Stats C183-Project6

Philip Huang

Spring 2022

PART A Downloading the training data

```
train <- read.csv("stockDataTrain.csv", sep=",", header=TRUE)
```

Computing the returns

```
returns <- (train[-1,3:ncol(train)]-train[-nrow(train),3:ncol(train)])/train[-nrow(train),3:ncol(train)]
```

Separating the returns with the index

```
rr <- returns[,-1]
index <- returns[,1]
head(rr) #gives the returns without the index
```

```
##            TSM       AAPL        NVDA          TSLA        MSFT         AMD
## 2  0.067966914 0.05121871  0.17070078  0.3494846108  0.01242055  0.08163265
## 3  0.107913530 0.02605846 -0.02117160 -0.1485233998  0.07797878  0.08086253
## 4  0.003996158 0.09939605  0.03126701 -0.0026864956 -0.01439350  0.01995012
## 5  0.022885268 0.07271754  0.02869540 -0.0005771803  0.01336632 -0.02200489
## 6  0.040369884 0.03340213 -0.01971600  0.1554122309  0.02574914  0.04750000
## 7 -0.064983478 0.02873098 -0.05609494 -0.0698158987  0.03501199 -0.06682578
##           GOOG           FB        NFLX       TWTR          CHT        CMCSA
## 2  0.029365727  0.094134553  0.08868150 -0.1486822  0.028911726 -0.050688595
## 3 -0.082750882 -0.120070072 -0.21003972 -0.1500638  0.014214828 -0.031921145
## 4 -0.054419409 -0.007636172 -0.08519163 -0.1649882  0.023142138  0.039089837
## 5  0.063095775  0.058882570  0.29744751 -0.1675648  0.014654464  0.008500881
## 6  0.027487499  0.063033208  0.05449585  0.2629470  0.006593393  0.028352277
## 7 -0.006396881  0.079655237 -0.04058098  0.1030021 -0.056144736  0.005109618
##            PLD         AMT          CCI          PSA        EQIX        WELL
## 2  0.062693927 0.007294700  0.069616931  0.072403346  0.02570208 0.014157854
## 3 -0.008740028 0.004909937 -0.027931708 -0.003017811 -0.02695303 0.029014868
## 4  0.003202099 0.020153397 -0.009602118  0.050427962  0.01606782 0.058557368
## 5  0.021658586 0.077430553  0.054997968 -0.017834140  0.05825051 0.002218882
## 6 -0.010118024 0.003905081 -0.032190846 -0.005975265  0.05705641 0.003683868
## 7  0.001015610 0.053068066  0.003718001  0.009755617  0.02108652 0.015318362
##            JNJ          UNH           MRK          ABT         ABBV         AZN
## 2  0.041256890  0.069036430  0.0758923301  0.091173470  0.042316607  0.06708666
## 3  0.073999511  0.061085122 -0.0038603275 -0.031925690  0.009624971 -0.01504280
## 4  0.031151227 -0.081459836  0.0399720579  0.005972766  0.013229588  0.21840318
## 5  0.001678525  0.061167487 -0.0119534510  0.038916678  0.052538949 -0.08665395
```

```
## 6  0.038331612  0.026623209 -0.0001728946  0.022244611  0.038836598  0.02922433
## 7 -0.043299888 -0.003903819 -0.0114363785  0.029828673 -0.072643602 -0.02045492
##             WMT          PG          PEP          MDLZ           KO
## 2  0.0002677824  0.03438440 -0.003608634  0.0387670709  0.010047332
## 3  0.0231594815  0.02466324  0.042837588  0.0152805604  0.012041789
## 4  0.0496429779  0.02419383  0.035893769  0.0360825217  0.063454976
## 5 -0.0368834760 -0.01350660  0.028408101  0.0552597191  0.002942074
## 6 -0.0160899923 -0.02723131  0.011434796 -0.0002659565  0.035443670
## 7 -0.0198480014 -0.01615955 -0.006503135 -0.0386491729 -0.065498296
##             EL
## 2  0.001454896
## 3 -0.025584005
## 4  0.085077885
## 5  0.055808312
## 6 -0.028230138
## 7 -0.010772920
```

```
index #gives the returns of the index
```

```
##  [1]  0.0431170300  0.0069321656  0.0062007890  0.0210302800  0.0190583317
##  [6] -0.0150798306  0.0376552955 -0.0155138372  0.0232014608  0.0245335888
## [11] -0.0041885879 -0.0310408058  0.0548925110 -0.0173961069  0.0085208197
## [16]  0.0104913824 -0.0210116724  0.0197420297 -0.0625808182 -0.0264428316
## [21]  0.0829831178  0.0005048693 -0.0175301852 -0.0507353220 -0.0041283604
## [26]  0.0659911146  0.0026993985  0.0153246024  0.0009109211  0.0356098011
## [31] -0.0012192431 -0.0012344508 -0.0194256793  0.0341745222  0.0182007622
## [36]  0.0178843582  0.0371981603 -0.0003891972  0.0090912085  0.0115762514
## [41]  0.0048137751  0.0193488261  0.0005464328  0.0193029785  0.0221881353
## [46]  0.0280826277  0.0098316305  0.0561787044 -0.0389473721 -0.0268844986
## [51]  0.0027187751  0.0216083420  0.0048424360  0.0360215562  0.0302632115
## [56]  0.0042942871 -0.0694033560  0.0178593568 -0.0917768946
```

BREAKPOINT 1: This is the first approach for finding X for the Multigroup Model

```
group1 = rr[1:6]
group2 = rr[7:12]
group3 = rr[13:18]
group4 = rr[19:24]
group5 = rr[25:30]

cor.matrix = cor(rr)
head(cor.matrix)
```

```
##              TSM      AAPL       NVDA        TSLA      MSFT        AMD
## TSM  1.000000000 0.4881521 0.51335805 0.002352407 0.5610358 0.28976420
## AAPL 0.488152139 1.0000000 0.46607525 0.132955720 0.3883190 0.25508531
## NVDA 0.513358050 0.4660753 1.00000000 0.091198792 0.4424173 0.41682653
## TSLA 0.002352407 0.1329557 0.09119879 1.000000000 0.1000357 0.05088927
## MSFT 0.561035768 0.3883190 0.44241733 0.100035732 1.0000000 0.31061651
## AMD  0.289764204 0.2550853 0.41682653 0.050889269 0.3106165 1.00000000
##            GOOG         FB      NFLX        TWTR         CHT     CMCSA
## TSM  0.41812563 0.09204639 0.2320642 -0.04916528  0.29111913 0.2885973
## AAPL 0.29738400 0.32209593 0.2715191  0.25514918  0.21085202 0.2562311
```

```
## NVDA 0.40521256 0.20242695 0.3525203  0.09768510  0.14273033 0.2786302
## TSLA 0.08753646 0.22938963 0.2598298  0.20023724  0.07437638 0.1848653
## MSFT 0.58646897 0.30209522 0.3194192 -0.01034658 -0.04565052 0.3746506
## AMD  0.29352490 0.12997820 0.1645240 -0.06180246  0.08940778 0.1971629
##            PLD        AMT        CCI         PSA       EQIX       WELL
## TSM  0.2988845 0.33037283 0.29291604  0.04869060 0.34303613  0.1365636
## AAPL 0.2801479 0.38915377 0.31429053  0.22728323 0.15500742  0.1184767
## NVDA 0.2763530 0.02292159 0.15452519  0.02401534 0.26038486 -0.1047239
## TSLA 0.1021972 0.16292178 0.08453724  0.07435971 0.12163911  0.1855199
## MSFT 0.2613861 0.35657676 0.16885831  0.04307035 0.42383415 -0.1077074
## AMD  0.3591972 0.17342958 0.12253440 -0.04786822 0.09538971  0.1876068
##             JNJ       UNH        MRK       ABT       ABBV       AZN        WMT
## TSM  0.4138310 0.2149548 0.16602888 0.4699059 0.25686311 0.2791651 0.27269290
## AAPL 0.1814671 0.1984693 0.12514394 0.4012236 0.08962962 0.1754471 0.14238957
## NVDA 0.1024046 0.2166770 0.06681852 0.2746761 0.25477425 0.1901820 0.10781126
## TSLA 0.0844957 0.1872566 0.19167680 0.2124842 0.20815838 0.1398365 0.06704248
## MSFT 0.3621385 0.1537371 0.19315174 0.4152522 0.38224601 0.1435684 0.09182490
## AMD  0.3030626 0.3335709 0.34979807 0.4516329 0.38781251 0.1379067 0.03907121
##               PG         PEP      MDLZ        KO         EL
## TSM   0.22332488 0.331596563 0.2746476 0.4295469 0.25282221
## AAPL  0.14023665 0.220589041 0.1558214 0.1710399 0.23472668
## NVDA -0.07196894 0.009185657 0.1425258 0.1295878 0.13086183
## TSLA  0.08356726 0.119718469 0.1349797 0.2218584 0.11535336
## MSFT  0.19265238 0.348626415 0.3954915 0.4343138 0.19395054
## AMD   0.16057256 0.222287672 0.1800234 0.1311243 0.08478151
```

```
means = colMeans(rr)
means
```

```
##          TSM         AAPL         NVDA         TSLA         MSFT          AMD
##  0.018022831  0.017597560  0.044567996  0.016879280  0.020687995  0.043437846
##         GOOG           FB         NFLX         TWTR          CHT        CMCSA
##  0.011206214  0.014473364  0.034101355 -0.003696091  0.007410518  0.006931851
##          PLD          AMT          CCI          PSA         EQIX         WELL
##  0.011089250  0.014264652  0.011227455  0.008176040  0.015260709  0.009094911
##          JNJ          UNH          MRK          ABT         ABBV          AZN
##  0.009525905  0.023671961  0.010107381  0.015006148  0.016591625  0.008827518
##          WMT           PG          PEP         MDLZ           KO           EL
##  0.007404403  0.006624564  0.008565794  0.006095891  0.007265480  0.013236073
```

```
library(matrixStats)
sd = colSds(as.matrix(rr[sapply(rr,is.numeric)]))
```

Compute correlation fro each group

```
rho11 = (sum(cor.matrix[1:6,1:6]) - 6) / (36-6)
rho12 = (sum(cor.matrix[1:6,7:12])) / 36
rho13 = (sum(cor.matrix[1:6,13:18])) / 36
rho14 = (sum(cor.matrix[1:6,19:24])) / 36
rho15 = (sum(cor.matrix[1:6,25:30])) / 36

rho21 = rho12
rho22 = (sum(cor.matrix[7:12,7:12])-6) / (36-6)
```

```
rho23 = (sum(cor.matrix[7:12,13:18])) / 36
rho24 = (sum(cor.matrix[7:12,19:24])) / 36
rho25 = (sum(cor.matrix[7:12,25:30])) / 36

rho31 = rho13
rho32 = rho23
rho33 = (sum(cor.matrix[13:18,13:18])-6) / (36-6)
rho34 = (sum(cor.matrix[13:18,19:24])) / 36
rho35 = (sum(cor.matrix[13:18,25:30])) / 36

rho41 = rho14
rho42 = rho24
rho43 = rho34
rho44 = (sum(cor.matrix[19:24,19:24])-6) / (36-6)
rho45 = (sum(cor.matrix[19:24,25:30])) / 36

rho51 = rho15
rho52 = rho25
rho53 = rho35
rho54 = rho45
rho55 = (sum(cor.matrix[25:30,25:30])-6) / (36-6)

P = rbind(c(rho11,rho12,rho13,rho14,rho15),
          c(rho21,rho22,rho23,rho24,rho25),
          c(rho31,rho32,rho33,rho34,rho35),
          c(rho41,rho42,rho43,rho44,rho45),
          c(rho51,rho52,rho53,rho54,rho55))
P
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.3006055 0.2150248 0.1759939 0.2420958 0.1809633
## [2,] 0.2150248 0.1624972 0.1367383 0.1219426 0.1329181
## [3,] 0.1759939 0.1367383 0.3889642 0.2418283 0.3176168
## [4,] 0.2420958 0.1219426 0.2418283 0.2942266 0.2577317
## [5,] 0.1809633 0.1329181 0.3176168 0.2577317 0.3697981
```

Computing matrix A

```
N = 6 #6 stocks per group
row1 = c(1+(N*rho11)/(1-rho11),N*rho12/(1-rho11),N*rho13/(1-rho11),N*rho14/(1-rho11),N*rho15/(1-rho11))

row2 = c((N*rho21)/(1-rho22),1+N*rho22/(1-rho22),N*rho23/(1-rho22),N*rho24/(1-rho22),N*rho25/(1-rho22))

row3 = c((N*rho31)/(1-rho33),N*rho32/(1-rho33),1+N*rho33/(1-rho33),N*rho34/(1-rho33),N*rho35/(1-rho33))

row4 = c((N*rho41)/(1-rho44),N*rho42/(1-rho44),N*rho43/(1-rho44),1+N*rho44/(1-rho44),N*rho45/(1-rho44))

row5 = c((N*rho51)/(1-rho55),N*rho52/(1-rho55),N*rho53/(1-rho55),N*rho54/(1-rho55),1+N*rho55/(1-rho55))
A = rbind(row1,row2,row3,row4,row5)
A
```

```
##          [,1]     [,2]      [,3]      [,4]      [,5]
## row1 3.578849 1.844665 1.5098255 2.0769030 1.5524565
```

```
## row2 1.540471 2.164155 0.9796142 0.8736154 0.9522458
## row3 1.728153 1.342687 4.8193915 2.3746066 3.1188037
## row4 2.058132 1.036672 2.0558578 3.5013126 2.1910574
## row5 1.722908 1.265481 3.0239529 2.4538009 4.5207586
```

Computing Matrix C

```
Rf = 0.002
c1 = 0
for (i in 1:6){
  c1 = c1 + as.numeric((means[i]-Rf)/(sd[i]*(1-rho11)))
}
c2 = 0
for (i in 7:12){
  c2 = c2 + as.numeric((means[i]-Rf)/(sd[i]*(1-rho22)))
}
c3 = 0
for (i in 13:18){
  c3 = c3 + as.numeric((means[i]-Rf)/(sd[i]*(1-rho33)))
}
c4 = 0
for (i in 19:24){
  c4 = c4 + as.numeric((means[i]-Rf)/(sd[i]*(1-rho44)))
}
c5 = 0
for (i in 25:30){
  c5 = c5 + as.numeric((means[i]-Rf)/(sd[i]*(1-rho55)))
}
C <- rbind(c1,c2,c3,c4,c5)
C
```

```
##         [,1]
## c1 2.1441207
## c2 0.9747475
## c3 1.8117479
## c4 1.8850972
## c5 1.2966100
```

cut off points for each group from phi

```
phi <- solve(A) %*% C
phi
```

```
##             [,1]
## [1,]  0.41352603
## [2,]  0.02910834
## [3,]  0.18078681
## [4,]  0.27329112
## [5,] -0.14820211
```

```
c1s = c(rho11,rho12,rho13,rho14,rho15) %*% phi
c2s = c(rho21,rho22,rho23,rho24,rho25) %*% phi
```

```
c3s = c(rho31,rho32,rho33,rho34,rho35) %*% phi
c4s = c(rho41,rho42,rho43,rho44,rho45) %*% phi
c5s = c(rho51,rho52,rho53,rho54,rho55) %*% phi
```

```
z1 <- numeric(0)
for(i in 1:6){
  z1[i] = 1/(sd[i]*(1-rho11)) * ((means[i]-Rf)/sd[i] - c1s)
}

z2 <- numeric(0)
for(i in 7:12){
  z2[i-6] = 1/(sd[i]*(1-rho22)) * ((means[i]-Rf)/sd[i] - c2s)
}

z3 <- numeric(0)
for(i in 13:18){
  z3[i-12] = 1/(sd[i]*(1-rho33)) * ((means[i]-Rf)/sd[i] - c3s)
}

z4 <- numeric(0)
for(i in 19:24){
  z4[i-18] = 1/(sd[i]*(1-rho44)) * ((means[i]-Rf)/sd[i] - c4s)
}

z5 <- numeric(0)
for(i in 25:30){
  z5[i-24] = 1/(sd[i]*(1-rho55)) * ((means[i]-Rf)/sd[i] - c5s)
}

Z = c(z1,z2,z3,z4,z5)
Z
```

```
##  [1]  1.11407657  0.30605812  1.83426225 -0.91364475  2.57355537  0.30604410
##  [7]  0.57044088  1.36838082  1.02047402 -1.45348036  1.24172661 -0.97367581
## [13]  0.23049866  2.36731389  1.26184626 -1.32748109  2.37771834 -1.26134559
## [19]  0.02709128  7.72486088 -0.76403279  1.08979086  0.02397025 -1.85258929
## [25] -1.52256586 -1.43707250  0.43742731 -2.21948757 -0.51298672  2.04883197
```

```
X = Z/sum(Z)
X
```

```
##  [1]  0.081402607  0.022362851  0.134024656 -0.066757588  0.188042833
##  [6]  0.022361827  0.041680595  0.099983940  0.074563317 -0.106201937
## [11]  0.090729655 -0.071143897  0.016841923  0.172973319  0.092199744
## [16] -0.096995507  0.173733544 -0.092163161  0.001979488  0.564435002
## [21] -0.055825840  0.079628114  0.001751442 -0.135363764 -0.111249830
## [26] -0.105003059  0.031961648 -0.162172043 -0.037482573  0.149702693
```

BREAKPOINT 2 This is the Second approach to finding X for multigroup model Computing the variance of the market

```r
exp_R_m = mean(index)
sigma_m2 <- var(index)
```

```r
exp_R = colMeans(rr)
```

Optimal Portfolio from Multigroup Model

```r
cormat <- cor(rr)
head(cormat)
```

```
##              TSM      AAPL       NVDA        TSLA      MSFT        AMD
## TSM  1.000000000 0.4881521 0.51335805 0.002352407 0.5610358 0.28976420
## AAPL 0.488152139 1.0000000 0.46607525 0.132955720 0.3883190 0.25508531
## NVDA 0.513358050 0.4660753 1.00000000 0.091198792 0.4424173 0.41682653
## TSLA 0.002352407 0.1329557 0.09119879 1.000000000 0.1000357 0.05088927
## MSFT 0.561035768 0.3883190 0.44241733 0.100035732 1.0000000 0.31061651
## AMD  0.289764204 0.2550853 0.41682653 0.050889269 0.3106165 1.00000000
##            GOOG         FB      NFLX        TWTR         CHT     CMCSA
## TSM  0.41812563 0.09204639 0.2320642 -0.04916528  0.29111913 0.2885973
## AAPL 0.29738400 0.32209593 0.2715191  0.25514918  0.21085202 0.2562311
## NVDA 0.40521256 0.20242695 0.3525203  0.09768510  0.14273033 0.2786302
## TSLA 0.08753646 0.22938963 0.2598298  0.20023724  0.07437638 0.1848653
## MSFT 0.58646897 0.30209522 0.3194192 -0.01034658 -0.04565052 0.3746506
## AMD  0.29352490 0.12997820 0.1645240 -0.06180246  0.08940778 0.1971629
##            PLD        AMT        CCI         PSA       EQIX       WELL
## TSM  0.2988845 0.33037283 0.29291604  0.04869060 0.34303613  0.1365636
## AAPL 0.2801479 0.38915377 0.31429053  0.22728323 0.15500742  0.1184767
## NVDA 0.2763530 0.02292159 0.15452519  0.02401534 0.26038486 -0.1047239
## TSLA 0.1021972 0.16292178 0.08453724  0.07435971 0.12163911  0.1855199
## MSFT 0.2613861 0.35657676 0.16885831  0.04307035 0.42383415 -0.1077074
## AMD  0.3591972 0.17342958 0.12253440 -0.04786822 0.09538971  0.1876068
##            JNJ       UNH        MRK       ABT       ABBV       AZN        WMT
## TSM  0.4138310 0.2149548 0.16602888 0.4699059 0.25686311 0.2791651 0.27269290
## AAPL 0.1814671 0.1984693 0.12514394 0.4012236 0.08962962 0.1754471 0.14238957
## NVDA 0.1024046 0.2166770 0.06681852 0.2746761 0.25477425 0.1901820 0.10781126
## TSLA 0.0844957 0.1872566 0.19167680 0.2124842 0.20815838 0.1398365 0.06704248
## MSFT 0.3621385 0.1537371 0.19315174 0.4152522 0.38224601 0.1435684 0.09182490
## AMD  0.3030626 0.3335709 0.34979807 0.4516329 0.38781251 0.1379067 0.03907121
##               PG         PEP      MDLZ        KO         EL
## TSM   0.22332488 0.331596563 0.2746476 0.4295469 0.25282221
## AAPL  0.14023665 0.220589041 0.1558214 0.1710399 0.23472668
## NVDA -0.07196894 0.009185657 0.1425258 0.1295878 0.13086183
## TSLA  0.08356726 0.119718469 0.1349797 0.2218584 0.11535336
## MSFT  0.19265238 0.348626415 0.3954915 0.4343138 0.19395054
## AMD   0.16057256 0.222287672 0.1800234 0.1311243 0.08478151
```

```r
stdev <- diag(cov(rr))^.5
```

```r
for (i in 1:5){
  g_a = (i*6-5):(i*6)
  cormat[g_a,g_a] <- (sum(cormat[g_a,g_a]) - 6 )/30
  for (j in (i+1):5){
```

```
    if (i >= 5){
      break
    }
    g_b = (j*6-5):(j*6)
    cormat[g_a,g_b] = mean(cormat[g_a, g_b])
    cormat[g_b,g_a] = mean(cormat[g_a,g_b])
  }
}
diag(cormat) = 1
head(cormat)
```

```
##            TSM      AAPL      NVDA      TSLA      MSFT       AMD      GOOG
## TSM  1.0000000 0.3006055 0.3006055 0.3006055 0.3006055 0.3006055 0.2150248
## AAPL 0.3006055 1.0000000 0.3006055 0.3006055 0.3006055 0.3006055 0.2150248
## NVDA 0.3006055 0.3006055 1.0000000 0.3006055 0.3006055 0.3006055 0.2150248
## TSLA 0.3006055 0.3006055 0.3006055 1.0000000 0.3006055 0.3006055 0.2150248
## MSFT 0.3006055 0.3006055 0.3006055 0.3006055 1.0000000 0.3006055 0.2150248
## AMD  0.3006055 0.3006055 0.3006055 0.3006055 0.3006055 1.0000000 0.2150248
##             FB      NFLX      TWTR       CHT     CMCSA       PLD       AMT
## TSM  0.2150248 0.2150248 0.2150248 0.2150248 0.2150248 0.1759939 0.1759939
## AAPL 0.2150248 0.2150248 0.2150248 0.2150248 0.2150248 0.1759939 0.1759939
## NVDA 0.2150248 0.2150248 0.2150248 0.2150248 0.2150248 0.1759939 0.1759939
## TSLA 0.2150248 0.2150248 0.2150248 0.2150248 0.2150248 0.1759939 0.1759939
## MSFT 0.2150248 0.2150248 0.2150248 0.2150248 0.2150248 0.1759939 0.1759939
## AMD  0.2150248 0.2150248 0.2150248 0.2150248 0.2150248 0.1759939 0.1759939
##            CCI       PSA      EQIX      WELL       JNJ       UNH       MRK
## TSM  0.1759939 0.1759939 0.1759939 0.1759939 0.2420958 0.2420958 0.2420958
## AAPL 0.1759939 0.1759939 0.1759939 0.1759939 0.2420958 0.2420958 0.2420958
## NVDA 0.1759939 0.1759939 0.1759939 0.1759939 0.2420958 0.2420958 0.2420958
## TSLA 0.1759939 0.1759939 0.1759939 0.1759939 0.2420958 0.2420958 0.2420958
## MSFT 0.1759939 0.1759939 0.1759939 0.1759939 0.2420958 0.2420958 0.2420958
## AMD  0.1759939 0.1759939 0.1759939 0.1759939 0.2420958 0.2420958 0.2420958
##            ABT      ABBV       AZN       WMT        PG       PEP      MDLZ
## TSM  0.2420958 0.2420958 0.2420958 0.1809633 0.1809633 0.1809633 0.1809633
## AAPL 0.2420958 0.2420958 0.2420958 0.1809633 0.1809633 0.1809633 0.1809633
## NVDA 0.2420958 0.2420958 0.2420958 0.1809633 0.1809633 0.1809633 0.1809633
## TSLA 0.2420958 0.2420958 0.2420958 0.1809633 0.1809633 0.1809633 0.1809633
## MSFT 0.2420958 0.2420958 0.2420958 0.1809633 0.1809633 0.1809633 0.1809633
## AMD  0.2420958 0.2420958 0.2420958 0.1809633 0.1809633 0.1809633 0.1809633
##             KO        EL
## TSM  0.1809633 0.1809633
## AAPL 0.1809633 0.1809633
## NVDA 0.1809633 0.1809633
## TSLA 0.1809633 0.1809633
## MSFT 0.1809633 0.1809633
## AMD  0.1809633 0.1809633
```

```
covmat = matrix(nrow = 30,ncol=30,dimnames= list(names(stdev),names(stdev)))
for (i in 1:30){
  for (j in 1:30){
    covmat[i,j] = cormat[i,j] * stdev[i] * stdev[j]
  }
}
```

```r
# diag(covmat) = 1
head(covmat)
```

```
##               TSM         AAPL         NVDA         TSLA         MSFT          AMD
## TSM  0.004062383 0.001376484 0.002294916 0.002250663 0.001154369 0.003323834
## AAPL 0.001376484 0.005161405 0.002586784 0.002536904 0.001301183 0.003746562
## NVDA 0.002294916 0.002586784 0.014346924 0.004229601 0.002169370 0.006246379
## TSLA 0.002250663 0.002536904 0.004229601 0.013798960 0.002127538 0.006125931
## MSFT 0.001154369 0.001301183 0.002169370 0.002127538 0.003630068 0.003142001
## AMD  0.003323834 0.003746562 0.006246379 0.006125931 0.003142001 0.030095678
##              GOOG           FB         NFLX         TWTR          CHT
## TSM  0.0007907413 0.0008439148 0.001800865 0.001939292 0.0004469362
## AAPL 0.0008913083 0.0009512445 0.002029900 0.002185933 0.0005037778
## NVDA 0.0014860157 0.0015859430 0.003384310 0.003644452 0.0008399133
## TSLA 0.0014573611 0.0015553615 0.003319051 0.003574177 0.0008237174
## MSFT 0.0007474831 0.0007977477 0.001702347 0.001833202 0.0004224861
## AMD  0.0021522666 0.0022969960 0.004901655 0.005278432 0.0012164860
##             CMCSA          PLD          AMT          CCI          PSA
## TSM  0.0008020364 0.0005877306 0.0005732038 0.0005138900 0.0005476926
## AAPL 0.0009040399 0.0006624785 0.0006461042 0.0005792469 0.0006173485
## NVDA 0.0015072422 0.0011045039 0.0010772041 0.0009657377 0.0010292617
## TSLA 0.0014781783 0.0010832059 0.0010564326 0.0009471155 0.0010094146
## MSFT 0.0007581603 0.0005555782 0.0005418461 0.0004857772 0.0005177305
## AMD  0.0021830099 0.0015997051 0.0015601655 0.0013987234 0.0014907282
##              EQIX         WELL          JNJ          UNH          MRK
## TSM  0.0006076410 0.0006585690 0.0006100724 0.0007408762 0.0007687610
## AAPL 0.0006849212 0.0007423263 0.0006876618 0.0008351013 0.0008665326
## NVDA 0.0011419210 0.0012376284 0.0011464902 0.0013923057 0.0014447088
## TSLA 0.0011199015 0.0012137634 0.0011243827 0.0013654581 0.0014168507
## MSFT 0.0005743995 0.0006225414 0.0005766979 0.0007003459 0.0007267053
## AMD  0.0016538980 0.0017925156 0.0016605159 0.0020165420 0.0020924399
##               ABT         ABBV          AZN          WMT           PG
## TSM  0.0008628042 0.001179512 0.0010276903 0.0006246802 0.0004618618
## AAPL 0.0009725362 0.001329523 0.0011583927 0.0007041275 0.0005206017
## NVDA 0.0016214412 0.002216620 0.0019313066 0.0011739422 0.0008679625
## TSLA 0.0015901752 0.002173877 0.0018940655 0.0011513053 0.0008512258
## MSFT 0.0008156037 0.001114986 0.0009714696 0.0005905065 0.0004365952
## AMD  0.0023484097 0.003210435 0.0027972024 0.0017002759 0.0012571111
##               PEP         MDLZ           KO           EL
## TSM  0.0004649805 0.0005818332 0.0004351805 0.0005938492
## AAPL 0.0005241170 0.0006558311 0.0004905271 0.0006693753
## NVDA 0.0008738235 0.0010934211 0.0008178213 0.0011160024
## TSLA 0.0008569737 0.0010723368 0.0008020514 0.0010944827
## MSFT 0.0004395433 0.0005500035 0.0004113736 0.0005613621
## AMD  0.0012655998 0.0015836533 0.0011844891 0.0016163588
```

```r
R_f = 0.002
R = exp_R - R_f

X_alloc = (solve(covmat)%*%R)/ as.numeric(rep(1,30) %*% solve(covmat) %*% R)

X_alloc
```

```
##               [,1]
## TSM    0.081402607
## AAPL   0.022362851
## NVDA   0.134024656
## TSLA  -0.066757588
## MSFT   0.188042833
## AMD    0.022361827
## GOOG   0.041680595
## FB     0.099983940
## NFLX   0.074563317
## TWTR  -0.106201937
## CHT    0.090729655
## CMCSA -0.071143897
## PLD    0.016841923
## AMT    0.172973319
## CCI    0.092199744
## PSA   -0.096995507
## EQIX   0.173733544
## WELL  -0.092163161
## JNJ    0.001979488
## UNH    0.564435002
## MRK   -0.055825840
## ABT    0.079628114
## ABBV   0.001751442
## AZN   -0.135363764
## WMT   -0.111249830
## PG    -0.105003059
## PEP    0.031961648
## MDLZ  -0.162172043
## KO    -0.037482573
## EL     0.149702693
```

```
port_exp_R = as.numeric(t(X_alloc)%*%exp_R)
port_stdev = as.numeric(t(X_alloc) %*% covmat %*% X_alloc)^.5
port_exp_R
```

```
## [1] 0.03362654
```

```
port_stdev
```

```
## [1] 0.04807148
```

Breakpoint 3 Project 5 a.

```
train <- read.csv("stockDataTrain.csv", sep=",", header=TRUE)
```

```
returns <- (train[-1,3:ncol(train)]-train[-nrow(train),3:ncol(train)])/train[-nrow(train),3:ncol(train)]
```

```
rr <- returns[,-1]
index <- returns[,1]
head(rr)
```

```
##            TSM       AAPL        NVDA          TSLA        MSFT         AMD
## 2  0.067966914 0.05121871  0.17070078  0.3494846108  0.01242055  0.08163265
## 3  0.107913530 0.02605846 -0.02117160 -0.1485233998  0.07797878  0.08086253
## 4  0.003996158 0.09939605  0.03126701 -0.0026864956 -0.01439350  0.01995012
## 5  0.022885268 0.07271754  0.02869540 -0.0005771803  0.01336632 -0.02200489
## 6  0.040369884 0.03340213 -0.01971600  0.1554122309  0.02574914  0.04750000
## 7 -0.064983478 0.02873098 -0.05609494 -0.0698158987  0.03501199 -0.06682578
##           GOOG           FB        NFLX       TWTR          CHT        CMCSA
## 2  0.029365727  0.094134553  0.08868150 -0.1486822  0.028911726 -0.050688595
## 3 -0.082750882 -0.120070072 -0.21003972 -0.1500638  0.014214828 -0.031921145
## 4 -0.054419409 -0.007636172 -0.08519163 -0.1649882  0.023142138  0.039089837
## 5  0.063095775  0.058882570  0.29744751 -0.1675648  0.014654464  0.008500881
## 6  0.027487499  0.063033208  0.05449585  0.2629470  0.006593393  0.028352277
## 7 -0.006396881  0.079655237 -0.04058098  0.1030021 -0.056144736  0.005109618
##            PLD          AMT          CCI          PSA        EQIX        WELL
## 2  0.062693927 0.007294700  0.069616931  0.072403346  0.02570208 0.014157854
## 3 -0.008740028 0.004909937 -0.027931708 -0.003017811 -0.02695303 0.029014868
## 4  0.003202099 0.020153397 -0.009602118  0.050427962  0.01606782 0.058557368
## 5  0.021658586 0.077430553  0.054997968 -0.017834140  0.05825051 0.002218882
## 6 -0.010118024 0.003905081 -0.032190846 -0.005975265  0.05705641 0.003683868
## 7  0.001015610 0.053068066  0.003718001  0.009755617  0.02108652 0.015318362
##            JNJ          UNH           MRK          ABT         ABBV          AZN
## 2  0.041256890  0.069036430  0.0758923301  0.091173470  0.042316607  0.06708666
## 3  0.073999511  0.061085122 -0.0038603275 -0.031925690  0.009624971 -0.01504280
## 4  0.031151227 -0.081459836  0.0399720579  0.005972766  0.013229588  0.21840318
## 5  0.001678525  0.061167487 -0.0119534510  0.038916678  0.052538949 -0.08665395
## 6  0.038331612  0.026623209 -0.0001728946  0.022244611  0.038836598  0.02922433
## 7 -0.043299888 -0.003903819 -0.0114363785  0.029828673 -0.072643602 -0.02045492
##             WMT          PG          PEP          MDLZ           KO
## 2  0.0002677824  0.03438440 -0.003608634  0.0387670709  0.010047332
## 3  0.0231594815  0.02466324  0.042837588  0.0152805604  0.012041789
## 4  0.0496429779  0.02419383  0.035893769  0.0360825217  0.063454976
## 5 -0.0368834760 -0.01350660  0.028408101  0.0552597191  0.002942074
## 6 -0.0160899923 -0.02723131  0.011434796 -0.0002659565  0.035443670
## 7 -0.0198480014 -0.01615955 -0.006503135 -0.0386491729 -0.065498296
##             EL
## 2  0.001454896
## 3 -0.025584005
## 4  0.085077885
## 5  0.055808312
## 6 -0.028230138
## 7 -0.010772920
```

```
index
```

```
##   [1]  0.0431170300  0.0069321656  0.0062007890  0.0210302800  0.0190583317
##   [6] -0.0150798306  0.0376552955 -0.0155138372  0.0232014608  0.0245335888
##  [11] -0.0041885879 -0.0310408058  0.0548925110 -0.0173961069  0.0085208197
##  [16]  0.0104913824 -0.0210116724  0.0197420297 -0.0625808182 -0.0264428316
##  [21]  0.0829831178  0.0005048693 -0.0175301852 -0.0507353220 -0.0041283604
##  [26]  0.0659911146  0.0026993985  0.0153246024  0.0009109211  0.0356098011
##  [31] -0.0012192431 -0.0012344508 -0.0194256793  0.0341745222  0.0182007622
##  [36]  0.0178843582  0.0371981603 -0.0003891972  0.0090912085  0.0115762514
##  [41]  0.0048137751  0.0193488261  0.0005464328  0.0193029785  0.0221881353
```

```
## [46]  0.0280826277  0.0098316305  0.0561787044 -0.0389473721 -0.0268844986
## [51]  0.0027187751  0.0216083420  0.0048424360  0.0360215562  0.0302632115
## [56]  0.0042942871 -0.0694033560  0.0178593568 -0.0917768946
```

```
sigma_m2 <- var(index)
```

Calculating the betas, alphas, sigmas, and the variance of betas through linear regression model

```
beta2 <- rep(0,30)

alpha2 <- rep(0,30)

sigma_e2 <- rep(0,30)

var_beta2 <- rep(0,30)

for(i in 1:30){
    q <- lm(data=rr, formula=rr[,i] ~ index)
    beta2[i] <- q$coefficients[2]
    alpha2[i] <- q$coefficients[1]
    sigma_e2[i] <- summary(q)$sigma^2
    var_beta2[i] <- vcov(q)[2,2]
}
```

```
beta2
```

```
##  [1] 1.1575430 1.1546817 2.0393210 0.7257803 1.2465438 3.3617924 1.0647161
##  [8] 0.6993114 1.2186013 0.2482677 0.1143192 1.0719706 0.9615390 0.6452853
## [15] 0.4331112 0.3109568 0.7489639 0.3200383 0.6801403 0.8557822 0.7613453
## [22] 1.2635732 1.4733349 0.5777802 0.3278997 0.3664477 0.6786786 0.8151057
## [29] 0.5684855 0.6411576
```

```
alpha2
```

```
##  [1]  0.0107539160  0.0103466129  0.0317618618  0.0123216653  0.0128601908
##  [6]  0.0223271122  0.0045202159  0.0100819627  0.0264490179 -0.0052551145
## [11]  0.0066926391  0.0002002969  0.0050511629  0.0102125141  0.0085076868
## [16]  0.0062233532  0.0105575104  0.0070851967  0.0052548916  0.0182979847
## [21]  0.0053264319  0.0070714054  0.0073396613  0.0051992860  0.0053453223
## [26]  0.0043234165  0.0043039592  0.0009773474  0.0036956149  0.0092098549
```

```
sigma_e2
```

```
##  [1] 0.002801742 0.003926622 0.010464609 0.013517433 0.002149154 0.019389460
##  [7] 0.002260526 0.003372151 0.016093300 0.020312993 0.001069156 0.002342581
## [13] 0.001874361 0.002243112 0.001949121 0.002329658 0.002428271 0.003405539
## [19] 0.001130768 0.001617797 0.001949505 0.001594339 0.003787930 0.004181755
## [25] 0.002877866 0.001498124 0.001195860 0.001928903 0.001127292 0.002288760
```

VAR Covariance Matrix for SIM

```r
var_i <- sigma_e2 + beta2^2*var(index)
covmat <- beta2 %*% t(beta2)
covmat <- covmat * as.numeric(var(index))
# covmat <- covmat + diag(var_i)
covmat <- covmat + diag(sigma_e2)
rownames(covmat) <- names(rr)
colnames(covmat) <- names(rr)
head(covmat)
```

```
##                TSM          AAPL         NVDA          TSLA         MSFT          AMD
## TSM  0.0041106888 0.0013057112 0.002306059 0.0008207105 0.001409589 0.003801506
## AAPL 0.0013057112 0.0052291052 0.002300359 0.0008186818 0.001406104 0.003792110
## NVDA 0.0023060592 0.0023003589 0.014527348 0.0014459007 0.002483367 0.006697369
## TSLA 0.0008207105 0.0008186818 0.001445901 0.0140320188 0.000883813 0.002383547
## MSFT 0.0014095886 0.0014061043 0.002483367 0.0008838130 0.003667122 0.004093795
## AMD  0.0038015065 0.0037921096 0.006697369 0.0023835472 0.004093795 0.030429979
##               GOOG           FB          NFLX         TWTR          CHT
## TSM  0.0012039783 0.0007907797 0.0013779914 0.0002807405 1.292718e-04
## AAPL 0.0012010023 0.0007888249 0.0013745851 0.0002800466 1.289523e-04
## NVDA 0.0021211293 0.0013931695 0.0024276997 0.0004945994 2.277468e-04
## TSLA 0.0007548953 0.0004958194 0.0008640016 0.0001760245 8.105353e-05
## MSFT 0.0012965494 0.0008515809 0.0014839419 0.0003023260 1.392112e-04
## AMD  0.0034966522 0.0022966205 0.0040020292 0.0008153402 3.754375e-04
##              CMCSA          PLD          AMT          CCI          PSA
## TSM  0.0012121817 0.0010873060 0.0007296871 0.0004897611 0.0003516292
## AAPL 0.0012091853 0.0010846183 0.0007278834 0.0004885504 0.0003507600
## NVDA 0.0021355817 0.0019155798 0.0012855386 0.0008628449 0.0006194887
## TSLA 0.0007600388 0.0006817416 0.0004575143 0.0003070805 0.0002204717
## MSFT 0.0013053835 0.0011709064 0.0007857910 0.0005274176 0.0003786651
## AMD  0.0035204767 0.0031578066 0.0021191925 0.0014223878 0.0010212185
##               EQIX         WELL          JNJ          UNH          MRK
## TSM  0.0008469266 0.0003618985 0.0007691010 0.0009677164 0.0008609274
## AAPL 0.0008448331 0.0003610039 0.0007671999 0.0009653243 0.0008587992
## NVDA 0.0014920873 0.0006375808 0.0013549768 0.0017048909 0.0015167534
## TSLA 0.0005310236 0.0002269106 0.0004822269 0.0006067589 0.0005398021
## MSFT 0.0009120448 0.0003897240 0.0008282354 0.0010421219 0.0009271221
## AMD  0.0024596851 0.0010510431 0.0022336604 0.0028104889 0.0025003469
##                ABT         ABBV          AZN          WMT           PG
## TSM  0.0014288455 0.001666043 0.0006533524 0.0003707881 0.0004143781
## AAPL 0.0014253135 0.001661925 0.0006517374 0.0003698716 0.0004133538
## NVDA 0.0025172928 0.002935180 0.0011510547 0.0006532423 0.0007300377
## TSLA 0.0008958871 0.001044610 0.0004096524 0.0002324844 0.0002598154
## MSFT 0.0015387061 0.001794142 0.0007035871 0.0003992972 0.0004462387
## AMD  0.0041497221 0.004838604 0.0018974976 0.0010768609 0.0012034570
##                PEP          MDLZ           KO           EL
## TSM  0.0007674481 0.0009217195 0.0006428420 0.0007250195
## AAPL 0.0007655511 0.0009194412 0.0006412530 0.0007232273
## NVDA 0.0013520648 0.0016238551 0.0011325378 0.0012773154
## TSLA 0.0004811905 0.0005779188 0.0004030624 0.0004545877
## MSFT 0.0008264554 0.0009925884 0.0006922686 0.0007807645
## AMD  0.0022288600 0.0026769025 0.0018669729 0.0021056366
```

```
Rf <- 0.002
R_ibar <- colMeans(rr)
R <- R_ibar-Rf
R_ibar
```

```
##          TSM         AAPL         NVDA         TSLA         MSFT          AMD
##  0.018022831  0.017597560  0.044567996  0.016879280  0.020687995  0.043437846
##         GOOG           FB         NFLX         TWTR          CHT        CMCSA
##  0.011206214  0.014473364  0.034101355 -0.003696091  0.007410518  0.006931851
##          PLD          AMT          CCI          PSA         EQIX         WELL
##  0.011089250  0.014264652  0.011227455  0.008176040  0.015260709  0.009094911
##          JNJ          UNH          MRK          ABT         ABBV          AZN
##  0.009525905  0.023671961  0.010107381  0.015006148  0.016591625  0.008827518
##          WMT           PG          PEP         MDLZ           KO           EL
##  0.007404403  0.006624564  0.008565794  0.006095891  0.007265480  0.013236073
```

```
df <- data.frame(matrix(ncol=6,nrow=0))
df <- cbind(colnames(rr),alpha2,beta2,R_ibar,sigma_e2,(R_ibar - Rf)/beta2)
df = data.frame(df)
colnames(df) <- c("stock","alpha","beta","expected return of stock","sigma^2","excess return to beta ra
df = df[order(df$`excess return to beta ratio`,decreasing=TRUE),]
df = cbind(df,(R_ibar-Rf)*beta2/sigma_e2,cumsum((R_ibar-Rf)*beta2/sigma_e2),beta2^2/sigma_e2,cumsum(beta
df <- cbind(df,(sigma_m2*df$`cumsum((R_ibar - Rf) * beta2/sigma_e2)`)/(1+sigma_m2*df$`cumsum(beta2^2/sig
colnames(df) <- c("stock","alpha","beta","expected return of stock","sigma^2","excess return to beta ra

#if short sales allowed:
C_star_allowed <- df$C[length(df$C)]
#if short sales not allowed
C_star <- df$C[which(df$`excess return to beta ratio` > df$C)[length(which(df$`excess return to beta ra
```

a.

```
df
```

```
##       stock                alpha               beta expected return of stock
## CHT     CHT  0.00669263907115864 0.114319164853984      0.00741051846790379
## NFLX   NFLX   0.0264490179120803  1.21860131508328       0.0341013549289067
## UNH     UNH   0.0182979847323322 0.855782220524718       0.0236719605219577
## WELL   WELL  0.00708519669666882 0.320038264897381      0.00909491113810431
## CCI     CCI  0.00850768680190838  0.43311118812552       0.0112274547327908
## NVDA   NVDA   0.0317618618371258  2.03932104594676       0.0445679959976041
## TSLA   TSLA    0.012321665289032 0.725780263879404       0.0168792799170672
## PSA     PSA   0.0062233532243196 0.310956795445091      0.00817603960908326
## AMT     AMT   0.0102125141479574 0.645285339536314       0.0142646523075258
## FB       FB   0.0100819626761704 0.699311446269394       0.0144733635390621
## EQIX   EQIX   0.0105575103587186  0.74896393799576       0.0152607093427123
## EL       EL  0.00920985493494317 0.641157624276606       0.0132360726659958
## WMT     WMT  0.00534532228914836 0.327899680310383      0.00740440332770216
## MSFT   MSFT   0.0128601907820131  1.24654376313014       0.0206879953835887
## TSM     TSM   0.0107539159973098  1.15754297987532       0.0180228306815699
## AAPL   AAPL   0.0103466129290547  1.15468167334493        0.017597559733576
## PG       PG   0.0043234164772652  0.36644768499596      0.00662456382538972
```

```
## AMD     AMD   0.0223271122321973  3.36179235105671       0.0434378460977005
## AZN     AZN   0.0051992860102525 0.577780172496018      0.00882751826870652
## JNJ     JNJ  0.00525489157321153 0.680140337002439      0.00952590540936918
## MRK     MRK  0.00532643189611972 0.761345276707195       0.0101073808179862
## ABT     ABT  0.00707140541624719  1.26357320932821       0.0150061482427406
## ABBV   ABBV  0.00733966128709105  1.47333490790803       0.0165916250872423
## PEP     PEP     0.00430395919008 0.678678633758669      0.00856579410471387
## PLD     PLD   0.0050511628986216   0.9615390329351       0.0110892498789807
## KO       KO  0.00369561494741026 0.568485507793538      0.00726548036686501
## GOOG   GOOG  0.00452021588076655  1.06471610489039       0.0112062142963412
## MDLZ   MDLZ 0.000977347440130093 0.815105726862176      0.00609589084648479
## CMCSA CMCSA 0.000200296894146275  1.07197058062099      0.00693185056432624
## TWTR   TWTR -0.00525511449052535 0.248267714738308     -0.00369609087873399
##                   sigma^2 excess return to beta ratio        Col1       Col2
## CHT   0.00106915625998746          0.0473281839909736  6.61985098   6.619851
## NFLX   0.0160932997123741          0.0263427870391827  4.58669520  11.206546
## UNH   0.00161779675597616          0.0253241537416723  8.29556175  19.502108
## WELL  0.00340553940263042          0.0221689463926423  0.79890079  20.301009
## CCI   0.00194912078269453           0.021305048185725 10.83933856  31.140347
## NVDA   0.0104646089943664          0.0208736118730348  7.18459591  38.324943
## TSLA   0.0135174326719554          0.0205010809160547  4.33616091  42.661104
## PSA   0.00232965769528159          0.0198614074352134  2.58670674  45.247811
## AMT   0.00224311178209009          0.0190065565666483  2.43074783  47.678559
## FB    0.00337215106808481          0.0178366357444935 -0.06961827  47.608940
## EQIX  0.00242827072727569          0.0177054043191962  0.57851782  48.187458
## EL    0.00228875970907348          0.0175246651378014  2.25682645  50.444285
## WMT   0.00287786636238358          0.0164818804415621  4.66274557  55.107030
## MSFT  0.00214915365616709          0.0149918486108037  3.52822378  58.635254
## TSM   0.00280174209968681          0.0138421043193539  2.05041879  60.685673
## AAPL  0.00392662158494458          0.0135081036563024  0.82436209  61.510035
## PG    0.00149812400535767          0.0126199837377625  4.09006829  65.600103
## AMD     0.019389459862153          0.0123261170740291  0.66674990  66.266853
## AZN    0.0041817551179446          0.0118168095648065  4.52672066  70.793574
## JNJ   0.00113076821477706          0.0110652243366977 11.46403492  82.257609
## MRK    0.0019495052350728          0.0106487569648431  3.16619621  85.423805
## ABT   0.00159433935288406          0.0102931497334099 10.30785601  95.731661
## ABBV  0.00378792990865251          0.0099038073481614 6 5.67548796 101.407149
## PEP   0.00119585975184197          0.0096743786795689 8  0.94333708 102.350486
## PLD   0.00187436101741367          0.0094528142567813 5  0.61576943 102.966255
## KO     0.0011272920141441          0.0092622948073063 7  1.13118854 104.097444
## GOOG  0.00226052603629044          0.0086466375910496 7  3.72624312 107.823687
## MDLZ   0.0019289030498278          0.0050249810687164 4  1.73082006 109.554507
## CMCSA 0.00234258097732937          0.0046007331296995 3  2.65534506 112.209852
## TWTR   0.0203129932936262          -0.022943341161931 1  3.14759720 115.357449
##              Col3      Col4           C
## CHT    478.240217  478.2402 0.004407674
## NFLX   339.551377  817.7916 0.006085740
## UNH    397.418607 1215.2102 0.008710723
## WELL    38.968716 1254.1789 0.008912431
## CCI    723.015476 1977.1944 0.010377189
## NVDA   582.875846 2560.0702 0.010694173
## TSLA   501.485214 3061.5555 0.010442825
## PSA    145.022121 3206.5776 0.010696302
## AMT     92.273753 3298.8513 0.011030312
```

```
## FB       3.034356 3301.8857 0.011006480
## EQIX    12.223537 3314.1092 0.011108832
## EL     490.536266 3804.6455 0.010447634
## WMT    493.265333 4297.9108 0.010355423
## MSFT   185.631930 4483.5427 0.010647028
## TSM     96.240984 4579.7837 0.010830083
## AAPL    41.505724 4621.2895 0.010896488
## PG     231.006771 4852.2962 0.011164173
## AMD     30.075850 4882.3721 0.011220214
## AZN    409.094341 5291.4664 0.011210174
## JNJ    452.691728 5744.1581 0.012154243
## MRK    297.330123 6041.4883 0.012090886
## ABT   1001.428744 7042.9170 0.011867705
## ABBV   573.061224 7615.9782 0.011737441
## PEP     79.830100 7695.8083 0.011738168
## PLD     37.360387 7733.1687 0.011758406
## KO      89.634707 7822.8034 0.011767136
## GOOG   385.166143 8207.9696 0.011679821
## MDLZ   344.443100 8552.4127 0.011440452
## CMCSA  286.683280 8839.0960 0.011377139
## TWTR   179.609549 9018.7055 0.011487089
```

```
# C_star <- df$C[nrow(df)]
# C_star
```

```
# z <- (as.numeric(df$beta) / as.numeric(df$`sigma^2`)) * (as.numeric(df$`excess return to beta ratio`)
# # z
# alloc <- z / sum(z)
# names(alloc) <- row.names(df)
# # alloc
# alloc <- alloc[colnames(rrr)] #reordering stocks
# alloc
#
# sigma_m2 * (alloc%*%beta2) * ((alloc %*% colMeans(rrr) - Rf)/ (alloc%*%covmat %*% alloc))
```

b.

If short sales are allowed, the answer of the composition matches the z_i and x_i computed from covariance matrix of single index model in project 4, which is a good way to double check our work.

```
z_i_allowed <- (beta2/sigma_e2)*(((R_ibar - Rf)/beta2)-C_star_allowed)
x_i_allowed <- z_i_allowed/sum(z_i_allowed)
x_i_allowed
```

```
##          TSM         AAPL         NVDA         TSLA         MSFT          AMD
##  0.055152410  0.033687943  0.103688140  0.027434045  0.115228543  0.008246005
##         GOOG           FB         NFLX         TWTR          CHT        CMCSA
## -0.075835640  0.074639392  0.063763382 -0.023853411  0.217230531 -0.178623842
##          PLD          AMT          CCI          PSA         EQIX         WELL
## -0.059154112  0.122616728  0.123664093  0.063360582  0.108717210  0.056901605
##          JNJ          UNH          MRK          ABT         ABBV          AZN
## -0.014383331  0.414901508 -0.018558158 -0.053636878 -0.034907550  0.002582333
##          WMT           PG          PEP         MDLZ           KO           EL
##  0.032258895  0.015707810 -0.058314155 -0.154788774 -0.063596685  0.095871382
```

If short sales are not allowed, the answer is a little different compared to when short sales are allowed.

```
z_i <- (beta2/sigma_e2)*(((R_ibar - Rf)/beta2)-C_star)
x_i <- z_i/sum(z_i)
x_i
```

```
##          TSM         AAPL         NVDA         TSLA         MSFT          AMD
##  0.053585865  0.033300200  0.092802810  0.024583059  0.108091390  0.009534856
##         GOOG           FB         NFLX         TWTR          CHT        CMCSA
## -0.059501874  0.067719257  0.056467341 -0.020570677  0.190313037 -0.149045831
##          PLD          AMT          CCI          PSA         EQIX         WELL
## -0.044426406  0.110525095  0.110542284  0.056905316  0.098724566  0.050751020
##          JNJ          UNH          MRK          ABT         ABBV          AZN
## -0.004296439  0.367921916 -0.010804623 -0.035815191 -0.025039834  0.004130464
##          WMT           PG          PEP         MDLZ           KO           EL
##  0.029599779  0.016993844 -0.042952070 -0.128802310 -0.048407407  0.087170562
```

```
cov.matrix <- cov(rr)
# cov.matrix <- covmat
head(cov.matrix)
```

```
##               TSM        AAPL        NVDA         TSLA        MSFT         AMD
## TSM  4.062383e-03 0.002235268 0.003919135 1.761271e-05 0.002154460 0.003203961
## AAPL 2.235268e-03 0.005161405 0.004010693 1.122055e-03 0.001680854 0.003179227
## NVDA 3.919135e-03 0.004010693 0.014346924 1.283192e-03 0.003192779 0.008661374
## TSLA 1.761271e-05 0.001122055 0.001283192 1.379896e-02 0.000708004 0.001037054
## MSFT 2.154460e-03 0.001680854 0.003192779 7.080040e-04 0.003630068 0.003246639
## AMD  3.203961e-03 0.003179227 0.008661374 1.037054e-03 0.003246639 0.030095678
##              GOOG           FB        NFLX          TWTR           CHT
## TSM  0.0015376332 0.0003612575 0.001943573 -4.434180e-04  6.051009e-04
## AAPL 0.0012326992 0.0014249149 0.002563224  2.593836e-03  4.940016e-04
## NVDA 0.0028003856 0.0014930262 0.005548375  1.655664e-03  5.575223e-04
## TSLA 0.0005932909 0.0016592685 0.004010646  3.328376e-03  2.849212e-04
## MSFT 0.0020387217 0.0011207815 0.002528836 -8.821019e-05 -8.969531e-05
## AMD  0.0029380051 0.0013884886 0.003750453 -1.517128e-03  5.058177e-04
##            CMCSA          PLD          AMT          CCI           PSA
## TSM  0.001076460 0.0009981227 0.0010760083 0.0008552944  0.0001515250
## AAPL 0.001077286 0.0010545360 0.0014286508 0.0010344209  0.0007972601
## NVDA 0.001953092 0.0017343381 0.0001402959 0.0008479314  0.0001404484
## TSLA 0.001270848 0.0006290024 0.0009779647 0.0004549391  0.0004264907
## MSFT 0.001320988 0.0008251445 0.0010978204 0.0004660815  0.0001267023
## AMD  0.002001670 0.0032649395 0.0015374327 0.0009738502 -0.0004054600
##              EQIX          WELL          JNJ          UNH          MRK
## TSM  0.0011843750  0.0005110208 0.0010428388 0.0006578176 0.0005272151
## AAPL 0.0006032473  0.0004997238 0.0005154489 0.0006846134 0.0004479273
## NVDA 0.0016894839 -0.0007364419 0.0004849562 0.0012461212 0.0003987402
## TSLA 0.0007740256  0.0012794605 0.0003924294 0.0010561565 0.0011217768
## MSFT 0.0013832868 -0.0003809920 0.0008626523 0.0004447379 0.0005797887
## AMD  0.0008964220  0.0019107934 0.0020786828 0.0027784858 0.0030233136
##              ABT         ABBV          AZN          WMT           PG
## TSM  0.001674696 0.0012514596 0.0011850485 0.0009413284  0.0005699788
## AAPL 0.001611777 0.0004922211 0.0008394886 0.0005540373  0.0004034379
## NVDA 0.001839648 0.0023327039 0.0015171674 0.0006993916 -0.0003451880
```

```
## TSLA 0.001395675 0.0018691396 0.0010940279 0.0004265306  0.0003930886
## MSFT 0.001398956 0.0017604554 0.0005761039 0.0002996365  0.0004647966
## AMD  0.004380990 0.0051427873 0.0015933901 0.0003671012  0.0011154614
##               PEP         MDLZ           KO           EL
## TSM  8.520290e-04 0.0008830470 0.0010329745 0.0008296615
## AAPL 6.388837e-04 0.0005647143 0.0004636284 0.0008682439
## NVDA 4.435509e-05 0.0008611732 0.0005856416 0.0008070263
## TSLA 5.669415e-04 0.0007998516 0.0009833035 0.0006976679
## MSFT 8.467819e-04 0.0012020212 0.0009873010 0.0006016497
## AMD  1.554610e-03 0.0015754280 0.0008582701 0.0007572661
```

```
ones <- rep(1,30)
```

```
A <- t(ones) %*% solve(cov.matrix) %*% R_ibar
A
```

```
##          [,1]
## [1,] 33.11504
```

```
B <- t(R_ibar) %*% solve(cov.matrix) %*% R_ibar
B
```

```
##          [,1]
## [1,] 1.049438
```

```
C <- t(ones) %*% solve(cov.matrix) %*% ones
C
```

```
##          [,1]
## [1,] 3530.478
```

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

```
##          [,1]
## [1,] 2608.411
```

```
Sigma30 <- cov(rr)
head(Sigma30)
```

```
##               TSM        AAPL        NVDA         TSLA        MSFT         AMD
## TSM  4.062383e-03 0.002235268 0.003919135 1.761271e-05 0.002154460 0.003203961
## AAPL 2.235268e-03 0.005161405 0.004010693 1.122055e-03 0.001680854 0.003179227
## NVDA 3.919135e-03 0.004010693 0.014346924 1.283192e-03 0.003192779 0.008661374
## TSLA 1.761271e-05 0.001122055 0.001283192 1.379896e-02 0.000708004 0.001037054
## MSFT 2.154460e-03 0.001680854 0.003192779 7.080040e-04 0.003630068 0.003246639
## AMD  3.203961e-03 0.003179227 0.008661374 1.037054e-03 0.003246639 0.030095678
##              GOOG           FB        NFLX          TWTR           CHT
## TSM  0.0015376332 0.0003612575 0.001943573 -4.434180e-04  6.051009e-04
## AAPL 0.0012326992 0.0014249149 0.002563224  2.593836e-03  4.940016e-04
```

```
## NVDA 0.0028003856 0.0014930262 0.005548375  1.655664e-03  5.575223e-04
## TSLA 0.0005932909 0.0016592685 0.004010646  3.328376e-03  2.849212e-04
## MSFT 0.0020387217 0.0011207815 0.002528836 -8.821019e-05 -8.969531e-05
## AMD  0.0029380051 0.0013884886 0.003750453 -1.517128e-03  5.058177e-04
##            CMCSA          PLD          AMT          CCI           PSA
## TSM  0.001076460 0.0009981227 0.0010760083 0.0008552944  0.0001515250
## AAPL 0.001077286 0.0010545360 0.0014286508 0.0010344209  0.0007972601
## NVDA 0.001953092 0.0017343381 0.0001402959 0.0008479314  0.0001404484
## TSLA 0.001270848 0.0006290024 0.0009779647 0.0004549391  0.0004264907
## MSFT 0.001320988 0.0008251445 0.0010978204 0.0004660815  0.0001267023
## AMD  0.002001670 0.0032649395 0.0015374327 0.0009738502 -0.0004054600
##              EQIX          WELL          JNJ          UNH          MRK
## TSM  0.0011843750  0.0005110208 0.0010428388 0.0006578176 0.0005272151
## AAPL 0.0006032473  0.0004997238 0.0005154489 0.0006846134 0.0004479273
## NVDA 0.0016894839 -0.0007364419 0.0004849562 0.0012461212 0.0003987402
## TSLA 0.0007740256  0.0012794605 0.0003924294 0.0010561565 0.0011217768
## MSFT 0.0013832868 -0.0003809920 0.0008626523 0.0004447379 0.0005797887
## AMD  0.0008964220  0.0019107934 0.0020786828 0.0027784858 0.0030233136
##              ABT         ABBV          AZN          WMT           PG
## TSM  0.001674696 0.0012514596 0.0011850485 0.0009413284  0.0005699788
## AAPL 0.001611777 0.0004922211 0.0008394886 0.0005540373  0.0004034379
## NVDA 0.001839648 0.0023327039 0.0015171674 0.0006993916 -0.0003451880
## TSLA 0.001395675 0.0018691396 0.0010940279 0.0004265306  0.0003930886
## MSFT 0.001398956 0.0017604554 0.0005761039 0.0002996365  0.0004647966
## AMD  0.004380990 0.0051427873 0.0015933901 0.0003671012  0.0011154614
##               PEP         MDLZ           KO           EL
## TSM  8.520290e-04 0.0008830470 0.0010329745 0.0008296615
## AAPL 6.388837e-04 0.0005647143 0.0004636284 0.0008682439
## NVDA 4.435509e-05 0.0008611732 0.0005856416 0.0008070263
## TSLA 5.669415e-04 0.0007998516 0.0009833035 0.0006976679
## MSFT 8.467819e-04 0.0012020212 0.0009873010 0.0006016497
## AMD  1.554610e-03 0.0015754280 0.0008582701 0.0007572661
```

```
sdev30 <- (diag(Sigma30))^.5
sdev30
```

```
##        TSM       AAPL       NVDA       TSLA       MSFT        AMD       GOOG
## 0.06373683 0.07184292 0.11977864 0.11746897 0.06025004 0.17348106 0.05769731
##         FB       NFLX       TWTR        CHT      CMCSA        PLD        AMT
## 0.06157717 0.13140208 0.14150259 0.03261119 0.05852146 0.05239503 0.05110000
##        CCI        PSA       EQIX       WELL        JNJ        UNH        MRK
## 0.04581229 0.04882572 0.05417001 0.05871015 0.03953700 0.04801401 0.04982115
##        ABT       ABBV        AZN        WMT         PG        PEP       MDLZ
## 0.05591581 0.07644070 0.06660160 0.05415978 0.04004342 0.04031382 0.05044494
##         KO         EL
## 0.03773016 0.05148673
```

```
sdev_m = sigma_m2^.5
R_m = mean(index)
```

```
R_allowed_SIM <- t(x_i_allowed) %*% R_ibar
sdev_allowed_SIM <- (t(x_i_allowed) %*% covmat %*% x_i_allowed)^.5
R_not_allowed_SIM <- t(x_i) %*% R_ibar
sdev_not_allowed_SIM <- (t(x_i) %*% covmat %*% x_i)^.5
```

```
popo = x_i
```

b.

```
#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.
```

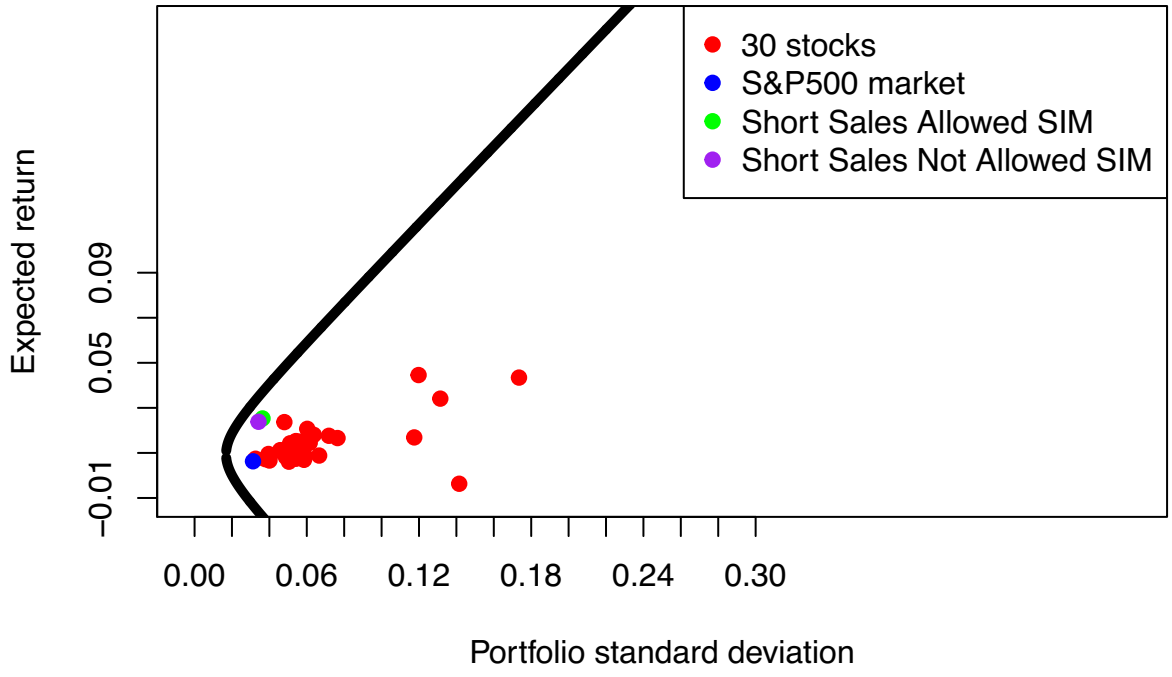
```
#    options(warn = -1)
    y1 <- (A + sqrt(D*(C*sdeff^2 - 1)))*(1/C)
```

```
## 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 deprecate
##   Use c() or as.vector() instead.

## Warning in sqrt(D * (C * sdeff^2 - 1)): NaNs produced

## 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 ari
##   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 sqrt(D * (C * sdeff^2 - 1)): NaNs produced

## 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 ari
##   Use c() or as.vector() instead.
```

```
#    options(warn = 0)

# plot(sdeff, y1, type = "n",xlim=c(0,0.3), ylim=c(-0.05,0.10), xlab="Portfolio standard deviation", yl
plot(sdeff, y1, type = "n",xlim=c(0,0.5), ylim=c(-0.01,0.2), xlab="Portfolio standard deviation", ylab="
axis(1, at=seq(0, 0.3, 0.02))
axis(2, at=seq(-0.05,0.10, 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 = "yellow")
    # #investors expected return portfolio
    # points(sd1,R1bar, pch = 19, col = "blue")
    #30 stocks
    points(sdev30,R_ibar, pch=19, col = "red")
    points(sdev_m,R_m,pch=19,col="blue")
    points(sdev_allowed_SIM, R_allowed_SIM,pch=19, col = "green")
    points(sdev_not_allowed_SIM,R_not_allowed_SIM, pch =19, col="purple")
    legend("topright",c("30 stocks","S&P500 market","Short Sales Allowed SIM","Short Sales Not Allowed S
```




```
    # #random weights
    # # points(sd_bar, r_bar, pch=19, col = "green")
    # points(sd_bar, rp_bar, pch=19, col = "green")
    # #equal allocation portfolio
    # points(sdp30_equal, Rp30_equal, pch=19, col="purple")
```

c.

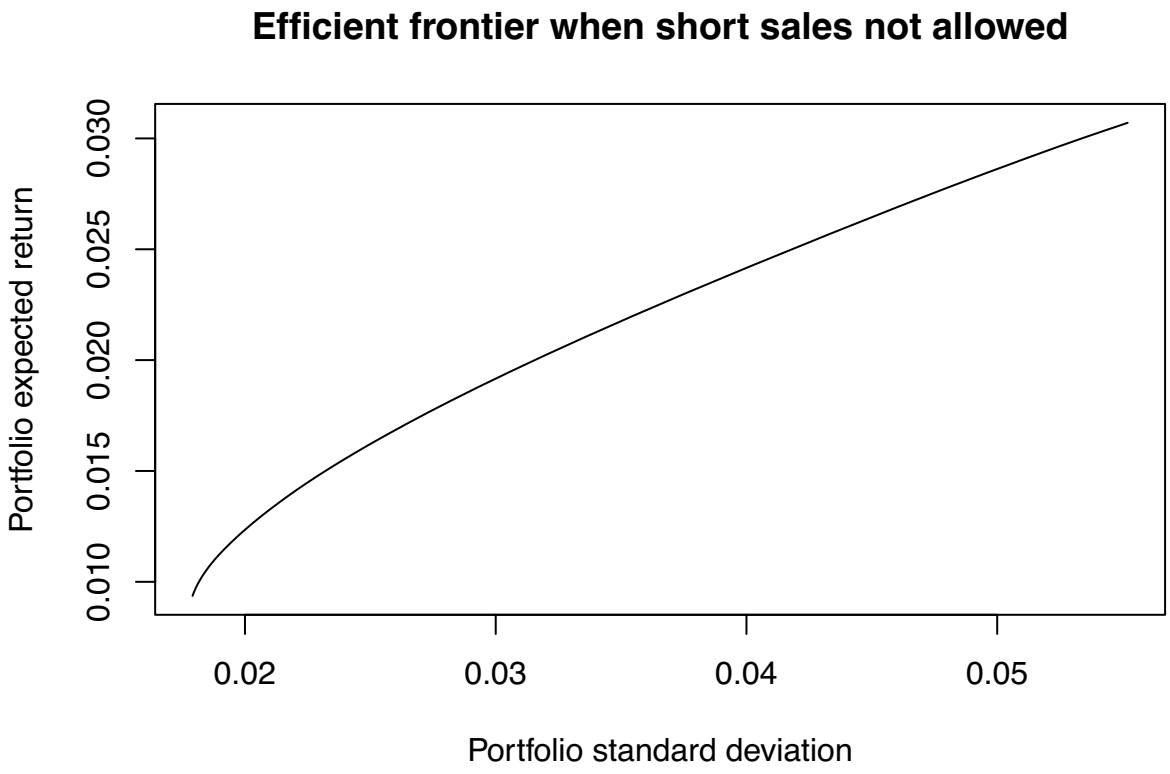
```
a <- read.csv("stockDataTrain.csv", sep=",", header=TRUE)
```

```
r <- (train[-1,3:ncol(train)]-train[-nrow(train),3:ncol(train)])/train[-nrow(train),3:ncol(train)]
```

```
# #Stocks used:
# ^GSPC,AAPL,AMZN,BA,COST,CVX,DIS,ELY,FL,GILD,GS,HD,JCP,LMT,M,MMM,PG,S,SBUX,SHOO,SMRT,TM,YUM,K,C,XOM,IB
#
# #Data
# 2015-01-31 to 2019-12-31


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


Rfr <- seq(-0.05,.01,0.0005)

#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)

#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
```



In part a, we used Rf <- 0.002

d.

```
#CONSTANT CORRELATION MODEL - EXAMPLE:
#Read the data:
data1 <- rrr

#Compute the average correlation:
rho <- (sum(cor(data1[1:30]))-30)/(30*29)

#Initialize the vectors:
col1 <- rep(0,30)
col2 <- rep(0,30)
col3 <- rep(0,30)

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

#Compute necessary quantities:
Rbar <- colMeans(data1[1:30])
Rbar_f <- Rbar-0.0002
sigma <- ( diag(var(data1[1:30])) )^0.5
Ratio <- Rbar_f/sigma
```

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

#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(1:30)) {

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

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

#Compute the Ci:
for(i in (1:30)) {

          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)

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

final table for short sales allowed

```
aaaa
```

```
##              Rbar       Rbar_f      sigma       Ratio       col1      col2
## UNH    0.023671961  0.023471961 0.04801401  0.48885647 0.21968758 0.4888565
## NVDA   0.044567996  0.044367996 0.11977864  0.37041658 0.18011791 0.8592730
## MSFT   0.020687995  0.020487995 0.06025004  0.34004946 0.15262704 1.1993225
## TSM    0.018022831  0.017822831 0.06373683  0.27963160 0.13241668 1.4789541
```

```
## EQIX   0.015260709  0.015060709 0.05417001  0.27802670 0.11693282 1.7569808
## AMT    0.014264652  0.014064652 0.05110000  0.27523783 0.10469101 2.0322186
## ABT    0.015006148  0.014806148 0.05591581  0.26479360 0.09476949 2.2970122
## NFLX   0.034101355  0.033901355 0.13140208  0.25799709 0.08656571 2.5550093
## EL     0.013236073  0.013036073 0.05148673  0.25319287 0.07966909 2.8082022
## AMD    0.043437846  0.043237846 0.17348106  0.24923670 0.07379029 3.0574389
## AAPL   0.017597560  0.017397560 0.07184292  0.24216110 0.06871946 3.2996000
## CCI    0.011227455  0.011027455 0.04581229  0.24070951 0.06430075 3.5403095
## JNJ    0.009525905  0.009325905 0.03953700  0.23587789 0.06041596 3.7761874
## FB     0.014473364  0.014273364 0.06157717  0.23179636 0.05697383 4.0079838
## CHT    0.007410518  0.007210518 0.03261119  0.22110568 0.05390278 4.2290894
## ABBV   0.016591625  0.016391625 0.07644070  0.21443582 0.05114588 4.4435253
## PLD    0.011089250  0.010889250 0.05239503  0.20782981 0.04865726 4.6513551
## PEP    0.008565794  0.008365794 0.04031382  0.20751680 0.04639958 4.8588719
## MRK    0.010107381  0.009907381 0.04982115  0.19885894 0.04434212 5.0577308
## GOOG   0.011206214  0.011006214 0.05769731  0.19075785 0.04245939 5.2484887
## KO     0.007265480  0.007065480 0.03773016  0.18726348 0.04073001 5.4357522
## PSA    0.008176040  0.007976040 0.04882572  0.16335733 0.03913600 5.5991095
## PG     0.006624564  0.006424564 0.04004342  0.16043994 0.03766206 5.7595494
## WELL   0.009094911  0.008894911 0.05871015  0.15150552 0.03629511 5.9110549
## TSLA   0.016879280  0.016679280 0.11746897  0.14198881 0.03502392 6.0530438
## WMT    0.007404403  0.007204403 0.05415978  0.13302128 0.03383875 6.1860650
## AZN    0.008827518  0.008627518 0.06660160  0.12953921 0.03273117 6.3156042
## MDLZ   0.006095891  0.005895891 0.05044494  0.11687774 0.03169379 6.4324820
## CMCSA  0.006931851  0.006731851 0.05852146  0.11503216 0.03072016 6.5475141
## TWTR  -0.003696091 -0.003896091 0.14150259 -0.02753371 0.02980455 6.5199804
##            col3           z            x
## UNH   0.1073957  7.86131157  0.460330091
## NVDA  0.1547705  1.88404119  0.110322666
## MSFT  0.1830491  3.09960415  0.181501655
## TSM   0.1958382  1.71523296  0.100437864
## EQIX  0.2054487  1.98018715  0.115952628
## AMT   0.2127550  2.02921182  0.118823336
## ABT   0.2176867  1.61507179  0.094572787
## NFLX  0.2211762  0.62097995  0.036362349
## EL    0.2237269  1.46525657  0.085800148
## AMD   0.2256093  0.40564243  0.023752959
## AAPL  0.2267467  0.85330092  0.049966228
## CCI   0.2276445  1.29754161  0.075979363
## JNJ   0.2281420  1.34687652  0.078868238
## FB    0.2283502  0.77984790  0.045665084
## CHT   0.2279597  1.05240926  0.061625296
## ABBV  0.2272680  0.33715847  0.019742786
## PLD   0.2263222  0.33031340  0.019341963
## PEP   0.2254496  0.41935126  0.024555699
## MRK   0.2242705  0.11662267  0.006829003
## GOOG  0.2228476 -0.07923383 -0.004639648
## KO    0.2213983 -0.23985447 -0.014045014
## PSA   0.2191268 -0.81281730 -0.047595653
## PG    0.2169165 -1.08445135 -0.063501565
## WELL  0.2145424 -0.93467550 -0.054731231
## TSLA  0.2120013 -0.57096743 -0.033433797
## WMT   0.2093287 -1.45058181 -0.084940847
## AZN   0.2067171 -1.24660065 -0.072996444
```

```
## MDLZ  0.2038698 -1.96752632 -0.115211255
## CMCSA 0.2011407 -1.73640440 -0.101677588
## TWTR  0.1943251 -2.00929546 -0.117657105
```

final table for short sales not allowed

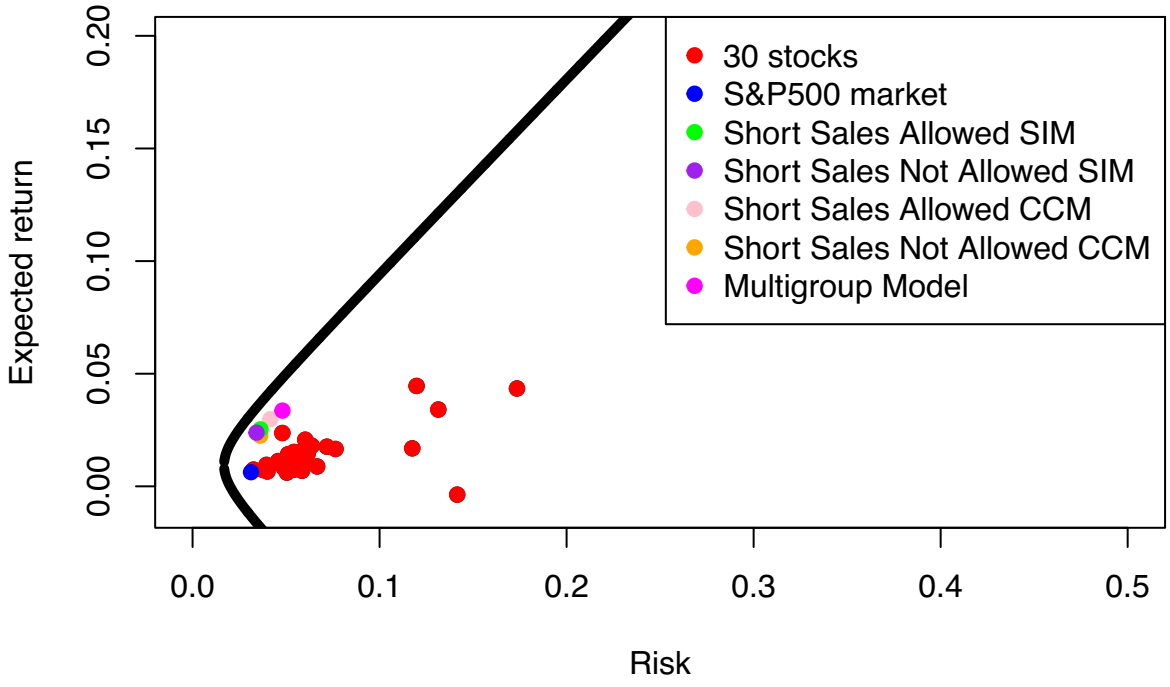
```
a_no
```

```
##             Rbar      Rbar_f      sigma     Ratio       col1      col2
## UNH  0.023671961 0.023471961 0.04801401 0.4888565 0.21968758 0.4888565
## NVDA 0.044567996 0.044367996 0.11977864 0.3704166 0.18011791 0.8592730
## MSFT 0.020687995 0.020487995 0.06025004 0.3400495 0.15262704 1.1993225
## TSM  0.018022831 0.017822831 0.06373683 0.2796316 0.13241668 1.4789541
## EQIX 0.015260709 0.015060709 0.05417001 0.2780267 0.11693282 1.7569808
## AMT  0.014264652 0.014064652 0.05110000 0.2752378 0.10469101 2.0322186
## ABT  0.015006148 0.014806148 0.05591581 0.2647936 0.09476949 2.2970122
## NFLX 0.034101355 0.033901355 0.13140208 0.2579971 0.08656571 2.5550093
## EL   0.013236073 0.013036073 0.05148673 0.2531929 0.07966909 2.8082022
## AMD  0.043437846 0.043237846 0.17348106 0.2492367 0.07379029 3.0574389
## AAPL 0.017597560 0.017397560 0.07184292 0.2421611 0.06871946 3.2996000
## CCI  0.011227455 0.011027455 0.04581229 0.2407095 0.06430075 3.5403095
## JNJ  0.009525905 0.009325905 0.03953700 0.2358779 0.06041596 3.7761874
## FB   0.014473364 0.014273364 0.06157717 0.2317964 0.05697383 4.0079838
##           col3         z          x       z_no        x_no
## UNH  0.1073957 7.8613116 0.46033009 6.95315146 0.408141927
## NVDA 0.1547705 1.8840412 0.11032267 1.51999958 0.089222212
## MSFT 0.1830491 3.0996041 0.18150166 2.37588003 0.139461402
## TSM  0.1958382 1.7152330 0.10043786 1.03110082 0.060524422
## EQIX 0.2054487 1.9801871 0.11595263 1.17523235 0.068984776
## AMT  0.2127550 2.0292118 0.11882334 1.17589650 0.069023761
## ABT  0.2176867 1.6150718 0.09457279 0.83524920 0.049028159
## NFLX 0.2211762 0.6209800 0.03636235 0.28914038 0.016972205
## EL   0.2237269 1.4652566 0.08580015 0.61835074 0.036296472
## AMD  0.2256093 0.4056424 0.02375296 0.15429274 0.009056805
## AAPL 0.2267467 0.8533009 0.04996623 0.24635994 0.014461043
## CCI  0.2276445 1.2975416 0.07597936 0.34573575 0.020294287
## JNJ  0.2281420 1.3468765 0.07886824 0.24400061 0.014322553
## FB   0.2283502 0.7798479 0.04566508 0.07172163 0.004209977
```

e.

```
#Var-covar matrix based on the constant correlation model:
for(i in 1:30){

    for(j in 1:30){

    if(i==j){
        mat[i,j]=aaaa[i,3]^2
        } else
                {
    mat[i,j]=rho*aaaa[i,3]*aaaa[j,3]
            }
            }
```

```
                }


#Calculate the expected return and sd of the point of tangency
#when short sales allowed
sd_p_opt <- (t(x) %*% mat %*% x)^.5
R_p_opt <- t(x) %*% aaaa[,1]


#Calculate the expected return and sd of the point of tangency
#when short sales are not allowed
sd_p_opt_no <- (t(x_no) %*% mat[1:which(aaaa[,7]==max(aaaa[,7])),1:which(aaaa[,7]==max(aaaa[,7]))] %*% x
R_p_opt_no <- t(x_no) %*% aaaaa[,1]


#Plot all the stocks and the two tangency points:
plot(aaaa[,3], aaaa[,1], xlim=c(0,0.5), ylim=c(-0.01,0.2), xlab="Risk", ylab="Expected return")

points(sdeff, y1, lwd=5,type = "l")
points(sdeff, y2, lwd=5,type = "l")
points(sd_p_opt,R_p_opt, col="pink", pch=19)

points(sd_p_opt_no,R_p_opt_no, col="orange", pch=19)

#30 stocks
points(sdev30,R_ibar, pch=19, col = "red")
points(sdev_m,R_m,pch=19,col="blue")
points(sdev_allowed_SIM, R_allowed_SIM,pch=19, col = "green")
points(sdev_not_allowed_SIM,R_not_allowed_SIM, pch =19, col="purple")
points(port_stdev,port_exp_R,pch=19,col="magenta")
legend("topright",c("30 stocks","S&P500 market","Short Sales Allowed SIM","Short Sales Not Allowed SIM"
```




```
#=====================================================================
#=====================================================================
#Note:  When short sales are allowed we can also find exactly the same #solution using Z=SIGMA^-1*R:
# rr <- aaa[,1]-.0002
# znew <- solve(mat) %*% rr
# xnew <- znew/sum(znew)

#=====================================================================
#=====================================================================
```

PART B 1.

```
index
```

```
##  [1]  0.0431170300  0.0069321656  0.0062007890  0.0210302800  0.0190583317
##  [6] -0.0150798306  0.0376552955 -0.0155138372  0.0232014608  0.0245335888
## [11] -0.0041885879 -0.0310408058  0.0548925110 -0.0173961069  0.0085208197
## [16]  0.0104913824 -0.0210116724  0.0197420297 -0.0625808182 -0.0264428316
## [21]  0.0829831178  0.0005048693 -0.0175301852 -0.0507353220 -0.0041283604
## [26]  0.0659911146  0.0026993985  0.0153246024  0.0009109211  0.0356098011
## [31] -0.0012192431 -0.0012344508 -0.0194256793  0.0341745222  0.0182007622
## [36]  0.0178843582  0.0371981603 -0.0003891972  0.0090912085  0.0115762514
## [41]  0.0048137751  0.0193488261  0.0005464328  0.0193029785  0.0221881353
## [46]  0.0280826277  0.0098316305  0.0561787044 -0.0389473721 -0.0268844986
## [51]  0.0027187751  0.0216083420  0.0048424360  0.0360215562  0.0302632115
## [56]  0.0042942871 -0.0694033560  0.0178593568 -0.0917768946
```

Here are the weights of ccm short sales allowed

```
x
```

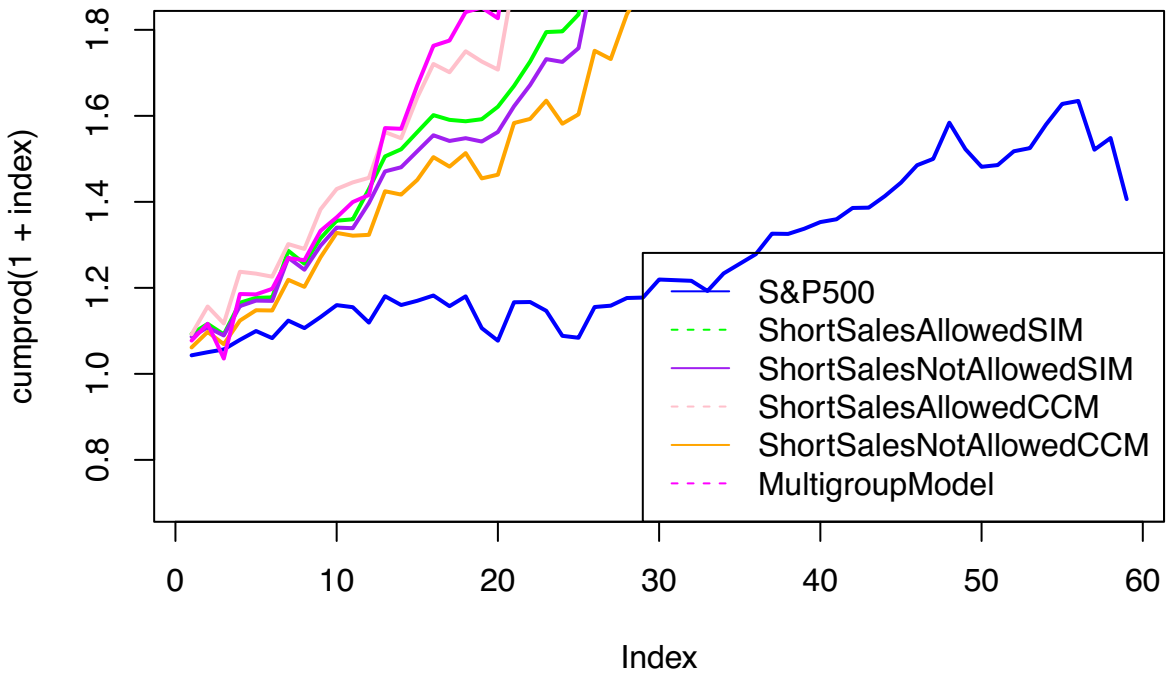
```
##          UNH         NVDA         MSFT          TSM         EQIX          AMT
##  0.460330091  0.110322666  0.181501655  0.100437864  0.115952628  0.118823336
##          ABT         NFLX           EL          AMD         AAPL          CCI
##  0.094572787  0.036362349  0.085800148  0.023752959  0.049966228  0.075979363
##          JNJ           FB          CHT         ABBV          PLD          PEP
##  0.078868238  0.045665084  0.061625296  0.019742786  0.019341963  0.024555699
##          MRK         GOOG           KO          PSA           PG         WELL
##  0.006829003 -0.004639648 -0.014045014 -0.047595653 -0.063501565 -0.054731231
##         TSLA          WMT          AZN         MDLZ        CMCSA         TWTR
## -0.033433797 -0.084940847 -0.072996444 -0.115211255 -0.101677588 -0.117657105
```

We need to reorder them

```
reordered_x_ccm_allowed = x[colnames(rr)]
reordered_x_ccm_allowed
```

```
##          TSM         AAPL         NVDA         TSLA         MSFT          AMD
##  0.100437864  0.049966228  0.110322666 -0.033433797  0.181501655  0.023752959
##         GOOG           FB         NFLX         TWTR          CHT        CMCSA
## -0.004639648  0.045665084  0.036362349 -0.117657105  0.061625296 -0.101677588
##          PLD          AMT          CCI          PSA         EQIX         WELL
##  0.019341963  0.118823336  0.075979363 -0.047595653  0.115952628 -0.054731231
##          JNJ          UNH          MRK          ABT         ABBV          AZN
##  0.078868238  0.460330091  0.006829003  0.094572787  0.019742786 -0.072996444
##          WMT           PG          PEP         MDLZ           KO           EL
## -0.084940847 -0.063501565  0.024555699 -0.115211255 -0.014045014  0.085800148
```

```
plot(cumprod(1+index), col="blue", lwd=2, type="l", ylim=c(0.7,1.8))
# points(cumprod(1+R_ibar), col="red", lwd=2, type="l")
r_allowed_sim <-  as.matrix(rr) %*% x_i_allowed
r_not_allowed_sim <- as.matrix(rr) %*% x_i
r_allowed_ccm <-  as.matrix(rr) %*% reordered_x_ccm_allowed
r_not_allowed_ccm <- as.matrix(rr[names(x_no)]) %*% x_no
r_multigroup <- as.matrix(rr) %*% X_alloc
points(cumprod(1+r_allowed_sim), col="green", lwd=2, type="l")
points(cumprod(1+r_not_allowed_sim), col="purple", lwd=2, type="l")
points(cumprod(1+r_allowed_ccm), col="pink", lwd=2, type="l")
points(cumprod(1+r_not_allowed_ccm), col="orange", lwd=2, type="l")
points(cumprod(1+r_multigroup),col="magenta", lwd=2,type ="l")
legend('bottomright', lty=1:2, c('S&P500','ShortSalesAllowedSIM','ShortSalesNotAllowedSIM','ShortSalesA
```



PART B 2.

```
comp <- cumprod(1+index)
geoMean_index <- comp[length(comp)]^(1/length(comp)) - 1
geoMean_index
```

```
## [1] 0.005795708
```

30 stocks in not a portfolio

```
# comp <- cumprod(1+R_ibar)
# geoMean_30stocks <- comp[length(comp)]^(1/length(comp)) - 1
# geoMean_30stocks
```

```
comp <- cumprod(1+r_allowed_sim)
geoMean_allowed_sim <- comp[length(comp)]^(1/length(comp)) - 1
geoMean_allowed_sim
```

```
## [1] 0.02460446
```

```
comp <- cumprod(1+r_not_allowed_sim)
geoMean_not_allowed_sim <- comp[length(comp)]^(1/length(comp)) - 1
geoMean_not_allowed_sim
```

```
## [1] 0.02321741
```

```
comp <- cumprod(1+r_allowed_ccm)
geoMean_allowed_ccm <- comp[length(comp)]^(1/length(comp)) - 1
geoMean_allowed_ccm
```

```
## [1] 0.02849642
```

```
comp <- cumprod(1+r_not_allowed_ccm)
geoMean_not_allowed_ccm <- comp[length(comp)]^(1/length(comp)) - 1
geoMean_not_allowed_ccm
```

```
## [1] 0.02182881
```

```
comp <- cumprod(1+r_multigroup)
geoMean_multigroup <- comp[length(comp)]^(1/length(comp)) - 1
geoMean_multigroup
```

```
## [1] 0.03181646
```

PART B 3. Sharpe Ratio of each Portfolio

For SIM when Short Sales are Allowed

```
(mean(r_allowed_sim) - Rf) / sdev_allowed_SIM
```

```
##           [,1]
## [1,] 0.6410813
```

For SIM when Short Sales not Allowed

```
(mean(r_not_allowed_sim) - Rf) / sdev_not_allowed_SIM
```

```
##           [,1]
## [1,] 0.6382121
```

For CCM with Short Sales Allowed

```
(t(reordered_x_ccm_allowed) %*% colMeans(rr) - Rf) / sd_p_opt
```

```
##           [,1]
## [1,] 0.6680151
```

```
(mean(r_allowed_ccm)-Rf)/sd_p_opt
```

```
##           [,1]
## [1,] 0.6680151
```

For CCM without Short Sales Allowed

```
(mean(r_not_allowed_ccm) - Rf) / sd_p_opt_no
```

```
##           [,1]
## [1,] 0.5669022
```

For Multigroup Model

```
(mean(r_multigroup) - Rf) / port_stdev
```

```
## [1] 0.6579066
```

Here are Rm_bar and sigma_m

```
R_m = mean(index)
s_m2 <- var(index)
```

Differential Excess Return R_a_bar - R_a'_bar For SIM when Short Sales are Allowed

```
# mean(r_allowed_sim) - (Rf + (R_m-Rf)/s_m2*sdev_allowed_SIM)
mean(r_allowed_sim) - Rf - ((R_m-Rf)/sdev_m) * sdev_allowed_SIM
```

```
##           [,1]
## [1,] 0.01832064
```

For SIM when Short Sales not Allowed

```
# mean(r_not_allowed_sim) - (Rf + (R_m-Rf)/s_m2*sdev_not_allowed_SIM)
mean(r_not_allowed_sim) - Rf - ((R_m-Rf)/sdev_m) * sdev_not_allowed_SIM
```

```
##           [,1]
## [1,] 0.01714445
```

For CCM with Short Sales Allowed

```
# (t(reordered_x_ccm_allowed) %*% colMeans(rr)) - (Rf +  (R_m-Rf)/s_m2* sd_p_opt)
# mean(r_allowed_ccm)-(Rf + (R_m-Rf)/s_m2*sd_p_opt)
mean(r_allowed_ccm) - Rf - ((R_m-Rf)/sdev_m) * sd_p_opt
```

```
##           [,1]
## [1,] 0.02211854
```

For CCM without Short Sales Allowed

```
# mean(r_not_allowed_ccm) - (Rf + (R_m-Rf)/s_m2* sd_p_opt_no)
mean(r_not_allowed_ccm) - Rf - ((R_m-Rf)/sdev_m) * sd_p_opt_no
```

```
##           [,1]
## [1,] 0.01556331
```

For Multigroup Model

```
# mean(r_multigroup) - (Rf + (R_m-Rf)/s_m2*port_stdev)
mean(r_multigroup) - Rf - ((R_m-Rf)/sdev_m) * port_stdev
```

```
## [1] 0.0250444
```

Treynor Measure SIM Short Sales Allowed

```
# (R_p_opt - Rf) / (t(beta)%*%x_i_allowed)
(mean(r_allowed_sim) - Rf) / (t(beta)%*%x_i_allowed)
```

```
##            [,1]
## [1,] 0.03495135
```

SIM Short Sales Not Allowed

```
(mean(r_not_allowed_sim)-Rf)/ (t(beta)%*%x_i_allowed)
```

```
##            [,1]
## [1,] 0.03274744
```

Jensen Differential Performance Index SIM Short Sales Allowed

```
# R_p_opt - Rf - (R_m - Rf) * (t(beta)%*%x_i_allowed)
mean(r_allowed_sim) - Rf - (R_m - Rf) * (t(beta)%*%x_i_allowed)
```

```
##            [,1]
## [1,] 0.02044384
```

SIM Short Sales Not Allowed

```
# R_p_opt_no - Rf - (R_m-Rf)*(t(beta)%*%x_i)
mean(r_not_allowed_sim) - Rf - (R_m-Rf)*(t(beta)%*%x_i)
```

```
##            [,1]
## [1,] 0.01882078
```

Part B 4.Fama's decomposition (net selectivity and diversification) for single index model when short sales are not allowed

```
R_a = mean(r_not_allowed_sim)
# R_a = R_p_opt_no
beta_a_double_prime = (sdev_not_allowed_SIM / sigma_m2^.5)
R_a_double_prime = Rf + ((R_m-Rf)/1)*beta_a_double_prime
R_from_net_selectivity = R_a - R_a_double_prime
R_from_net_selectivity
```

```
##            [,1]
## [1,] 0.01714445
```

```r
# beta_a = mean(beta)
beta_a = (t(beta)%*%x_i)
R_a_prime = Rf + ((R_m-Rf)/1)*beta_a
R_a_double_prime - R_a_prime
```

```
##              [,1]
## [1,] 0.001676325
```

```r
plot(c(1,beta_a,beta_a_double_prime,beta_a), c(R_m,R_a_prime,R_a_double_prime,R_p_opt_no),xlab = "beta"
abline(lm(c(R_a_double_prime,Rf)~c(beta_a_double_prime,0)))
```