STATS183 Project 6

Takao Oba

2023-05-05

a. Use only the stocks with positive betas in your data. Rank the stocks based on the excess return to beta ratio and complete the entire table based on handout #28:

```
#Read your csv file:
a <- read.csv("/Users/takaooba/STATS 183/stockData.csv", sep=",", header=TRUE)
train <- a[1:60,]
test <- a[61:dim(a)[1],]

#Convert adjusted close prices into returns:
r <- (train[-1,3:ncol(train)]-train[-nrow(train),3:ncol(train)])/train[-nrow(train),3:ncol(train)]
```

using C*

```
#Compute the betas:
covmat <- var(r)
beta <- covmat[1,-1] / covmat[1,1]

#Keep only the stocks with positive betas:
rrr <- r[,-c(1,which(beta<0)+1)]

#all of the stocks have positive betas
length(rrr)
```

```
## [1] 30
```

```
#Note: which(beta<0) gives the element in the beta vector with negative beta and add 1 because
#the first column in the iitial data set is the index.
# We also remove column 1 (index) from the initial data #set.

#Initialize
beta <- rep(0,ncol(rrr))
alpha <- rep(0,ncol(rrr))
mse <- rep(0,ncol(rrr))
Ribar <- rep(0,ncol(rrr))
Ratio <- rep(0,ncol(rrr))
stock <- rep(0,ncol(rrr))

#Risk free asset:
rf <- 0.001
```

```r
#This for loop computes the required inputs:
for(i in 1:ncol(rrr)){
    q <- lm(data=rrr, formula=rrr[,i] ~ r[,1])
    beta[i] <- q$coefficients[2]
    alpha[i] <- q$coefficients[1]
     mse[i] <- summary(q)$sigma^2
    Ribar[i] <- q$coefficients[1]+q$coefficients[2]*mean(r[,1])
    Ratio[i] <- (Ribar[i]-rf)/beta[i] # (R_i - R_f)/B_i
    stock[i] <- i
}


#So far we have this table:
xx <- (cbind(stock,alpha, beta, Ribar, mse, Ratio))




#Order the table based on the excess return to beta ratio:
A <- xx[order(-xx[,6]),]

col1 <- rep(0,nrow(A))
col2 <- rep(0,nrow(A))
col3 <- rep(0,nrow(A))
col4 <- rep(0,nrow(A))
col5 <- rep(0,nrow(A))


#Create the last 5 columns of the table:
col1 <- (A[,4]-rf)*A[,3]/A[,5]
col3 <- A[,3]^2/A[,5]
for(i in(1:nrow(A))) {
col2[i] <- sum(col1[1:i])
col4[i] <- sum(col3[1:i])
}

#So far we have:
# head(cbind(A, col1, col2, col3, col4))

#Compute the Ci (col5):
for(i in (1:nrow(A))) {
col5[i] <- var(r[,1])*col2[i]/(1+var(r[,1])*col4[i])
}

A1 <- A
#The final table when short sales allowed:
B1 <- cbind(A1, col1, col2, col3, col4, col5)
rownames(B1) <- NULL
head(B1)
```

```
##      stock      alpha      beta      Ribar         mse      Ratio     col1
## [1,]     1 0.01230233 0.4415918 0.01618445 0.001534396 0.03438572 4.370012
## [2,]     6 0.01936647 0.7979014 0.02638098 0.009592057 0.03180968 2.111280
## [3,]    14 0.01422696 0.6752743 0.02016343 0.002543818 0.02837873 5.087066
## [4,]     3 0.01003061 0.5401055 0.01477879 0.002532305 0.02551129 2.938824
## [5,]    18 0.01434742 0.8004113 0.02138399 0.002228809 0.02546690 7.320310
## [6,]    29 0.03383867 2.0248857 0.05163985 0.011146357 0.02500874 9.199409
##           col2      col3      col4        col5
## [1,]  4.370012 127.08801  127.0880 0.004509038
## [2,]  6.481293  66.37226  193.4603 0.006258855
## [3,] 11.568359 179.25628  372.7166 0.009522882
## [4,] 14.507183 115.19700  487.9136 0.010907714
## [5,] 21.827493 287.44414  775.3577 0.013495111
## [6,] 31.026902 367.84772 1143.2054 0.015628438
```

b. Find the composition of the point of tangency with and without short sales allowed. Place the two portfolios on the plot with the 30 stocks, S&P500, and the efficient frontier that you constructed in the previous projects. Your answer for the short sales case should be the same as in project 4, part (a).

When short sales are allowed, we can consider all of the stocks. However, when short sales are not allowed, we have to see when the C_star value becomes greater than the ratio.

```r
# # B1
table2 <- data.frame(B1[1:which(col5==max(col5)), ])


table3 <- data.frame(ncol = 59)
for (i in table2$stock){
  table3 <- cbind(table3, r[i+1])
}

r2 <- table3[,-1]

means2 <- colMeans(r2)
covmat2 <- cov(r2)
```

```r
Rfr <- seq(-0.05,.03,0.0005)

#Initialize the two vectors:
rbar_opt <- rep(0,length(Rfr))
risk_opt <- rep(0,length(Rfr))


for(l in 1:length(Rfr)){
#Risk free asset:
rf <- Rfr[l]
#rf <- .002
#Initialize
beta <- rep(0,ncol(rrr))
alpha <- rep(0,ncol(rrr))
mse <- rep(0,ncol(rrr))
Ribar <- rep(0,ncol(rrr))
Ratio <- rep(0,ncol(rrr))
stocknum <- rep(0,ncol(rrr))
```

```r
#stock <- names(rrr)

#This for loop computes the required inputs:
for(i in 1:ncol(rrr)){
    q <- lm(data=rrr, formula=rrr[,i] ~ r[,1])
    beta[i] <- q$coefficients[2]
    alpha[i] <- q$coefficients[1]
     mse[i] <- summary(q)$sigma^2
    Ribar[i] <- q$coefficients[1]+q$coefficients[2]*mean(r[,1])
    Ratio[i] <- (Ribar[i]-rf)/beta[i]
    stocknum[i] <- i
}

#So far we have this table:
#xx <- (cbind(stock,alpha, beta, Ribar, mse, Ratio))
xx <- (data.frame(stocknum,alpha, beta, Ribar, mse, Ratio))


#Order the table based on the excess return to beta ratio:
A <- xx[order(-xx[,6]),]

col1 <- rep(0,nrow(A))
col2 <- rep(0,nrow(A))
col3 <- rep(0,nrow(A))
col4 <- rep(0,nrow(A))
col5 <- rep(0,nrow(A))


#Create the last 5 columns of the table:
col1 <- (A[,4]-rf)*A[,3]/A[,5]
col3 <- A[,3]^2/A[,5]
for(i in(1:nrow(A))) {
col2[i] <- sum(col1[1:i])
col4[i] <- sum(col3[1:i])
}

#So far we have:
cbind(A, col1, col2, col3, col4)


#Compute the Ci (col5):
for(i in (1:nrow(A))) {
col5[i] <- var(r[,1])*col2[i]/(1+var(r[,1])*col4[i])
}


#The final table when short sales allowed:
B <- cbind(A, col1, col2, col3, col4, col5)
rownames(B) <- NULL

#SHORT SALES NOT ALLOWED:
#First create a matrix up to the maximum of col5:
#table1 <- cbind(A, col1, col2, col3, col4, col5)
```

```
#table2 <- (B[1:which(col5==max(col5)), ], nrow=which(col5==max(col5)), ncol=ncol(B))
table2 <- B[1:which(col5==max(col5)), ]

#Compute the Zi:
z_no_short <- (table2[,3]/table2[,5])*(table2[,6]-max(col5))

#Compute the xi:
x_no_short <- z_no_short/sum(z_no_short)

#Compute the mean and variance for each portfolio when short sales not allowed:
#First match the columns of the data with the composition of the portfolio:
r1 <- data.frame(rrr[,table2[,1]])

beta1 <- rep(0,ncol(r1))
sigma_e1 <- rep(0,ncol(r1))
alpha1 <- rep(0,ncol(r1))

for(i in 1:ncol(r1)){
    q1<- lm(r1[,i] ~ r[,1])
beta1[i] <- q1$coefficients[2]
sigma_e1[i] <- summary(q1)$sigma^2
alpha1[i] <- q1$coefficients[1]
}

means1 <- colMeans(r1)
#means1 <- alpha1 + beta1*mean(r[,1])


#Construct the variance covariance matrix using SIM:
xx <- rep(0,ncol(r1)*(ncol(r1)))             #Initialize
varcovar <- matrix(xx,nrow=ncol(r1),ncol=ncol(r1))  #the variance covariance matrix


for (i in 1:ncol(r1)){
    for (j in 1:ncol(r1)){
        varcovar[i,j]=beta1[i]*beta1[j]*var(r[,1])
        if(i==j){varcovar[i,j]=beta1[i]^2*var(r[,1])+ sigma_e1[i]}
        }
        }


rbar_opt[l] <- t(x_no_short) %*% means1
risk_opt[l] <- ( t(x_no_short) %*% varcovar %*% x_no_short )^.5

}
```

```
#Compute variance covariance matrix:
covmat <- cov(r[-1]) #Without ^GSPC

#Compute the vector of standard deviations:
stdev <- diag(covmat)^.5

#Compute mean vector:
```

```r
means <- colMeans(r[-1]) #Without ^GSPC

# one vector
ones <- rep(1,30)

# mean vector of SP500
means_sp500 <- mean(r[,1])
stdev_sp500 <- sd(r[,1])

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

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

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

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


#Hyperbola:
#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.
```

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

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

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

## Warning in A + sqrt(D * (C * sdeff^2 - 1)): Recycling array of length 1 in array-vector arithmetic i
##   Use c() or as.vector() instead.

## Warning in (A + sqrt(D * (C * sdeff^2 - 1))) * (1/C): Recycling array of length 1 in vector-array ar
##   Use c() or as.vector() instead.
```

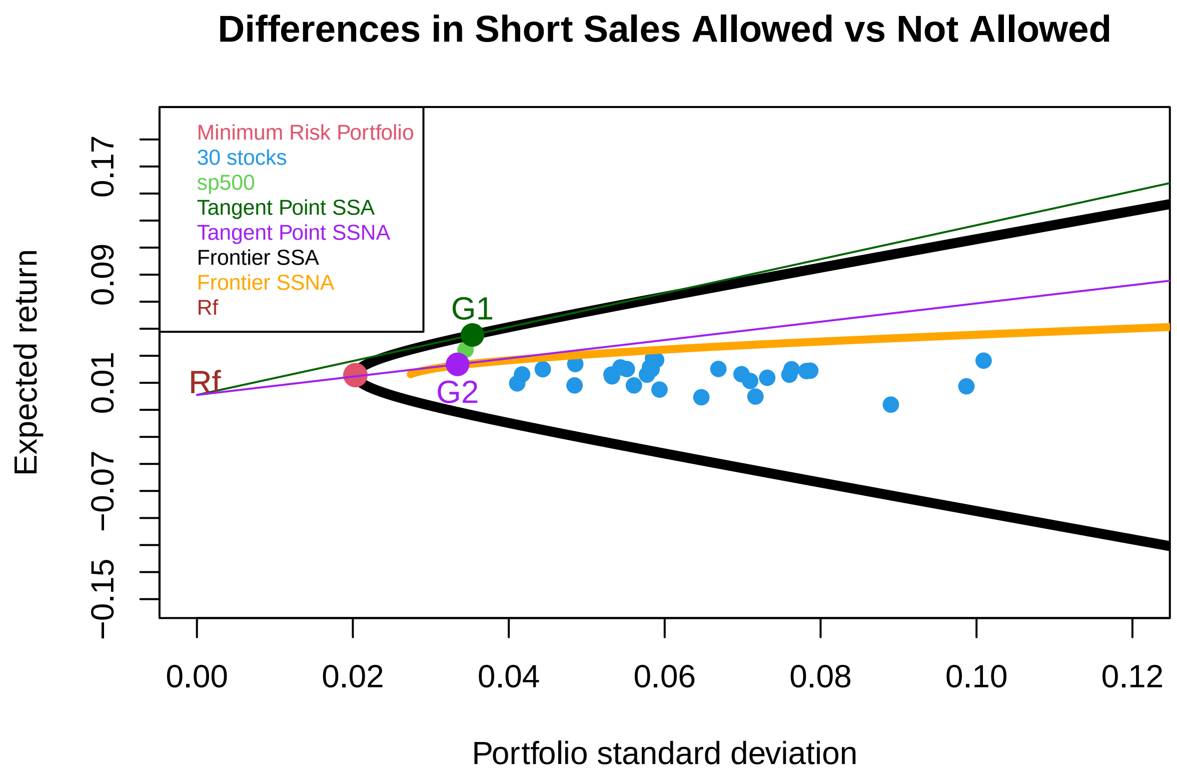
```
    y2 <- (A - sqrt(D*(C*sdeff^2 - 1)))*(1/C)
```

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

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

## Warning in A - sqrt(D * (C * sdeff^2 - 1)): Recycling array of length 1 in array-vector arithmetic is
##   Use c() or as.vector() instead.

## Warning in (A - sqrt(D * (C * sdeff^2 - 1))) * (1/C): Recycling array of length 1 in vector-array ar
##   Use c() or as.vector() instead.
```

```
#    options(warn = 0)
```

```
plot(sdeff, y1, type = "n",xlim=c(0 ,0.12), ylim=c(-0.15,0.2),
     xlab="Portfolio standard deviation", ylab="Expected return",
     xaxt="no", yaxt="no", main = "Differences in Short Sales Allowed vs Not Allowed")

axis(1, at=seq(0, 0.15, 0.02))
axis(2, at=seq(-0.15,0.2, 0.02))

    points(sdeff, y1, lwd=5,type = "l")
    points(sdeff, y2, lwd=5,type = "l")

# min risk portfolio
points(sqrt(1/C), A/C, pch = 19, col = 10, lwd = 5)
# 30 stocks
points(stdev, means, pch = 19, col = 12)
# sp500
points(stdev_sp500, stdev_sp500, pch = 19, col = 3)



legend(x = 'topleft',cex  = 0.65,
       legend = c("Minimum Risk Portfolio","30 stocks", "sp500", "Tangent Point SSA", "Tangent Point SS
       text.col = c(10, 12, 3, "Dark Green", "Purple", "Black", "Orange", "Brown"))



########## Add Point of Tangency (Short Selling Allowed)
rf <- 0.001
R2 <- means-rf
z2 <- solve(covmat) %*% R2

xx2 <- z2/sum(z2)

rr2 <- t(xx2) %*% means
varr2 <- t(xx2) %*% covmat %*% xx2
sdev2 <- varr2^.5
```

```r
points(sdev2,rr2, col = "Dark Green", pch = 19, cex = 1.5)
text(sdev2,rr2+0.02, "G1", col = "Dark Green")

# Capital Allocation Line 2

slope2 <- as.numeric((rr2-rf)/sdev2)
x_ax2 <- seq(0,0.45, 0.001)
y_ax2 <- rf + slope2*x_ax2
text(0 + 0.001, rf+0.01, "Rf", col = "Dark Green")
lines(x_ax2, y_ax2, col = "Dark Green")



########## Add Point of Tangency (Short Selling NOT Allowed)

# frontier
points(risk_opt, rbar_opt, type="l", col = "orange", lwd = 4)

means2 <- colMeans(r2)
covmat2 <- cov(r2)

R3 <- means2-rf
z3 <- solve(covmat2) %*% R3

xx3 <- z3/sum(z3)

rr3 <- t(xx3) %*% means2
varr3 <- t(xx3) %*% covmat2 %*% xx3
sdev3 <- varr3^.5

# Tangent Point
points(sdev3,rr3, col = "Purple", pch = 19, cex = 1.5)
text(sdev3,rr3-0.02, "G2", col = "Purple")



# Capital Allocation Line 2

slope3 <- as.numeric((rr3-rf)/sdev3)
x_ax3 <- seq(0,0.45, 0.001)
y_ax3 <- rf + slope3*x_ax3
text(0 + 0.001, rf+0.01, "Rf", col = "Brown")
lines(x_ax3, y_ax3, col = "Purple")
```

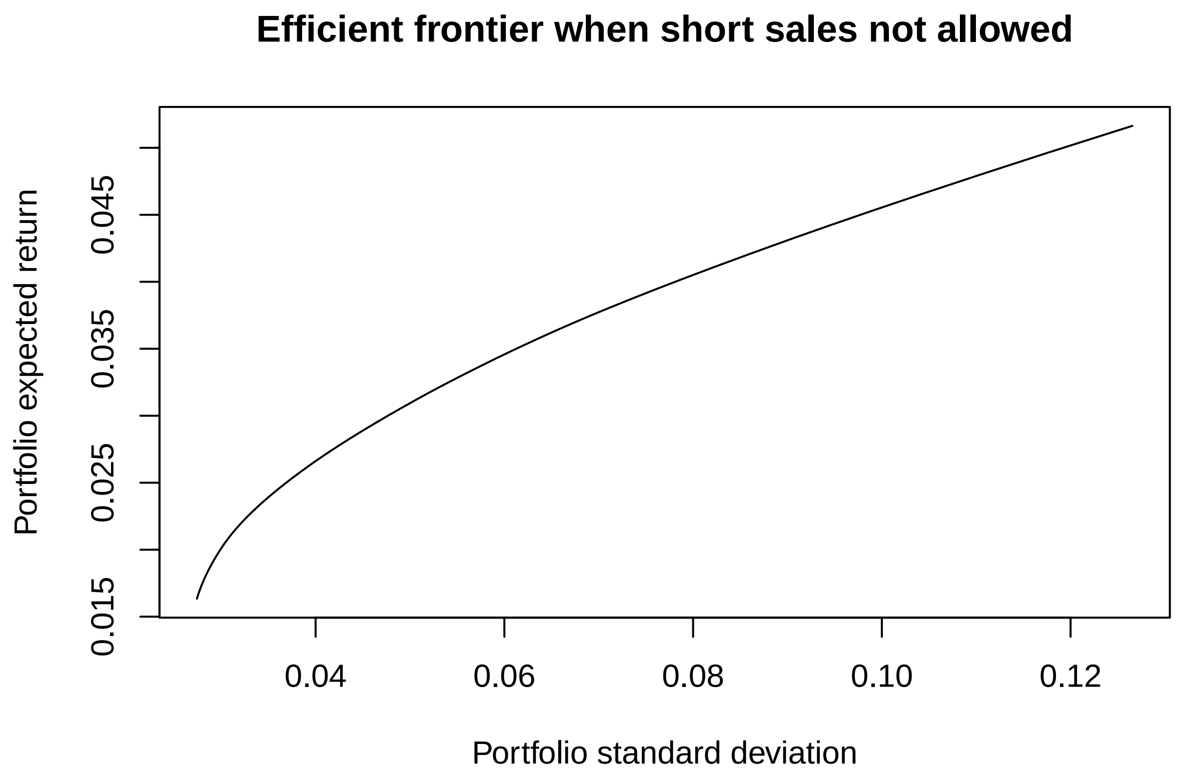


c. We want now to draw the efficient frontier when short sale are not allowed. One way to this is to use a for loop where you vary Rf . For each Rf you find the composition of the optimal portfolio (tangency point) and its expected return and standard deviation. Finally connect the points to draw the efficient frontier. Note: See handout #14 under "Labs".

```
Rfr <- seq(-0.05,.03,0.0005)

#Initialize the two vectors:
rbar_opt <- rep(0,length(Rfr))
risk_opt <- rep(0,length(Rfr))


for(l in 1:length(Rfr)){
#Risk free asset:
rf <- Rfr[l]
#rf <- .002
#Initialize
beta <- rep(0,ncol(rrr))
alpha <- rep(0,ncol(rrr))
mse <- rep(0,ncol(rrr))
Ribar <- rep(0,ncol(rrr))
Ratio <- rep(0,ncol(rrr))
stocknum <- rep(0,ncol(rrr))
#stock <- names(rrr)
```

```r
#This for loop computes the required inputs:
for(i in 1:ncol(rrr)){
    q <- lm(data=rrr, formula=rrr[,i] ~ r[,1])
    beta[i] <- q$coefficients[2]
    alpha[i] <- q$coefficients[1]
     mse[i] <- summary(q)$sigma^2
    Ribar[i] <- q$coefficients[1]+q$coefficients[2]*mean(r[,1])
    Ratio[i] <- (Ribar[i]-rf)/beta[i]
    stocknum[i] <- i
}

#So far we have this table:
#xx <- (cbind(stock,alpha, beta, Ribar, mse, Ratio))
xx <- (data.frame(stocknum,alpha, beta, Ribar, mse, Ratio))


#Order the table based on the excess return to beta ratio:
A <- xx[order(-xx[,6]),]

col1 <- rep(0,nrow(A))
col2 <- rep(0,nrow(A))
col3 <- rep(0,nrow(A))
col4 <- rep(0,nrow(A))
col5 <- rep(0,nrow(A))


#Create the last 5 columns of the table:
col1 <- (A[,4]-rf)*A[,3]/A[,5]
col3 <- A[,3]^2/A[,5]
for(i in(1:nrow(A))) {
col2[i] <- sum(col1[1:i])
col4[i] <- sum(col3[1:i])
}

#So far we have:
cbind(A, col1, col2, col3, col4)


#Compute the Ci (col5):
for(i in (1:nrow(A))) {
col5[i] <- var(r[,1])*col2[i]/(1+var(r[,1])*col4[i])
}


#The final table when short sales allowed:
B <- cbind(A, col1, col2, col3, col4, col5)
rownames(B) <- NULL

#SHORT SALES NOT ALLOWED:
#First create a matrix up to the maximum of col5:
#table1 <- cbind(A, col1, col2, col3, col4, col5)
#table2 <- (B[1:which(col5==max(col5)), ], nrow=which(col5==max(col5)), ncol=ncol(B))
table2 <- B[1:which(col5==max(col5)), ]
```

```
#Compute the Zi:
z_no_short <- (table2[,3]/table2[,5])*(table2[,6]-max(col5))

#Compute the xi:
x_no_short <- z_no_short/sum(z_no_short)

#Compute the mean and variance for each portfolio when short sales not allowed:
#First match the columns of the data with the composition of the portfolio:
r1 <- data.frame(rrr[,table2[,1]])

beta1 <- rep(0,ncol(r1))
sigma_e1 <- rep(0,ncol(r1))
alpha1 <- rep(0,ncol(r1))

for(i in 1:ncol(r1)){
    q1<- lm(r1[,i] ~ r[,1])
beta1[i] <- q1$coefficients[2]
sigma_e1[i] <- summary(q1)$sigma^2
alpha1[i] <- q1$coefficients[1]
}

means1 <- colMeans(r1)
#means1 <- alpha1 + beta1*mean(r[,1])


#Construct the variance covariance matrix using SIM:
xx <- rep(0,ncol(r1)*(ncol(r1)))             #Initialize
varcovar <- matrix(xx,nrow=ncol(r1),ncol=ncol(r1))  #the variance covariance matrix


for (i in 1:ncol(r1)){
    for (j in 1:ncol(r1)){
        varcovar[i,j]=beta1[i]*beta1[j]*var(r[,1])
        if(i==j){varcovar[i,j]=beta1[i]^2*var(r[,1])+ sigma_e1[i]}
        }
        }


rbar_opt[l] <- t(x_no_short) %*% means1
risk_opt[l] <- ( t(x_no_short) %*% varcovar %*% x_no_short )^.5

}

plot(risk_opt, rbar_opt, type="l", main="Efficient frontier when short sales not allowed", ylab="Portfo
```



d. Assume the constant correlation model holds. Rank the stocks based on the excess return to standard deviation ratio and complete the entire table based on handout #33: Note: Please use the same Rf as the one in (a) if possible.

```
#Compute the average correlation:
rho <- (sum(cor(r[2:31]))-30)/870 # 1st column is index so ignore, 30 diagonal elements, 870 other elem

#Initialize the vectors:
col1 <- rep(0,10)
col2 <- rep(0,10)
col3 <- rep(0,10)

#Initialize the var-covar matrix:
y <- rep(0,100)
mat <- matrix(y, ncol=30, nrow=30)

#Compute necessary quantities:
Rbar <- colMeans(r[2:31])
Rbar_f <- Rbar-0.0001
sigma <- ( diag(var(r[2:31])) )^0.5
Ratio <- Rbar_f/sigma

#Initial table:
xx <- (cbind(Rbar, Rbar_f, sigma, Ratio))
```

```r
#Order the table based on the excess return to sigma ratio:
aaa <- xx[order(-Ratio),]


#Create the last 3 columns of the table:
for(i in(1:10)) {

         col1[i] <- rho/(1-rho+i*rho)

         col2[i] <- sum(aaa[,4][1:i])
              }

#Compute the Ci:
for(i in (1:10)) {

          col3[i] <- col1[i]*col2[i]

               }

#Create the entire table until now:
xxx <- cbind(aaa, col1, col2, col3)

#SHORT SALES ALLOWED:
#Compute the Zi:
z <- (1/((1-rho)*xxx[,3]))*(xxx[,4]-xxx[,7][nrow(xxx)])

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

#The final table:
aaaa <- cbind(xxx, z, x)

head(aaaa)
```

```
##            Rbar     Rbar_f      sigma     Ratio      col1      col2      col3
## MA   0.02387403 0.02377403 0.04853263 0.4898565 0.2967132 0.4898565 0.1453469
## ADBE 0.02822558 0.02812558 0.05850966 0.4806999 0.2288194 0.9705564 0.2220822
## MSFT 0.02687025 0.02677025 0.05890516 0.4544637 0.1862108 1.4250201 0.2653541
## V    0.02004904 0.01994904 0.04436175 0.4496900 0.1569795 1.8747101 0.2942911
## NVDA 0.05163985 0.05153985 0.12579058 0.4097274 0.1356805 2.2844374 0.3099535
## ZTS  0.02138399 0.02128399 0.05432482 0.3917912 0.1194706 2.6762287 0.3197307
##              z           x
## MA   4.5602627 -0.10215356
## ADBE 3.5601272 -0.07974972
## MSFT 2.9029149 -0.06502763
## V    3.7015893 -0.08291859
## NVDA 0.8536916 -0.01912338
## ZTS  1.5072860 -0.03376442
```

```r
#SHORT SALES NOT ALLOWED:
#Find composition of optimum portfolio when short sales are not allowed:
aaaaa <- aaaa[1:which(aaaa[,7]==max(aaaa[,7])), ]
```

```
## Warning in 1:which(aaaa[, 7] == max(aaaa[, 7])): numerical expression has 3
## elements: only the first used
```

```
z_no <- (1/((1-rho)*aaaaa[,3]))*(aaaaa[,4]-aaaaa[,7][nrow(aaaaa)])
x_no <- z_no/sum(z_no)

#Final table:
a_no <- cbind(aaaaa, z_no, x_no)
head(a_no)
```

```
##            Rbar     Rbar_f      sigma     Ratio      col1      col2      col3
## MA   0.02387403 0.02377403 0.04853263 0.4898565 0.2967132 0.4898565 0.1453469
## ADBE 0.02822558 0.02812558 0.05850966 0.4806999 0.2288194 0.9705564 0.2220822
## MSFT 0.02687025 0.02677025 0.05890516 0.4544637 0.1862108 1.4250201 0.2653541
## V    0.02004904 0.01994904 0.04436175 0.4496900 0.1569795 1.8747101 0.2942911
## NVDA 0.05163985 0.05153985 0.12579058 0.4097274 0.1356805 2.2844374 0.3099535
## ZTS  0.02138399 0.02128399 0.05432482 0.3917912 0.1194706 2.6762287 0.3197307
##              z           x      z_no       x_no
## MA   4.5602627 -0.10215356 4.5602627 0.22154939
## ADBE 3.5601272 -0.07974972 3.5601272 0.17296022
## MSFT 2.9029149 -0.06502763 2.9029149 0.14103114
## V    3.7015893 -0.08291859 3.7015893 0.17983281
## NVDA 0.8536916 -0.01912338 0.8536916 0.04147455
## ZTS  1.5072860 -0.03376442 1.5072860 0.07322786
```

e. Find the composition of the point of tangency with and without short sales allowed. Place the two portfolios on the plot with the 30 stocks, S&P500, and the efficient frontier that you constructed in the previous projects.

Same as Part b?