Markowitz Mean-Variance Optimization

Hypothetical Portfolio of SPY and SPAB

```
names <- c("SPY", "SPAB")
date.from <- "2010-12-31"; date.to <- "2013-12-31"

getMonthlyReturns <- function(name, from = date.from, to = date.to) {
   prices <- getSymbols(name, from = date.from, to = date.to, auto.assign = F)
   monthly <- to.monthly(prices)
   returns <- Delt(monthly[, 6])[-1]

   ret.data <- data.table(Month = index(returns), Company = name, Return = returns[, 1])
   colnames(ret.data) <- c("Month", "Company", "Return")

   return(ret.data)
}

returns <- data.table::rbindlist(lapply(names, getMonthlyReturns))
```

```
'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.
```

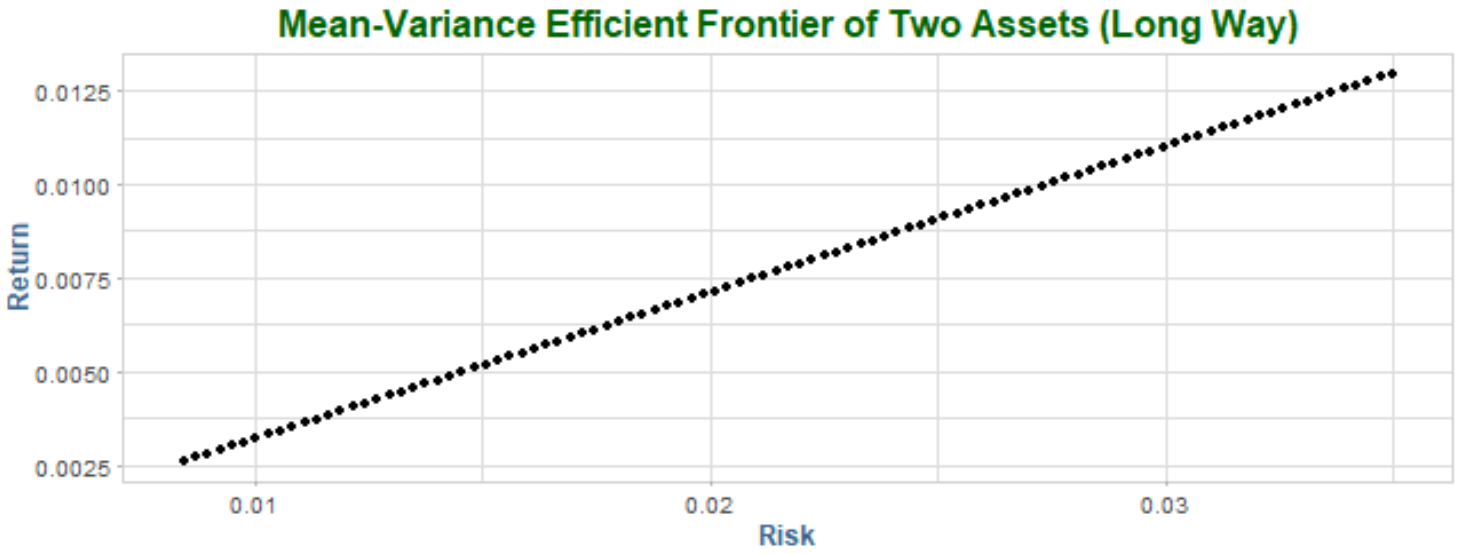
```
portfolio <- dcast(returns, Month ~ Company, value.var = c("Return"))

portfolio.avg <- apply(portfolio[, 2:3], 2, mean)
portfolio.sd <- apply(portfolio[, 2:3], 2, sd)
portfolio.covar <- cov(portfolio[, 2:3])[1, 1]

portfolio.wgt.spy <- data.table(Weight = seq(0, 1, by = 0.01))
portfolio.wgt.spab <- 1 - portfolio.wgt.spy

portfolio.weights <- cbind(Wgt.SPY = portfolio.wgt.spy, Wgt.SPAB = portfolio.wgt.spab)

portfolio.returns <- portfolio.weights$Wgt.SPY.Weight * portfolio.avg[2] + portfolio.weights$Wg
portfolio.risk <- portfolio.weights$Wgt.SPY.Weight * portfolio.sd[2] + portfolio.weights$Wgt.SP

portfolio.profiles <- cbind(portfolio.weights, portfolio.returns, portfolio.risk)

portfolio.profiles[which.min(portfolio.profiles$portfolio.risk)]
```

```
   Wgt.SPY.Weight Wgt.SPAB.Weight portfolio.returns portfolio.risk
1:              0               1       0.002600346    0.008439225
```

```
riskFree <- 0.007/12

portfolio.Sharpe <- (portfolio.returns - riskFree) / portfolio.risk

port.stats <- data.table(SPY.Wgt = portfolio.wgt.spy,
                         SPAB.Wgt = portfolio.wgt.spab,
                         Return = portfolio.returns,
                         Risk = portfolio.risk,
                         Sharpe = portfolio.Sharpe)

port.stats[which.max(port.stats$Sharpe)]
```

```
   SPY.Wgt.Weight SPAB.Wgt.Weight     Return       Risk    Sharpe
1:              1               0 0.01295431 0.03498053 0.3536533
```

```
ggplot(port.stats, aes(Risk, Return)) +
   geom_point() +
   labs(title = "Mean-Variance Efficient Frontier of Two Assets (Long Way)")
```



Quadratic Way

```
mat.ret <- as.matrix(portfolio[, 2:3])

tail(mat.ret)
```

```
               SPAB         SPY
[31,] -0.0003699898  0.05167731
```

```
[32,] -0.0078708850 -0.02999255
[33,]  0.0151824158  0.03164635
[34,]  0.0055572991  0.04630651
[35,] -0.0034871141  0.02963788
[36,] -0.0038379023  0.02109404
```

```
VCOV <- cov(mat.ret)

avg.ret <- matrix(apply(mat.ret, 2, mean))
colnames(avg.ret) <- paste("Avg.Ret")
rownames(avg.ret) <- paste(c("Lag", "Spy"))

min.ret <- min(avg.ret)
max.ret <- max(avg.ret)

increments <- 100
tgt.ret <- seq(min.ret, max.ret, length = increments)

head(tgt.ret)
```

```
[1] 0.002600346 0.002704931 0.002809517 0.002914102 0.003018688 0.003123273
```

```
tail(tgt.ret)
```

```
[1] 0.01243138 0.01253597 0.01264055 0.01274514 0.01284973 0.01295431
```

```
tgt.sd <- rep(0, length = increments)
tgt.sd
```

```
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
wgt <- matrix(0, nrow=increments, ncol = length(avg.ret))

head(wgt)
```

```
     [,1] [,2]
[1,]    0    0
[2,]    0    0
[3,]    0    0
[4,]    0    0
[5,]    0    0
[6,]    0    0
```

```
for(i in 1:increments){
   Dmat <- 2*VCOV
   dvec <- c(rep(0, length(avg.ret)))
```

```
   Amat <- cbind(rep(1, length(avg.ret)), avg.ret, diag(1, nrow=2))
   bvec <- c(1, tgt.ret[i], rep(0, 2))
   soln <- solve.QP(Dmat, dvec, Amat, bvec=bvec, meq=2)
   tgt.sd[i] <- sqrt(soln$value)
   wgt[i, ] <- soln$solution
}

head(tgt.sd)
```

```
[1] 0.008439225 0.008299713 0.008175279 0.008066620 0.007974380 0.007899136
```

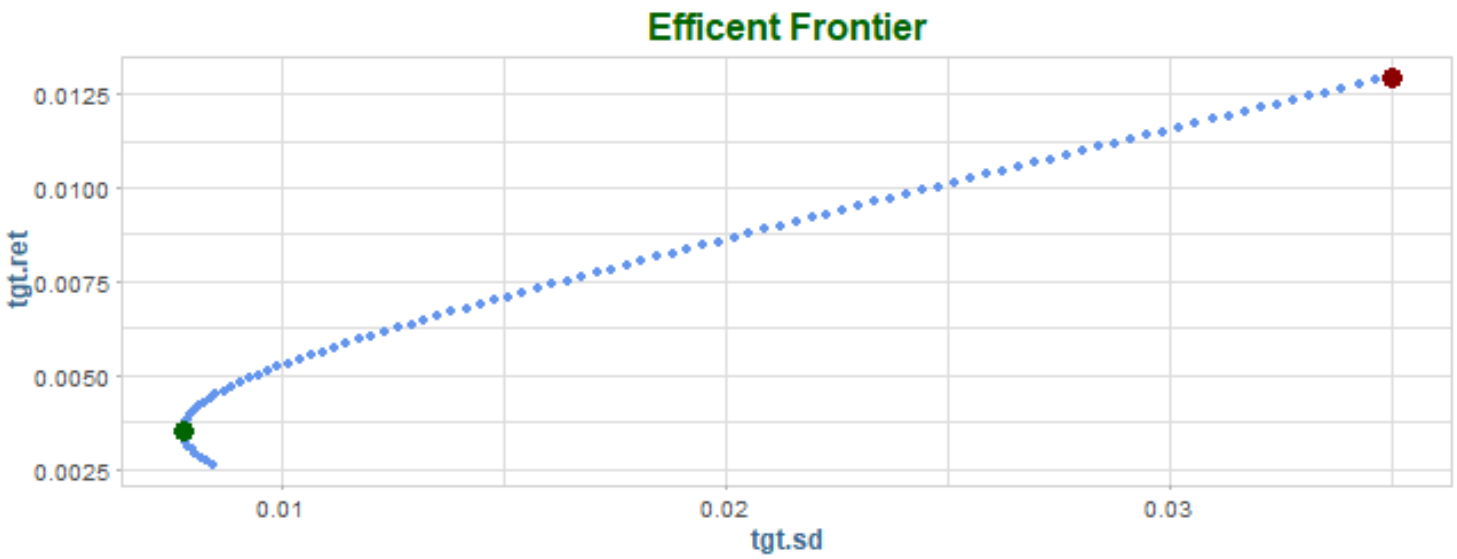
```
tail(tgt.sd)
```

```
[1] 0.03314223 0.03350945 0.03387689 0.03424456 0.03461244 0.03498053
```

```
head(wgt)
```

```
          [,1]       [,2]
[1,] 1.0000000 0.00000000
[2,] 0.9898990 0.01010101
[3,] 0.9797980 0.02020202
[4,] 0.9696970 0.03030303
[5,] 0.9595960 0.04040404
[6,] 0.9494949 0.05050505
```

```
colnames(wgt) <- c("Lag", "Spy")
tgt.port <- data.table(cbind(tgt.ret, tgt.sd, wgt))

min.var <- tgt.port[which.min(tgt.port$tgt.sd)]
max.ret <- tgt.port[which.max(tgt.port$tgt.ret)]

ggplot(tgt.port, aes(tgt.sd, tgt.ret)) +
   geom_point(col = "cornflowerblue") +
   geom_point(data = min.var, aes(x = tgt.sd, y = tgt.ret), col = "darkgreen", size = 4) +
   geom_point(data = max.ret, aes(x = tgt.sd, y = tgt.ret), col = "darkred", size = 4) +
   labs(title = "Efficent Frontier")
```


Efficent Frontier
0.0125
0.0100
0.0075
0.0050
0.0025
tgt.ret
0.01
0.02
0.03
tgt.sd