This case study was prepared to the same extent by Luca Bruno and Simon Gühring

CaseStudy_3-1.R

guehrings

2024-01-05

```
library(tseries)

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

library(zoo)

## Warning: Paket 'zoo' wurde unter R Version 4.3.2 erstellt

##
## Attache Paket: 'zoo'

## Die folgenden Objekte sind maskiert von 'package:base':
##
##     as.Date, as.Date.numeric

library(moments)
library(quantmod)

## Lade nötiges Paket: xts

## Lade nötiges Paket: TTR

##Download stock prices for at least 100 stocks for the year 2022.

SPShares <- read.table("C:/Users/guehrings/Downloads/SP500Ticker (1).csv",
sep=";", header=TRUE)
SPTickers <- SPShares$Symbol
SPTickers

##   [1] "MMM"    "AOS"    "ABT"    "ABBV"   "ACN"    "ATVI"   "ADM"    "ADBE"
"ADP"
##  [10] "AAP"    "AES"    "AFL"    "A"      "APD"    "AKAM"   "ALK"    "ALB"
"ARE"
##  [19] "ALGN"   "ALLE"   "LNT"    "ALL"    "GOOGL"  "GOOG"   "MO"     "AMZN"
"AMCR"
##  [28] "AMD"    "AEE"    "AAL"    "AEP"    "AXP"    "AIG"    "AMT"    "AWK"
```

```
"AMP"
## [37] "ABC"    "AME"    "AMGN"   "APH"    "ADI"    "ANSS"   "AON"    "APA"
"AAPL"
## [46] "AMAT"   "APTV"   "ACGL"   "ANET"   "AJG"    "AIZ"    "T"      "ATO"

…

## Get 100 Stocks Tickers out of S&P
HundredTickers_SP = SPTickers[seq(5, length(SPTickers), by = 5)]
length(HundredTickers_SP)

## [1] 100

##Get the Prices

getPrices <- function(TickerSymbols,start,end,type){
  NumberOfStocks <- length(TickerSymbols)
  prices <- get.hist.quote(TickerSymbols[1],start=start,end=end,quote=type)
  goodSymbols <- TickerSymbols[1]
  for (d in 2:NumberOfStocks) {
    tryCatch({
      P <- get.hist.quote(TickerSymbols[d],start=start,end=end,quote=type)
      prices <- cbind(prices,P)
      goodSymbols <- c(goodSymbols,TickerSymbols[d])
    }, error=function(err) { print(paste("Download ERROR: ",
TickerSymbols[d])) } )
  }
  prices <- data.frame(coredata(prices))
  colnames(prices) <- goodSymbols
  NumberOfGoodStocks <- dim(prices)[2]
  T <- dim(prices)[1]
  badSymbols <- rep(FALSE,NumberOfGoodStocks)
  for (d in 1:NumberOfGoodStocks) {
    if (is.na(prices[1,d]) || is.na(prices[T,d])) {
      badSymbols[d] <- TRUE
    } else {
      if ( sum(is.na(prices[,d]))>0) {
        print(paste(goodSymbols[d]," NAs filled: ", sum(is.na(prices[,d]))))
        prices[,d]<-na.approx(prices[,d])
      }
    }
  }
  if (sum(badSymbols)>0){
    prices <- prices[!badSymbols]
    print(paste("Removed due to NAs: ", goodSymbols[badSymbols]))
  }
  if (sum(is.na(prices))==0 ) {
    if (sum(prices == 0) > 0) {print("Check Zeros!")}
  } else {print("Check NAs and Zeros")}
  prices
}
```

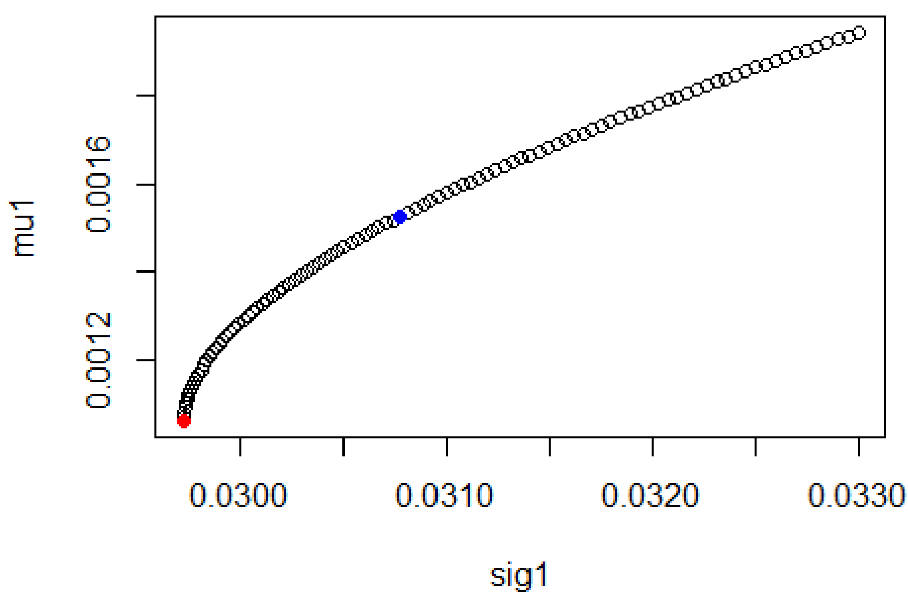
```
HundredSP_Prices <- getPrices(HundredTickers_SP, start="2022-01-01",
end="2022-12-31", type="Adj")

## time series starts 2022-01-03
## time series ends   2022-12-30
## time series starts 2022-01-03
## time series ends   2022-12-30
…

## Calculate the Returns

getReturns <- function(prices) {
  NumberOfStocks <- dim(prices)[2]
  T <- dim(prices)[1]
  returns <- c()
  for (ind in 1:NumberOfStocks) {
    returns <- cbind(returns, diff(log(prices[,ind])))
  }
  returns
}

HundredSP_Returns <- data.frame(getReturns(HundredSP_Prices))

names(HundredSP_Returns) <- names(HundredSP_Prices)
HundredSP_Returns

##               ACN           AAP          AKAM          ALLE            MO
## 1   -7.171901e-03  1.139713e-03 -0.0047769876  0.0232285300  0.0218565473
## 2   -1.776756e-02 -2.534439e-03 -0.0126490831 -0.0236150895 -0.0079862866
## 3   -4.949533e-02  2.175364e-02 -0.0316705401 -0.0096355432  0.0116506022
## 4   -1.936628e-02 -1.484155e-02 -0.0120493891 -0.0113076846  0.0113156188
## 5    6.050340e-03 -1.677215e-02  0.0189082257 -0.0122359854  0.0153525945
## 6    5.641050e-03 -9.096219e-03  0.0068120460  0.0199453121 -0.0211963840
## 7    8.118986e-03  2.351342e-02  0.0060651744 -0.0002350617  0.0002020764
## 8   -4.284686e-02  1.583003e-02 -0.0105718272 -0.0115098280  0.0164330947
…

## Calculate the covariance matrix and the correlation matrix

Correlation_HundredSP_Returns <- cor(HundredSP_Returns)
Correlation_HundredSP_Returns

##            ACN       AAP      AKAM      ALLE           MO        AAL
AWK
## ACN  1.0000000 0.4617644 0.5973131 0.6946401  0.236488146 0.54237081
0.6211141
## AAP  0.4617644 1.0000000 0.3229661 0.4906257  0.205208137 0.39097459
0.3472652
## AKAM 0.5973131 0.3229661 1.0000000 0.5538290  0.116641124 0.37947142
0.5219272
## ALLE 0.6946401 0.4906257 0.5538290 1.0000000  0.255683417 0.49111930
```

```r
Covariance_HundredSP_Returns <- cov(HundredSP_Returns)
Covariance_HundredSP_Returns

##               ACN          AAP         AKAM         ALLE            MO
## ACN  4.376362e-04 2.323310e-04 2.371014e-04 2.933837e-04  7.898094e-05
## AAP  2.323310e-04 5.784426e-04 1.473882e-04 2.382319e-04  7.879180e-05
##

## Find the pair of stocks with the highest correlation

Max_Position <- which(Correlation_HundredSP_Returns ==
max(Correlation_HundredSP_Returns[Correlation_HundredSP_Returns < 1]),
arr.ind = TRUE)
Max_Position

##      row col
## MRO   60  30
## FANG  30  60

HundredTickers_SP[30]

## [1] "FANG"

HundredTickers_SP[60]

## [1] "MRO"

Correlation_HundredSP_Returns[30,60]

## [1] 0.9038399

## FANG and MRO is the pair of stocks with the highest correlation.

## CHECK!
FANG <- get.hist.quote("FANG", start="2022-01-01", end="2022-12-31", "Adj")

## time series starts 2022-01-03
## time series ends   2022-12-30

MRO <- get.hist.quote("MRO", start="2022-01-01", end="2022-12-31", "Adj")

## time series starts 2022-01-03
## time series ends   2022-12-30

Matrix <- cbind(diff(log(FANG)),diff(log(MRO)))
cor(Matrix)

##                           Adjusted.diff(log(FANG)) Adjusted.diff(log(MRO))
## Adjusted.diff(log(FANG))                1.0000000               0.9038398
## Adjusted.diff(log(MRO))                 0.9038398               1.0000000

##Find the pair of stocks with the lowest correlation
Min_Position <- which(Correlation_HundredSP_Returns ==
```

```r
min(Correlation_HundredSP_Returns), arr.ind = TRUE)
Min_Position

##      row col
## VTRS  95  59
## LMT   59  95

HundredTickers_SP[95]

## [1] "VTRS"

HundredTickers_SP[59]

## [1] "LMT"

Correlation_HundredSP_Returns[95,59]

## [1] -0.1005962

##VTRS and LMT is the pair of stock with the lowest correlation

## Compare the stand alone Risk and return ("lowest correlation")
SD_VTRS <- sd(HundredSP_Returns[,95])
SD_LMT <- sd(HundredSP_Returns[,59])
SD_VTRS

## [1] 0.02804002

SD_LMT

## [1] 0.01661673

MeanRet_VTRS <- mean(HundredSP_Returns[,95])
MeanRet_LMT <- mean(HundredSP_Returns[,59])
MeanRet_VTRS

## [1] -0.0008130285

MeanRet_LMT

## [1] 0.001371292

data.frame(Stock= c("VTRS", "LMT"), Volatility=c(SD_VTRS, SD_LMT),
Mean=c(MeanRet_VTRS,MeanRet_LMT))

##   Stock Volatility          Mean
## 1  VTRS 0.02804002 -0.0008130285
## 2   LMT 0.01661673  0.0013712916

## Compare the stand alone Risk and return  (highest correlation)
SD_FANG <- sd(HundredSP_Returns[,30])
SD_MRO <- sd(HundredSP_Returns[,60])
SD_FANG

## [1] 0.02972532
```

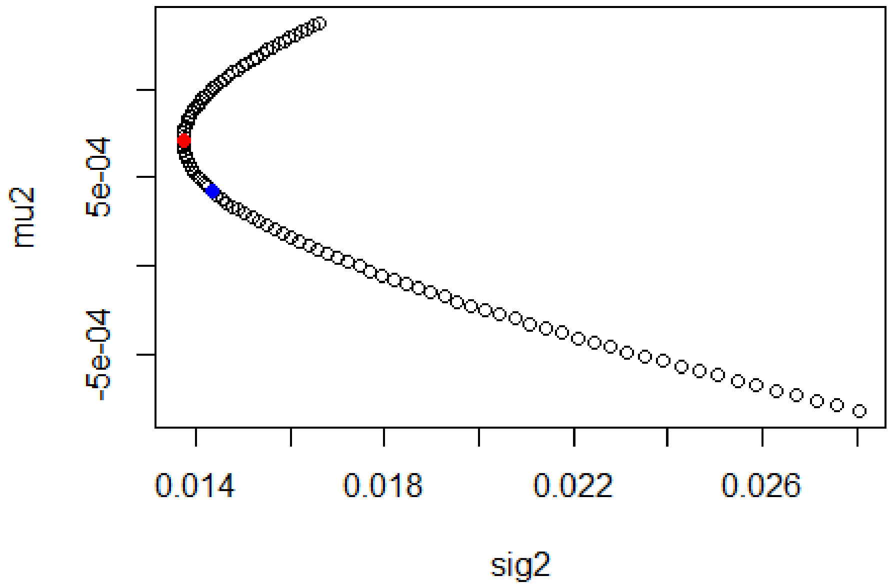
```r
SD_MRO

## [1] 0.03299777

MeanRet_FANG <- mean(HundredSP_Returns[,30])
MeanRet_MRO <- mean(HundredSP_Returns[,60])
MeanRet_FANG

## [1] 0.001065097

MeanRet_MRO

## [1] 0.00194103

data.frame(Stock= c("FANG", "MRO"), Volatility=c(SD_FANG, SD_MRO),
Mean=c(MeanRet_FANG,MeanRet_MRO))

##   Stock Volatility        Mean
## 1  FANG 0.02972532 0.001065097
## 2   MRO 0.03299777 0.001941030

##Use the first pair of stocks to built portfolios with different
combinations of weights.

PortfolioReturns <- function(StockReturns, weights) {
  if (sum(weights)==1 && length(weights)==dim(StockReturns)[2]) {
    NumStocks <- dim(StockReturns)[2]
    Length <- dim(StockReturns)[1]
    P <- rep(0,Length)
    for (t in 1:Length) {
      for (d in 1:NumStocks) {
        P[t] <- P[t] + weights[d]*exp(sum(StockReturns[1:t,d]))
      }
    }
    P <- c(1,P)
    diff(log(P))
  } else {print("Error: weights do not match")}
}

TwoReturns1 <- data.frame(HundredSP_Returns[30],HundredSP_Returns[60])

Means_Pair1 <- c(MeanRet_FANG,MeanRet_MRO)
Volas_Pair1 <- c(SD_FANG, SD_MRO)

sig1 <- c()
mu1 <- c()

for (i in 0:100){
  PRet <- PortfolioReturns(TwoReturns1,c(i/100,1-i/100))
  mu1 <- c(mu1,mean(PRet))
  sig1 <- c(sig1,sd(PRet))
}
```

```r
plot(sig1,mu1)

##Where to find the Equal Weighted Portfolio?

Equal_Weighted <- PortfolioReturns(TwoReturns1, c(0.5,0.5))
points(sd(Equal_Weighted),mean(Equal_Weighted), col="blue", pch=19)
sd(Equal_Weighted)

## [1] 0.03078003

##Find the Portfolio with the smallest Volatility

MinVola1 <- min(sig1)
MinVola1

## [1] 0.02972532

MinVolaIndex1 <- which(sig1 == MinVola1)

points(sig1[MinVolaIndex1], mu1[MinVolaIndex1], col = "red", pch = 19)
```




```r
##Check if the Minimum Volatility equals the Volatility of the stock FANG
MinVola1 == SD_FANG

## [1] TRUE

weights_MinVola1 <- c((MinVolaIndex1 - 1)/100, 1 - (MinVolaIndex1 - 1)/100)
weights_MinVola1
```

```
## [1] 1 0

##The Portfolio with the smallest volatility is therefore the following: 100%
FANG, 0% MRO

##Repeat with the second pair of stocks

TwoReturns2 <- data.frame(HundredSP_Returns[95],HundredSP_Returns[59])

Means_Pair2 <- c(MeanRet_VTRS,MeanRet_LMT)
Volas_Pair2 <- c(SD_VTRS, SD_LMT)

sig2 <- c()
mu2 <- c()

for (i in 0:100){
  PRet <- PortfolioReturns(TwoReturns2,c(i/100,1-i/100))
  mu2 <- c(mu2,mean(PRet))
  sig2 <- c(sig2,sd(PRet))
}

plot(sig2,mu2)

##Where to find the Equal Weighted Portfolio?

Equal_Weighted2 <- PortfolioReturns(TwoReturns2, c(0.5,0.5))
points(sd(Equal_Weighted2),mean(Equal_Weighted2), col="blue", pch=19)
sd(Equal_Weighted2)

## [1] 0.01434945

## Find the Portfolio with the smallest Volatility
MinVola2 <- min(sig2)
MinVola2

## [1] 0.01373065

MinVolaIndex2 <- which(sig2 == MinVola2)

points(sig2[MinVolaIndex2], mu2[MinVolaIndex2], col = "red", pch = 19)
```

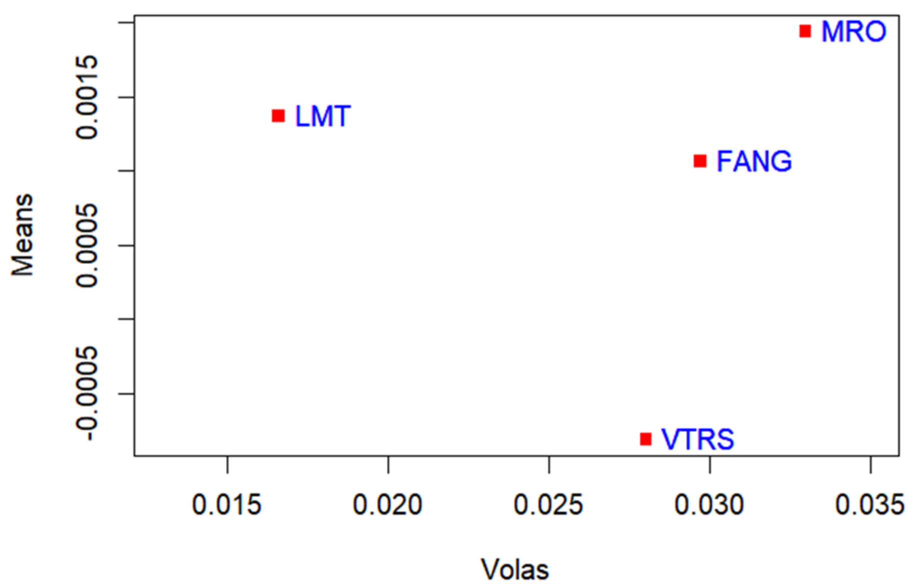



```
weights_MinVola2 <- c((MinVolaIndex2 - 1)/100, 1 - (MinVolaIndex2 - 1)/100)
weights_MinVola2

## [1] 0.36 0.64

##The portfolio with the smallest volatility is the one with the following
weights: 64% LMT and 36% VTRS.

##Compare the results

data.frame( StockPair = c("LMT and VTRS", "FANG and MRO"),
            Weights = c(paste0(weights_MinVola2[2], " and ",
weights_MinVola2[1]), paste0(weights_MinVola1[1], " and ",
weights_MinVola1[2])))

##      StockPair       Weights
## 1 LMT and VTRS 0.64 and 0.36
## 2 FANG and MRO       1 and 0

data.frame(Case= c("Highest Correlation", "Lowest
Correlation"),Volatility=c(MinVola1, MinVola2))

##                  Case Volatility
## 1 Highest Correlation 0.02972532
## 2  Lowest Correlation 0.01373065

##The Minimum Volatility of the portfolio with the HIGHEST correlation is
higher than
##the Minimum Volatility of the portfolio with the LOWEST correlation.
```

```r
##If the correlation between a pair of stocks is high, it means that they
tend to move together, and the benefits of diversification in terms of risk
reduction are limited.
##And as diversification is limited as they tend to move together, it also
chose 100% of one stock (FANG) and 0% of another (MRO).
##Conversely, if the correlation is low or negative, the assets exhibit more
independent price movements, potentially leading to a portfolio with lower
volatility.


##Now combine the four stocks and calculate mean return and volatility of an
equal weighted portfolio.

Returns_Four_Stocks <-
data.frame(HundredSP_Returns[30],HundredSP_Returns[60],
HundredSP_Returns[95],HundredSP_Returns[59])
Returns_Four_Stocks

##              FANG           MRO          VTRS           LMT
## 1    0.0637334688  0.0429301620  0.0111966127  2.130329e-02
## 2   -0.0091863136 -0.0229763690  0.0048594295 -1.069254e-02
## 3    0.0458384306  0.0465464588  0.0041466153 -3.912422e-04
## 4   -0.0173760439  0.0115799354  0.0164163438  5.960169e-03
## 5   -0.0013999605 -0.0277951052  0.0181516316  8.598224e-03


Portfolio_Returns <- PortfolioReturns(Returns_Four_Stocks,
c(0.25,0.25,0.25,0.25))
Mean_Portfolio <- mean(Portfolio_Returns)
Mean_Portfolio

## [1] 0.001014438

Volatility_Portfolio <- sd(Portfolio_Returns)
Volatility_Portfolio

## [1] 0.02078818

##Compare with the portfolios above

Mean_of_Minimum_Volatility1 <- mu1[MinVolaIndex1]


Mean_of_Minimum_volatility2 <- mu2[MinVolaIndex2]

data.frame(Case= c("Highest Correlation Only", "Lowest Correlation Only",
"Equal Weighted Portfolio"),Volatility=c(MinVola1, MinVola2,
Volatility_Portfolio),
Mean=c(Mean_of_Minimum_Volatility1,Mean_of_Minimum_volatility2,Mean_Portfolio
))

##                         Case Volatility         Mean
## 1 Highest Correlation Only 0.02972532 0.0010650973
```

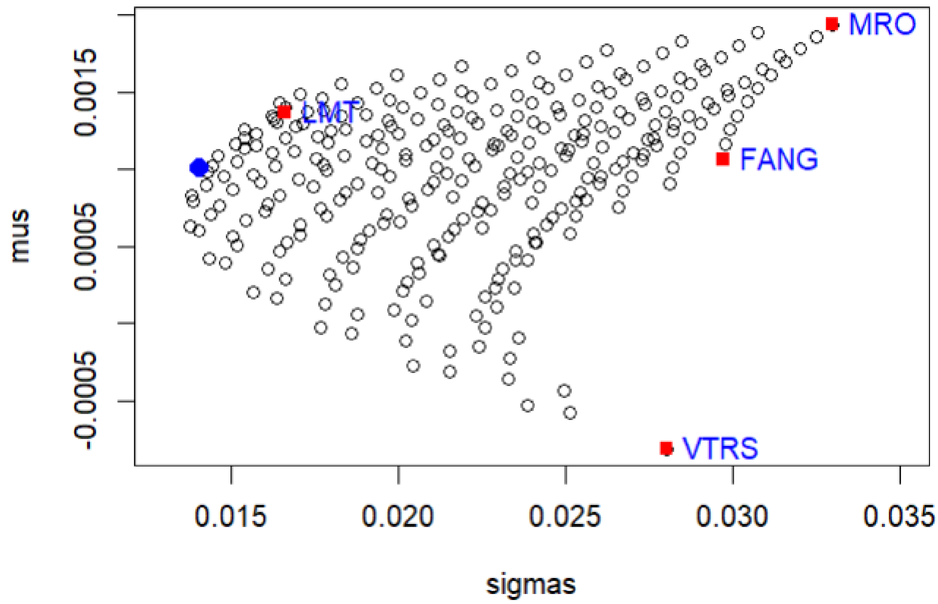
```
## 2  Lowest Correlation Only 0.01373065 0.0007142344
## 3 Equal Weighted Portfolio 0.02078818 0.0010144381
```

```r
##Equal Weighted Portfolio has a lower volatility (Risk) than the one with the Highest Correlation only and still a very good Mean Return, that is near to the Mean Return of Highest Correlation Only.
##In other words: The Equal Weighted Portfolio has the best Mean to Volatility Ratio


## Try to minimize portfolio volatility for the portfolio of four stocks.

Means <- c()
Volas <- c()
for (i in 1:4){
  Means <- c(Means, mean(Returns_Four_Stocks[,i]))
  Volas <- c(Volas,sd(Returns_Four_Stocks[,i]))
}

plot(Volas,Means,col="red",xlim=c(0.013,0.035), pch=15)
text(Volas, Means, labels = c("FANG", "MRO", "VTRS", "LMT"), pos = 4, col = "blue")
```




```r
mus <- c()
sigmas <- c()


for (i in seq(0,1,0.1)){
```

```r
  for (j in seq(0,1-i,0.1)){
    for (k in seq(0,1-i-j,0.1)){
      PRet <- PortfolioReturns(Returns_Four_Stocks,c(i,j,k,max(1-(i+j+k),0)))
      mus <- c(mus,mean(PRet)); sigmas <- c(sigmas,sd(PRet))
    } } }

plot(sigmas,mus, xlim=c(0.013,0.035))
points(Volas,Means,col="red",pch=15)
text(Volas, Means, labels = c("FANG", "MRO", "VTRS", "LMT"), pos = 4, col =
"blue")

library(quadprog)

FindWeightsForMu <- function(Mu,CovMat,Means){
  N <- dim(CovMat)[1]
  AMat <- cbind(rep(1,N),Means,diag(1,nrow=N))
  bVec = c(1,Mu,rep(0,N))
  result = solve.QP(2*CovMat,rep(0,N),AMat,bVec,2)
  result
}

COV <- cov(Returns_Four_Stocks)

FindWeightsForMu(Mean_Portfolio,COV,Means)

## $solution
## [1] 0.00000000 0.05763983 0.17840480 0.76395537
##
## $value
## [1] 0.0001974833
##
## $unconstrained.solution
## [1] 0 0 0 0
##
## $iterations
## [1] 4 0
##
## $Lagrangian
## [1] 3.011745e-04 9.245727e-02 2.584535e-05 0.000000e+00 0.000000e+00
## [6] 0.000000e+00
##
## $iact
## [1] 1 2 3

points(sqrt(FindWeightsForMu(Mean_Portfolio,COV,Means)$value),Mean_Portfolio,
col="blue",pch=16,cex=1.5)
```




```
Solution <- FindWeightsForMu(Mean_Portfolio,COV,Means)
Solution$solution

## [1] 0.00000000 0.05763983 0.17840480 0.76395537

##With the above weight, the variance is minimal.



###
### PART 2
##Calculate mean return and volatility for an equal weighted portfolio of all
downloaded stocks.

PortfolioRet_All_Stocks <- PortfolioReturns(HundredSP_Returns,
rep(0.01,times=100))
Mean_Portfolio_All_Stocks <- mean(PortfolioRet_All_Stocks)
Mean_Portfolio_All_Stocks

## [1] -0.0002648815

PortfolioVola_All_Stocks <- sd(PortfolioRet_All_Stocks)
PortfolioVola_All_Stocks

## [1] 0.01396972

##Compare with the results of part 1

data.frame(Case= c("Highest Corr Equal Weighted Portfolio", "Lowest Corr
Equal Weighted Portfolio", "Portfolio of Four Stocks", "Portfolio of 100
```

```
Stocks"),Volatility=c(sd(Equal_Weighted) ,
sd(Equal_Weighted2),Volatility_Portfolio, PortfolioVola_All_Stocks),
Mean=c(mean(Equal_Weighted), mean(Equal_Weighted2), Mean_Portfolio,
Mean_Portfolio_All_Stocks))

##                                     Case Volatility          Mean
## 1 Highest Corr Equal Weighted Portfolio 0.03078003  0.0015269928
## 2  Lowest Corr Equal Weighted Portfolio 0.01434945  0.0004264167
## 3                Portfolio of Four Stocks 0.02078818  0.0010144381
## 4                 Portfolio of 100 Stocks 0.01396972 -0.0002648815

##Volatility is lower with 100 Stocks as it is more diversified; Even more
than the Equal weighted Portfolio with the two stocks with the lowest
correlation


## As a general rule portfolio volatility of a large equal weighted portfolio
is close to the root of the average covariance of its constituents
##Check this relation for the equal weighted portfolio of all downloaded
stocks
COV <- cov(HundredSP_Returns)
COV

##               ACN          AAP         AKAM          ALLE            MO
## ACN  4.376362e-04 2.323310e-04 2.371014e-04 2.933837e-04  7.898094e-05
## AAP  2.323310e-04 5.784426e-04 1.473882e-04 2.382319e-04  7.879180e-05
## AKAM 2.371014e-04 1.473882e-04 3.600402e-04 2.121635e-04  3.533325e-05
## ALLE 2.933837e-04 2.382319e-04 2.121635e-04 4.076046e-04  8.240971e-05
## MO   7.898094e-05 7.879180e-05 3.533325e-05 8.240971e-05  2.548664e-04
## GWW  1.483921e-04 1.375650e-04 1.850172e-04 3.214729e-04 2.448778e-04
## ZBRA 2.598378e-04 1.106446e-04 3.290882e-04 2.448778e-04 8.775403e-04

…

avg_covariance <- (mean(COV[upper.tri(COV)]))^(1/2)
avg_covariance

## [1] 0.0143814

## The use of the upper.tri function is essential to extract the upper
triangular part of the covariance matrix. This is necessary because the
matrix contains duplicated variances and includes the diagonal line, which
represents the covariance of one stock with itself.

data.frame(Case= c("Portfolio Volatility", "Estimator Covariance"), Value=
c(PortfolioVola_All_Stocks, avg_covariance))

##                   Case      Value
## 1 Portfolio Volatility 0.01396972
## 2 Estimator Covariance 0.01438140

##Repeat part 2 for the year 2021
```

```
SP_Prices2021 <- getPrices(HundredTickers_SP, start="2021-01-01", end="2021-
12-31", type="Adj")

## time series starts 2021-01-04
## time series ends   2021-12-30


SP_Returns2021 <- getReturns(SP_Prices2021)

PortfolioRet2021 <- PortfolioReturns(SP_Returns2021, rep(0.01,times=100))
Mean_Portfolio2021 <- mean(PortfolioRet2021)
Mean_Portfolio2021

## [1] 0.001254293

PortfolioVola2021 <- sd(PortfolioRet2021)
PortfolioVola2021

## [1] 0.009173637

COV21 <- cov(SP_Returns2021)
COV21

##                  [,1]         [,2]          [,3]          [,4]
[,5]
##   [1,]  1.524632e-04 6.980169e-05  5.335897e-05  7.980420e-05  5.630731e-
05
##   [2,]  6.980169e-05 2.668643e-04  1.949372e-05  7.845309e-05  7.684162e-
05
##   [3,]  5.335897e-05 1.949372e-05  2.651781e-04  5.973737e-05  2.844404e-
05
##   [4,]  7.980420e-05 7.845309e-05  5.973737e-05  2.124795e-04  6.468370e-
05
##   [5,]  5.630731e-05 7.684162e-05  2.844404e-05  6.468370e-05  1.812570e-

…

avg_covariance21 <- (mean(COV21[upper.tri(COV21)]))^(1/2)
avg_covariance21

## [1] 0.008600109

data.frame(Case= c("Portfolio Volatility", "Estimator Covariance"), Value=
c(PortfolioVola2021, avg_covariance21))

##                    Case       Value
## 1 Portfolio Volatility 0.009173637
## 2 Estimator Covariance 0.008600109

##And for the year 2020

SP_Prices2020 <- getPrices(HundredTickers_SP, start="2020-01-01", end="2020-
12-31", type="Adj")
```

```
## time series starts 2020-01-02
## time series ends   2020-12-30
## time series starts 2020-01-02


SP_Returns2020 <- getReturns(SP_Prices2020)

PortfolioRet2020 <- PortfolioReturns(SP_Returns2020, rep(0.01,times=100))
Mean_Portfolio2020 <- mean(PortfolioRet2020)
Mean_Portfolio2020

## [1] 0.0006419715

PortfolioVola2020 <- sd(PortfolioRet2020)
PortfolioVola2020

## [1] 0.02296594

COV20 <- cov(SP_Returns2020)
COV20

##                [,1]         [,2]          [,3]         [,4]         [,5]
##   [1,] 0.0006874362 0.0004560028  3.037809e-04 0.0005447751 3.408584e-04
##   [2,] 0.0004560028 0.0008922744  2.156688e-04 0.0005742801 3.681927e-04
##   [3,] 0.0003037809 0.0002156688  5.811859e-04 0.0002661739 1.891566e-04
##   [4,] 0.0005447751 0.0005742801  2.661739e-04 0.0008444493 4.209408e-04
##   [5,] 0.0003408584 0.0003681927  1.891566e-04 0.0004209408 5.304618e-04
##   [6,] 0.0006348969 0.0007586608  1.055607e-04 0.0008370975 7.031500e-04
##   [7,] 0.0003935501 0.0004860291  3.143446e-04 0.0005045385 3.597892e-04
##   [8,] 0.0005273172 0.0005154806  2.514380e-04 0.0005679669 3.547365e-04


avg_covariance20 <- (mean(COV20[upper.tri(COV20)]))^(1/2)
avg_covariance20

## [1] 0.02446958

data.frame(Case= c("Portfolio Volatility", "Estimator Covariance"), Value=
c(PortfolioVola2020, avg_covariance20))

##                   Case      Value
## 1 Portfolio Volatility 0.02296594
## 2 Estimator Covariance 0.02446958

##As a general rule, one can say that the volatility of a large, equally
weighted portfolio approximates the root of the average covariance of its
constituents.
```