = T) %>%
  tbl_df() %>%
  dmap_at("date", ymd)
```

Part A

In the first step of B-L allocation, we want to calculate the equilibrium risk prima. To do so, we will first calculate market capitalization weights w with a risk-aversion factor of $\gamma = 2.5$ to solve for the equilibrium returns pi.

Then, we will compare the equilibrium risk prima with historical risk prima.

We first obtain data for our risk-free asset. We chose our risk-free asset to be the 3-month Treasury Bills from the Federal Reserve Bank of St.Louis. This is the same risk-free asset we used in other aspects of the class.

```
risk_free <- read_csv("TB3MS.csv") %>%
  dplyr::filter(year(DATE) > 1991, DATE < '2012-02-01') %>%
  rename(TreasuryBill = TB3MS,
         Date = DATE) %>%
  mutate(TreasuryBill = (1 + (TreasuryBill/100))^(1/12) - 1)
```

For our market returns, we pulled data from the Standard & Poor official website. We went to the link for monthly and annualized returns and downloaded our returns from there.

```
sp500 <- read_csv("sp500_m_ret.csv") %>%
  dmap_at("Date", mdy) %>%
  dmap_at("Returns", as.numeric) %>%
  dplyr::filter(year(Date) > 1991) %>%
  .$Returns
```

We combine the two data sets and then calculate the market excess returns.

```
market_returns <- cbind(risk_free[1:length(sp500),], sp500) %>%
  tbl_df() %>%
  mutate(returns = TreasuryBill - sp500)
market_returns <- market_returns[-1,]
```

Next, we define data sets where we pull out the returns, the firm size, and the average market capitalization for each type of firm. The market capitalization gives us an idea of the size of the companies that the specific portfolio invests in.

```
returns <- df %>%
  dplyr::select(smlo_vwret, smme_vwret, smhi_vwret, bilo_vwret,
         bime_vwret, bihi_vwret)

firms <- df %>%
  dplyr::select(smlo_nfirms, smme_nfirms, smhi_nfirms, bilo_nfirms,
         bime_nfirms, bihi_nfirms)

# In millions
market_cap <- df %>%
  dplyr::select(smlo_msize, smme_msize, smhi_msize, bilo_msize,
         bime_msize, bihi_msize)
```

Here we define some constants. We need to have the number of observations and the window size in handy. For an expanding window estimate, we need to increase the size of our data set starting with the window size.

```
# Number of stocks
n <- 6

# Number of time periods
t <- nrow(df)

# Risk aversion factor
gamma <- 2.5

# Expanding window size
window <- 12*5
```

We use the market capitalization sizes to estimate what the weights w are supposed to be. We also calculate the excess returns in order to obtain our σ for the covariance matrix. Then, we use the formula $\pi = \gamma\sigma w$ to calculate the equilibrium risk primas.

```
excess_returns <- returns - market_returns$returns
pi <- matrix(0, ncol = 6, nrow = (t - window + 1))
w <- matrix(0, ncol = 6, nrow = (t - window + 1))

for (i in 0:(t - window)) {
  k <- window + i
  w[i+1,] <- apply(market_cap[1:k,]*firms[1:k,], 2, mean)/
  sum(apply(market_cap[1:k,]*firms[1:k,], 2, mean))
  sigma <- cov(excess_returns[1:k,])
  pi[i+1,] <- gamma*(sigma %*% w[i+1,]) %>% t()}
```

We obtain the following weights for each of the 6 portfolios.

```
head(w)
```

```
##            [,1]       [,2]       [,3]      [,4]      [,5]      [,6]
## [1,] 0.04389526 0.03897748 0.02416476 0.4394165 0.3341779 0.1193681
## [2,] 0.04381620 0.03899651 0.02411380 0.4408438 0.3332497 0.1189800
## [3,] 0.04371865 0.03898244 0.02404898 0.4424185 0.3323083 0.1185232
## [4,] 0.04356112 0.03895814 0.02398792 0.4439201 0.3314392 0.1181335
## [5,] 0.04338025 0.03894288 0.02394685 0.4451378 0.3308039 0.1177883
## [6,] 0.04312875 0.03888903 0.02387945 0.4465878 0.3301470 0.1173680
```

We calculated the historical risk prima earlier as the expected S&P 500 excess returns on the risk-free asset. We plot the equilibrium risk primas against our historical risk prima.

We see that our equilibrium risk primas are a lot more stable and less variable than the historical risk prima. This makes a lot of sense since ultimately the equilibrium risk primas were calculated in very different ways. The historical risk prima follows the market (S&P 500 in our case) where as the equilibrium risk prima is calculated on a rolling window and by adding in a data point past our window. This means that in the beginning of our expanding window, the data for the previous 5 years is weighted heavily in our equilibrium risk calculation.

Here is part of our equilibrium risk prima.

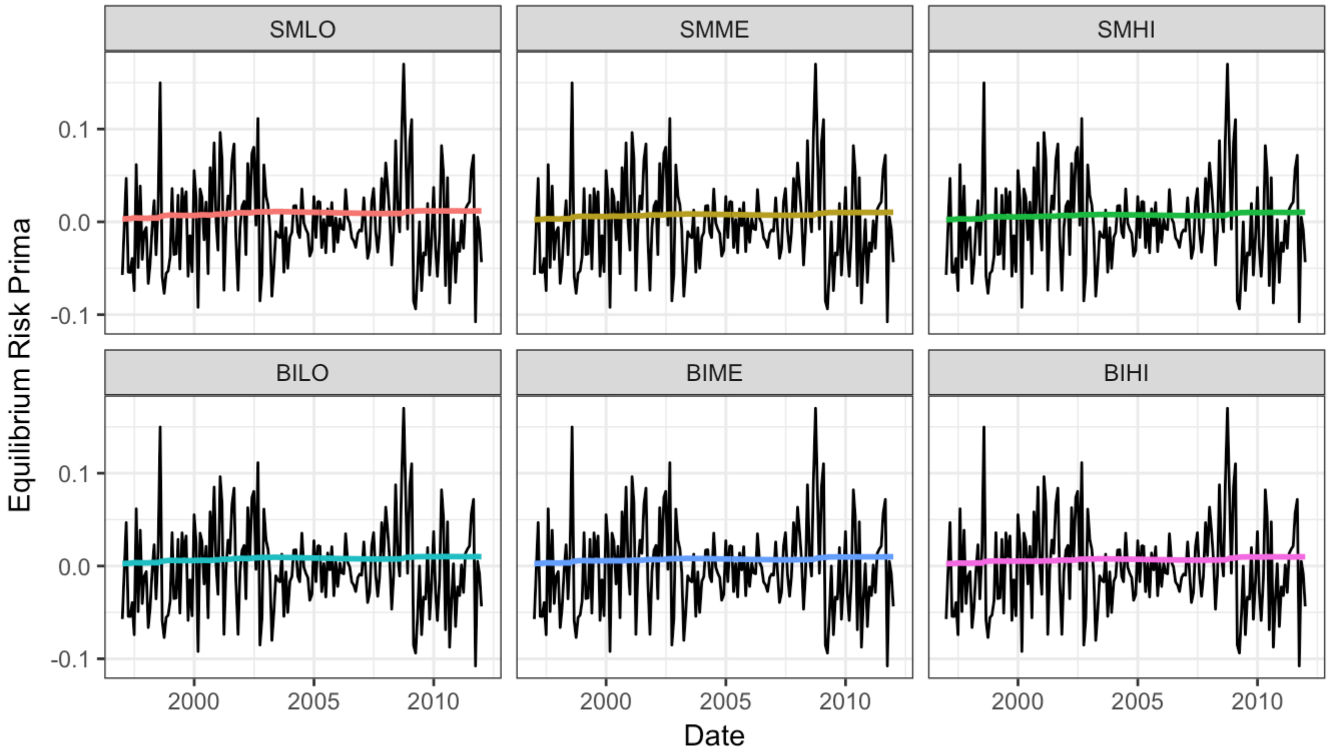
```
head(pi)
```

```
##             [,1]        [,2]        [,3]        [,4]        [,5]
## [1,] 0.003169209 0.002572723 0.002563770 0.002622205 0.002668472
## [2,] 0.003157298 0.002541366 0.002532393 0.002650898 0.002669819
## [3,] 0.003439219 0.002691524 0.002652567 0.002784983 0.002798907
## [4,] 0.003399943 0.002647308 0.002604803 0.002747507 0.002750698
## [5,] 0.003374181 0.002730606 0.002651077 0.003029434 0.002951337
## [6,] 0.003935454 0.003081171 0.002923734 0.003289103 0.003162158
##             [,6]
## [1,] 0.002667416
## [2,] 0.002636602
## [3,] 0.002720123
## [4,] 0.002679110
## [5,] 0.002792181
## [6,] 0.003009025
```

In our graph, the historical risk prima is the black line. The other colors represent the equilibrium risk prima of the other firm types.

```
# Historical Risk Prima
risk_prima <- market_returns %>% .$returns

pi %>%
  tbl_df() %>%
  rename(SMLO = V1, SMME = V2, SMHI = V3,
         BILO = V4, BIME = V5, BIHI = V6) %>%
  gather(firm_type, value, SMLO:BIHI) %>%
  mutate(firm_type = factor(firm_type, levels = c("SMLO", "SMME", "SMHI",
                                                  "BILO", "BIME", "BIHI"))) %>%
  ggplot() +
  geom_line(mapping = aes(x = rep(df$date[window:nrow(df)], times = 6),
                          y = rep(risk_prima[window:nrow(df)], times = 6)),
            color = "black", size = 0.5) +
  geom_line(mapping = aes(x = rep(df$date[window:nrow(df)], times = 6),
                          y = value,
                          color = firm_type), size = 1) +
  guides(color = FALSE) +
  theme_bw() +
  facet_wrap(~firm_type) +
  labs(x = "Date", y = "Equilibrium Risk Prima")
```



Part B and Part C

To beat the market, we need to specify expected views or build models that forecast expected returns well. For simplicity, we will assume the projection matrix is $P = I_6$. We want to find the view vector Q where return forecasts are made. We will use a momentum-based factor model to calculate Q.

Next, we will use the Black-Litterman master formula provided in the handout to calculate μ_{BL} and σ_{BL}. Allowing short selling, use the mean-variance optimization results provided in the textbook (Chapter 3) to calculate the optimal allocation weights for the next period. We will also use the plug-in estimates for the mean-variance estimation and compare the two allocations.

First, we want to create matrices for our portfolio weights and the view vector Q.

```r
w_eff <- matrix(0, nrow = t - window + 1, ncol = n)
w_eff_BL <- matrix(0, nrow = t - window + 1, ncol = n)
w_eff_BL_GARCH <- matrix(0, nrow = t - window + 1, ncol = n)
Q <- matrix(0, ncol = n, nrow = t - window + 1)
Q_GARCH <- Q <- matrix(0, ncol = n, nrow = t - window + 1)

for (i in 0:(t - window - 1)) {
  # Set our rolling window
  k <- window + i
  P <- diag(1, n)
  # Set tau and calculate our covariance estiamtes
  tau <- 1/k
  sigma <- cov(excess_returns[1:k,])
  sigma_pi <- tau*sigma
  # Omega is the diagonal of our variance
  omega <- diag(c(1, diag(tau * P %*% sigma_pi %*% t(P))))[-1,-1]

  # Four-factor momentum model
  smb <- (apply(returns[,1:3], 1, mean) -
            apply(returns[,4:6], 1, mean))[-1]
  hml <- (apply(returns[,c(3,6)], 1, mean) -
            apply(returns[,c(1,4)], 1, mean))[-1]
  rf <- (risk_free$TreasuryBill)[-1]
  rmrf <- (market_returns$returns)[-1]
  excess_returns <- (returns - rf)[-1,]

  maxes <- apply(cbind(apply(returns[,1:3], 1, max),
                       apply(returns[,4:6], 1, max)), 1, mean)
  mins <- apply(cbind(apply(returns[,1:3], 1, min),
                      apply(returns[,4:6], 1, min)), 1, mean)
  momentums <- (maxes - mins)[-241]

  # Calculate Q from the momentum model
  # Momentum model does pretty well - high R^2 value
  for (j in 1:n) {
    fit <- lm(excess_returns[1:k,j] ~ 1 + rmrf[1:k] +
                smb[1:k] + hml[1:k] + momentums[1:k])
    Q[i+1,j] <- sum(gamma*(coef(fit)))
  }

  # Calculate Q from the AR(1)-GARCH(1, 1) model
  for (j in 1:n) {
    fit <- garchFit(excess_returns[1:k,j] ~ arma(1, 0) + garch(1, 1), trace = F)
    Q_GARCH[i+1,j] <- sum(gamma*(coef(fit)))
  }

  # Part C
  S <- solve(solve(sigma_pi) +
    t(P)%*% solve(omega) %*% P)
```

```
m <- S %*% (solve(sigma_pi) %*% matrix(pi[i+1,], ncol = 1) +
  t(P)%*% solve(omega) %*% Q[i+1,])

m_GARCH <- S %*% (solve(sigma_pi) %*% matrix(pi[i+1,], ncol = 1) +
  t(P)%*% solve(omega) %*% Q_GARCH[i+1,])

# MV mu and sigma estimates
sigma <- cov(excess_returns)
u <- colMeans(excess_returns)

# BL mu and sigma estimates
u_BL <- m
u_BL_GARCH <- m_GARCH
sigma_BL <- sigma + S

ones <- array(1, n)

# From Chapter 3
A <- t(u) %*% solve(sigma) %*% ones
A_BL <- t(u_BL) %*% solve(sigma_BL) %*% ones
A_BL_GARCH <- t(u_BL_GARCH) %*% solve(sigma_BL) %*% ones

B <- t(u) %*% solve(sigma) %*% u
B_BL <- t(u_BL) %*% solve(sigma_BL) %*% u_BL
B_BL_GARCH <- t(u_BL_GARCH) %*% solve(sigma_BL) %*% u_BL_GARCH

C <- t(ones) %*% solve(sigma) %*% ones
C_BL <- t(ones) %*% solve(sigma_BL) %*% ones
C_BL_GARCH <- C_BL

D <- B*C - A^2
D_BL <- B_BL*C_BL - A_BL^2
D_BL_GARCH <- B_BL_GARCH*C_BL_GARCH - A_BL_GARCH^2

# mu* was given by the TAs during office hours
u_star <- matrix(0.003, nrow = 1, ncol = 1)
u_star_BL <- matrix(0.003, nrow = 1, ncol = 1)
u_star_BL_GARCH <- matrix(0.003, nrow = 1, ncol = 1)

term1 <- B[1,1] * solve(sigma) %*% ones
term2 <- A[1,1] * solve(sigma) %*% u
term3 <- u_star[1,1]*(C[1,1] * solve(sigma) %*% u - A[1,1] *
                        solve(sigma) %*% ones)

# Weights for MV optimization
w_eff[i+1,] <- (term1 - term2 + term3)/D[1,1]

term1_BL <- B_BL[1,1] * solve(sigma_BL) %*% ones
term2_BL <- A_BL[1,1] * solve(sigma_BL) %*% u_BL
```

```
  term3_BL <- u_star_BL[1,1]*(C_BL[1,1] * solve(sigma_BL) %*%
                               u_BL - A_BL[1,1] * solve(sigma_BL)
                             %*% ones)

  # Weights for BL optimization
  w_eff_BL[i+1,] <- (term1_BL - term2_BL + term3_BL)/D_BL[1,1]

  term1_BL_GARCH <- B_BL_GARCH[1,1] * solve(sigma_BL) %*% ones
  term2_BL_GARCH <- A_BL_GARCH[1,1] * solve(sigma_BL) %*% u_BL_GARCH
  term3_BL_GARCH <- u_star_BL_GARCH[1,1]*(C_BL_GARCH[1,1] * solve(sigma_BL) %*%
                               u_BL_GARCH - A_BL_GARCH[1,1] * solve(sigma_BL)
                             %*% ones)

  # Weights for BL optimization
  w_eff_BL_GARCH[i+1,] <- (term1_BL_GARCH - term2_BL_GARCH +
                            term3_BL_GARCH)/D_BL_GARCH[1,1]
}
```

Our view vector is now given as follows from the four-factor momentum model.

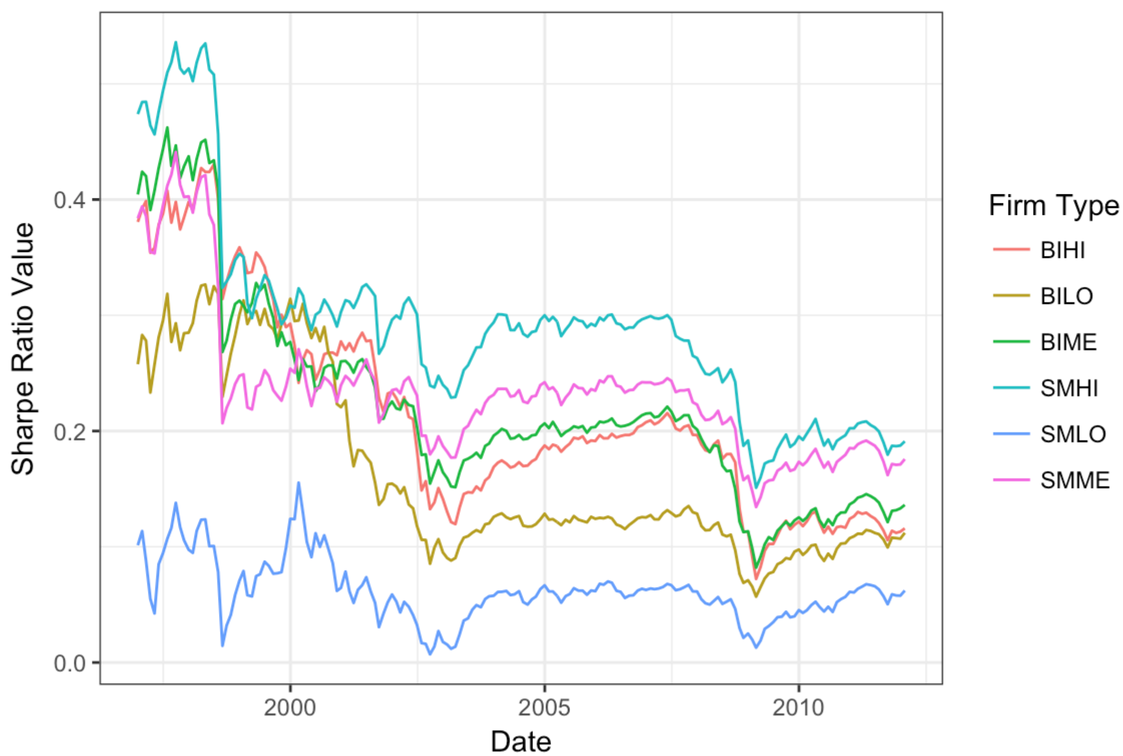
```
head(Q)
```

```
##           [,1]     [,2]     [,3]      [,4]       [,5]      [,6]
## [1,] 1.1376993 2.172387 3.147393 -1.766575 -0.4961540 1.2327074
## [2,] 1.0730406 2.192472 3.154056 -1.721582 -0.5499349 1.2064734
## [3,] 0.8926649 2.074696 3.048322 -1.833744 -0.6562439 1.0199138
## [4,] 0.8804278 1.899090 2.880624 -2.060848 -0.7098062 0.9483754
## [5,] 1.0278062 1.997019 2.957456 -1.987979 -0.6030772 1.0909954
## [6,] 1.0366587 2.014416 2.980476 -1.966087 -0.5835572 1.0986052
```

We also calculated the Sharpe ratio for the data using an expanded window. We see that the average return per unit risk goes down as we look at more and more data. We see a fairly large dip around the recession of 2008, which makes sense since the market was not fruitful during that time.

```
sharpe_ratio <- matrix(0, nrow = t - window + 1, ncol = n)
colnames(sharpe_ratio) <- c("SMLO", "SMME", "SMHI", "BILO",
                            "BIME", "BIHI")
for (i in 0:(t - window)) {
  k <- window + i
  sharpe_ratio[i+1,] <- (colMeans(returns[1:k,] - risk_free$TreasuryBill[1:k]))/
  apply(returns[1:k,], 2, sd)
}

sharpe_ratio %>%
  tbl_df() %>%
  gather(type, value, SMLO:BIHI) %>%
  ggplot() +
  geom_line(mapping = aes(x = rep(df$date[60:nrow(df)], 6), y = value,
                          color = type)) +
  scale_color_discrete(name = "Firm Type") +
  theme_bw() +
  labs(x = "Date", y = "Sharpe Ratio Value")
```



We see that the portfolio weights for both the mean-variance allocation and the Black-Litterman allocation suggest roughly the same action. They both suggest to short SMLO, SMHI and to buy SMME, BILO, and BIHI. The MV allocation suggest to buy BIME, while BL suggests to short BIME.

The magnitude of the weights are a little different as well. The BL allocation actually shorts SMLO by a much larger proportion than the MV allocation. The MV allocation places a greater weight on BIHI, while BL only considers a small weight for that portfolio.

These weights make sense because the BL allocation is a riskier allocation, since the Markowitz allocation minimizes the variance.

```
head(w_eff)
```

```
##             [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] -0.04803964 0.7876661 -1.083901 0.6411456 0.1003089 0.6028202
## [2,] -0.04803964 0.7876661 -1.083901 0.6411456 0.1003089 0.6028202
## [3,] -0.04803964 0.7876661 -1.083901 0.6411456 0.1003089 0.6028202
## [4,] -0.04803964 0.7876661 -1.083901 0.6411456 0.1003089 0.6028202
## [5,] -0.04803964 0.7876661 -1.083901 0.6411456 0.1003089 0.6028202
## [6,] -0.04803964 0.7876661 -1.083901 0.6411456 0.1003089 0.6028202
```

```
head(w_eff_BL)
```

```
##            [,1]     [,2]       [,3]      [,4]       [,5]       [,6]
## [1,] -1.0071044 1.802175 -0.3902837 0.9968236 -0.4477888 0.04617861
## [2,] -1.0117215 1.804938 -0.4049119 1.0170142 -0.4312290 0.02590987
## [3,] -1.0225916 1.815238 -0.3885715 1.0157918 -0.4428550 0.02298794
## [4,] -1.0132580 1.822572 -0.3347486 0.9407012 -0.5097695 0.09450286
## [5,] -0.9987075 1.811578 -0.3469672 0.9301431 -0.5008651 0.10481850
## [6,] -1.0017978 1.808909 -0.3464925 0.9382366 -0.4912665 0.09241104
```

With the GARCH model, we actally see similar behavior to the BL allocation using the momentum model. However, the GARCH model gives us a weighting
that is even riskier. It shorts SMLO, SMHI, and BIME by a lot, but buys SMME in a larger proportion than BILO and BIHI. Note that this model also gives us a view vector that Shorts BIHI as well, where the momentum model gave us a weighting that buys BIHI in a small weighting.

Moving forward, we will focus on the four-factor momentum model.

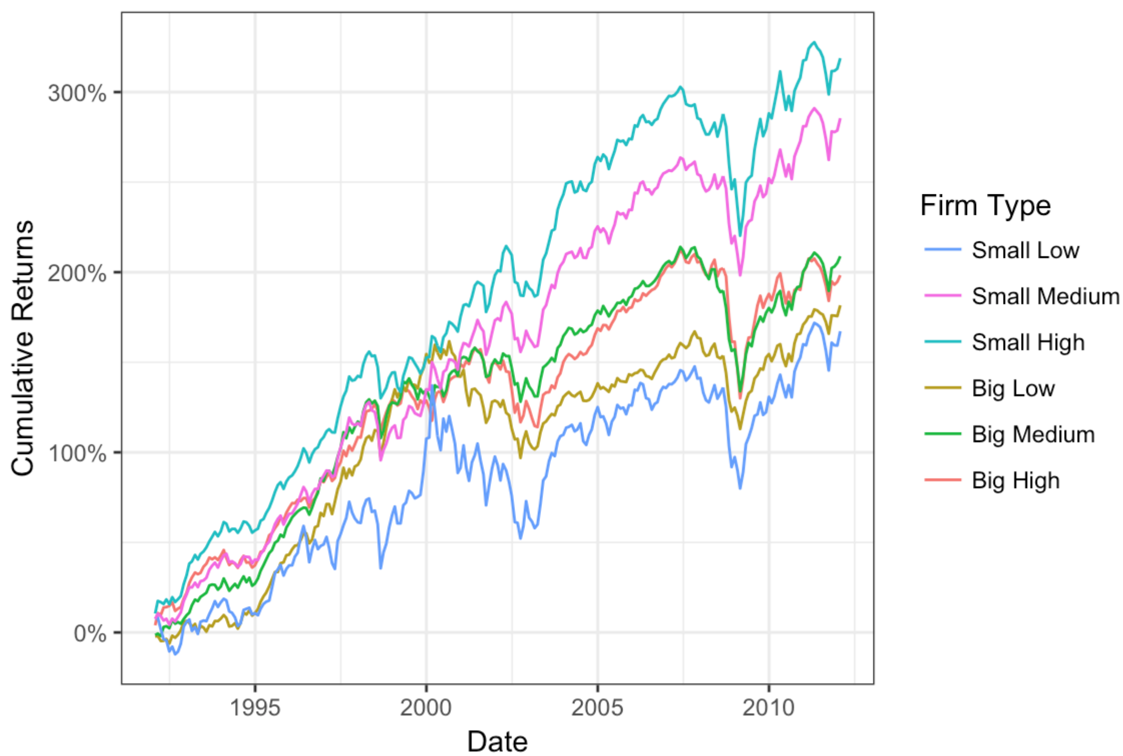
```
head(w_eff_BL_GARCH)
```

```
##            [,1]     [,2]      [,3]      [,4]       [,5]       [,6]
## [1,] -14.652158 23.53677 -3.476299 11.016392 -11.019380 -4.4053250
## [2,]  -7.723843 17.62547 -7.153321  4.623956  -6.574091  0.2018254
## [3,]  -7.983296 17.33304 -6.480953  4.824026  -6.313361 -0.3794585
## [4,]  -9.135970 20.20488 -7.723387  5.270057  -7.293165 -0.3224184
## [5,]  -8.884802 18.41008 -6.285686  5.356403  -6.827557 -0.7684339
## [6,]  -9.028478 19.75859 -7.469239  5.327698  -7.468709 -0.1198635
```

Finally, we plot the cumulative returns for each type of portfolio. We see that SMHI actually allows us to triple our initial investment over the span of 20 years. Interestingly enough, we also see a dip around the recession in 2008.

```r
cumulative_df <- cbind(cumsum(returns[,1]), cumsum(returns[,2]),
                       cumsum(returns[,3]), cumsum(returns[,4]),
                       cumsum(returns[,5]), cumsum(returns[,6]))

cumulative_df %>%
  gather(firmtype, returns, smlo_vwret:bihi_vwret) %>%
  group_by(firmtype) %>%
  ggplot() +
  geom_line(mapping = aes(x = rep(df$date, 6), y = returns,
                          color = firmtype)) +
  scale_color_discrete(breaks = c("smlo_vwret", "smme_vwret", "smhi_vwret",
                                  "bilo_vwret", "bime_vwret", "bihi_vwret"),
                       labels = c("Small Low", "Small Medium", "Small High",
                                  "Big Low", "Big Medium", "Big High"),
                       name = "Firm Type") +
  scale_y_continuous(labels = scales::percent) +
  theme_bw() +
  labs(x = "Date", y = "Cumulative Returns")
```



Part D

In this section, we will plot the turnover rates for both the mean-variance optimization and the Black-Litterman on expanding windows.

We first calculate the MV allocation turnover rates using the weights calculated in Part C.

```
investment <- 1000000
mv_allocation <- w_eff[1,] * investment

# Turnover Rate for BL Optimization
MVTR <- array(0, t - window)

current_returns <- mv_allocation

for (i in 0:(t-window - 1)) {
  k <- window + i
  past_returns <- w_eff[i + 1,] * investment
  current_returns <- (past_returns %>% unlist()) * (returns[k,] %>% unlist())
  # Formula as given in the project description
  num <- sum(abs(current_returns - past_returns))
  denom <- sum(abs(current_returns))
  MVTR[i+1] <- abs(num)/abs(denom)
}
```

Next, we calculate the BL allocation turnover rates using the weights calculated in Part C.

```
# Turnover Rate for BL Optimization
bl_allocation <- w_eff_BL[1,] * investment

BLTR <- array(0, t - window)

current_returns <- bl_allocation

for (i in 0:(t-window - 1)) {
  k <- window + i
  past_returns <- w_eff_BL[i + 1,] * investment
  # Formula as given in the project description
  current_returns <- (past_returns %>% unlist()) * (returns[k,] %>% unlist())
  num <- sum(abs(current_returns - past_returns))
  denom <- sum(abs(current_returns))
  BLTR[i+1] <- abs(num)/abs(denom)
}
```

In our plot, we see that the two turnover rates are generally very similar over time. Note that for the data, the point corresponding to a date uses data from that date and all the dates before. Several of the larger spikes in turnover rate come from the BL portfolio. We justify that since the BL allocation is a lot riskier than the MV portfolio, it is likely that there is more selling and buying transactions occuring with the BL portfolio. But we do see some spikes where the MV portfolio also indicates a lot of transactions.

Overall, we see the two methods being pretty comparable.

```
cbind(MVTR, BLTR) %>%
  tbl_df() %>%
  gather(portfolio, turnover, MVTR:BLTR) %>%
  ggplot() +
  geom_line(mapping = aes(x = rep(df$date[window:(t-1)], 2), y = turnover,
            color = portfolio)) +
  theme_bw() +
  scale_color_discrete(breaks = c("MVTR", "BLTR"),
                       labels = c("Mean-Variance", "Black-Litterman"),
                       name = "Allocation Type") +
  labs(x = "Date", y = "Turnover Rate")
```