

Global Rotation Strategy

Philipp Epstein

Introduction

Trading Idea

Global Rotation Strategy

For the design of a well performing rotation strategy, it is important that the selected ETFs are not too volatile, show longer-term visible trends and have a good market volume, so that they cannot be manipulated. They all should have more or less the same volatility. The 5 global markets ETFs fulfill this condition. They all are capitalized enough, so that they cannot be manipulated in the short term. They follow slightly different economic cycles and have long periods where one market outperforms the others until it becomes so overpriced and investors begin to remove their money from that market in order to invest in other cheaper valued markets.

Implementation of the trading strategy

Part A: Initialization

Step 1: General Setup

Pre-Step: Clear the environment.

```
rm(list=ls()) # clear the workspace
```

Define Currency, Instruments, Time zone and other simulation parameters. As well as Currency, Instruments and Time zone, you need to set starting and ending dates for the simulation, names for the portfolio and the account (often both are set to the strategy name), and the amount of starting capital.

```
library(quantstrat)
currency("USD")
Sys.setenv(TZ="UTC")
initdate <- '2002-12-31'
startdate <- '2003-01-01'
enddate <- '2018-12-31'
portfolioname <- "GRS"
accountname <- portfolioname
strategyname <- portfolioname
```

Step 2: Setting the variables

```
#Set values:
```

```
startingcapital <- 10000
txncost = -70

buypositions <- 1
roc.var <- 3
```

```
dir <- "~/Desktop/R/GRS/"
daterange <- '2003::2018'
```

Step 3: Preset for plotting graphs

```
myTheme <- chart_theme()
myTheme$col$dn.col <- 'red'
myTheme$col$up.col <- 'lightblue'
myTheme$col$dn.border <- 'lightgray'
myTheme$col$up.border <- 'lightgray'
```

Step 4: Load and transform data

Loading data into R environment. Transform it to monthly data, use only Close prices and change the columnname to the name of the symbol used.

```
library(INFT361Course)
symbollist <- GetCourseDirList(dir)

# Load data into the environment
LoadCourseDir(dir)

# transform data to monthly data
for (symbol in symbollist)
{
  symname <- symbol
  stock(symbollist, currency="USD")
  # convert data to monthly
  x <- get(symbol)
  x = to.monthly(x, indexAt="startof", drop.time=FALSE, OHLC=TRUE) # needs "endof" and drop.time=FALSE
  indexFormat(x) <- '%Y-%m-%d'
  x <- x[, "x.Close"]
  colnames(x) <- gsub("x.", "", colnames(x)) # Change columnnames to Symbol.Open etc.
  colnames(x) <- symbol

  assign(symbol, x)
}
```

Merge all symbols together

Merge all symbols together in order to perform the ranking of the ETFs.

```
symbols_close <- do.call(merge, lapply(symbollist, get))
```

Calculate ROC

Calculate the ROC for the last n months for the used symbols and store it for ranking in new roc matrix.

```
roc <- ROC(symbols_close, n=roc.var, type="discrete")
roc <- na.omit(roc)
```

Rank ETFs

Rank the ETFs based on their rate of change and store the ranks in new ranks matrix.

```
ranks <- as.xts(t(apply(-roc,1,rank)))
```

Identify positiv returns

Create new matrix pos_rets for the ETFs which assigns a 0 to negative returns and a 1 to positive returns. This is needed to be able to easy access the information about the returns at a specific date.

```
pos_rets <- roc
pos_rets[pos_rets<=0] <- 0
pos_rets[pos_rets>0] <- 1
```

Merge all data together

Merge the created matrices together with the original data to have all data of the symbols available together with the created indicators (ranks and pos_rets).

```
# Load all original files
LoadCourseDir(dir)

# Convert the symbols to monthly data
for (symbol in symbollist)
{
  symname <- symbol
  stock(symbollist,currency="USD")
  # convert data to monthly
  x<-get(symbol)
  x=to.monthly(x,indexAt="startof",drop.time=FALSE, OHLC=TRUE)
  indexFormat(x) <- '%Y-%m-%d'
  colnames(x)<-gsub("x.", "", colnames(x))

  assign(symbol,x)
}

# lag the ranks by one period as the calculations
# are based on the Close price of the month
# so the rank is the indicator for the next month
# to perform the buy at the opening price

ranks <- lag(ranks,k=1,na.pad=TRUE)
ranks <- na.omit(ranks)

# merge the ranks values to their respective original symbols
for (col in colnames(ranks))
{
  x <- get(col)
  x <- merge(x,ranks[,col])
  colnames(x)[colnames(x) == col] <- "Rank"
  x<-na.omit(x)
  assign(col,x)
}

# merge the returns indicator values back to their original symbols
for (col in colnames(pos_rets))
{
  x <- get(col)
```

```

x <- merge(x,pos_rets[,col])
colnames(x)[colnames(x) == col] <- "PosRets"
x<-na.omit(x)
assign(col,x)
}

```

Step 5: Order sizing function

The order sizing function is used to ensure that money is equally split between the investment products

```

# order sizing function
osInvestAll <- function (data, timestamp, orderqty, ordertype, orderside, portfolio, symbol, ruletype,
{
  datePos <- format(timestamp,"%Y-%m-%d")

  updatePortf(Portfolio=portfolio,Symbol=symbol,Dates=datePos)
  updateAcct(portfolio,Dates=datePos)
  updateEndEq(portfolio,Dates=datePos)

  Posn <- getPosQty(portfolio,Symbol=symbol,Date=datePos)
  equity <- getEndEq(portfolio,datePos) / buypositions

  OpenPrice <- getPrice(get(symbol), prefer="Open")[datePos]

  UnitSize <- as.numeric(trunc(equity/OpenPrice))

  # check whether orderside is long or short and return
  # either positive value for the unitsize or a negative value.

  if (Posn == 0) {

    if (orderside == "long"){
      print(orderside)
      osInvestAll <- UnitSize
    } else{
      print(orderside)
      osInvestAll <- -UnitSize
    }
  }else
    {osInvestAll <- 0
    }
}

```

Part B: Apply Strategy

Step 1: Create Strategy, Portfolio and Account

```

rm.strat(strategyname) # remove if it already exists
initPortf(portfolioname,symbolist,initDate=initdate,currency="USD")
initAcct(accountname,portfolios=portfolioname,initDate=initdate,initEq=startingcapital,currency="USD")
initOrders(portfolio=portfolioname,initDate=initdate)
strategy(strategyname,store=TRUE)
strat <- getStrategy(strategyname)

```

Step 2: Define indicators, signals and rules

The original strategy is based on the simple idea that the ETFs are ranked by the ROC (return of the last n months). The best ETF is bought and hold for one month. Therefore two signals and two rules are created:

Signal 1: If the Rank is lower or equal the Buypositions (which is 1 in case we want to buy only one ETF) then the signal is set to TRUE.

Signal 2: If the Rank is above the buyposition variable the signal is set to TRUE.

Rule 1: Long Entry rule which buys the ETF when the longEntry Signal is TRUE. Buy is performed at the Open price and osInvestAll function is used to calculate the ordersize.

Rule 2: Exit rule for the strategy. Based on the longExit signal the rule exits the current position and sells the entire amount in the portfolio.

```
add.signal(strategyname,
            name="sigThreshold",
            arguments=list(column="Rank",
                           threshold = buypositions,
                           relationship="lte"),
            label="longEntry")

add.signal(strategyname,
            name="sigThreshold",
            arguments=list(column="Rank",
                           threshold=buypositions,
                           relationship="gt"),
            label="longExit")

add.rule(strategyname,
          name="ruleSignal",
          arguments=list(sigcol="longEntry",
                         sigval=TRUE,
                         orderqty=100,
                         ordertype="market",
                         orderside="long",
                         replace=FALSE,
                         prefer="Open",
                         osFUN='osInvestAll',
                         TxnFees=txncost,
                         orderset="ocolong"),
          type="enter",
          path.dep=TRUE,
          label="enterLONG")

add.rule(strategyname,
          name="ruleSignal",
          arguments=list(sigcol="longExit",
                         sigval=TRUE,
                         orderqty="all",
                         ordertype="market",
                         orderside="long",
                         replace=FALSE,
                         prefer="Open",
                         TxnFees=txncost,
                         orderset="ocolong"),
```

```

        type="exit",
        path.dep=TRUE,
        label="exitLONG")

###
# STOP LOSS RULE
###
# stopLossPercent <- 0.001
#
# add.rule(strategyname, name = "ruleSignal", arguments = list(sigcol="longEntry", signal=TRUE, orderqt
###
# STOP LOSS RULE
###

```

Step 3: Run the strategy

```
applyStrategy(strategy=strategyname,portfolios=portfolioname)
```

```

## Warning in .updatePosPL(Portfolio = pname, Symbol = as.character(symbol), :
## Could not parse 2018-11-01 as ISO8601 string, or one/bothends of the range
## were outside the available prices: 2003-09-02/2018-11-01. Using 2018-11-01/
## instead.

```

```

updatePortf(strategyname)
updateAcct(strategyname)
updateEndEq(strategyname)

```

Step 4: Check orderbook

Check the orderbook for some symbols in order to make sure that the orders are palced correctly and as intended.

```

subset <- "2007::20010"
ob <- getOrderBook(portfolioname)
ob$`GRS`$MDY.csv[subset,]

```

	Order.Qty	Order.Price	Order.Type	Order.Side
## 2007-03-01 00:00:00	"74"	"151.38"	"market"	"long"
## 2007-04-02 00:00:00	"all"	"154.64"	"market"	"long"
## 2008-08-01 00:00:00	"77"	"146.17"	"market"	"long"
## 2008-12-01 00:00:00	"all"	"90.94"	"market"	"long"
## 2010-02-01 00:00:00	"55"	"127.8"	"market"	"long"
## 2010-08-02 00:00:00	"all"	"140.34"	"market"	"long"
## 2010-12-01 00:00:00	"46"	"157.43"	"market"	"long"
## 2011-03-01 00:00:00	"all"	"176.22"	"market"	"long"
## 2011-04-01 00:00:00	"43"	"180.57"	"market"	"long"
## 2011-05-02 00:00:00	"all"	"184.84"	"market"	"long"
## 2011-10-03 00:00:00	"55"	"140.97"	"market"	"long"
## 2012-03-01 00:00:00	"all"	"178.43"	"market"	"long"
## 2012-04-02 00:00:00	"48"	"180.47"	"market"	"long"
## 2012-08-01 00:00:00	"all"	"171.82"	"market"	"long"
## 2013-02-01 00:00:00	"41"	"199.69"	"market"	"long"
## 2013-10-01 00:00:00	"all"	"225.99"	"market"	"long"
## 2014-01-02 00:00:00	"38"	"243.31"	"market"	"long"

##	2014-04-01 00:00:00	"all"	"251.25"	"market"	"long"
##	2014-11-03 00:00:00	"37"	"258.71"	"market"	"long"
##	2015-05-01 00:00:00	"all"	"273.68"	"market"	"long"
##	2015-08-03 00:00:00	"37"	"273.95"	"market"	"long"
##	2016-01-04 00:00:00	"all"	"249.92"	"market"	"long"
##	2016-02-01 00:00:00	"42"	"238.03"	"market"	"long"
##	2016-04-01 00:00:00	"all"	"260.75"	"market"	"long"
##	2016-08-01 00:00:00	"38"	"283.93"	"market"	"long"
##	2016-09-01 00:00:00	"all"	"285.26"	"market"	"long"
##	2016-12-01 00:00:00	"36"	"297.66"	"market"	"long"
##	2017-03-01 00:00:00	"all"	"317.83"	"market"	"long"
##	2017-12-01 00:00:00	"32"	"346.06"	"market"	"long"
##	2018-02-01 00:00:00	"all"	"354.28"	"market"	"long"
##	2018-05-01 00:00:00	"34"	"339.88"	"market"	"long"
##	2018-10-01 00:00:00	"all"	"368.93"	"market"	"long"
##		Order.Threshold	Order.Status	Order.StatusTime	
##	2007-03-01 00:00:00	NA	"closed"	"2007-04-02 00:00:00"	
##	2007-04-02 00:00:00	NA	"closed"	"2007-05-01 00:00:00"	
##	2008-08-01 00:00:00	NA	"closed"	"2008-09-02 00:00:00"	
##	2008-12-01 00:00:00	NA	"closed"	"2009-01-02 00:00:00"	
##	2010-02-01 00:00:00	NA	"closed"	"2010-03-01 00:00:00"	
##	2010-08-02 00:00:00	NA	"closed"	"2010-09-01 00:00:00"	
##	2010-12-01 00:00:00	NA	"closed"	"2011-01-03 00:00:00"	
##	2011-03-01 00:00:00	NA	"closed"	"2011-04-01 00:00:00"	
##	2011-04-01 00:00:00	NA	"closed"	"2011-05-02 00:00:00"	
##	2011-05-02 00:00:00	NA	"closed"	"2011-06-01 00:00:00"	
##	2011-10-03 00:00:00	NA	"closed"	"2011-11-01 00:00:00"	
##	2012-03-01 00:00:00	NA	"closed"	"2012-04-02 00:00:00"	
##	2012-04-02 00:00:00	NA	"closed"	"2012-05-01 00:00:00"	
##	2012-08-01 00:00:00	NA	"closed"	"2012-09-04 00:00:00"	
##	2013-02-01 00:00:00	NA	"closed"	"2013-03-01 00:00:00"	
##	2013-10-01 00:00:00	NA	"closed"	"2013-11-01 00:00:00"	
##	2014-01-02 00:00:00	NA	"closed"	"2014-02-03 00:00:00"	
##	2014-04-01 00:00:00	NA	"closed"	"2014-05-01 00:00:00"	
##	2014-11-03 00:00:00	NA	"closed"	"2014-12-01 00:00:00"	
##	2015-05-01 00:00:00	NA	"closed"	"2015-06-01 00:00:00"	
##	2015-08-03 00:00:00	NA	"closed"	"2015-09-01 00:00:00"	
##	2016-01-04 00:00:00	NA	"closed"	"2016-02-01 00:00:00"	
##	2016-02-01 00:00:00	NA	"closed"	"2016-03-01 00:00:00"	
##	2016-04-01 00:00:00	NA	"closed"	"2016-05-02 00:00:00"	
##	2016-08-01 00:00:00	NA	"closed"	"2016-09-01 00:00:00"	
##	2016-09-01 00:00:00	NA	"closed"	"2016-10-03 00:00:00"	
##	2016-12-01 00:00:00	NA	"closed"	"2017-01-03 00:00:00"	
##	2017-03-01 00:00:00	NA	"closed"	"2017-04-03 00:00:00"	
##	2017-12-01 00:00:00	NA	"closed"	"2018-01-02 00:00:00"	
##	2018-02-01 00:00:00	NA	"closed"	"2018-03-01 00:00:00"	
##	2018-05-01 00:00:00	NA	"closed"	"2018-06-01 00:00:00"	
##	2018-10-01 00:00:00	NA	"closed"	"2018-11-01 00:00:00"	
##		Prefer	Order.Set	Txn.Fees	Rule Time.In.Force
##	2007-03-01 00:00:00	"Open"	"ocolong"	"-70"	"enterLONG" ""
##	2007-04-02 00:00:00	"Open"	"ocolong"	"-70"	"exitLONG" ""
##	2008-08-01 00:00:00	"Open"	"ocolong"	"-70"	"enterLONG" ""
##	2008-12-01 00:00:00	"Open"	"ocolong"	"-70"	"exitLONG" ""
##	2010-02-01 00:00:00	"Open"	"ocolong"	"-70"	"enterLONG" ""

```
## 2010-08-02 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2010-12-01 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2011-03-01 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2011-04-01 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2011-05-02 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2011-10-03 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2012-03-01 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2012-04-02 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2012-08-01 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2013-02-01 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2013-10-01 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2014-01-02 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2014-04-01 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2014-11-03 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2015-05-01 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2015-08-03 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2016-01-04 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2016-02-01 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2016-04-01 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2016-08-01 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2016-09-01 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2016-12-01 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2017-03-01 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2017-12-01 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2018-02-01 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
## 2018-05-01 00:00:00 "Open" "ocolong" "-70" "enterLONG" ""
## 2018-10-01 00:00:00 "Open" "ocolong" "-70" "exitLONG" ""
```

Analyze strategy

Step 1: Endequity and Return

Calculate endequity and return of the strategy

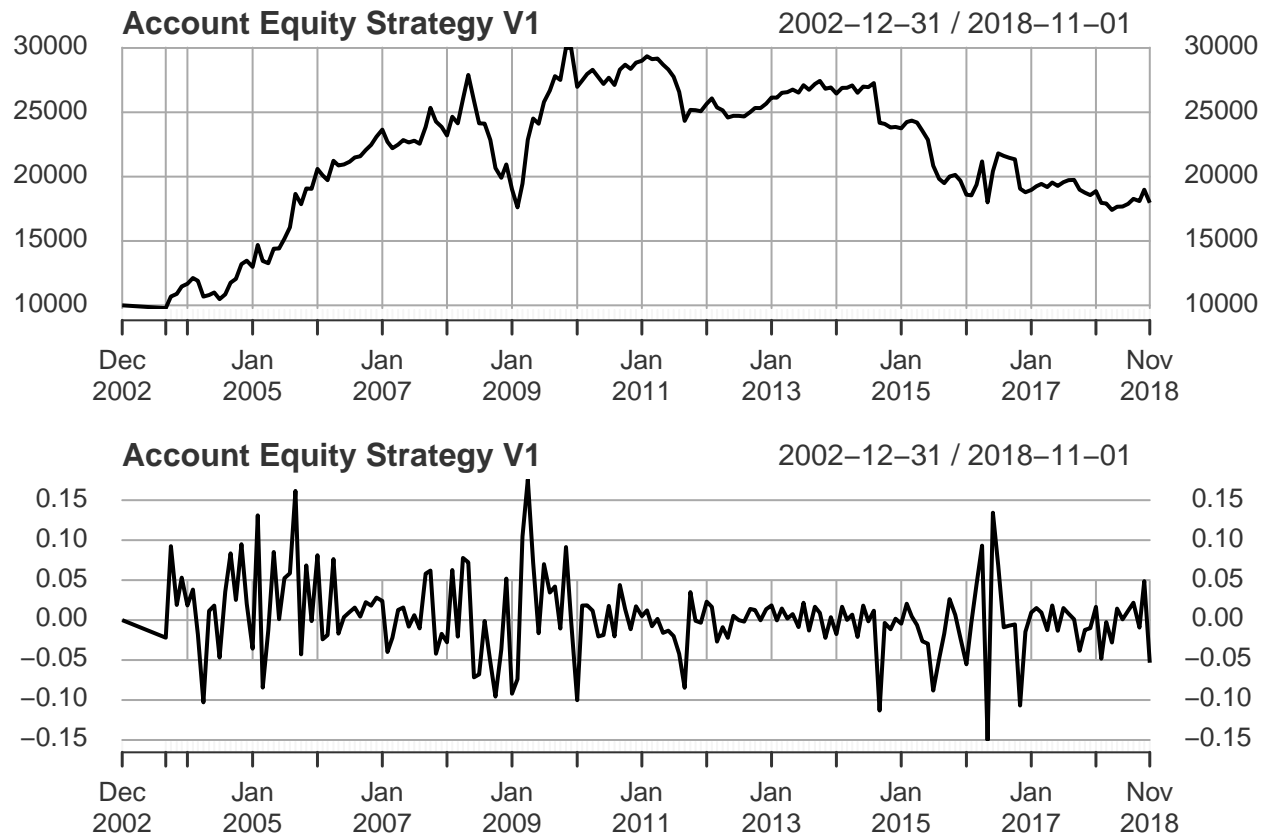
```
# end equity
acct <- getAccount(accountname)
equity.v1 <- acct$summary$End.Eq
colnames(equity.v1) <- c("Equity V1")

# returns of the strategy
rets.v1 <- monthlyReturn(equity.v1, type="arithmetic")
colnames(rets.v1) <- c("Returns V1")
```

Step 2: Plot Endequity and Return

The Endequity and the return of the strategy are plotted below. We can see that the strategy performs quiet well until the financial crisis 2008. After the crisis the performance is positive as well. However, from 2010 until today the strategy does not generate returns.

```
par(mfrow=c(2,1))
plot(equity.v1, main="Account Equity Strategy V1")
plot(rets.v1, main="Account Equity Strategy V1")
```

```
par(mfrow=c(1,1))
```

Step 3: Visualize the data original data

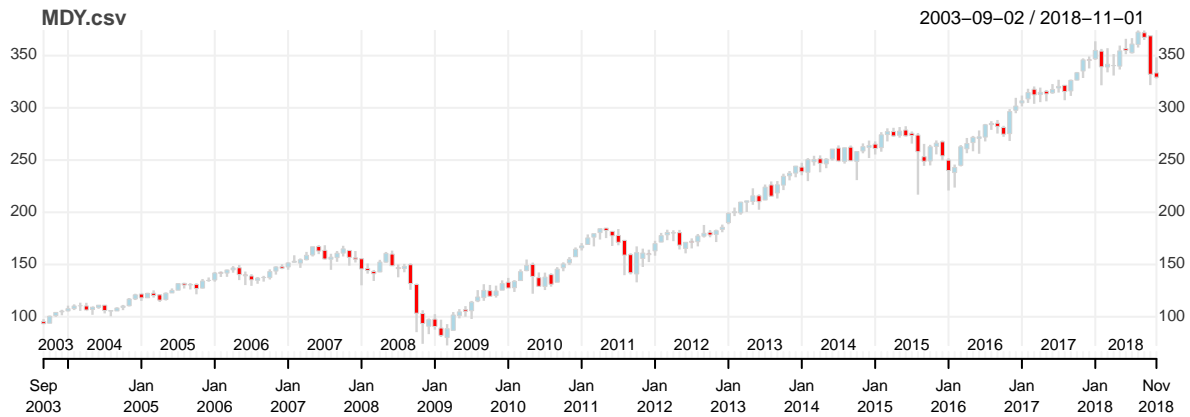
Below are visualizations of the original instruments which are traded. This plots can be used to get a first impression of the long term trend of the instrument and to see which general behaviour of the price movement can be observed.

```
par(mfrow=c(2,1))
for (symbol in symbollist){
  chart <- get(symbol)
  if (nrow(getTxns(portfolioname,Symbol=symbol)) > 1)
  {
    print(chart_Series(x=chart,name=symbol,
                      theme=myTheme,
                      subset=daterange))
  }
}
```





```
par(mfrow=c(1,1))
```

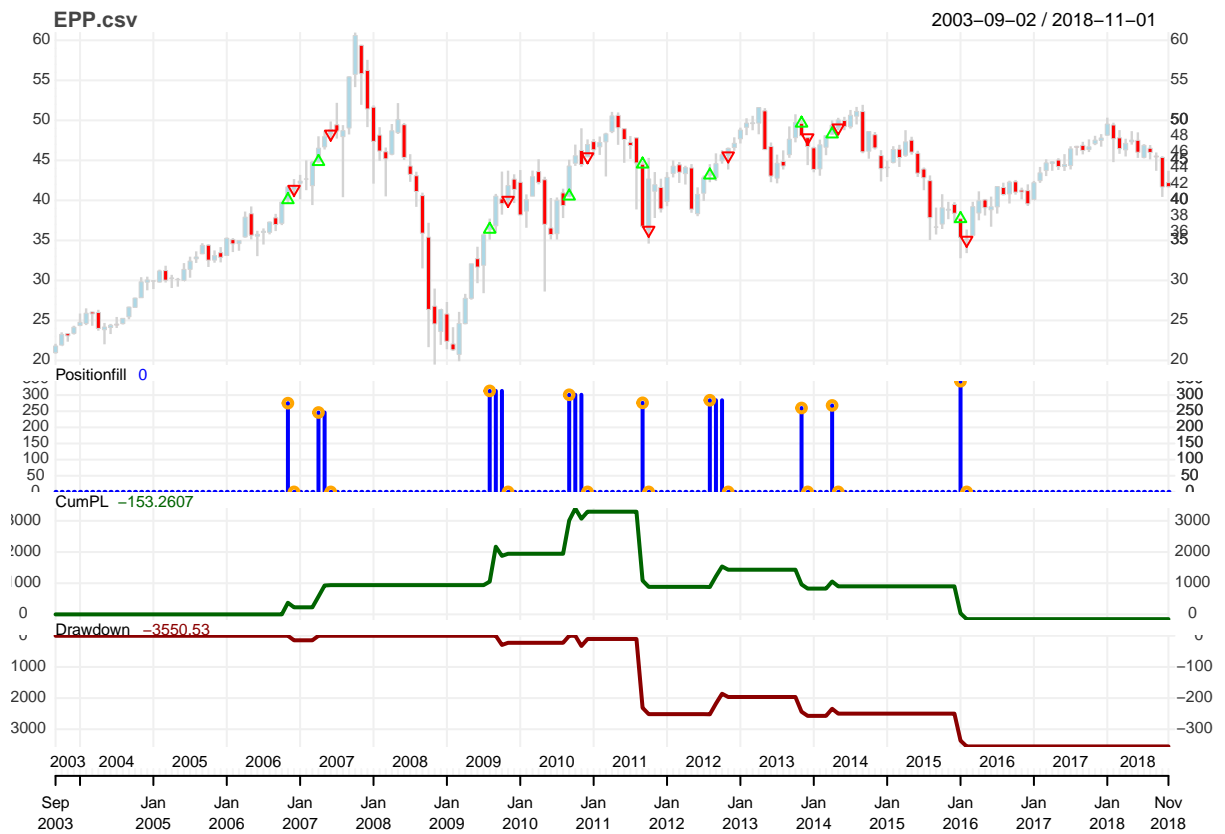


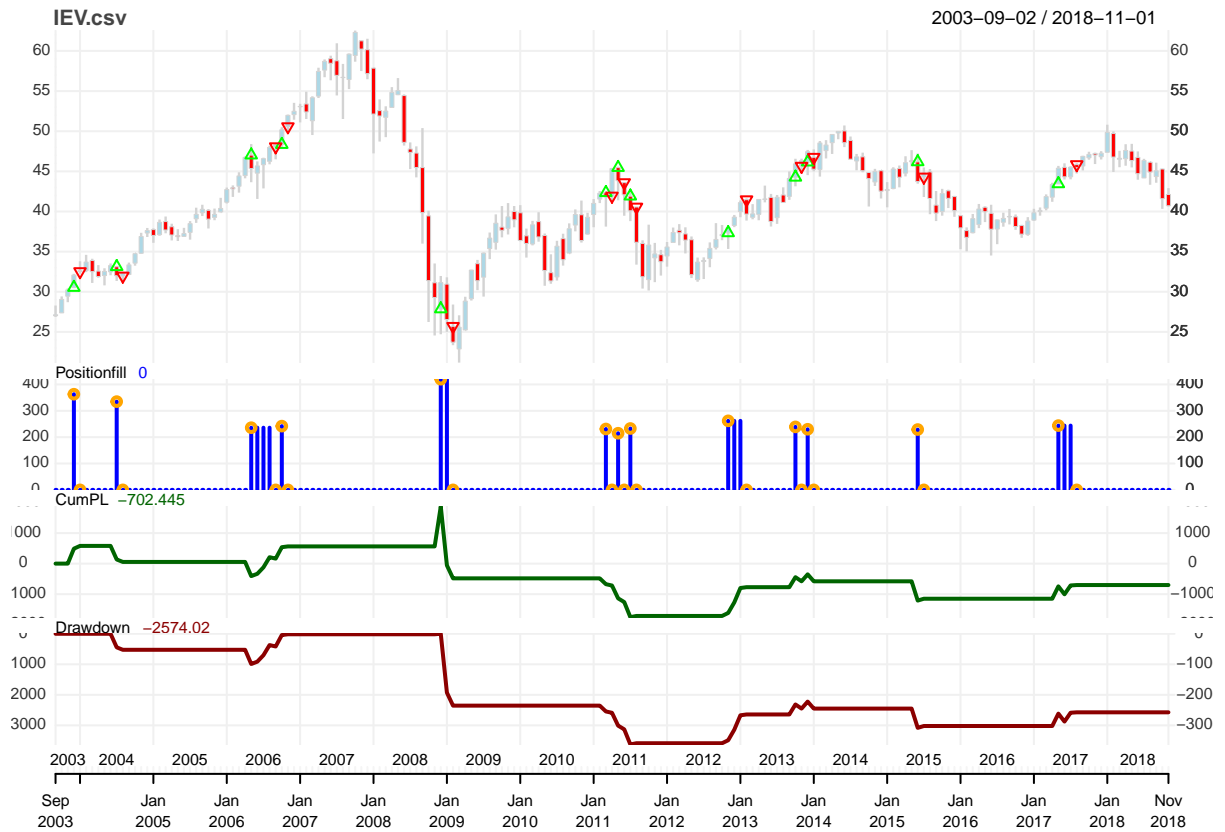
Step 4: Visualize all transactions

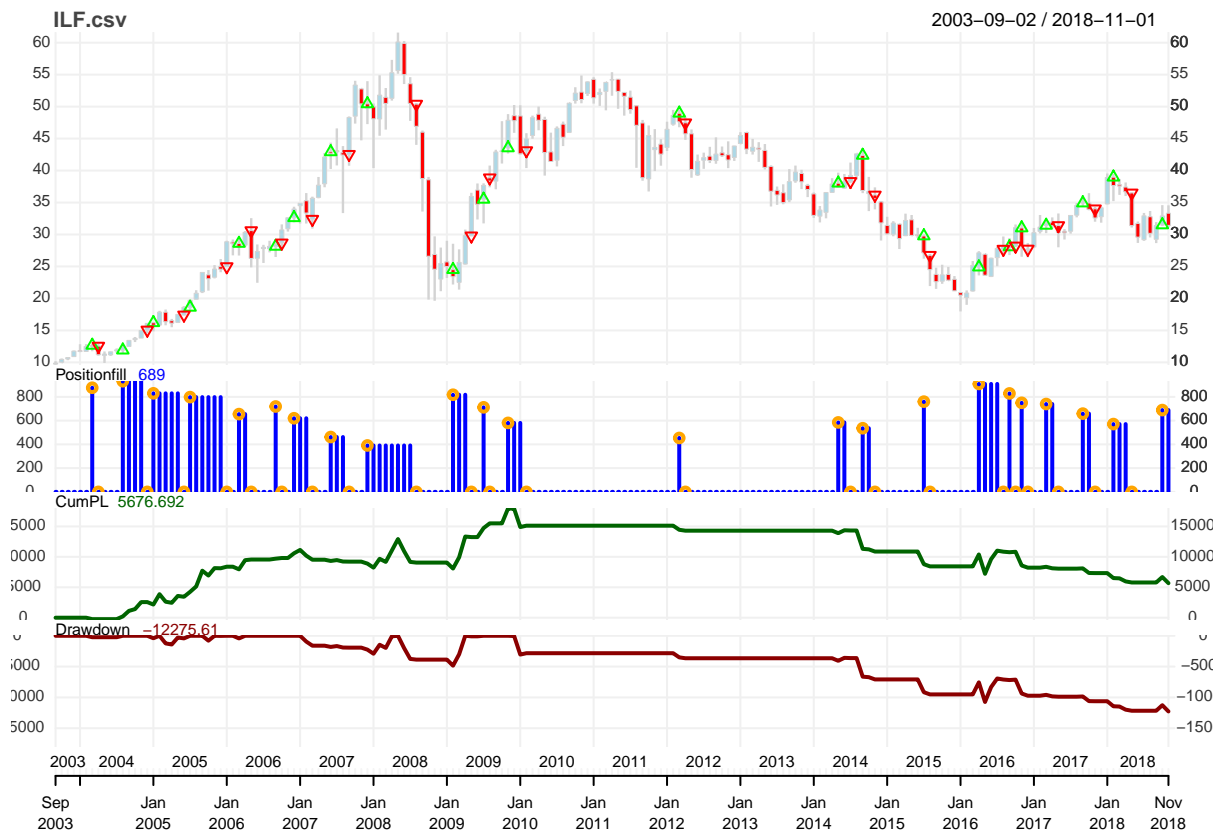
The following graphs show the combined view of the performance of the GRS trading system. It visualizes the trades (buy-transactions are visualized in green and sell-transactions are visualized in red). Moreover, the size of the blue squares indicates the size of the position (height) and the holding duration of the position (width). The green line shows the cumulative net profit curve, while the red line indicates the drawdown on each day compared to the last reached high.

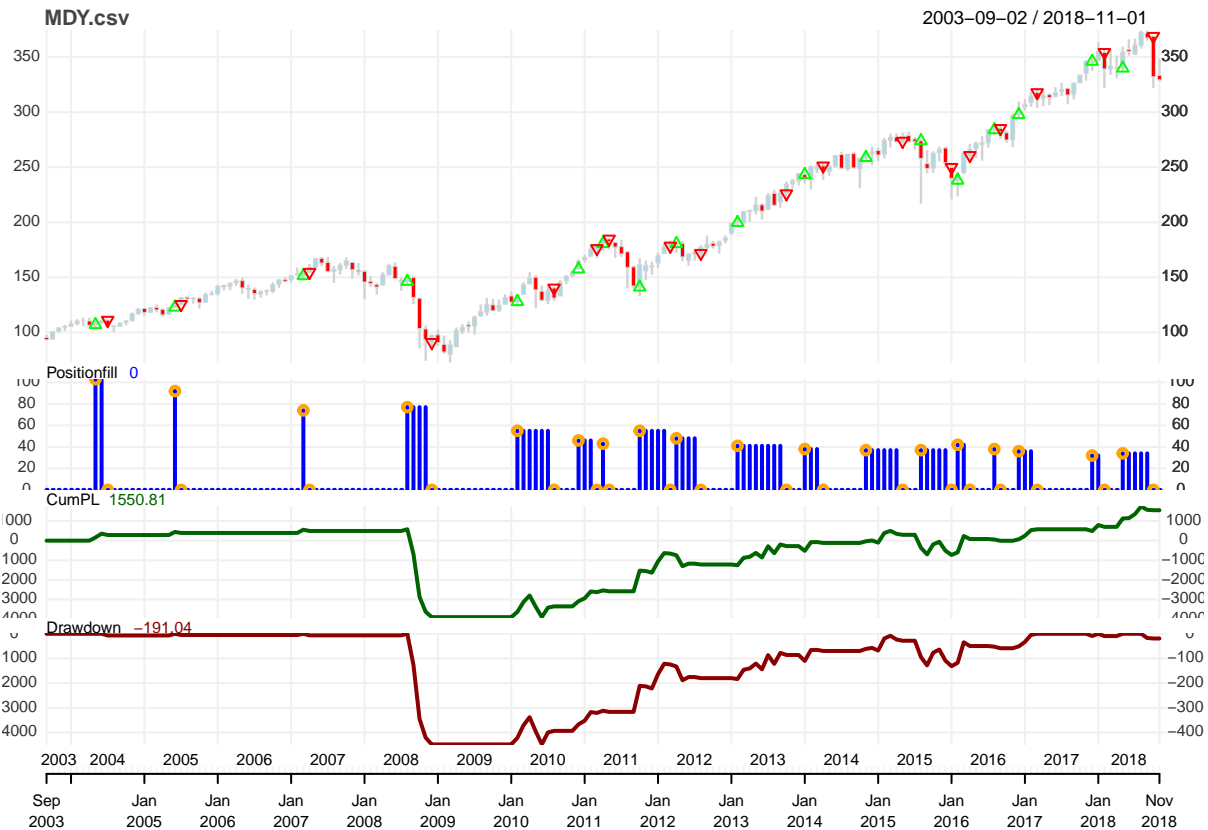
```
par(mfrow=c(1,1))
# plot graph with indicators for transactions
for (symbol in symbollist)
{
  if (nrow(getTxns(portfolioname,Symbol=symbol)) > 1)
  {
    symboldata <- get(symbol)
    chart.Posn(portfolioname,Symbol=symbol,type='candlesticks', theme=myTheme,subset=daterange)
  }
}
```











Step 5: Performance statistics

To analyze the results of our strategy appropriately we need to create some statistical measures which indicates how well the strategy performs. The following table summarizes some important trading statistics for all instruments. This statistic is just an excerpt of the available statistics which can be calculated and displayed. It is printed here as it can be assumed that they are interesting for investors to rate the trading strategy.

```
library(PerformanceAnalytics)
# Get returns for all instruments in the portfolio
rets.portfolio.v1 <- PortfReturns(Account=accountname)
rownames(rets.portfolio.v1) <- NULL

rets.v1.comb <- cbind(rets.v1, rets.portfolio.v1)

tstats <- tradeStats(Portfolio=portfolioname, Symbols=symbollist)

for (i in 1:nrow(tstats)) {
  trades.tab <- cbind(
    c("Trades", "Win Percent", "Loss Percent", "W/L Ratio"),
    c(tstats[i, "Num.Trades"],
      round(tstats[i, "Percent.Positive"], 2),
      round(tstats[i, "Percent.Negative"], 2),
      round((tstats[i, "Percent.Positive"] / tstats[i, "Percent.Negative"]), 2)))
  trades1 <- trades.tab
  rownames(trades1) <- c("Trades",
```

```

        "Win Percent",
        "Loss Percent",
        "W/L Ratio")

trades1 <- trades1[,2]
print(row.names(tstats[i,]))
print(trades1)
writeLines("")
}

## [1] "EEM.csv"
##      Trades  Win Percent Loss Percent   W/L Ratio
##      "15"    "53.33"    "46.67"    "1.14"
##
## [1] "EPP.csv"
##      Trades  Win Percent Loss Percent   W/L Ratio
##      "9"     "66.67"    "33.33"    "2"
##
## [1] "IEV.csv"
##      Trades  Win Percent Loss Percent   W/L Ratio
##      "13"    "53.85"    "46.15"    "1.17"
##
## [1] "ILF.csv"
##      Trades  Win Percent Loss Percent   W/L Ratio
##      "22"    "45.45"    "54.55"    "0.83"
##
## [1] "MDY.csv"
##      Trades  Win Percent Loss Percent   W/L Ratio
##      "18"    "83.33"    "16.67"    "5"

tab.perf <- table.Arbitrary(rets.v1.comb, metrics=c("Return.cumulative",
        "Return.annualized",
        "SharpeRatio.annualized",
        "CalmarRatio"),
        metricsNames=c("Cumulative Return",
        "Annualized Return",
        "Annualized Sharp Ratio",
        "Calmar Ratio"))
tab.risk <- table.Arbitrary(rets.v1.comb, metrics=c("StdDev.annualized",
        "maxDrawdown",
        "VaR",
        "ES"),
        metricsNames=c("Annualized StdDev",
        "Max Drawdown",
        "Value-at-Risk",
        "Conditional VaR"),
        method=c("historical"))
colnames(tab.perf) <- substr(colnames(tab.perf),1,3)
colnames(tab.risk) <- substr(colnames(tab.risk),1,3)

writeLines("Performance Metrics")

## Performance Metrics

```

```
tab.perf
```

```
##                               Ret      EEM      EPP      IEV
## Cumulative Return      0.79688763 0.088293969 -0.064194858 -0.107968540
## Annualized Return      0.03896087 0.005563702 -0.004341238 -0.007464062
## Annualized Sharp Ratio 0.23919069 0.055147473 -0.054962595 -0.100559974
## Calmar Ratio           0.09272764 0.020011491 -0.013395831 -0.023243454
##                               ILF      MDY
## Cumulative Return      -0.043596425 0.08111090
## Annualized Return      -0.002918703 0.00512714
## Annualized Sharp Ratio -0.010403177 0.05239138
## Calmar Ratio           -0.003717909 0.01317575
```

```
writeLines("Risk Metrics")
```

```
## Risk Metrics
```

```
tab.risk
```

```
##                               Ret      EEM      EPP      IEV
## Annualized StdDev  0.16288622 0.10088771 0.07898532 0.07422498
## Max Drawdown       0.42016457 0.27802538 0.32407384 0.32112533
## Value-at-Risk      -0.08283979 -0.02130370 -0.01295680 -0.01846805
## Conditional VaR     -0.10182500 -0.06312026 -0.04993423 -0.05490760
##                               ILF      MDY
## Annualized StdDev  0.2805588 0.09786228
## Max Drawdown       0.7850389 0.38913460
## Value-at-Risk      -0.0974546 -0.03307680
## Conditional VaR     -0.2051515 -0.07664500
```

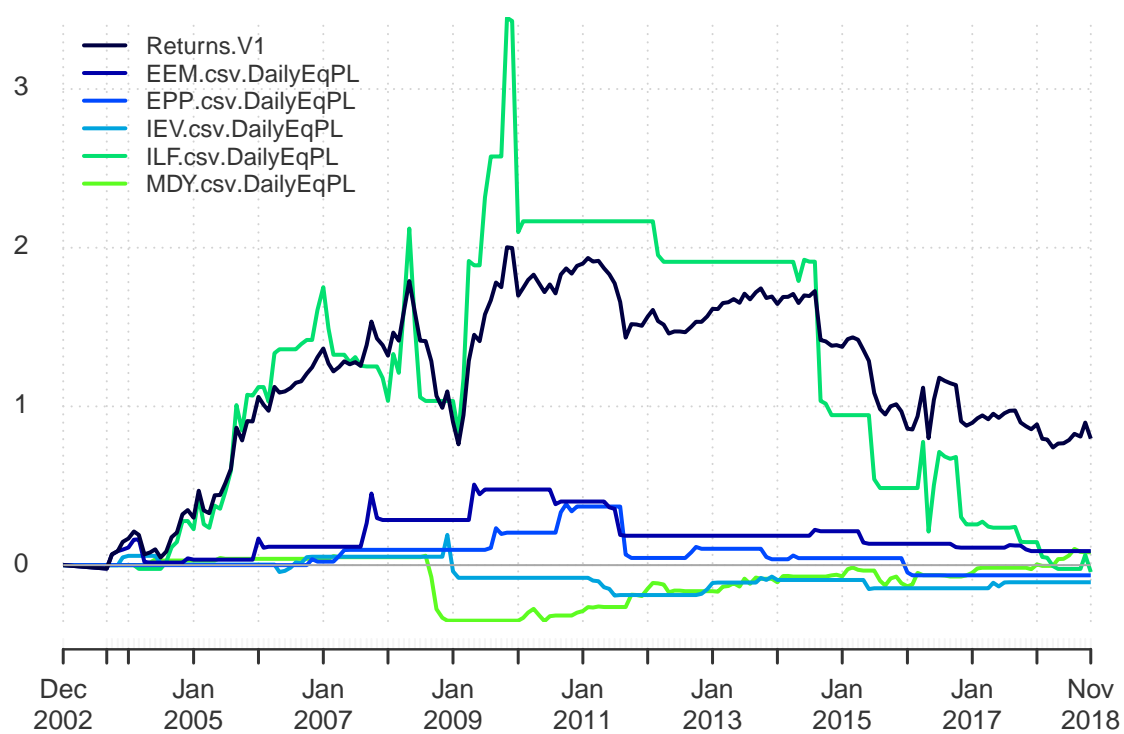
Step 6: Visualize portfolio statistics

The following plot shows the cumulative returns of the instruments which are used in the strategy in order to see whether the returns of the instruments differentiate significantly from each other.

```
par(mfrow=c(1,1))
chart.CumReturns(rets.v1.comb,colorset=rich10equal,legend.loc="topleft",main="Cumulative Returns")
```

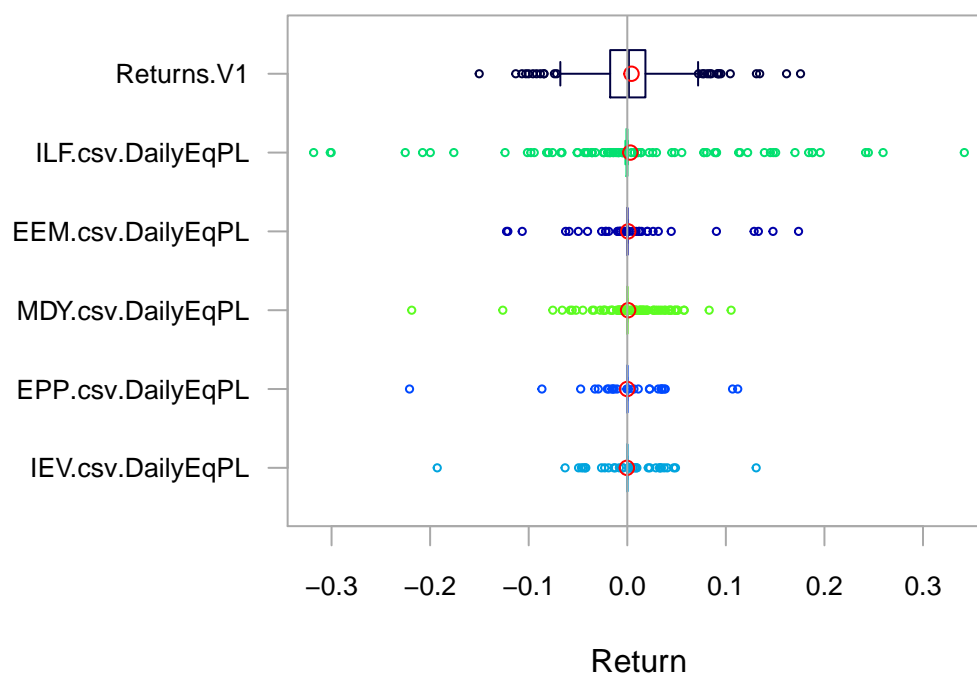
Cumulative Returns

2002-12-31 / 2018-11-01



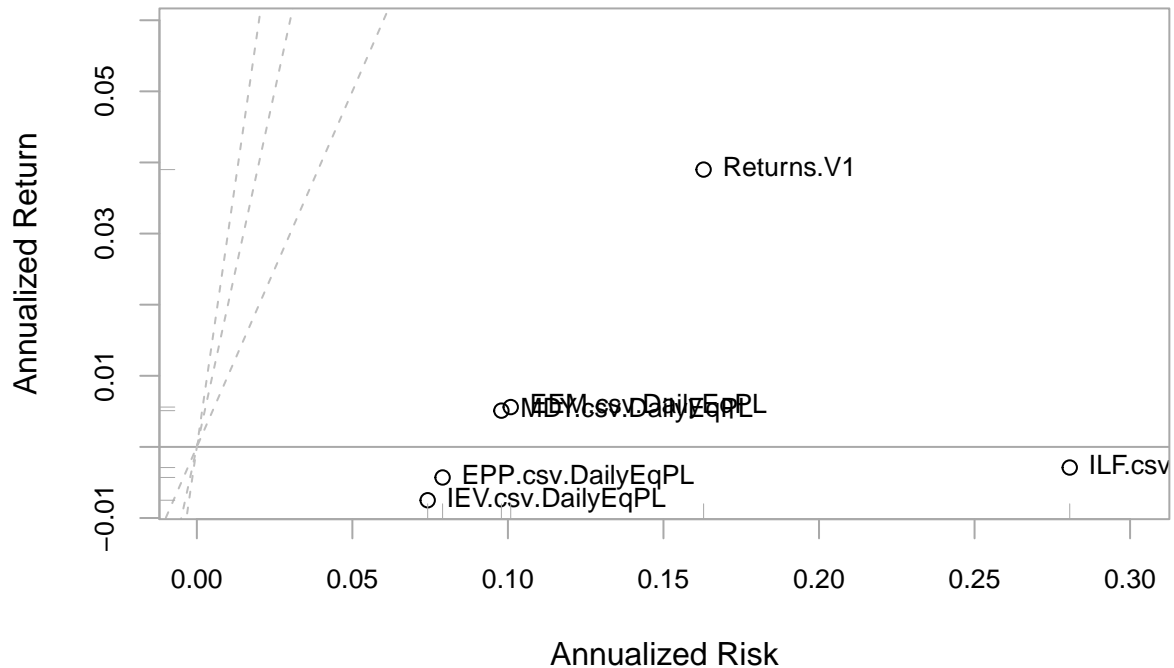
```
chart.Boxplot(rets.v1.comb,colorset=rich10equal)
```

Return Distribution Comparison



```
chart.RiskReturnScatter(rets.v1.comb)
```

Annualized Return and Risk

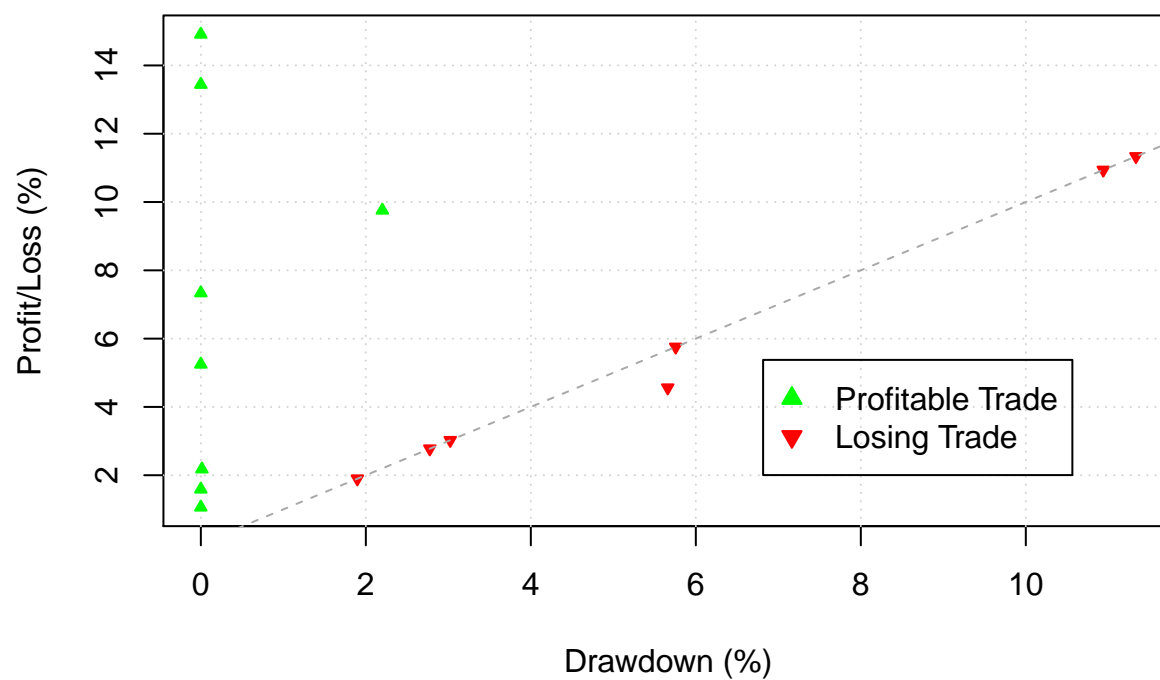


Step 7: Could we have performed better or even worse? MAE / MFE

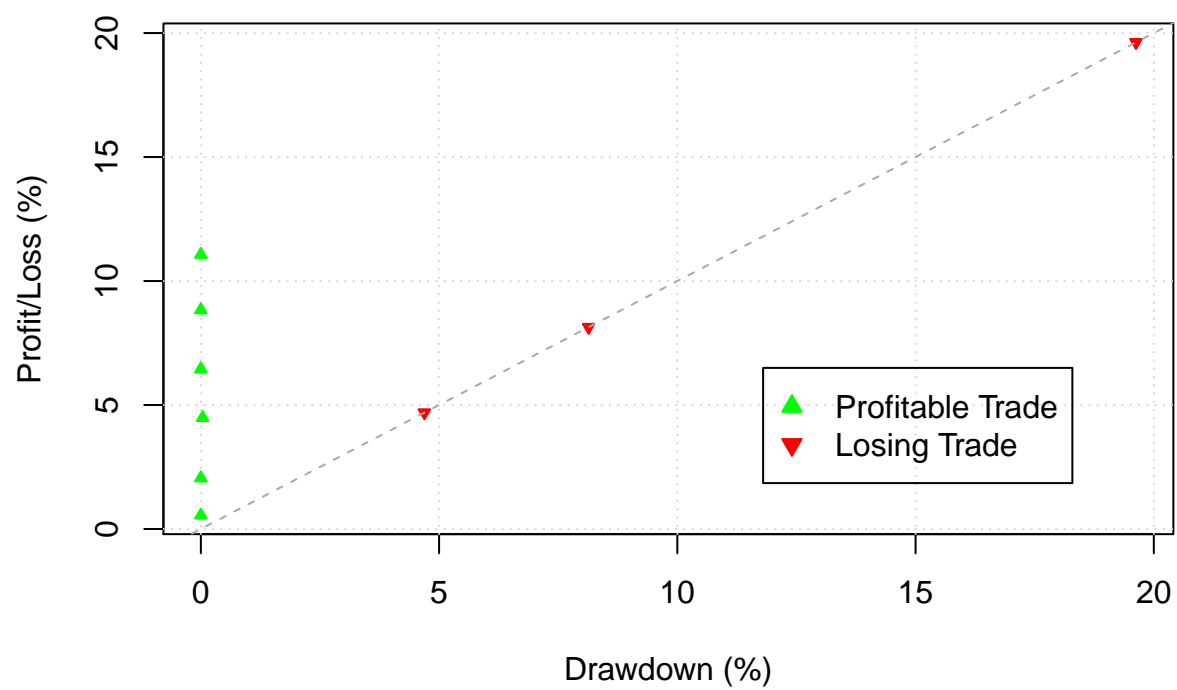
The efficiency of a strategy is also dependent on the question whether we could have performed better or worse. This can be shown with the maximum adverse excursion and the maximum favorable excursion. The Maximum Adverse excursion shows how much we could have lost at most during the trade, while the maximum favorable excursion shows how much we could have earned more than we did. In the MFE case we want all trades to be as close to the diagonal line as possible. Every movement to the bottom right sight can be interpreted as a not realized opportunity to earn higher returns.

```
for (instrument in symbolist){
  chart.ME(portfolioname, instrument, type="MAE", scale="percent")
}
```

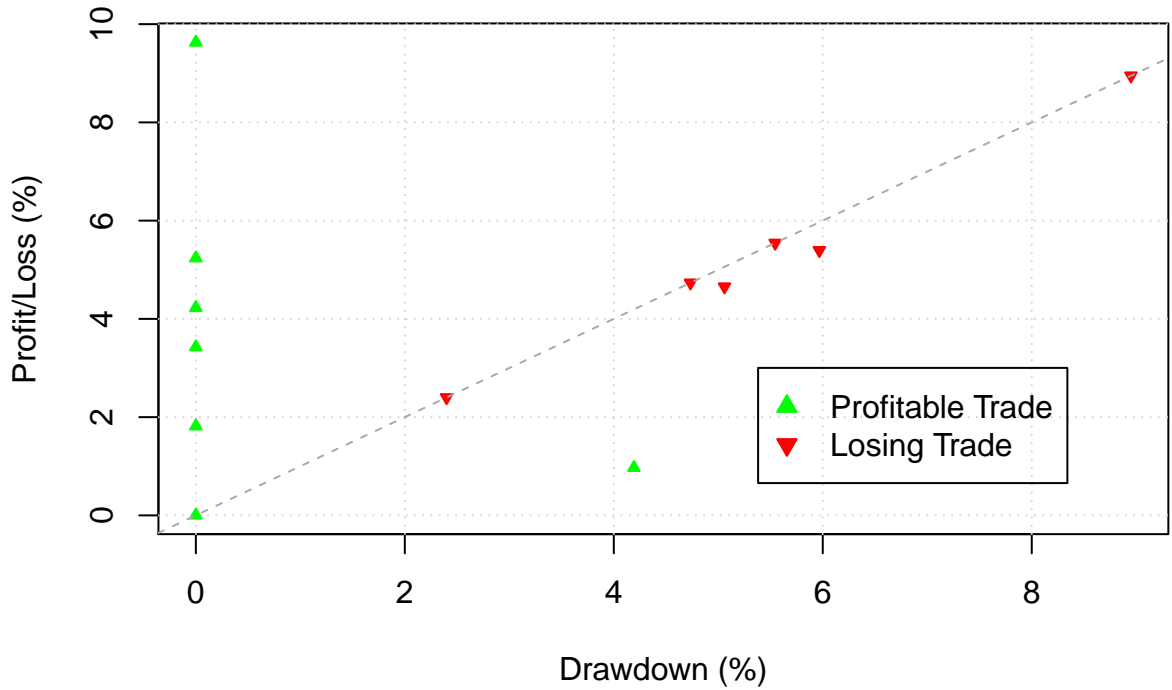
EEM.csv Maximum Adverse Excursion (MAE)



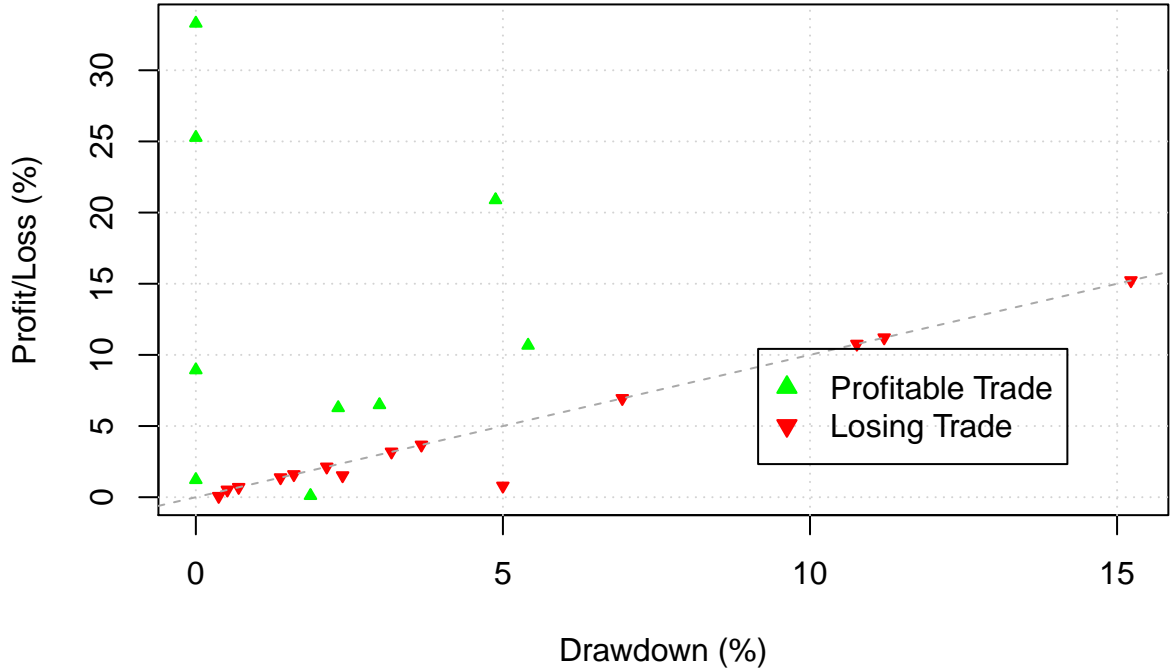
EPP.csv Maximum Adverse Excursion (MAE)



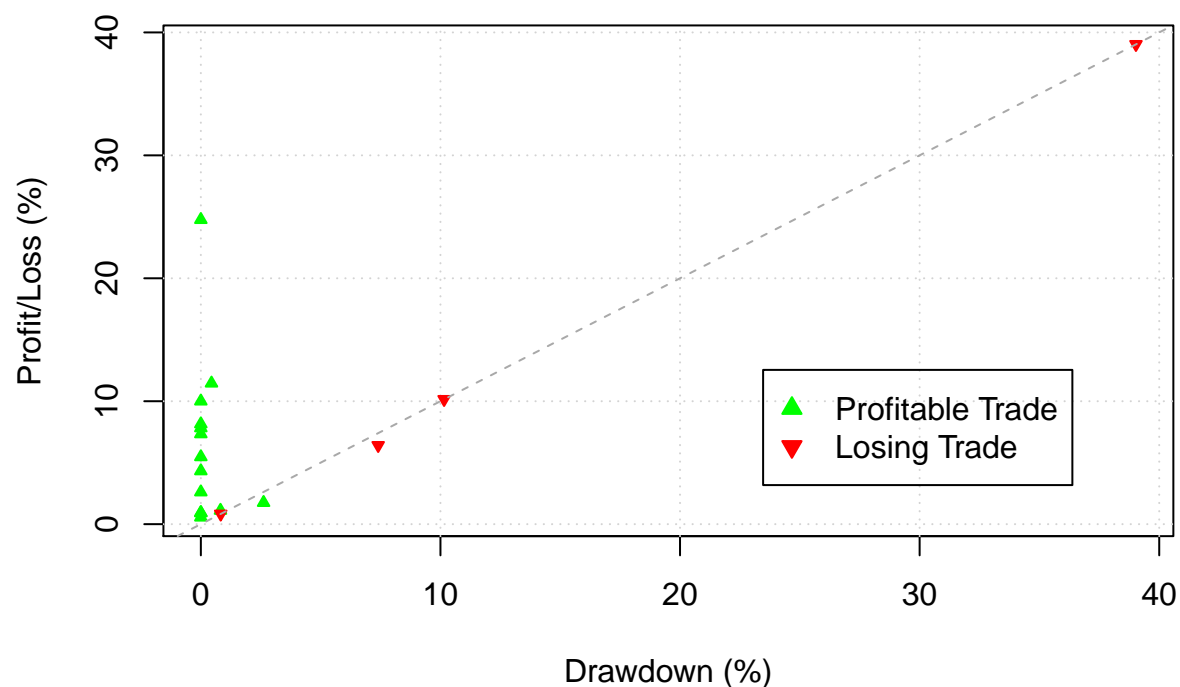
IEV.csv Maximum Adverse Excursion (MAE)



ILF.csv Maximum Adverse Excursion (MAE)



MDY.csv Maximum Adverse Excursion (MAE)



Step 8: Strategy Risk

```
logR <- monthlyReturn(equity.v1,type="arithmetic")
index(logR) <- as.yearmon(index(logR))

library(quantmod)

dir <- "~/Desktop/R/GRS/Risk/"
ff <- read.csv(paste0(dir,"FFFactors.csv"))
ffxts <- as.xts(ff[,-1],order.by=as.Date(ff$dates))
index(ffxts) <- as.yearmon(index(ffxts))

funddata <- merge(logR,ffxts)
funddata <- funddata[complete.cases(funddata),]
# Calculate Excess Returns above the risk free rate
funddata$excessrtn <- funddata$monthly.returns*100 - funddata$rf
```

Perform a CAPM Regression

```
capm <- lm(excessrtn ~ mkt.rf, data=funddata)
summary(capm)

##
## Call:
## lm(formula = excessrtn ~ mkt.rf, data = funddata)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.0897  -1.7315  -0.3492   1.3355  15.6863
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.15170     0.29527  -0.514   0.608
## mkt.rf       0.68018     0.07366   9.234 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.888 on 177 degrees of freedom
## Multiple R-squared:  0.3251, Adjusted R-squared:  0.3213
## F-statistic: 85.27 on 1 and 177 DF,  p-value: < 2.2e-16
```

Perform a Fama-French 3 factor Regression

```
# Run Fama-French 3 factor Regression
ff3regression <- lm(excessrtn ~ mkt.rf + smb + hml, data=funddata)
summary(ff3regression)
```

```
##
## Call:
## lm(formula = excessrtn ~ mkt.rf + smb + hml, data = funddata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.8237  -1.6443  -0.3099   1.2969  15.6739
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.14468     0.29606  -0.489   0.626
## mkt.rf       0.64425     0.08086   7.967 1.99e-13 ***
## smb         0.09618     0.13644   0.705   0.482
## hml         0.10223     0.12221   0.837   0.404
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.896 on 175 degrees of freedom
## Multiple R-squared:  0.3299, Adjusted R-squared:  0.3184
## F-statistic: 28.72 on 3 and 175 DF,  p-value: 3.763e-15
```

Part C: Compare with Buy and Hold Strategy

After having analyzed the strategy extensively, a comparison to another strategy should be made to evaluate the performance based on a benchmark which could have been an alternative investment. In this case a simple buy and hold strategy is used. At the first date of the trading period we place a buy order and sell our position at the last day of the selected period. In order to do this, we create a new Portfolio and a new Account. As ETFs are used which cover the entire world market, the appropriate Buy and Hold comparison would be an index which covers also the entire world market. Therefore, the MSCI World Index is used as a benchmark.

Step 1: Generate the buy-and-hold portfolio, account, and transactions

```
BuyHoldDirectory = "~/Desktop/R/GRS/BH/"
BuyHoldInstrument = "MXWO.csv"

# Any objects, in case there was a buyhold
# portfolio initialized before are removed
suppressWarnings(try(rm(list=c("account.buyhold",
                              "portfolio.buyhold"),
                        pos=.blotter)))

# The Buy and hold instrument is loaded
LoadCourseFile(BuyHoldDirectory,BuyHoldInstrument,debugme=TRUE,dates=daterange)

symname <- BuyHoldInstrument
# The Buy and hold instrument is initialized
stock(BuyHoldInstrument,currency="USD")
# convert data to monthly
x<-get(BuyHoldInstrument)
x=to.monthly(x,indexAt="startof",drop.time=FALSE, OHLC=TRUE)
colnames(x)<-gsub("x.", "", colnames(x))

assign(BuyHoldInstrument,x)

BuyHoldSymbol<-get(BuyHoldInstrument)

# The portfolio and account "buyhold" is initialized
initPortf("buyhold",
          BuyHoldInstrument,
          initDate=initdate,
          currency="USD")
initAcct("buyhold",
         portfolios="buyhold",
         initDate=initdate,
         initEq=startingcapital,
         currency="USD")

# The first date of the defined daterange is selected
currentdate <- first(time(BuyHoldSymbol))

# The close price at this date is selected
closeprice <- as.numeric(Cl(BuyHoldSymbol[currentdate,]))

# Calculate the unitsize we can buy with our startingcapital
unitsize <- as.numeric(trunc(startingcapital/closeprice))

# Place the transaction for the instrument at the first date
addTxn("buyhold",
       Symbol=BuyHoldInstrument,
       TxnDate=currentdate,
       TxnPrice=closeprice,
       TxnQty=unitsize,
       TxnFees=txncost)
```

```

# Select the last date of the daterange period
lastdate <-last(time(BuyHoldSymbol))

# Select the price at the last date
lastprice <- as.numeric(Cl(BuyHoldSymbol[lastdate,]))

# Sell the position at the last date of the daterange
addTxn("buyhold",
      Symbol=BuyHoldInstrument,
      TxnDate=lastdate,
      TxnPrice=lastprice,
      TxnQty=-unitsize,
      TxnFees=txncost)

# update portfolio and account
updatePortf(Portfolio="buyhold")
updateAcct(name="buyhold")
updateEndEq(Account="buyhold")

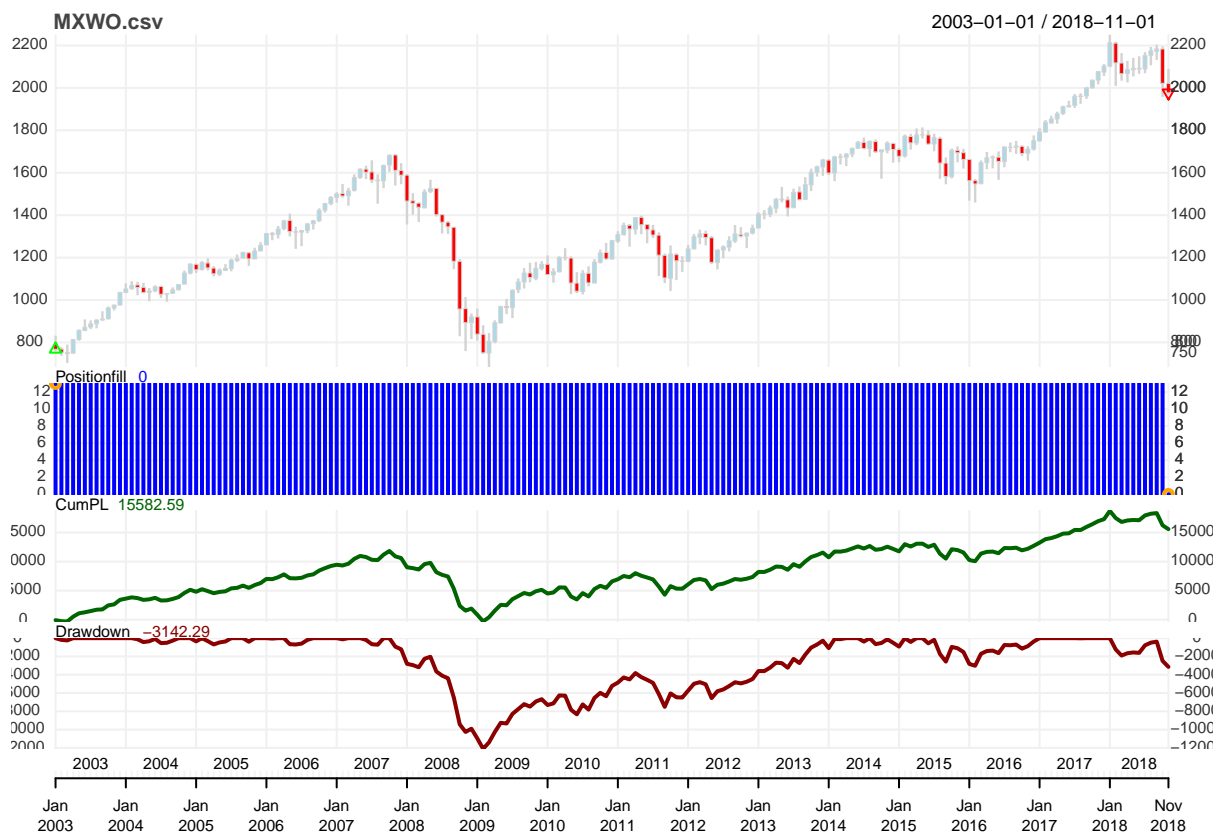
```

Step 2: Visualize the Buy and Hold strategy

```

chart.Posn("buyhold",Symbol=BuyHoldInstrument, type='candlesticks', theme=myTheme, subset=daterange)

```



Step 3: Compare Buy and Hold with GRS

Generate statistics for the buy-and-hold portfolio. Generate returns and metrics of the buy-and-hold portfolio/account and compare statistics for the quantitative/technical portfolio to the buy-and-hold portfolio.

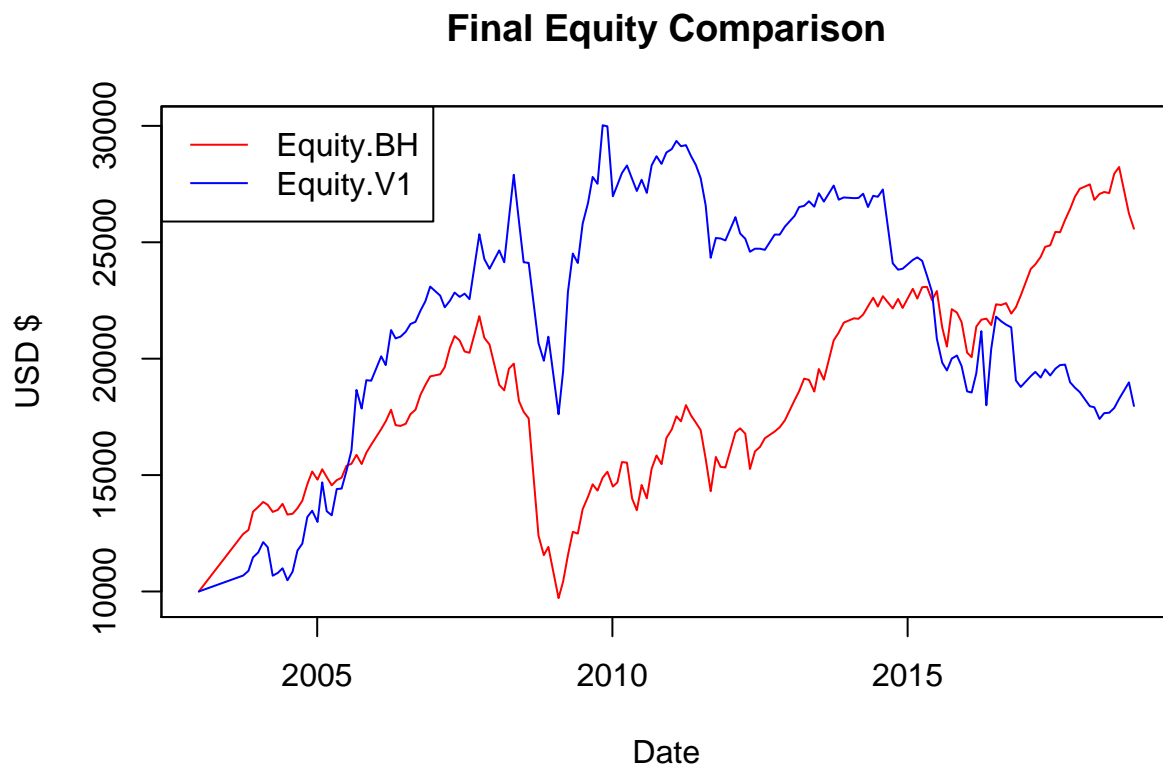
Generate equity and return

```
equity.bh <- getAccount("buyhold")$summary$End.Eq
colnames(equity.bh) <- c("Equity BH")
equity.bh.v1 <- cbind(equity.bh, equity.v1)
equity.bh.v1 <- na.omit(equity.bh.v1)

rets.bh <- monthlyReturn(equity.bh, type="arithmetic")
colnames(rets.bh) <- c("Returns BH")
rets.bh.v1 <- cbind(rets.bh, rets.v1)
rets.bh.v1 <- na.omit(rets.bh.v1)
```

Visually compare Equity curves

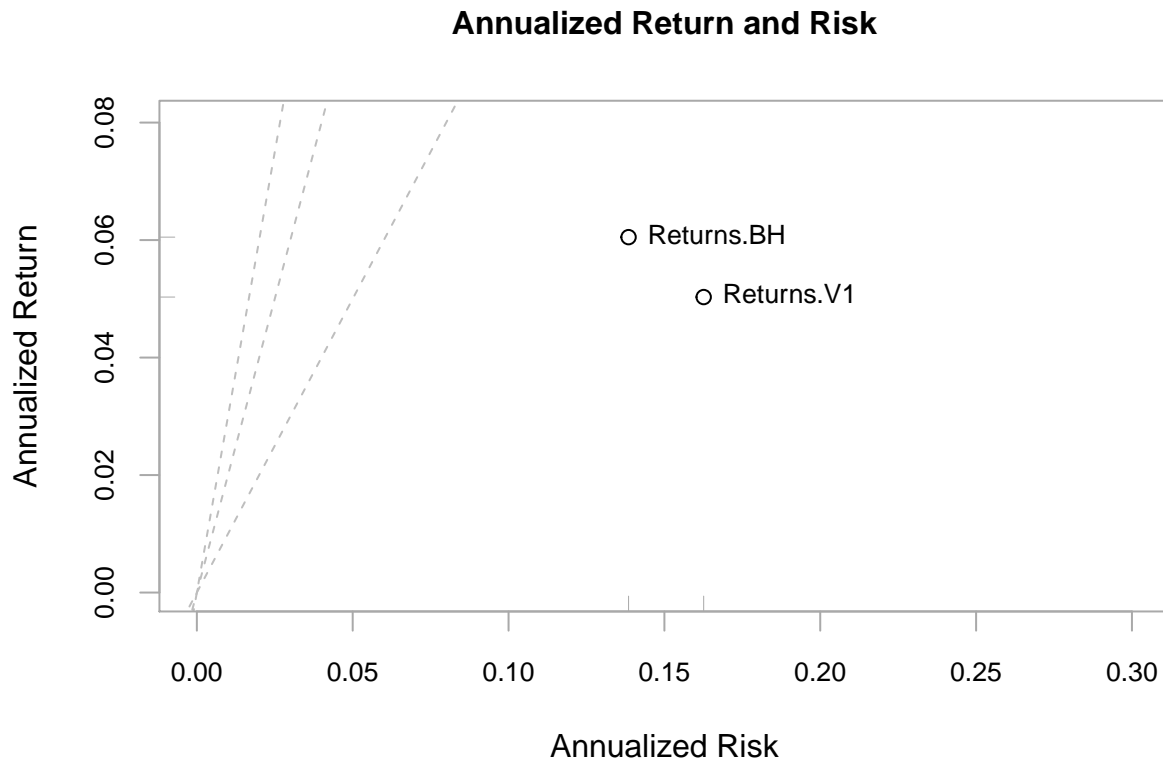
```
plot.zoo(equity.bh.v1, plot.type="single", col=c('red', 'blue'), main="Final Equity Comparison", xlab="Date",
legend(x="topleft", legend=colnames(equity.bh.v1), col=c("red", "blue"), lty=1) # add a legend to the chart
```



Visualization of risk and return

We can see that the Buy and Hold strategy generates higher returns while having a smaller annualized risk. The sharpe ratio is higher and therefore the buy and hold strategy is better, compared to the GRS strategy.

```
chart.RiskReturnScatter(rets.bh.v1, xlim = c(0,0.3))
```



```
table.AnnualizedReturns(rets.bh.v1)
```

##	Returns.BH	Returns.V1
## Annualized Return	0.0605	0.0503
## Annualized Std Dev	0.1385	0.1626
## Annualized Sharpe (Rf=0%)	0.4371	0.3094

```
table.Stats(rets.bh.v1)
```

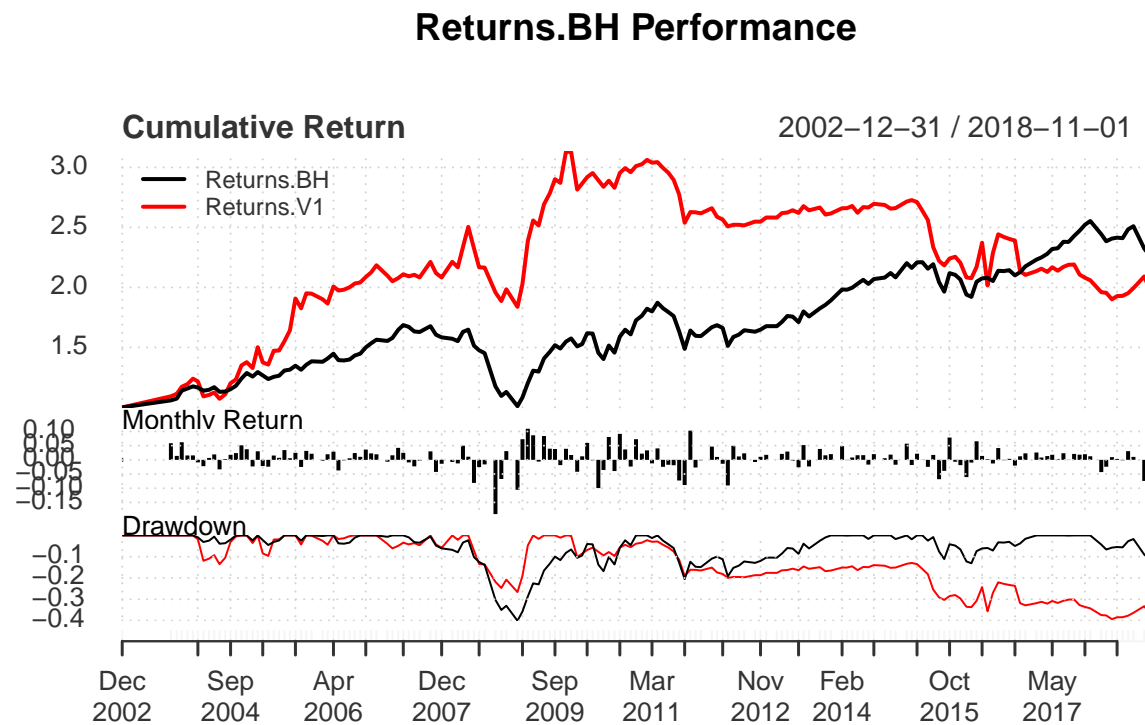
##	Returns.BH	Returns.V1
## Observations	167.0000	167.0000
## NAs	0.0000	0.0000
## Minimum	-0.1910	-0.1502
## Quartile 1	-0.0164	-0.0170
## Median	0.0104	0.0015
## Arithmetic Mean	0.0057	0.0052
## Geometric Mean	0.0049	0.0041
## Quartile 3	0.0243	0.0184
## Maximum	0.1095	0.1758
## SE Mean	0.0031	0.0036
## LCL Mean (0.95)	-0.0004	-0.0020
## UCL Mean (0.95)	0.0118	0.0123
## Variance	0.0016	0.0022
## Stdev	0.0400	0.0469
## Skewness	-0.8399	0.3969


```
## Kurtosis          3.4691      2.1143
```

Visualization of cumulative Returns

The following plot visualizes the cumulative returns of both strategies. Also here we can see that the GRS outperformed the buy and hold strategy before ca. 2016. However, the returns could not be sustained, thus, the GRS performed over the long run not as good as a simple buy and hold strategy.

```
charts.PerformanceSummary(rets.bh.v1, geometric=T, wealth.index=TRUE)
```



Strategy Improvement

As we saw in the analysis above, the GRS strategy in its plain form cannot outperform a buy and hold strategy of the MSCI World index. A first approach to improve the GRS is only to perform Long trades, when the ROC (the momentum) was positive. For this, we can use the already calculated data from `pos_rets`. This data indicates for every month whether the ROC was positive in the past period or not. The signals of the standard GRS strategy can be still used with a slight adjustment to the entryLong signal, which is only triggered when both, the rank meets the requirements and the returns were positive.

Step 1: Initialize account, strategy and portfolio

```
strategyname.alt <- paste(strategyname, ".", "v2", sep = "")
portfolioname.alt <- paste(portfolioname, ".", "v2", sep = "")
accountname.alt <- paste(accountname, ".", "v2", sep = "")
rm.strat(strategyname.alt)
initPortf(portfolioname.alt, symbollist, initDate=initdate, currency="USD")
```

```

initAcct(accountname.alt, portfolios=portfolioname.alt, initDate=initdate, initEq=startingcapital, currency=
initOrders(portfolio=portfolioname.alt, initDate=initdate)
strategy(strategyname.alt, store=TRUE)
strat <- getStrategy(strategyname.alt)

```

Step 2: New Strategy Rules

As described above, these are the new rules which includes that the signal longEntry is only set to TRUE, when the Rank requirement is fulfilled and the returns were positive.

```

# Indicator
add.indicator(strategyname.alt,
               name = "sigThreshold",
               arguments=list(column="Rank",
                             threshold=buypositions,
                             relationship="lte"),
               label = "bpos")

# Signal
add.signal(strategyname.alt,
            name="sigFormula",
            arguments=list(columns=c("Rank.bpos", "PosRets"),
                          formula="(Rank.bpos == 1) & (PosRets==1)",
                          label="longEntry")

add.signal(strategyname.alt,
            name="sigThreshold",
            arguments=list(column="Rank",
                          threshold=buypositions,
                          relationship="gt"),
            label="longExit")

# Rules
add.rule(strategyname.alt,
          name="ruleSignal",
          arguments=list(sigcol="longEntry",
                        signal=TRUE,
                        orderqty=100,
                        ordertype="market",
                        orderside="long",
                        replace=FALSE,
                        prefer="Open",
                        osFUN='osInvestAll',
                        orderset="ocolong",
                        TxnFees=txncost),
          type="enter",
          label="LE",
          path.dep=TRUE)

add.rule(strategyname.alt,
          name="ruleSignal",
          arguments=list(sigcol="longExit",
                        signal=TRUE,
                        orderqty="all",

```

```

ordertype="market",
orderside="long",
replace=FALSE,
prefer="Open",
orderset="ocolong",
TxnFees=txncost),

type="exit",
label = "LX",
path.dep=TRUE)

```

##Step 3: Apply the strategy

```

applyStrategy(strategy=strategyname.alt,portfolios=portfolioname.alt)

updatePortf(strategyname.alt)
updateAcct(strategyname.alt)
updateEndEq(strategyname.alt)

```

Analysis of improved strategy GRS.V2

Step 1: Calculate equity and return of the improved strategy

```

acct.alt <- getAccount(accountname.alt)

equity.v2 <- acct.alt$summary$End.Eq
colnames(equity.v2) <- c("Equity.V2")

rets.v2 <- monthlyReturn(equity.v2, type = "arithmetic")
colnames(rets.v2) <- c("Return V2")

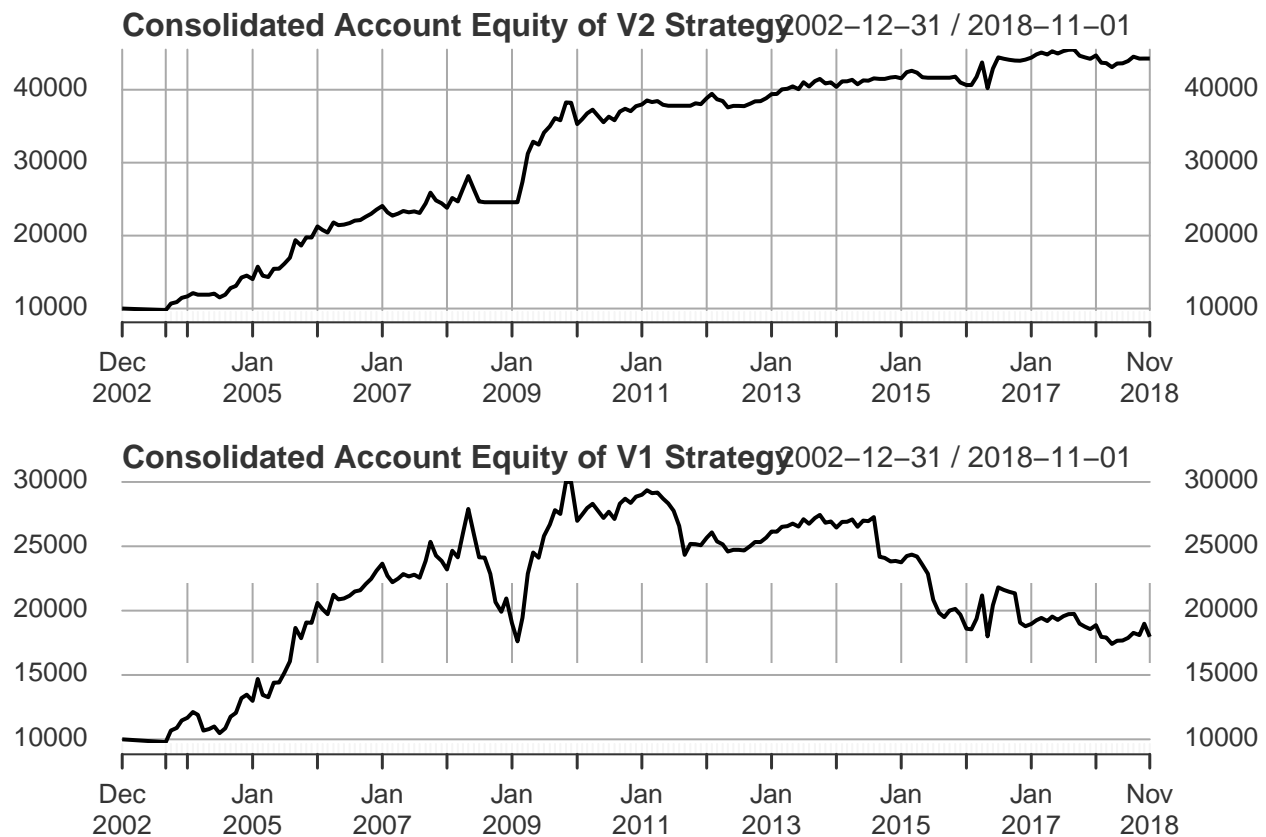
```

Step 2: Visualize Equity of GRS.V2

```

par(mfrow=c(2,1))
plot(equity.v2, main="Consolidated Account Equity of V2 Strategy")
plot(equity.v1, main="Consolidated Account Equity of V1 Strategy")

```



```
par(mfrow=c(1,1))
```

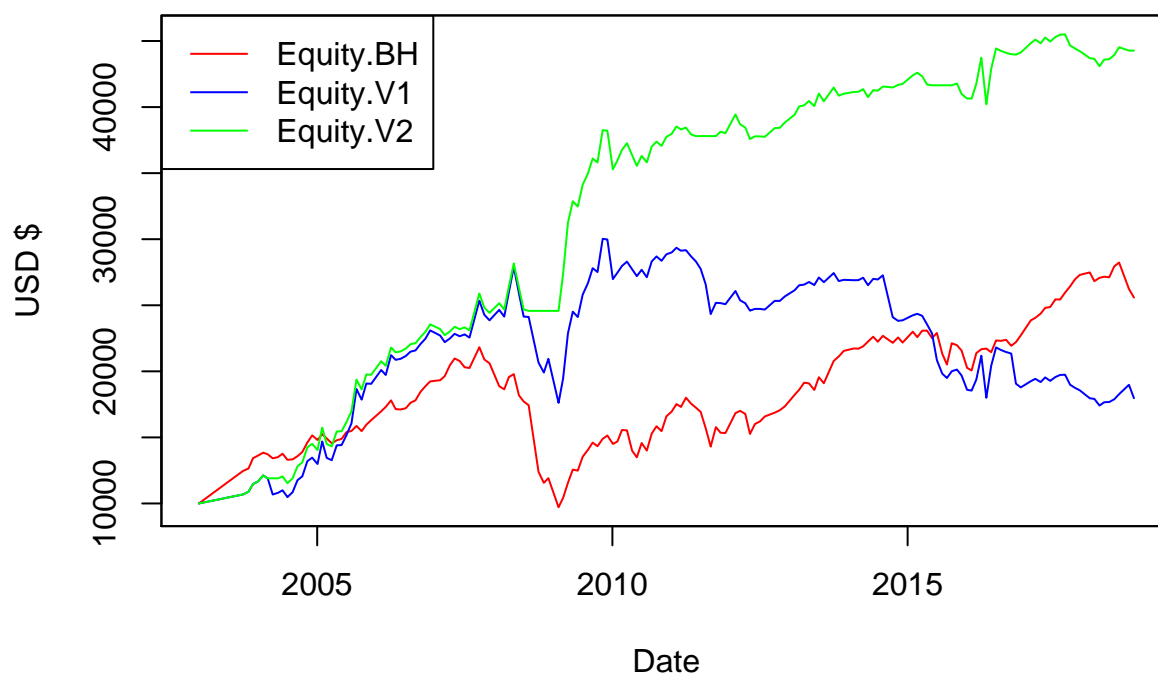
We can clearly see that the improved strategy generates more return. Phases where the original strategy generates negative returns, e.g., 2008 - 2009 are improved in the V2 strategy by simply not being invested in the market.

Step 3: Visualize Equity comparison

```
equity.bh.v1.v2 <- cbind(equity.bh, equity.v1, equity.v2)
equity.bh.v1.v2 <- na.omit(equity.bh.v1.v2)
```

```
plot.zoo(equity.bh.v1.v2, plot.type="single", col=c('red', 'blue', 'green'), main="Final Equity Comparison",
legend(x="topleft", legend=colnames(equity.bh.v1.v2), col=c("red", "blue", "green"), lty=1) # add a legend to
```

Final Equity Comparison

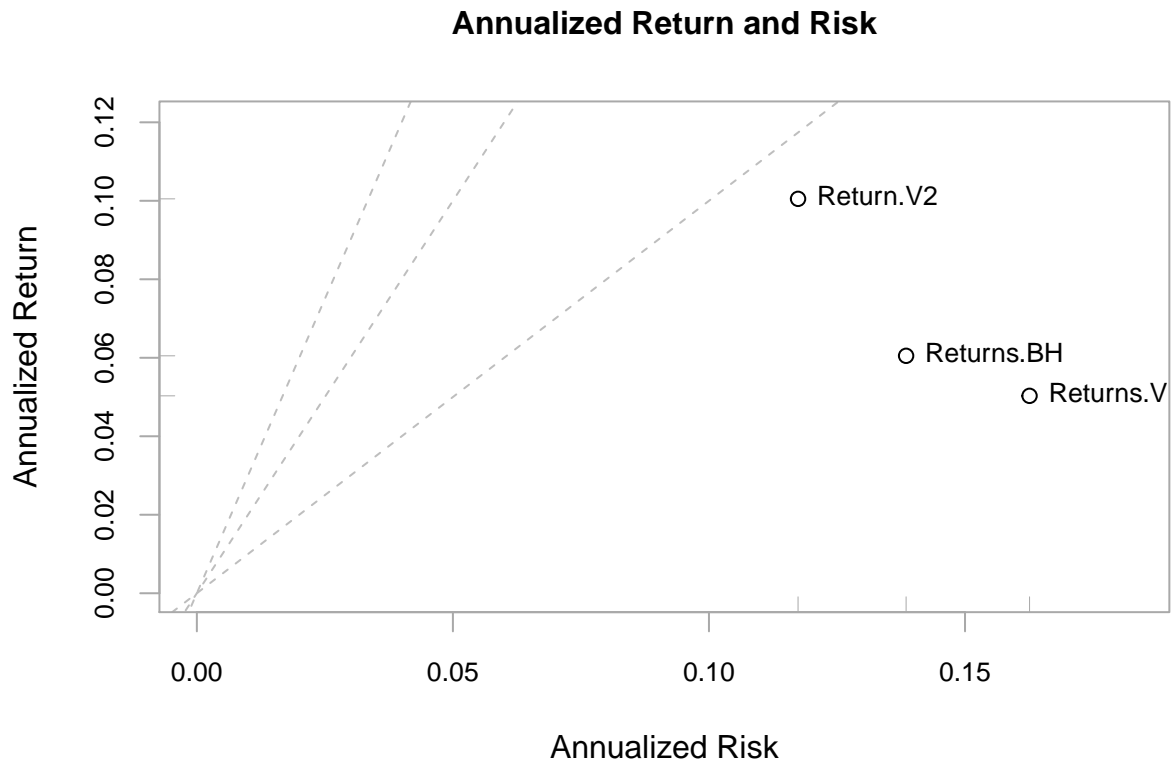


Step 4: Compare risk and return

We can clearly see here that with the improved strategy we were not only able to increase the return drastically by eliminating all long transactions during phases when the market was not generating any returns, but we also reduced the risk of the strategy. The sharpe ratio of the improved strategy is 0.85. The Annualized Return Risk scatter plot shows the impact of the simple improvement of the strategy. The annualized return of the GRS.V2 is about 10 %.

```
rets.bh.v1.v2 <- cbind(rets.v2, rets.v1, rets.bh)
rets.bh.v1.v2 <- na.omit(rets.bh.v1.v2)

chart.RiskReturnScatter(rets.bh.v1.v2)
```



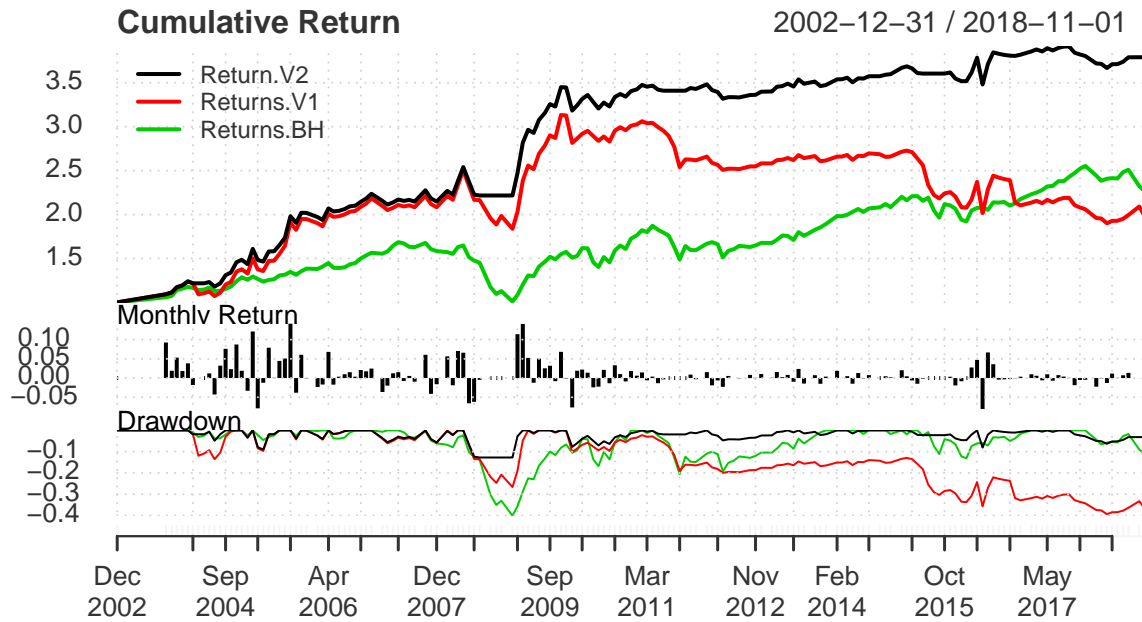
```
table.AnnualizedReturns(rets.bh.v1.v2)
```

Step 5: Compare returns

When we compare the returns of the Buy and hold strategy, the original strategy and the improved GRS.V2 strategy, we can see that the GRS strategies performed equally well until the financial crisis. While the standard strategy was not able to prevent drawdowns, the improved strategy is robust against such situations as investments are only made when the momentum (measured by ROC) is positive. Therefore, during the financial crisis no investments were done. After that the performance is also comparable. From 2011 the GRS.V2 outperforms the standard strategy strongly.

```
charts.PerformanceSummary(rets.bh.v1.v2,geometric=T,wealth.index=TRUE)
```

Return.V2 Performance



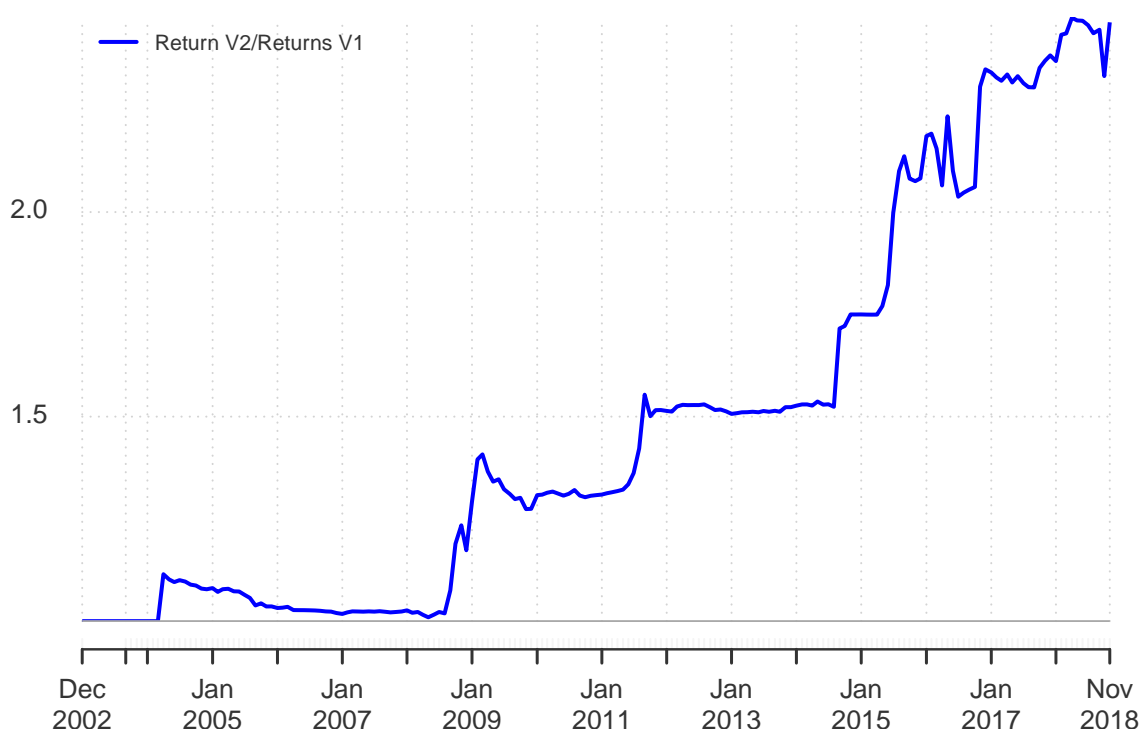
Step 6: Impact of improvement

The impact of improvement can be seen in the following Relative Performance plot which visualizes the relative performance of the GRS.V2 to the standard strategy. We see that there are phases of comparable performance and phases of clear outperformance. The outperformance happens during phases where the market is in a negative trend.

```
chart.RelativePerformance(rets.v2, rets.v1, main="Comparison of strategies",
                           colorset = c("blue", "red"),
                           legend.loc = "topleft")
```

Comparison of strategies

2002-12-31 / 2018-11-01



Step 7: Strategy Risk

```
rets.v2.tmp <- rets.v2
index(rets.v2.tmp) <- as.yearmon(index(rets.v2.tmp))

funddata.alt <- merge(rets.v2.tmp,ffxts)
funddata.alt <- funddata.alt[complete.cases(funddata.alt),]
# Calculate Excess Returns above the risk free rate
funddata.alt$excessrtn <- funddata.alt$monthly.returns*100 - funddata.alt$rf
```

Perform a CAPM Regression

```
capm <- lm(excessrtn ~ mkt.rf, data=funddata.alt)
summary(capm)

##
## Call:
## lm(formula = excessrtn ~ mkt.rf, data = funddata.alt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12205 -0.09166 -0.07019  0.04447  0.35024
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.099463   0.010276   9.679  <2e-16 ***
```



```
## mkt.rf      -0.002781   0.002563  -1.085    0.279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1353 on 177 degrees of freedom
## Multiple R-squared:  0.006607, Adjusted R-squared:  0.0009943
## F-statistic: 1.177 on 1 and 177 DF, p-value: 0.2794
```

Perform a Fama-French 3 factor Regression

```
# Run Fama-French 3 factor Regression
ff3regression <- lm(excessrtn ~ mkt.rf + smb + hml, data=funddata.alt)
summary(ff3regression)
```

```
##
## Call:
## lm(formula = excessrtn ~ mkt.rf + smb + hml, data = funddata.alt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12647 -0.09160 -0.07147  0.04214  0.35207
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.099745   0.010321   9.664  <2e-16 ***
## mkt.rf       -0.003063   0.002819  -1.086    0.279
## smb         -0.001293   0.004757  -0.272    0.786
## hml          0.003294   0.004260   0.773    0.440
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1358 on 175 degrees of freedom
## Multiple R-squared:  0.0103, Adjusted R-squared:  -0.006665
## F-statistic: 0.6071 on 3 and 175 DF, p-value: 0.6112
```

Strategy Improvement V3

Now, that we eliminated the negative impact of down-phases in the market, the next improvement is to see whether we can generate profits during these phases by adding short-rules. A short position is opened for the ETF with the worse performance according to the momentum (ROC). Moreover, it is only opened if the return over the momentum period was negativ. The exit of the short position is triggered when ETF is not ranked at the last position anymore OR when the return over the last momentum period is positiv again.

```
strategyname.alt.v3 <- paste(strategyname, ".", "v3", sep = "")
portfolioname.alt.v3 <- paste(portfolioname, ".", "v3", sep = "")
accountname.alt.v3 <- paste(accountname, ".", "v3", sep = "")
rm.strat(strategyname.alt.v3)
initPortf(portfolioname.alt.v3, symbollist, initDate=initdate, currency="USD")
initAcct(accountname.alt.v3, portfolios=portfolioname.alt.v3, initDate=initdate, initEq=startingcapital, cu
initOrders(portfolio=portfolioname.alt.v3, initDate=initdate)
strategy(strategyname.alt.v3, store=TRUE)
strat <- getStrategy(strategyname.alt.v3)
```

New Strategy Rules

```
# Indicator
add.indicator(strategyname.alt.v3,
              name = "sigThreshold",
              arguments=list(column="Rank",
                             threshold=buypositions, relationship="lte"),
              label = "bpos")

#Signal Long
add.signal(strategyname.alt.v3,
           name="sigFormula",
           arguments=list(columns=c("Rank.bpos", "PosRets"),
                          formula="(Rank.bpos == 1) & (PosRets==1)",
                          label="longEntry")

add.signal(strategyname.alt.v3,
           name="sigThreshold",
           arguments=list(column="Rank",
                          threshold=buypositions,
                          relationship="gt"),
           label="longExit")

# Rules Long
add.rule(strategyname.alt.v3,
         name="ruleSignal",
         arguments=list(sigcol="longEntry",
                        sigval=TRUE,
                        orderqty=100,
                        ordertype="market",
                        orderside="long",
                        replace=FALSE,
                        prefer="Open",
                        osFUN='osInvestAll',
                        orderset="ocolong",
                        TxnFees=txncost),
         type="enter",
         label="LE",
         path.dep=TRUE)

add.rule(strategyname.alt.v3,
         name="ruleSignal",
         arguments=list(sigcol="longExit",
                        sigval=TRUE,
                        orderqty="all",
                        ordertype="market",
                        orderside="long",
                        replace=FALSE,
                        prefer="Open",
                        orderset="ocolong",
                        TxnFees=txncost),
         type="exit",
         label = "LX",
         path.dep=TRUE)
```

```

# Short indicator
symbolCount <- length(symbollist)
shortThreshold <- symbolCount-buypositions

add.indicator(strategyname.alt.v3,
              name = "sigThreshold",
              arguments=list(column="Rank",
                             threshold=shortThreshold,
                             relationship="gt"),
              label = "ShortBpos")

# Short Signals
add.signal(strategyname.alt.v3,
           name="sigFormula",
           arguments=list(columns=c("Rank.ShortBpos", "PosRets"),
                          formula="(Rank.ShortBpos == 1) & (PosRets==0)"),
           label="shortEntry")

add.signal(strategyname.alt.v3,
           name="sigFormula",
           arguments=list(columns=c("Rank.ShortBpos", "PosRets"),
                          formula="(Rank.ShortBpos == 0) | (PosRets == 1)"),
           label="shortExit")

# Short Rules
add.rule(strategyname.alt.v3,
         name="ruleSignal",
         arguments=list(sigcol="shortEntry",
                        sigval=TRUE,
                        orderqty=100,
                        ordertype="market",
                        orderside="short",
                        replace=FALSE,
                        prefer="Open",
                        osFUN='osInvestAll',
                        TxnFees=txncost),
         type="enter",
         path.dep=TRUE,
         label="ShortEntry")

add.rule(strategyname.alt.v3,
         name="ruleSignal",
         arguments=list(sigcol="shortExit",
                        sigval=TRUE,
                        orderqty="all",
                        ordertype="market",
                        orderside="short",
                        replace=FALSE,
                        prefer="Open",
                        TxnFees=txncost),
         type="exit",
         path.dep=TRUE,
         label="ShortExit")

```

Apply the strategy

```
applyStrategy(strategy=strategyname.alt.v3,portfolios=portfolioname.alt.v3)

## Warning in .updatePosPL(Portfolio = pname, Symbol = as.character(symbol), :
## Could not parse 2018-11-01 as ISO8601 string, or one/bothends of the range
## were outside the available prices: 2003-09-02/2018-11-01. Using 2018-11-01/
## instead.

updatePortf(strategyname.alt.v3)
updateAcct(strategyname.alt.v3)
updateEndEq(strategyname.alt.v3)
```

Analysis

Retrieve Equity of GRS.V3 strategy

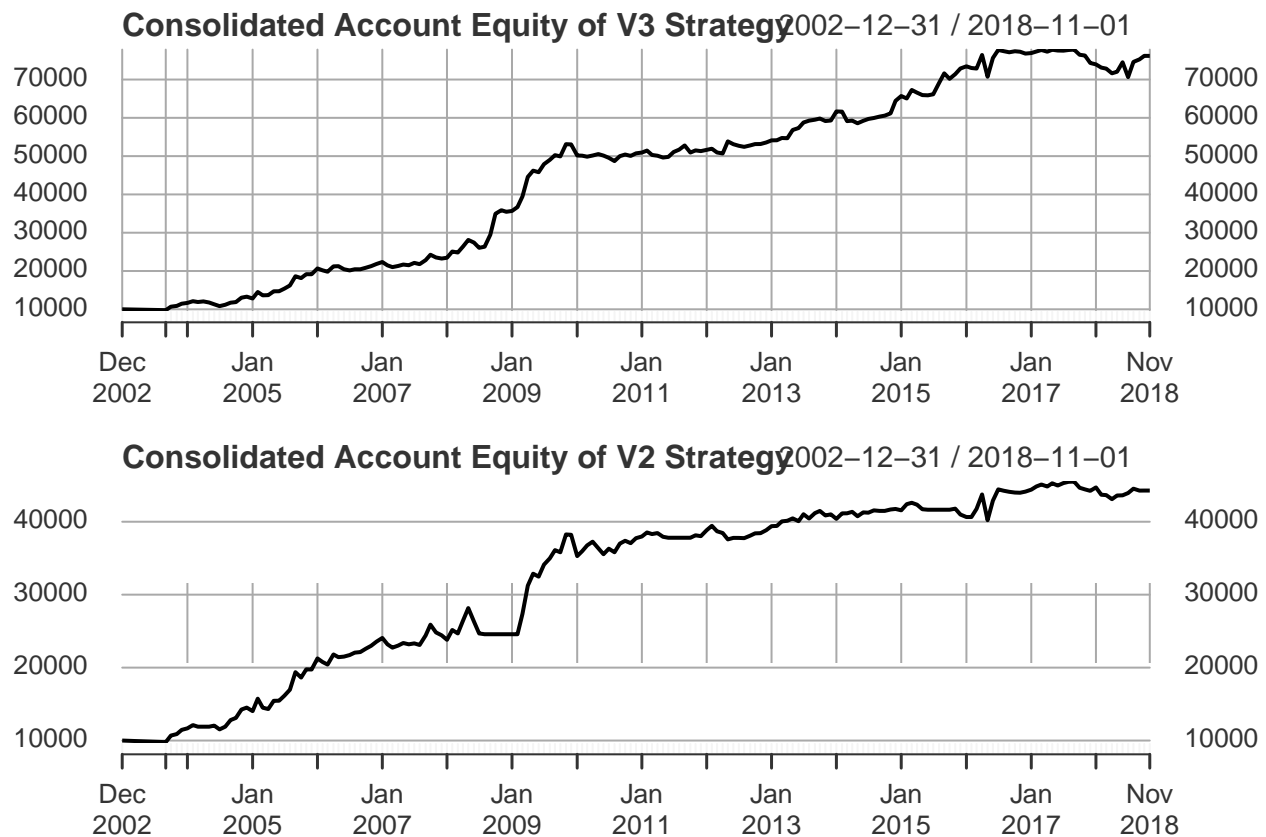
```
acct.alt.v3 <- getAccount(accountname.alt.v3)

equity.v3 <- acct.alt.v3$summary$End.Eq
colnames(equity.v3) <- c("Equity.V3")
```

Visualize Equity of GRS.V3

As we can see, the curves look very similar, besides the fact that the scale is different for the GRS.V3 strategy. Through the implementation of short selling rules we are able to increase the returns by a significant amount.

```
par(mfrow=c(2,1))
plot(equity.v3, main="Consolidated Account Equity of V3 Strategy")
plot(equity.v2, main="Consolidated Account Equity of V2 Strategy")
```



```
par(mfrow=c(1,1))
```

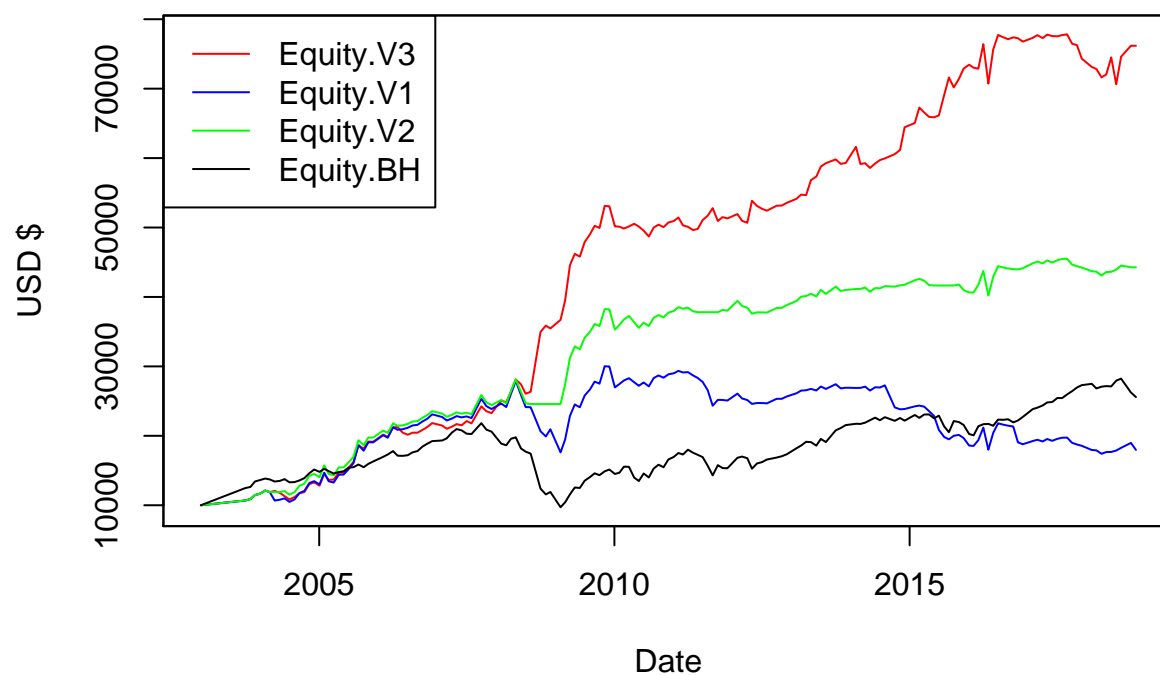
Comparison of all strategies

Especially the comparison of the equity development of all 4 strategies shows the significant difference between the strategies and the impact of the implemented improvements. While the first improvement just eliminated the negative impacts of down-markets, the second improvement used short sells in down-phases in order to generate even more profits.

```
equity.bh.v1.v2.v3 <- cbind(equity.v3, equity.v1, equity.v2, equity.bh)
equity.bh.v1.v2.v3 <- na.omit(equity.bh.v1.v2.v3)
```

```
plot.zoo(equity.bh.v1.v2.v3, plot.type="single", col=c('red', 'blue', 'green', 'black'), main="Final Equity (")
legend(x="topleft", legend=colnames(equity.bh.v1.v2.v3), col=c("red", "blue", "green", "black"), lty=1)
```

Final Equity Comparison



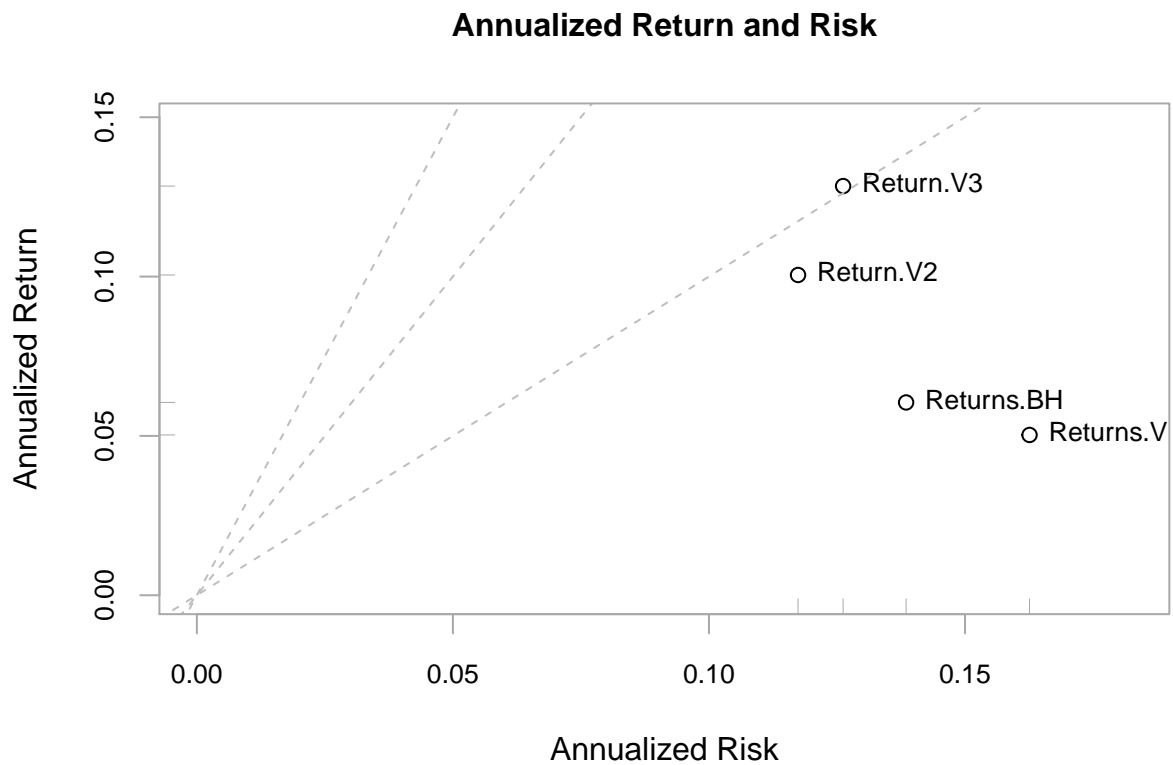
Visualizing Risk and Return

The visualization and comparison of the Risk and Return scatter shows that the risk increased minimally but the return increased significantly, compared to the GRS.V2 strategy. The Sharpe Ratio is now above 1. The annualized return is 12.8 % with an annualized standard deviation of 12.6%.

```
rets.v3 <- monthlyReturn(equity.v3, type = "arithmetic")
colnames(rets.v3) <- c("Return V3")

rets.bh.v1.v2.v3 <- cbind(rets.v3, rets.v1, rets.v2, rets.bh)
rets.bh.v1.v2.v3 <- na.omit(rets.bh.v1.v2.v3)

chart.RiskReturnScatter(rets.bh.v1.v2.v3)
```



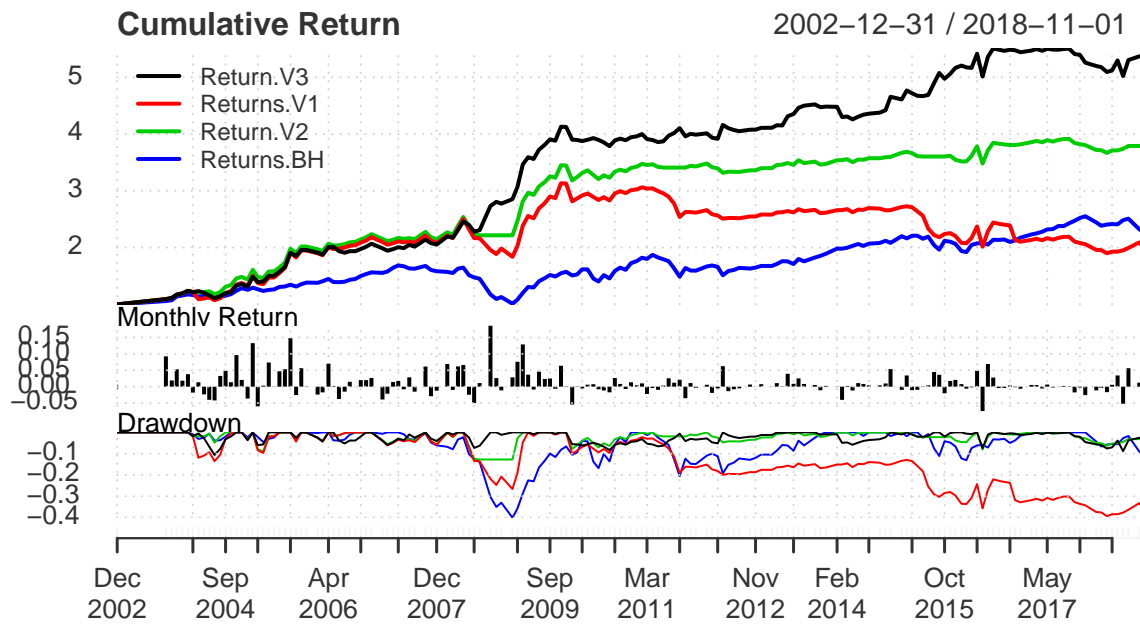
```
table.AnnualizedReturns(rets.bh.v1.v2.v3)
```

Compare performance

The comparison of the returns of all strategies shows again what we already saw in the equity comparison plot. In contrast to the V2 strategy, the GRS.V3 performed trades throughout the entire period and generated even in phases of global crisis returns.

```
charts.PerformanceSummary(rets.bh.v1.v2.v3,geometric=T,wealth.index=TRUE)
```

Return.V3 Performance



Direct comparison GRS.V2 and GRS.V3

The relative performance of the V3 strategy compared to the V2 strategy shows that especially during the financial crisis the strategy which included shorting outperformed the GRS.V2 strategy.

```
chart.RelativePerformance(rets.v3, rets.v2, main="Comparison of strategies", colorset = c("blue", "red"))
```

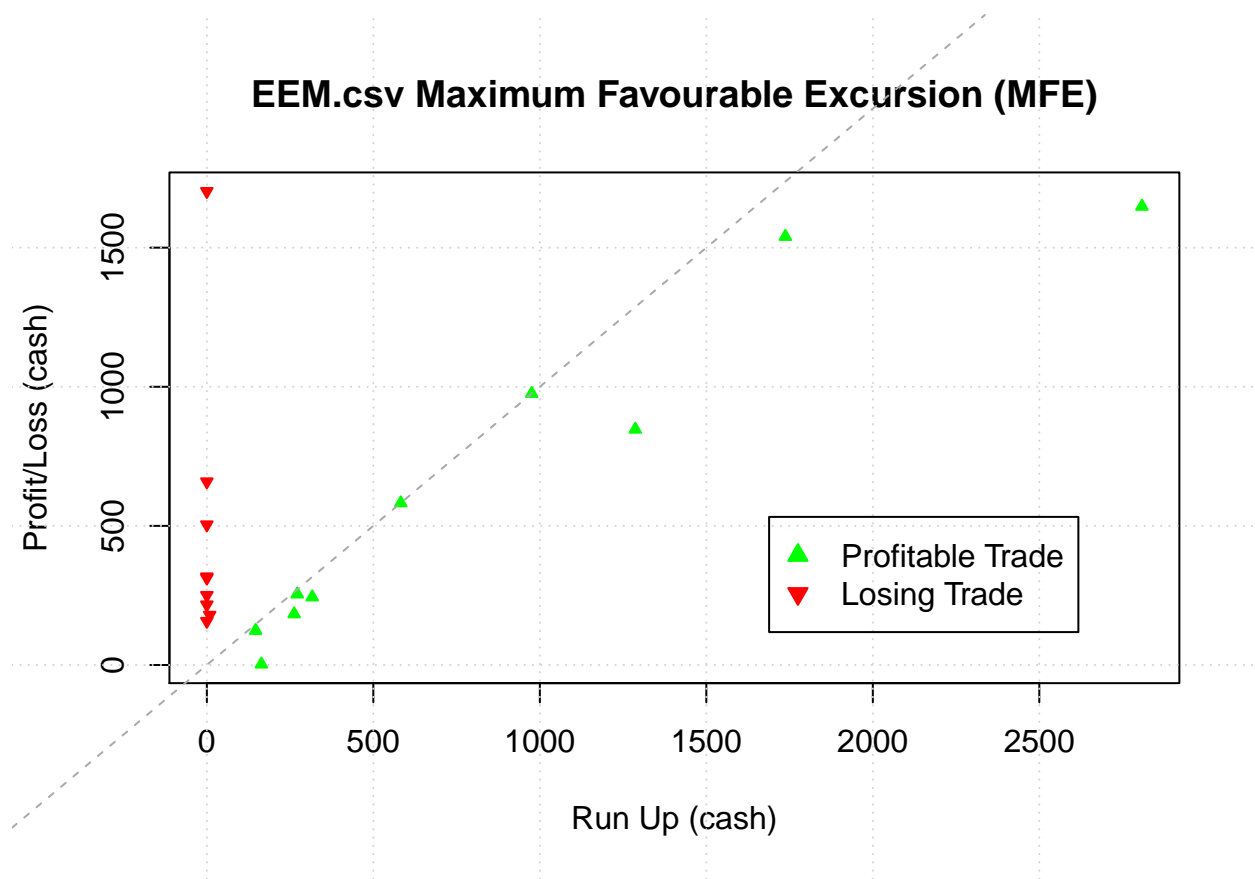


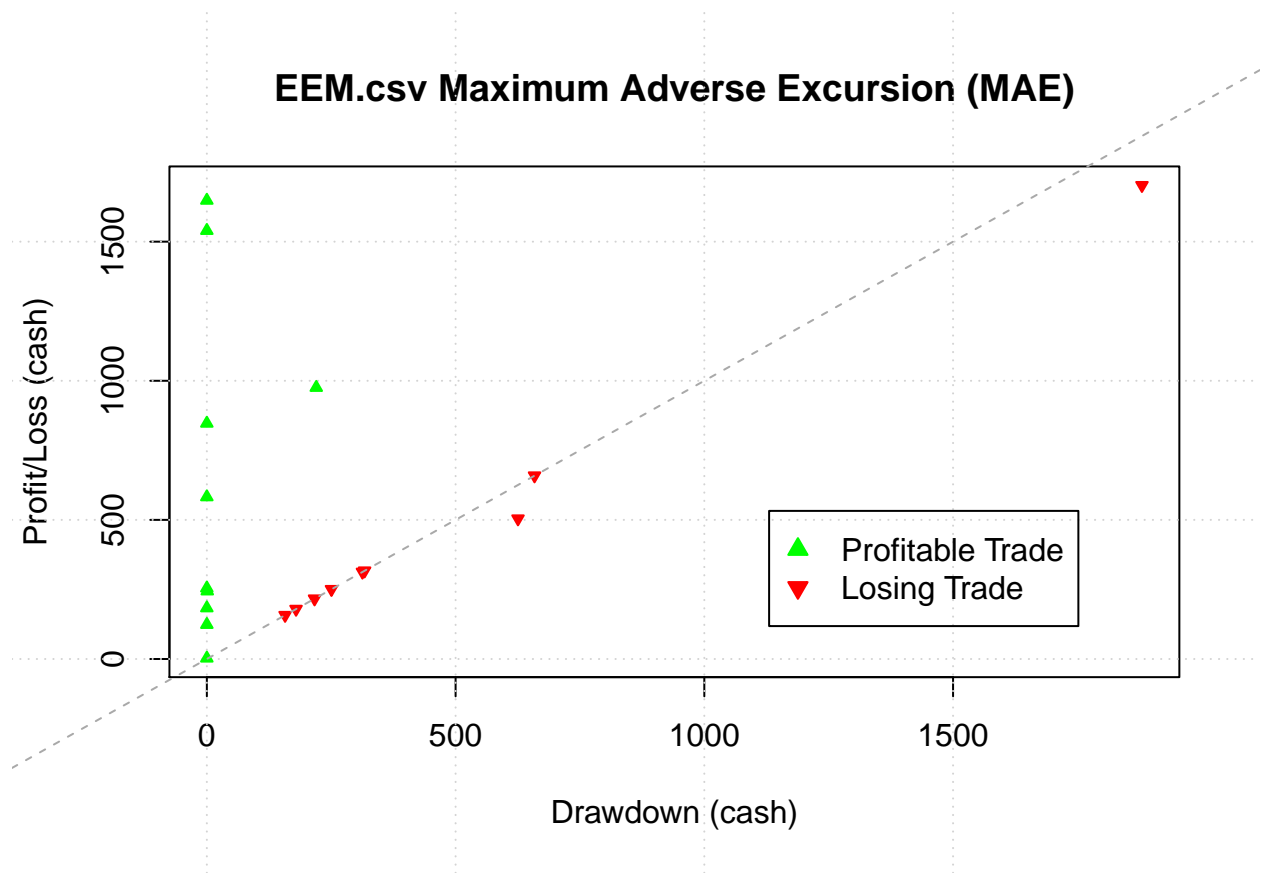

Could we have performed better

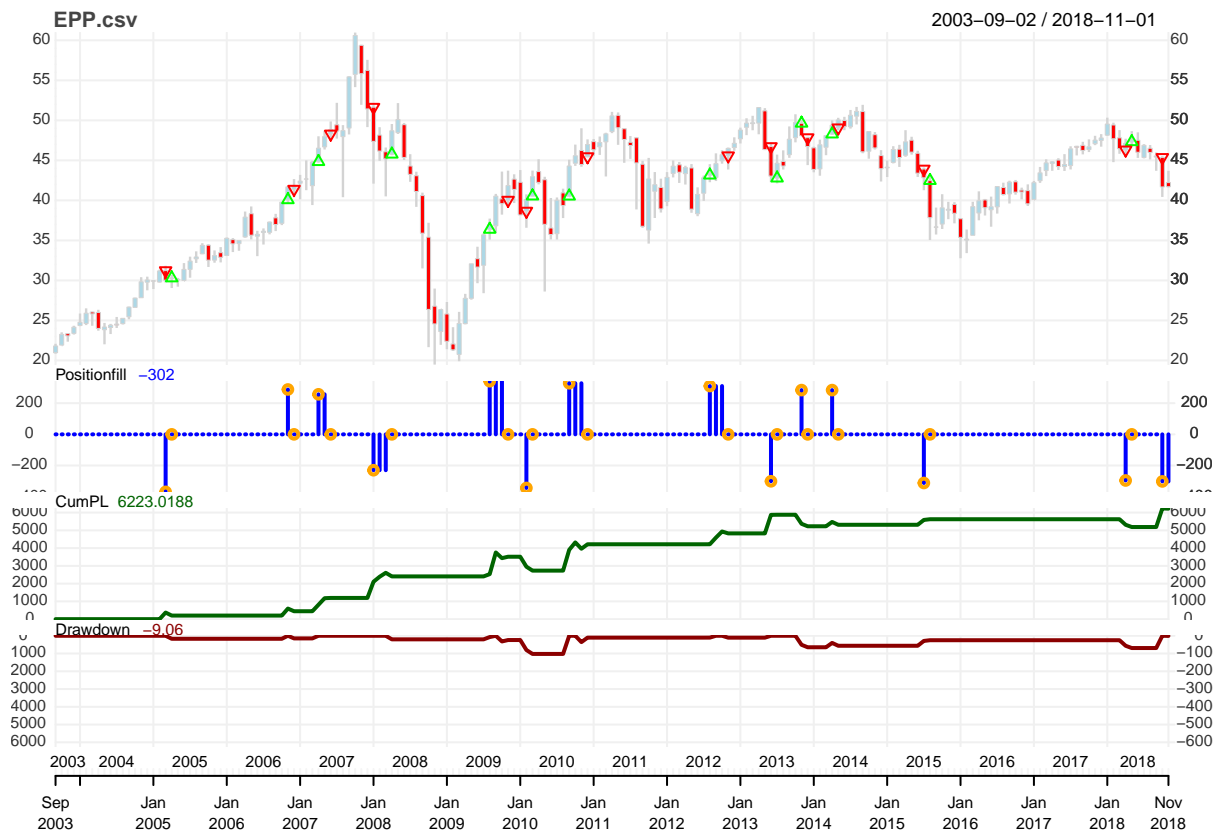
In the following, the MAE and MFE plots for all traded instruments are plotted. We can see, that there is still some room for improvement which could be improved by initiating stop loss or trailing stop orders.

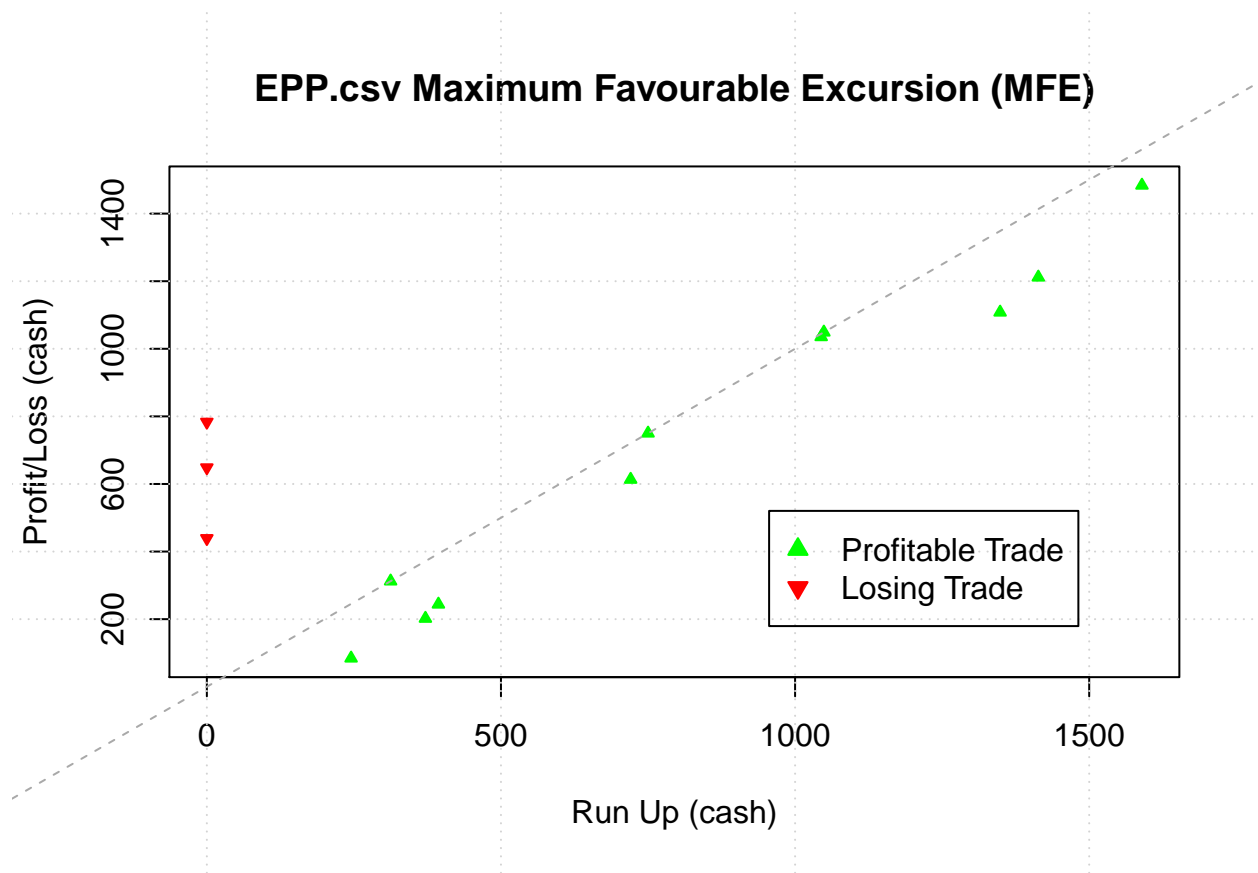
```
for (symbol in symbollist)
{
  if (nrow(getTxns(portfolioname.alt.v3,Symbol=symbol)) > 1)
  {
    symboldata <- get(symbol)
    chart.Posn(portfolioname.alt.v3,Symbol=symbol,type='candlesticks', theme=myTheme,subset=daterang
    chart.ME(portfolioname.alt.v3, Symbol=symbol, type="MFE")
    chart.ME(portfolioname.alt.v3, Symbol=symbol, type="MAE")
  }
}
```



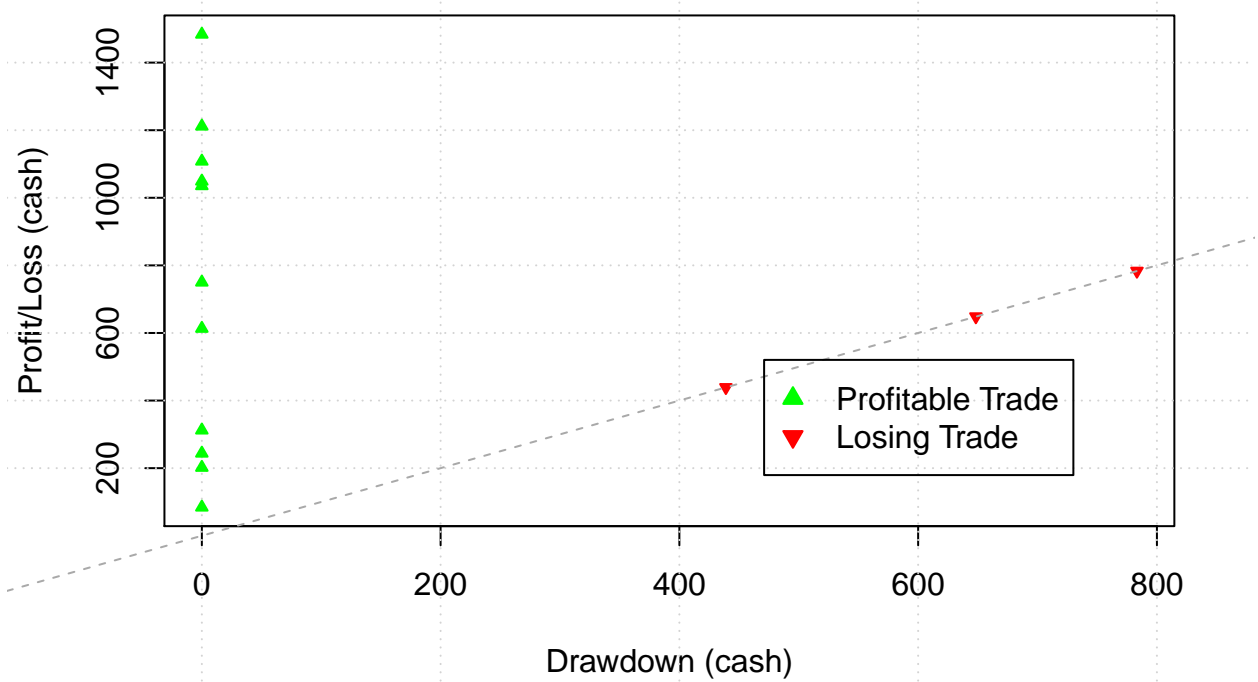


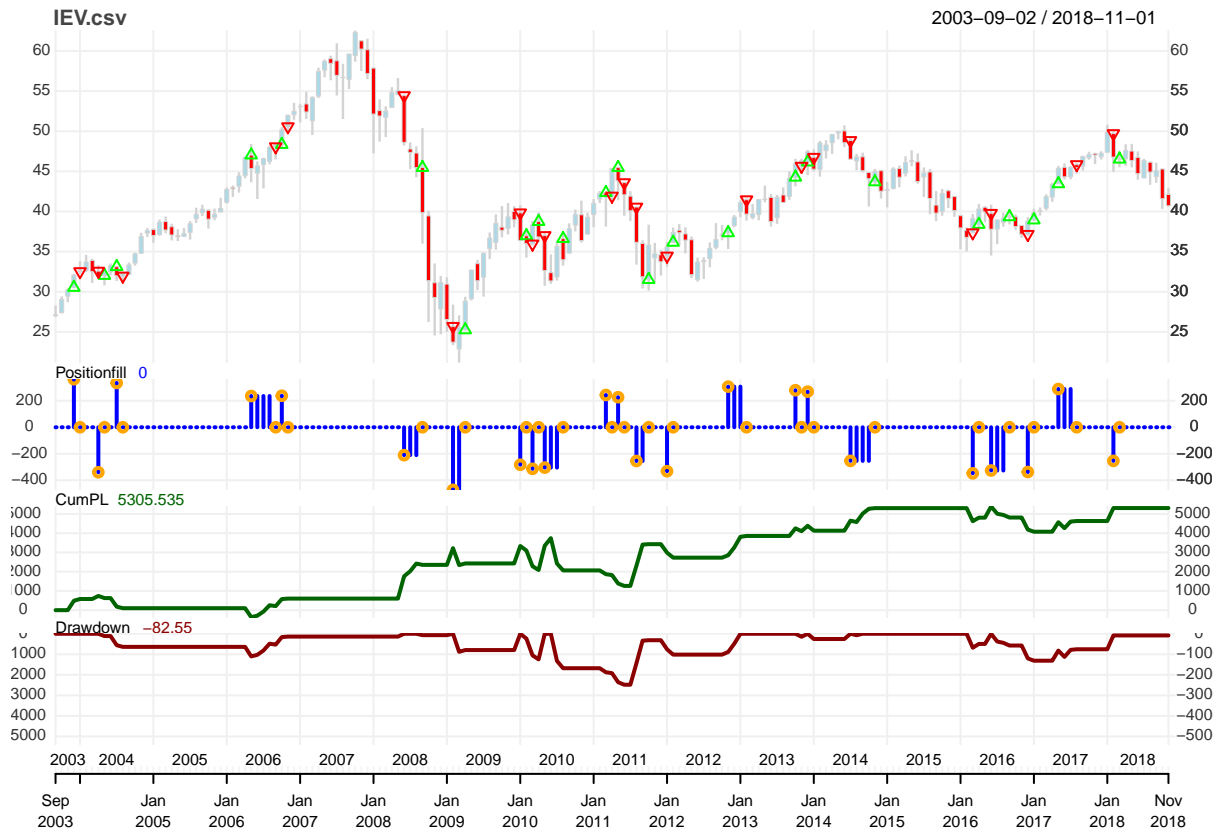




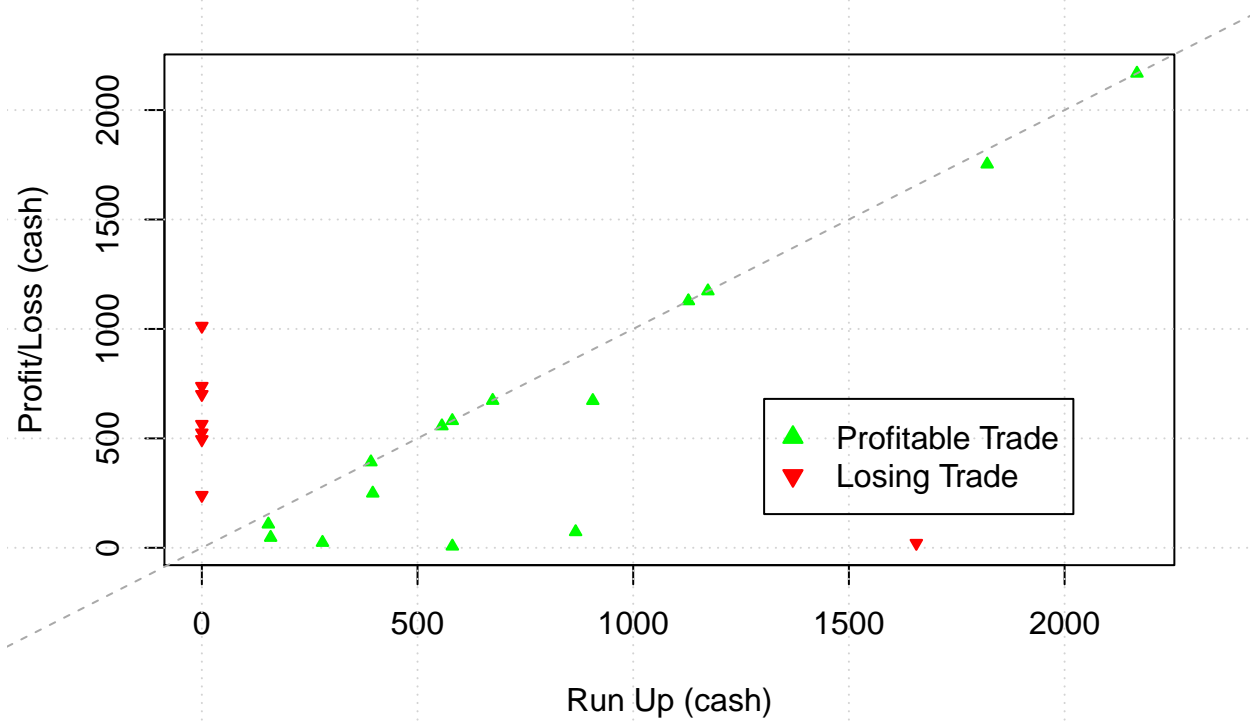


EPP.csv Maximum Adverse Excursion (MAE)

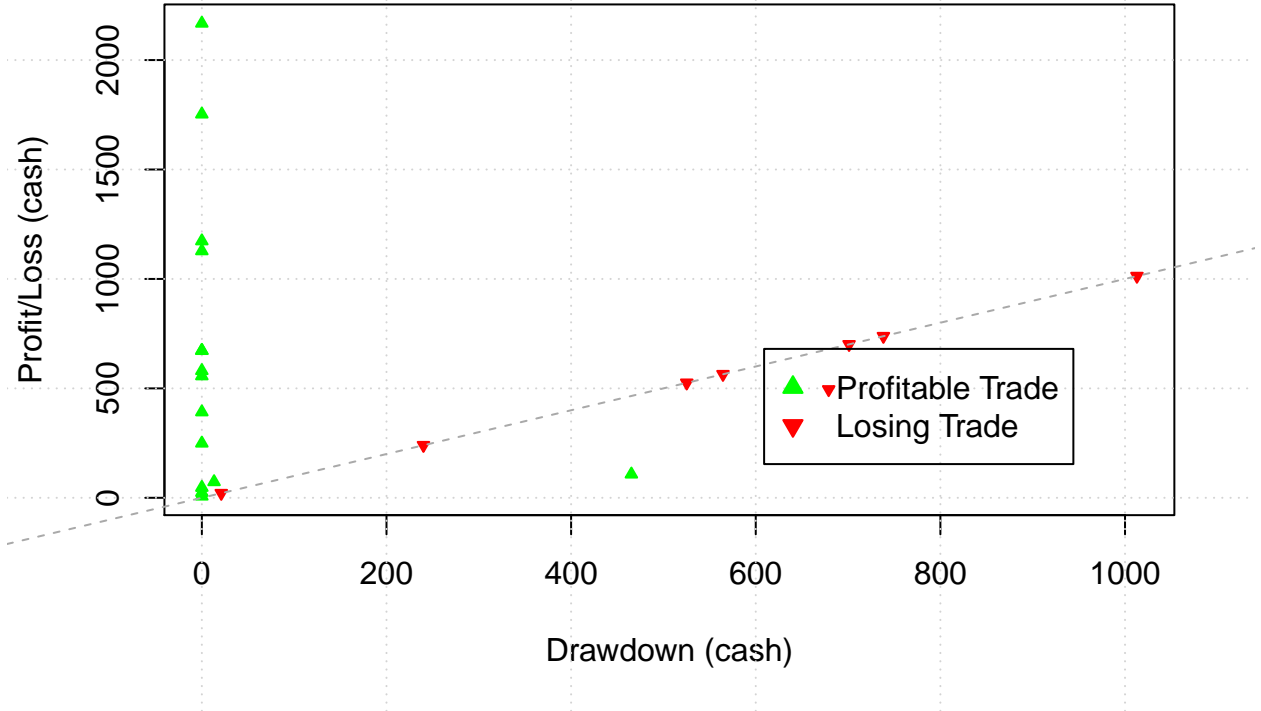


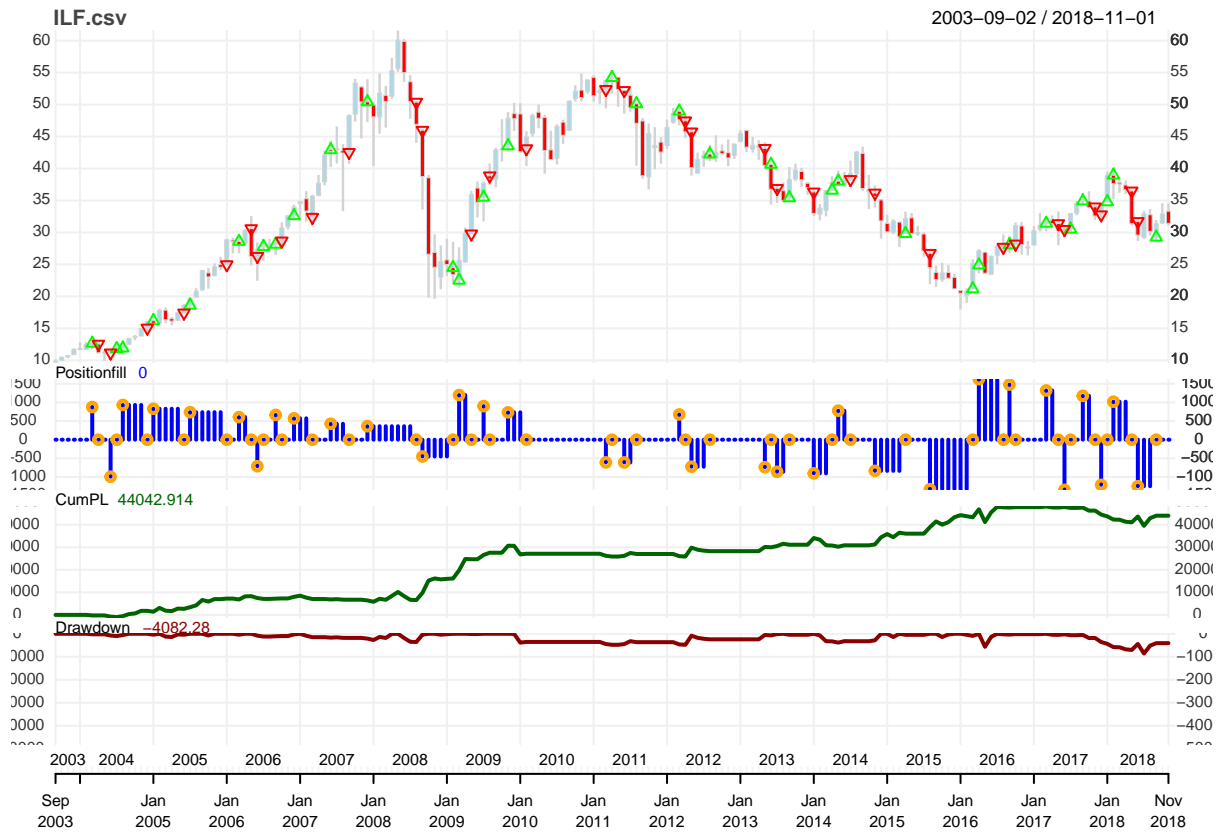


IEV.csv Maximum Favourable Excursion (MFE)

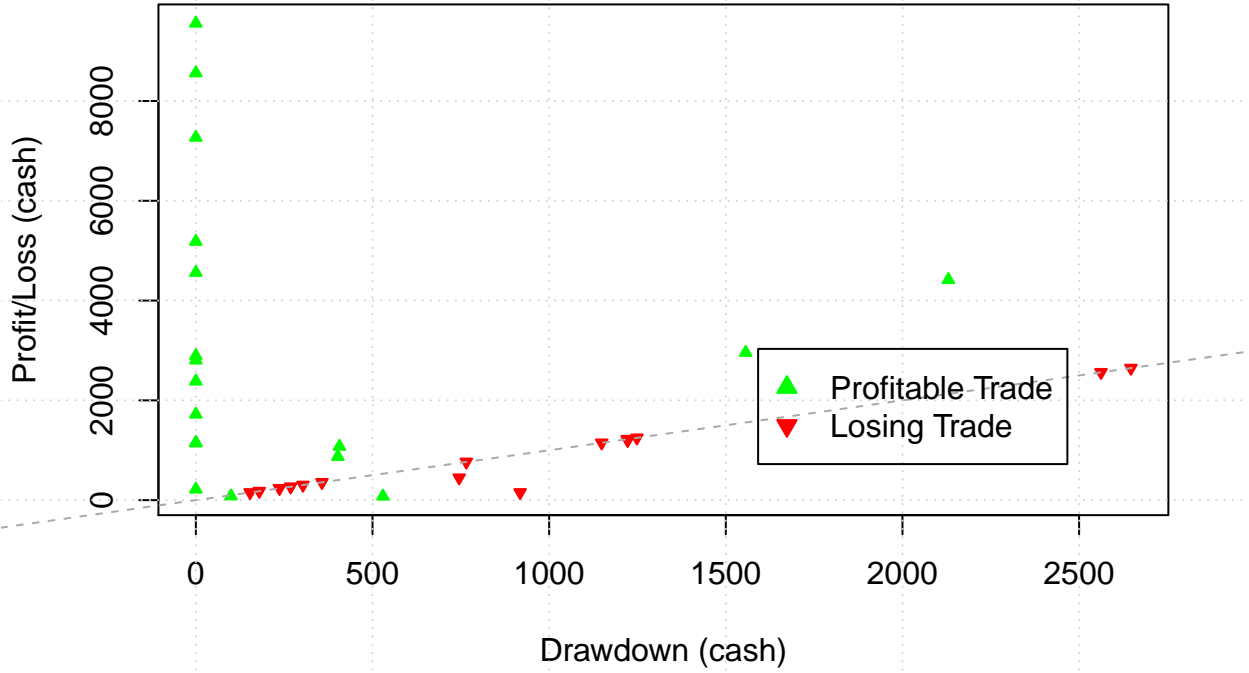


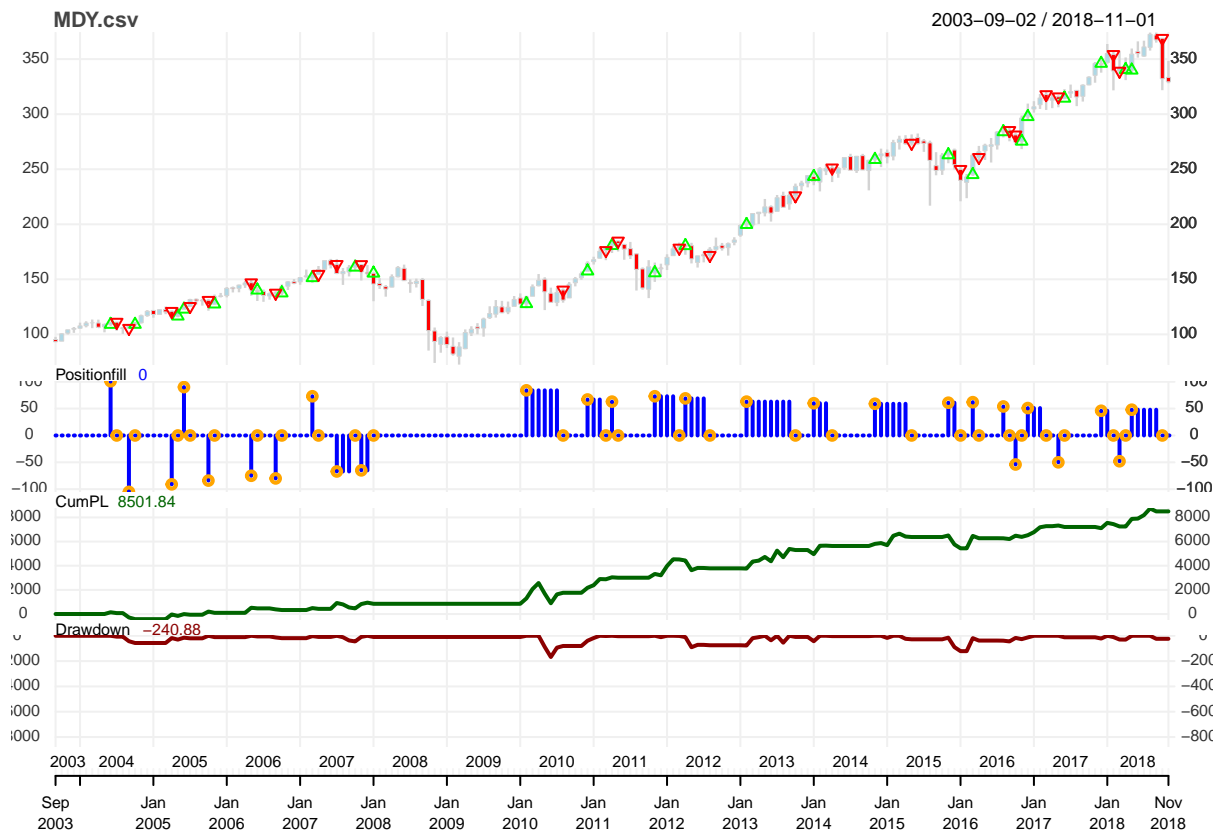
IEV.csv Maximum Adverse Excursion (MAE)

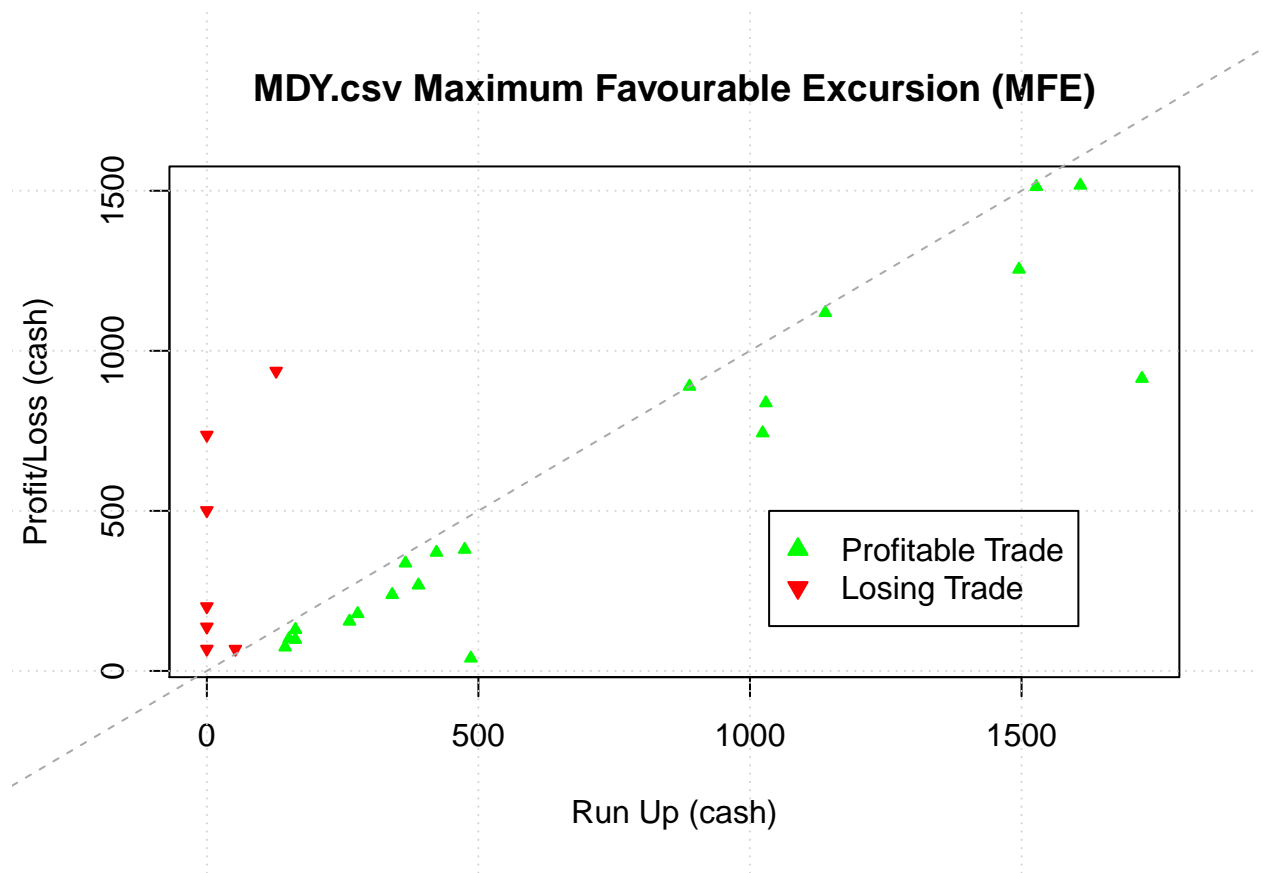




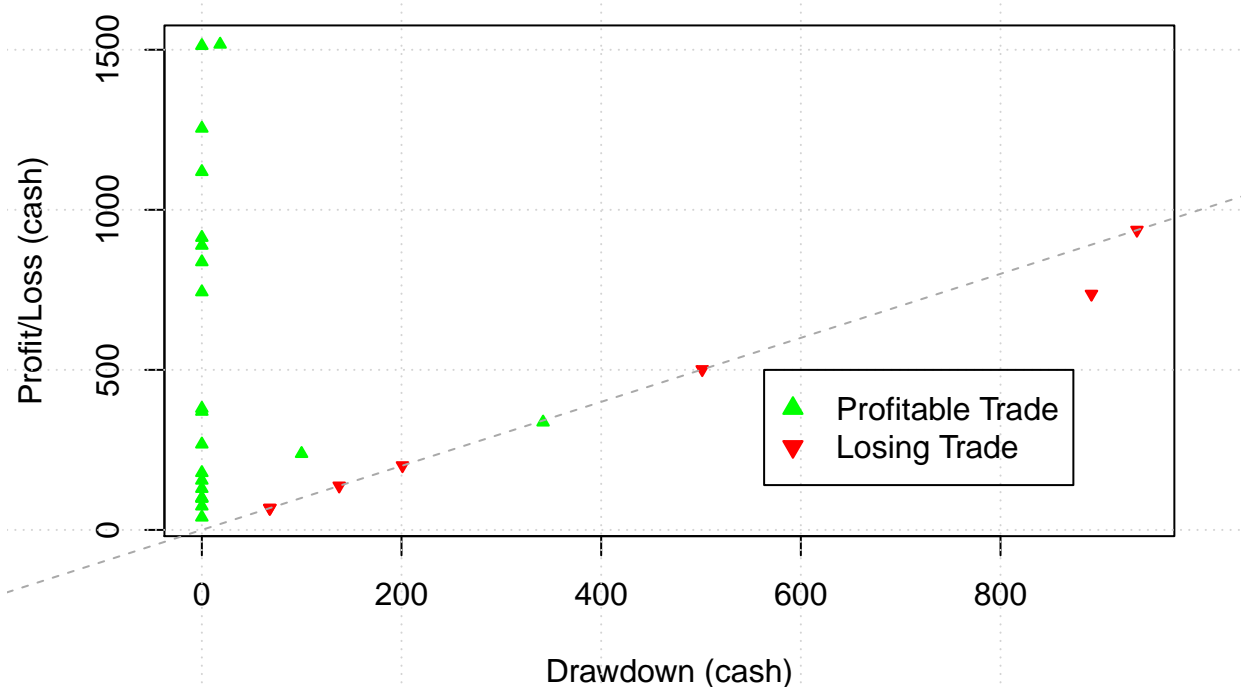
ILF.csv Maximum Adverse Excursion (MAE)







MDY.csv Maximum Adverse Excursion (MAE)



Strategy Risk

```
rets.v3.tmp <- rets.v3
index(rets.v3.tmp) <- as.yearmon(index(rets.v3.tmp))

funddata.alt.v3 <- merge(rets.v3.tmp,ffxts)
funddata.alt.v3 <- funddata.alt.v3[complete.cases(funddata.alt.v3),]
# Calculate Excess Returns above the risk free rate
funddata.alt.v3$excessrtn <- funddata.alt.v3$monthly.returns*100 - funddata.alt.v3$rf
head(funddata.alt.v3)
tail(funddata.alt.v3)
```

Perform a CAPM Regression

```
capm <- lm(excessrtn ~ mkt.rf, data=funddata.alt.v3)
summary(capm)
```

```
##
## Call:
## lm(formula = excessrtn ~ mkt.rf, data = funddata.alt.v3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12205 -0.09166 -0.07019  0.04447  0.35024
##
## Coefficients:
```



```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.099463   0.010276   9.679  <2e-16 ***
## mkt.rf      -0.002781   0.002563  -1.085   0.279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1353 on 177 degrees of freedom
## Multiple R-squared:  0.006607, Adjusted R-squared:  0.0009943
## F-statistic: 1.177 on 1 and 177 DF, p-value: 0.2794
```

Perform a Fama-French 3 factor Regression

```
# Run Fama-French 3 factor Regression
ff3regression <- lm(excessrtn ~ mkt.rf + smb + hml, data=funddata.alt.v3)
summary(ff3regression)
```

```
##
## Call:
## lm(formula = excessrtn ~ mkt.rf + smb + hml, data = funddata.alt.v3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12647 -0.09160 -0.07147  0.04214  0.35207
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.099745   0.010321   9.664  <2e-16 ***
## mkt.rf      -0.003063   0.002819  -1.086   0.279
## smb        -0.001293   0.004757  -0.272   0.786
## hml         0.003294   0.004260   0.773   0.440
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1358 on 175 degrees of freedom
## Multiple R-squared:  0.0103, Adjusted R-squared: -0.006665
## F-statistic: 0.6071 on 3 and 175 DF, p-value: 0.6112

#Conclusion
```