Global Rotation Strategy

Philipp Epstein

by

Philipp Epstein

Student ID: 13591160

Date: November 30, 2018

Submitted to:
Dr Bruce Vanstone
Bond Business School
Bond University

Contents

1 Introduction

The following report summarizes the results of an analysis of a Financial Trading System. It backtests the Global Rotation Strategy (GRS) with 5 ETFs over a period from 2003 until 2018. First, the GRS is implemented and trades are performed based on the rules defined by the strategy which was published on the website SeekingAlpha. Additionally, the strategy is benachmarked against a buy and hold strategy. As the GRS uses ETFs which represent the entire world market, the MSCI World Index is used as a benchmark. After the benchmark a possible improvement of the strategy is presented and implemented. Moreover, another improvement step is proposed which includes short selling. All the results of the strategies are presented and measured against each other. Furthermore, a risk analysis is performed based on the CAPM and 3-factor model to evaluate whether the strategies generate excess returns compared to the market.

2 Trading Idea

2.1 Global Rotation Strategy

The GRS is based on a ranking approach. The used ETFs are ranked based on momentum which is measured by the Rate of Change (ROC) indicator. The lookback period is variable and can be tested for different values. The same is true for the invested period. For this report the lookback period will be tested for different values between one month and 12 months. The best result will be presented in the report. The time period for being invested is one month. The strategy suggests to buy the best ranked ETF. Many variations and extensions of the strategy exist which will be used and presented throughout the report.

2.2 Choice of ETFs

In order to achieve substantial returns it is benefitial for a ranking strategy that the selected ETFs are not too volatile and have long-term visible trends. It can be assumed that the best results can be achieved if the used ETFs have a comparable volatility. A large market volume makes sure that the ETFs cannot be easily manipulated. This is the case for all proposed ETFs in the strategy. The 5 chosen ETFs cover the world market. The following ETFs are used:

US Market (MDY- S&P MidCap 400 SPDRs) Europe (IEV- iShares S&P Europe 350 Index Fund) Emerging Markets (EEM- iShares MSCI Emerging Markets) Latin America (ILF- iShares S&P Latin America) Pacific region (EPP - iShares MSCI Pacific ex-Japan)

According to the strategy, the ETFs follow slightly different economic cycles. There are certain periods where one market outperforms another until it becomes overpriced. As a result investors begin to shift there money in other cheaper valued markets. With the GRS strategy, this trends can be followed.

3 Implementation of the trading strategy

3.1 Part A: Initialization

3.1.1 Step 1: General Setup

Pre-Step: Clear the environment.

```r
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.

```r
library(quantstrat)
currency("USD")
Sys.setenv(TZ="UTC")
initdate <- '2003-08-31'
startdate <- '2003-09-02'
enddate <- '2018-12-31'
portfolioname <- "GRS"
accountname <- portfolioname
strategyname <- portfolioname
```

3.1.2 Step 2: Setting the variables

```r
#Set values:

startingcapital <- 10000

buypositions <- 1
roc.var <- 3

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

3.1.3 Step 3: Preset for plotting graphs

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

3.1.4 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 dr
    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)
}
```

3.1.4.1 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))
```

3.1.4.2 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)
```

3.1.4.3 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)))
```

3.1.4.4 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
```

3.1.4.5 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)
}
```

```r
# 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)
}
```

3.1.5 Step 5: Order sizing function

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

```r
# order sizing function
osInvestAll <- function (data, timestamp, orderqty,
                         ordertype, orderside, portfolio,
                         symbol, ruletype, ..., orderprice)
{
    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
           }
}
```

3.1.6 Step 6: Transaction cost function

In order to include realistic transaction costs, the following transaction costs function is used. Depending on the order volume we have increeasing transactioncosts. Ordervolume above 28000

```
# transaction fees function
txnc <- function(TxnQty, TxnPrice, Symbol, ...)
{
  ordervalue = abs(TxnQty) * TxnPrice

  if (ordervalue <= 5000)
    return (-19.95)

  if (ordervalue <= 10000)
    return (-24.95)

  if (ordervalue <= 28000)
    return (-29.95)

  return(-0.11/100 * ordervalue)
}
```

3.2 Part B: Apply Strategy

3.2.1 Step 1: Create Strategy, Portfolio and Account

```
rm.strat(strategyname) # remove if it already exists
initPortf(portfolioname,symbollist,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)
```

3.2.2 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 euqal 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.

```r
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="txnc",
                        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="txnc",
                        orderset="ocolong"),
         type="exit",
         path.dep=TRUE,
         label="exitLONG")
```

3.2.3 Step 3: Run the strategy

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

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

3.2.4 Step 4: Check orderbook

Check the orderbook to verify that all transactions are made correctly. The orderbook can be found in the appendix.

3.3 Analyze strategy

3.3.1 Step 1: Endequity and Return

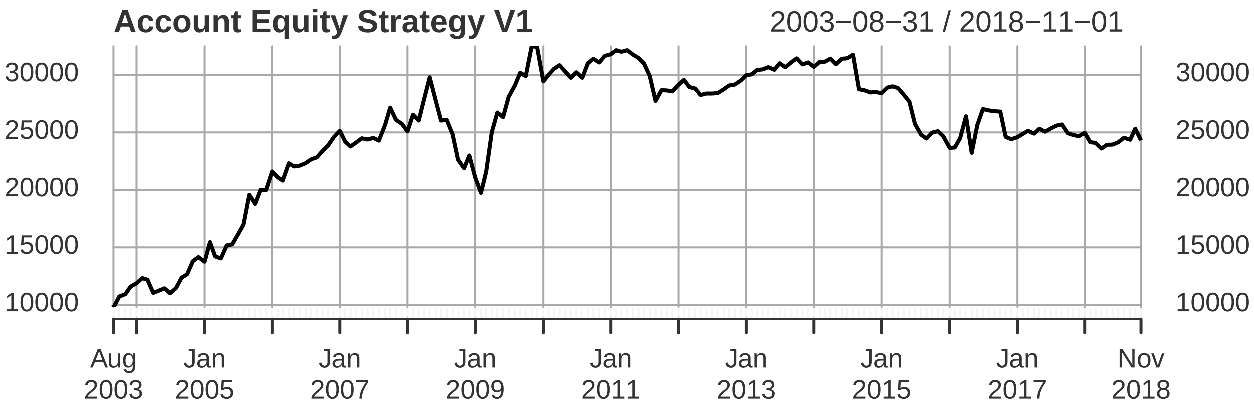
Calculate endequity and return of the strategy

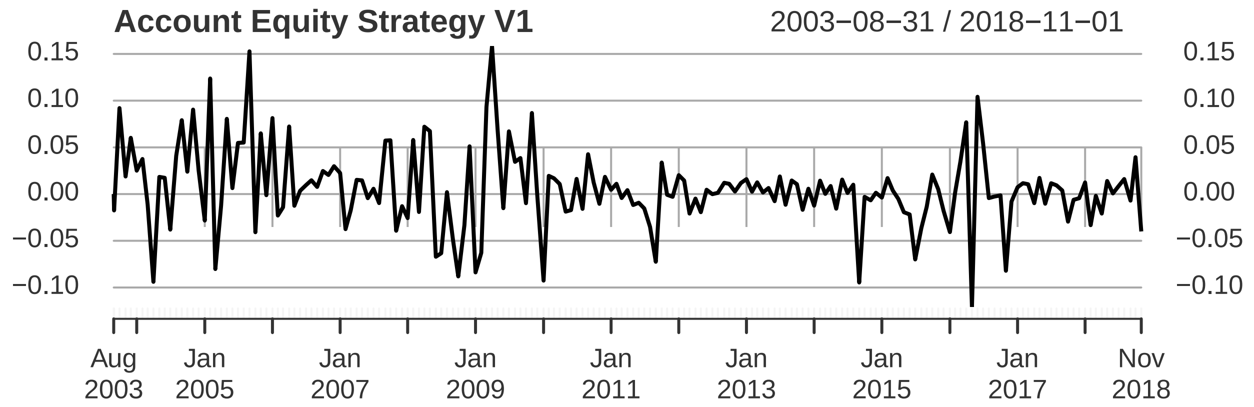
```r
# end equity
acct <- getAccount(accountname)
equity.v1 <- acct$summary$End.Eq
colnames(equity.v1) <- c("Equity V1")

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

3.3.2 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.

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



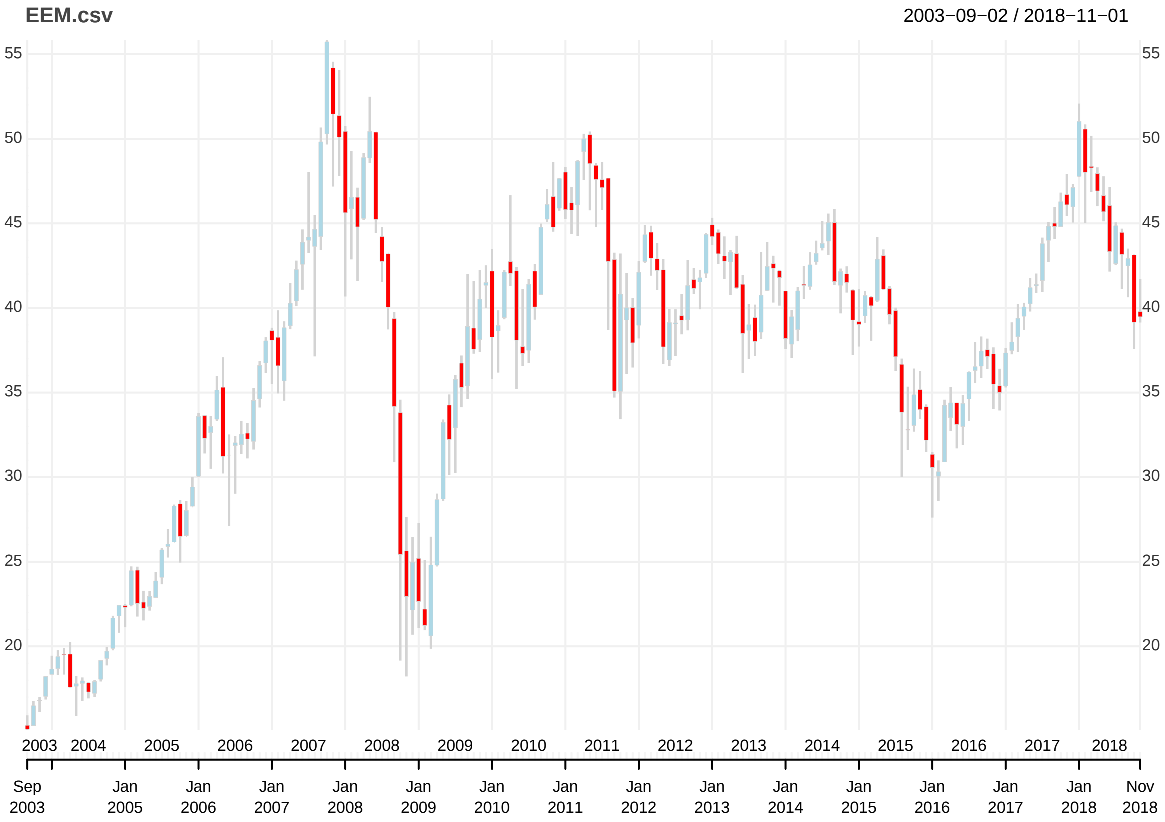


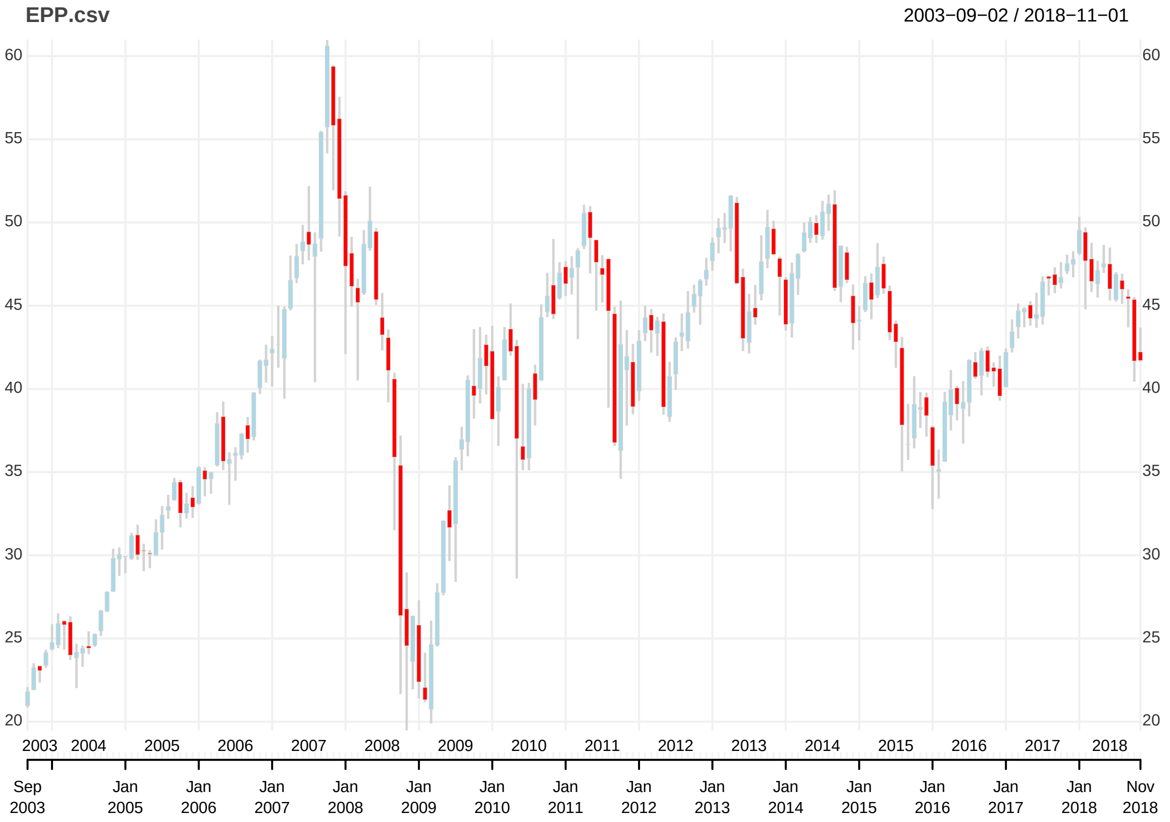
```
par(mfrow=c(1,1))
```

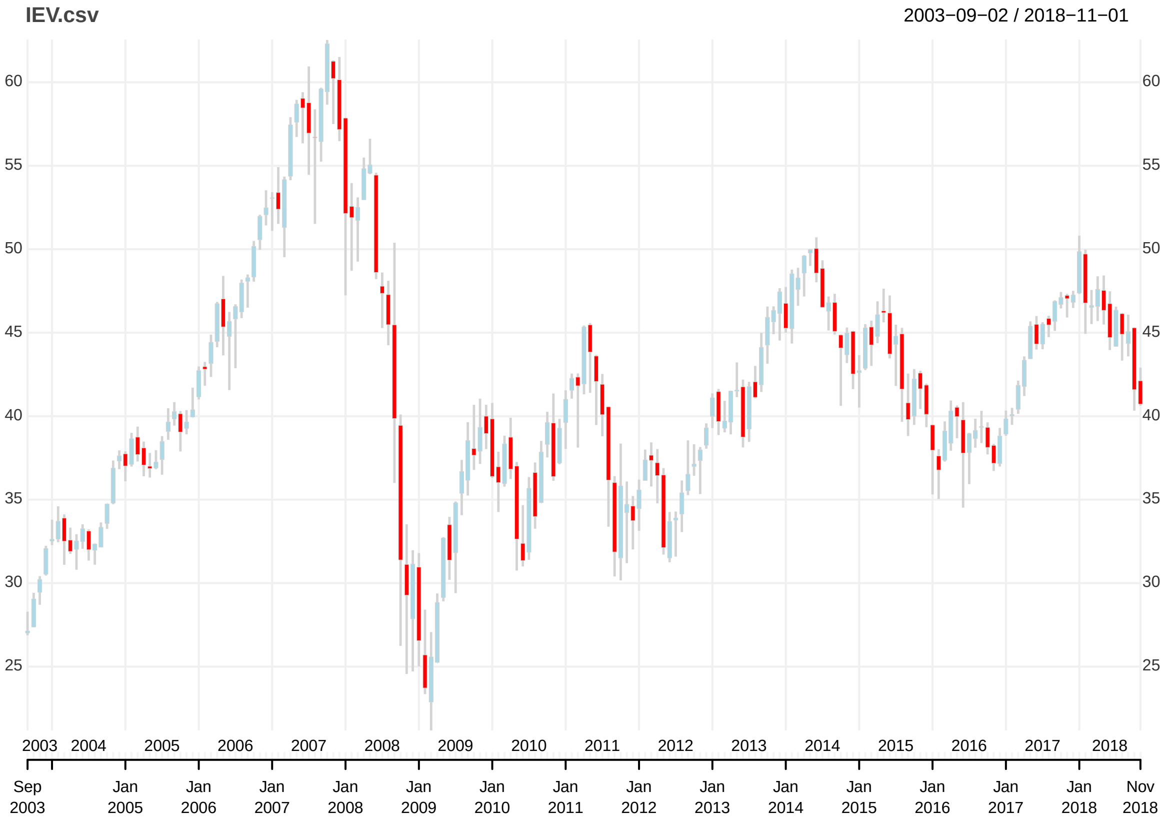
3.3.3 Step 3: Visualize the original data

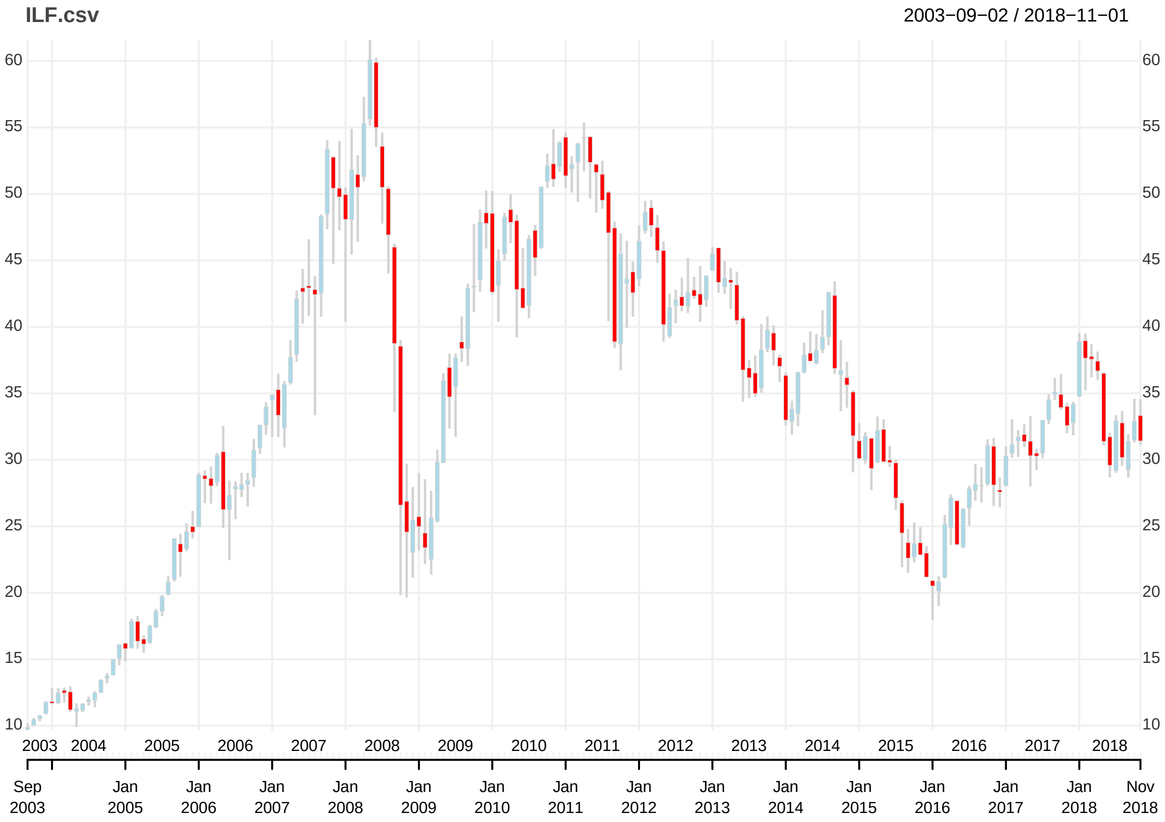
Below are visualizations of the original instruments which are traded. This plots can be used to get a first impression of the long term trend of the instrument and to see which general behaviour of the price movement can be observed.

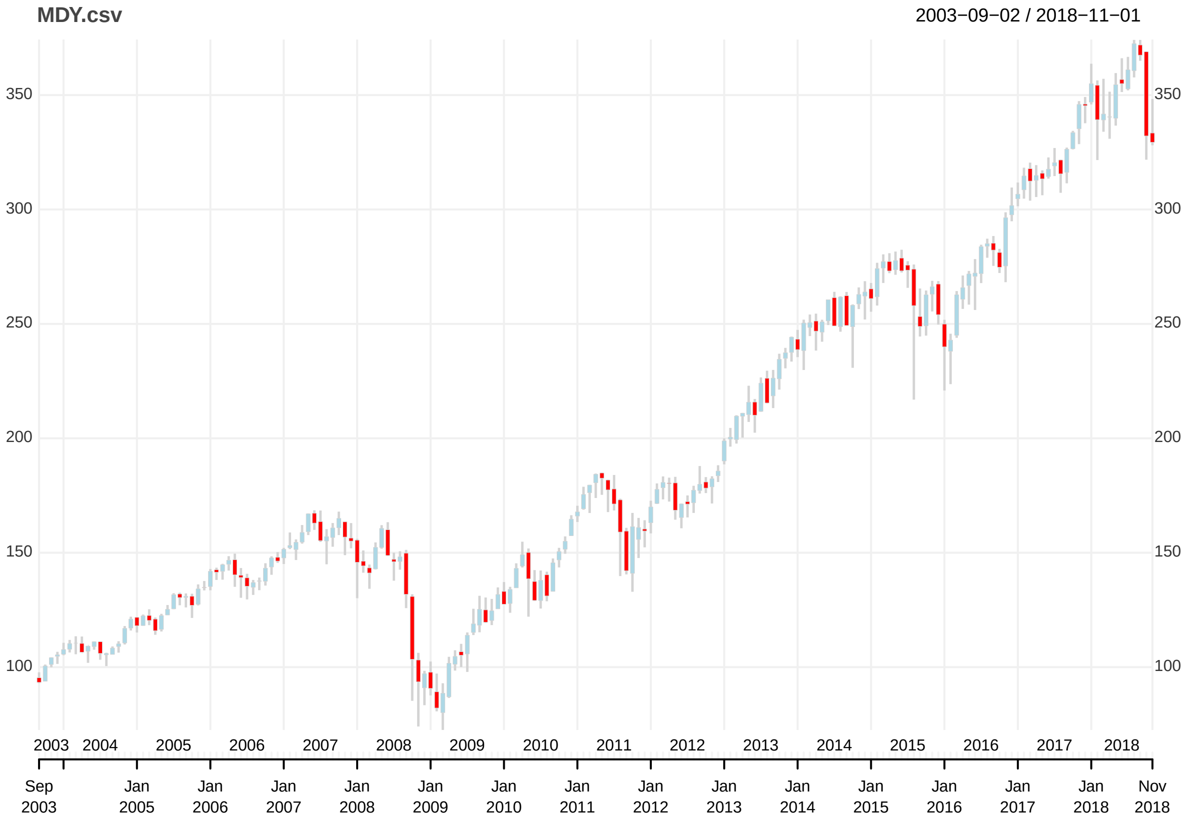
```
for (symbol in symbollist){
  chart <- get(symbol)
   if (nrow(getTxns(portfolioname,Symbol=symbol)) > 1)
    {
      print(chart_Series(x=chart,name=symbol,
                         theme=myTheme,
                         subset=daterange))
    }

}
```


EEM.csv
2003−09−02 / 2018−11−01
55
50
45
40
35
30
25
20
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018


EPP.csv
2003−09−02 / 2018−11−01
60
55
50
45
40
35
30
25
20
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018


IEV.csv
2003−09−02 / 2018−11−01
60
55
50
45
40
35
30
25
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018

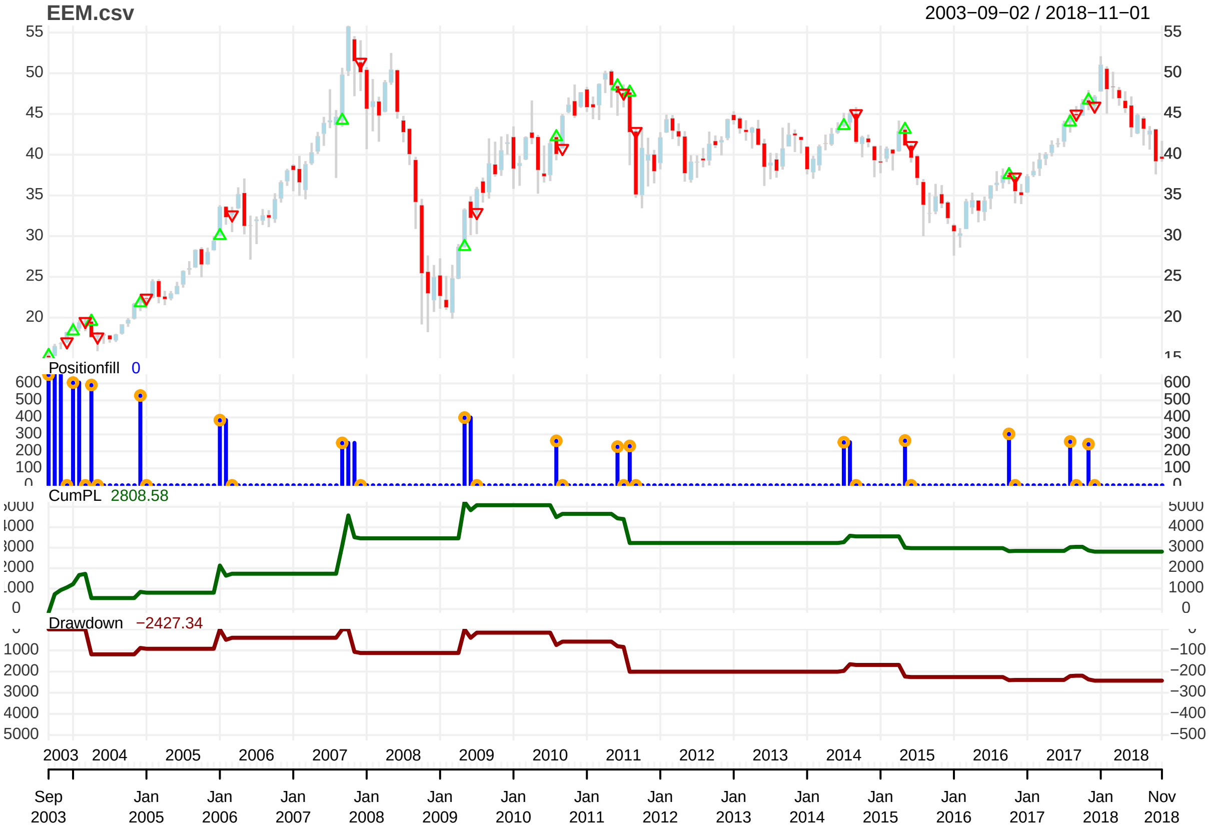

ILF.csv
2003−09−02 / 2018−11−01
60
55
50
45
40
35
30
25
20
15
10
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018

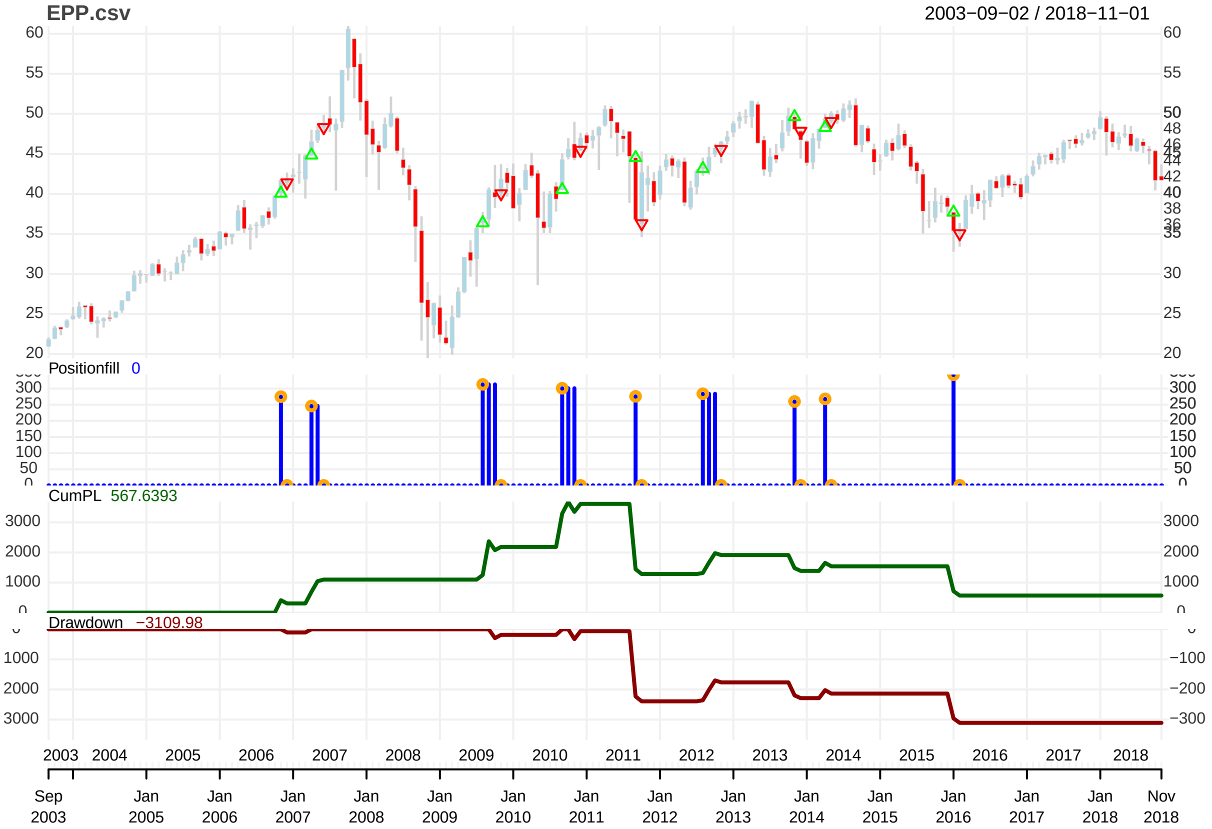


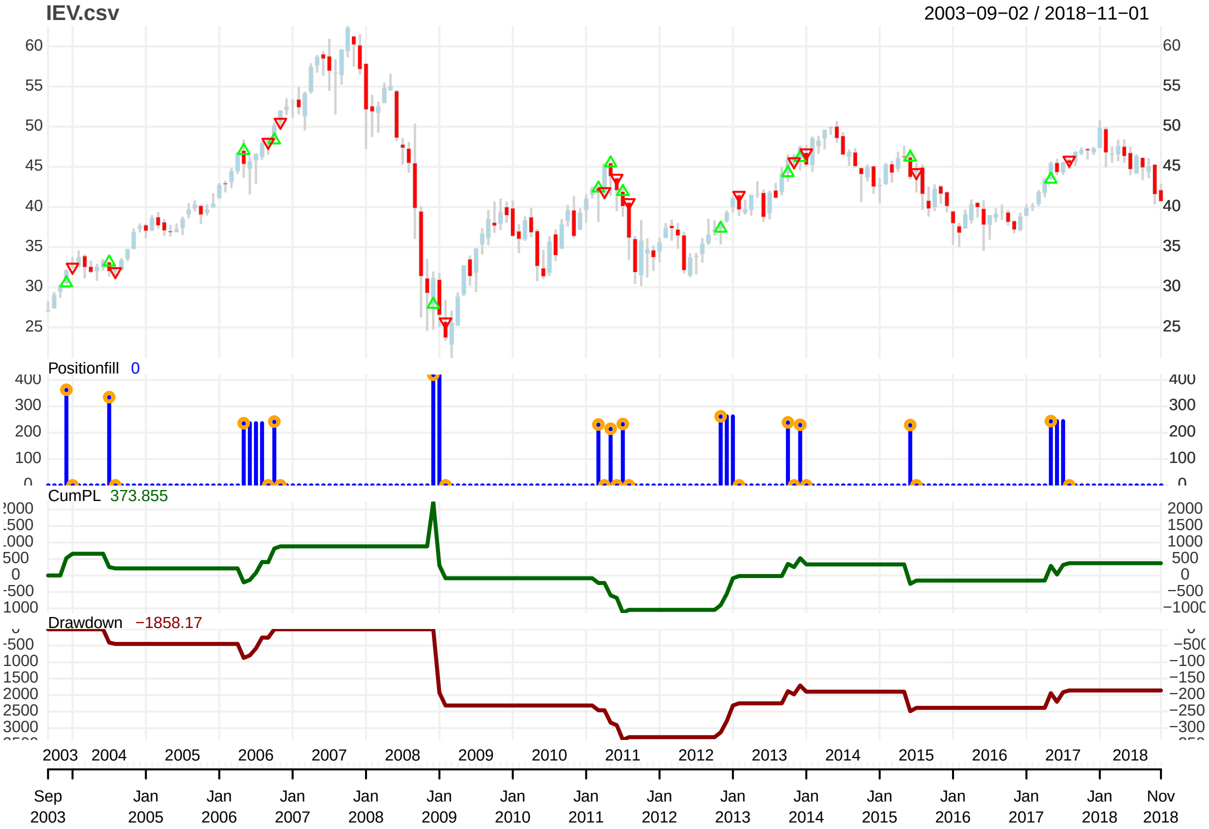
3.3.3.1 Step 4: Visualize all transactions

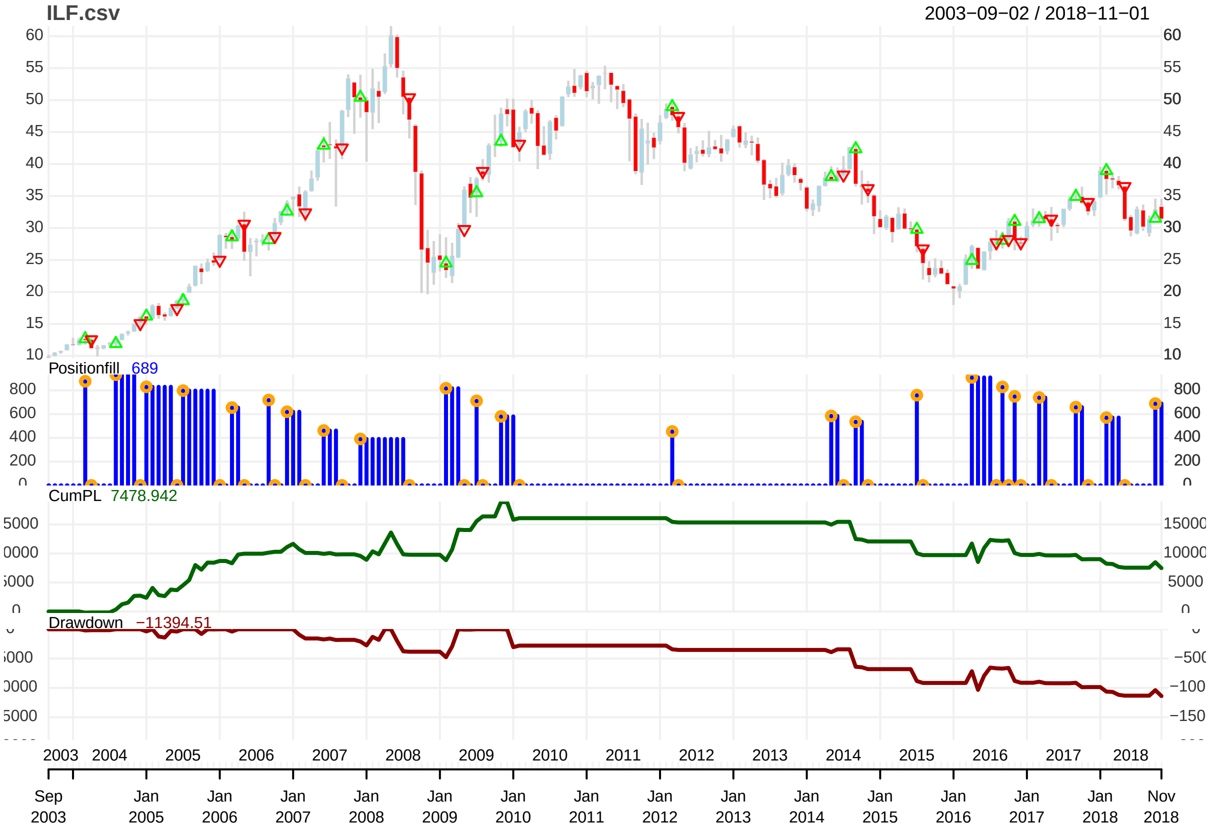
The following graphs show the combined view of the performance of the GRS trading system. It visualizes the trades (buy-transactions are visualized in green and sell-transactions are visualized in red). Moreover, the size of the blue squares indicates the size of the position (height) and the holding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

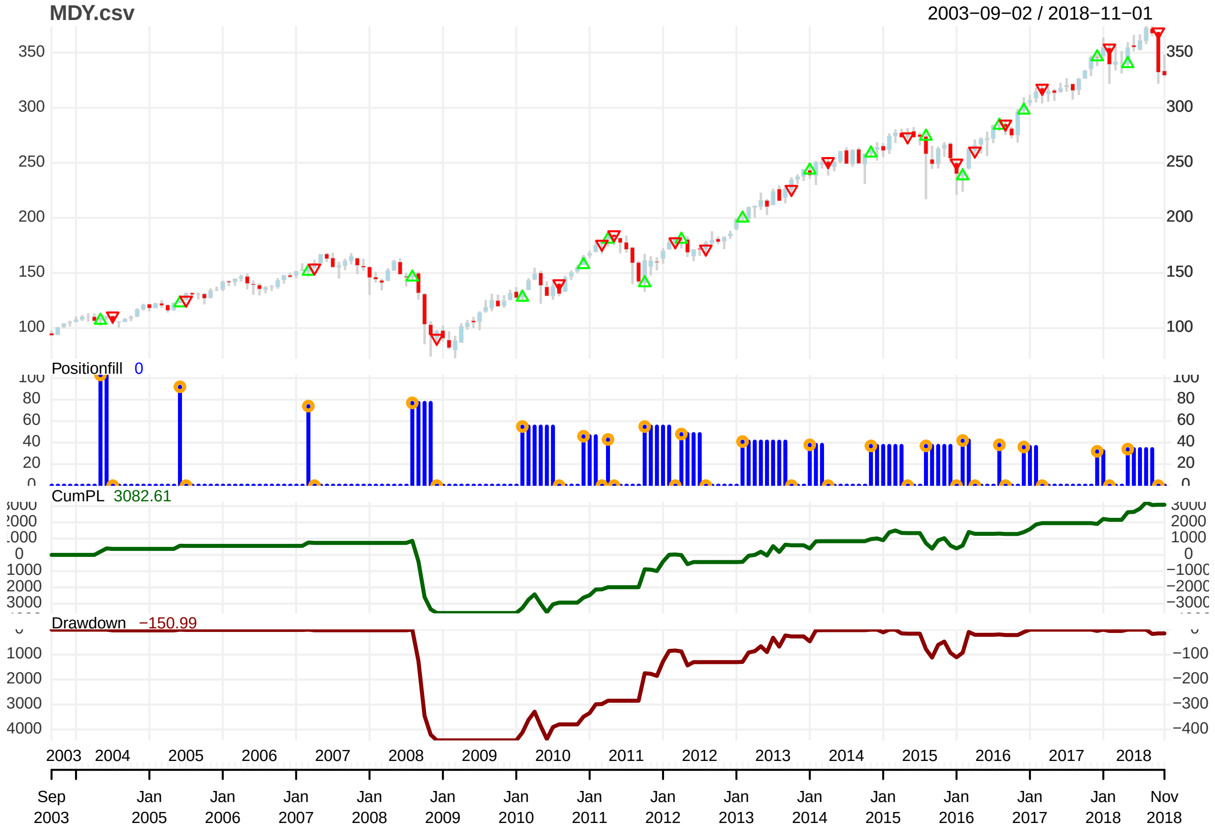
```r
par(mfrow=c(1,1))
# plot graph with indicators for transactions
for (symbol in symbollist)
{
    if (nrow(getTxns(portfolioname,Symbol=symbol)) > 1)
    {
        symboldata <- get(symbol)
        chart.Posn(portfolioname,
                   Symbol=symbol,
                   type='candlesticks',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


EEM.csv
2003−09−02 / 2018−11−01
55
50
45
40
35
30
25
20
15
Positionfill 0
600
500
400
300
200
100
0
CumPL 2808.58
5000
4000
3000
2000
1000
0
Drawdown −2427.34
0
−100
−200
−300
−400
−500
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018


EPP.csv
2003−09−02 / 2018−11−01
60
55
50
45
40
35
30
25
20
Positionfill 0
300
250
200
150
100
50
0
CumPL 567.6393
3000
2000
1000
0
Drawdown −3109.98
1000
2000
3000
−100
−200
−300
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018


IEV.csv
2003−09−02 / 2018−11−01
60
55
50
45
40
35
30
25
Positionfill 0
400
300
200
100
0
CumPL 373.855
2000
1500
1000
500
0
−500
−1000
Drawdown −1858.17
0
−500
−1000
−1500
−2000
−2500
−3000
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018


ILF.csv
2003−09−02 / 2018−11−01
60
55
50
45
40
35
30
25
20
15
10
Positionfill 689
800
600
400
200
0
CumPL 7478.942
5000
0000
5000
0
15000
10000
5000
0
Drawdown −11394.51
5000
0000
5000
−500
−100
−150
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018



3.3.3.2 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 performes. 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.

```r
library(PerformanceAnalytics)
# Get returns for all instruments in the portfoilio
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",
```

```r
                                           "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
```

```r
tab.perf
```

	Ret	EEM	EPP	IEV	ILF	MDY
Cumulative Return	1.4311626	0.2285680	0.0067913	-0.0065177	0.1485727	0.2594299
Annualized Return	0.0596484	0.0135898	0.0004439	-0.0004287	0.0091247	0.0152402
Annualized Sharp Ratio	0.4096519	0.1345520	0.0056614	-0.0057701	0.0325740	0.1552906
Calmar Ratio	0.1769151	0.0601695	0.0015183	-0.0014177	0.0119390	0.0397280

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

```
## Risk Metrics
```

```r
tab.risk
```

	Ret	EEM	EPP	IEV	ILF	MDY
Annualized StdDev	0.1456075	0.1010005	0.0784119	0.0742959	0.2801212	0.0981396
Max Drawdown	0.3371584	0.2258592	0.2923854	0.3023776	0.7642705	0.3836126
Value-at-Risk	-0.0695126	-0.0172231	-0.0089518	-0.0140131	-0.0943069	-0.0326733
Conditional VaR	-0.0878293	-0.0602668	-0.0467302	-0.0515536	-0.2036066	-0.0762445

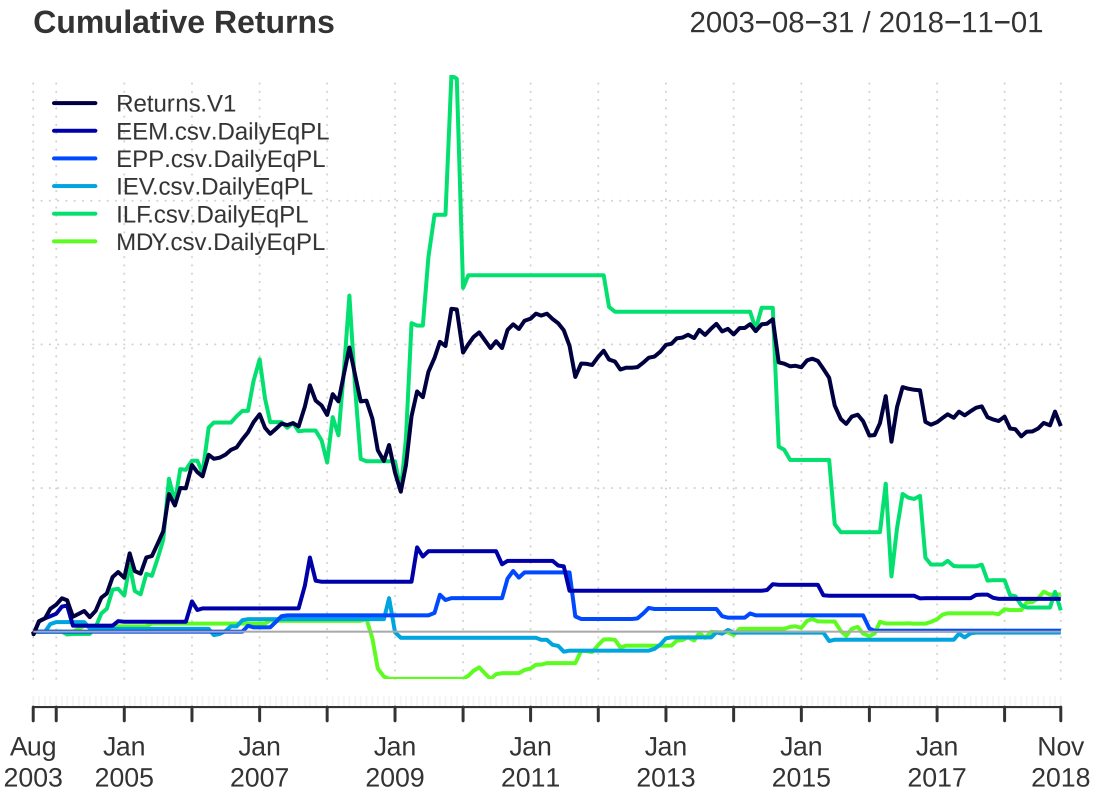
```r
# present performance metrics for the entire strategy
somestats <- data.frame(rownames(tab.perf),
                        tab.perf[,1],
                        rownames(tab.risk),
                        tab.risk[,1])
colnames(somestats) <- c("Performance Metric",
                         "Performance Value",
                         "Risk Metric",
                         "Risk Value")
somestats
```

Performance Metric	Performance Value	Risk Metric	Risk Value
Cumulative Return	1.4311626	Annualized StdDev	0.1456075
Annualized Return	0.0596484	Max Drawdown	0.3371584
Annualized Sharp Ratio	0.4096519	Value-at-Risk	-0.0695126
Calmar Ratio	0.1769151	Conditional VaR	-0.0878293

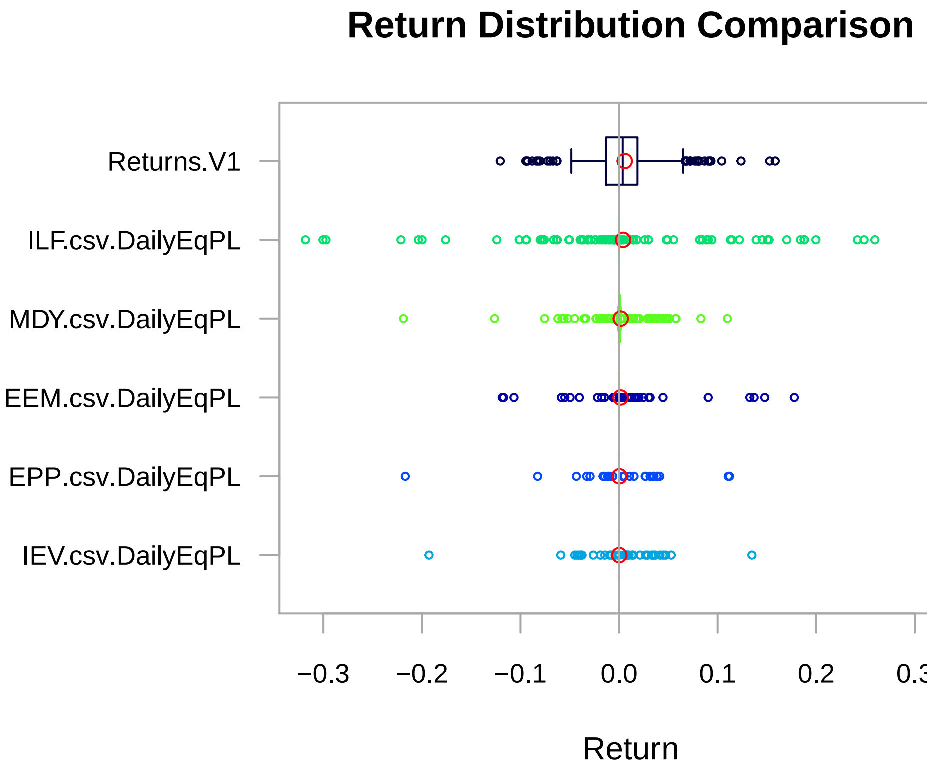
3.3.3.3 Step 6: Visualize portfolio statistics

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.

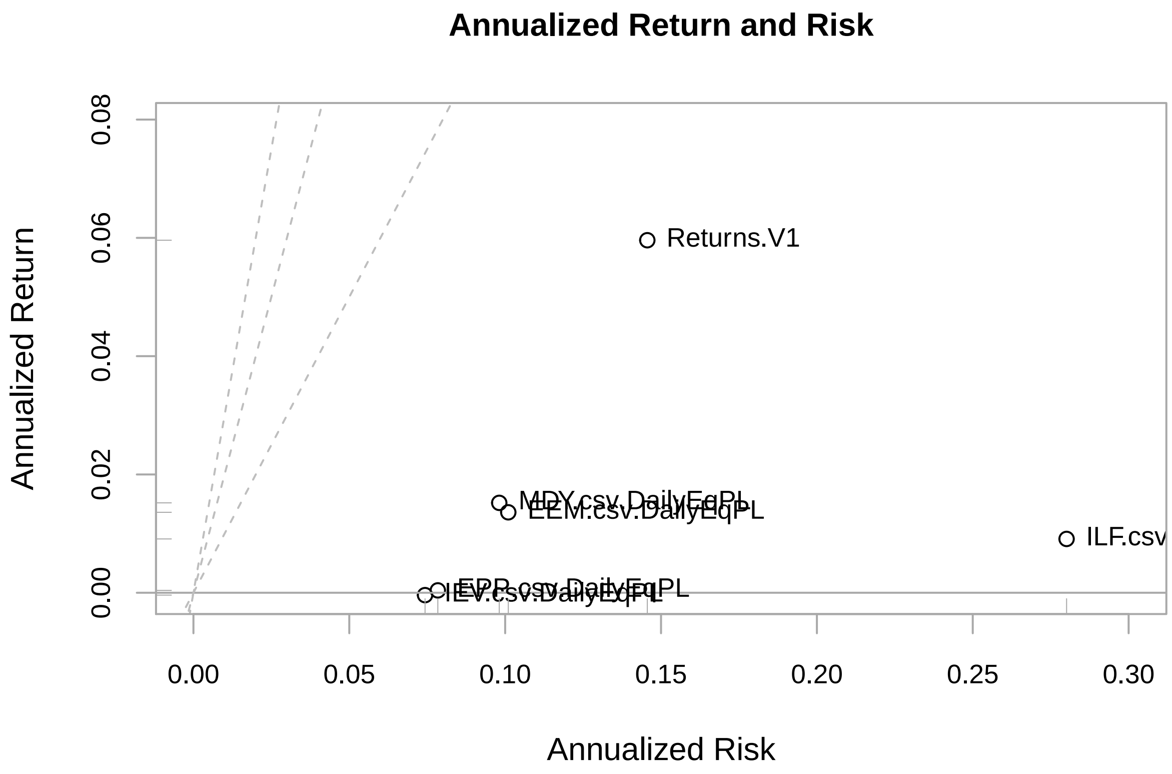
```
par(mfrow=c(1,1))
chart.CumReturns(rets.v1.comb,
                 colorset=rich10equal,
                 legend.loc="topleft",
                 main="Cumulative Returns")
```



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



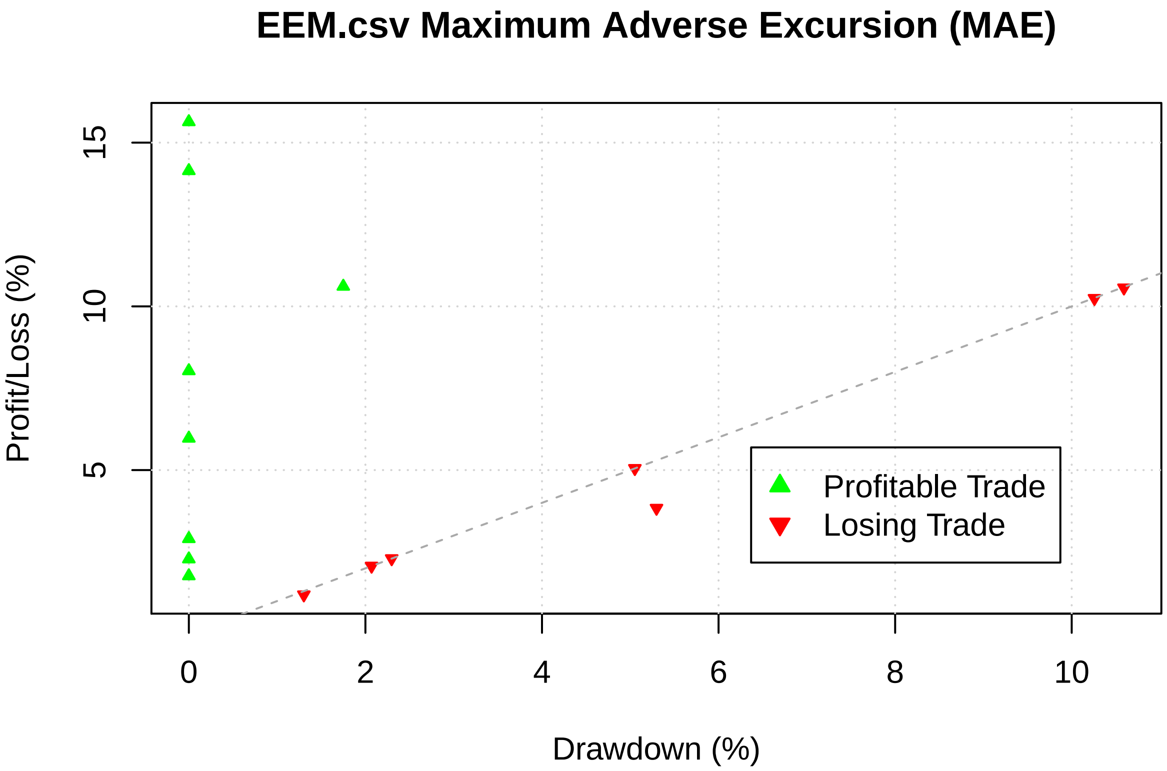
```
chart.RiskReturnScatter(rets.v1.comb)
```

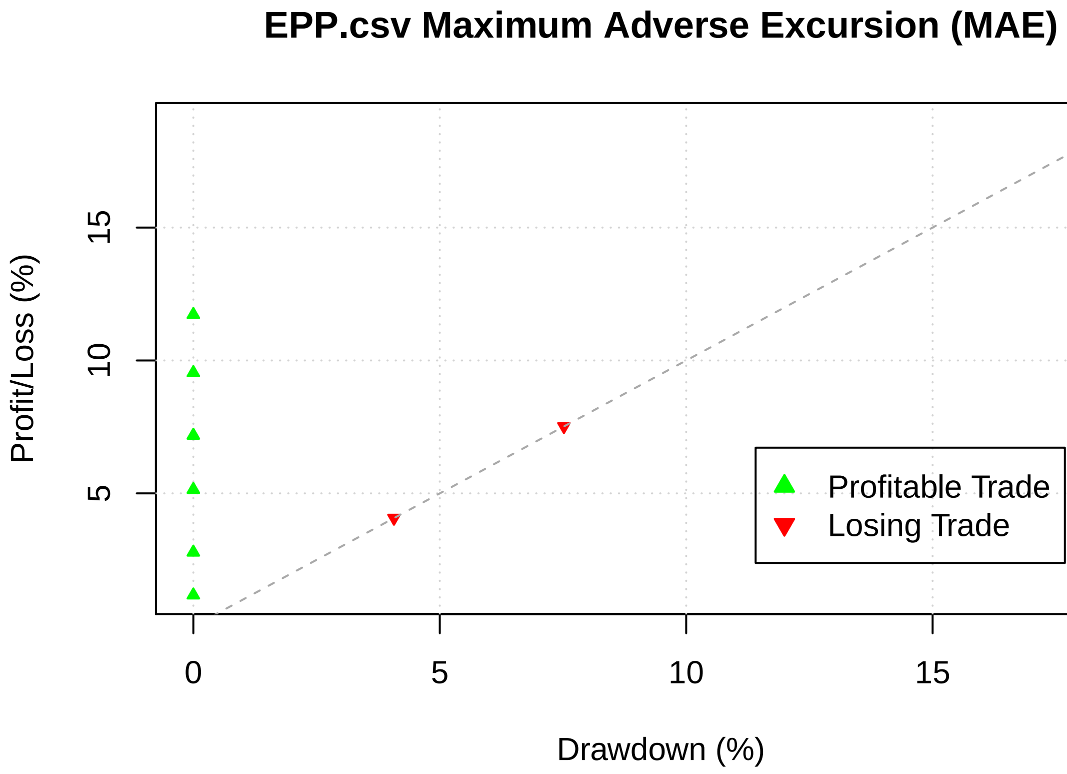


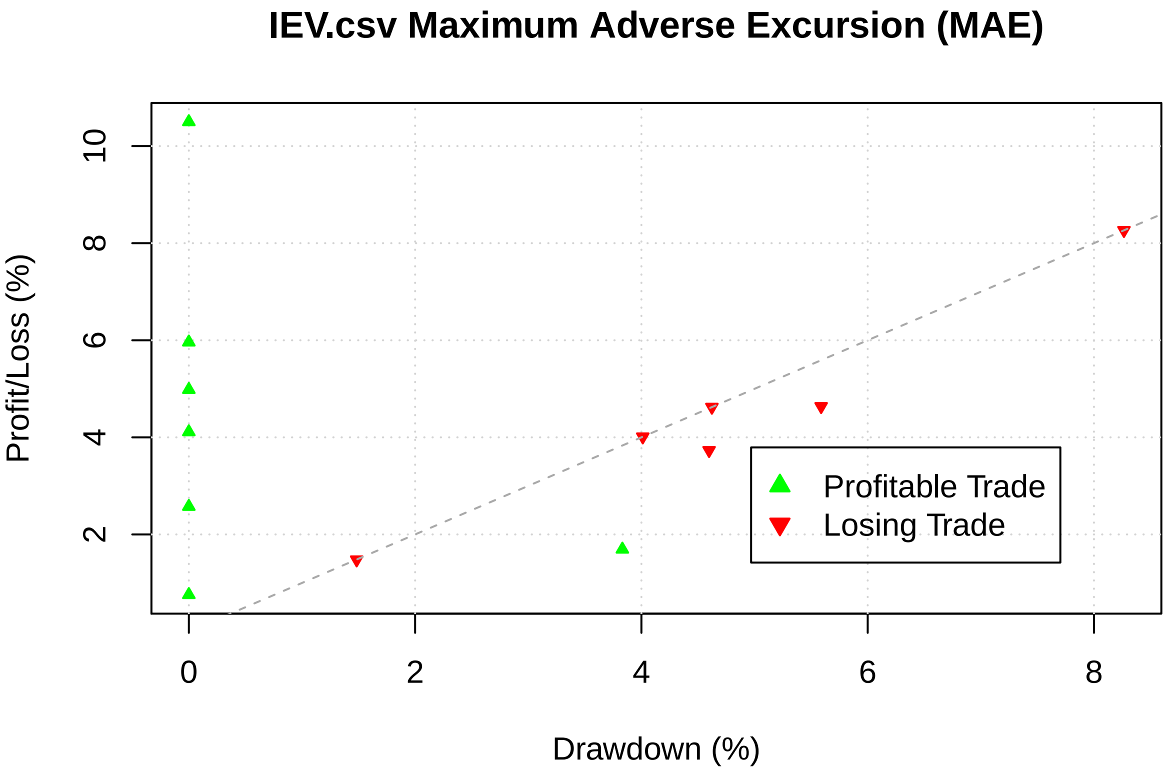
3.3.3.4 Step 7: Could we have performed better or even worse? MAE / MFE

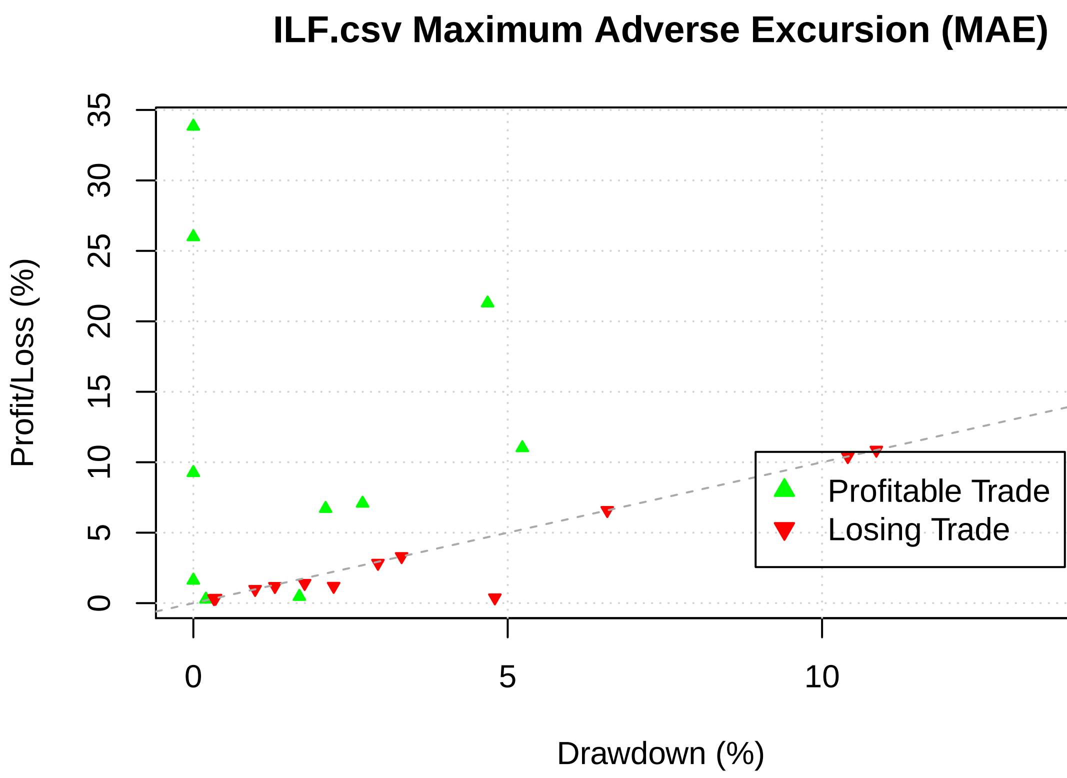
The efficiency of a strategy is also dependent on the question whether we could have performed better or worse. This can be shown with the maximum adverse excursion and the maximum favorable excursion. The Maximum Adverse excursion shows how much we could have lost at most during the trade, while the 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.

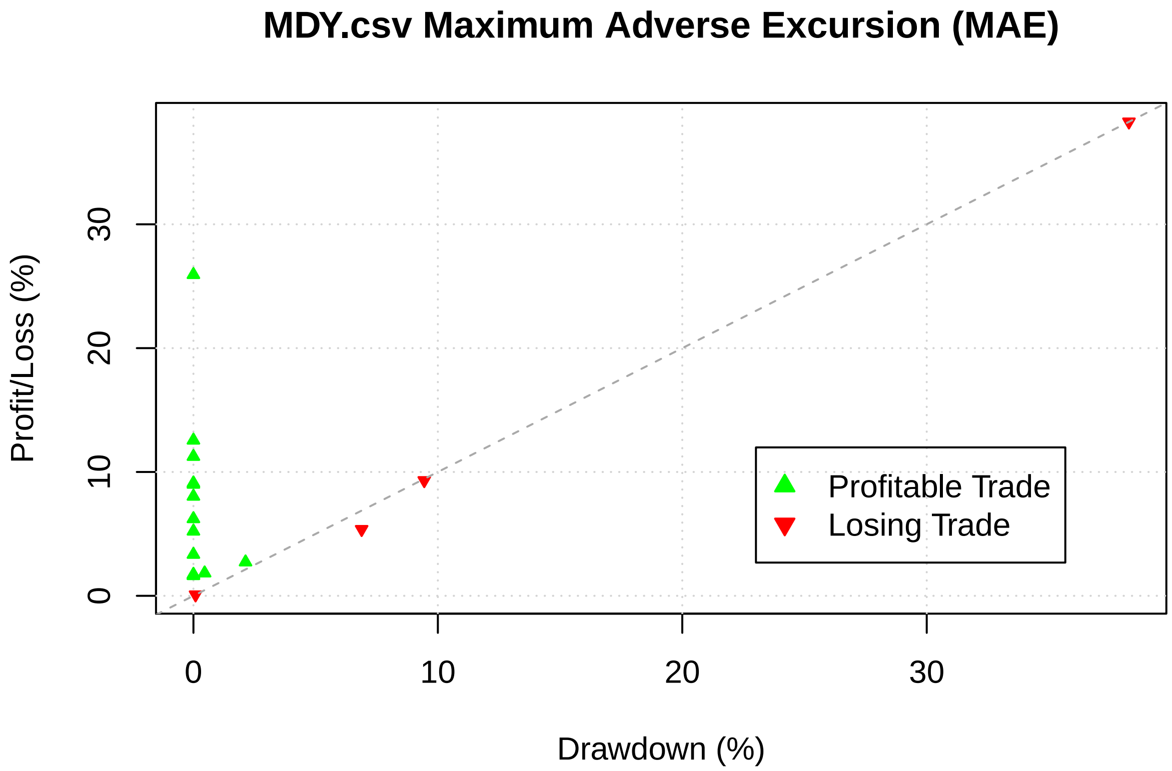
```r
for (instrument in symbollist){
  chart.ME(portfolioname, instrument, type="MAE", scale="percent")
}
```


EEM.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
15
10
5
Drawdown (%)
0
2
4
6
8
10
▲ Profitable Trade
▼ Losing Trade


EPP.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
5
10
15
0
5
10
15
Drawdown (%)
▲ Profitable Trade
▼ Losing Trade


IEV.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
10
8
6
4
2
0
2
4
6
8
Drawdown (%)
▲ Profitable Trade
▼ Losing Trade


ILF.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
0
5
10
15
20
25
30
35
Drawdown (%)
0
5
10
15
▲ Profitable Trade
▼ Losing Trade



3.3.3.5 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
```

3.3.3.5.1 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
## -13.1819  -1.5270  -0.3787   1.0936  14.6738
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.005985   0.262019   0.023    0.982
## mkt.rf      0.626846   0.065683   9.543   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.442 on 177 degrees of freedom
## Multiple R-squared:  0.3397, Adjusted R-squared:  0.336
## F-statistic: 91.08 on 1 and 177 DF,  p-value: < 2.2e-16
```

3.3.3.5.2 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
## -12.9402  -1.5833  -0.3621   1.1228  14.6474
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01417    0.26282   0.054    0.957
## mkt.rf       0.59547    0.07241   8.224  4.3e-14 ***
## smb          0.07154    0.12056   0.593    0.554
## hml          0.09746    0.10863   0.897    0.371
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.45 on 175 degrees of freedom
## Multiple R-squared:  0.3443, Adjusted R-squared:  0.333
## F-statistic: 30.63 on 3 and 175 DF,  p-value: 5.771e-16
```

The results of the strategy's risk analysis indicate that no excess returns are generated. The intercept of the regression models (CAPM as well as Fama-French model) is negative and not significant. This means that risk adjusted and compared to the market we cannot generate an alpha (excess returns). However, the beta-coefficient for the market risk free return is significant and with about 0.68 below 0. Thus, the strategy is not significantly market sensitiv.

3.4 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.

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

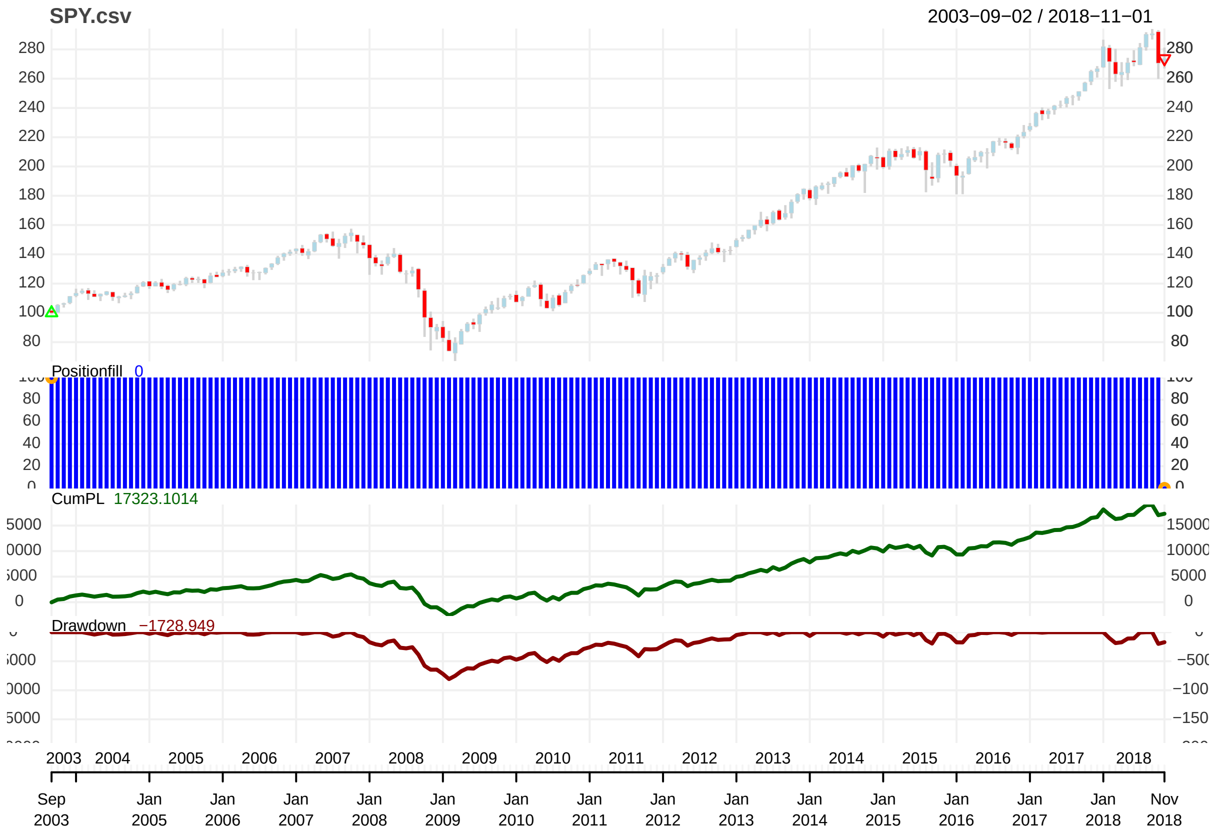
```
BuyHoldDirectory = "~/Desktop/R/GRS/BH/"
BuyHoldInstrument = "SPY.csv"

# 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="2003-08-31::2018")

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))
indexFormat(x) <- '%Y-%m'
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="txnc")

# 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="txnc")

# update portfolio and account
updatePortf(Portfolio="buyhold")
```

```
updateAcct(name="buyhold")
updateEndEq(Account="buyhold")
```

3.4.2 Step 2: Visualize the Buy and Hold strategy

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



3.4.3 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

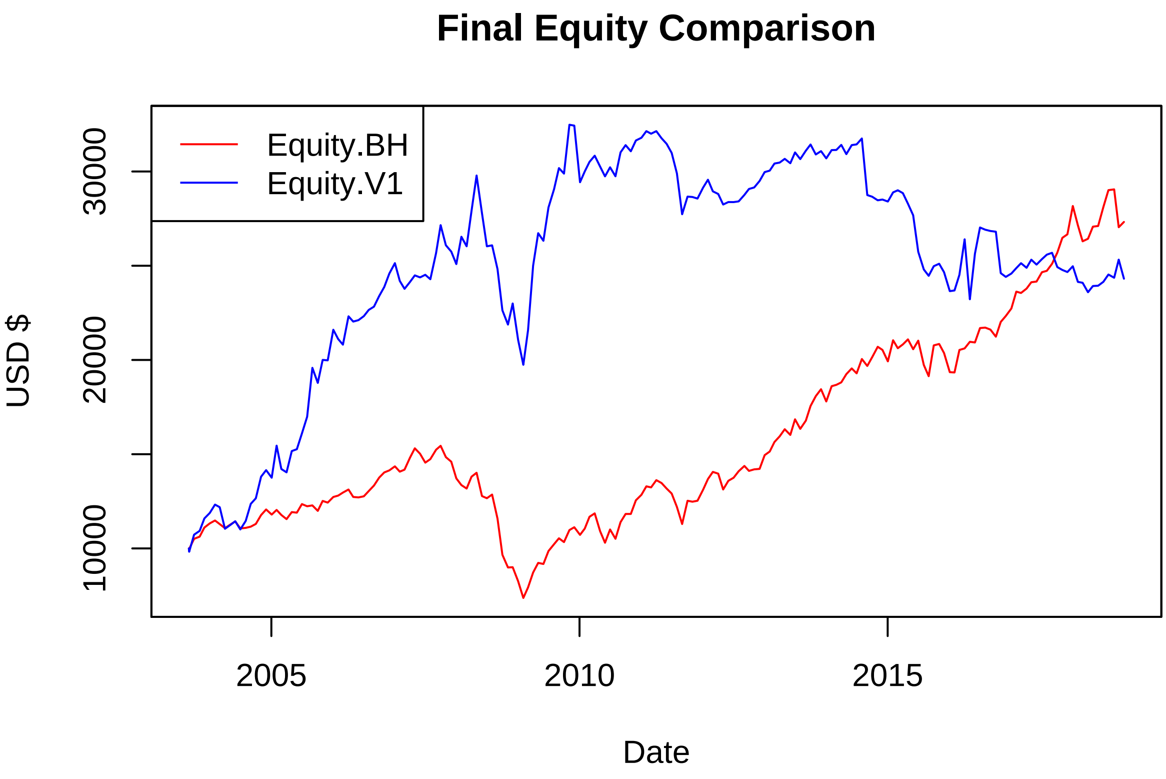
3.4.3.1 Generate equity and return

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

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

rets.bh.v1 <- cbind(rets.bh, rets.v1)
```

3.4.3.2 Visually compare Equity curves

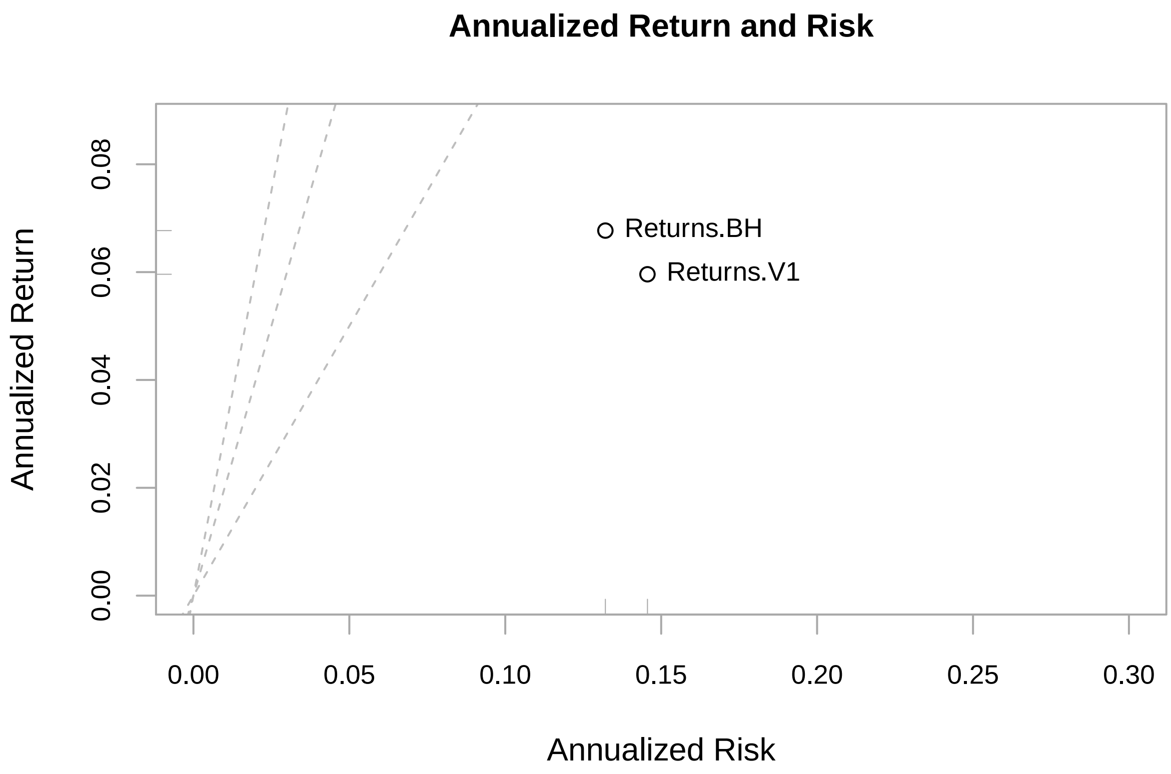
```
plot.zoo(equity.bh.v1,
         plot.type="single",
         col=c('red','blue'),
         main="Final Equity Comparison",
         xlab="Date",
         ylab="USD $")
legend(x="topleft",
       legend=colnames(equity.bh.v1),
       col=c("red","blue"),
       lty=1)
```



3.4.3.3 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 we cannot outperform the buy and hold strategy.

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



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

	Returns.BH	Returns.V1
Annualized Return	0.0677	0.0596
Annualized Std Dev	0.1321	0.1456
Annualized Sharpe (Rf=0%)	0.5128	0.4097

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

	Returns V1
Annualized Return	0.0596
Annualized Std Dev	0.1456
Annualized Sharpe (Rf=0%)	0.4097

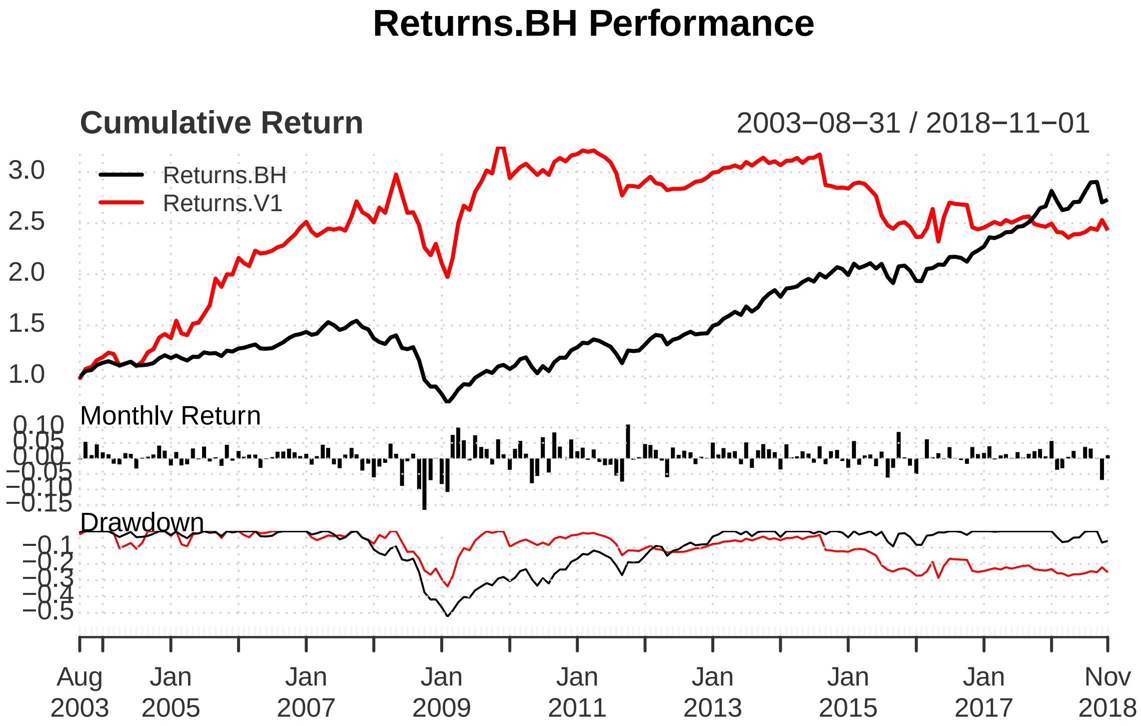
```r
table.Stats(rets.bh.v1)
```

	Returns.BH	Returns.V1
Observations	184.0000	184.0000
NAs	0.0000	0.0000
Minimum	-0.1655	-0.1205
Quartile 1	-0.0166	-0.0131
Median	0.0106	0.0037
Arithmetic Mean	0.0062	0.0057
Geometric Mean	0.0055	0.0048
Quartile 3	0.0292	0.0186
Maximum	0.1093	0.1584
SE Mean	0.0028	0.0031
LCL Mean (0.95)	0.0007	-0.0004
UCL Mean (0.95)	0.0118	0.0118
Variance	0.0015	0.0018
Stdev	0.0381	0.0420
Skewness	-0.7724	0.3769
Kurtosis	2.2969	1.9048

3.4.3.4 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. Hoewever, the returns could not be sustained, thus, the GRS performed over the long run not as good as a simple buy and hold strategy.

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



3.5 Conclusion

The Global Rotation Strategy in its plain generates 7% annulaized return while the compared buy and hold strategy generates about 6%. The Sharp Ratio for the strategy is 0.48 compared to 0.43 for the buy and hold strategy. As we saw, the risk analysis has shown that we cannot generate excess returns with this strategy compared to the market. Especially after 2009 the strategy does not generate enough returns to outperform the buy and hols strategy. In fact, after 2009 the returns of the strategy are not significantly positive. The following chapters describe improvement options for the strategy, in order to increase the returns and the performance, especially for the time horizon after the GFC.

4 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.

4.1 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="USD")
initOrders(portfolio=portfolioname.alt,initDate=initdate)
strategy(strategyname.alt,store=TRUE)
strat <- getStrategy(strategyname.alt)
```

4.2 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",
                        sigval=TRUE,
                        orderqty=100,
                        ordertype="market",
                        orderside="long",
                        replace=FALSE,
                        prefer="Open",
                        osFUN='osInvestAll',
                        orderset="ocolong",
                        TxnFees="txnc"),
         type="enter",
         label="LE",
         path.dep=TRUE)

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

4.3 Step 3: Apply the strategy

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

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

4.4 Analysis of improved strategy GRS.V2

4.4.1 Step 1: Calculate equity and return of the improved strategy

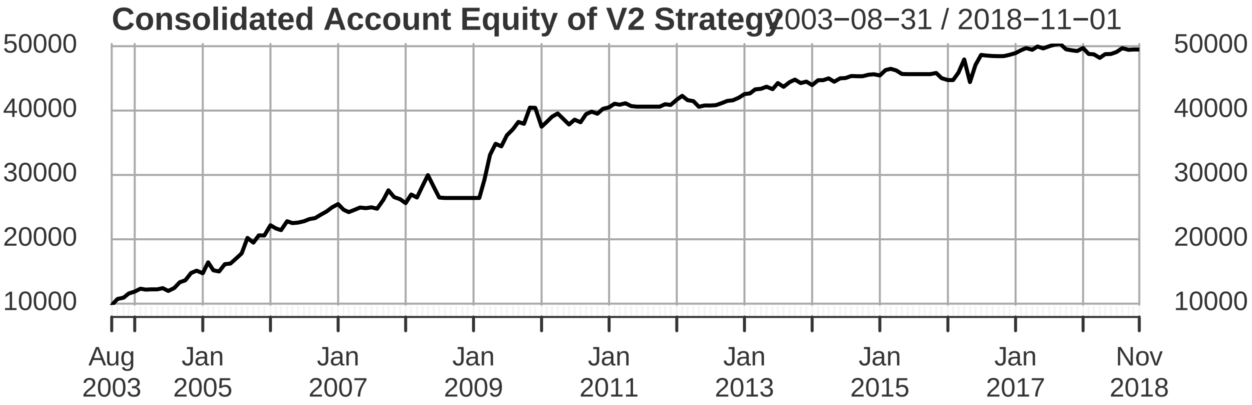
```
acct.alt <- getAccount(accountname.alt)

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

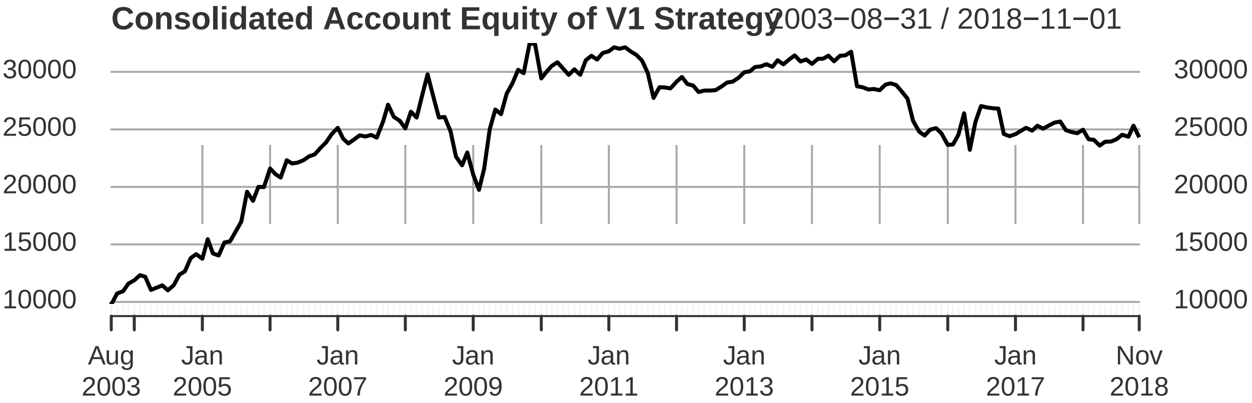
```r
rets.v2 <- monthlyReturn(equity.v2, type = "arithmetic")
colnames(rets.v2) <- c("Return V2")
```

4.4.2 Step 2: Visualize Equity of GRS.V2

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. This makes sense, as we only enter a trade, when the past 3 months generated positive returns, we avoid phases in which the entire world market (2008-2009) is in a negative trend. As a result, we can see that we are not invested during this time at all.

```r
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")
```



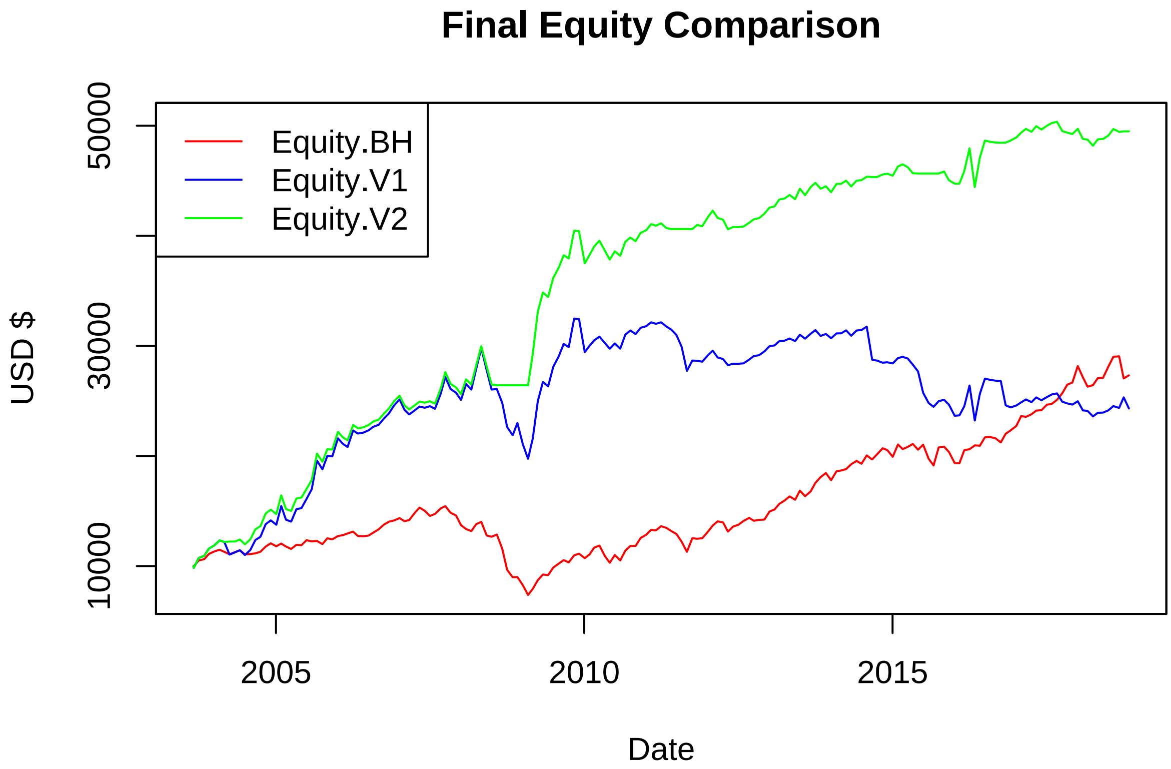


```r
par(mfrow=c(1,1))
```

4.4.3 Step 3: Visualize Equity comparison

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

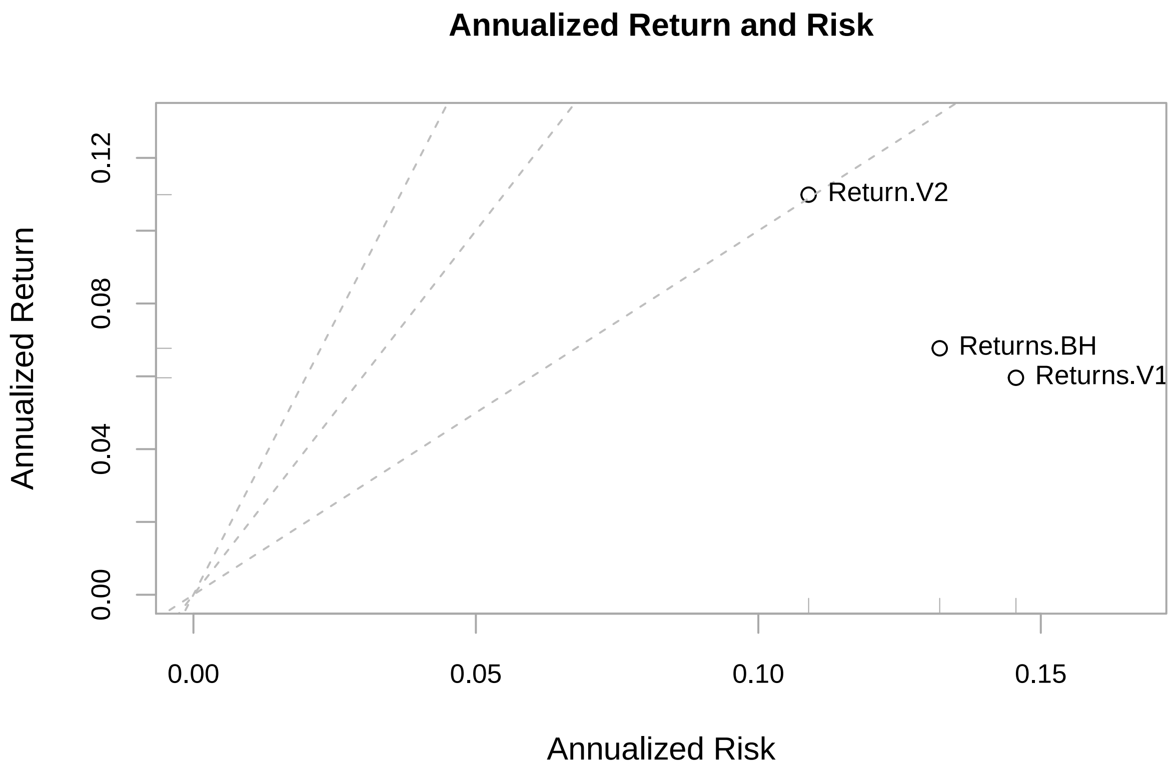
```
plot.zoo(equity.bh.v1.v2,plot.type="single",col=c('red','blue','green'),main="Final Equi
legend(x="topleft",legend=colnames(equity.bh.v1.v2),col=c("red","blue","green"),lty=1) #
```



4.4.4 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)
chart.RiskReturnScatter(rets.bh.v1.v2)
```



4.4.4.1 General Strategy Statistics

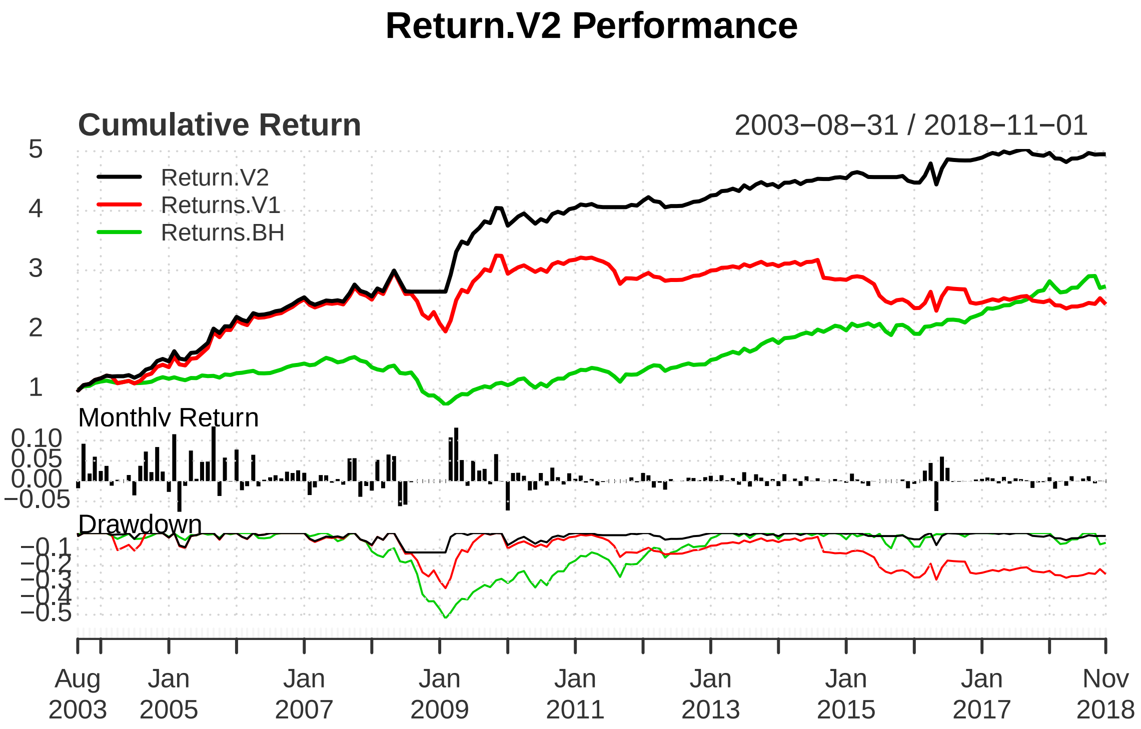
```
tab.perf <- table.Arbitrary(rets.bh.v1.v2, metrics=c("Return.cumulative",
                                             "Return.annualized",
                                             "SharpeRatio.annualized",
                                             "CalmarRatio"),
                            metricsNames=c("Cumulative Return",
                                            "Annualized Return",
                                            "Annualized Sharp Ratio",
                                            "Calmar Ratio"))
tab.risk <- table.Arbitrary(rets.bh.v1.v2, metrics=c("StdDev.annualized",
                                             "maxDrawdown",
                                             "VaR",
                                             "ES"),
                            metricsNames=c("Annualized StdDev",
                                            "Max Drawdown",
                                            "Value-at-Risk",
                                            "Conditional VaR"),
                            method=c("historical"))

somestats <- data.frame(rownames(tab.perf),tab.perf[,1], rownames(tab.risk),tab.risk[,1
colnames(somestats) <- c("Performance Metric", "Performance Value", "Risk Metric", "Risk
somestats
```

Performance Metric	Performance Value	Risk Metric	Risk Value
Cumulative Return	3.9484875	Annualized StdDev	0.1088522
Annualized Return	0.1099200	Max Drawdown	0.1184020
Annualized Sharp Ratio	1.0098101	Value-at-Risk	-0.0260376
Calmar Ratio	0.9283629	Conditional VaR	-0.0511959

4.4.5 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. Moreover, we can see that the drawdown decreases drastically as we do not trade during the GFC.

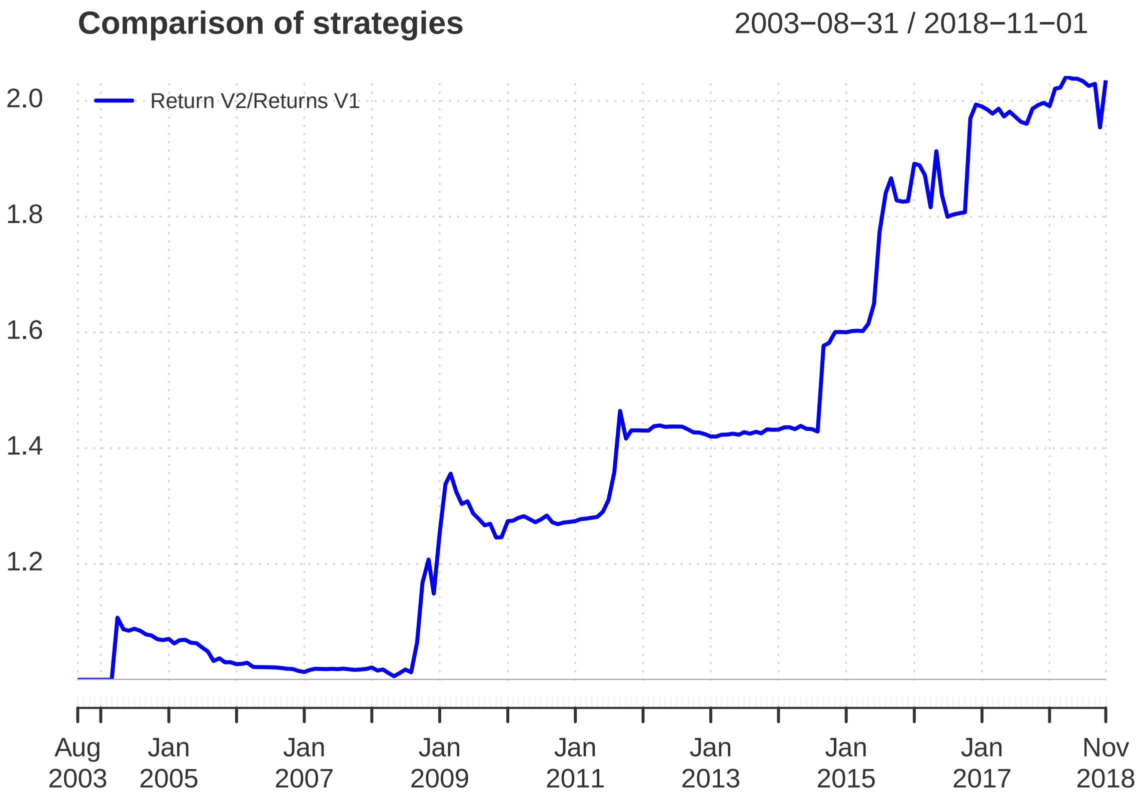
```
tab.perf
tab.risk
charts.PerformanceSummary(rets.bh.v1.v2,geometric=T,wealth.index=TRUE)
```



4.4.6 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")
```



4.4.7 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
```

4.4.7.1 Perform a CAPM Regression

```r
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
## -0.12218 -0.09151 -0.06993  0.04444  0.35041
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.099385   0.010300   9.649   <2e-16 ***
## mkt.rf      -0.002808   0.002582  -1.087    0.278
## ---
## 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:  0.006635,	Adjusted R-squared:  0.001023
## F-statistic: 1.182 on 1 and 177 DF,  p-value: 0.2784
```

4.4.7.2 Perform a Fama-French 3 factor Regression

```r
# 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
## -0.12668 -0.09158 -0.07133  0.04209  0.35214
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.099722   0.010348   9.637   <2e-16 ***
## mkt.rf      -0.003104   0.002851  -1.089    0.278
## smb         -0.001342   0.004747  -0.283    0.778
## hml          0.003299   0.004277   0.771    0.442
## ---
## 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:  0.01034,	Adjusted R-squared:  -0.006623
## F-statistic: 0.6096 on 3 and 175 DF,  p-value: 0.6096
```

The improved strategy's risk analysis shows that we can generate excess risk adjusted returns. The intercept of the regression is in both tested models (CAPM and Fama-French Model) significant and positive with about 0.1. This leads to a risk-adjusted outperformance of about 0.1*12 = 1.2% per year.

5 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 accoring 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=sta
initOrders(portfolio=portfolioname.alt.v3,initDate=initdate)
strategy(strategyname.alt.v3,store=TRUE)
strat <- getStrategy(strategyname.alt.v3)
```

5.1 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,
```

```r
                          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="txnc"),
         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="txnc"),
         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="txnc"),
         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="txnc"),
         type="exit",
         path.dep=TRUE,
         label="ShortExit")
```

5.2 Apply the strategy

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

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

5.3 Analysis of the improved (V3) Strategy

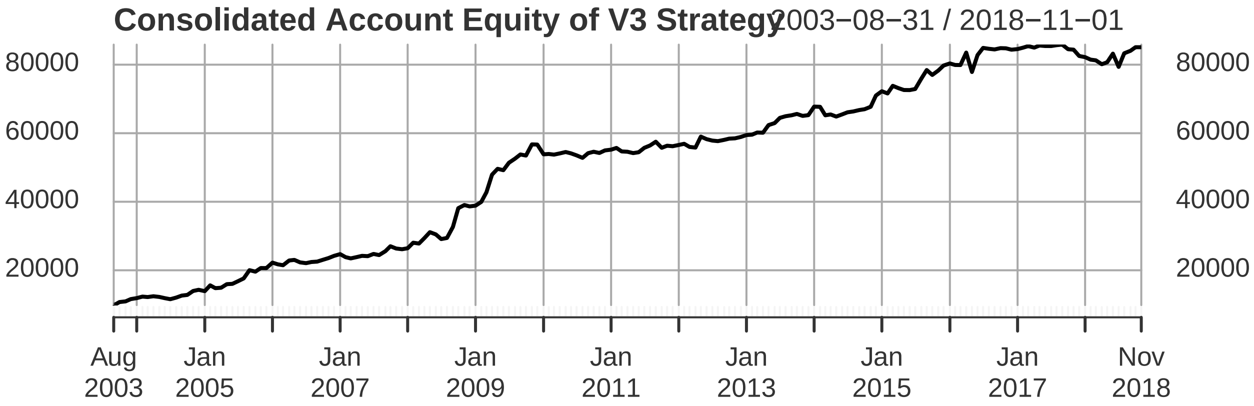
5.3.1 Retrieve Equity of GRS.V3 strategy

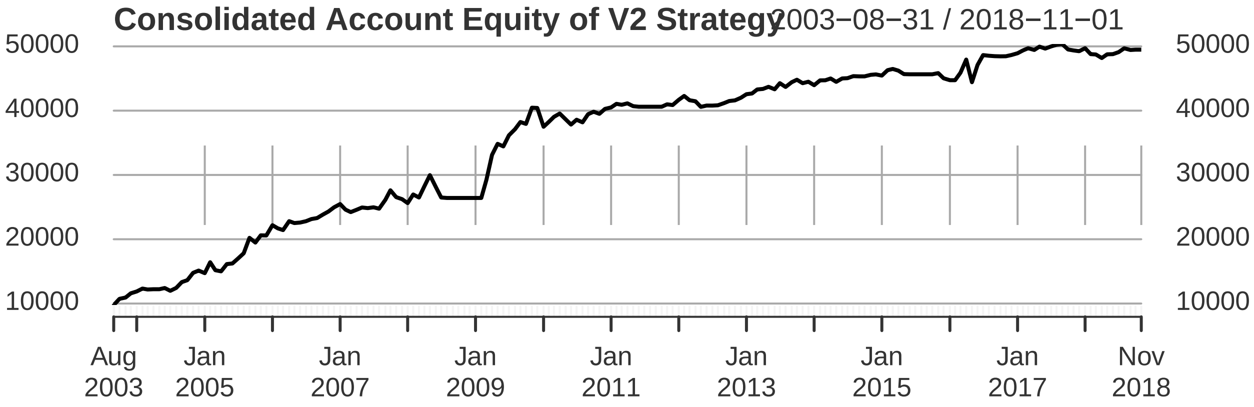
```
acct.alt.v3 <- getAccount(accountname.alt.v3)

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

5.3.2 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. Moreover, we are able to generate returns during phases where the V2 strategy simply did not perform any trades. By this we generate returns during strong and weak market phases.

```
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")
```



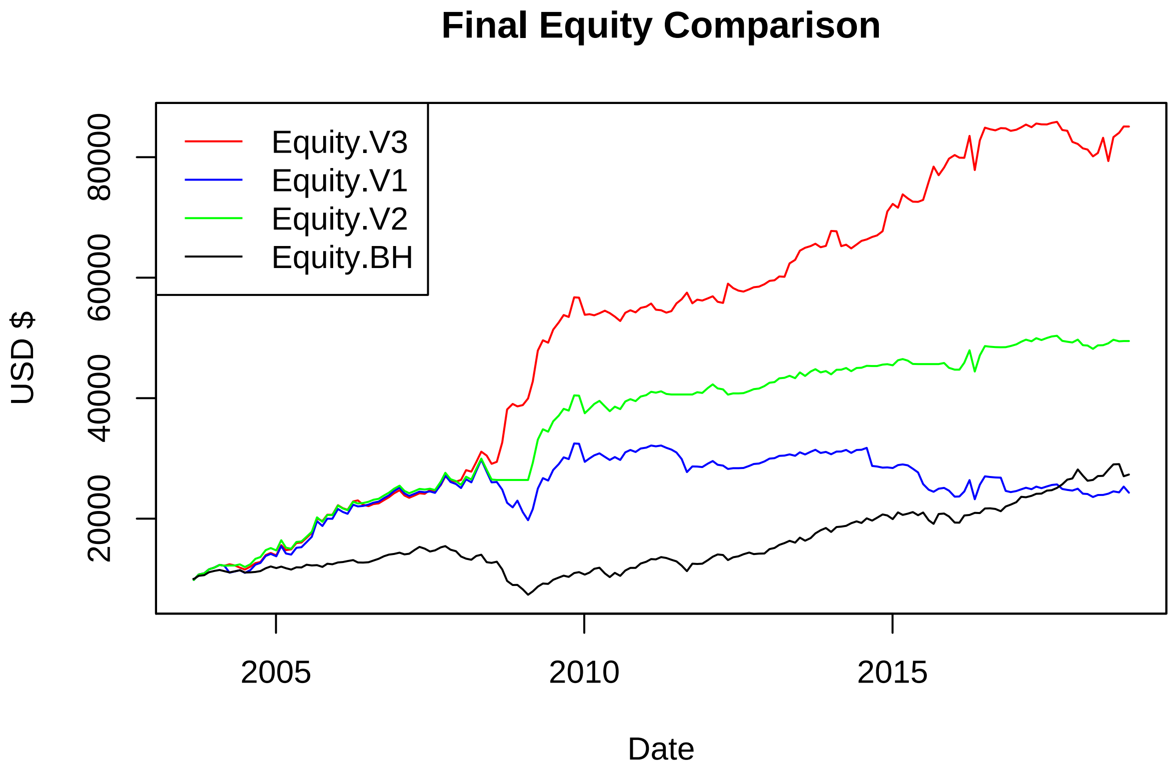


```
par(mfrow=c(1,1))
```

5.3.3 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.

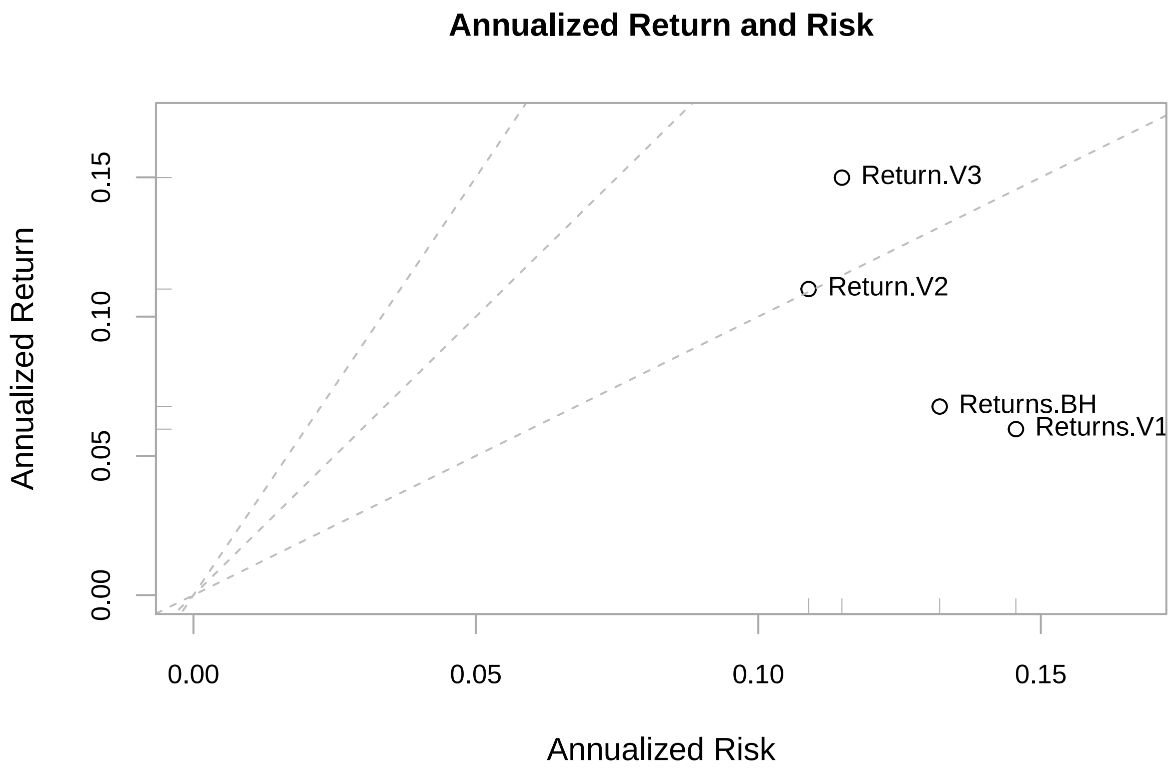
```
equity.bh.v1.v2.v3 <- cbind(equity.v3, equity.v1, equity.v2, equity.bh)
equity.bh.v1.v2.v3 <- na.omit(equity.bh.v1.v2.v3)

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



5.3.4 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)
```

5.3.4.1 General Strategy Statistics of the improved V3 strategy

```
tab.perf <- table.Arbitrary(rets.bh.v1.v2.v3, 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.bh.v1.v2.v3, metrics=c("StdDev.annualized",
                                            "maxDrawdown",
                                            "VaR",
                                            "ES"),
                            metricsNames=c("Annualized StdDev",
                                           "Max Drawdown",
                                           "Value-at-Risk",
                                           "Conditional VaR"),
                            method=c("historical"))
```

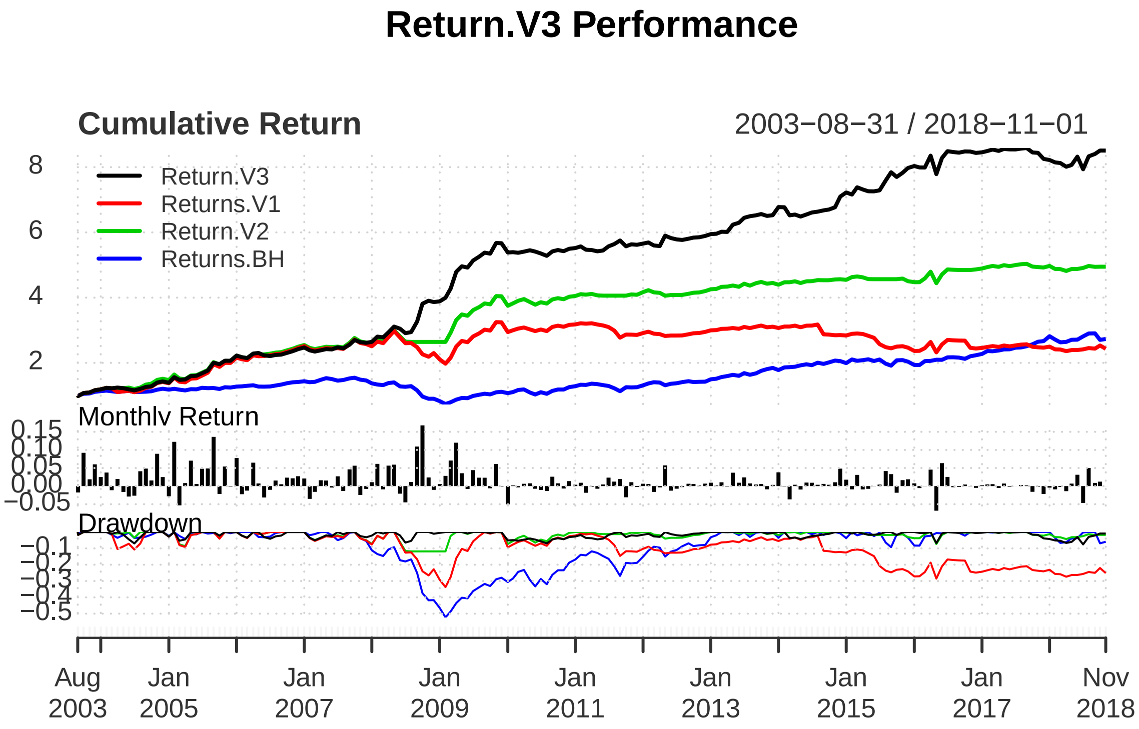
```
somestats <- data.frame(rownames(tab.perf),tab.perf[,1], rownames(tab.risk),tab.risk[,1
colnames(somestats) <- c("Performance Metric", "Performance Value", "Risk Metric", "Risk
somestats
```

Performance Metric	Performance Value	Risk Metric	Risk Value
Cumulative Return	7.5087708	Annualized StdDev	0.1148201
Annualized Return	0.1498561	Max Drawdown	0.0758307
Annualized Sharp Ratio	1.3051385	Value-at-Risk	-0.0286469
Calmar Ratio	1.9761937	Conditional VaR	-0.0425340

5.3.5 Compare returns of all strategies

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.

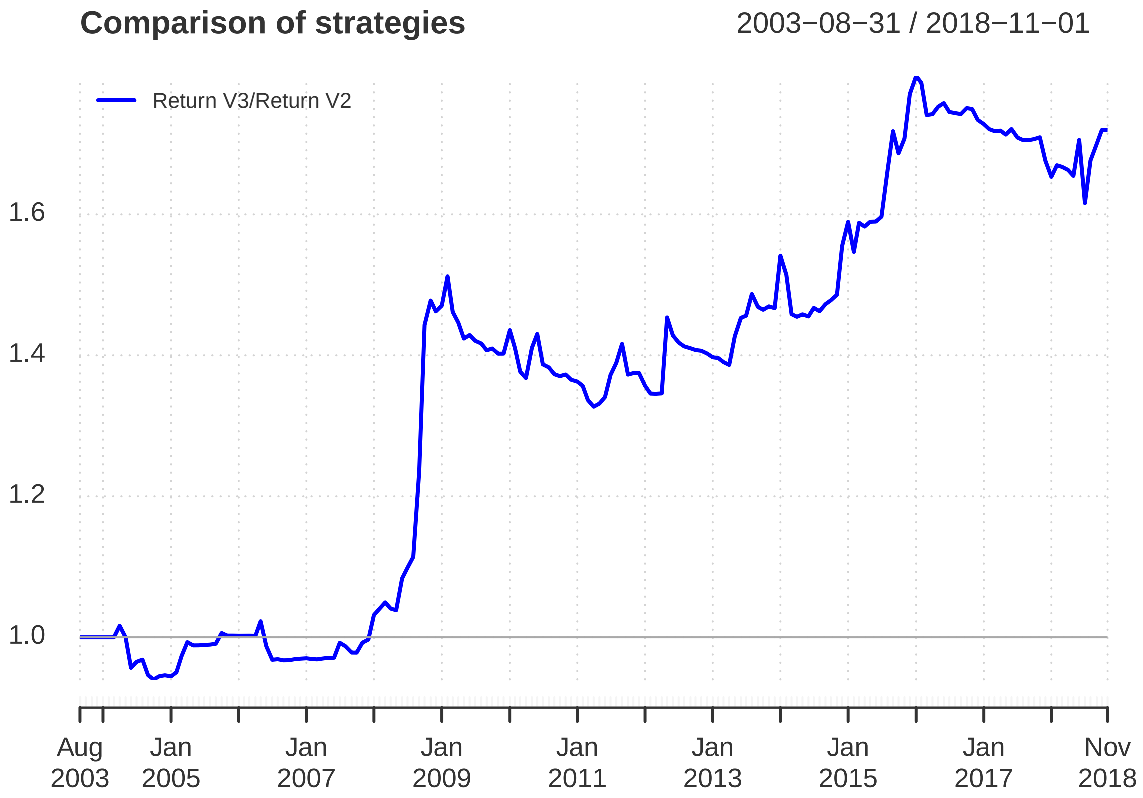
```
tab.perf
tab.risk
charts.PerformanceSummary(rets.bh.v1.v2.v3,geometric=T,wealth.index=TRUE)
```



5.3.6 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 short-selling outperformed the GRS.V2 strategy.

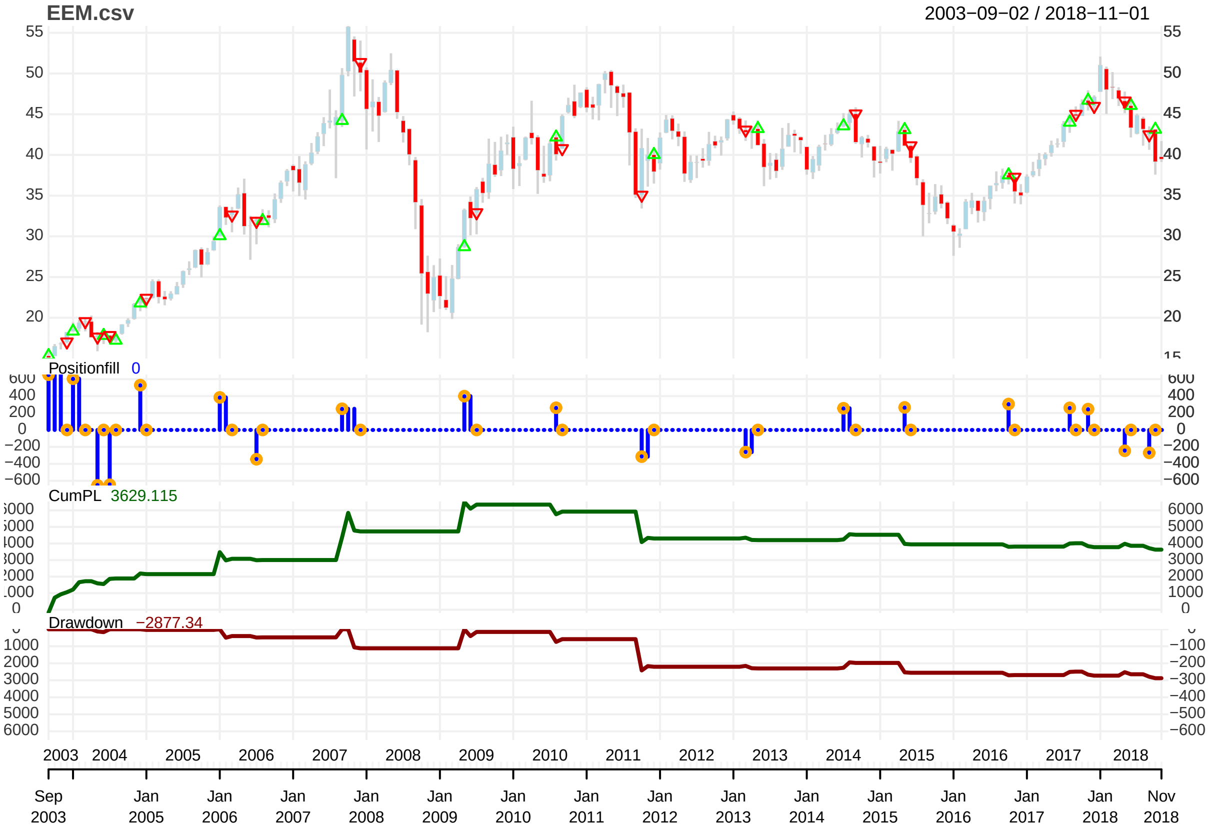
```
chart.RelativePerformance(rets.v3, rets.v2, main="Comparison of strategies", colorset =
```

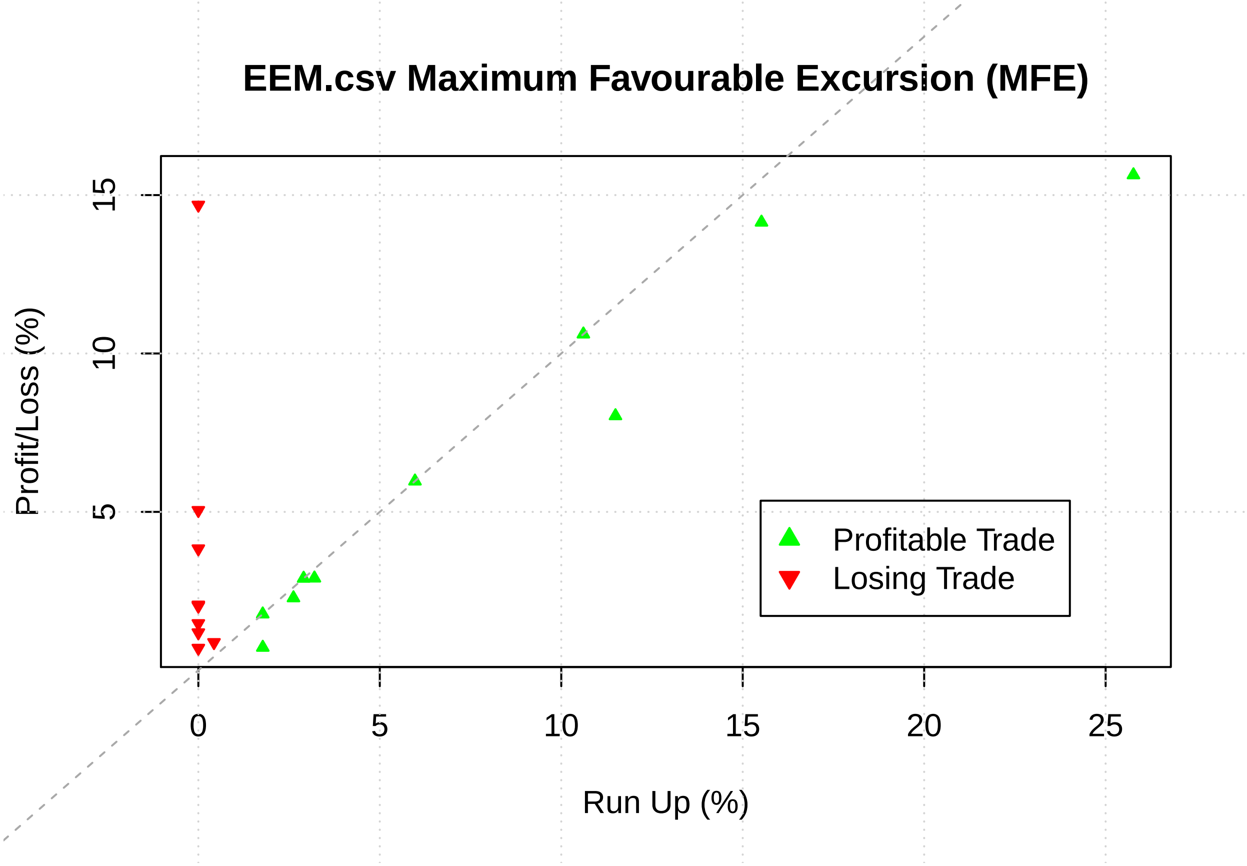


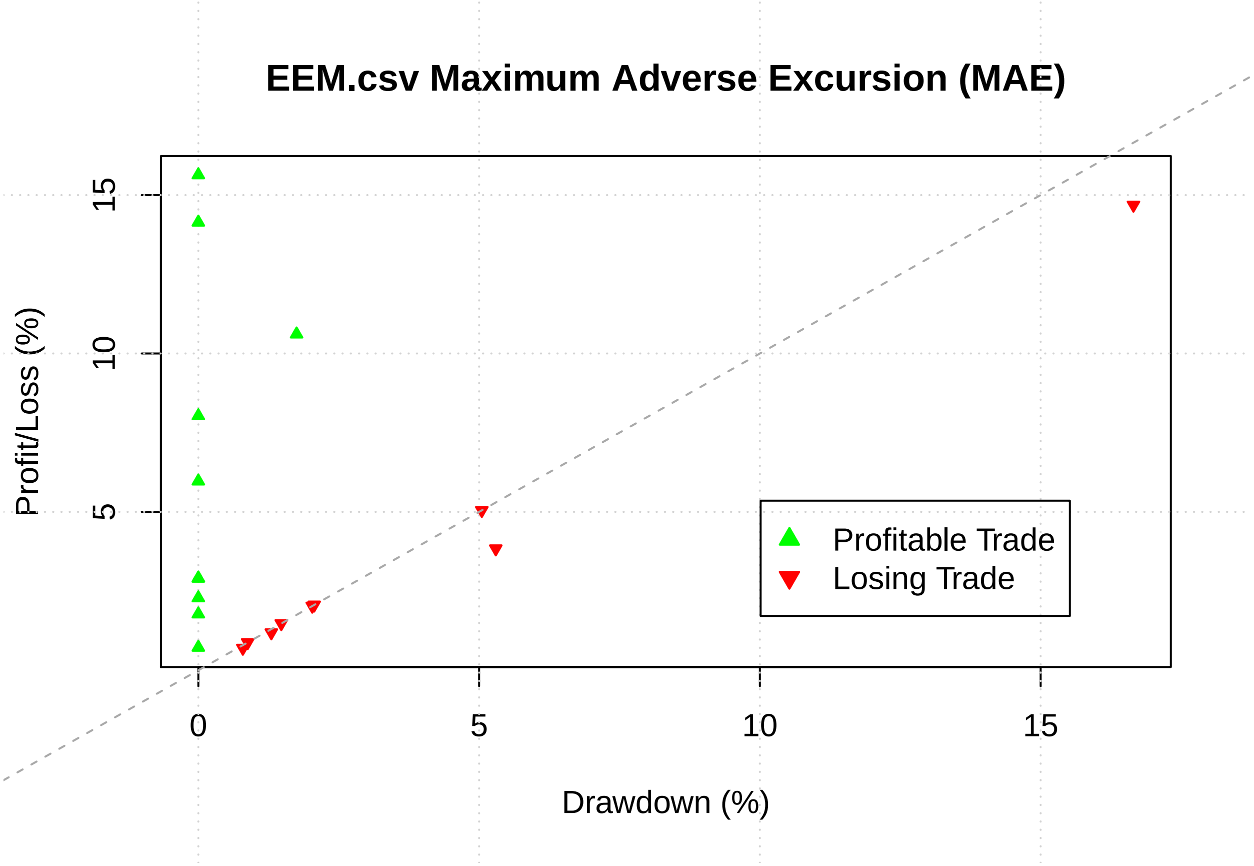
5.3.7 Could we have performed better

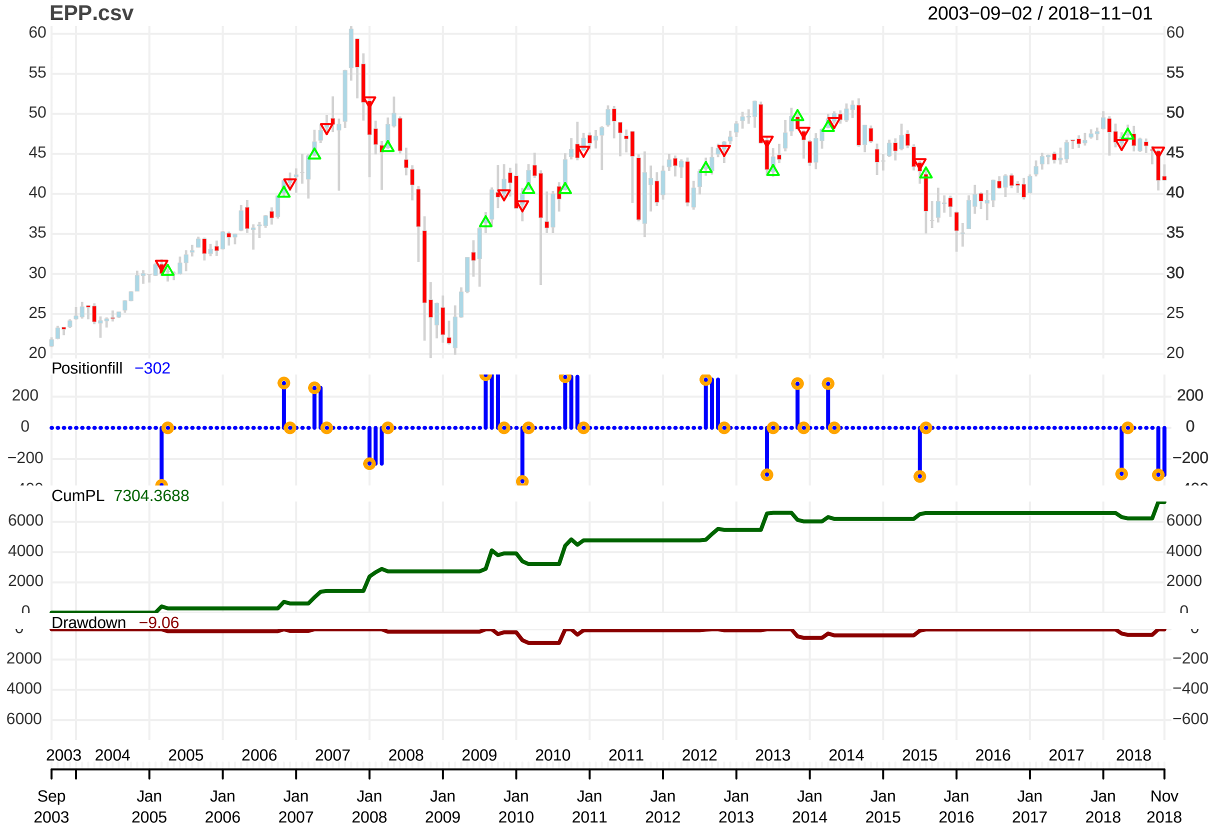
In the following, the MAE and MFE plots for all traded instruments are plotted. We can see, that there is still some room for improvement which could be improved by initiating stop loss or trailing stop orders.

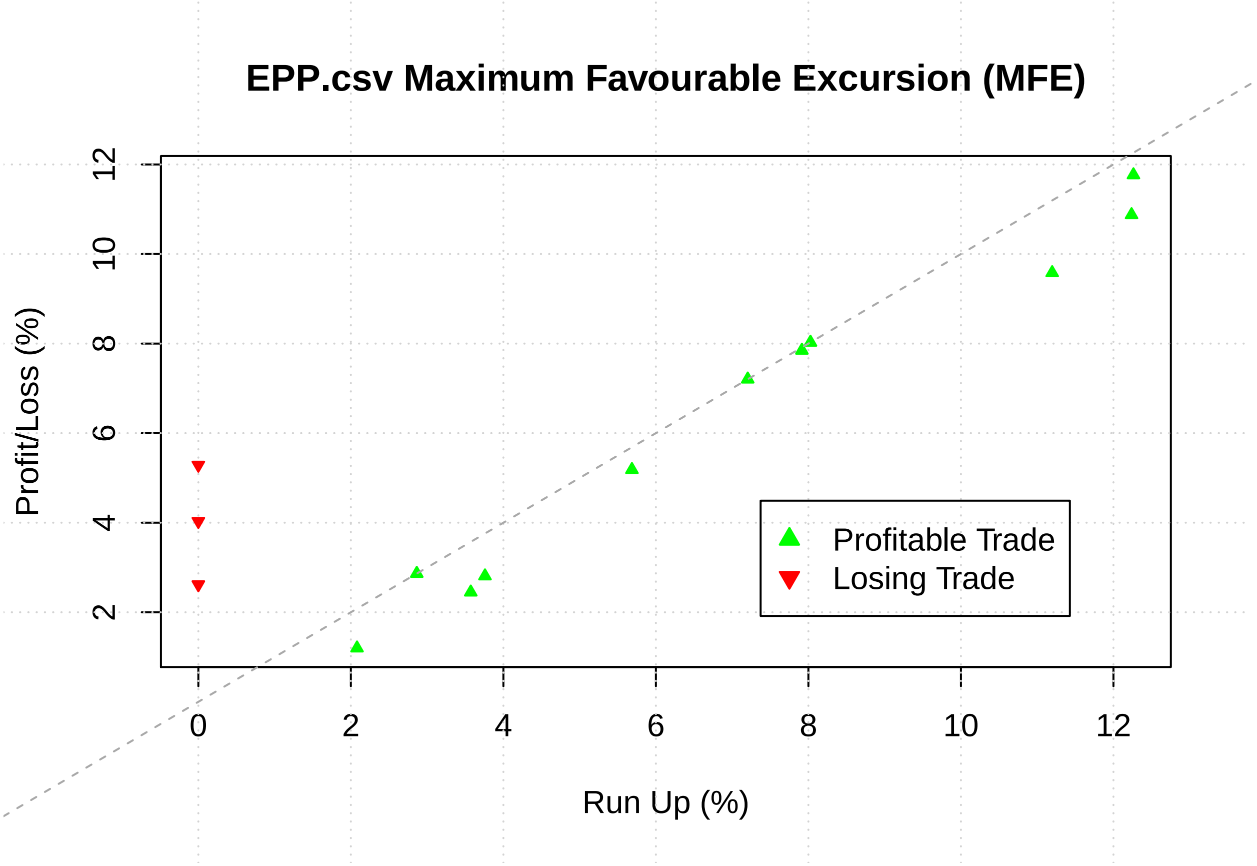
```
for (symbol in symbollist)
{
    if (nrow(getTxns(portfolioname.alt.v3,Symbol=symbol)) > 1)
    {
        symboldata <- get(symbol)
        chart.Posn(portfolioname.alt.v3,Symbol=symbol,type='candlesticks', theme=myTheme
        chart.ME(portfolioname.alt.v3, Symbol=symbol, type="MFE", scale="percent")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

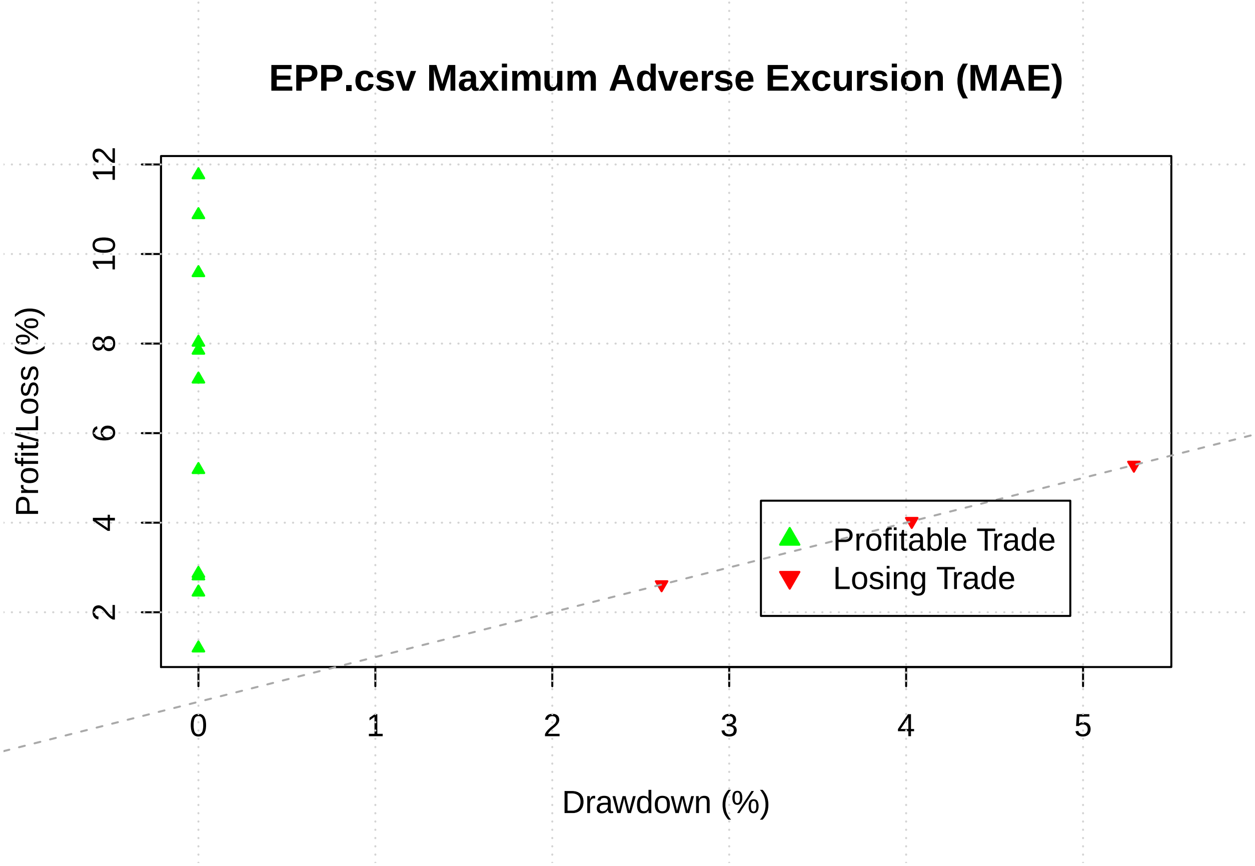

EEM.csv
2003−09−02 / 2018−11−01
55
50
45
40
35
30
25
20
15
Positionfill 0
600
400
200
0
−200
−400
−600
CumPL 3629.115
6000
5000
4000
3000
2000
1000
0
Drawdown −2877.34
0
1000
2000
3000
4000
5000
6000
−100
−200
−300
−400
−500
−600
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018

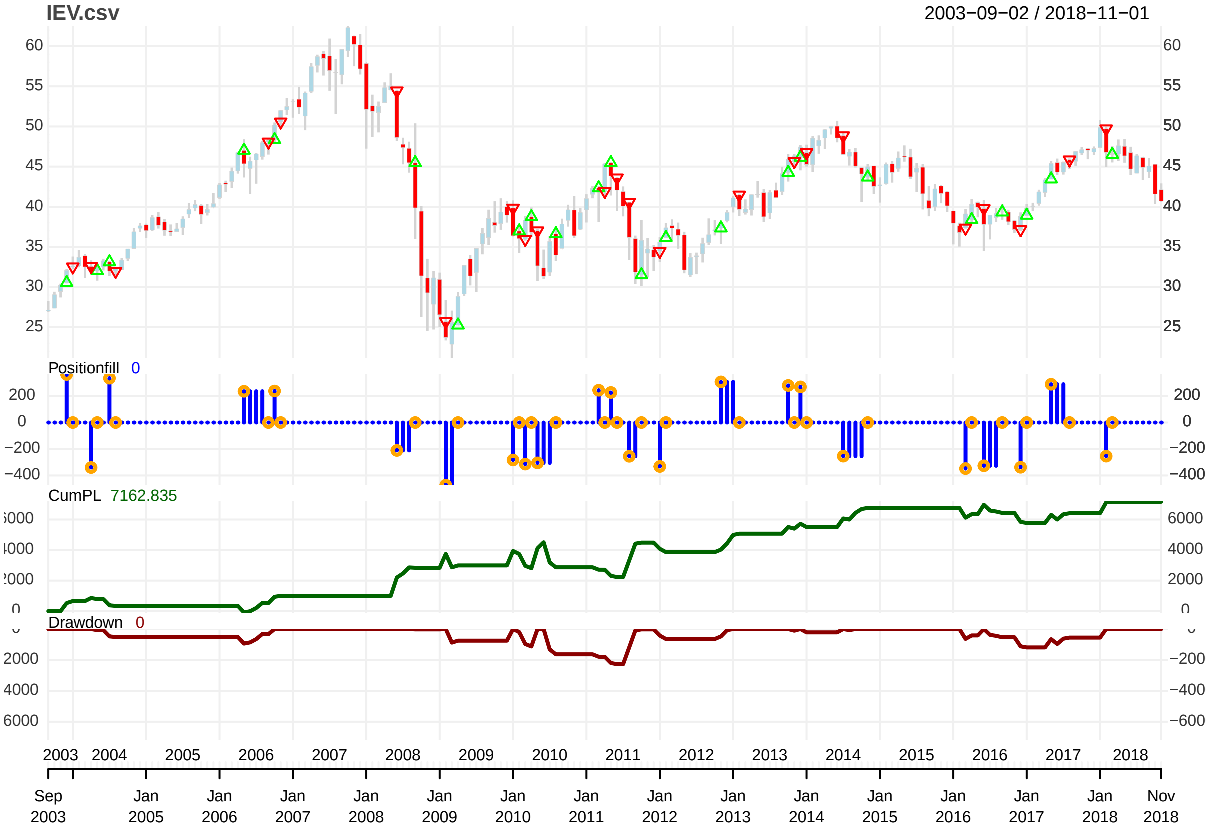

EEM.csv Maximum Favourable Excursion (MFE)
Profit/Loss (%)
15
10
5
0
5
10
15
20
25
Run Up (%)
▲ Profitable Trade
▼ Losing Trade

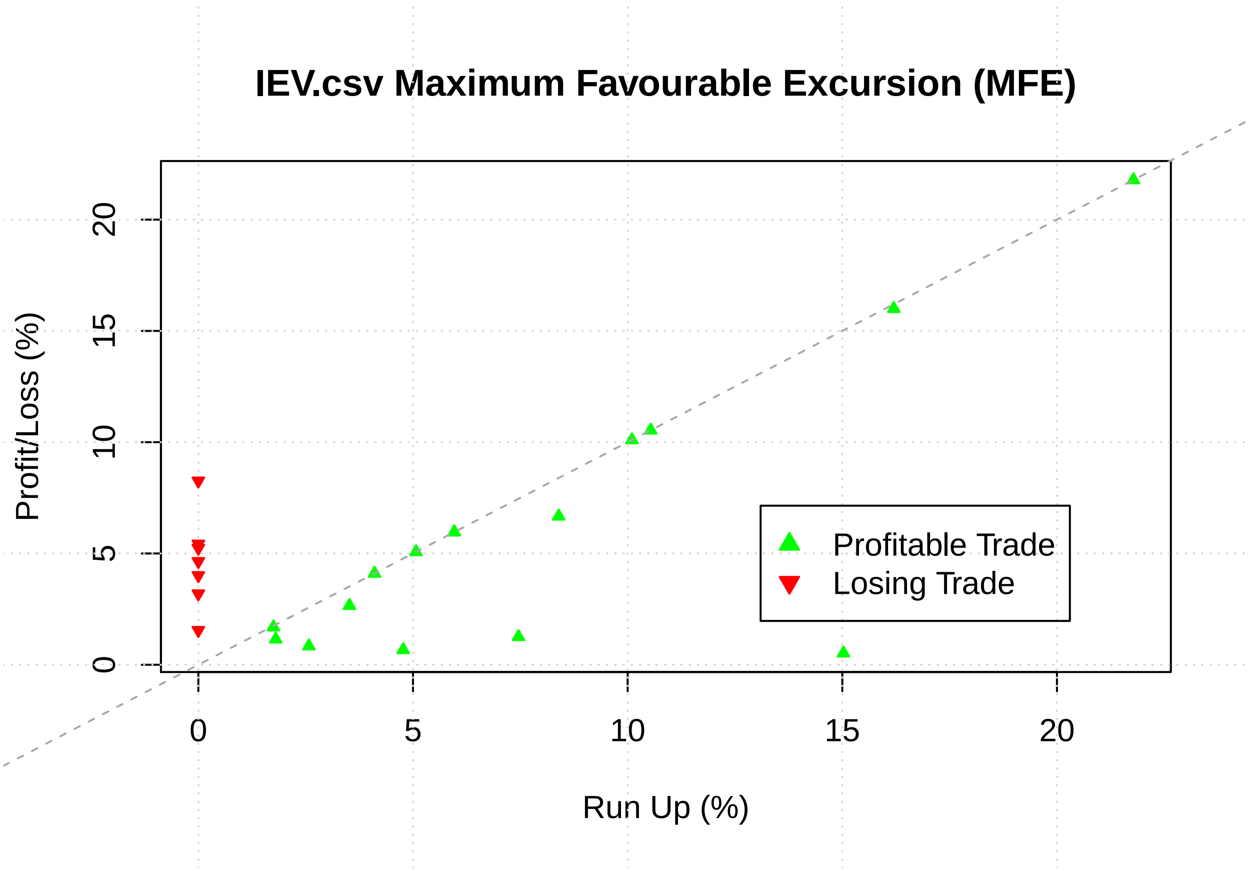

EEM.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
15
10
5
0
5
10
15
Drawdown (%)
▲ Profitable Trade
▼ Losing Trade

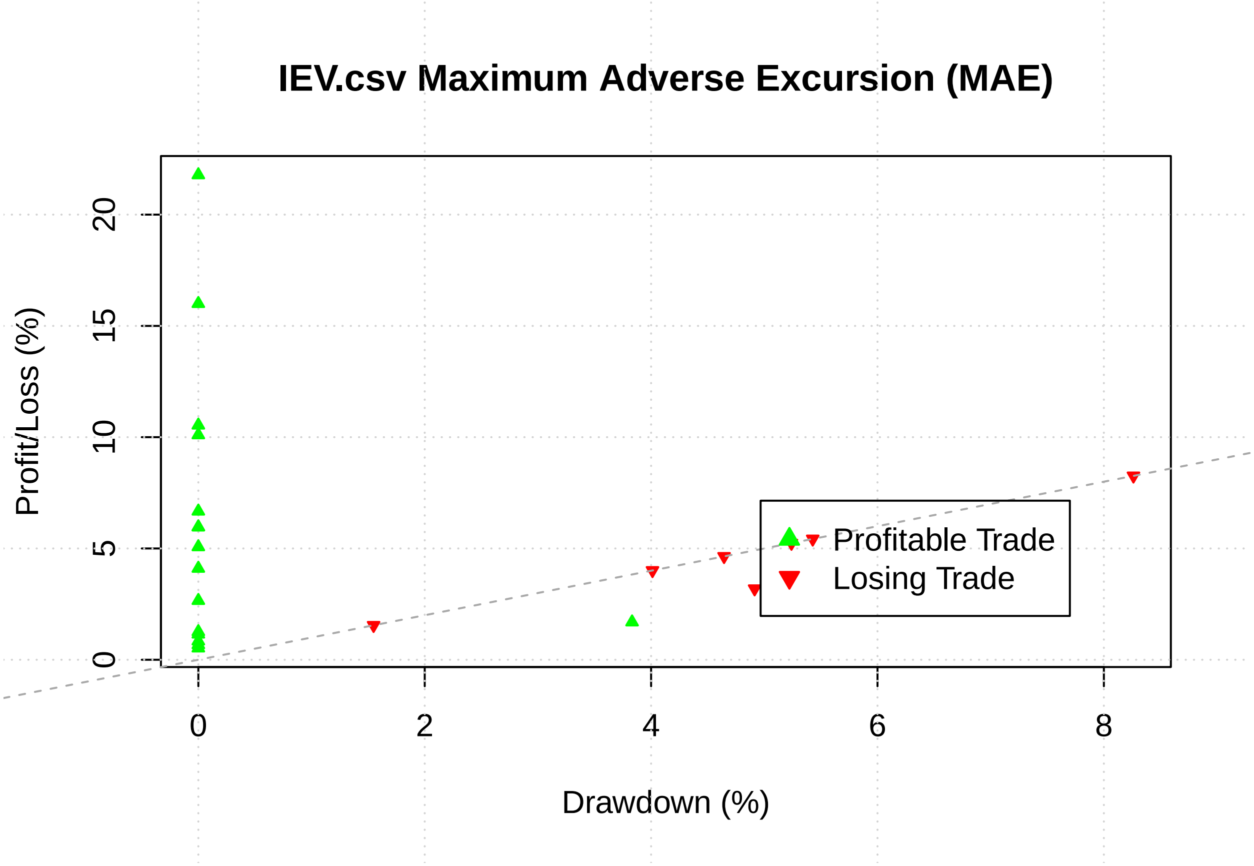

EPP.csv
2003−09−02 / 2018−11−01
60
55
50
45
40
35
30
25
20
Positionfill −302
200
0
−200
CumPL 7304.3688
6000
4000
2000
0
Drawdown −9.06
2000
4000
6000
−200
−400
−600
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018

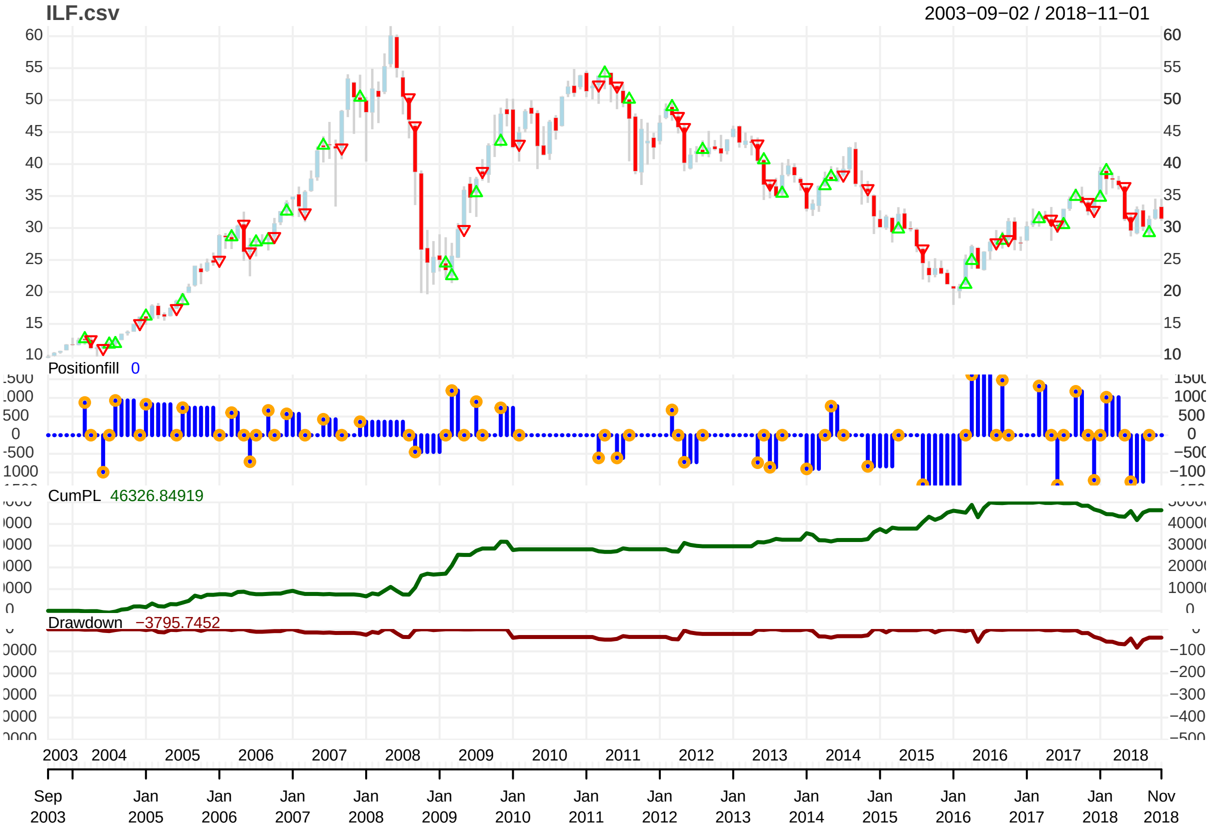

EPP.csv Maximum Favourable Excursion (MFE)
Profit/Loss (%)
2
4
6
8
10
12
0
2
4
6
8
10
12
Run Up (%)
▲ Profitable Trade
▼ Losing Trade

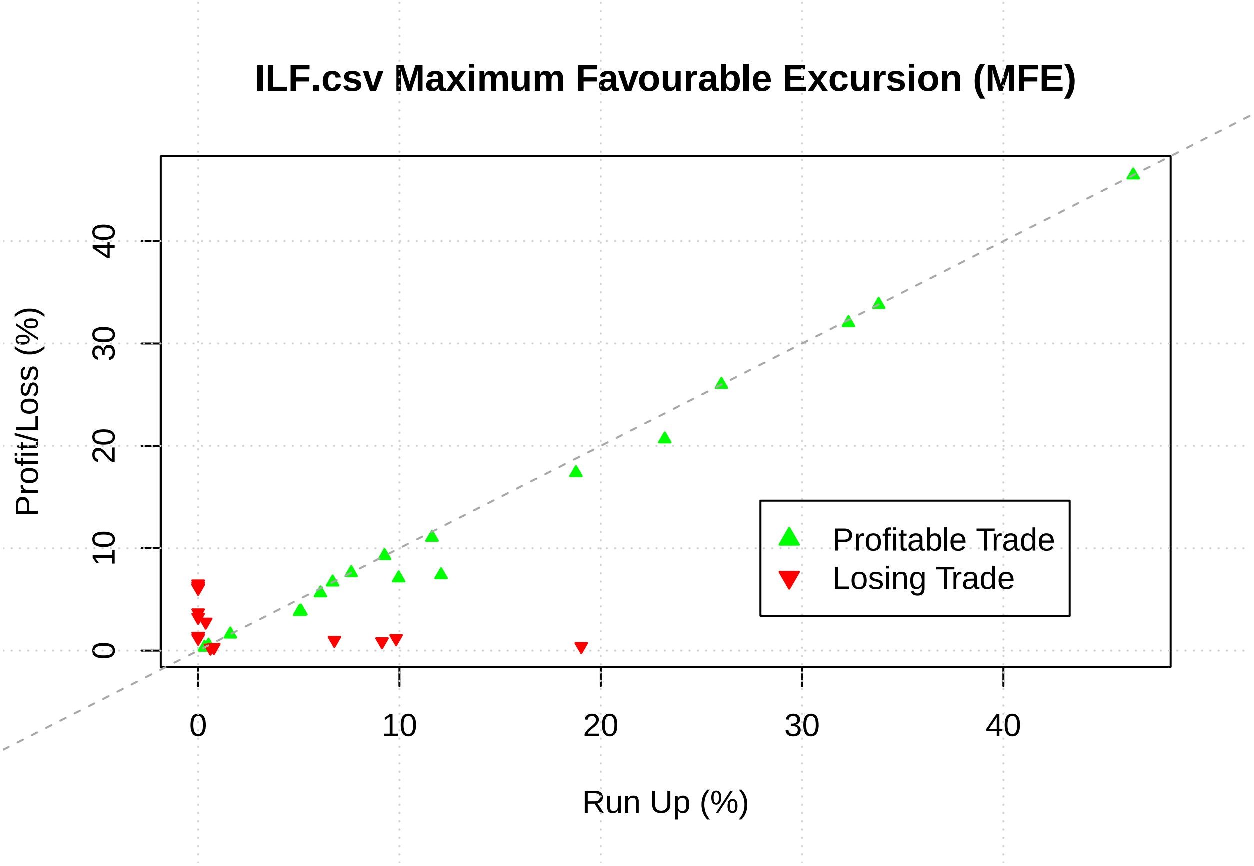

EPP.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
2
4
6
8
10
12
0
1
2
3
4
5
Drawdown (%)
▲ Profitable Trade
▼ Losing Trade

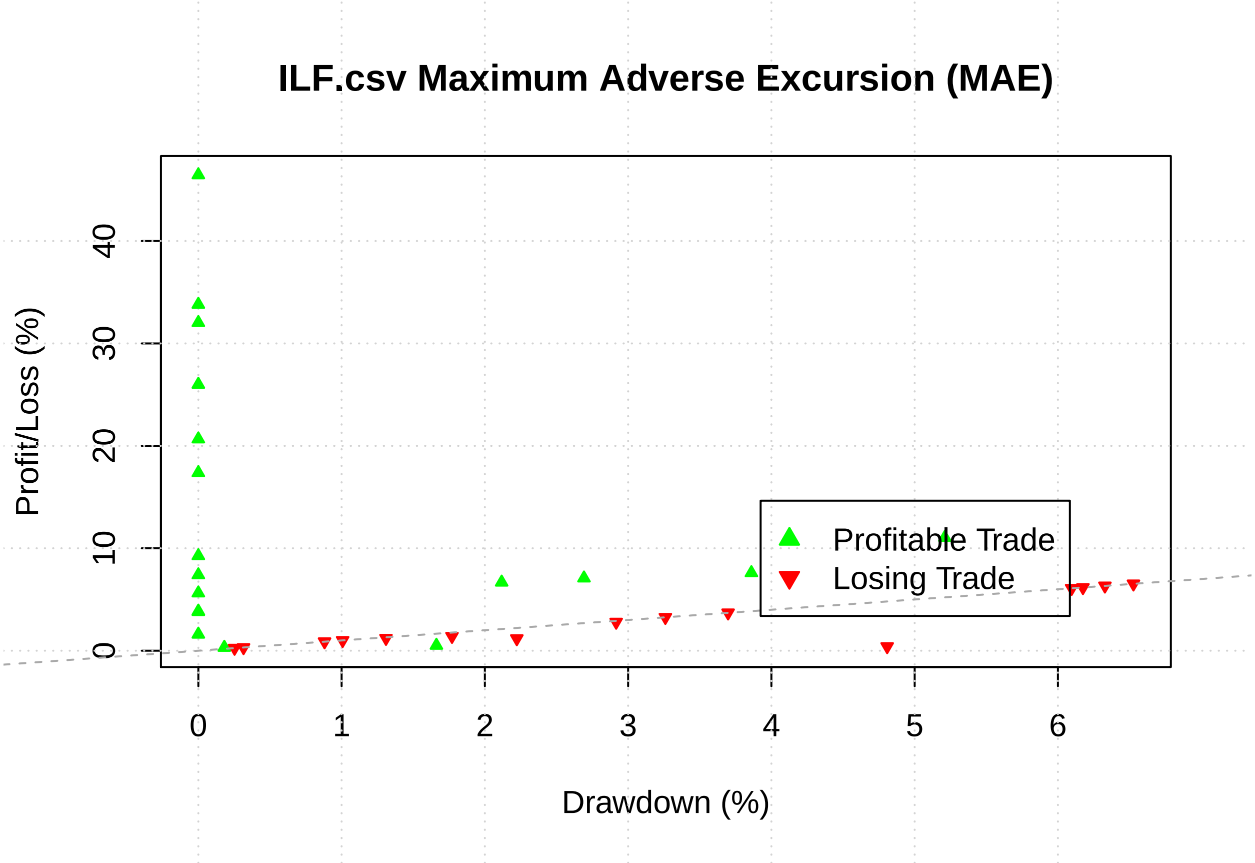

IEV.csv
2003−09−02 / 2018−11−01
60
55
50
45
40
35
30
25
Positionfill 0
200
0
−200
−400
CumPL 7162.835
6000
4000
2000
0
Drawdown 0
0
−2000
−4000
−6000
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018

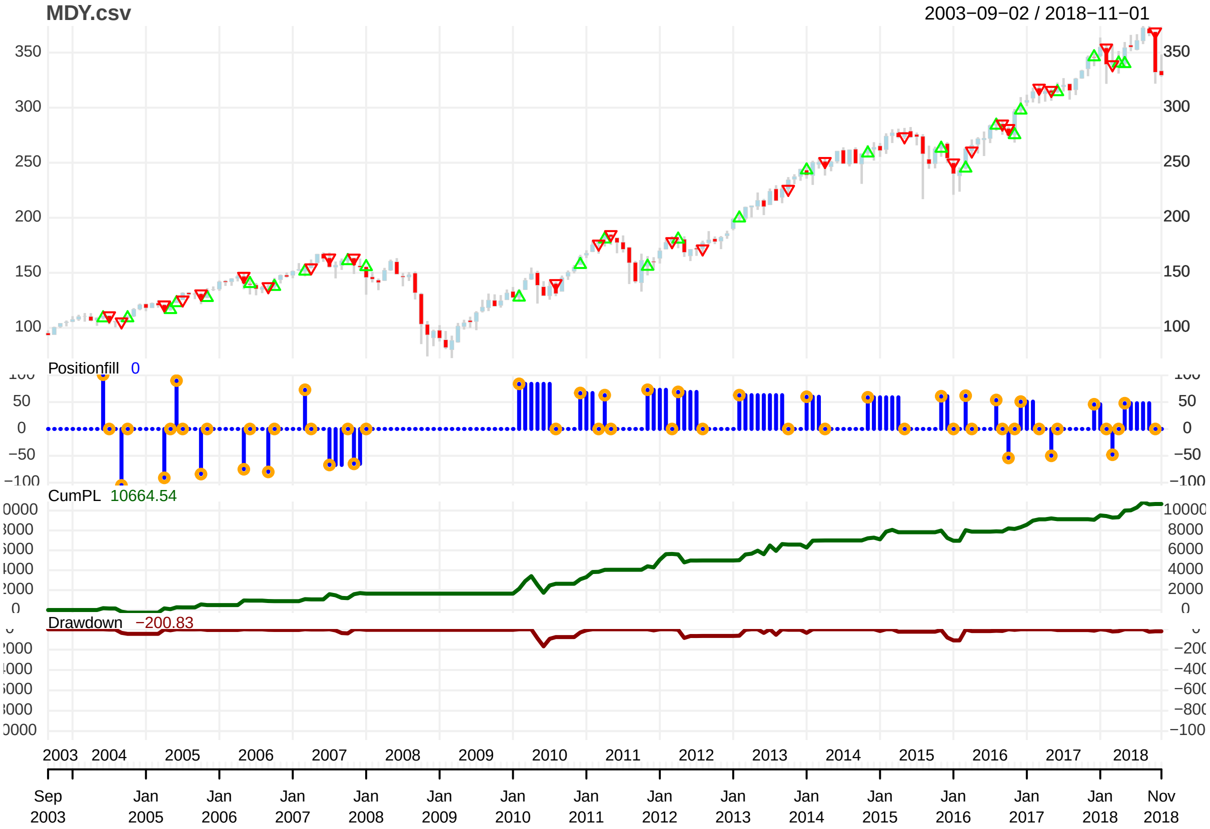

IEV.csv Maximum Favourable Excursion (MFE)
Profit/Loss (%)
0
5
10
15
20
Run Up (%)
0
5
10
15
20
▲ Profitable Trade
▼ Losing Trade

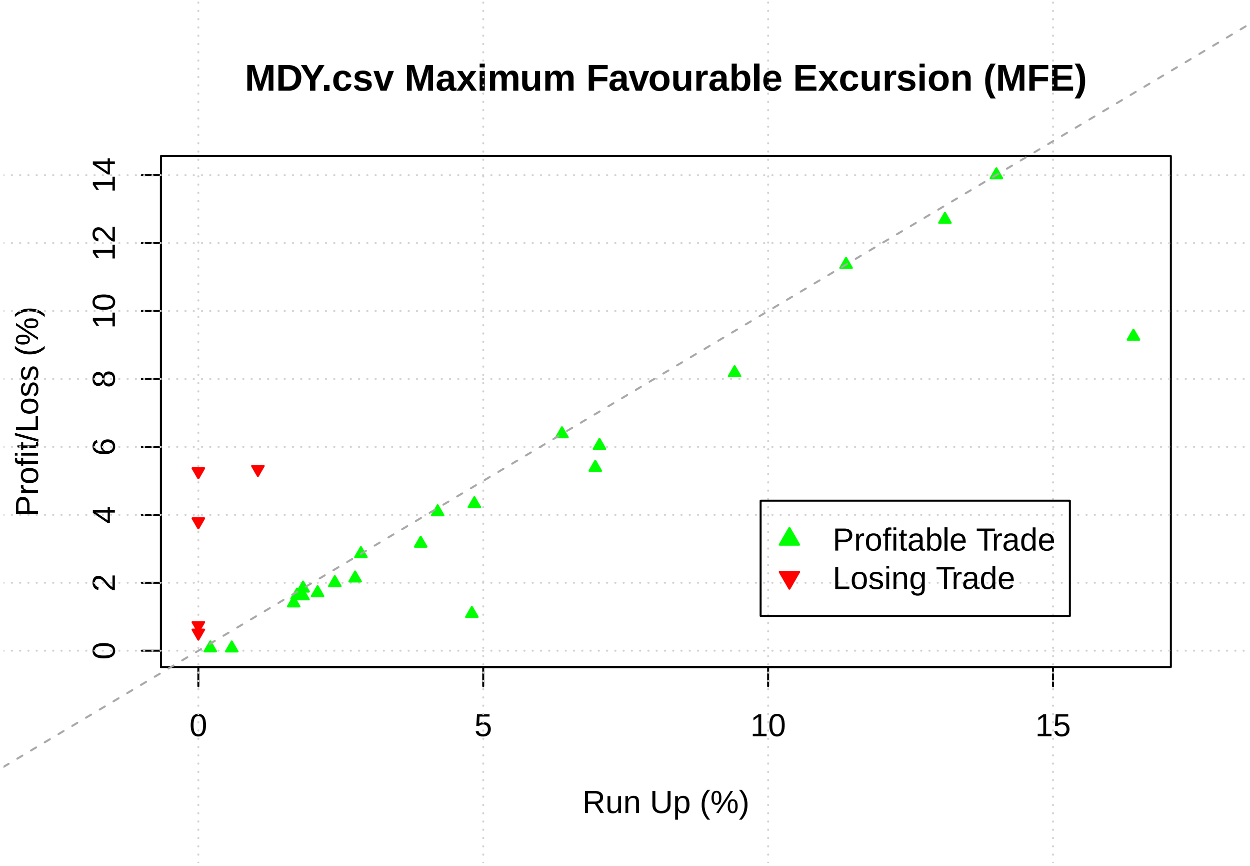

IEV.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
0
5
10
15
20
Drawdown (%)
0
2
4
6
8
Profitable Trade
Losing Trade

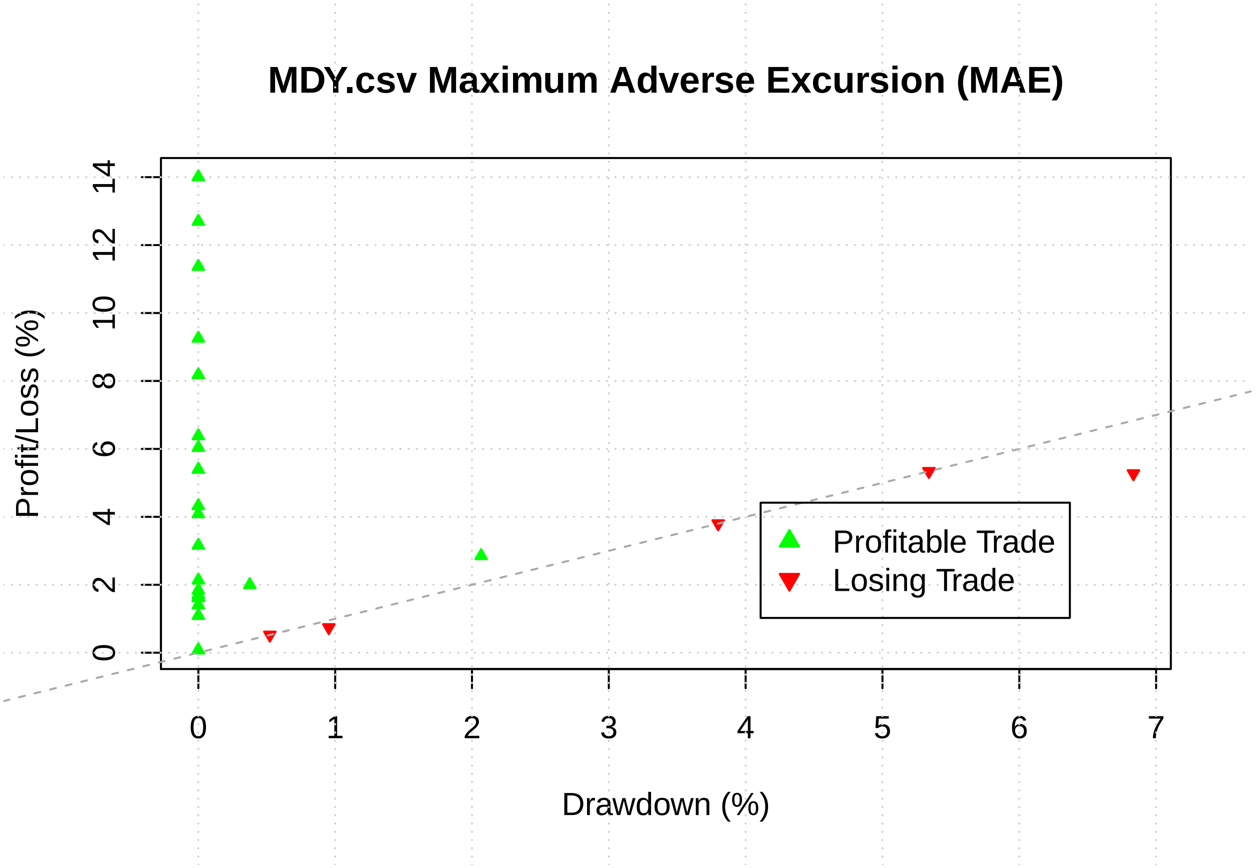

ILF.csv
2003−09−02 / 2018−11−01
60
55
50
45
40
35
30
25
20
15
10
Positionfill 0
1500
1000
500
0
−500
−1000
CumPL 46326.84919
50000
40000
30000
20000
10000
0
Drawdown −3795.7452
0
−1000
−2000
−3000
−4000
−5000
2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018


ILF.csv Maximum Favourable Excursion (MFE)
Profit/Loss (%)
0
10
20
30
40
Run Up (%)
0
10
20
30
40
▲ Profitable Trade
▼ Losing Trade


ILF.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
0
10
20
30
40
Drawdown (%)
0
1
2
3
4
5
6
▲ Profitable Trade
▼ Losing Trade


MDY.csv
2003−09−02 / 2018−11−01
350
300
250
200
150
100
Positionfill 0
100
50
0
−50
−100
CumPL 10664.54
10000
8000
6000
4000
2000
0
Drawdown −200.83
0
−2000
−4000
−6000
−8000
−10000
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Sep 2003
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Nov 2018


MDY.csv Maximum Favourable Excursion (MFE)
Profit/Loss (%)
0
2
4
6
8
10
12
14
Run Up (%)
0
5
10
15
▲ Profitable Trade
▼ Losing Trade



5.3.8 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
```

5.3.8.1 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
## -0.12218 -0.09151 -0.06993  0.04444  0.35041
```

```
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.099385   0.010300   9.649   <2e-16 ***
## mkt.rf      -0.002808   0.002582  -1.087    0.278    
## ---
## 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:  0.006635,   Adjusted R-squared:  0.001023 
## F-statistic: 1.182 on 1 and 177 DF,  p-value: 0.2784
```

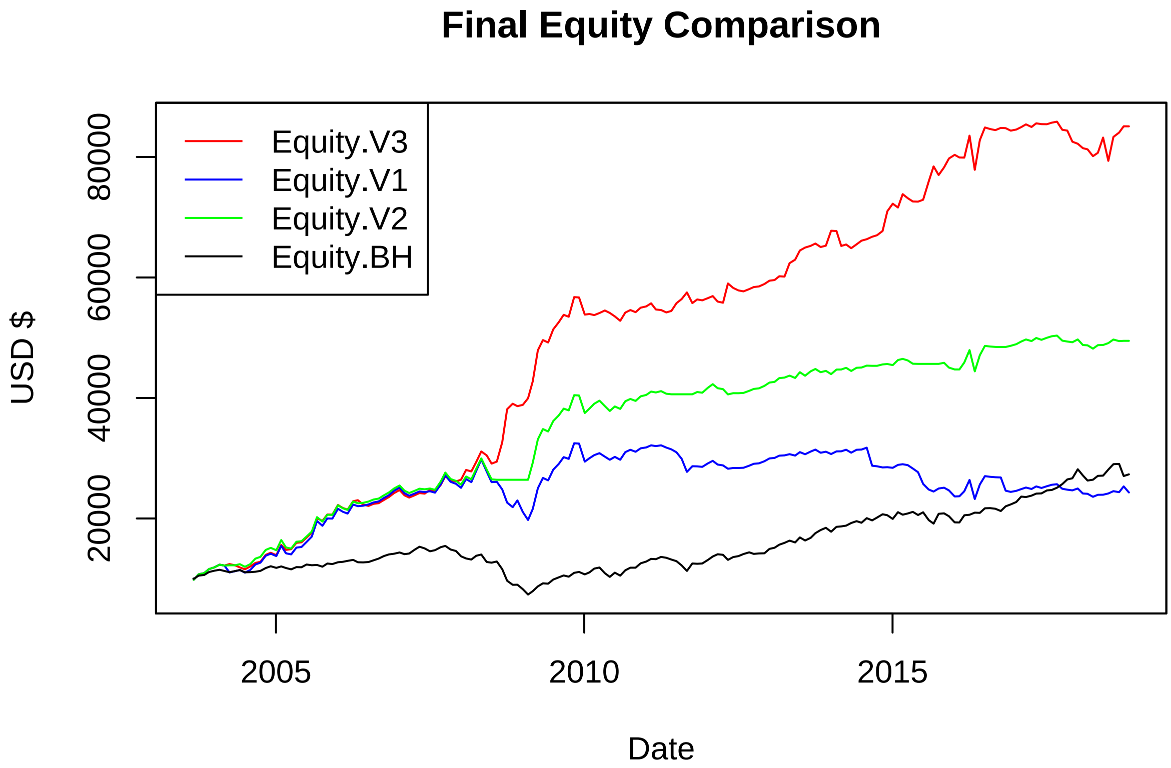
5.3.8.2 Perform a Fama-French 3 factor Regression

```
# 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 
## -0.12668 -0.09158 -0.07133  0.04209  0.35214 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.099722   0.010348   9.637   <2e-16 ***
## mkt.rf      -0.003104   0.002851  -1.089    0.278    
## smb         -0.001342   0.004747  -0.283    0.778    
## hml          0.003299   0.004277   0.771    0.442    
## ---
## 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:  0.01034,    Adjusted R-squared:  -0.006623 
## F-statistic: 0.6096 on 3 and 175 DF,  p-value: 0.6096
```

The analysis of the improved V3 strategy's risk shows again what we have seen with the V2 improvement strategy. We can generate a significant alpha of approx. 0.1 and by this generate risk adjusted excess returns.

6 Conclusion


Final Equity Comparison

```
## Performance Metrics
```

	Returns.V1	Returns.BH	Return.V2	Return.V3
Cumulative Return	1.4311626	1.7323101	3.9484875	7.5087708
Annualized Return	0.0596484	0.0677494	0.1099200	0.1498561
Annualized Sharp Ratio	0.4096519	0.5127792	1.0098101	1.3051385
Calmar Ratio	0.1769151	0.1296325	0.9283629	1.9761937

```
## Risk Metrics
```

	Returns.V1	Returns.BH	Return.V2	Return.V3
Annualized StdDev	0.1456075	0.1321220	0.1088522	0.1148201
Max Drawdown	0.3371584	0.5226270	0.1184020	0.0758307
Value-at-Risk	-0.0695126	-0.0612741	-0.0260376	-0.0286469
Conditional VaR	-0.0878293	-0.0897504	-0.0511959	-0.0425340

7 Appendix

7.1 Orderbook

```
ob <- getOrderBook(portfolioname)
ob
```

```
## $GRS
## $GRS$EEM.csv
##                     Order.Qty Order.Price Order.Type Order.Side
## 2003-09-02 00:00:00 "652"     "15.333"    "market"   "long"
## 2003-12-01 00:00:00 "all"     "17.044"    "market"   "long"
## 2004-01-02 00:00:00 "605"     "18.344"    "market"   "long"
## 2004-03-01 00:00:00 "all"     "19.538"    "market"   "long"
## 2004-04-01 00:00:00 "591"     "19.543"    "market"   "long"
## 2004-05-03 00:00:00 "all"     "17.642"    "market"   "long"
## 2004-12-01 00:00:00 "529"     "21.8"      "market"   "long"
## 2005-01-03 00:00:00 "all"     "22.411"    "market"   "long"
## 2006-01-03 00:00:00 "384"     "30.06"     "market"   "long"
## 2006-03-01 00:00:00 "all"     "32.63"     "market"   "long"
## 2007-09-04 00:00:00 "250"     "44.217"    "market"   "long"
## 2007-12-03 00:00:00 "all"     "51.37"     "market"   "long"
## 2009-05-01 00:00:00 "399"     "28.71"     "market"   "long"
## 2009-07-01 00:00:00 "all"     "32.92"     "market"   "long"
## 2010-08-02 00:00:00 "262"     "42.18"     "market"   "long"
## 2010-09-01 00:00:00 "all"     "40.79"     "market"   "long"
## 2011-06-01 00:00:00 "228"     "48.43"     "market"   "long"
## 2011-07-01 00:00:00 "all"     "47.58"     "market"   "long"
## 2011-08-01 00:00:00 "232"     "47.67"     "market"   "long"
## 2011-09-01 00:00:00 "all"     "42.87"     "market"   "long"
## 2014-07-01 00:00:00 "254"     "43.55"     "market"   "long"
## 2014-09-02 00:00:00 "all"     "45.05"     "market"   "long"
## 2015-05-01 00:00:00 "264"     "43.09"     "market"   "long"
## 2015-06-01 00:00:00 "all"     "41.14"     "market"   "long"
## 2016-10-03 00:00:00 "303"     "37.53"     "market"   "long"
## 2016-11-01 00:00:00 "all"     "37.28"     "market"   "long"
## 2017-08-01 00:00:00 "258"     "44"        "market"   "long"
## 2017-09-01 00:00:00 "all"     "45.01"     "market"   "long"
## 2017-11-01 00:00:00 "243"     "46.7"      "market"   "long"
## 2017-12-01 00:00:00 "all"     "45.98"     "market"   "long"
##                     Order.Threshold Order.Status Order.StatusTime
## 2003-09-02 00:00:00 NA              "closed"     "2003-10-01 00:00:00"
## 2003-12-01 00:00:00 NA              "closed"     "2004-01-02 00:00:00"
## 2004-01-02 00:00:00 NA              "closed"     "2004-02-02 00:00:00"
## 2004-03-01 00:00:00 NA              "closed"     "2004-04-01 00:00:00"
## 2004-04-01 00:00:00 NA              "closed"     "2004-05-03 00:00:00"
## 2004-05-03 00:00:00 NA              "closed"     "2004-06-01 00:00:00"
## 2004-12-01 00:00:00 NA              "closed"     "2005-01-03 00:00:00"
## 2005-01-03 00:00:00 NA              "closed"     "2005-02-01 00:00:00"
## 2006-01-03 00:00:00 NA              "closed"     "2006-02-01 00:00:00"
## 2006-03-01 00:00:00 NA              "closed"     "2006-04-03 00:00:00"
## 2007-09-04 00:00:00 NA              "closed"     "2007-10-01 00:00:00"
## 2007-12-03 00:00:00 NA              "closed"     "2008-01-02 00:00:00"
## 2009-05-01 00:00:00 NA              "closed"     "2009-06-01 00:00:00"
## 2009-07-01 00:00:00 NA              "closed"     "2009-08-03 00:00:00"
## 2010-08-02 00:00:00 NA              "closed"     "2010-09-01 00:00:00"
## 2010-09-01 00:00:00 NA              "closed"     "2010-10-01 00:00:00"
## 2011-06-01 00:00:00 NA              "closed"     "2011-07-01 00:00:00"
## 2011-07-01 00:00:00 NA              "closed"     "2011-08-01 00:00:00"
## 2011-08-01 00:00:00 NA              "closed"     "2011-09-01 00:00:00"
## 2011-09-01 00:00:00 NA              "closed"     "2011-10-03 00:00:00"
## 2014-07-01 00:00:00 NA              "closed"     "2014-08-01 00:00:00"
## 2014-09-02 00:00:00 NA              "closed"     "2014-10-01 00:00:00"
## 2015-05-01 00:00:00 NA              "closed"     "2015-06-01 00:00:00"
## 2015-06-01 00:00:00 NA              "closed"     "2015-07-01 00:00:00"
## 2016-10-03 00:00:00 NA              "closed"     "2016-11-01 00:00:00"
## 2016-11-01 00:00:00 NA              "closed"     "2016-12-01 00:00:00"
## 2017-08-01 00:00:00 NA              "closed"     "2017-09-01 00:00:00"
## 2017-09-01 00:00:00 NA              "closed"     "2017-10-02 00:00:00"
## 2017-11-01 00:00:00 NA              "closed"     "2017-12-01 00:00:00"
## 2017-12-01 00:00:00 NA              "closed"     "2018-01-02 00:00:00"
##                     Prefer Order.Set Txn.Fees Rule        Time.In.Force
## 2003-09-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2003-12-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2004-01-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2004-03-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2004-04-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2004-05-03 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2004-12-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2005-01-03 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2006-01-03 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2006-03-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2007-09-04 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2007-12-03 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2009-05-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2009-07-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2010-08-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2010-09-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2011-06-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2011-07-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2011-08-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2011-09-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2014-07-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2014-09-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2015-05-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2015-06-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2016-10-03 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2016-11-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2017-08-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2017-09-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2017-11-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2017-12-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
##
## $GRS$EPP.csv
##                     Order.Qty Order.Price Order.Type Order.Side
## 2006-11-01 00:00:00 "275"     "40.0667"   "market"   "long"
## 2006-12-01 00:00:00 "all"     "41.4"      "market"   "long"
## 2007-04-02 00:00:00 "246"     "44.8367"   "market"   "long"
## 2007-06-01 00:00:00 "all"     "48.3"      "market"   "long"
## 2009-08-03 00:00:00 "313"     "36.37"     "market"   "long"
## 2009-11-02 00:00:00 "all"     "40.03"     "market"   "long"
## 2010-09-01 00:00:00 "301"     "40.52"     "market"   "long"
## 2010-12-01 00:00:00 "all"     "45.47"     "market"   "long"
## 2011-09-01 00:00:00 "276"     "44.52"     "market"   "long"
## 2011-10-03 00:00:00 "all"     "36.29"     "market"   "long"
## 2012-08-01 00:00:00 "284"     "43.15"     "market"   "long"
## 2012-11-01 00:00:00 "all"     "45.58"     "market"   "long"
## 2013-11-01 00:00:00 "260"     "49.63"     "market"   "long"
## 2013-12-02 00:00:00 "all"     "47.84"     "market"   "long"
## 2014-04-01 00:00:00 "268"     "48.28"     "market"   "long"
## 2014-05-01 00:00:00 "all"     "49.07"     "market"   "long"
## 2016-01-04 00:00:00 "343"     "37.7"      "market"   "long"
## 2016-02-01 00:00:00 "all"     "35.04"     "market"   "long"
##                     Order.Threshold Order.Status Order.StatusTime
## 2006-11-01 00:00:00 NA              "closed"     "2006-12-01 00:00:00"
## 2006-12-01 00:00:00 NA              "closed"     "2007-01-03 00:00:00"
## 2007-04-02 00:00:00 NA              "closed"     "2007-05-01 00:00:00"
## 2007-06-01 00:00:00 NA              "closed"     "2007-07-02 00:00:00"
## 2009-08-03 00:00:00 NA              "closed"     "2009-09-01 00:00:00"
## 2009-11-02 00:00:00 NA              "closed"     "2009-12-01 00:00:00"
## 2010-09-01 00:00:00 NA              "closed"     "2010-10-01 00:00:00"
## 2010-12-01 00:00:00 NA              "closed"     "2011-01-03 00:00:00"
## 2011-09-01 00:00:00 NA              "closed"     "2011-10-03 00:00:00"
## 2011-10-03 00:00:00 NA              "closed"     "2011-11-01 00:00:00"
## 2012-08-01 00:00:00 NA              "closed"     "2012-09-04 00:00:00"
## 2012-11-01 00:00:00 NA              "closed"     "2012-12-03 00:00:00"
## 2013-11-01 00:00:00 NA              "closed"     "2013-12-02 00:00:00"
## 2013-12-02 00:00:00 NA              "closed"     "2014-01-02 00:00:00"
## 2014-04-01 00:00:00 NA              "closed"     "2014-05-01 00:00:00"
## 2014-05-01 00:00:00 NA             "closed"     "2014-06-02 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"
##                     Prefer Order.Set Txn.Fees Rule        Time.In.Force
## 2006-11-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2006-12-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2007-04-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2007-06-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2009-08-03 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2009-11-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2010-09-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2010-12-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2011-09-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2011-10-03 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2012-08-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2012-11-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2013-11-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2013-12-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2014-04-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2014-05-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2016-01-04 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2016-02-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
##
## $GRS$IEV.csv
##                     Order.Qty Order.Price Order.Type Order.Side
## 2003-12-01 00:00:00 "363"     "30.535"    "market"   "long"
## 2004-01-02 00:00:00 "all"     "32.52"     "market"   "long"
## 2004-07-01 00:00:00 "335"     "33.12"     "market"   "long"
## 2004-08-02 00:00:00 "all"     "31.97"     "market"   "long"
## 2006-05-01 00:00:00 "236"     "47.025"    "market"   "long"
## 2006-09-01 00:00:00 "all"     "48.075"    "market"   "long"
## 2006-10-02 00:00:00 "242"     "48.335"    "market"   "long"
## 2006-11-01 00:00:00 "all"     "50.57"     "market"   "long"
## 2008-12-01 00:00:00 "420"     "27.86"     "market"   "long"
## 2009-02-02 00:00:00 "all"     "25.7"      "market"   "long"
## 2011-03-01 00:00:00 "231"     "42.34"     "market"   "long"
## 2011-04-01 00:00:00 "all"     "41.93"     "market"   "long"
## 2011-05-02 00:00:00 "215"     "45.47"     "market"   "long"
## 2011-06-01 00:00:00 "all"     "43.6"      "market"   "long"
## 2011-07-01 00:00:00 "233"     "41.91"     "market"   "long"
## 2011-08-01 00:00:00 "all"     "40.56"     "market"   "long"
## 2012-11-01 00:00:00 "262"     "37.34"     "market"   "long"
## 2013-02-01 00:00:00 "all"     "41.47"     "market"   "long"
## 2013-10-01 00:00:00 "239"     "44.26"     "market"   "long"
## 2013-11-01 00:00:00 "all"     "45.65"     "market"   "long"
## 2013-12-02 00:00:00 "230"     "46.15"     "market"   "long"
## 2014-01-02 00:00:00 "all"     "46.76"     "market"   "long"
## 2015-06-01 00:00:00 "229"     "46.18"     "market"   "long"
## 2015-07-01 00:00:00 "all"     "44.3"      "market"   "long"
## 2017-05-01 00:00:00 "244"     "43.44"     "market"   "long"
## 2017-08-01 00:00:00 "all"     "45.85"     "market"   "long"
##                     Order.Threshold Order.Status Order.StatusTime
## 2003-12-01 00:00:00 NA              "closed"     "2004-01-02 00:00:00"
## 2004-01-02 00:00:00 NA              "closed"     "2004-02-02 00:00:00"
## 2004-07-01 00:00:00 NA              "closed"     "2004-08-02 00:00:00"
## 2004-08-02 00:00:00 NA              "closed"     "2004-09-01 00:00:00"
## 2006-05-01 00:00:00 NA              "closed"     "2006-06-01 00:00:00"
## 2006-09-01 00:00:00 NA              "closed"     "2006-10-02 00:00:00"
## 2006-10-02 00:00:00 NA              "closed"     "2006-11-01 00:00:00"
## 2006-11-01 00:00:00 NA              "closed"     "2006-12-01 00:00:00"
## 2008-12-01 00:00:00 NA              "closed"     "2009-01-02 00:00:00"
## 2009-02-02 00:00:00 NA              "closed"     "2009-03-02 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"
## 2011-06-01 00:00:00 NA              "closed"     "2011-07-01 00:00:00"
## 2011-07-01 00:00:00 NA              "closed"     "2011-08-01 00:00:00"
## 2011-08-01 00:00:00 NA              "closed"     "2011-09-01 00:00:00"
## 2012-11-01 00:00:00 NA              "closed"     "2012-12-03 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"
## 2013-11-01 00:00:00 NA              "closed"     "2013-12-02 00:00:00"
## 2013-12-02 00:00:00 NA              "closed"     "2014-01-02 00:00:00"
## 2014-01-02 00:00:00 NA              "closed"     "2014-02-03 00:00:00"
## 2015-06-01 00:00:00 NA              "closed"     "2015-07-01 00:00:00"
## 2015-07-01 00:00:00 NA              "closed"     "2015-08-03 00:00:00"
## 2017-05-01 00:00:00 NA              "closed"     "2017-06-01 00:00:00"
## 2017-08-01 00:00:00 NA              "closed"     "2017-09-01 00:00:00"
##                     Prefer Order.Set Txn.Fees Rule        Time.In.Force
## 2003-12-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2004-01-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2004-07-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2004-08-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2006-05-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2006-09-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2006-10-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2006-11-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2008-12-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2009-02-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2011-03-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2011-04-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2011-05-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2011-06-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2011-07-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2011-08-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2012-11-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2013-02-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2013-10-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2013-11-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2013-12-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2014-01-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2015-06-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2015-07-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2017-05-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2017-08-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
##
## $GRS$ILF.csv
##                     Order.Qty Order.Price Order.Type Order.Side
## 2004-03-01 00:00:00 "877"     "12.66"     "market"   "long"
## 2004-04-01 00:00:00 "all"     "12.55"     "market"   "long"
## 2004-08-02 00:00:00 "932"     "11.91"     "market"   "long"
## 2004-12-01 00:00:00 "all"     "15.07"     "market"   "long"
## 2005-01-03 00:00:00 "830"     "16.2"      "market"   "long"
## 2005-06-01 00:00:00 "all"     "17.42"     "market"   "long"
## 2005-07-01 00:00:00 "798"     "18.61"     "market"   "long"
## 2006-01-03 00:00:00 "all"     "24.982"    "market"   "long"
## 2006-03-01 00:00:00 "655"     "28.6"      "market"   "long"
## 2006-05-01 00:00:00 "all"     "30.61"     "market"   "long"
## 2006-09-01 00:00:00 "719"     "28.144"    "market"   "long"
## 2006-10-02 00:00:00 "all"     "28.684"    "market"   "long"
## 2006-12-01 00:00:00 "620"     "32.618"    "market"   "long"
## 2007-03-01 00:00:00 "all"     "32.394"    "market"   "long"
## 2007-06-01 00:00:00 "462"     "42.92"     "market"   "long"
## 2007-09-04 00:00:00 "all"     "42.538"    "market"   "long"
## 2007-12-03 00:00:00 "391"     "50.43"     "market"   "long"
## 2008-08-01 00:00:00 "all"     "50.39"     "market"   "long"
## 2009-02-02 00:00:00 "818"     "24.5"      "market"   "long"
## 2009-05-01 00:00:00 "all"     "29.79"     "market"   "long"
## 2009-07-01 00:00:00 "712"     "35.5"      "market"   "long"
## 2009-08-03 00:00:00 "all"     "38.87"     "market"   "long"
## 2009-11-02 00:00:00 "581"     "43.54"     "market"   "long"
## 2010-02-01 00:00:00 "all"     "43.12"     "market"   "long"
## 2012-03-01 00:00:00 "454"     "48.96"     "market"   "long"
## 2012-04-02 00:00:00 "all"     "47.47"     "market"   "long"
## 2014-05-01 00:00:00 "585"     "38.02"     "market"   "long"
## 2014-07-01 00:00:00 "all"     "38.3"      "market"   "long"
## 2014-09-02 00:00:00 "536"     "42.37"     "market"   "long"
## 2014-11-03 00:00:00 "all"     "36.18"     "market"   "long"
## 2015-07-01 00:00:00 "760"     "29.77"     "market"   "long"
## 2015-08-03 00:00:00 "all"     "26.75"     "market"   "long"
## 2016-04-01 00:00:00 "909"     "24.9"      "market"   "long"
## 2016-08-01 00:00:00 "all"     "27.71"     "market"   "long"
## 2016-09-01 00:00:00 "829"     "28.06"     "market"   "long"
## 2016-10-03 00:00:00 "all"     "28.21"     "market"   "long"
## 2016-11-01 00:00:00 "750"     "31.02"     "market"   "long"
## 2016-12-01 00:00:00 "all"     "27.73"     "market"   "long"
## 2017-03-01 00:00:00 "740"     "31.43"     "market"   "long"
## 2017-05-01 00:00:00 "all"     "31.4"      "market"   "long"
## 2017-09-01 00:00:00 "659"     "34.92"     "market"   "long"
## 2017-11-01 00:00:00 "all"     "34.02"     "market"   "long"
## 2018-02-01 00:00:00 "571"     "38.96"     "market"   "long"
## 2018-05-01 00:00:00 "all"     "36.5"      "market"   "long"
## 2018-10-01 00:00:00 "689"     "31.49"     "market"   "long"
##                     Order.Threshold Order.Status Order.StatusTime
## 2004-03-01 00:00:00 NA              "closed"     "2004-04-01 00:00:00"
## 2004-04-01 00:00:00 NA              "closed"     "2004-05-03 00:00:00"
## 2004-08-02 00:00:00 NA              "closed"     "2004-09-01 00:00:00"
## 2004-12-01 00:00:00 NA              "closed"     "2005-01-03 00:00:00"
## 2005-01-03 00:00:00 NA              "closed"     "2005-02-01 00:00:00"
## 2005-06-01 00:00:00 NA              "closed"     "2005-07-01 00:00:00"
## 2005-07-01 00:00:00 NA              "closed"     "2005-08-01 00:00:00"
## 2006-01-03 00:00:00 NA              "closed"     "2006-02-01 00:00:00"
## 2006-03-01 00:00:00 NA              "closed"     "2006-04-03 00:00:00"
## 2006-05-01 00:00:00 NA              "closed"     "2006-06-01 00:00:00"
## 2006-09-01 00:00:00 NA              "closed"     "2006-10-02 00:00:00"
## 2006-10-02 00:00:00 NA              "closed"     "2006-11-01 00:00:00"
## 2006-12-01 00:00:00 NA              "closed"     "2007-01-03 00:00:00"
## 2007-03-01 00:00:00 NA              "closed"     "2007-04-02 00:00:00"
## 2007-06-01 00:00:00 NA              "closed"     "2007-07-02 00:00:00"
## 2007-09-04 00:00:00 NA              "closed"     "2007-10-01 00:00:00"
## 2007-12-03 00:00:00 NA              "closed"     "2008-01-02 00:00:00"
## 2008-08-01 00:00:00 NA              "closed"     "2008-09-02 00:00:00"
## 2009-02-02 00:00:00 NA              "closed"     "2009-03-02 00:00:00"
## 2009-05-01 00:00:00 NA              "closed"     "2009-06-01 00:00:00"
## 2009-07-01 00:00:00 NA              "closed"     "2009-08-03 00:00:00"
## 2009-08-03 00:00:00 NA              "closed"     "2009-09-01 00:00:00"
## 2009-11-02 00:00:00 NA              "closed"     "2009-12-01 00:00:00"
## 2010-02-01 00:00:00 NA              "closed"     "2010-03-01 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"
## 2014-05-01 00:00:00 NA              "closed"     "2014-06-02 00:00:00"
## 2014-07-01 00:00:00 NA              "closed"     "2014-08-01 00:00:00"
## 2014-09-02 00:00:00 NA              "closed"     "2014-10-01 00:00:00"
## 2014-11-03 00:00:00 NA              "closed"     "2014-12-01 00:00:00"
## 2015-07-01 00:00:00 NA              "closed"     "2015-08-03 00:00:00"
## 2015-08-03 00:00:00 NA              "closed"     "2015-09-01 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"
## 2016-10-03 00:00:00 NA              "closed"     "2016-11-01 00:00:00"
## 2016-11-01 00:00:00 NA              "closed"     "2016-12-01 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"
## 2017-05-01 00:00:00 NA              "closed"     "2017-06-01 00:00:00"
## 2017-09-01 00:00:00 NA              "closed"     "2017-10-02 00:00:00"
## 2017-11-01 00:00:00 NA              "closed"     "2017-12-01 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
## 2004-03-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2004-04-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2004-08-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2004-12-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2005-01-03 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2005-06-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2005-07-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2006-01-03 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2006-03-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2006-05-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2006-09-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2006-10-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2006-12-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2007-03-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2007-06-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2007-09-04 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2007-12-03 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2008-08-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2009-02-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2009-05-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2009-07-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2009-08-03 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2009-11-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2010-02-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2012-03-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2012-04-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2014-05-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2014-07-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2014-09-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2014-11-03 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2015-07-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2015-08-03 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2016-04-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2016-08-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2016-09-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2016-10-03 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2016-11-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2016-12-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2017-03-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2017-05-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2017-09-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2017-11-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2018-02-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2018-05-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2018-10-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
##
## $GRS$MDY.csv
##                     Order.Qty Order.Price Order.Type Order.Side
## 2004-05-03 00:00:00 "103"     "106.95"    "market"   "long"
## 2004-07-01 00:00:00 "all"     "111.1"     "market"   "long"
## 2005-06-01 00:00:00 "92"      "122.79"    "market"   "long"
## 2005-07-01 00:00:00 "all"     "125.46"    "market"   "long"
## 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
## 2004-05-03 00:00:00 NA              "closed"     "2004-06-01 00:00:00"
## 2004-07-01 00:00:00 NA              "closed"     "2004-08-02 00:00:00"
## 2005-06-01 00:00:00 NA              "closed"     "2005-07-01 00:00:00"
## 2005-07-01 00:00:00 NA              "closed"     "2005-08-01 00:00:00"
## 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
## 2004-05-03 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2004-07-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2005-06-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2005-07-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2007-03-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2007-04-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2008-08-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2008-12-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2010-02-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2010-08-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2010-12-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2011-03-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2011-04-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2011-05-02 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2011-10-03 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2012-03-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2012-04-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2012-08-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2013-02-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2013-10-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2014-01-02 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2014-04-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2014-11-03 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2015-05-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2015-08-03 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2016-01-04 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2016-02-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2016-04-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2016-08-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2016-09-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2016-12-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2017-03-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2017-12-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2018-02-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
## 2018-05-01 00:00:00 "Open" "ocolong" "txnc"   "enterLONG" ""
## 2018-10-01 00:00:00 "Open" "ocolong" "txnc"   "exitLONG"  ""
##
##
## attr(,"class")
## [1] "order_book"
```