Stats_C183_Project_5_Charles_Liu

Charles Liu (304804942)

5/15/2020

Loading Necessary Packages/Files:

```
library(readr)
# Loading the data for all stocks:
a <- read.csv("C:/Users/cliuk/Documents/UCLA Works/UCLA Spring 2020/Stats C183/Project/stockData_all.csv
```

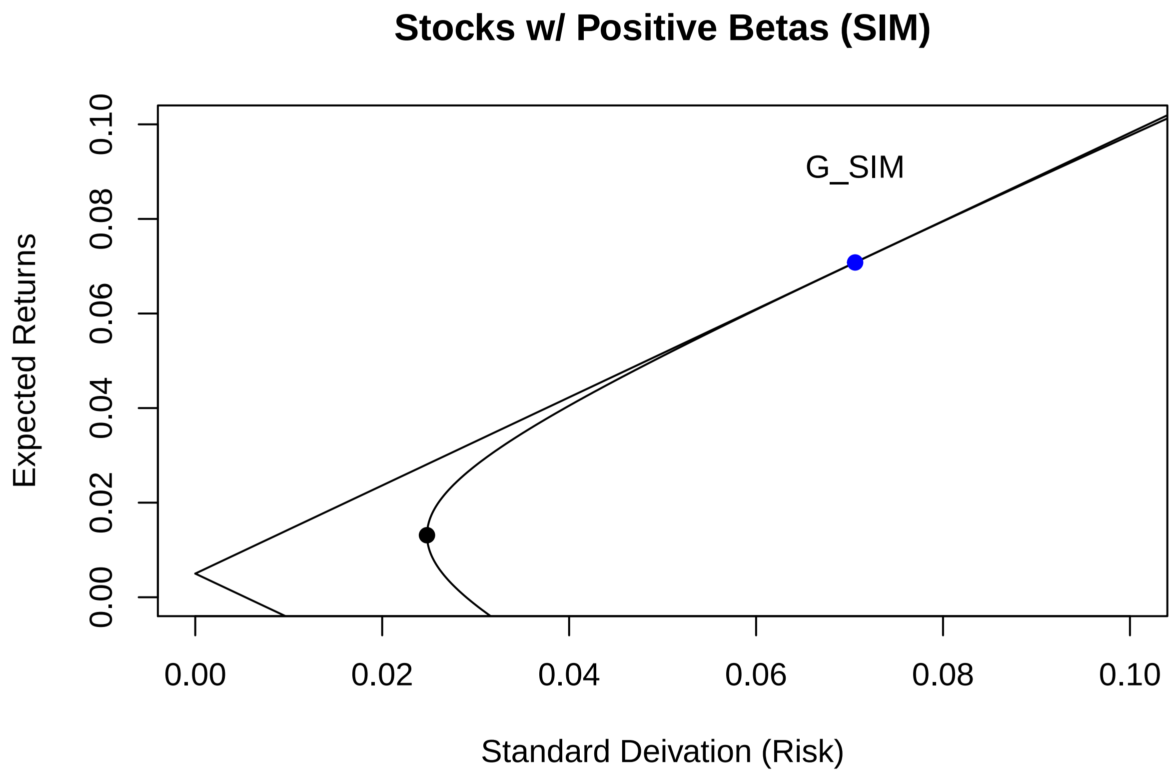
a)

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

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

```
##        BHP        GOLD        VALE        GOOG           T        NFLX
## 1.03167339 -0.14116913  0.96606144  0.99316900  0.63252260  1.15829301
##       AMZN         MCD        TSLA         WMT          KO        COST
## 1.20704704  0.60859600  1.00025367  0.25604736  0.66946453  0.73848356
##        XOM         CVX         TRP       BRK.B           V         JPM
## 1.22485080  1.17242167  0.84150937  0.86005529  0.95455176  1.41838539
##        JNJ        AMGN         CVS         UNP          BA          GE
## 0.71108254  0.82660026  0.81853689  0.91680193  1.53779231  1.33457006
##        DLR         BXP           O        MSFT        AAPL        NVDA
## 0.07297199  0.97358604  0.57217761  0.91629371  1.14510667  1.36654407
```

```
# For Beta, only GOLD is negative, so we exclude it in our SIM (Single Index Model)

###########################################################################
means_SIM <- colMeans(r[,-c(1,3)]) # 29 stocks instead of 31 (removes GOLD, S&P500)
covmat_SIM <- var(r[,-c(1,3)]) # Removed GOLD as its Beta is NOT positive.

# Set up A - E formulas & column of Ones:
ones_SIM <- rep(1, 29)
A_SIM <- t(ones_SIM) %*% solve(covmat_SIM) %*% means_SIM
B_SIM <- t(means_SIM) %*% solve(covmat_SIM) %*% means_SIM
C_SIM <- t(ones_SIM) %*% solve(covmat_SIM) %*% ones_SIM
D_SIM <- B_SIM * C_SIM - A_SIM^2
E_SIM <- seq(-5,5,.1)
```

```
# Set up the plot:
plot(sqrt(1/C_SIM), A_SIM/C_SIM,  xlim=c(0,0.1), ylim=c(0,0.1),pch=19, ylab = "Expected Returns", xlab =
minvar_SIM <- 1/C_SIM
minE_SIM <- A_SIM/C_SIM
sdeff_SIM <- seq((minvar_SIM)^0.5, 1, by = 0.0001)
options(warn = -1)
y1_SIM <- (A_SIM + sqrt(D_SIM*(C_SIM*sdeff_SIM^2 - 1)))*(1/C_SIM)
y2_SIM <- (A_SIM - sqrt(D_SIM*(C_SIM*sdeff_SIM^2 - 1)))*(1/C_SIM)
options(warn = 0)
points(sdeff_SIM, y1_SIM, type = "l")
points(sdeff_SIM, y2_SIM, type = "l")

# Choose risk-free return:
Rf <- 0.005

# Range of expected return:
sigma <- seq(0,.5, .001)
Rp1_SIM <- Rf + sigma*sqrt(C_SIM*Rf^2-2*Rf*A_SIM+B_SIM)
Rp2_SIM <- Rf - sigma*sqrt(C_SIM*Rf^2-2*Rf*A_SIM+B_SIM)
points(sigma, Rp1_SIM, type="l")
points(sigma, Rp2_SIM, type="l")

# Point of tangency:
Ri_SIM <- means_SIM-Rf
z_SIM <- solve(covmat_SIM) %*% Ri_SIM
x_G_SIM <- z_SIM/sum(z_SIM)
Ri_bar_SIM <- t(x_G_SIM) %*% means_SIM
varr_G_SIM <- t(x_G_SIM) %*% covmat_SIM %*% x_G_SIM
sdev_G_SIM <- varr_G_SIM^.5
points(sdev_G_SIM, Ri_bar_SIM, pch=19, col = "blue")
text(sdev_G_SIM, Ri_bar_SIM + 0.02, "G_SIM")
```




```
# Composition of Optimal Portfolio:
x_G_SIM
```

```
##               [,1]
## BHP   -0.33723772
## VALE  -0.03448230
## GOOG  -0.22340547
## T      0.25570333
## NFLX   0.02755105
## AMZN   0.31380130
## MCD    0.25978438
## TSLA   0.12209782
## WMT    0.15861048
## KO    -0.59357945
## COST  -0.32149018
## XOM   -1.81623157
## CVX    0.94854647
## TRP    0.15432028
## BRK.B -0.52345533
## V      0.82362571
## JPM    0.48436237
## JNJ    0.50293396
## AMGN  -0.21516512
## CVS   -0.10995796
## UNP    1.16474227
## BA     0.18991178
```

```
## GE    -0.12012387
## DLR    0.15905596
## BXP   -1.57521453
## O      0.98674874
## MSFT  -0.03961945
## AAPL   0.19703010
## NVDA   0.16113696
```

```
Ri_bar_SIM
```

```
##            [,1]
## [1,] 0.07079141
```

```
sdev_G_SIM
```

```
##            [,1]
## [1,] 0.07059549
```

b)

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

# 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 <- 0.005

# 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]
  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_ordered <- xx[order(-xx[,6]),]
col1 <- rep(0,nrow(A_ordered))
col2 <- rep(0,nrow(A_ordered))
col3 <- rep(0,nrow(A_ordered))
col4 <- rep(0,nrow(A_ordered))
col5 <- rep(0,nrow(A_ordered))
```

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

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

table1 <- cbind(A_ordered, col1, col2, col3, col4, col5)

# Express the entire table:
table1
```

```
##       stock         alpha       beta         Ribar          mse         Ratio
##  [1,]    24  0.0118107716 0.07297199  0.0123631195 0.0036757586  0.1009033728
##  [2,]     8  0.0340296617 1.00025367  0.0416008955 0.0260069622  0.0365916135
##  [3,]     5  0.0331898847 1.15829301  0.0419573679 0.0213044632  0.0319067522
##  [4,]    29  0.0258610181 1.36654407  0.0362048189 0.0091800225  0.0228348428
##  [5,]     6  0.0176942699 1.20704704  0.0268307877 0.0045011394  0.0180861118
##  [6,]     9  0.0076468285 0.25604736  0.0095849313 0.0024518651  0.0179065753
##  [7,]    16  0.0145218598 0.95455176  0.0217471616 0.0013347556  0.0175445295
##  [8,]    27  0.0139147478 0.91629371  0.0208504624 0.0022388395  0.0172984516
##  [9,]    11  0.0106703983 0.73848356  0.0162602120 0.0016088686  0.0152477491
## [10,]    19  0.0089435868 0.82660026  0.0152003835 0.0031739754  0.0123401649
## [11,]    28  0.0096519746 1.14510667  0.0183196462 0.0041941589  0.0116317952
## [12,]     3  0.0084803876 0.99316900  0.0159979953 0.0022926955  0.0110736394
## [13,]    21  0.0061436706 0.91680193  0.0130832320 0.0019932995  0.0088167703
## [14,]    26  0.0050196653 0.57217761  0.0093506571 0.0040877734  0.0076036829
## [15,]    18  0.0049620446 0.71108254  0.0103444515 0.0011132114  0.0075159369
## [16,]    17  0.0030707954 1.41838539  0.0138069995 0.0022783396  0.0062091724
## [17,]     7  0.0041343280 0.60859600  0.0087409821 0.0013987540  0.0061469055
## [18,]    22  0.0017475356 1.53779231  0.0133875680 0.0044395254  0.0054542918
## [19,]    15  0.0029891655 0.86005529  0.0094991938 0.0007351871  0.0052312844
## [20,]    20  0.0013003125 0.81853689  0.0074960750 0.0034275847  0.0030494350
## [21,]    14  0.0006211814 0.84150937  0.0069908298 0.0021959108  0.0023657845
## [22,]    10  0.0013410007 0.66946453  0.0064083877 0.0013278936  0.0021037525
## [23,]     4  0.0008445761 0.63252260  0.0056323382 0.0019083649  0.0009997084
## [24,]    25 -0.0049149844 0.97358604  0.0024543939 0.0021197381 -0.0026146699
## [25,]    13 -0.0074148364 1.17242167  0.0014595910 0.0019192874 -0.0030197403
## [26,]     1 -0.0084386036 1.03167339 -0.0006295440 0.0056509671 -0.0054567115
## [27,]    23 -0.0124704813 1.33457006 -0.0023687018 0.0044047254 -0.0055214050
## [28,]     2 -0.0077152214 0.96606144 -0.0004027993 0.0166931433 -0.0055926042
## [29,]    12 -0.0126171140 1.22485080 -0.0033458340 0.0015371372 -0.0068137557
##             col1      col2       col3       col4         col5
##  [1,] 0.1461743 0.1461743   1.448656   1.448656 0.0001823629
##  [2,] 1.4077069 1.5538812  38.470752  39.919408 0.0018497973
##  [3,] 2.0093189 3.5632001  62.974724 102.894132 0.0039459469
##  [4,] 4.6451695 8.2083696 203.424631 306.318763 0.0074188068
```

```
##  [5,]  5.8542483 14.0626178  323.687499   630.006262 0.0098332099
##  [6,]  0.4788027 14.5414205   26.738931   656.745193 0.0099813875
##  [7,] 11.9767488 26.5181693  682.648616  1339.393809 0.0123945509
##  [8,]  6.4871462 33.0053156  375.013117  1714.406926 0.0131259153
##  [9,]  5.1685274 38.1738430  338.969861  2053.376787 0.0133779712
## [10,]  2.6564918 40.8303348  215.271986  2268.648773 0.0133051695
## [11,]  3.6365851 44.4669199  312.641775  2581.290549 0.0131504505
## [12,]  4.7642035 49.2311234  430.229245  3011.519794 0.0129160350
## [13,]  3.7178170 52.9489404  421.675616  3433.195409 0.0125077109
## [14,]  0.6089742 53.5579146   80.089375  3513.284784 0.0124166548
## [15,]  3.4138586 56.9717732  454.215976  3967.500760 0.0119497574
## [16,]  5.4828171 62.4545903  883.018975  4850.519736 0.0110526810
## [17,]  1.6276963 64.0822866  264.799309  5115.319044 0.0108330779
## [18,]  2.9053416 66.9876282  532.670721  5647.989765 0.0103887420
## [19,]  5.2633615 72.2509897 1006.131786  6654.121551 0.0096926161
## [20,]  0.5960843 72.8470741  195.473695  6849.595246 0.0095228622
## [21,]  0.7629189 73.6099930  322.480323  7172.075569 0.0092333535
## [22,]  0.7100461 74.3200391  337.514060  7509.589629 0.0089437713
## [23,]  0.2095869 74.5296259  209.647982  7719.237611 0.0087482802
## [24,] -1.1691853 73.3604406  447.163630  8166.401241 0.0081816050
## [25,] -2.1627049 71.1977357  716.189031  8882.590272 0.0073530880
## [26,] -1.0277623 70.1699735  188.348289  9070.938561 0.0071086657
## [27,] -2.2326134 67.9373601  404.356026  9475.294587 0.0066116492
## [28,] -0.3126695 67.6246906   55.907667  9531.202254 0.0065456061
## [29,] -6.6502858 60.9744048  976.008846 10507.211100 0.0053924721
```

c)

```r
# SHORT SALES ALLOWED:
z_short <- (A_ordered[,3]/A_ordered[,5])*(A_ordered[,6]-col5[nrow(A_ordered)])
x_short <- z_short/sum(z_short)
x_short <- x_short[order(A_ordered[,1])]
names(x_short) <- names(a)[-c(1,2,3,5)]
Rbar_short <- t(x_short) %*% Ribar
sdev_short <- sqrt(t(x_short) %*% covmat_SIM %*% x_short)

#SHORT SALES NOT ALLOWED:
table1 <- cbind(A_ordered, col1, col2, col3, col4, col5)
table2 <- table1[1:which(col5==max(col5)), ]
z_no_short <- (table2[,3]/table2[,5])*(table2[,6]-max(col5))
x_no_short <- z_no_short/sum(z_no_short)
names(x_no_short) <- names(means_SIM)[c(24,8,5,29,6,9,16,27,11)]
Rbar_no_short <- t(x_no_short) %*% Ribar[c(24,8,5,29,6,9,16,27,11)]
sdev_no_short <- sqrt(t(x_no_short) %*% covmat_SIM[c(24,8,5,29,6,9,16,27,11), c(24,8,5,29,6,9,16,27,11)

# Composition of Point of Tangency (Short Sales Allowed):
x_short
```

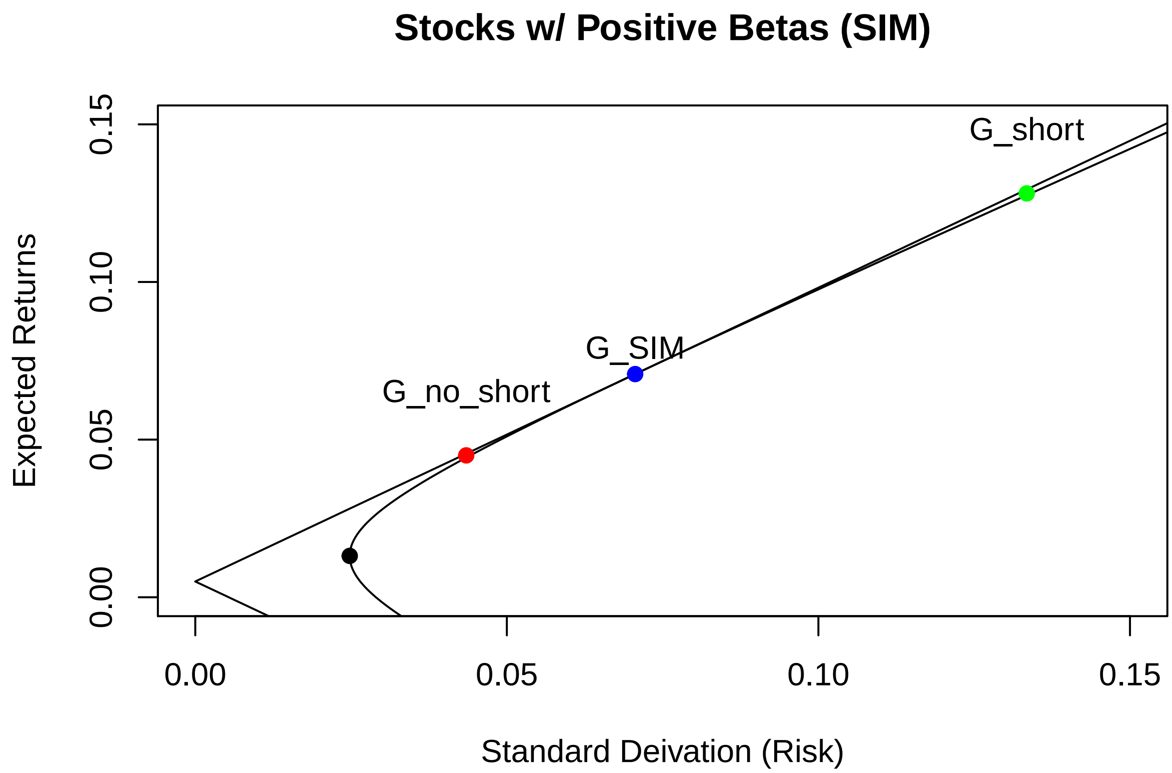
```
##          BHP         VALE         GOOG            T         NFLX         AMZN
## -0.188334201 -0.060448063  0.234006026 -0.138441138  0.137069353  0.323668611
##          MCD         TSLA          WMT           KO         COST          XOM
##  0.031211982  0.114097428  0.124261375 -0.157653576  0.430132025 -0.924835985
##          CVX          TRP        BRK.B            V          JPM          JNJ
```

```
## -0.488614940 -0.110287057 -0.017929700  0.826342765  0.048344961  0.128973398
##         AMGN          CVS          UNP           BA           GE          DLR
##  0.172046276 -0.053203709  0.149757036  0.002036109 -0.314422985  0.180291319
##          BXP            O         MSFT         AAPL         NVDA
## -0.349689307  0.029429771  0.463329090  0.161976207  0.246886927
```

```r
# Composition of Point of Tangency (Short Sales NOT Allowed):
x_no_short
```

```
##        DLR       TSLA       NFLX       NVDA       AMZN        WMT          V
## 0.14215012 0.07304117 0.08241355 0.11516787 0.10328924 0.03868940 0.24376936
##       MSFT       COST
## 0.13126682 0.07021246
```

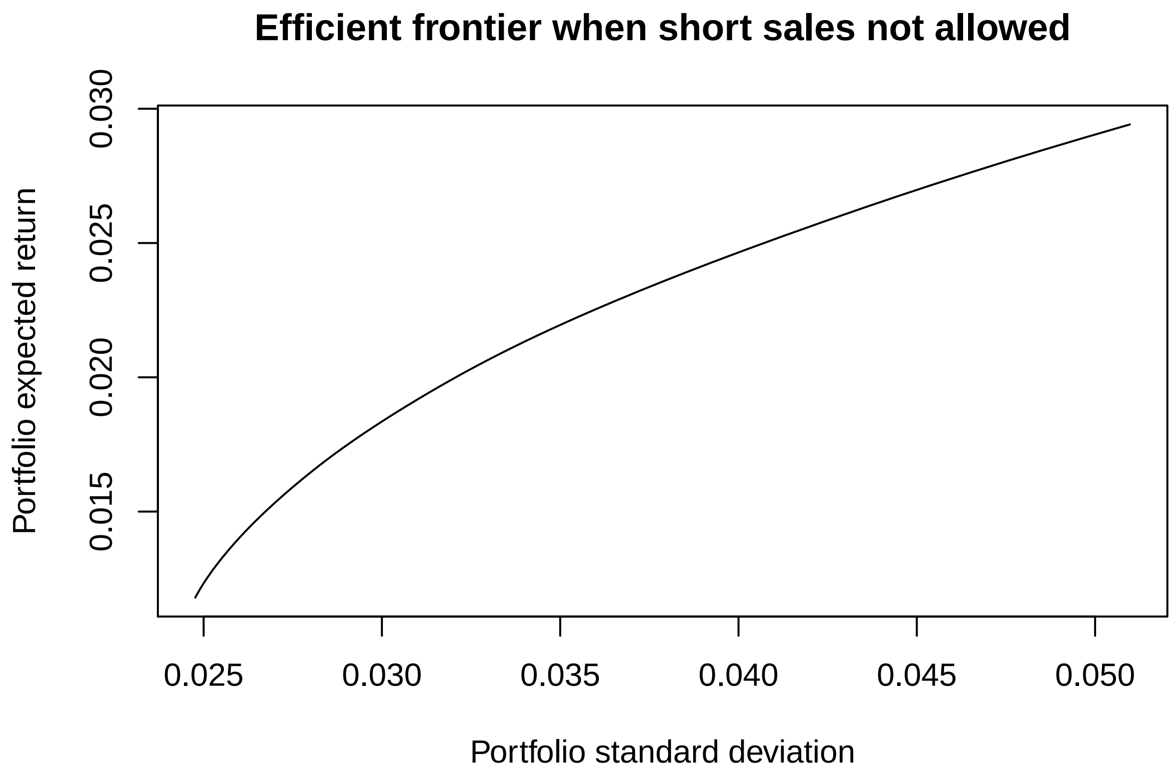
```r
# Previous Project Plot:
plot(sqrt(1/C_SIM), A_SIM/C_SIM,  xlim=c(0,0.15), ylim=c(0,0.15),pch=19, ylab = "Expected Returns", xlab
minvar_SIM <- 1/C_SIM
minE_SIM <- A_SIM/C_SIM
sdeff_SIM <- seq((minvar_SIM)^0.5, 1, by = 0.0001)
options(warn = -1)
Rbar_short <- 0.1281
Rbar_no_short <- 0.045
y1_SIM <- (A_SIM + sqrt(D_SIM*(C_SIM*sdeff_SIM^2 - 1)))*(1/C_SIM)
y2_SIM <- (A_SIM - sqrt(D_SIM*(C_SIM*sdeff_SIM^2 - 1)))*(1/C_SIM)
options(warn = 0)
points(sdeff_SIM, y1_SIM, type = "l")
points(sdeff_SIM, y2_SIM, type = "l")
points(sigma, Rp1_SIM, type="l")
points(sigma, Rp2_SIM, type="l")
Ri_SIM <- means_SIM-Rf
z_SIM <- solve(covmat_SIM) %*% Ri_SIM
x_G_SIM <- z_SIM/sum(z_SIM)
Ri_bar_SIM <- t(x_G_SIM) %*% means_SIM
varr_G_SIM <- t(x_G_SIM) %*% covmat_SIM %*% x_G_SIM
sdev_G_SIM <- varr_G_SIM^.5
points(sdev_G_SIM, Ri_bar_SIM, pch=19, col = "blue")
text(sdev_G_SIM, Ri_bar_SIM + 0.008, "G_SIM")

# Plot it with Short Sales' and No Short Sales' Tangency Point:
points(sdev_short, Rbar_short, pch=19, col = "green") # Short Sales Allowed
text(sdev_short, Rbar_short + 0.02, "G_short")
points(sdev_no_short, Rbar_no_short, pch=19, col = "red") # Short Sales NOT Allowed
text(sdev_no_short, Rbar_no_short + 0.02, "G_no_short")
```



d)

```
# Initialize for the for() loop:
Rfr <- seq(-0.05,.01,0.0005)
rbar_opt <- rep(0,length(Rfr))
risk_opt <- rep(0,length(Rfr))


# Start the for() loop:
for(l in 1:length(Rfr)){
  rf <- Rfr[l]
  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))
  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
  }
  xx <- (data.frame(stocknum,alpha, beta, Ribar, mse, 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))
  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])
  }
  cbind(A, col1, col2, col3, col4)
  for(i in (1:nrow(A))) {
    col5[i] <- var(r[,1])*col2[i]/(1+var(r[,1])*col4[i])
  }
  B <- cbind(A, col1, col2, col3, col4, col5)
  rownames(B) <- NULL
  table2 <- B[1:which(col5==max(col5)), ]
  z_no_short <- (table2[,3]/table2[,5])*(table2[,6]-max(col5))
  x_no_short <- z_no_short/sum(z_no_short)
  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)
  xx <- rep(0,ncol(r1)*(ncol(r1)))
  varcovar <- matrix(xx,nrow=ncol(r1),ncol=ncol(r1))
  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
}

# Create the Plot for Efficient Frontier When Short Sales NOT Allowed:
plot(risk_opt, rbar_opt, type="l", main="Efficient frontier when short sales not allowed", ylab="Portfo
```



e)

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

# Compute inputs:
stock_CCM <- 1:30
Ribar_CCM <- colMeans(r)
diff_CCM <- Ribar_CCM - Rf
covmat_CCM <- cov(r)
stdev_CCM <- diag(covmat)^.5
ratio_CCM <- diff_CCM / stdev_CCM
cormat_CCM <- cor(r)
avg_cor_CCM <- (sum(cormat_CCM) - 30) / (30*29)

# Table Construction
xx_CCM <- (cbind(stock_CCM, Ribar_CCM, diff_CCM, stdev_CCM, ratio_CCM))
A_CCM <- xx_CCM[order(-xx_CCM[,5]),]
col1 <- rep(0, nrow(A_CCM))
col2 <- rep(0, nrow(A_CCM))
col3 <- rep(0, nrow(A_CCM))
col1 <- avg_cor_CCM / (1 - avg_cor_CCM + stock_CCM*avg_cor_CCM)
for(i in(1:nrow(A_CCM))){
col2[i] <- sum(A_CCM[1:i, 5])
}
```

```r
col3 <- col1 * col2 # Ci
table3 <- cbind(A_CCM, col1, col2, col3)

# Express the Table:
table3
```

```
##         stock_CCM     Ribar_CCM      diff_CCM  stdev_CCM   ratio_CCM       col1
## V              18  0.0217471616  0.0167471616 0.04959639  0.33766894 0.26907667
## NVDA            1  0.0362048189  0.0312048189 0.10686145  0.29201193 0.21202554
## MSFT           29  0.0208504624  0.0158504624 0.05714113  0.27739146 0.17493488
## AMZN            8  0.0268307877  0.0218307877 0.07921927  0.27557422 0.14888900
## NFLX            7  0.0419573679  0.0369573679 0.15086966  0.24496222 0.12959389
## COST           13  0.0162602120  0.0112602120 0.04768530  0.23613594 0.11472609
## TSLA           10  0.0416008955  0.0366008955 0.16428425  0.22279005 0.10291864
## GOOG            5  0.0159979953  0.0109979953 0.05917663  0.18585032 0.09331481
## AAPL           30  0.0183196462  0.0133196462 0.07609063  0.17504975 0.08535036
## AMGN           21  0.0152003835  0.0102003835 0.06320778  0.16137861 0.07863853
## UNP            23  0.0130832320  0.0080832320 0.05498422  0.14701002 0.07290536
## JNJ            20  0.0103444515  0.0053444515 0.04163769  0.12835610 0.06795134
## JPM            19  0.0138069995  0.0088069995 0.06905999  0.12752680 0.06362775
## DLR            26  0.0123631195  0.0073631195 0.06036985  0.12196682 0.05982145
## BRK.B          17  0.0094991938  0.0044991938 0.04064605  0.11069203 0.05644483
## BA             24  0.0133875680  0.0083875680 0.08572843  0.09783882 0.05342904
## WMT            11  0.0095849313  0.0045849313 0.05008520  0.09154264 0.05071917
## MCD             9  0.0087409821  0.0037409821 0.04297973  0.08704061 0.04827091
## X.GSPC          1  0.0075693137  0.0025693137 0.03535294  0.07267609 0.04604812
## O              28  0.0093506571  0.0043506571 0.06674436  0.06518389 0.04402103
## CVS            22  0.0074960750  0.0024960750 0.06503567  0.03838009 0.04216489
## TRP            16  0.0069908298  0.0019908298 0.05530213  0.03599916 0.04045895
## KO             12  0.0064083877  0.0014083877 0.04329384  0.03253090 0.03888567
## T               6  0.0056323382  0.0006323382 0.04887463  0.01293796 0.03743017
## GOLD            3 -0.0001718661 -0.0051718661 0.13304506 -0.03887304 0.03607970
## VALE            4 -0.0004027993 -0.0054027993 0.13299430 -0.04062429 0.03482329
## BXP            27  0.0024543939 -0.0025456061 0.05729365 -0.04443086 0.03365144
## CVX            15  0.0014595910 -0.0035404090 0.06014552 -0.05886405 0.03255588
## BHP             2 -0.0006295440 -0.0056295440 0.08320437 -0.06765924 0.03152942
## GE             25 -0.0023687018 -0.0073687018 0.08115024 -0.09080321 0.03056570
## XOM            14 -0.0033458340 -0.0083458340 0.05827830 -0.14320654 0.26907667
##              col2       col3
## V       0.3376689 0.09085883
## NVDA    0.6296809 0.13350843
## MSFT    0.9070723 0.15867859
## AMZN    1.1826465 0.17608306
## NFLX    1.4276088 0.18500938
## COST    1.6637447 0.19087493
## TSLA    1.8865348 0.19415959
## GOOG    2.0723851 0.19338421
## AAPL    2.2474348 0.19181936
## AMGN    2.4088134 0.18942554
## UNP     2.5558235 0.18633323
## JNJ     2.6841796 0.18239360
## JPM     2.8117064 0.17890255
## DLR     2.9336732 0.17549657
```

```
## BRK.B  3.0443652 0.17183869
## BA     3.1422040 0.16788495
## WMT    3.2337467 0.16401293
## MCD    3.3207873 0.16029741
## X.GSPC 3.3934634 0.15626261
## O      3.4586473 0.15225323
## CVS    3.4970273 0.14745178
## TRP    3.5330265 0.14294253
## KO     3.5655574 0.13864910
## T      3.5784954 0.13394371
## GOLD   3.5396223 0.12770853
## VALE   3.4989980 0.12184663
## BXP    3.4545672 0.11625115
## CVX    3.3957031 0.11055012
## BHP    3.3280439 0.10493128
## GE     3.2372407 0.09894852
## XOM    3.0940341 0.83253240
```

f)

```
# SHORT SALES ALLOWED:
z_short_CCM <- (A_CCM[,5] - col3[nrow(A_CCM)]) / ((1 - avg_cor_CCM)*A_CCM[,4])
x_short_CCM <- z_short_CCM/sum(z_short_CCM)
names(x_short_CCM) <- names(a)[-c(1,2)]


# SHORT SALES NOT ALLOWED:
table3 <- cbind(A_CCM, col1, col2, col3)
table4 <- table3[1:which(col3 == max(col3)), ]
z_no_short_CCM <- (table4[,5] - col3[nrow(table4)]) / ((1 - avg_cor_CCM)*table4[,4])
x_no_short_CCM <- z_no_short_CCM/sum(z_no_short_CCM)
x_no_short_CCM <- x_no_short_CCM[order(A_CCM[,1])]
x_no_short_CCM <- ifelse(is.na(x_no_short_CCM), 0, x_no_short_CCM)
names(x_no_short_CCM) <- names(a)[-c(1,2)]


# Composition of short sales and no short sales:
x_short_CCM
```

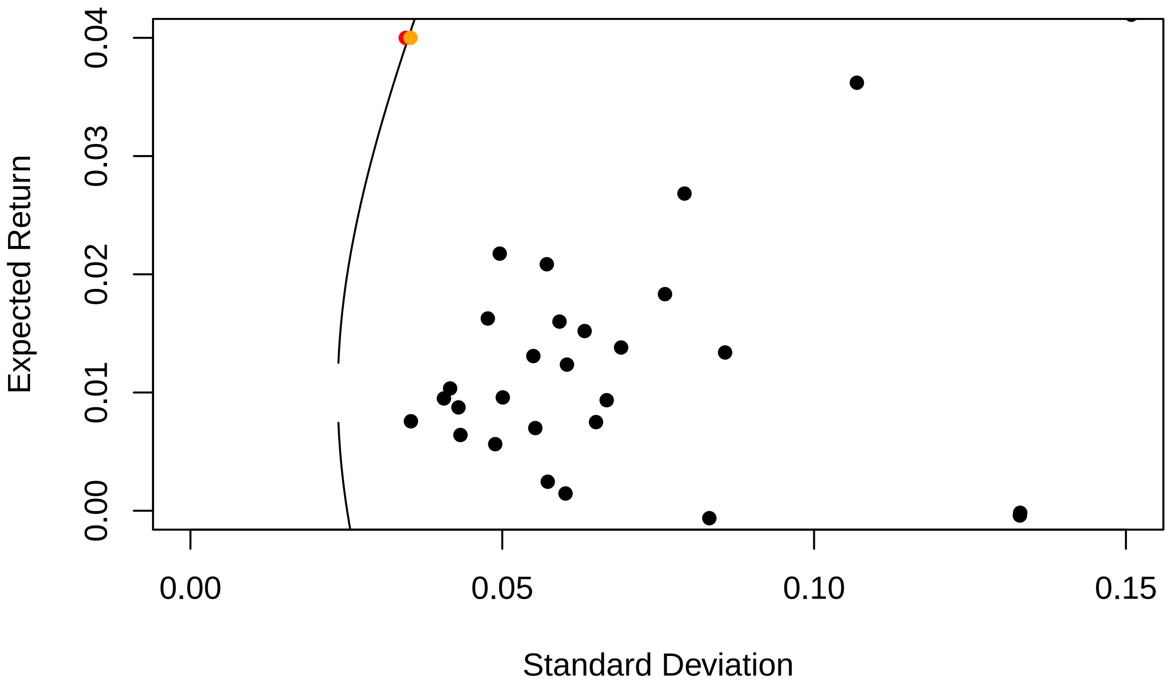
```
##     X.GSPC        BHP       GOLD       VALE       GOOG          T       NFLX
## 0.02704427 0.01370980 0.02633264 0.01905600 0.01055596 0.03389929 0.01005983
##       AMZN        MCD       TSLA        WMT         KO       COST        XOM
## 0.02961970 0.02342034 0.02878005 0.03379275 0.04583896 0.02766980 0.03190245
##        CVX        TRP      BRK.B          V        JPM        JNJ       AMGN
## 0.04813521 0.02322851 0.04009986 0.04701313 0.05825671 0.03116148 0.03309727
##        CVS        UNP         BA         GE        DLR        BXP          O
## 0.03903931 0.05008466 0.04545227 0.01775258 0.01779505 0.04148728 0.04017054
##       MSFT       AAPL       NVDA
## 0.02932438 0.03083968 0.04538024
```

```
x_no_short_CCM
```

```
##     X.GSPC        BHP       GOLD       VALE       GOOG          T       NFLX
## 0.01370980 0.05825671 0.02932438 0.01775258 0.01779505 0.02961970 0.04545227
```

```
##       AMZN        MCD       TSLA        WMT         KO       COST        XOM
## 0.01055596 0.01905600 0.04701313 0.01005983 0.04009986 0.05008466 0.03389929
##        CVX        TRP      BRK.B          V        JPM        JNJ       AMGN
## 0.04538024 0.04017054 0.03903931 0.04813521 0.02704427 0.02766980 0.04583896
##        CVS        UNP         BA         GE        DLR        BXP          O
## 0.02878005 0.03309727 0.03379275 0.02322851 0.03083968 0.03190245 0.04148728
##       MSFT       AAPL       NVDA
## 0.03116148 0.02633264 0.02342034
```

```
# Plot it all:
mean_short <- t(x_short_CCM) %*% means
sd_short <- sqrt(t(x_short_CCM) %*% covmat %*% x_short_CCM)
mean_no_short <- t(x_no_short_CCM) %*% means
sd_no_short <- sqrt(t(x_no_short_CCM) %*% covmat %*% x_no_short_CCM)
ones <- matrix(1, nrow = 31, ncol = 1)

A2 <- t(ones) %*% solve(covmat) %*% means
B <- t(means) %*% solve(covmat) %*% means
C <- t(ones) %*% solve(covmat) %*% ones
D <- B*C - A2^2
mean_short <- 0.04
mean_no_short <- 0.04
plot(stdev_CCM, Ribar_CCM, xlab = "Standard Deviation", ylab = "Expected Return", pch = 16,
xlim = c(0,0.15), ylim = c(0, 0.04))
# Efficient frontier:
minvar <- 1/C
minE <- A2/C
sdeff <- seq((minvar)^0.5, 1, by = 0.0001)
options(warn = -1)
y1 <- (A2 + sqrt(D*(C*sdeff^2 - 1)))*(1/C)
y2 <- (A2 - sqrt(D*(C*sdeff^2 - 1)))*(1/C)
options(warn = 0)
points(sdeff, y1, type = "l")
points(sdeff, y2, type = "l")
points(sd_short, mean_short, pch = 16, col = "red")
points(sd_no_short, mean_no_short, pch = 16, col = "orange")
```


0.04
0.03
0.02
0.01
0.00
Expected Return
0.00
0.05
0.10
0.15
Standard Deviation