STATS183 Project 3

Takao Oba

2023-04-14

Access the following data: http://www.stat.ucla.edu/~nchristo/statistics_c183_c283/statc183c283_5stocks.txt. In R you can access the data from the command line as follows: 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 the prices into returns for all the 5 stocks. Important note: In this data set the most recent data are at the beginning. You will need to consider this when converting the prices into returns.

```r
a <- read.table("http://www.stat.ucla.edu/~nchristo/statistics_c183_c283/statc183c283_5stocks.txt", hea
# ordering properly by date
a <- a[order(a$date),]

# converting to return
r <- (a[-1,2:ncol(a)] - a[-nrow(a), 2:ncol(a)])/a[-nrow(a), 2:ncol(a)]
head(r)
```

```
##               P1          P2           P3          P4          P5
## 215  0.009661836  0.05067568  0.114285714  0.11893584  0.10621762
## 214  0.066985646  0.10932476  0.002849003  0.07832168  0.06791569
## 213  0.015695067 -0.07246377  0.028409091  0.03242542 -0.01315789
## 212  0.057395143 -0.00312500  0.022099448  0.03140704  0.03555556
## 211  0.016701461 -0.02664577 -0.113513514 -0.28745432  0.08154506
## 210 -0.002053388 -0.12238325 -0.039634146 -0.12478632 -0.06150794
```

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

```r
# mean return for each stock
means <- colMeans(r)
means
```

```
##           P1           P2           P3           P4           P5
## 0.0027625075 0.0035831363 0.0066229478 0.0004543727 0.0045679106
```

```r
# variance-covariance matrix
covmat <- cov(r)
covmat
```

```
##             P1          P2          P3          P4          P5
## P1 0.005803160 0.001389264 0.001666854 0.000789581 0.001351044
## P2 0.001389264 0.009458804 0.003944643 0.002281200 0.002578939
## P3 0.001666854 0.003944643 0.016293581 0.002863584 0.001469964
## P4 0.000789581 0.002281200 0.002863584 0.009595202 0.003210827
## P5 0.001351044 0.002578939 0.001469964 0.003210827 0.009242440
```

c. Use only Exxon-Mobil and Boeing stocks: For these 2 stocks find the composition, expected return, and standard deviation of the minimum risk portfolio.

```r
# Exxon-Mobil is represented as P1
# Boeing is represented as P5
exxon_boeing <- r[,c(1,5)]
head(exxon_boeing)
```

```
##               P1          P5
## 215  0.009661836  0.10621762
## 214  0.066985646  0.06791569
## 213  0.015695067 -0.01315789
## 212  0.057395143  0.03555556
## 211  0.016701461  0.08154506
## 210 -0.002053388 -0.06150794
```

```r
means2 <- colMeans(exxon_boeing)
covmat2 <- cov(exxon_boeing)

ones2 <- rep(1,2)

# composition
weights <- (solve(covmat2)%*%ones2/(as.numeric(t(ones2)%*%solve(covmat2)%*%ones2)))
weights
```

```
##         [,1]
## P1 0.6393153
## P5 0.3606847
```

```r
# expected return
exp_return <- t(weights) %*% means2
exp_return
```

```
##             [,1]
## [1,] 0.003413689
```

```
# standard deviation of minimum risk portfolio
sd2 <- sqrt(t(weights) %*% covmat2 %*% weights)
sd2
```

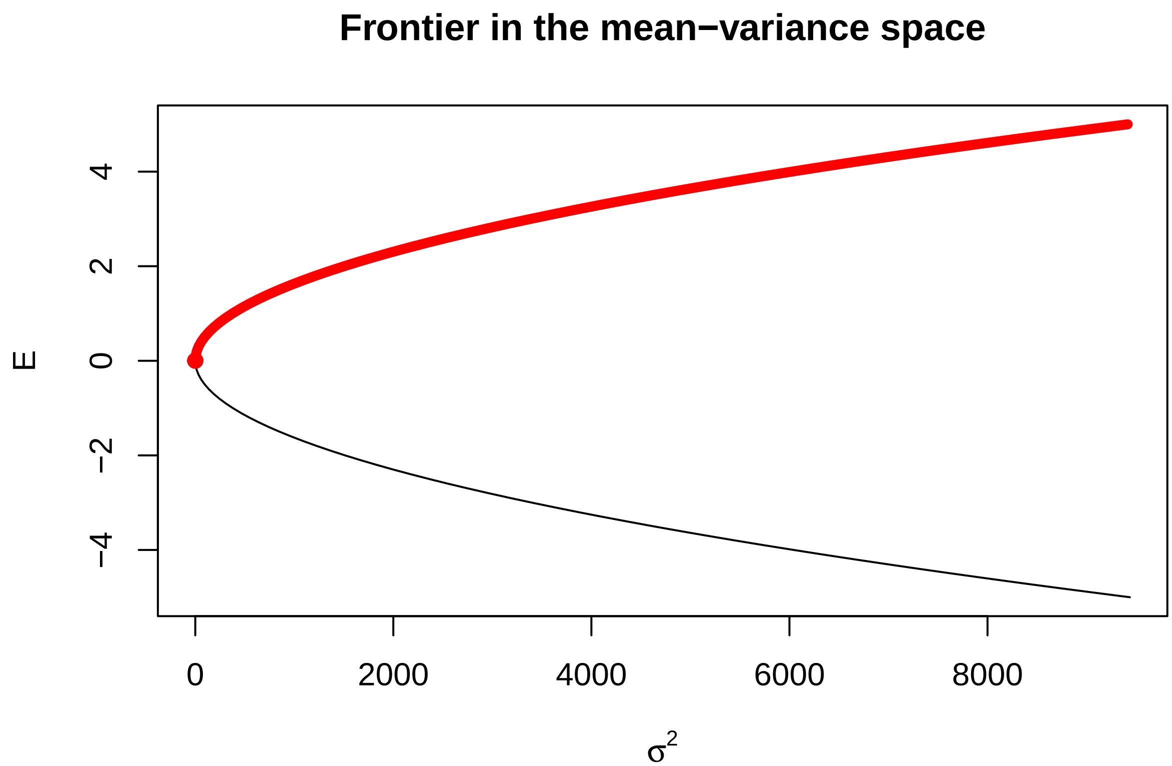
```
##            [,1]
## [1,] 0.06478695
```

d. Plot the portfolio possibilities curve and identify the efficient frontier on it.

```
ones <- rep(1,5)
# Computing A,B,C,D
A <- t(ones) %*% solve(covmat) %*% means
B <- t(means) %*% solve(covmat) %*% means
C <- t(ones) %*% solve(covmat) %*% ones
D <- B*C - A^2

E <- seq(-5,5, 0.1)
sigma2 <- (C*E^2 - 2*A*E +B) /D
```

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

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

## Warning in C * E^2 - 2 * A * E + B: Recycling array of length 1 in vector-array arithmetic is depreca
##   Use c() or as.vector() instead.

## Warning in (C * E^2 - 2 * A * E + B)/D: Recycling array of length 1 in vector-array arithmetic is dep
##   Use c() or as.vector() instead.
```

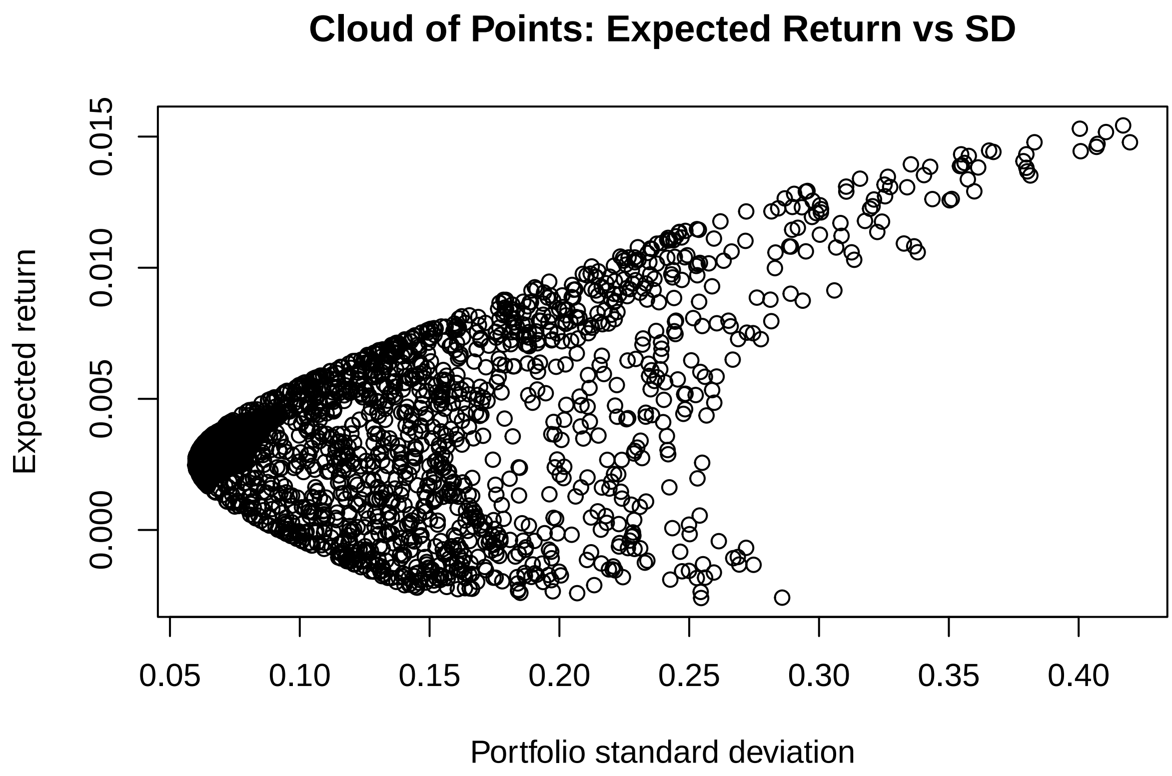
```
#plot E against sigma2:
plot(sigma2, E,type="l", xlab=expression(sigma^2), main = "Frontier in the mean-variance space")
points(1/C, A/C, col = "red", pch = 19)

# coloring the side that is considered efficient frontier
E_2 <- E[51:101]
sigma2 <- sigma2[51:101]
points(sigma2, E_2, lwd = 5, type = "l", col = "red")
```



e. Use only Exxon-Mobil, McDonalds and Boeing stocks and assume short sales are allowed to answer the following question: For these 3 stocks compute the expected return and standard deviation for many combinations of xa, xb, xc with xa + xb + xc = 1 and plot the cloud of points. You can use the following combinations of the three stocks: a <- read.table("http://www.stat.ucla.edu/~nchristo/datac183c283/statc183c283_abc.txt", header=T)

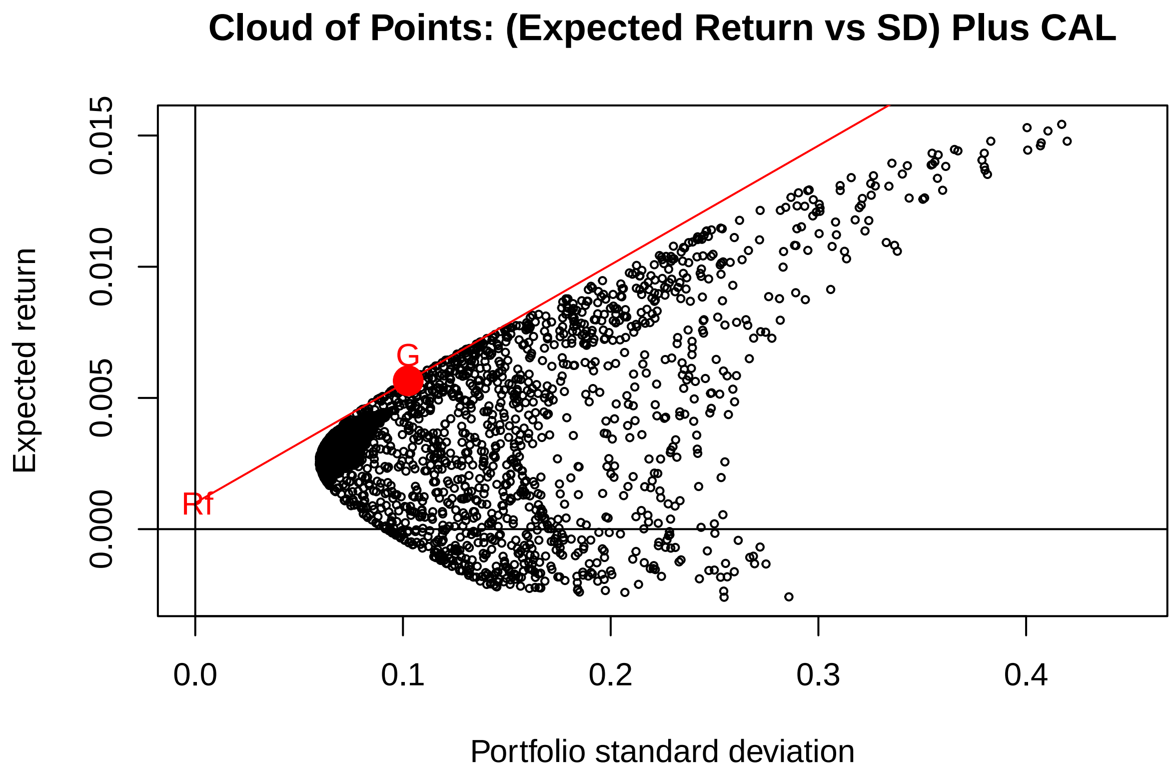
```
w <- read.table("http://www.stat.ucla.edu/~nchristo/datac183c283/statc183c283_abc.txt", header=T)

# Exxon-Mobil is represented as P1
# McDonalds is represented as P4
# Boeing is represented as P5
new_port <- r[,c(1,4,5)]

#Compute the standard deviation of each portfolio:
mexon <- mean(new_port$P1)
mmcd <- mean(new_port$P4)
mboeing <- mean(new_port$P5)
```

```
vexon <- var(new_port$P1)
vmcd <- var(new_port$P4)
vboeing <- var(new_port$P5)

c12 <- cov(new_port$P1, new_port$P4)
c13 <- cov(new_port$P1, new_port$P5)
c23 <- cov(new_port$P4, new_port$P5)

#Compute the expected return of each portfolio:
rp_bar <- w$a*mexon +w$b*mmcd +w$c*mboeing

# risk
sigma_p <- ((w$a)^2*vexon +(w$b)^2* vmcd+(w$c)^2*vboeing+2*w$a*w$b*c12+2*w$a*w$c*c13+2*w$b*w$c*c23)^.5

rp_bar <- as.matrix(rp_bar)

#Plot:
plot(sigma_p, rp_bar, type = "n", xlab="Portfolio standard deviation",  ylab="Expected return", main =

    points(sigma_p, rp_bar)
```



f. Assume Rf = 0.001 and that short sales are allowed. Find the composition, expected return and standard deviation of the portfolio of the point of tangency G and draw the tangent to the efficient frontier of question (e).

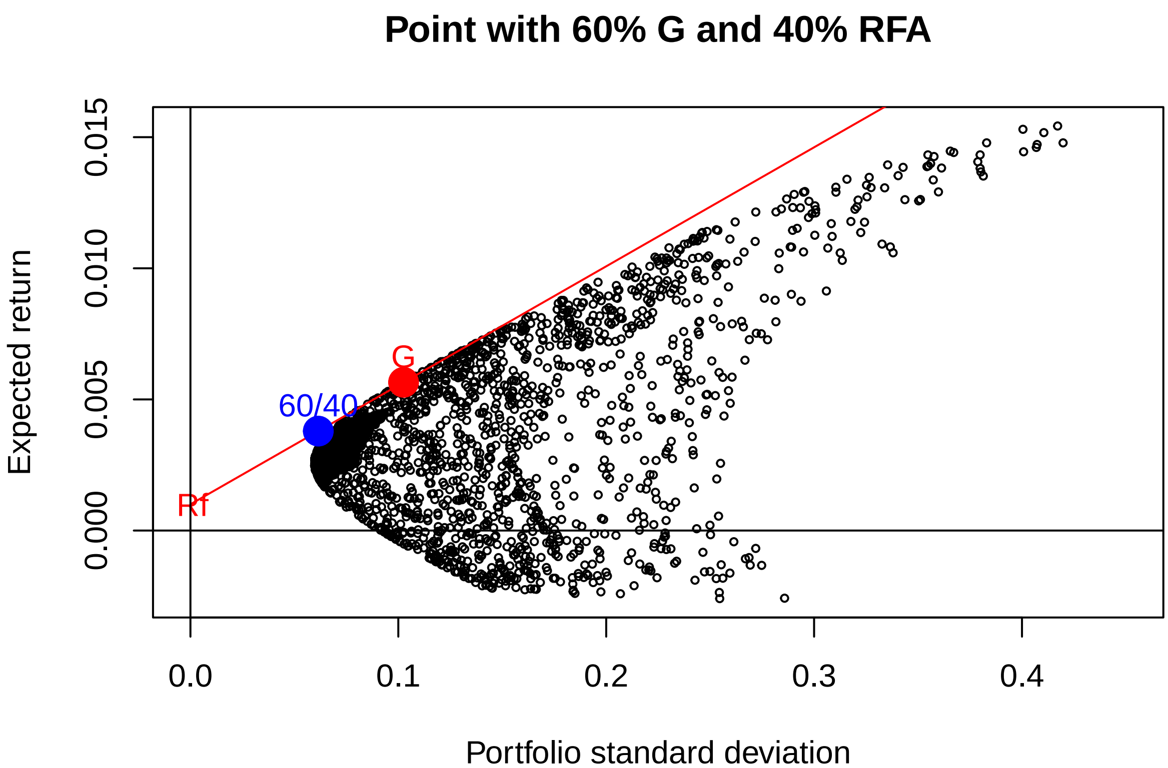
```
#Choose risk-free return:
Rf <- 0.001
means <- colMeans(new_port)
covmat <- cov(new_port)
#===================================
#===================================
#Point of tangency:
R <- means-Rf
z <- solve(covmat) %*% R

xx <- z/sum(z)

rr <- t(xx) %*% means
varr <- t(xx) %*% covmat %*% xx
sdev <- varr^.5

#Plot:
plot(sigma_p, rp_bar, type = "n", xlab="Portfolio standard deviation",  ylab="Expected return", xlim = c

points(sigma_p, rp_bar, cex = 0.5)

points(sdev,rr, col = "red", pch = 19, cex = 2)
text(sdev,rr+0.001, "G", col = "red")


# Capital Allocation Line
slope <- as.numeric((rr - Rf) / sdev)
x_ax <- seq(0, 0.45, 0.001)
y_ax <- Rf + slope*x_ax
text(0+0.001, Rf, "Rf", col = "red")

lines(x_ax, y_ax, col = "red")
abline(v = 0)
abline(h = 0)
```



g. Find the expected return and standard deviation of the portfolio that consists of 60% and G 40% risk free asset. Show this position on the capital allocation line (CAL).

```
port_on_CAL <- sdev*0.6

##########################
#Plot:
plot(sigma_p, rp_bar, type = "n", xlab="Portfolio standard deviation",  ylab="Expected return", xlim = c

points(sigma_p, rp_bar, cex = 0.5)

points(sdev,rr, col = "red", pch = 19, cex = 2)
text(sdev,rr+0.001, "G", col = "red")


# Capital Allocation Line
slope <- as.numeric((rr - Rf) / sdev)
x_ax <- seq(0, 0.45, 0.001)
y_ax <- Rf + slope*x_ax
text(0+0.001, Rf, "Rf", col = "red")
```

```
lines(x_ax, y_ax, col = "red")
abline(v = 0)
abline(h = 0)
###########################

points(port_on_CAL, Rf + port_on_CAL*slope, col = "Blue", pch = 19, cex = 2)
text(port_on_CAL, Rf + port_on_CAL*slope+0.001,"60/40" , col = "Blue")
```



h. Refer to question (g). Use the expected value (E) you found in (g) to compute

$$X = \frac{(E - R_f)\Sigma^{-1}(\bar{R} - R_f 1)}{(\bar{R} - R_f 1)'\Sigma^{-1}(\bar{R} - R_f 1)}$$

What does this x represent?

```
ones <- c(1,1,1)
E <- Rf + port_on_CAL*slope

X <- ((as.numeric(E - Rf))*solve(covmat)%*%(t(means - Rf %*%ones)))/as.numeric(((means - Rf %*%ones))%*
X
```

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

```
## P1  0.3170869
## P4 -0.2973529
## P5  0.5802660
```

```
sum(X)
```

```
## [1] 0.6
```

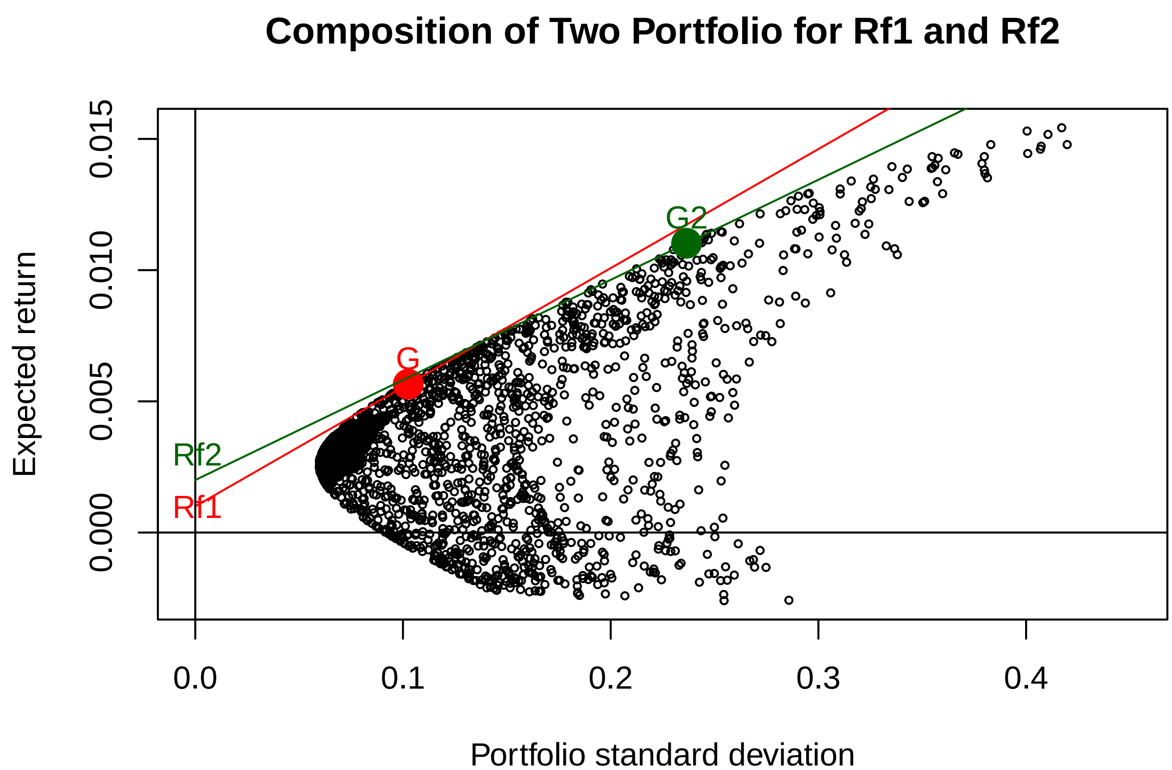
ANSWER: X has a size of the amount of stocks that are in the portfoliio. In this case, we have three values that corresponds to the percentage invested in P1 (Exxon), P4 (McDonald's), P5 (Boeing). We notice that the sum of these three values is 0.6 as we will be investing the other components in the risk free asset.

i. Now assume that short sales are allowed but risk free asset does not exist.

1. Using Rf1 = 0.001 and Rf2 = 0.002 find the composition of two portfolios A and B (tangent to the efficient frontier - you found the one with Rf1 = 0.001 in question (f)).

```
Rf2 <- 0.002


##########################
#Plot:
plot(sigma_p, rp_bar, type = "n", xlab="Portfolio standard deviation",  ylab="Expected return", xlim = c

points(sigma_p, rp_bar, cex = 0.5)

points(sdev,rr, col = "red", pch = 19, cex = 2)
text(sdev,rr+0.001, "G", col = "red")


# Capital Allocation Line
slope <- as.numeric((rr - Rf) / sdev)
x_ax <- seq(0, 0.45, 0.001)
y_ax <- Rf + slope*x_ax
text(0+0.001, Rf, "Rf1", col = "red")

lines(x_ax, y_ax, col = "red")
abline(v = 0)
abline(h = 0)
##########################


# Point of Tangency 2
R2 <- means-Rf2
z2 <- solve(covmat) %*% R2
```

```r
xx2 <- z2/sum(z2)

rr2 <- t(xx2) %*% means
varr2 <- t(xx2) %*% covmat %*% xx2
sdev2 <- varr2^.5
points(sdev2,rr2, col = "Dark Green", pch = 19, cex = 2)
text(sdev2,rr2+0.001, "G2", col = "Dark Green")

# Capital Allocation Line 2

slope2 <- as.numeric((rr2-Rf2)/sdev2)
x_ax2 <- seq(0,0.45, 0.001)
y_ax2 <- Rf2 + slope2*x_ax2
text(0 + 0.001, Rf2+0.001, "Rf2", col = "Dark Green")
lines(x_ax2, y_ax2, col = "Dark Green")
```



2. Compute the covariance between portfolios A and B?

```r
# Weights of A
xx
```

```
##          [,1]
## P1  0.5284782
```

```
## P4 -0.4955882
## P5  0.9671100
```

```r
# Weights of B
xx2
```

```
##           [,1]
## P1  0.5312205
## P4 -1.8026632
## P5  2.2714427
```

```r
# covariance between portfolio A and B
cov_ab <- t(xx) %*% covmat %*% xx2
cov_ab
```

```
##            [,1]
## [1,] 0.02264823
```

3. Use your answers to (1) and (2) to trace out the efficient frontier of the stocks Exxon-Mobil, McDonalds, Boeing. Use a different color to show that the frontier is located on top of the cloud of points from question (e). Your graph should look like the one below

```r
################################################################################
###########################
#Plot:
plot(sigma_p, rp_bar, type = "n", xlab="Portfolio standard deviation",  ylab="Expected return", xlim = c

points(sigma_p, rp_bar, cex = 0.5)

points(sdev,rr, col = "red", pch = 19, cex = 2)
text(sdev,rr+0.001, "G", col = "red")


# Capital Allocation Line
text(0+0.001, Rf, "Rf1", col = "red")

lines(x_ax, y_ax, col = "red")
abline(v = 0)
abline(h = 0)
###########################


# Point of Tangency 2
points(sdev2,rr2, col = "Dark Green", pch = 19, cex = 2)
text(sdev2,rr2+0.001, "G2", col = "Dark Green")

# Capital Allocation Line 2
text(0 + 0.001, Rf2+0.001, "Rf2", col = "Dark Green")
lines(x_ax2, y_ax2, col = "Dark Green")
```

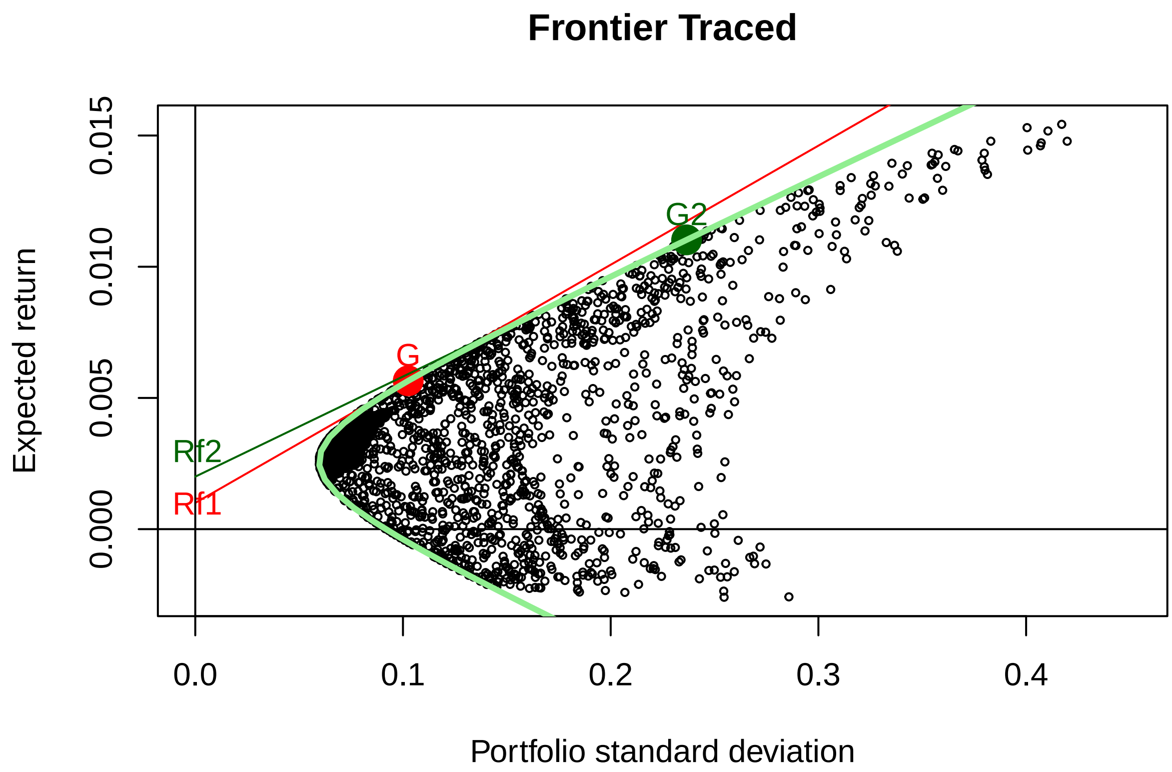
```r
################################################################################

#Mean1:
m1 <- t(xx) %*% means

#Variance1:
v1 <- t(xx) %*% covmat %*% xx


#Mean2:
m2 <- t(xx2) %*% means

#Variance2:
v2 <- t(xx2) %*% covmat %*% xx2



a <- seq(-3,3,.1)
b <- 1-a

r_ab <- a*m1 + b*m2
```

```
## Warning in a * m1: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.

## Warning in b * m2: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```r
var_ab <- a^2*v1 + b^2*v2 + 2*a*b*cov_ab
```

```
## Warning in a^2 * v1: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.

## Warning in b^2 * v2: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.

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

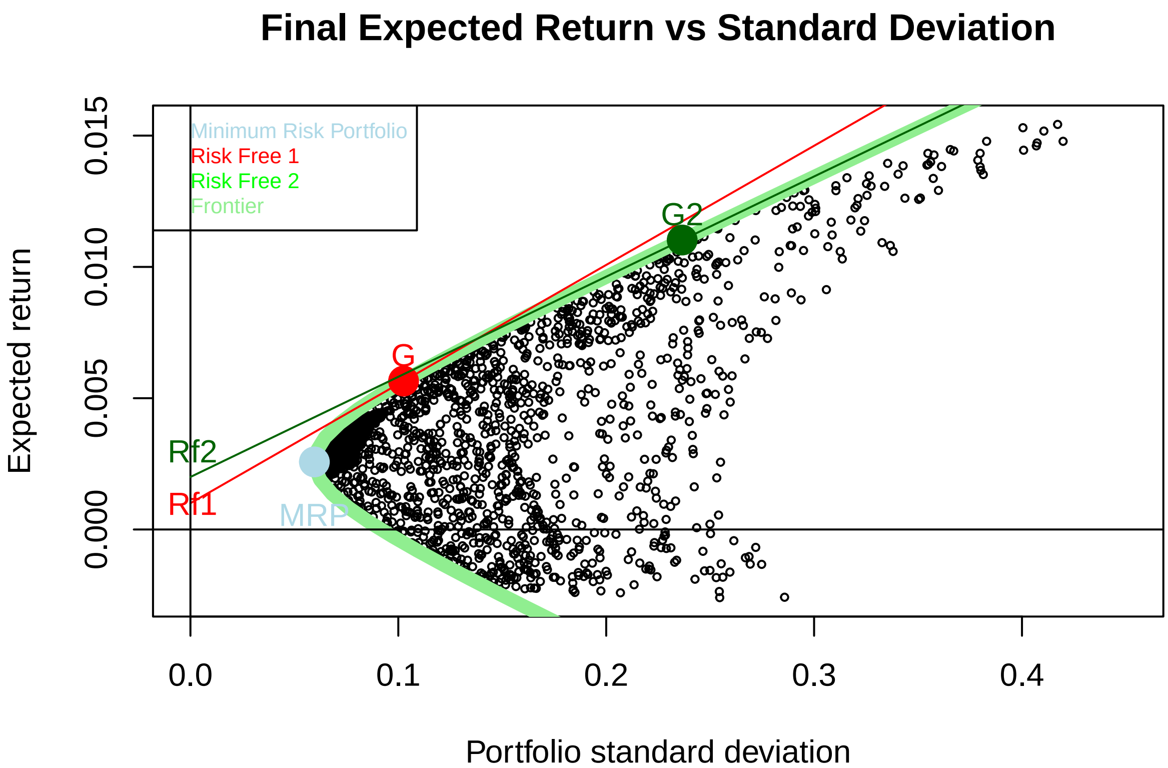
```r
sd_ab <- var_ab^.5

# tracing out the efficient frontier
points(sd_ab, r_ab, type = "l", lwd = 3, col="light green")
```



4. Find the composition of the minimum risk portfolio using the three stocks (how much of each stock) and its expected return, and standard deviation.

```
##########################################################################################################
################################################################################
###########################
#Plot:
plot(sigma_p, rp_bar, type = "n", xlab="Portfolio standard deviation",  ylab="Expected return", xlim = c

points(sigma_p, rp_bar, cex = 0.5)

# tracing out the efficient frontier
points(sd_ab, r_ab, type = "l", lwd = 7, col="light green")

points(sdev,rr, col = "red", pch = 19, cex = 2)
text(sdev,rr+0.001, "G", col = "red")


# Capital Allocation Line
text(0+0.001, Rf, "Rf1", col = "red")

lines(x_ax, y_ax, col = "red")
abline(v = 0)
abline(h = 0)
```

```r
###########################


# Point of Tangency 2
points(sdev2,rr2, col = "Dark Green", pch = 19, cex = 2)
text(sdev2,rr2+0.001, "G2", col = "Dark Green")

# Capital Allocation Line 2
text(0 + 0.001, Rf2+0.001, "Rf2", col = "Dark Green")
lines(x_ax2, y_ax2, col = "Dark Green")
#################################################################################

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


#Compute A:
A <- t(ones) %*% solve(covmat) %*% means

#Compute B:
B <- t(means) %*% solve(covmat) %*% means

#Compute C:
C <- t(ones) %*% solve(covmat) %*% ones

#Compute D:
D <- B*C - A^2

points(sqrt(1/C), A/C, pch = 19, cex = 2, col = "light blue")
text(sqrt(1/C), A/C - 0.002, "MRP", col = "light blue")

legend(x = 'topleft',cex  = 0.65,
       legend = c("Minimum Risk Portfolio","Risk Free 1", "Risk Free 2", "Frontier"),
       text.col = c("light blue", "Red", "Green", "light green"))
```




```r
x <- solve(covmat) %*% ones / as.numeric(t(ones) %*% solve(covmat) %*% ones)
# composition
x
```

```
##         [,1]
## P1 0.5269063
## P4 0.2536533
## P5 0.2194404
```