

**CFRM 543 –**

**Portfolio Optimization and Asset Management**

**Final Project**

**Jeffrey Lea and Joanna Nowak**

Spring 2015

Table of Contents

[The Challenge 3](#_Toc421581364)

[Portfolios – Created 4](#_Toc421581365)

[Performance Charts and Statistics 5](#_Toc421581366)

[Analysis – Optimal Portfolios 8](#_Toc421581367)

[Appendix – R Code 9](#_Toc421581368)

# The Challenge

The General Problem of Selecting an Optimal Portfolio Strategy – our objective is to create an optimal portfolio strategy using multi-objectives taking into consideration many performance indicators, such risk-adjusted performance measures, cumulative geometric return, risk measures, maximum drawdown, turnover, and diversification. In an asset management organization, final decisions on which of several competing strategy proposals are chosen for a product roll-out would be made after extensive discussions by the Investment Committee, including CIO, Managing Directors, etc.

The returns data to be used for this project represents returns classified into four different sizes of market capitalization: micro-cap, small-cap, mid-cap and large-cap. The returns data used is from 1997 through 2001 and represents weekly returns.

In this project, we will identify several different portfolios and compare each of the portfolios using different performance measurements to determine the optimal portfolio for these returns (note there may be other performance measurements that are important and have not been considered). The performance measures used will be: Sharpe Ratio, STARR ratio and geometric cumulative returns.

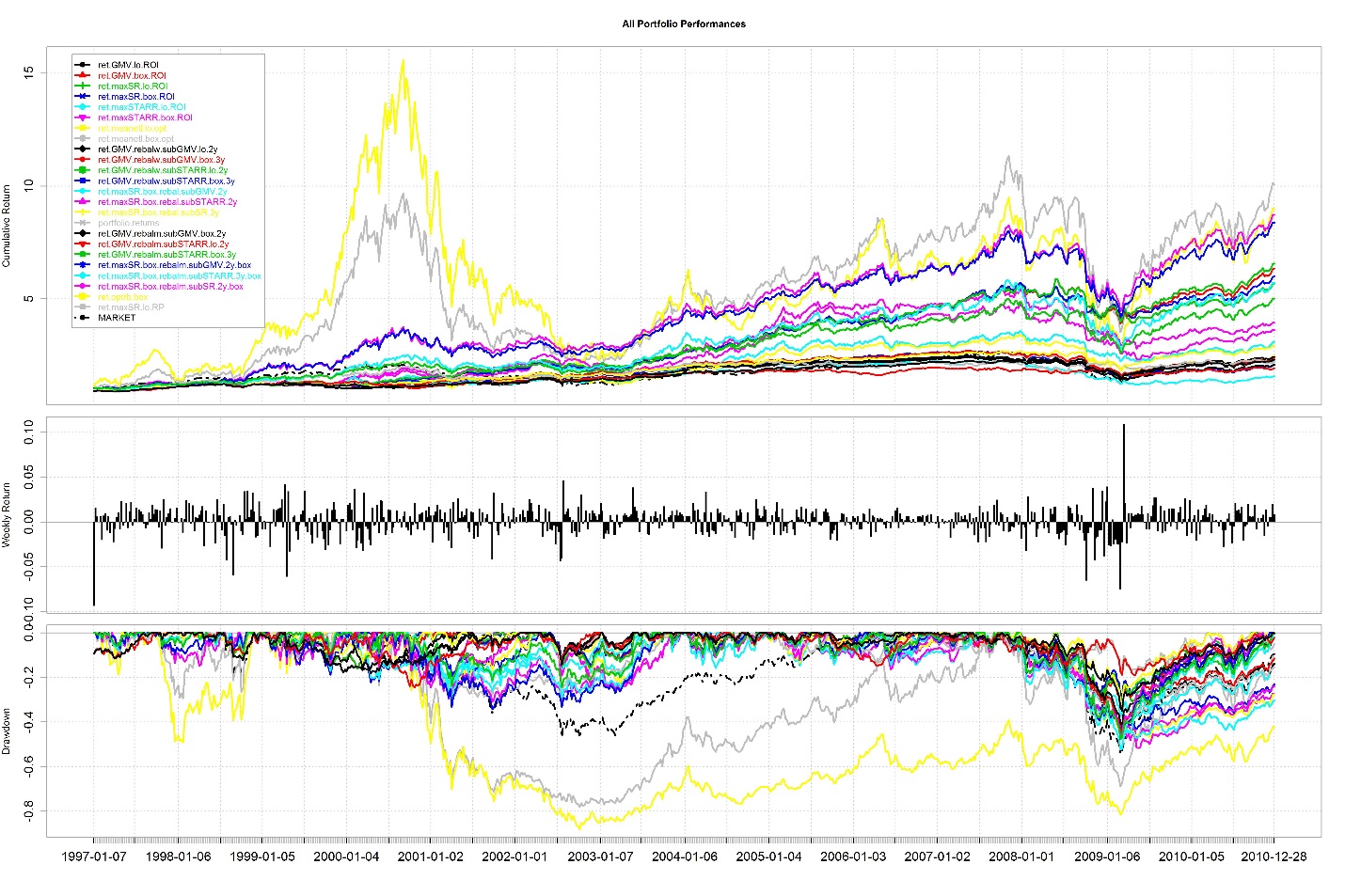
# Portfolios – Created

In determining the optimal portfolios using the supplied data we generated the following portfolios. Due to the ability to generate many portfolios our box constrained optimizations all utilized a maximum shorting of 5% and maximum weight of 35%. In order to reduce the number of optimizations to a reasonable amount, we chose a mixture of rebalancing periods of weeks or months and rolling windows of either 2 or 3 years (i.e. there are other portfolio optimizations that could have been created). Additionally, unless otherwise stated all of the portfolios utilized all 80 assets in the optimization:

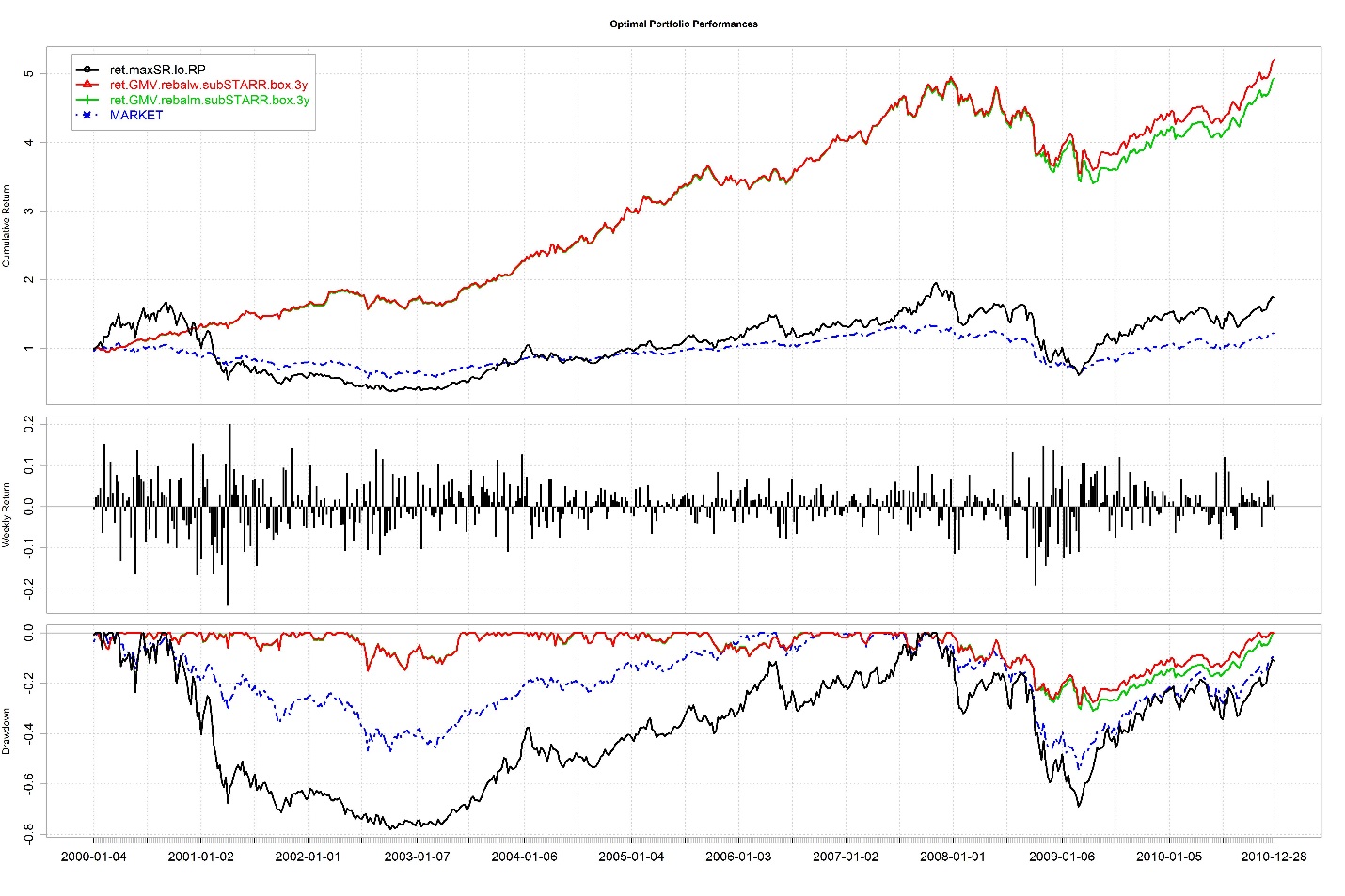
1. Long-only Minimum Variance
2. Box Constrained Minimum Variance
3. Long-only Maximizing the Sharpe-Ratio
4. Box Constrained Maximizing the Sharpe-Ratio
5. Long-only Maximizing the STARR-Ratio
6. Box Constrained Maximizing the STARR-Ratio
7. Maximum Sharpe Ratio from 100 Random Portfolios
8. The following rebalanced portfolios used the 16 assets that had absolute weights greater than 1% in the Minimum Variance portfolio chosen in the first optimization above.
   1. Long-only minimum variance rebalancing weekly 2 year rolling window
   2. Box constrained minimum variance rebalancing weekly 3 year rolling window
   3. Long-only minimum variance rebalancing monthly 3 year rolling window
   4. Box constrained minimum variance rebalancing monthly 2 year rolling window
   5. Box constrained maximize SR rebalancing weekly 2 year rolling window
   6. Box constrained maximize SR rebalancing monthly 2 year rolling window
9. The following rebalanced portfolios used the 20 assets that had the highest Sharpe Ratio over the period.
   1. Box constrained maximize SR rebalancing weekly 3 year rolling window
   2. Box constrained maximize SR rebalancing monthly 3 year rolling window
10. The following rebalanced portfolios used the 20 assets that had the highest STARR Ratio over the period.
    1. Long-only minimum variance rebalancing weekly 2 year rolling window
    2. Box constrained minimum variance rebalancing weekly 3 year rolling window
    3. Long-only minimum variance rebalancing monthly 2 year rolling window
    4. Box constrained minimum variance rebalancing monthly 3 year rolling window
    5. Box constrained maximize SR rebalancing weekly 2 year rolling window
    6. Box constrained maximize SR rebalancing monthly 3 year rolling window
11. Long-only Mean-ETL
12. Box constrained Mean-ETL
13. Maximize STARR Risk-Budgeted Portfolio (using DEoptim)

# Performance Charts and Statistics

The following performance charts were generated based on the PortfolioAnalytics optimization of each portfolio. Additionally the portfolio statistics for the Sharpe Ratio (SR), STARR ratio and Geometric Cumulative Return (GeomRet) were stored in a dataset providing easy numerical comparison of the strategies involved. It is also noted that due to the training windows/rolling windows the rebalancing portfolios did not produce any results until sometime after the beginning of the period of returns (i.e. this affected comparability of the returns).



Note the large peaks/volatility in the random portfolio and risk balanced portfolio.



This chart shows the performance for the three optimal portfolios. Note that due to the rebalancing portfolios, the chart and returns begin in 2000 rather than 1997. Also, the returns and drawdown on the two rebalancing portfolios appear to be performing better than the market.



The plot above shows the full period performance for the random portfolio maximizing the Sharpe ratio. Note the contrast between this and the plot comparing the optimal portfolios, which begins calculating returns in 2000.



Note the SR and STARR ratios generated for the non-rebalancing portfolios are calculated using the full period returns, whereas the rebalancing portfolios did not compute weights and therefore returns until after the beginning of the time series (i.e. due to the rolling window these portfolios don't compute optimal weights in the beginning of the time series).



The table above provides the performance statistics, along with the turnover and diversification, for the three optimal portfolios.

# Analysis – Optimal Portfolios

It appears that the maximum Sharpe ratio random portfolio produced the highest geometric returns, however since this return was calculated over the full period whereas the returns for the rebalancing portfolios were computed using a shorter period the comparison is lacking. Additionally, we can see that the drawdown on the chart is pretty significant for this portfolio. When we compare the rebalancing portfolios with the random portfolio starting at the later date the geometric returns become much more comparable (we should note that this is in the later half of the peak for the random portfolio).

It is clear that the volatility in the random portfolio is far greater than the optimal rebalancing portfolios. Based on the performance graphs above and the performance statistics dataset we can conclude that the optimal portfolio is the box-constrained minimum variance weekly rebalancing portfolio using the subset of 20 returns with the highest STARR ratio. We should also note that these statistics do not include the transaction costs, noting that for this same portfolio constraints but with monthly rebalancing the performance statistics are not significantly reduced we may wish to sacrifice to the monthly rebalancing due to transaction costs not considered in this report.

# Appendix – R Code

*#max Sharpe ratio*

library(xts)

library(PortfolioAnalytics)

library(ROI)

library(ROI.plugin.quadprog)

library(ROI.plugin.glpk)

library(DEoptim)

library(mpo)

library(lattice)

library(doParallel)

cl <- makeCluster(8)

registerDoParallel(cl)

*#read the data*

dat.largecap <- read.table(file="largecap\_weekly.csv",sep=",",header=TRUE,as.is=TRUE)

dat.microcap <- read.table(file="microcap\_weekly.csv",sep=",",header=TRUE,as.is=TRUE)

dat.midcap <- read.table(file="midcap\_weekly.csv",sep=",",header=TRUE,as.is=TRUE)

dat.smallcap <- read.table(file="smallcap\_weekly.csv",sep=",",header=TRUE,as.is=TRUE)

dat.largecap[,1]<-as.Date(dat.largecap[,1],format="%m/%d/%Y")

dat.microcap[,1]<-as.Date(dat.microcap[,1],format="%m/%d/%Y")

dat.midcap[,1]<-as.Date(dat.midcap[,1],format="%m/%d/%Y")

dat.smallcap[,1]<-as.Date(dat.smallcap[,1],format="%m/%d/%Y")

*#convert to zoo objects*

dat.largecap<-xts(dat.largecap[,-1],order.by=dat.largecap$DATE)

dat.microcap<-xts(dat.microcap[,-1],order.by=dat.microcap$DATE)

dat.midcap<-xts(dat.midcap[,-1],order.by=dat.midcap$DATE)

dat.smallcap<-xts(dat.smallcap[,-1],order.by=dat.smallcap$DATE)

*#remove duplicated columns and date (already captured in zoo index)*

dat.largecap2<-dat.largecap[,-c(21,22)]

dat.microcap2<-dat.microcap[,-c(21,22)]

dat.midcap2<-dat.midcap[,-c(21,22)]

*#merge returns*

dat.all<-merge.zoo(dat.largecap2,dat.microcap2,dat.midcap2,dat.smallcap,all=TRUE)

returns = dat.all[,1:80]*#[,1:80]*

names(returns)

MARKET = dat.largecap[,"VWMKT"]

RiskFree = dat.largecap[,"RiskFree"]

*#plot.zoo(returns, main = "SMALL-CAPS")*

funds = colnames(returns)

*#Performance Measurements to be used*

*#ETL Function*

etlf <- function(r, alpha=0.05) {

r <- sort(r); mean <- mean(r)

n.tail <- ifelse( alpha == 0, 1, ceiling(alpha\*length(r)))

-1/n.tail \* sum(r[which((1:length(r)) <= n.tail)])

}

*#normalized ETL Function*

netlf <- function(r, alpha=0.05) {

r <- sort(r); mean <- mean(r)

n.tail <- ifelse( alpha == 0, 1, ceiling(alpha\*length(r)))

etlr <- -1/n.tail \* sum(r[which((1:length(r)) <= n.tail)])

alpha\*(etlr + mean(r))/ dnorm(qnorm(alpha))

}

STARR <-function(r,rf,alpha=0.05){

er = r - rf

mean(er)/etlf(r)

}

TO = function(weights){

if(class(weights)=="numeric"){

return(0)

}

dates = index(weights)

weights=coredata(weights)

n.asset=ncol(weights)

n.dates=nrow(weights)

if(n.dates<2){

print("Less than 2 balancing dates!")

return()

}

TurnOver=rep(0,n.dates-1)

for(i in 1:length(TurnOver)){

TurnOver[i]=sum(abs(weights[i+1,]-weights[i,]))

}

dates=dates[-1]

res=zoo(TurnOver,order.by = dates)

res

}

*# DIV is the function to calculate the diversification*

*# Input: a zoo object containing the weights at balance dates*

*# Output: a zoo object containing the DIV values at the balance dates*

DIV = function(weights){

if(class(weights)=="numeric"){

return(1-sum(t(weights)%\*%weights))

}

n.dates=nrow(weights)

if(n.dates<1){

print("empty data set")

return()

}

diversification=rep(0,n.dates)

for(i in 1:n.dates){

diversification[i]=1-sum(weights[i,]^2)

}

dates=index(weights)

Div=zoo(diversification,dates)

return(Div)

}

PerfStats<-function(r,rf=RiskFree,period=52,alpha=0.05){

g = na.omit(r)

er = g - rf

GeomRet = Return.cumulative(g,geometric=TRUE)

STARR.port = STARR(g,rf,alpha) \* sqrt(period)

SR = mean(g)/sd(g) \* sqrt(period)

n = length(g)

sk=0

krt=0

sk = skewness(r)

krt = kurtosis(r)

sr.var=52\*(1 + .25\*(krt+2)\*SR^2-SR\*sk)/n

SESR = sqrt(sr.var)

stats=c(SR,SESR,STARR.port,GeomRet)

names(stats)=c("SR","Std Err SR","STARR","GeomRet")

stats

}

*#Individual Assets Performance*

individ.perfstats<-t(apply(returns,2,FUN=PerfStats))

individ.perfstats.STARR<-individ.perfstats[order(individ.perfstats[,"STARR"],decreasing=TRUE),]

individ.perfstats.SR<-individ.perfstats[order(individ.perfstats[,"SR"],decreasing=TRUE),]

optassets.STARR<-rownames(individ.perfstats.STARR[c(1:20),])

optassets.SR<-rownames(individ.perfstats.SR[c(1:20),])

ret.optassets.STARR = returns[,optassets.STARR]

ret.optassets.SR = returns[,optassets.SR]

*#initialize basic portfolio with constraints.*

init.portf <- portfolio.spec(assets=funds)

*#initialize mvo portfolios with leveraged constraints*

init.portf.mvo <- add.constraint(portfolio=init.portf, type="leverage",min\_sum=0.9, max\_sum=1.2)

init.portf.mvo <- add.objective(portfolio=init.portf.mvo, type="risk", name="var")

init.portf.mvo.box <- add.constraint(portfolio=init.portf.mvo, type="box", min = -0.1, max = 0.35)

init.portf.mvo.lo <-add.constraint(portfolio=init.portf.mvo, type="long\_only")

*#Optimize mvo portfolios*

mvo.lo.ROI <- optimize.portfolio(R=returns, portfolio=init.portf.mvo.lo,optimize\_method="ROI", trace=TRUE)

mvo.box.ROI <- optimize.portfolio(R=returns, portfolio=init.portf.mvo.box,optimize\_method="ROI", trace=TRUE)

wts.mvo.lo.ROI = extractWeights(mvo.lo.ROI)

wts.mvo.box.ROI = extractWeights(mvo.box.ROI)

ret.mvo.lo.ROI = Return.rebalancing(returns, wts.mvo.lo.ROI)

ret.mvo.box.ROI = Return.rebalancing(returns, wts.mvo.box.ROI)

*#extract mvo portfolios where weights are > 1% (use this to find assets to rebalance)*

optassets.mvo<-names(wts.mvo.lo.ROI[(abs(wts.mvo.lo.ROI) > 0.01)])

ret.optassets.mvo<-dat.all[,optassets.mvo]

*#initialize rebalancing mvo and maxSR portfolios with full investment constraints*

init.portf.rebal <- add.constraint(portfolio=init.portf, type="full\_investment")

init.portf.mvo.rebal <- add.objective(portfolio=init.portf.rebal, type="risk", name="var")

init.portf.mvo.rebal.box <- add.constraint(portfolio=init.portf.rebal, type="box", min = -0.1, max = 0.35)

init.portf.mvo.rebal.lo <- add.constraint(portfolio=init.portf.rebal, type="long\_only")

*#initialize Max Sharpe Portfolios Full investment*

init.portf.maxSR <- add.constraint(portfolio=init.portf, type="full\_investment")

init.portf.maxSR <- add.objective(portfolio=init.portf.maxSR, type="return", name="mean")

init.portf.maxSR <- add.objective(portfolio=init.portf.maxSR, type="risk", name="StdDev")

init.portf.maxSR.lo <- add.constraint(portfolio=init.portf.maxSR, type="long\_only")

init.portf.maxSR.box <- add.constraint(portfolio=init.portf.maxSR, type="box", min = -0.1, max = 0.35)

*#initialize Max STARR Portfolios full investment*

init.portf.maxSTARR <- add.constraint(portfolio=init.portf, type="full\_investment")

init.portf.maxSTARR <- add.constraint(portfolio=init.portf.maxSTARR, type="long\_only")

init.portf.maxSTARR <- add.objective(portfolio=init.portf, type="return", name="mean")

init.portf.maxSTARR <- add.objective(portfolio=init.portf.maxSTARR, type="risk", name="ES",

arguments=list(p=0.925))

init.portf.maxSTARR.lo <- add.constraint(portfolio=init.portf.maxSTARR, type="long\_only")

init.portf.maxSTARR.box <- add.constraint(portfolio=init.portf.maxSTARR, type="box", min = -0.1, max = 0.35)

*#initialize Mean ETL Portfolios*

init.portf.meanetl <- add.objective(portfolio=init.portf, type="return", name="mean")

init.portf.meanetl <- add.objective(portfolio=init.portf, type="risk", name="ES")

init.portf.meanetl.box <- add.constraint(portfolio=init.portf.meanetl, type="box", min = -0.1, max = 0.35)

init.portf.meanetl.lo <-add.constraint(portfolio=init.portf.meanetl, type="long\_only")

*#Optimize MaxSR portfolios*

maxSR.lo.ROI <- optimize.portfolio(R=returns, portfolio=init.portf.maxSR.lo,optimize\_method="ROI",maxSR=TRUE, trace=TRUE)

maxSR.box.ROI <- optimize.portfolio(R=returns, portfolio=init.portf.maxSR.box,optimize\_method="ROI",maxSR=TRUE, trace=TRUE)

wts.maxSR.lo.ROI = extractWeights(maxSR.lo.ROI)

wts.maxSR.box.ROI = extractWeights(maxSR.box.ROI)

ret.maxSR.lo.ROI = Return.rebalancing(returns, wts.maxSR.lo.ROI)

ret.maxSR.box.ROI = Return.rebalancing(returns, wts.maxSR.box.ROI)

objm.maxSR.lo.ROI = extractObjectiveMeasures(maxSR.lo.ROI)

objm.maxSR.box.ROI = extractObjectiveMeasures(maxSR.box.ROI)

*#Optimize MaxSTARR portfolios*

maxSTARR.lo.ROI <- optimize.portfolio(R=returns, portfolio=init.portf.maxSTARR.lo,

optimize\_method="ROI",

maxSTARR=TRUE, trace=TRUE)

maxSTARR.box.ROI <- optimize.portfolio(R=returns, portfolio=init.portf.maxSTARR.box,

optimize\_method="ROI",

maxSTARR=TRUE, trace=TRUE)

wts.maxSTARR.lo.ROI = extractWeights(maxSTARR.lo.ROI)

wts.maxSTARR.box.ROI = extractWeights(maxSTARR.box.ROI)

ret.maxSTARR.lo.ROI = Return.rebalancing(returns, wts.maxSTARR.lo.ROI)

ret.maxSTARR.box.ROI = Return.rebalancing(returns, wts.maxSTARR.box.ROI)

objm.maxSTARR.lo.ROI = extractObjectiveMeasures(maxSTARR.lo.ROI)

objm.maxSTARR.box.ROI = extractObjectiveMeasures(maxSTARR.box.ROI)

*#Optimize mvo Rebalancing portfolios using subset of returns chosen by mvo portfolio*

init.portf.rebal.subGMV = portfolio.spec(assets=optassets.mvo)

init.portf.rebal.subGMV = add.constraint(init.portf.rebal.subGMV, type="full\_investment")

init.portf.rebal.subGMV.lo = add.constraint(init.portf.rebal.subGMV, type="long\_only")

init.portf.rebal.subGMV.lo = add.objective(init.portf.rebal.subGMV.lo, type="risk", name="var")

init.portf.rebal.subGMV.box = add.constraint(init.portf.rebal.subGMV, type="box", min = -0.05, max = 0.35)

init.portf.rebal.subGMV.box = add.objective(init.portf.rebal.subGMV.box, type="risk", name="var")

bt.mvo.rebalw.subGMV.Lo.2y <- optimize.portfolio.rebalancing(ret.optassets.mvo, init.portf.rebal.subGMV.lo,

optimize\_method="ROI",rebalance\_on="weeks",

, search\_size=2000,training\_period=104,rolling\_window=104)

bt.mvo.rebalw.subGMV.box.3y <- optimize.portfolio.rebalancing(ret.optassets.mvo, init.portf.rebal.subGMV.box,

optimize\_method="ROI",rebalance\_on="weeks",

,training\_period=156,rolling\_window=156)

wts.mvo.rebalw.subGMV.lo.2y = extractWeights(bt.mvo.rebalw.subGMV.Lo.2y)

wts.mvo.rebalw.subGMV.box.3y = extractWeights(bt.mvo.rebalw.subGMV.box.3y)

ret.mvo.rebalw.subGMV.lo.2y = Return.rebalancing(ret.optassets.mvo, wts.mvo.rebalw.subGMV.lo.2y)

ret.mvo.rebalw.subGMV.box.3y = Return.rebalancing(ret.optassets.mvo, wts.mvo.rebalw.subGMV.box.3y)

bt.mvo.rebalm.subGMV.Lo.3y <- optimize.portfolio.rebalancing(ret.optassets.mvo, init.portf.rebal.subGMV.lo,

optimize\_method="ROI",rebalance\_on="months",

, search\_size=2000,training\_period=156,rolling\_window=156)

bt.mvo.rebalm.subGMV.box.2y <- optimize.portfolio.rebalancing(ret.optassets.mvo, init.portf.rebal.subGMV.box,

optimize\_method="ROI",rebalance\_on="months",

training\_period=104,rolling\_window=104)

wts.mvo.rebalm.subGMV.lo.3y = extractWeights(bt.mvo.rebalm.subGMV.Lo.3y)

wts.mvo.rebalm.subGMV.box.2y = extractWeights(bt.mvo.rebalm.subGMV.box.2y)

ret.mvo.rebalm.subGMV.lo.3y = Return.rebalancing(ret.optassets.mvo, wts.mvo.rebalm.subGMV.lo.3y)

ret.mvo.rebalm.subGMV.box.2y = Return.rebalancing(ret.optassets.mvo, wts.mvo.rebalm.subGMV.box.2y)

*#Optimize mvo Rebalancing portfolios using max STARR assets*

init.portf.rebal.subSTARR = portfolio.spec(assets=optassets.STARR)

init.portf.rebal.subSTARR = add.constraint(init.portf.rebal.subSTARR, type="full\_investment")

init.portf.rebal.subSTARR.lo = add.constraint(init.portf.rebal.subSTARR, type="long\_only")

init.portf.rebal.subSTARR.lo = add.objective(init.portf.rebal.subSTARR.lo, type="risk", name="var")

init.portf.rebal.subSTARR.box = add.constraint(init.portf.rebal.subSTARR, type="box", min = -0.05, max = 0.35)

init.portf.rebal.subSTARR.box = add.objective(init.portf.rebal.subSTARR.box, type="risk", name="var")

bt.mvo.rebalw.subSTARR.Lo.2y <- optimize.portfolio.rebalancing(ret.optassets.STARR, init.portf.rebal.subSTARR.lo,

optimize\_method="ROI",rebalance\_on="weeks",

training\_period=104,rolling\_window=104)

bt.mvo.rebalw.subSTARR.box.3y <- optimize.portfolio.rebalancing(ret.optassets.STARR, init.portf.rebal.subSTARR.box,

optimize\_method="ROI",rebalance\_on="weeks",

training\_period=156,rolling\_window=156)

wts.mvo.rebalw.subSTARR.lo.2y = extractWeights(bt.mvo.rebalw.subSTARR.Lo.2y)

wts.mvo.rebalw.subSTARR.box.3y = extractWeights(bt.mvo.rebalw.subSTARR.box.3y)

ret.mvo.rebalw.subSTARR.lo.2y = Return.rebalancing(ret.optassets.STARR, wts.mvo.rebalw.subSTARR.lo.2y)

ret.mvo.rebalw.subSTARR.box.3y = Return.rebalancing(ret.optassets.STARR, wts.mvo.rebalw.subSTARR.box.3y)

bt.mvo.rebalm.subSTARR.Lo.2y <- optimize.portfolio.rebalancing(ret.optassets.STARR, init.portf.rebal.subSTARR.lo,

optimize\_method="ROI",rebalance\_on="months",

training\_period=104,rolling\_window=104)

bt.mvo.rebalm.subSTARR.box.3y <- optimize.portfolio.rebalancing(ret.optassets.STARR, init.portf.rebal.subSTARR.box,

optimize\_method="ROI",rebalance\_on="months",

training\_period=156,rolling\_window=156)

wts.mvo.rebalm.subSTARR.lo.2y = extractWeights(bt.mvo.rebalm.subSTARR.Lo.2y)

wts.mvo.rebalm.subSTARR.box.3y = extractWeights(bt.mvo.rebalm.subSTARR.box.3y)

ret.mvo.rebalm.subSTARR.lo.2y = Return.rebalancing(ret.optassets.STARR, wts.mvo.rebalm.subSTARR.lo.2y)

ret.mvo.rebalm.subSTARR.box.3y = Return.rebalancing(ret.optassets.STARR, wts.mvo.rebalm.subSTARR.box.3y)

*#initialize rebalancing maxSR portfolios with full investment constraints*

init.portf.maxSR.subGMV.rebal <- portfolio.spec(assets=optassets.mvo)

init.portf.maxSR.subGMV.rebal <- add.constraint(portfolio=init.portf.maxSR.subGMV.rebal, type="full\_investment")

init.portf.maxSR.subGMV.rebal <- add.constraint(portfolio=init.portf.maxSR.subGMV.rebal, type="box",min=-.03,max=0.35)

init.portf.maxSR.subGMV.rebal <- add.objective(init.portf.maxSR.subGMV.rebal, type="quadratic\_utility")

*#initialize rebalancing maxSR portfolios with full investment constraints*

init.portf.maxSR.subSTARR.rebal <- portfolio.spec(assets=optassets.STARR)

init.portf.maxSR.subSTARR.rebal <- add.constraint(portfolio=init.portf.maxSR.subSTARR.rebal, type="full\_investment")

init.portf.maxSR.subSTARR.rebal <- add.constraint(portfolio=init.portf.maxSR.subSTARR.rebal, type="box",min=-.03,max=0.35)

init.portf.maxSR.subSTARR.rebal <- add.objective(init.portf.maxSR.subSTARR.rebal, type="quadratic\_utility")

*#initialize rebalancing maxSR portfolios with full investment constraints*

init.portf.maxSR.subSR.rebal <- portfolio.spec(assets=optassets.SR)

init.portf.maxSR.subSR.rebal <- add.constraint(portfolio=init.portf.maxSR.subSR.rebal, type="full\_investment")

init.portf.maxSR.subSR.rebal <- add.constraint(portfolio=init.portf.maxSR.subSR.rebal, type="box",min=-.03,max=0.35)

init.portf.maxSR.subSR.rebal <- add.objective(init.portf.maxSR.subSR.rebal, type="quadratic\_utility")

*#Optimize maxSR Rebalancing portfolios with mvo portfolio*

bt.maxSR.box.subGMV.rm.2y <- optimize.portfolio.rebalancing(ret.optassets.mvo, init.portf.maxSR.subGMV.rebal,

optimize\_method="quadprog",rebalance\_on="months",search\_size = 200

,training\_period=104,rolling\_window=104

,maxSR=TRUE,trace=TRUE)

wts.maxSR.box.subGMV.rm.2y = extractWeights(bt.maxSR.box.subGMV.rm.2y)

ret.maxSR.box.rebalm.subGMV.2y = Return.rebalancing(ret.optassets.mvo, wts.maxSR.box.subGMV.rm.2y)

bt.maxSR.box.subGMV.rw.2y <- optimize.portfolio.rebalancing(ret.optassets.mvo, init.portf.maxSR.subGMV.rebal,

optimize\_method="quadprog",rebalance\_on="weeks",search\_size = 300,

,training\_period=104,rolling\_window=104,

,maxSR=TRUE,trace=TRUE)

wts.maxSR.box.subGMV.rw.2y = extractWeights(bt.maxSR.box.subGMV.rw.2y)

ret.maxSR.box.rebalw.subGMV.2y = Return.rebalancing(ret.optassets.mvo, wts.maxSR.box.subGMV.rw.2y)

*#Optimize maxSR Rebalancing portfolios with maxSTARR portfolio*

bt.maxSR.box.subSTARR.rw.2y <- optimize.portfolio.rebalancing(ret.optassets.STARR, init.portf.maxSR.subSTARR.rebal,

optimize\_method="quadprog",rebalance\_on="weeks",search\_size = 300

,training\_period=104,rolling\_window=104

,maxSR=TRUE,trace=TRUE)

wts.maxSR.box.subSTARR.rw.2y = extractWeights(bt.maxSR.box.subSTARR.rw.2y)

ret.maxSR.box.rebalw.subSTARR.2y = Return.rebalancing(ret.optassets.STARR, wts.maxSR.box.subSTARR.rw.2y)

bt.maxSR.box.subSTARR.rm.3y <- optimize.portfolio.rebalancing(ret.optassets.STARR, init.portf.maxSR.subSTARR.rebal,

optimize\_method="quadprog",rebalance\_on="months",search\_size = 300

,training\_period=156,rolling\_window=156,

,maxSR=TRUE,trace=TRUE)

wts.maxSR.box.subSTARR.rm.3y = extractWeights(bt.maxSR.box.subSTARR.rm.3y)

ret.maxSR.box.rebalm.subSTARR.3y = Return.rebalancing(ret.optassets.STARR, wts.maxSR.box.subSTARR.rm.3y)

*#Optimize maxSR Rebalancing portfolios with maxSR portfolio*

bt.maxSR.box.subSR.rm.2y <- optimize.portfolio.rebalancing(ret.optassets.SR, init.portf.maxSR.subSR.rebal,

optimize\_method="quadprog",rebalance\_on="months",search\_size = 200

,training\_period=104,rolling\_window=104

,maxSR=TRUE,trace=TRUE)

wts.maxSR.box.subSR.rm.2y = extractWeights(bt.maxSR.box.subSR.rm.2y)

ret.maxSR.box.rebalm.subSR.2y = Return.rebalancing(ret.optassets.SR, wts.maxSR.box.subSR.rm.2y)

bt.maxSR.box.subSR.rw.3y <- optimize.portfolio.rebalancing(ret.optassets.SR, init.portf.maxSR.subSR.rebal,

optimize\_method="quadprog",rebalance\_on="weeks",search\_size = 300

,training\_period=156,rolling\_window=156,

,maxSR=TRUE,trace=TRUE)

wts.maxSR.box.subSR.rw.3y = extractWeights(bt.maxSR.box.subSR.rw.3y)

ret.maxSR.box.rebalw.subSR.3y = Return.rebalancing(ret.optassets.SR, wts.maxSR.box.subSR.rw.3y)

*#Optimize mean-etl portfolios*

meanetl.lo <- meanetl.efficient.frontier(init.portf.meanetl.lo ,returns,n.portfolios=25)

meanetl.box <- meanetl.efficient.frontier(init.portf.meanetl.box ,returns,n.portfolios=25)

wts.meanetl.lo.opt <- meanetl.lo[1,]

wts.meanetl.lo.opt <- wts.meanetl.lo.opt[-c(1:3)]

wts.meanetl.box.opt <- meanetl.box[1,]

wts.meanetl.box.opt <- wts.meanetl.box.opt[-c(1:3)]

ret.meanetl.lo.opt <- Return.rebalancing(returns, wts.meanetl.lo.opt)

ret.meanetl.box.opt <- Return.rebalancing(returns, wts.meanetl.box.opt)

*#initialize Risk Parity/Risk Budgeted Portfolio*

init.portf.rb <- portfolio.spec(assets=optassets.STARR)

init.portf.rb <- add.constraint(portfolio=init.portf.rb, type="full\_investment")

init.portf.rb <- add.objective(portfolio=init.portf.rb, type="return", name="mean")

init.portf.rb <- add.objective(portfolio=init.portf.rb, type="risk\_budget", name="ETL", arguments=list(p=0.95), max\_prisk=0.3)

init.portf.box.rb <- add.constraint(portfolio=init.portf.rb, type="box", min = -0.1, max = 0.35)

*#optimize risk budget portfolio using DEoptim*

init.portf.box.rb$constraints[[1]]$min\_sum=0.99

init.portf.box.rb$constraints[[1]]$max\_sum=1.01

optrb.box <- optimize.portfolio(R=ret.optassets.STARR, portfolio=init.portf.box.rb,

optimize\_method="DEoptim",

search\_size =100, trace=TRUE,traceDE=5)

wts.optrb.box = extractWeights(optrb.box)

ret.optrb.box <- Return.rebalancing(ret.optassets.STARR, wts.optrb.box)

*# Random portfolios*

maxSR.lo.RP <- optimize.portfolio(R=ret.optassets.STARR, portfolio=init.portf.box.rb,

optimize\_method="random",

search\_size=100,

trace=TRUE)

chart.RiskReward(maxSR.lo.RP, risk.col="ETL", return.col="mean")

wts.maxSR.lo.RP = extractWeights(maxSR.lo.RP)

ret.maxSR.lo.RP = Return.rebalancing(ret.optassets.STARR, wts.maxSR.lo.RP)

objm.maxSR.lo.RP = extractObjectiveMeasures(maxSR.lo.RP)

*# Use DEoptim to run the optimization.*

maxSR.lo.DE <- optimize.portfolio(R=returns, portfolio=init.portf.box.rb,

optimize\_method="DEoptim",

search\_size=10,

trace=TRUE)

maxSR.lo.DE

chart.RiskReward(maxSR.lo.DE, risk.col="ETL", return.col="mean")

wts.maxSR.lo.DE = extractWeights(maxSR.lo.DE)

ret.maxSR.lo.DE = Return.rebalancing(ret.optassets.STARR, wts.maxSR.lo.DE)

objm.maxSR.lo.DE = extractObjectiveMeasures(maxSR.lo.DE)

*#give names to returrns vectors*

colnames(ret.mvo.lo.ROI)<-"ret.mvo.lo.ROI"

colnames(ret.mvo.box.ROI)<-"ret.mvo.box.ROI"

colnames(ret.maxSR.lo.ROI)<-"ret.maxSR.lo.ROI"

colnames(ret.maxSR.box.ROI)<-"ret.maxSR.box.ROI"

colnames(ret.maxSTARR.lo.ROI)<-"ret.maxSTARR.lo.ROI"

colnames(ret.maxSTARR.box.ROI)<-"ret.maxSTARR.box.ROI"

colnames(ret.meanetl.lo.opt)<-"ret.meanetl.lo.opt"

colnames(ret.meanetl.box.opt)<-"ret.meanetl.box.opt"

colnames(ret.mvo.rebalw.subGMV.lo.2y)<-"ret.mvo.rebalw.subGMV.lo.2y"

colnames(ret.mvo.rebalw.subGMV.box.3y)<-"ret.mvo.rebalw.subGMV.box.3y"

colnames(ret.mvo.rebalw.subSTARR.lo.2y)<-"ret.mvo.rebalw.subSTARR.lo.2y"

colnames(ret.mvo.rebalw.subSTARR.box.3y)<-"ret.mvo.rebalw.subSTARR.box.3y"

colnames(ret.mvo.rebalm.subGMV.box.2y)<-"ret.mvo.rebalm.subGMV.lo.3y"

colnames(ret.mvo.rebalm.subGMV.box.2y)<-"ret.mvo.rebalm.subGMV.box.2y"

colnames(ret.mvo.rebalm.subSTARR.lo.2y)<-"ret.mvo.rebalm.subSTARR.lo.2y"

colnames(ret.mvo.rebalm.subSTARR.box.3y)<-"ret.mvo.rebalm.subSTARR.box.3y"

colnames(ret.maxSR.box.rebalw.subGMV.2y)<-"ret.maxSR.box.rebal.subGMV.2y"

colnames(ret.maxSR.box.rebalw.subSTARR.2y)<-"ret.maxSR.box.rebal.subSTARR.2y"

colnames(ret.maxSR.box.rebalw.subSR.3y)<-"ret.maxSR.box.rebal.subSR.3y"

colnames(ret.maxSR.box.rebalm.subGMV.2y)<-"ret.maxSR.box.rebalm.subGMV.2y"

colnames(ret.maxSR.box.rebalm.subSTARR.3y)<-"ret.maxSR.box.rebalm.subSTARR.3y"

colnames(ret.maxSR.box.rebalm.subSR.2y)<-"ret.maxSR.box.rebalm.subSR.2y"

colnames(ret.optrb.box)<-"ret.optrb.box"

colnames(ret.maxSR.lo.RP)<-"ret.maxSR.lo.RP"

colnames(ret.maxSR.lo.DE)<-"ret.maxSR.lo.DE"

colnames(MARKET)<-"MARKET"

*#save as sep variable without removing NA so we can calculate proper performance stats for non-rebalancing portfolios*

ret.comb.temp <- merge(ret.mvo.lo.ROI, ret.mvo.box.ROI,

ret.maxSR.lo.ROI,ret.maxSR.box.ROI,

ret.maxSTARR.lo.ROI,ret.maxSTARR.box.ROI,

ret.meanetl.lo.opt, ret.meanetl.box.opt,

ret.mvo.rebalw.subGMV.lo.2y, ret.mvo.rebalw.subGMV.box.3y,

ret.mvo.rebalw.subSTARR.lo.2y, ret.mvo.rebalw.subSTARR.box.3y,

ret.maxSR.box.rebalw.subGMV.2y, ret.maxSR.box.rebalw.subSTARR.2y,

ret.maxSR.box.rebalw.subSR.3y,

ret.mvo.rebalm.subGMV.lo.3y, ret.mvo.rebalm.subGMV.box.2y,

ret.mvo.rebalm.subSTARR.lo.2y, ret.mvo.rebalm.subSTARR.box.3y,

ret.maxSR.box.rebalm.subGMV.2y, ret.maxSR.box.rebalm.subSTARR.3y,

ret.maxSR.box.rebalm.subSR.2y,

ret.optrb.box, ret.maxSR.lo.RP)

ret.comb <- merge(ret.comb.temp,MARKET)

jpeg(file="PerformanceSummary.jpg",width=15,height=10,units="in",res=400) *#open png file*

charts.PerformanceSummary(ret.comb,main="All Portfolio Performances",wealth.index = T,

lty = c(rep(1,ncol(ret.comb)-1),4), colorset = c(seq(rep(1,ncol(ret.comb)))),

cex.legend = 0.9,cex.axis = 1.3)

dev.off()

ret.comb <- na.omit(merge(ret.comb.temp,MARKET,all=F))

pspec.list = combine.portfolios(list(init.portf.mvo.lo, init.portf.mvo.box,

init.portf.maxSR.lo,init.portf.maxSR.box,

init.portf.maxSTARR.lo,init.portf.maxSTARR.box,

init.portf.rebal.lo, init.portf.rebal.box,

init.portf.meanetl.lo,init.portf.meanetl.box,

init.portf.rebalw.lo,init.portf.rebalw.box,

init.portf.rebalw.subGMV.lo,init.portf.rebalw.subGMV.box,

init.portf.rebalw.subSTARR.box,init.portf.rebalw.subSR.box,

init.portf.box.rb))

*#chart.EfficientFrontierOverlay(returns, pspec.list,*

*# type="mean-StdDev",match.col="StdDev",*

*# legend.loc="topleft",*

*# legend.labels=legend.labels,*

*# cex.legend=0.8,labels.assets=T,xlim =*

*# c(.001,.14), lty = c(1,2,5),lwd =*

*# c(1.3,1.7,1.7))*

PerformStats.nontruncated<-t(apply(ret.comb.temp,2,PerfStats))

PerformStats.truncated<-t(apply(ret.comb,2,PerfStats))

*#sort performance stats by optimal performance measures*

PerformStats.nontruncated.sortSR = PerformStats.nontruncated[order(PerformStats.nontruncated[,"SR"],decreasing=TRUE),]

PerformStats.nontruncated.sortSTARR = PerformStats.nontruncated[order(PerformStats.nontruncated[,"STARR"],decreasing=TRUE),]

PerformStats.nontruncated.sortGeomRet = PerformStats.nontruncated[order(PerformStats.nontruncated[,"GeomRet"],decreasing=TRUE),]

write.csv(PerformStats.nontruncated.sortSR,file="perfstats.csv",append=FALSE,col.names=TRUE,row.names=TRUE)

*#maxSTARR.ret = ret.comb.temp[,which.max(PerformStats.nontruncated[,"STARR"])]*

*#maxGeomRet.ret = ret.comb.temp[,which.max(PerformStats.nontruncated[,"GeomRet"])]*

maxSR.ret = ret.comb.temp[,which.max(PerformStats.nontruncated[,"SR"])]

maxSTARR.ret = ret.comb.temp[,which.max(PerformStats.nontruncated[,"STARR"])]

*#If maxSTARR portfolio is same as maxSR choose 2nd best maxSTARR*

if (all.equal(maxSR.ret,maxSTARR.ret)){

which.max(PerformStats.nontruncated[,"SR"])

ord<-order(PerformStats.nontruncated[,"STARR"],decreasing=TRUE)

maxSTARR.ret=ret.comb.temp[,ord[2]]

}

maxGeomRet.ret = ret.comb.temp[,which.max(PerformStats.nontruncated[,"GeomRet"])]

maxstatreturns = na.omit(merge(maxGeomRet.ret,maxSR.ret,maxSTARR.ret,MARKET,all=F))

maxstatreturns.perfstats = PerformStats.nontruncated[colnames(maxstatreturns[,-4]),]

wts.opt.names<-paste("wts.",sep="",substr(colnames(maxstatreturns)[c(1:3)],5,30))

div.optport.1 = mean(DIV(eval(parse(text=wts.opt.names[1]))))

div.optport.2 = mean(DIV(eval(parse(text=wts.opt.names[2]))))

div.optport.3 = mean(DIV(eval(parse(text=wts.opt.names[3]))))

to.optport.1 = mean(TO(eval(parse(text=wts.opt.names[1]))))

to.optport.2 = mean(TO(eval(parse(text=wts.opt.names[2]))))

to.optport.3 = mean(TO(eval(parse(text=wts.opt.names[3]))))

DIVTO.MAT<-cbind(c(div.optport.1,div.optport.2,div.optport.3),c(to.optport.1,to.optport.2,to.optport.3))

colnames(DIVTO.MAT)=c("DIV","TO")

maxstatreturns.perfstats = cbind(maxstatreturns.perfstats,DIVTO.MAT)

write.csv(maxstatreturns.perfstats,file="optperfstats.csv",append=FALSE,col.names=TRUE,row.names=TRUE)

jpeg(file="PerformanceSummaryretmaxSR.jpg",width=15,height=10,units="in",res=400) *#open png file*

charts.PerformanceSummary(ret.optrb.box,main="Optimal Portfolio Performances",wealth.index = T,

lty = 1, colorset = 1,

cex.legend = 1.3,cex.axis = 1.3)

dev.off()

jpeg(file="PerformanceSummaryOptimal.jpg",width=15,height=10,units="in",res=400) *#open png file*

charts.PerformanceSummary(maxstatreturns,main="Optimal Portfolio Performances",wealth.index = T,

lty = c(rep(1,ncol(maxstatreturns)-1),4), colorset = c(seq(rep(1,ncol(maxstatreturns)))),

cex.legend = 1.3,cex.axis = 1.3)

dev.off()