```r
require(fImport)
require(PerformanceAnalytics)
require(tseries)
require(stats)

options(scipen=100)
options(digits=4)

#Function to load stock data into a Time Series object
importSeries = function (symbol,from,to) {


#Read data from Yahoo! Finance
      input = yahooSeries(symbol,from=from,to=to)

#Character Strings for Column Names
      adjClose = paste(symbol,".Adj.Close",sep="")
      inputReturn = paste(symbol,".Return",sep="")
      CReturn = paste(symbol,".CReturn",sep="")

#Calculate the Returns and put it on the time series
      input.Return = returns(input[,adjClose])
      colnames(input.Return)[1] = inputReturn
      input = merge(input,input.Return)

#Calculate the cumulative return and put it on the time series
      input.first = input[,adjClose][1]

      input.CReturn = fapply(input[,adjClose],FUN=function(x) log(x) - log(input.first))
      colnames(input.CReturn)[1] = CReturn
      input = merge(input,input.CReturn)

#Deleting things (not sure I need to do this, but I can't not delete things if
# given a way to...
      rm(input.first,input.Return,input.CReturn,adjClose,inputReturn,CReturn)

#Return the timeseries
      return(input)
}

#Get Return Series for Short Medium and Long Term Gov Bonds
from = "2001-01-01"
to = "2011-12-16"
tlt = importSeries("tlt",from,to)
shy = importSeries("shy",from,to)
ief = importSeries("ief",from,to)

merged = merge(tlt,shy)
merged = merge(merged,ief)

vars = c("tlt.Return","shy.Return","ief.Return")

#Calculate Annualized Returns
t = table.AnnualizedReturns(merged[,vars],Rf=mean(merged[,"shy.Return"],na.rm=TRUE))
t

#Get the annualized return and StdDev for each series from the table
rTLT = t['Annualized Return',"tlt.Return"]
rSHY = t['Annualized Return',"shy.Return"]
rIEF = t['Annualized Return','ief.Return']
sTLT = t['Annualized Std Dev',"tlt.Return"]
sSHY = t['Annualized Std Dev',"shy.Return"]
sIEF = t['Annualized Std Dev',"ief.Return"]
```

```
#Check the correlations
corr = cor(merged[,vars],use='complete.obs')
c = corr['tlt.Return','shy.Return']

#Assuming a barbell strategy of long and short holdings, what is the risk return profile
ws = NULL
wt = NULL
mu = NULL
sigma = NULL

#50 observations
n=50

#Loop through the weights of the barbell
for (i in 0:n){
      wsi = i/n;
      wti = 1-wsi;

      mui = wsi * rSHY + wti * rTLT
      sigmai = wsi*wsi*sSHY*sSHY + wti*wti*sTLT*sTLT + wsi*wti*sSHY*sTLT*c


      ws = c(ws,wsi)
      wt = c(wt,wti)
      mu = c(mu,mui)
      sigma = c(sigma,sigmai)
}

#Risk Return Profile Data Frame
rrProfile = data.frame(ws=ws,wt=wt,mu=mu,sigma=sigma)

#Plot the profile
plot(rrProfile$sigma,
     rrProfile$mu,
       xlim=c(0,.022),
       ylim=c(0,.08),
       ylab="Expected Yearly Return",
       xlab="Expected Yearly Variance",
       main="Efficient Frontier for Government Bond Portfolios")

#Fit a quadratic function to the profile
fit = lm(rrProfile$sigma ~ rrProfile$mu + I(rrProfile$mu^2))

#How is the fit?
summary(fit)

#get the coefficients
coe = fit$coefficients

#Get predicted values risk values for each return
muf = NULL
sfit = NULL
for (i in seq(0,.08,by=.001)){
      muf = c(muf,i)

      s = coe[1] + coe[2]*i + coe[3]*i^2
      sfit = c(sfit,s)
}

#plot the predicted frontier
lines(sfit,muf,col='red')
```

```
#now let's add the 3rd asset.  Unless we want to do a grid search, we need
#to optimize the portfolio, minimizing the risk for each level of return

#portfolio.optim cannot have NA values in the time series, filter them out
m2 = removeNA(merged[,vars])

wSHY = NULL
wIEF = NULL
wTLT = NULL
er = NULL
eStd = NULL

#loop through finding the optimum portfolio for return levels between
#the minimum (rSHY) and the max(rTLT)
#
#portfolio.optim uses daily returns, so we have to adjust accordingly
for (i in seq((rSHY+.001),(rTLT-.001),length.out=100)){
      pm = 1+i
      pm = log(pm)/255
      opt = portfolio.optim(m2,pm=pm)
      er = c(er,exp(pm*255)-1)
      eStd = c(eStd,opt$ps*sqrt(255))
      wTLT = c(wTLT,opt$pw[1])
      wSHY = c(wSHY,opt$pw[2])
      wIEF = c(wIEF,opt$pw[3])
}

#Plot the efficient frontier for the 3 assets.
lines(eStd^2,er,col='blue')
legend(.014,0.015,c("'Barbell' Strategy","All Assets"),
            col=c("red","blue"),
            lty=c(1,1))

solution = data.frame(wTLT,wSHY,wIEF,er,eStd)
```