HW\_01

Chantsal 108035487

4/3/2020

## Q1. Try to find the min variance portfolio for three stocks

Using 2007-2009 daily returns as the insample data

library(quantmod)

## Loading required package: xts

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: TTR

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

## Version 0.4-0 included new data defaults. See ?getSymbols.

library(plyr)  
tickers <- c("COST", "WMT", "TGT")  
getSymbols(tickers, from= "2007-01-01", to= "2009-12-31")

## 'getSymbols' currently uses auto.assign=TRUE by default, but will  
## use auto.assign=FALSE in 0.5-0. You will still be able to use  
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")  
## and getOption("getSymbols.auto.assign") will still be checked for  
## alternate defaults.  
##   
## This message is shown once per session and may be disabled by setting   
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

## [1] "COST" "WMT" "TGT"

data.env<-new.env()  
l\_ply(tickers, function(sym) try(getSymbols(sym, env=data.env), silent=T))  
stocks <- tickers[tickers %in% ls(data.env)]  
data <- xts()  
  
for(i in seq\_along(stocks)) {  
 symbol <- stocks[i]  
 data <- merge(data, Ad(get(symbol, envir=data.env)))  
}  
  
head(data)

## COST.Adjusted WMT.Adjusted TGT.Adjusted  
## 2007-01-03 39.47904 34.71480 41.03054  
## 2007-01-04 40.39801 34.88271 41.32474  
## 2007-01-05 39.90489 34.59798 41.09513  
## 2007-01-08 40.06926 34.31325 41.15252  
## 2007-01-09 40.36813 34.59798 41.80551  
## 2007-01-10 40.65204 34.51767 41.84137

str(data)

## An 'xts' object on 2007-01-03/2020-05-22 containing:  
## Data: num [1:3371, 1:3] 39.5 40.4 39.9 40.1 40.4 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr [1:3] "COST.Adjusted" "WMT.Adjusted" "TGT.Adjusted"  
## Indexed by objects of class: [POSIXct,POSIXt] TZ:   
## xts Attributes:   
## NULL

con = gzcon(url('https://github.com/systematicinvestor/SIT/raw/master/sit.gz', 'rb'))  
source(con)  
close(con)

## In other words, you have to compute optimal weights for 2010-01

Also given the weights from Q1, compute realized returns for 2010-01.

library(fBasics)

## Loading required package: timeDate

## Loading required package: timeSeries

##   
## Attaching package: 'timeSeries'

## The following object is masked from 'package:zoo':  
##   
## time<-

##   
## Attaching package: 'fBasics'

## The following objects are masked \_by\_ '.GlobalEnv':  
##   
## inv, vec

## The following object is masked from 'package:TTR':  
##   
## volatility

library(xts)  
library(PerformanceAnalytics)

##   
## Attaching package: 'PerformanceAnalytics'

## The following objects are masked from 'package:timeDate':  
##   
## kurtosis, skewness

## The following object is masked from 'package:graphics':  
##   
## legend

firm.data= data  
str(firm.data)

## An 'xts' object on 2007-01-03/2020-05-22 containing:  
## Data: num [1:3371, 1:3] 39.5 40.4 39.9 40.1 40.4 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr [1:3] "COST.Adjusted" "WMT.Adjusted" "TGT.Adjusted"  
## Indexed by objects of class: [POSIXct,POSIXt] TZ:   
## xts Attributes:   
## NULL

Sigma = cov(firm.data)  
std = sqrt(diag(Sigma))  
ones = rep(1,3)   
one.vec = matrix(ones, ncol=1)  
a = inv(Sigma)%\*%one.vec  
b = t(one.vec)%\*%a  
mvp.w =a / as.numeric(b)  
mvp.w

## [,1]  
## COST.Adjusted -0.3732524  
## WMT.Adjusted 0.7787558  
## TGT.Adjusted 0.5944966

mvp.ret<-sum((mvp.w)\*colMeans(firm.data))  
mvp.ret

## [1] 37.58636

## Give me weights, realized returns and standard deviation for 2010-01.

mu<-0.05/12  
return <- firm.data  
Ax <- rbind(2\*cov(return), colMeans(return), rep(1, ncol(return)))  
Ax <- cbind(Ax, rbind(t(tail(Ax, 2)), matrix(0, 2, 2)))  
b0 <- c(rep(0, ncol(return)), mu, 1)  
out<-solve(Ax, b0)  
wgt<-out[1:3]  
wgt

## COST.Adjusted WMT.Adjusted TGT.Adjusted   
## -0.8827369 -0.1147639 1.9975008

sum(wgt)

## [1] 1

ret.out<-sum(wgt\*colMeans(return))  
ret.out.annual<-ret.out\*12  
ret.out.annual

## [1] 0.05

std.out<-sqrt(t(wgt)%\*%cov(return)%\*%wgt)  
std.out.annual<-std.out\*sqrt(12)  
std.out.annual

## [,1]  
## [1,] 118.4935

## Q2. Based on Q1, you have to use weekly and monthly returns to get the answers to Q1

rm(list=ls())  
con = gzcon(url('https://github.com/systematicinvestor/SIT/raw/master/sit.gz', 'rb'))  
source(con)  
close(con)  
library(quantmod)  
tickers <- c("COST","WMT","TGT")  
n <- length(tickers)  
  
data <- new.env()  
getSymbols(tickers, src = 'yahoo', from = '2007-01-01',to= '2009-12-31', env = data, auto.assign = T)

## [1] "COST" "WMT" "TGT"

for(i in ls(data)) data[[i]] = adjustOHLC(data[[i]], use.Adjusted=T)  
  
data.monthly <- new.env()  
for(i in tickers) data.monthly[[i]] = to.monthly(data[[i]], indexAt='endof')  
data.monthly

## <environment: 0x55c5fffb1bb0>

bt.prep(data, align='remove.na', fill.gaps = T)  
names(data)

## [1] "prices" "COST" "dates" "WMT"   
## [5] "weight" ".getSymbols" "TGT" "symbolnames"   
## [9] "execution.price"

head(data$prices)

## COST TGT WMT  
## 2007-01-03 39.47904 41.03054 34.71480  
## 2007-01-04 40.39801 41.32474 34.88271  
## 2007-01-05 39.90489 41.09513 34.59798  
## 2007-01-08 40.06926 41.15252 34.31325  
## 2007-01-09 40.36813 41.80551 34.59798  
## 2007-01-10 40.65204 41.84137 34.51767

head(data$weight)

## COST TGT WMT  
## 2007-01-03 NA NA NA  
## 2007-01-04 NA NA NA  
## 2007-01-05 NA NA NA  
## 2007-01-08 NA NA NA  
## 2007-01-09 NA NA NA  
## 2007-01-10 NA NA NA

prices <- data$prices   
ret <- prices/mlag(prices)- 1  
head(ret)

## COST TGT WMT  
## 2007-01-03 NA NA NA  
## 2007-01-04 0.023277568 0.0071702932 0.004836986  
## 2007-01-05 -0.012206590 -0.0055561389 -0.008162467  
## 2007-01-08 0.004119069 0.0013962723 -0.008229757  
## 2007-01-09 0.007458760 0.0158676329 0.008298048  
## 2007-01-10 0.007032999 0.0008579013 -0.002321349

ret<- ret[-1, ]  
head(ret)

## COST TGT WMT  
## 2007-01-04 0.023277568 0.0071702932 0.004836986  
## 2007-01-05 -0.012206590 -0.0055561389 -0.008162467  
## 2007-01-08 0.004119069 0.0013962723 -0.008229757  
## 2007-01-09 0.007458760 0.0158676329 0.008298048  
## 2007-01-10 0.007032999 0.0008579013 -0.002321349  
## 2007-01-11 0.016725065 0.0257248489 0.006768215

### Use weekly

library(tibbletime)

##   
## Attaching package: 'tibbletime'

## The following object is masked from 'package:timeSeries':  
##   
## filter

## The following object is masked from 'package:stats':  
##   
## filter

library(timetk)

## Loading required package: recipes

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following objects are masked \_by\_ '.GlobalEnv':  
##   
## count, lst

## The following objects are masked from 'package:timeSeries':  
##   
## filter, lag

## The following objects are masked from 'package:plyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following objects are masked from 'package:xts':  
##   
## first, last

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

##   
## Attaching package: 'recipes'

## The following object is masked from 'package:stats':  
##   
## step

weekly <- prices %>%   
 tk\_tbl(rename\_index= "date") %>%  
 as\_tbl\_time(index= date) %>%  
 as\_period(period= "week", side= "end")  
  
#Q1. min variance   
str(weekly[,2:4])

## Classes 'tbl\_df', 'tbl' and 'data.frame': 157 obs. of 3 variables:  
## $ COST: num 39.9 41.8 42.4 41.4 41.8 ...  
## $ TGT : num 41.1 43.3 44.7 43.2 44.5 ...  
## $ WMT : num 34.6 35 35.3 34.8 35.1 ...

Sigma.w = cov(weekly[,2:4])  
std = sqrt(diag(Sigma.w))  
ones = rep(1,3)   
one.vec = matrix(ones, ncol=1)  
a = inv(Sigma.w)%\*%one.vec  
b = t(one.vec)%\*%a  
mvp.w.w =a / as.numeric(b)  
mvp.w.w

## [,1]  
## COST -0.2189428  
## TGT 0.3649845  
## WMT 0.8539583

mvp.ret.w<-sum((mvp.w.w)\*colMeans(weekly[,2:4]))  
mvp.ret.w

## [1] 36.19774

mu.w<-0.05/12  
return<- weekly[,2:4]  
Ax <- rbind(2\*cov(return), colMeans(return), rep(1, ncol(return)))  
Ax <- cbind(Ax, rbind(t(tail(Ax, 2)), matrix(0, 2, 2)))  
b0 <- c(rep(0, ncol(return)), mu.w, 1)  
out.w<-solve(Ax, b0)  
wgt<-out.w[1:3]  
wgt

## COST TGT WMT   
## -5.473517 3.153192 3.320325

sum(wgt)

## [1] 1

ret.out<-sum(wgt\*colMeans(return))  
ret.out.annual<-ret.out\*12  
ret.out.annual

## [1] 0.05

std.out<-sqrt(t(wgt)%\*%cov(return)%\*%wgt)  
std.out.annual<-std.out\*sqrt(12)  
std.out.annual

## [,1]  
## [1,] 80.04875

### use montly data

monthly <- prices %>%   
 tk\_tbl(rename\_index= "date") %>%  
 as\_tbl\_time(index= date) %>%  
 as\_period(period= "month", side= "end")  
  
#Q1. min variance   
str(monthly[,2:4])

## Classes 'tbl\_df', 'tbl' and 'data.frame': 36 obs. of 3 variables:  
## $ COST: num 42 41.9 40.3 40.2 42.4 ...  
## $ TGT : num 44 44.2 42.6 42.7 45 ...  
## $ WMT : num 34.8 35.3 34.4 35.2 35.1 ...

Sigma.m = cov(monthly[,2:4])  
std = sqrt(diag(Sigma.m))  
ones = rep(1,3)   
one.vec = matrix(ones, ncol=1)  
a = inv(Sigma.m)%\*%one.vec  
b = t(one.vec)%\*%a  
mvp.w.m =a / as.numeric(b)  
mvp.w.m

## [,1]  
## COST -0.2573111  
## TGT 0.4126788  
## WMT 0.8446323

mvp.ret.m<-sum((mvp.w.m)\*colMeans(monthly[,2:4]))  
mvp.ret.m

## [1] 35.97238

mu.m<-0.05/12  
return.m<-monthly[,2:4]  
Ax.m <- rbind(2\*cov(return.m), colMeans(return.m), rep(1, ncol(return.m)))  
Ax.m <- cbind(Ax, rbind(t(tail(Ax.m, 2)), matrix(0, 2, 2)))  
b0 <- c(rep(0, ncol(return.m)), mu.m, 1)  
out.m<-solve(Ax, b0)  
wgt.m<-out.m[1:3]  
wgt.m

## COST TGT WMT   
## -5.473517 3.153192 3.320325

sum(wgt.m)

## [1] 1

ret.out.m<-sum(wgt.m\*colMeans(return.m))  
ret.out.annual.m<-ret.out.m\*12  
ret.out.annual.m

## [1] -1.669898

std.out<-sqrt(t(wgt)%\*%cov(return)%\*%wgt)  
std.out.annual<-std.out\*sqrt(12)  
std.out.annual

## [,1]  
## [1,] 80.04875

## Q3. Find the tangency portfolio based on Q2.

mean <- apply(ret, 2, mean)  
mean

## COST TGT WMT   
## 0.0004121229 0.0002460886 0.0003854026

var <- apply(ret, 2, var)  
var

## COST TGT WMT   
## 0.0003964410 0.0008174913 0.0002671033

summary <- apply(ret, 2, summary)  
apply(ret, 2, skewness)

## COST TGT WMT   
## 0.2087881 0.2928746 0.4343347

cov.matrix <- matrix(cov(ret), ncol =3)  
cov.matrix

## [,1] [,2] [,3]  
## [1,] 0.0003964410 0.0003702950 0.0002106665  
## [2,] 0.0003702950 0.0008174913 0.0002875380  
## [3,] 0.0002106665 0.0002875380 0.0002671033

mu <- matrix(mean, ncol=1)  
mu

## [,1]  
## [1,] 0.0004121229  
## [2,] 0.0002460886  
## [3,] 0.0003854026

rf <- 0.005  
numerator <- solve(cov.matrix)%\*% (mu - rf\*one.vec)  
denominator <-t(one.vec)%\*% numerator  
tangency.weight <-numerator/as.numeric(denominator)  
tangency.return<- t(mu)%\*% tangency.weight  
tangency.var <- t(tangency.weight) %\*% cov.matrix %\*%tangency.weight  
tangency.std <- sqrt(as.numeric(tangency.var))  
Sharperatio <- (tangency.return/tangency.std)  
Sharperatio

## [,1]  
## [1,] 0.02589656

#HW\_02 ## Q1. Find the tangency portfolio with no short-sale constraint. Risk-free rate is assumed to be 1% per annum.

library(quantmod)  
library(plyr)  
library(xts)  
library(quadprog)  
library(fBasics)  
library(stats)  
tickers <- c("COST", "WMT", "TGT")  
getSymbols(tickers, from= "2007-01-01", to= "2009-12-31")

## [1] "COST" "WMT" "TGT"

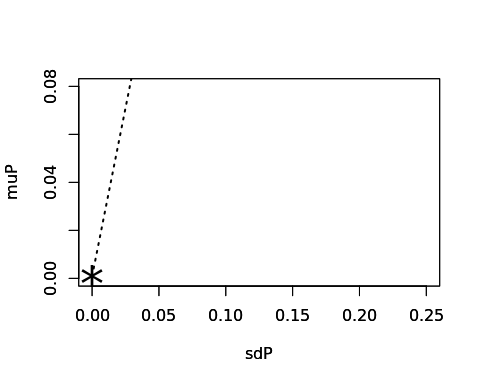
data.env<-new.env()  
l\_ply(tickers, function(sym) try(getSymbols(sym, env=data.env), silent=T))  
stocks <- tickers[tickers %in% ls(data.env)]  
data <- xts()  
  
for(i in seq\_along(stocks)) {  
 symbol <- stocks[i]  
 data <- merge(data, Ad(get(symbol, envir=data.env)))  
}  
  
  
mu = apply(data, 2, mean)  
Sigma = cov(data)  
std = sqrt(diag(Sigma))  
Amat1 = cbind(rep(1,3),mu, diag(1,nrow=3))   
t(Amat1)

## COST.Adjusted WMT.Adjusted TGT.Adjusted  
## 1.0000 1.00000 1.0000  
## mu 114.8834 61.85593 54.3253  
## 1.0000 0.00000 0.0000  
## 0.0000 1.00000 0.0000  
## 0.0000 0.00000 1.0000

# muP = seq(.01,.08,length=300) # set of 300 possible target values   
# When short sales are prohibited, the target expected return on the   
# portfolio must lie between the smallest   
# and largest expected returns on the stocks.   
muP = seq(min(mu)+.0001,max(mu)-.0001,length=300)  
sdP = muP # set up storage for standard deviations of portfolio returns  
weights = matrix(0,nrow=300,ncol=3) # storage for portfolio weights  
  
i=1  
for (i in 1:length(muP)) # find the optimal portfolios for each target expected return  
{  
 bvec1 = c(1,muP[i], rep(0,3)) # constraint vector  
 result = solve.QP(Dmat=2\*Sigma,dvec=rep(0,3),Amat=Amat1,bvec=bvec1,meq=2)  
 sdP[i] = sqrt(result$value)  
 weights[i,] = result$solution  
}  
  
  
plot(sdP,muP,type="l",xlim=c(0,0.25),ylim=c(0,0.08),lty=1)  
par(new=TRUE)  
plot(sdP,muP,type="l",xlim=c(0,0.25),ylim=c(0,0.08),lty=1, col="green")  
mufree = 0.01/12 # input value of risk-free interest rate  
points(0,mufree,cex=3, pch="\*") # show risk-free asset  
sharpe =( muP-mufree)/sdP # compute Sharpe ratios  
ind = (sharpe == max(sharpe)) # Find maximum Sharpe ratio  
options(digits=3)  
weights[ind,] # Find tangency portfolio# show line of optimal portfolios

## [1] 0.000 0.377 0.623

lines(c(0,2),mufree+c(0,2)\*(muP[ind]-mufree)/sdP[ind],lwd=2,lty=3)  
# show line of optimal portfolios  
points(sdP[ind],muP[ind],cex=1,pch=19, col="red") # show tangency portfolio  
ind2 = (sdP == min(sdP)) # find the minimum variance portfolio  
points(sdP[ind2],muP[ind2],cex=2,pch="+", col="blue") # show minimum variance portfolio  
ind3 = (muP > muP[ind2])  
lines(sdP[ind3],muP[ind3],type="l",xlim=c(0,.25),  
 ylim=c(0,.08),lwd=1) # plot the efficient frontier  
points(c(std[1],std[2], std[3]), c(mu[1], mu[2], mu[3]), cex=1, pch="o", col="red")   
text(std[1],mu[1],"Costco",cex=1, pos=4)  
text(std[2],mu[2],"Wallmart",cex=1, pos=4)  
text(std[3],mu[3],"Target",cex=1, pos=4)



graphics.off()

## Q2.Plot the efficient frontier with and without short-sale constraint.

con = gzcon(url('https://github.com/systematicinvestor/SIT/raw/master/sit.gz', 'rb'))  
source(con)  
close(con)  
  
tickers <- c("COST", "WMT", "TGT")  
n <- length(tickers)  
#with short sale constraint.  
constraints = new.constraints(n, lb = -Inf, ub = +Inf)  
  
constraints = add.constraints(rep(1, n), 1, type = '=', constraints)   
ia<- create.historical.ia(ret, 250)  
  
weight <- min.risk.portfolio(ia, constraints)  
#without short sale constraint.  
ia<- create.historical.ia(ret, 250)  
constraints = new.constraints(n, lb = 0, ub = 1)  
constraints = add.constraints(diag(n), type='>=', b=0, constraints)  
constraints = add.constraints(diag(n), type='<=', b=1, constraints)  
   
constraints = add.constraints(rep(1, n), 1, type = '=', constraints)   
  
weight <- min.risk.portfolio(ia, constraints)  
weight

## [1] 0.233 0.000 0.767

ifelse(!require(corpcor), install.packages("corpcor"), library(corpcor))

## Loading required package: corpcor

##   
## Attaching package: 'corpcor'

## The following object is masked \_by\_ '.GlobalEnv':  
##   
## cov.shrink

## [1] "corpcor"

ifelse(!require(lpSolve), install.packages("lpSolve"), library(lpSolve))

## Loading required package: lpSolve

## [1] "lpSolve"

ef = portopt(ia, constraints, 50, 'Efficient Frontier')

##   
## Attaching package: 'kernlab'

## The following object is masked \_by\_ '.GlobalEnv':  
##   
## cross

ef

## $weight  
## COST TGT WMT  
## [1,] 0.233 0.00e+00 0.7670  
## [2,] 0.249 6.49e-18 0.7513  
## [3,] 0.264 4.78e-18 0.7357  
## [4,] 0.280 3.08e-18 0.7200  
## [5,] 0.296 1.37e-18 0.7044  
## [6,] 0.311 -3.33e-19 0.6887  
## [7,] 0.327 -2.04e-18 0.6731  
## [8,] 0.343 -3.74e-18 0.6574  
## [9,] 0.358 -5.45e-18 0.6418  
## [10,] 0.374 -7.16e-18 0.6261  
## [11,] 0.390 -8.86e-18 0.6105  
## [12,] 0.405 -1.06e-17 0.5948  
## [13,] 0.421 -1.23e-17 0.5792  
## [14,] 0.436 -1.40e-17 0.5635  
## [15,] 0.452 -1.57e-17 0.5479  
## [16,] 0.468 -1.74e-17 0.5322  
## [17,] 0.483 -1.91e-17 0.5165  
## [18,] 0.499 -2.08e-17 0.5009  
## [19,] 0.515 -2.25e-17 0.4852  
## [20,] 0.530 -2.42e-17 0.4696  
## [21,] 0.546 -2.59e-17 0.4539  
## [22,] 0.562 -2.76e-17 0.4383  
## [23,] 0.577 -2.93e-17 0.4226  
## [24,] 0.593 -3.10e-17 0.4070  
## [25,] 0.609 -3.27e-17 0.3913  
## [26,] 0.624 -3.45e-17 0.3757  
## [27,] 0.640 -3.62e-17 0.3600  
## [28,] 0.656 -3.79e-17 0.3444  
## [29,] 0.671 -3.96e-17 0.3287  
## [30,] 0.687 -4.13e-17 0.3131  
## [31,] 0.703 -4.30e-17 0.2974  
## [32,] 0.718 -4.47e-17 0.2818  
## [33,] 0.734 -4.64e-17 0.2661  
## [34,] 0.750 -4.81e-17 0.2504  
## [35,] 0.765 -4.98e-17 0.2348  
## [36,] 0.781 -5.15e-17 0.2191  
## [37,] 0.797 -5.32e-17 0.2035  
## [38,] 0.812 -5.49e-17 0.1878  
## [39,] 0.828 -5.66e-17 0.1722  
## [40,] 0.843 -5.83e-17 0.1565  
## [41,] 0.859 -6.00e-17 0.1409  
## [42,] 0.875 -6.17e-17 0.1252  
## [43,] 0.890 -6.35e-17 0.1096  
## [44,] 0.906 -6.52e-17 0.0939  
## [45,] 0.922 -6.69e-17 0.0783  
## [46,] 0.937 -6.86e-17 0.0626  
## [47,] 0.953 -7.03e-17 0.0470  
## [48,] 0.969 -7.20e-17 0.0313  
## [49,] 0.984 -7.37e-17 0.0157  
## [50,] 1.000 0.00e+00 0.0000  
##   
## $return  
## [,1]  
## [1,] 0.103  
## [2,] 0.103  
## [3,] 0.103  
## [4,] 0.103  
## [5,] 0.103  
## [6,] 0.103  
## [7,] 0.104  
## [8,] 0.104  
## [9,] 0.104  
## [10,] 0.104  
## [11,] 0.104  
## [12,] 0.104  
## [13,] 0.104  
## [14,] 0.104  
## [15,] 0.104  
## [16,] 0.105  
## [17,] 0.105  
## [18,] 0.105  
## [19,] 0.105  
## [20,] 0.105  
## [21,] 0.105  
## [22,] 0.105  
## [23,] 0.105  
## [24,] 0.105  
## [25,] 0.106  
## [26,] 0.106  
## [27,] 0.106  
## [28,] 0.106  
## [29,] 0.106  
## [30,] 0.106  
## [31,] 0.106  
## [32,] 0.106  
## [33,] 0.107  
## [34,] 0.107  
## [35,] 0.107  
## [36,] 0.107  
## [37,] 0.107  
## [38,] 0.107  
## [39,] 0.107  
## [40,] 0.107  
## [41,] 0.107  
## [42,] 0.108  
## [43,] 0.108  
## [44,] 0.108  
## [45,] 0.108  
## [46,] 0.108  
## [47,] 0.108  
## [48,] 0.108  
## [49,] 0.108  
## [50,] 0.109  
##   
## $risk  
## [1] 0.252 0.252 0.252 0.252 0.252 0.253 0.253 0.253 0.254 0.254 0.255 0.256  
## [13] 0.256 0.257 0.258 0.259 0.259 0.260 0.261 0.262 0.263 0.265 0.266 0.267  
## [25] 0.268 0.270 0.271 0.273 0.274 0.276 0.277 0.279 0.281 0.282 0.284 0.286  
## [37] 0.288 0.289 0.291 0.293 0.295 0.297 0.299 0.302 0.304 0.306 0.308 0.310  
## [49] 0.313 0.315  
##   
## $name  
## [1] "Efficient Frontier"

## Q3. Repeat Q1 and Q2 using weekly and monthly returns.

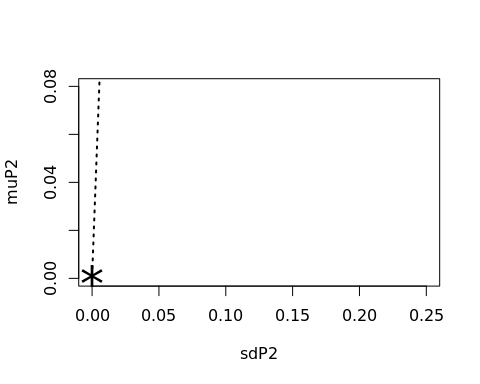
library(tibbletime)  
library(timetk)  
library(fBasics)  
library(quadprog)  
   
weekly <- prices %>%   
 tk\_tbl(rename\_index= "date") %>%  
 as\_tbl\_time(index= date) %>%  
 as\_period(period= "week", side= "end")  
Sigma1 = cov(weekly[,2:4])  
std1 = sqrt(diag(Sigma1))  
mu = apply(weekly[,2:4], 2, mean)  
  
Amat2 = cbind(rep(1,3),mu, diag(1,nrow=3))   
t(Amat2)

## COST TGT WMT  
## 1.0 1.0 1.0  
## mu 44.1 36.3 38.2  
## 1.0 0.0 0.0  
## 0.0 1.0 0.0  
## 0.0 0.0 1.0

muP2 = seq(min(mu)+.0001, max(mu)-.0001, length = 150)  
sdP2 = muP2  
weights2 = matrix(0,nrow=150,ncol=3)   
  
i=1  
for (i in 1:length(muP2))   
   
{  
 bvec2 = c(1,muP2[i], rep(0,3))   
 result = solve.QP(Dmat=2\*Sigma1,dvec=rep(0,3),Amat=Amat2,bvec=bvec2,meq=2)  
 sdP2[i] = sqrt(result$value)  
 weights2[i,] = result$solution  
}  
  
plot(sdP2,muP2,type="l",xlim=c(0,0.25),ylim=c(0,0.08),lty=1)  
mufree.w = 0.01/12   
points(0,mufree.w,cex=3, pch="\*")   
sharpe =( muP2-mufree.w)/sdP2  
ind1 = (sharpe == max(sharpe))   
options(digits=3)  
weights2[ind1,]

## [1] 0.000255 0.248673 0.751072

lines(c(0,2),mufree.w+c(0,2)\*(muP2[ind1]-mufree.w)/sdP2[ind1],lwd=2,lty=3)  
  
points(sdP2[ind1],muP2[ind1],cex=1,pch=19, col="red")   
ind2 = (sdP2 == min(sdP2))   
points(sdP2[ind2],muP2[ind2],cex=2,pch="+", col="blue")   
ind3 = (muP2 > muP2[ind2])  
lines(sdP2[ind3],muP2[ind3],type="l",xlim=c(0,.25),  
 ylim=c(0,.08),lwd=1)   
points(c(std1[1],std1[2], std1[3]), c(mu[1], mu[2], mu[3]), cex=1, pch="o", col="red")   
text(std1[1],mu[1],"Costco",cex=1, pos=4)  
text(std1[2],mu[2],"Wallmart",cex=1, pos=4)  
text(std1[3],mu[3],"Target",cex=1, pos=4)



graphics.off()

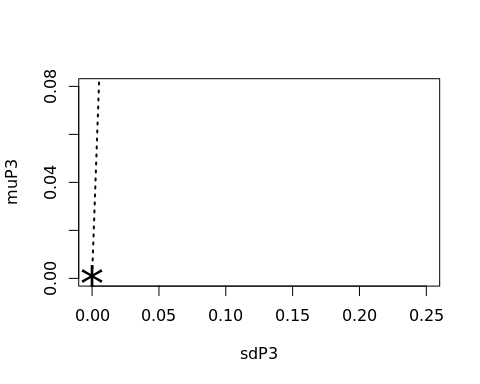
monthly <- prices %>%   
 tk\_tbl(rename\_index= "date") %>%  
 as\_tbl\_time(index= date) %>%  
 as\_period(period= "month", side= "end")  
  
Sigma2 = cov(monthly[,2:4])  
std2= sqrt(diag(Sigma2))  
mu = apply(monthly[,2:4], 2, mean)  
Amat3 = cbind(rep(1,3),mu, diag(1,nrow=3))  
t(Amat3)

## COST TGT WMT  
## 1.0 1.0 1.0  
## mu 44.2 36.3 38.3  
## 1.0 0.0 0.0  
## 0.0 1.0 0.0  
## 0.0 0.0 1.0

muP3 = seq(min(mu)+.0001,max(mu)-.0001,length=300)  
sdP3= muP3  
weights3= matrix(0,nrow=300,ncol=3)   
  
i=1  
for (i in 1:length(muP3))   
   
{  
 bvec3 = c(1,muP3[i], rep(0,3))   
 result = solve.QP(Dmat=2\*Sigma2,dvec=rep(0,3),Amat=Amat3,bvec=bvec3,meq=2)  
 sdP3[i] = sqrt(result$value)  
 weights3[i,] = result$solution  
}  
  
plot(sdP3,muP3,type="l",xlim=c(0,0.25),ylim=c(0,0.08),lty=1)  
mufree.m = 0.01/12   
points(0,mufree.m,cex=3, pch="\*")   
sharpe =( muP3-mufree.m)/sdP3  
ind2 = (sharpe == max(sharpe))   
options(digits=3)  
weights3[ind2,]

## [1] -2.02e-18 2.80e-01 7.20e-01

lines(c(0,2),mufree.m+c(0,2)\*(muP3[ind2]-mufree.m)/sdP3[ind2],lwd=2,lty=3)  
points(sdP3[ind2],muP3[ind2],cex=1,pch=19, col="red")   
ind2 = (sdP3 == min(sdP3))   
points(sdP3[ind2],muP3[ind2],cex=2,pch="+", col="blue")   
  
ind3 = (muP3 > muP3[ind2])  
lines(sdP3[ind3],muP3[ind3],type="l",xlim=c(0,.25),  
 ylim=c(0,.08),lwd=1)   
points(c(std[1],std[2], std[3]), c(mu[1], mu[2], mu[3]), cex=1, pch="o", col="red")   
text(std[1],mu[1],"Costco",cex=1, pos=4)  
text(std[2],mu[2],"Wallmart",cex=1, pos=4)  
text(std[3],mu[3],"Target",cex=1, pos=4)



graphics.off()