HW #1 - Portfolio Construction

Ivan Rodriguez

Contents

Data Notes **1**

Questions **2**

1a. The 6x1 vector of mean returns, $\boldsymbol{\mu}$. 2

1b. The 6x6 covariance matrix, $\boldsymbol{\Omega}$. 2

1c. Find the intermediate values of A, B, C, and D. 2

1d. Find the 6x1 vector **g** 3

1e. Find the 6x1 vector **h** 4

1f. Find the globabl minimum variance portfolio, w_g 4

1g. Find the globabl minimum variance portfolio, μ_g 4

1h. Find the globabl minimum variance portfolio, σ_g 4

1i. Find weight for an efficient portfolio with a mean equal to 3.5%, and call this portfolio p 4

1j. Find the weights, $\mathbf{w}_{op}$, and the mean , μ_{op}, for portfolio p's zero beta portfolio 5

1k. Find the regression beta of the first portfolio's return with respect to portfolio p. 5

1m. Use all 6 portfolios and plot them in mean-standard deviation space. Which portfolio shows the best return per unit of risk? 6

1n. Use the matrix based techniques learned in class to derive the mean-variance frontier and plot the 6 portfolios 7

Data Notes

The data used in the analysis is the Fama French portfolio and factors database from WRDS. The data are monthly value weighted return observations on the 6 portfolio constructed by Fama and French formed on size and book to market. The time period of the data is from the end of March 2000 to the end of July 2014. The portfolio variable names are as follows:

- SMLO (small size and low book-to-market ratio equities),
- SMME (small size and medium book-to-market ratio equities),
- SMHI (small size and high book-to-market ratio equities),
- BILO (big size and low book-to-market ratio equities),
- BIME (big size and medium book-to-market ratio equities),
- BIHI (big size and high book-to-market ratio equities)

Here is a small sample of the data that was used:

```
## Source: local data frame [4 x 8]
##
##       date smlo_vwret smme_vwret smhi_vwret bilo_vwret bime_vwret bihi_vwret     rf
## 1 20000331    -0.1338    -0.0383    -0.0102     0.0870     0.1064     0.1165 0.0047
## 2 20000428    -0.1416    -0.0555    -0.0456    -0.0429    -0.0117     0.0428 0.0046
## 3 20000531    -0.0808    -0.0553    -0.0370    -0.0343     0.0105    -0.0037 0.0050
## 4 20000630     0.1746     0.0926     0.0796     0.0502    -0.0587    -0.0553 0.0040
```

Questions

Before proceeding with answering the homework questions, we need to initialize a few things first, namely the unity vector. The unity vector $\boldsymbol{\iota}$ is the 6 x 1 vector of ones.

```
i <- 1 + numeric(n)
```

1a. The 6x1 vector of mean returns, $\boldsymbol{\mu}$.

The vector $\boldsymbol{\mu}$ is the 6 x 1 vector of the mean returns.

```
mu <- sapply(data.portfolio, mean)
mu
```

```
## smlo_vwret smme_vwret smhi_vwret bilo_vwret bime_vwret bihi_vwret
##  0.0039168  0.0103225  0.0116838  0.0042509  0.0072699  0.0080746
```

1b. The 6x6 covariance matrix, $\boldsymbol{\Omega}$.

The matrix $\boldsymbol{\Omega}$ is the 6 x 6 matrix whose elements are the covariances between the portfolios

```
V <- cov(data.portfolio)
V
```

```
##            smlo_vwret smme_vwret smhi_vwret bilo_vwret bime_vwret bihi_vwret
## smlo_vwret  0.0047331  0.0034930  0.0035378  0.0025346  0.0022701  0.0024439
## smme_vwret  0.0034930  0.0030936  0.0032499  0.0019477  0.0021714  0.0024387
## smhi_vwret  0.0035378  0.0032499  0.0036386  0.0020124  0.0023604  0.0027476
## bilo_vwret  0.0025346  0.0019477  0.0020124  0.0019581  0.0017149  0.0018303
## bime_vwret  0.0022701  0.0021714  0.0023604  0.0017149  0.0021182  0.0022926
## bihi_vwret  0.0024439  0.0024387  0.0027476  0.0018303  0.0022926  0.0028706
```

1c. Find the intermediate values of A, B, C, and D.

The intermediate values, which will make the later computation easier, are found through solving the original optimization problem and some matrix algebraic manipulation.

Intermediate Value A

The value of A is found to be $A = \boldsymbol{\iota}'\boldsymbol{\Omega}^{-1}\boldsymbol{\mu}$.

```r
A <- t(i) %*% solve(V) %*% mu
A
```

```
##        [,1]
## [1,] 6.2384
```

Intermediate Value B

The value of B is found to be $B = \boldsymbol{\mu}'\boldsymbol{\Omega}^{-1}\boldsymbol{\mu}$.

```r
B <- t(mu) %*% solve(V) %*% mu
B
```

```
##         [,1]
## [1,] 0.14741
```

Intermediate Value C

The value of C is found to be $C = \boldsymbol{\iota}'\boldsymbol{\Omega}^{-1}\boldsymbol{\iota}$.

```r
C <- t(i) %*% solve(V) %*% i
C
```

```
##        [,1]
## [1,] 711.38
```

Intermediate Value D

The value of D is found to be $D = BC - A^2$.

```r
D <- B * C - A^2
D
```

```
##       [,1]
## [1,] 65.95
```

1d. Find the 6x1 vector g

The vector **g** is the 6 x 1 vector which is equal to $\frac{1}{D}\left[B(\boldsymbol{\Omega}^{-1}\boldsymbol{\iota}) - A(\boldsymbol{\Omega}^{-1}\boldsymbol{\mu})\right]$.

```r
g <- 1/drop(D) * (drop(B) * (solve(V) %*% i) - drop(A) * (solve(V) %*% mu))
g <- g[,1]
g
```

```
## smlo_vwret smme_vwret smhi_vwret bilo_vwret bime_vwret bihi_vwret
##  0.1753291 -0.0037682 -0.8519829  0.9557043  0.7553438 -0.0306261
```

1e. Find the 6x1 vector h

The vector $\mathbf{h}$ is the 6 x 1 vector which is equal to $\frac{1}{D}\left[C(\boldsymbol{\Omega}^{-1}\boldsymbol{\mu}) - A(\boldsymbol{\Omega}^{-1}\boldsymbol{\iota})\right]$.

```
h <- 1/drop(D) * (drop(C) * (solve(V) %*% mu) - drop(A) * (solve(V) %*% i))
h <- h[,1]
g
```

```
## smlo_vwret smme_vwret smhi_vwret bilo_vwret bime_vwret bihi_vwret
##  0.1753291 -0.0037682 -0.8519829  0.9557043  0.7553438 -0.0306261
```

1f. Find the globabl minimum variance portfolio, w_g

The formula for the weights of the global minimum portfolio is $w_g = \frac{1}{C}\boldsymbol{\Omega}^{-1}\boldsymbol{\iota}$.

```
w.gl <- 1/drop(C) * solve(V) %*% i
w.gl <- w.gl[,1]
w.gl
```

```
## smlo_vwret smme_vwret smhi_vwret bilo_vwret bime_vwret bihi_vwret
##   -0.91735    1.37903   -0.43101    1.19338    0.12495   -0.34900
```

1g. Find the globabl minimum variance portfolio, μ_g

The formula for the return of the global minimum portfolio is $\mu_g = \frac{A}{C}$.

```
mu.gl <- drop(A) / drop(C)
mu.gl
```

```
## [1] 0.0087694
```

1h. Find the globabl minimum variance portfolio, σ_g

The formula for the standard deviationof the global minimum portfolio is $\sigma_g = \frac{1}{C}$.

```
sigma.gl <- 1 / drop(C)
sigma.gl
```

```
## [1] 0.0014057
```

1i. Find weight for an efficient portfolio with a mean equal to 3.5%, and call this portfolio p

The formula for the optimal weight for a portfolio p is $w_p = \mathbf{g} + \mathbf{h}\mu_p$.

```
mu.p <- .035
w.p <- g + h * mu.p
w.p
```

```
## smlo_vwret smme_vwret smhi_vwret bilo_vwret bime_vwret bihi_vwret
##   -4.18569    5.51516    0.82817    1.90431   -1.76064   -1.30131
```

1j. Find the weights, $\mathbf{w}_{op}$, and the mean , μ_{op}, for portfolio p's zero beta portfolio

The return of the portfolio which is uncorrelated with portfolio p is $\mu_{op} = \frac{D/C^2}{\mu_p - D/C} + \frac{A}{C}$.

```
mu.0p <- (D/C^2)/(mu.p - D/C) + A/C
mu.0p
```

```
##            [,1]
## [1,] 0.0065111
```

The weight vector of the portfolio is given by $\mathbf{w}_{op} < -\mathbf{g} + \mathbf{h}\mu_{op}$.

```
w.0p <- g + h * mu.0p
w.0p
```

```
## smlo_vwret smme_vwret smhi_vwret bilo_vwret bime_vwret bihi_vwret
##   -0.63596    1.02293   -0.53942    1.13218    0.28729   -0.26701
```

1k. Find the regression beta of the first portfolio's return with respect to portfolio p.

Now, we will SMLO's return's on the portfolio p's return. In order to run the regression, first we have to find the return of our portfolio p over the same time period as SMLO, denoted as $\boldsymbol{\mu}_P$. We simply use the following formula, $\boldsymbol{\mu}_P = \mathbf{X}\mathbf{w}_p$ where $\mathbf{X}$ is our data matrix.

```
port_vwret <- as.matrix(data.portfolio) %*% w.p
port_vwret <- data.frame(port_vwret)
data <- tbl_df(cbind(data, port_vwret))

#Creating separate regression data table in percent instead of decimals
data.regression <- select(data, smlo_vwret, port_vwret) * 100
head(data.regression, n=4)
```

```
##   smlo_vwret port_vwret
## 1     -13.38    16.7108
## 2     -14.16    13.2046
## 3      -8.08    -7.6417
## 4      17.46    11.6713
```

Now that we have calculated the necessary returns, we can run our regression, $\boldsymbol{\mu}_{SMLO} = \alpha + \beta\boldsymbol{\mu}_p$.

```
reg <- lm(smlo_vwret ~ port_vwret, data=data.regression)
summary(reg)
```

```
##
## Call:
## lm(formula = smlo_vwret ~ port_vwret, data = data.regression)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -22.204  -4.543   0.708   4.575  17.038
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.3787     0.5600    0.68     0.50
## port_vwret    0.0037     0.0560    0.07     0.95
##
## Residual standard error: 6.9 on 171 degrees of freedom
## Multiple R-squared:  2.56e-05,   Adjusted R-squared:  -0.00582
## F-statistic: 0.00438 on 1 and 171 DF,  p-value: 0.947
```

As we can see, our regression beta is $\beta = 0.0037$.

1m. Use all 6 portfolios and plot them in mean-standard deviation space. Which portfolio shows the best return per unit of risk?

To plot our points and find the portfolio with the highest return per unit of risk, we first have to find their standard deviations, which we denote as the vector $\boldsymbol{\sigma}$.

```r
v <- diag(V)
s <- sqrt(v)
s
```

```
## smlo_vwret smme_vwret smhi_vwret bilo_vwret bime_vwret bihi_vwret
##   0.068797   0.055620   0.060321   0.044250   0.046024   0.053578
```

Now, we can find the portfolio with the largest return per unit of risk

```r
retrisk <- mu/s
retrisk
```

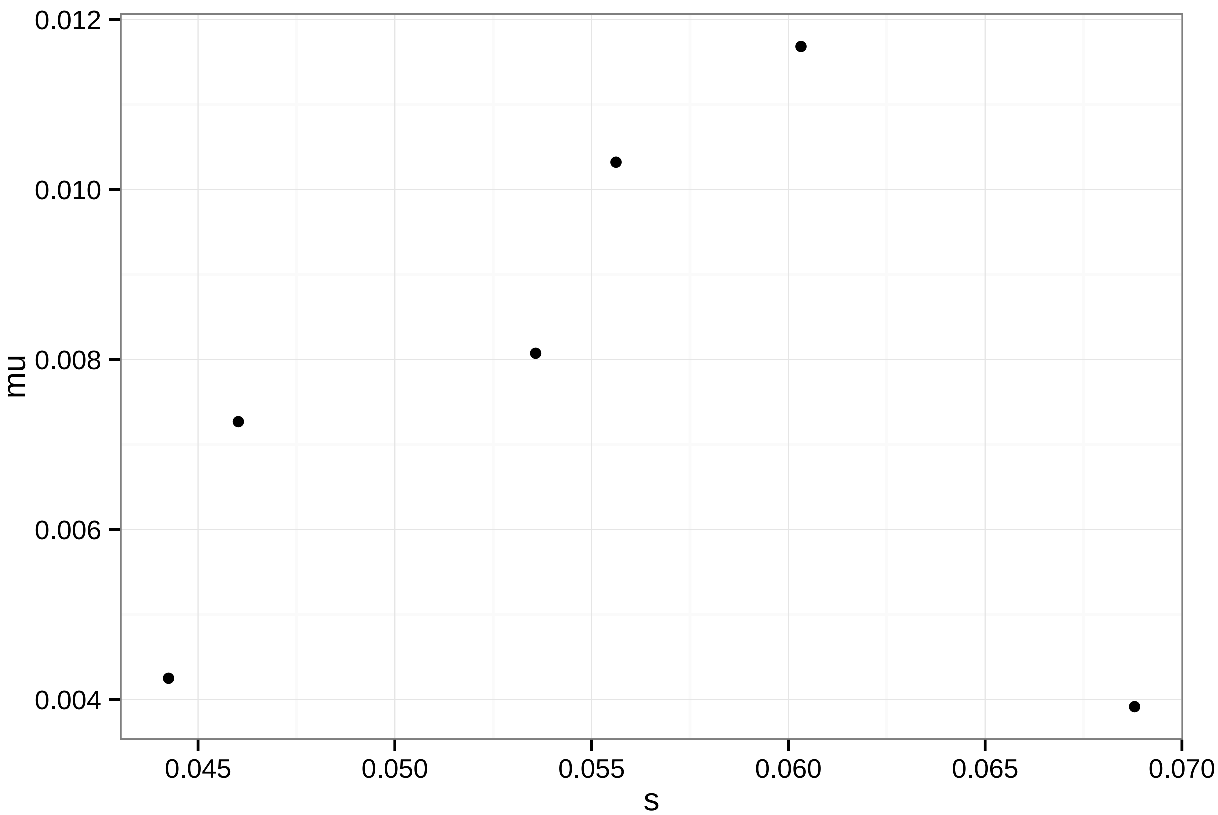
```
## smlo_vwret smme_vwret smhi_vwret bilo_vwret bime_vwret bihi_vwret
##   0.056932   0.185590   0.193695   0.096065   0.157961   0.150707
```

```r
max(retrisk)
```

```
## [1] 0.1937
```

and plot them in mean-standard deviation space

```r
qplot(s, mu) + theme_bw()
```



1n. Use the matrix based techniques learned in class to derive the mean-variance frontier and plot the 6 portfolios

To plot the mean variance frontier, we first have to find the weights of the optimal weights of any efficient portfolio. We can plot each efficient frontier if we had their return and variance. To find the return, we can use this formula: $\mu_e = \alpha\mu_g + (1-\alpha)\mu_p$. To find the variance, we first have to find the weight of the efficient frontier this using this formula: $w_e = \alpha(\mathbf{g} + \mathbf{h}\mu_g) + \alpha(\mathbf{g} + \mathbf{h}\mu_p)$.

```
#Initialize vectors that are going to be used
alpha <- seq(0,2,.01)
return.eff = NULL
std.eff = NULL

#loop for every alpha and write to our intialized vectors
for (i in 1:length(alpha)) {
  return.eff[i] = alpha[i] * mu.gl + (1 - alpha[i]) * mu.p
  weight.eff = alpha[i] * (g + h*mu.gl) + (1 - alpha[i]) * (g + h*mu.p)
  std.eff[i] = sqrt(diag(t(weight.eff) %*% V %*% weight.eff))
}

#Create data table for graph
frontier.data <- data.frame(cbind(return.eff, std.eff))

#Plot frontier and portfolios
ggplot() +
  geom_path(data=frontier.data, aes(x=std.eff, y=return.eff)) +
```

```
geom_point(aes(x=s, y=mu)) +
theme_bw()
```