Project-3

Nicholas Davidson

4/19/2023

```
# Read in csv
a <- read.table("http://www.stat.ucla.edu/~nchristo/statistics_c183_c283/statc183c283_5stocks.txt", header=T)
```

These are close monthly prices from January 1986 to December 2003. The first column is the date and P1, P2, P3, P4, P5 represent the close monthly prices for the stocks Exxon-Mobil, General Motors, Hewlett Packard, McDonalds, and Boeing respectively.

```
# a. Convert prices into returns for all 5 stocks. The data has most recent data at beginning
a <- a[rev(1:nrow(a)),]
returns <- a[2:nrow(a),2:6]/a[1:(nrow(a)-1),2:6]-1
```

```
# b. Compute the mean return for each stock and the variance-covariance matrix.

# Mean returns
mean_returns <- apply(returns, 2, mean)
cat('The mean returns for the stocks are', mean_returns, '\n')
```

```
## The mean returns for the stocks are 0.002762508 0.003583136 0.006622948 0.0004543727 0.004567911
```

```
# Variance-covariance matrix
var_cov <- cov(returns)
cat('The variance covariance matrix is', var_cov, '\n')
```

```
## The variance covariance matrix is 0.00580316 0.001389264 0.001666854 0.000789581 0.001351044 0.001389264
0.009458804 0.003944643 0.0022812 0.002578939 0.001666854 0.003944643 0.01629358 0.002863584 0.001469964 0.00
0789581 0.0022812 0.002863584 0.009595202 0.003210827 0.001351044 0.002578939 0.001469964 0.003210827 0.00924
244
```

```
# c. Using only Exxon-Mobil and Boeing Stocks find the composition, expected return and standard deviation of
minimum risk portfolio

# Extract Exxon-Mobil and Boeing stocks
exxon_boeing <- returns[,c(1,5)]

# Variance-covariance matrix
var_cov <- cov(exxon_boeing)

# Find the weights of the minimum risk portfolio
weights <- solve(var_cov) %*% rep(1,2)
weights <- weights/sum(weights)
cat('The composition is', weights, '\n')
```

```
## The composition is 0.6393153 0.3606847
```

```
# Expected return
expected_return <- t(weights) %*% mean_returns[c(1,5)]
cat('The expected return is:', expected_return, '\n')
```

```
## The expected return is: 0.003413689
```

```
# Standard deviation
standard_deviation <- sqrt(t(weights) %*% var_cov %*% weights)
cat('The standard deviation is:', standard_deviation)
```

```
## The standard deviation is: 0.06478695
```

```
# d. Plot the portfolio possibilities curve and identify the efficient frontier

# Find the weights of the minimum risk portfolio

ones <- rep(1, ncol(exxon_boeing))
A <- t(ones) %*% solve(var_cov) %*% mean_returns[c(1,5)]
B <- t(mean_returns[c(1,5)]) %*% solve(var_cov) %*% mean_returns[c(1,5)]
C <- t(ones) %*% solve(var_cov) %*% ones
D <- B*C-A^2

plot(0, A/C, main = "Portfolio possibilities curve", xlab = "Risk (standard deviation)",
  ylab = "Expected Return", type = "n",
  xlim = c(-2*sqrt(1/C), 4*sqrt(1/C)),
  ylim = c(-2*A/C, 4*A/C))


#Plot transverse and conjugate axes:
    abline(v = 0) #Also this is the y-axis.
    abline(h = A/C)

#Plot the x-axis:
    abline(h = 0)

#Find the asymptotes:
    V <- seq(-1, 1, 0.001)
    A1 <- A/C + V * sqrt(D/C)
```

```
## Warning in V * sqrt(D/C): Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
## Warning in A/C + V * sqrt(D/C): Recycling array of length 1 in array-vector arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
    A2 <- A/C - V * sqrt(D/C)
```

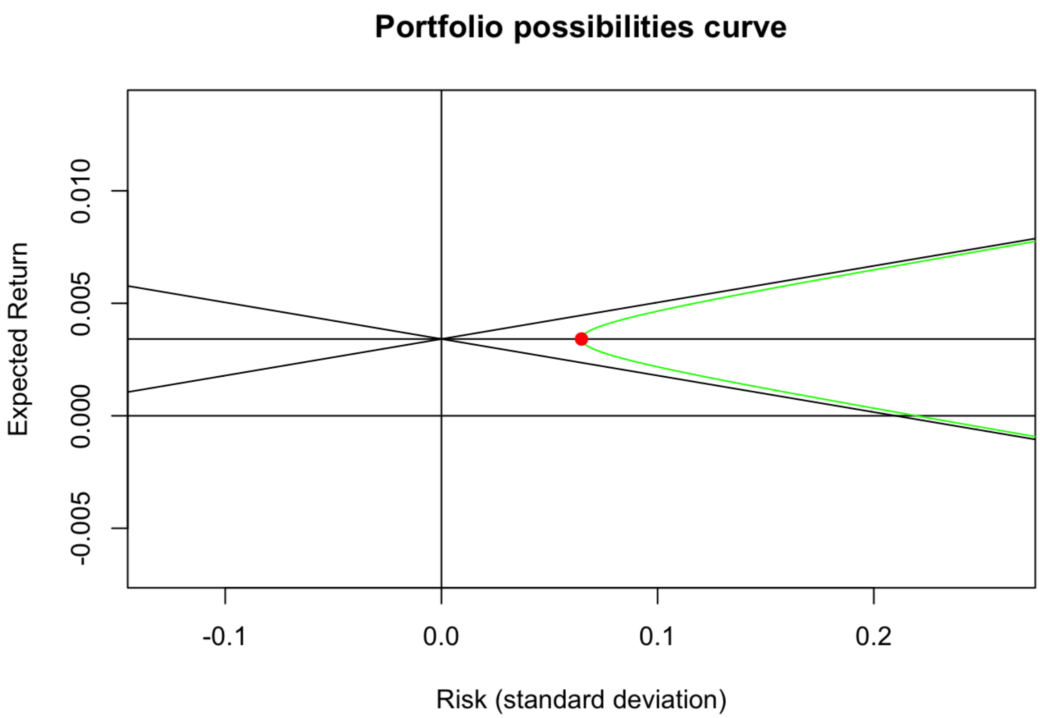
```
## Warning in V * sqrt(D/C): Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
## Warning in A/C - V * sqrt(D/C): Recycling array of length 1 in array-vector arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
    points(V, A1, type = "l")
    points(V, A2, type = "l")

#Efficient frontier:
    minvar <- 1/C
    minE <- A/C
    sdeff <- seq((minvar)^0.5, 1, by = 0.0001)
```

```
## Warning in from + (0L:n) * by: Recycling array of length 1 in array-vector arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
options(warn = -1)
y1 <- (A + sqrt(D*(C*sdeff^2 - 1)))*(1/C)
y2 <- (A - sqrt(D*(C*sdeff^2 - 1)))*(1/C)
options(warn = 0)

points(sdeff, y1, type = "l", col = 'green')
points(sdeff, y2, type = "l", col = 'green')

# Plot minimum risk portfolio
points(standard_deviation, expected_return, pch = 19, col = 'red')
```



```
exxon <- returns[,c(1)]
mcdonalds <- returns[,c(4)]
boeing <- returns[,c(5)]

# Part (e)
return_p <- rep(0,10000000);
sd_p <- rep(0,10000000);
j <- 0
i <- 0

for (a in seq(-.2, 1, 0.1)) {
    for (b in seq(-.2, 1, 0.1)) {
        for(c in seq(-.2, 1, 0.1)){
        if(a+b+c==1) {
            j=j+1
    return_p[j]=a*mean(exxon)+b*mean(mcdonalds)+
    c*mean(boeing)
    sd_p[j]=(a^2*var(exxon) +
            b^2*var(mcdonalds)+
            c^2*var(boeing) +
            2*a*b*cov(exxon,mcdonalds)+
            2*a*c*cov(exxon,boeing)+
            2*b*c*cov(mcdonalds,boeing))^.5
        }
        }
    }
}

R_p <- return_p[1:j]
sigma_p <- sd_p[1:j]

plot(sigma_p, R_p,xlab="Risk (standard deviation)", ylab="Expected return", main = "Portfolio Possibilites Cu
rve of Exxon, McDonalds, and Boeing", xlim=c(0.0,.24), ylim=c(0.0,.016),axes=FALSE, cex=0.4)
axis(1, at=c(0, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14))
axis(2,at=c(0,0.002, 0.004, 0.006, 0.008, 0.010, 0.012, 0.014, 0.016, 0.018, 0.02, 0.022, 0.024))

# Part (f)

#Construct the variance-covariance matrix:
var_covar <- cov(cbind(exxon, mcdonalds, boeing))

#Construct the vector of the expected returns:
R_ibar <- apply(cbind(exxon, mcdonalds, boeing), 2, mean)

#Compute the inverse of the variance-covariance matrix:
var_covar_inv <- solve(var_covar)

#Create the vector R:
Rf <- 0.001
R <- R_ibar-Rf


#Compute the vector Z:
z <- var_covar_inv %*% R


#Compute the vector X:
x <- z/sum(z)


#Compute the expected return of portfolio G:
R_Gbar <- t(x) %*% R_ibar
```

```
#Compute the variance and standard deviation of portfolio G:
var_G <- t(x) %*% var_covar %*% x


sd_G <- var_G^0.5


#Compute the slope:
slope <- (R_Gbar-Rf)/(sd_G)


#We have already two points on the line:  (0, 0.002) and (sd_G, R_Gbar)
#Find one more point (borrowing segment):
#(1.3*sd_G,  0.002+slope*(1.3*sd_G))

lines(c(0,sd_G, 1.3*sd_G),c(.001,R_Gbar,0.001+slope*(1.3*sd_G)))

#Identify portfolio G:
points(sd_G, R_Gbar, cex=2, col="blue", pch=19)
text(sd_G, R_Gbar+.001, "G")

# Part (g)

# Construct a portfolio of 60% G and 40% risk free asset

#Compute the expected return of portfolio F:
R_Fbar <- 0.6*R_Gbar + 0.4*Rf

#Compute the variance and standard deviation of portfolio F:
var_F <- 0.6^2*var_G

sd_F <- var_F^0.5

#Compute the slope:
slope <- (R_Fbar-Rf)/(sd_F)

points(sd_F, R_Fbar, cex=2, col="blue", pch=19)
text(sd_F, R_Fbar+.001, "F")

### Part (h)

E <- rep((R_Fbar-Rf), 3)
X <- E %*% var_covar_inv %*% (R_ibar-Rf) / (t(R_ibar-Rf) %*% var_covar_inv %*% (R_ibar-Rf))
cat("X represents the weighting of the risky asset.", X)
```

```
## X represents the weighting of the risky asset. 0.6
```

```
### Part (i)

# 1.

#Trace out the efficient frontier.  We need to find two portfolios on the efficient frontier:

#Construct the variance-covariance matrix:
var_covar <- cov(cbind(exxon, mcdonalds, boeing))

#Construct the vector of the expected returns:
R_ibar <- apply(cbind(exxon, mcdonalds, boeing), 2, mean)

#Choose two risk free rates:
Rf1 <- 0.001
Rf2 <- 0.002
#Construct the vectors RA and RB:
RA <- R_ibar-Rf1
RB <- R_ibar-Rf2

#Find the composition of the two portfolios A, B:
zA <- solve(var_covar) %*% RA
xA <- zA/sum(zA)
sum(xA)
```

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

```
zB <- solve(var_covar) %*% RB
xB <- zB/sum(zB)
sum(xB)
```

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

```
# 2.

#Compute the expected return and variance of portfolios A and B.  Also compute the covariance between portfol
io A an B:
RA_bar <- t(xA) %*% R_ibar
RB_bar <- t(xB) %*% R_ibar
var_A <- t(xA) %*% var_covar %*% xA
var_B <- t(xB) %*% var_covar %*% xB
cov_AB <- t(xA) %*% var_covar %*% xB
cat("The covariance between portfolio's A and B are", cov_AB)
```

```
## The covariance between portfolio's A and B are 0.02264823
```

```
sd_A <- var_A^.5
sd_B <- var_B^.5


#We can find now the portfolio possibilities by mixing the two  portfolios A and B:L

xa <- seq(-3, 5, 0.01)
xb <- 1-xa

#Compute the expected return and standard deviation for each combination of xa, xb:

sigma_p <- (xa^2*var_A + xb^2*var_B+ 2*xa*xb*cov_AB)^.5
```

```
## Warning in xa^2 * var_A: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
## Warning in xb^2 * var_B: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
## Warning in 2 * xa * xb * cov_AB: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
rp_bar <- xa*RA_bar + xb*RB_bar
```

```
## Warning in xa * RA_bar: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
## Warning in xb * RB_bar: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
# 3.

#Plot:
points(sigma_p, rp_bar, xlim=c(0,0.2), ylim=c(0,0.35), xlab="Portfolio standard deviation", ylab="Expected re
turn", cex=0.3, col = 'red')

points(sd_B, RB_bar, col="black", pch=19, cex=0.6)
text(sd_B-0.005, RB_bar+0.001, "B")

# 4.

#Compute the minimum risk portfolio in terms of the portfolios A and B:
xA_min <- (var_B - cov_AB)/(var_A+var_B-2*cov_AB)
xB_min <- 1-xA_min

#Find the composition of the minimum risk portfolio in terms of the three stocks:
x1_min <- xA_min*xA[1] + xB_min*xB[1]
x2_min <- xA_min*xA[2] + xB_min*xB[2]
x3_min <- xA_min*xA[3] + xB_min*xB[3]

cat("Composition of Exxon is", x1_min)
```

```
## Composition of Exxon is 0.5269063
```

```
cat("Composition of McDonalds is", x2_min)
```

```
## Composition of McDonalds is 0.2536533
```

```
cat("Composition of Boeing is", x1_min)
```

```
## Composition of Boeing is 0.5269063
```

```
#Find the expected return and standard deviation of the minimum risk portfolio:

xx <- as.matrix(c(x1_min,x2_min,x3_min))

rp_minimum <- t(xx) %*% R_ibar
sd_minimum <- (t(xx) %*% var_covar %*% xx)^.5

cat("The expected return of the min Risk Portfolio is:", rp_minimum)
```

```
## The expected return of the min Risk Portfolio is: 0.00257322
```

```
cat('The standard deviation of the minimum risk portfolio is:', sd_minimum)
```

```
## The standard deviation of the minimum risk portfolio is: 0.05961942
```

```
#Add this point on the frontier:
points(sd_minimum, rp_minimum, cex=2, pch=19)
```