# Economist at Large

# Modern Portfolio Theory

# Use solve.QP to solve for efficient frontier

# Last Edited 5/3/13

# This file uses the solve.QP function in the quadprog package to solve for the

# efficient frontier.

# Since the efficient frontier is a parabolic function, we can find the solution

# that minimizes portfolio variance and then vary the risk premium to find

# points along the efficient frontier. Then simply find the portfolio with the

# largest Sharpe ratio (expected return / sd) to identify the most

# efficient portfolio

library(stockPortfolio) # Base package for retrieving returns

library(ggplot2) # Used to graph efficient frontier

library(reshape2) # Used to melt the data

library(quadprog) #Needed for solve.QP

# Create the portfolio using ETFs, incl. hypothetical non-efficient allocation

stocks <- c(

 "VTSMX" = .0,

 "SPY" = .20,

 "EFA" = .10,

 "IWM" = .10,

 "VWO" = .30,

 "LQD" = .20,

 "HYG" = .10)

# Retrieve returns, from earliest start date possible (where all stocks have

# data) through most recent date

returns <- getReturns(names(stocks[-1]), freq="week") #Currently, drop index

#### Efficient Frontier function ####

eff.frontier <- function (returns, short="no", max.allocation=NULL,

 risk.premium.up=.5, risk.increment=.005){

 # return argument should be a m x n matrix with one column per security

 # short argument is whether short-selling is allowed; default is no (short

 # selling prohibited)max.allocation is the maximum % allowed for any one

 # security (reduces concentration) risk.premium.up is the upper limit of the

 # risk premium modeled (see for loop below) and risk.increment is the

 # increment (by) value used in the for loop

 covariance <- cov(returns)

 print(covariance)

 n <- ncol(covariance)

 # Create initial Amat and bvec assuming only equality constraint

 # (short-selling is allowed, no allocation constraints)

 Amat <- matrix (1, nrow=n)

 bvec <- 1

 meq <- 1

 # Then modify the Amat and bvec if short-selling is prohibited

 if(short=="no"){

 Amat <- cbind(1, diag(n))

 bvec <- c(bvec, rep(0, n))

 }

 # And modify Amat and bvec if a max allocation (concentration) is specified

 if(!is.null(max.allocation)){

 if(max.allocation > 1 | max.allocation <0){

 stop("max.allocation must be greater than 0 and less than 1")

 }

 if(max.allocation \* n < 1){

 stop("Need to set max.allocation higher; not enough assets to add to 1")

 }

 Amat <- cbind(Amat, -diag(n))

 bvec <- c(bvec, rep(-max.allocation, n))

 }

 # Calculate the number of loops

 loops <- risk.premium.up / risk.increment + 1

 loop <- 1

 # Initialize a matrix to contain allocation and statistics

 # This is not necessary, but speeds up processing and uses less memory

 eff <- matrix(nrow=loops, ncol=n+3)

 # Now I need to give the matrix column names

 colnames(eff) <- c(colnames(returns), "Std.Dev", "Exp.Return", "sharpe")

 # Loop through the quadratic program solver

 for (i in seq(from=0, to=risk.premium.up, by=risk.increment)){

 dvec <- colMeans(returns) \* i # This moves the solution along the EF

 sol <- solve.QP(covariance, dvec=dvec, Amat=Amat, bvec=bvec, meq=meq)

 eff[loop,"Std.Dev"] <- sqrt(sum(sol$solution\*colSums((covariance\*sol$solution))))

 eff[loop,"Exp.Return"] <- as.numeric(sol$solution %\*% colMeans(returns))

 eff[loop,"sharpe"] <- eff[loop,"Exp.Return"] / eff[loop,"Std.Dev"]

 eff[loop,1:n] <- sol$solution

 loop <- loop+1

 }

 return(as.data.frame(eff))

}

# Run the eff.frontier function based on no short and 50% alloc. restrictions

eff <- eff.frontier(returns=returns$R, short="no", max.allocation=.50,

 risk.premium.up=1, risk.increment=.001)

# Find the optimal portfolio

eff.optimal.point <- eff[eff$sharpe==max(eff$sharpe),]

# graph efficient frontier

# Start with color scheme

ealred <- "#7D110C"

ealtan <- "#CDC4B6"

eallighttan <- "#F7F6F0"

ealdark <- "#423C30"

ggplot(eff, aes(x=Std.Dev, y=Exp.Return)) + geom\_point(alpha=.1, color=ealdark) +

 geom\_point(data=eff.optimal.point, aes(x=Std.Dev, y=Exp.Return, label=sharpe),

 color=ealred, size=5) +

 annotate(geom="text", x=eff.optimal.point$Std.Dev,

 y=eff.optimal.point$Exp.Return,

 label=paste("Risk: ",

 round(eff.optimal.point$Std.Dev\*100, digits=3),"\nReturn: ",

 round(eff.optimal.point$Exp.Return\*100, digits=4),"%\nSharpe: ",

 round(eff.optimal.point$sharpe\*100, digits=2), "%", sep=""),

 hjust=0, vjust=1.2) +

 ggtitle("Efficient Frontier\nand Optimal Portfolio") +

 labs(x="Risk (standard deviation of portfolio)", y="Return") +

 theme(panel.background=element\_rect(fill=eallighttan),

 text=element\_text(color=ealdark),

 plot.title=element\_text(size=24, color=ealred))

ggsave("Efficient Frontier.png")