Assignment 1

Philipp Epstein

10/14/2018

Introduction

This report is written for the 1st assignment of the Financial Trading Systems course 2018 at the Bond University. During the first weeks we learned how to use the Blotter Framework in R to implement a trading strategy, backtest the strategy on a set of instruments (stock, futures, etc.) and visualize meaningful results which give insights about the strategy. Moreover, we learned how to compare the strategy to another alternative (e.g. a buy and hold strategy). The trading idea, implementation and analysis and results are summarized in the following report.

Trading Idea

Smash Day by Larry Williams

The trading idea which is used for this report is based on the Smash Day strategy introduced by Larry Williams in his book "Long Term Secrets to short term trading" 2012. The basic theoretical foundation is set by the concept of autocorrelation. There are two autocorrelation principles which can be empirically observed. The positive autocorrelation which describes the fact that if we observe an upward movement it is more likely to observe an upward movement in the future. The same is valid for a negative trend. However, the negative autocorreleation can be observed on short term time periods. We can expect a price to bounce up again after a sharp down movement and vice versa. Larry William's Smash Day pattern is based on the observation of this autocorrelation behaviour. A Smash Day pattern is initialized when we observe a close which is lower than the previous day's low. This looks like a breakout to the downside. If the very next day the price moves opposite the Smash Day and trades above the high of the Smash Day this is according to Williams a buy signal. Williams explains this buy signal with a feeling of hurt felt by the public as a result of an unfulfilled breakout. The public would want to get back on track and the price responds with a reversal. The exact opposite is true for a Sell setup. The Smash Day is set up when we observe a close above the previous day's high. The sell signal is initiated when the price reverses immidiately the next day and falls below the smash day's low.

Own variation of the Smash Day trading strategy

In addition to the Smash Day pattern, I have incorporated another rule for the buy signal and 2 additional components for the sell signal. I want to make sure that I only enter long when we are observing a longlasting upward trend. In order to do this buy orders are only performed when the current price is above the exponential moving average of the last 200 days. After that the Smash Day pattern is checked. For the sell I incorporated two "harder" conditions before checking whether we observe a Smash Day (sell). The first is that a maximum holding period can be set in order to make sure that a position is not open for too long. The second sell case is when the price falls below the 200 day exponential moving average the position is liquidated. If both is not the case a check for the smash day pattern is performed. If we observe it, we sell the current position.

These variations are incorporated to reduce the risk of falling prices.

Implementation of the trading idea

Part A: Initialization

Step 1: General Setup

The environment has to be cleared in order to ensure a reproducable setup. Moreover, the blotter library and the INF361Course library are loaded.

```
# Clear Environment
rm(list=ls())

# Loading libraries
library(blotter)
library(INFT361Course)
```

Step 2: Setting the Variables

The variables set in the next section can be adjusted to test the strategy with different parameters. The critical variables which should be adjusted to test the strategy are: * daterange * emaPeriod * maxHoldingPeriod * instrumentlist * BuyHoldInstrument

```
# Set values:
startCapital <- 1e+6
transactionCost <- -20
daterange <- '2015::2018'
emaPeriod <- 400
maxHoldingPeriod <- 4

InstrumentDirectory <- "~/Desktop/R/DownloadedData/"
instrumentlist <- c("BMW.csv","DAI.csv", "DTE.csv", "FME.csv", "FRE.csv")
BuyHoldDirectory <- "~/Desktop/R/DownloadedData/"
BuyHoldInstrument <- "DAXEX.csv"

currency("EUR")
Sys.setenv(TZ="UTC")
initdate <- '1999-12-31'
startdate <- '2000-01-01'
enddate <- '2018-12-31'
portfolioname <- "Smash Day"
accountname <- portfolioname
```

Step 3: Presetup for plotting graphs

Some setup for plotting the graphs in the visualization is performed here to ensure that the Theme is available later, when it is needed and used.

```
# Settings for graph
myTheme <- chart_theme()
myTheme$col$up.col <- 'lightblue'
myTheme$col$dn.col <- 'brown'
myTheme$col$dn.border <- 'lightgray'
myTheme$col$up.border <- 'lightgray'

# Concatenate string for EMA with input parameter
addEMAString <- paste("add_EMA(n=",emaPeriod,")",sep = "")
```

Step 4: Initializing the portfolio

The portfolio takes the instrumentlist which includes all the stocks we defined in Step 2.

```
# Clear portfolio and Account
suppressWarnings(rm("account.Smash Day","portfolio.Smash Day",
                    "account.buyhold","portfolio.buyhold",
                    pos=.blotter))

# Initialize Portfolio and Account
initPortf(portfolioname,
          instrumentlist,
          initDate=initdate,
          currency="EUR")

initAcct(accountname,
         portfolios=portfolioname,
         initDate=initdate,
         initEq=startCapital,
         currency="EUR")
```

Part B: Bar by bar processing

Step 1: Go through the data bar by bar

In this step the bar by bar processing is implemented. Thus, a for loop is implemented to loop through the instrumentlist. For each instrument in the list the data is loaded and the exponential moving average is added to the data. Furthermore, another for loop is used to loop through the dates bar by bar. In this step the defined strategy is applied.

```
for (instrument in instrumentlist) {
    LoadCourseFile(InstrumentDirectory, instrument, debugme = TRUE, dates = daterange)

    # Initialize the instrument
    stock(instrument, currency = "EUR")

    # Load the XTS file
    symbol <- get(instrument)

    # Calculate the Exponential Moving Average
    ema <- EMA(symbol$Close, n=emaPeriod)

    # Merge the xts file with the Exponential Moving Average
    symbol <- merge(symbol,ema)
    assign(instrument,symbol)
  # Starting to go bar by bar through using a "for loop"
  for (i in (emaPeriod + 1):(nrow(symbol) - 1)) {
    # Dates
    CurrentDate <- time(symbol[i])
    TomorrowDate <- time(symbol[i + 1])

    # Today's variables
    CloseToday <- as.numeric(symbol[i, "Close"])
    EMA_today <- as.numeric(symbol[i, "EMA"])
    LowToday <- as.numeric(symbol[i, "Low"])
    HighToday <- as.numeric(symbol[i, "High"])
```

```
# Yesterday's variables
LowYesterday <- as.numeric(symbol[i - 1, "Low"])
HighYesterday <- as.numeric(symbol[i - 1, "High"])

# Tomorrow's variables
OpenTomorrow <- as.numeric(symbol[i + 1, "Open"])
LowTomorrow <- as.numeric(symbol[i + 1, "Low"])
HighTomorrow <- as.numeric(symbol[i + 1, "High"])

# Config
Equity <- getEndEq(accountname, CurrentDate)
Position <-
  getPosQty(portfolioname, Symbol = instrument, Date = CurrentDate)

# Check whether we have a position
if (Position == 0) {
  # Start checking BUY rules

  # Check whether we have a Smash Day (Buy).
  # Smash Day (Buy) is when Todays Close is below Yesterdays Low.
  if (CloseToday < LowYesterday) {
    # Smash Day (Buy)

    #Check whether todays close is above today's EMA
    if (CloseToday > EMA_today) {

      # BUY RULE: If today was a smash day place a STOP BUY order
      # at todays high price for the next day.
      # (Buy tomorrow for 'price >= todays high')

      ######
      # Simulate STOP BUY order:
      ######

      # Option 1 to check: Check whether the open price tomorrow
      # is above today's high and add the transaction tomorrow at
      # tomorrow's open price.

      # Option 2 to check: Check whether today's high was lower
      # than tomorrows high and add the transaction tomorrow
      # at today's high price

      # Check Option 1
      if (OpenTomorrow > HighToday) {
        # Don't trade at the day before the last day
        if (CurrentDate != time(symbol[nrow(symbol) - 1])) {
          # Calculate the buy quantity
          BuyQuantity <- as.numeric(trunc(Equity / OpenTomorrow))
          # Add transaction
          addTxn(
            portfolioname,
            Symbol = instrument,
            TxnDate = TomorrowDate ,
```

```
            TxnPrice = OpenTomorrow,
            TxnQty = BuyQuantity,
            TxnFees = transactionCost
          )
          # Store the bar at which we placed the transaction
          BuyBar <- i
        }

      } else {
        # Check Option 2
        if (HighToday < HighTomorrow) {
          # Don't trade at the day before the last day
          if (CurrentDate != time(symbol[nrow(symbol) - 1])) {
            # Calculate the buy quantity
            BuyQuantity <- as.numeric(trunc(Equity / HighToday))
            # Add transaction
            addTxn(
              portfolioname,
              Symbol = instrument,
              TxnDate = TomorrowDate ,
              TxnPrice = HighToday,
              TxnQty = BuyQuantity,
              TxnFees = transactionCost
            )
            # Store the bar at which we placed the transaction
            BuyBar <- i
          }
        }
      }
    }
  }
} else {
  # We already have a position

  # Check the sell rules in the following order and sell at the
  # first condition which is satisfied.

  #####
  # SELL rules:
  #####

  # Rule 1: Sell if we hold the position longer than the specified
  # maximum holding period

  # Rule 2: Sell at tomorrow's opening price if the close price
  # today falls below the EMA

  # Rule 3: Sell if we meet the Smash Day (Sell) requirements.
  # Today's close must be higher than yesterday's high

  # Rule 4: If no sell rule can be applied and we reach the
  # second last day. Sell at the last day.
```

```
    # Check Rule 1:
    if ((i - BuyBar) > maxHoldingPeriod) {
      # Place the sell transaction at todays close price
      addTxn(
        portfolioname,
        Symbol = instrument,
        TxnDate = CurrentDate,
        TxnPrice = as.numeric(symbol[i, "Close"]),
        TxnQty = -Position,
        TxnFees = transactionCost
      )

    } else {
      # Check Rule 2:
      if (as.numeric(symbol[i, "Close"]) < EMA_today) {
        # Place the sell transaction at tomorrow's open price
        addTxn(
          portfolioname,
          Symbol = instrument,
          TxnDate = time(symbol[i + 1]),
          TxnPrice = OpenTomorrow,
          TxnQty = -Position,
          TxnFees = transactionCost
        )

      } else {
        # Check Rule 3:

        # Sell Rule 3: If today is a Smash Day (Sell) place an order tomorrow at todays
        # low price.

        # Simulate this behaviour:

        # Option 1 to check: Check whether the open price tomorrow is below today's
        # low and add the transaction tomorrow at tomorrow's open price.

        # Option 2 to check: Check whether today's low was larger than tomorrow's
        # low and add the transaction tomorrow at today's low price.

        # Check for Smash Day (Sell)
        if (CloseToday > HighYesterday) {
          # Check for Option 1
          if (OpenTomorrow < LowToday) {
            # Add Sell transaction tomorrow at tomorrow's open price
            addTxn(
              portfolioname,
              Symbol = instrument,
              TxnDate = time(symbol[i + 1]),
              TxnPrice = OpenTomorrow,
              TxnQty = -Position,
              TxnFees = transactionCost
            )
```

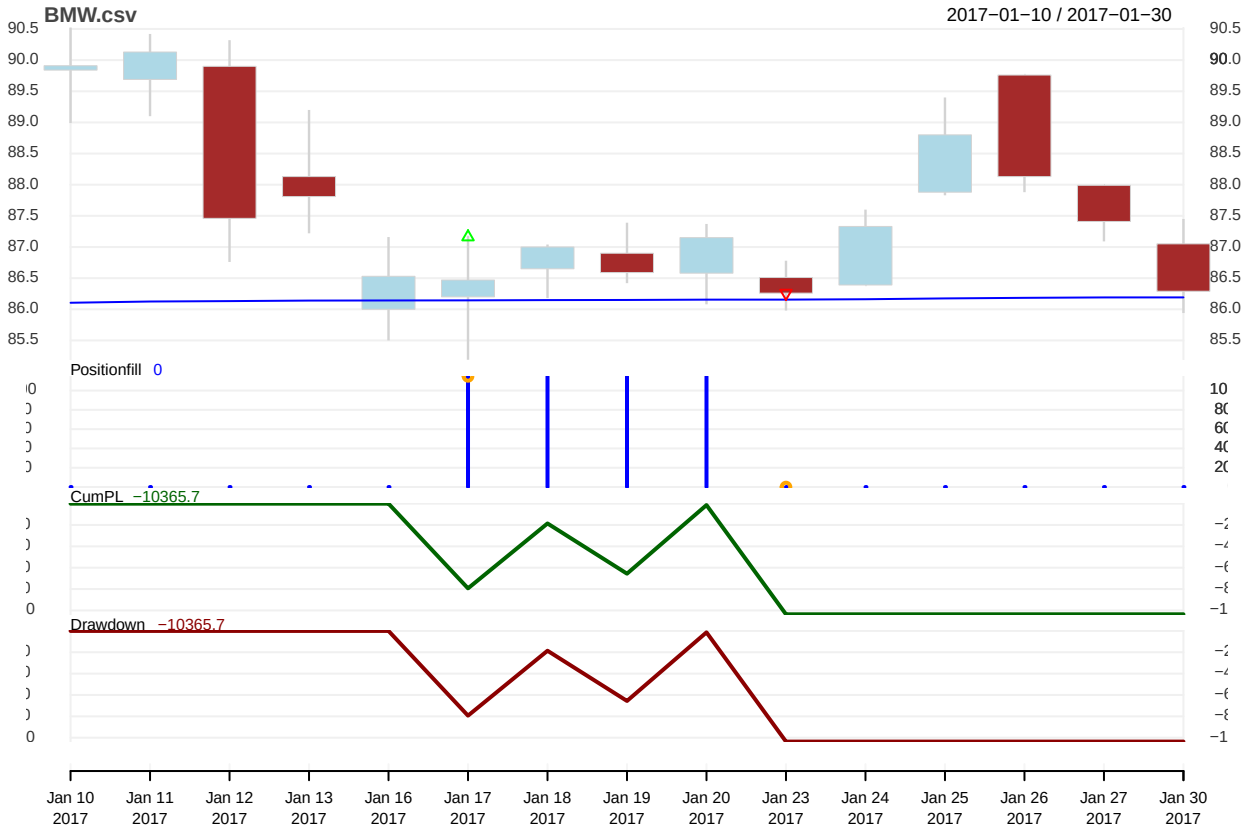
```
            } else {
              # Check for Option 2
              if (LowToday > LowTomorrow) {
                # Add Sell transaction tomorrow at today's low price
                addTxn(
                  portfolioname,
                  Symbol = instrument,
                  TxnDate = time(symbol[i + 1]),
                  TxnPrice = LowToday,
                  TxnQty = -Position,
                  TxnFees = transactionCost
                )
              }
            }
          } else {
            # Check Rule 4
            if (i == nrow(symbol) - 1) {
              # Add Sell transaction for the last day at the close price
              addTxn(
                portfolioname,
                Symbol = instrument,
                TxnDate = time(symbol[i + 1]),
                TxnPrice = as.numeric(symbol[i, "Close"]),
                TxnQty = -Position,
                TxnFees = transactionCost
              )
            }
          }
        }
      }
    }

    updatePortf(portfolioname, Symbols = instrument, Dates = CurrentDate)
    updateAcct(accountname, Dates = CurrentDate)
    updateEndEq(accountname, CurrentDate)

  } # End Bar-by-bar processing
} # End for loop for multiple instruments
```

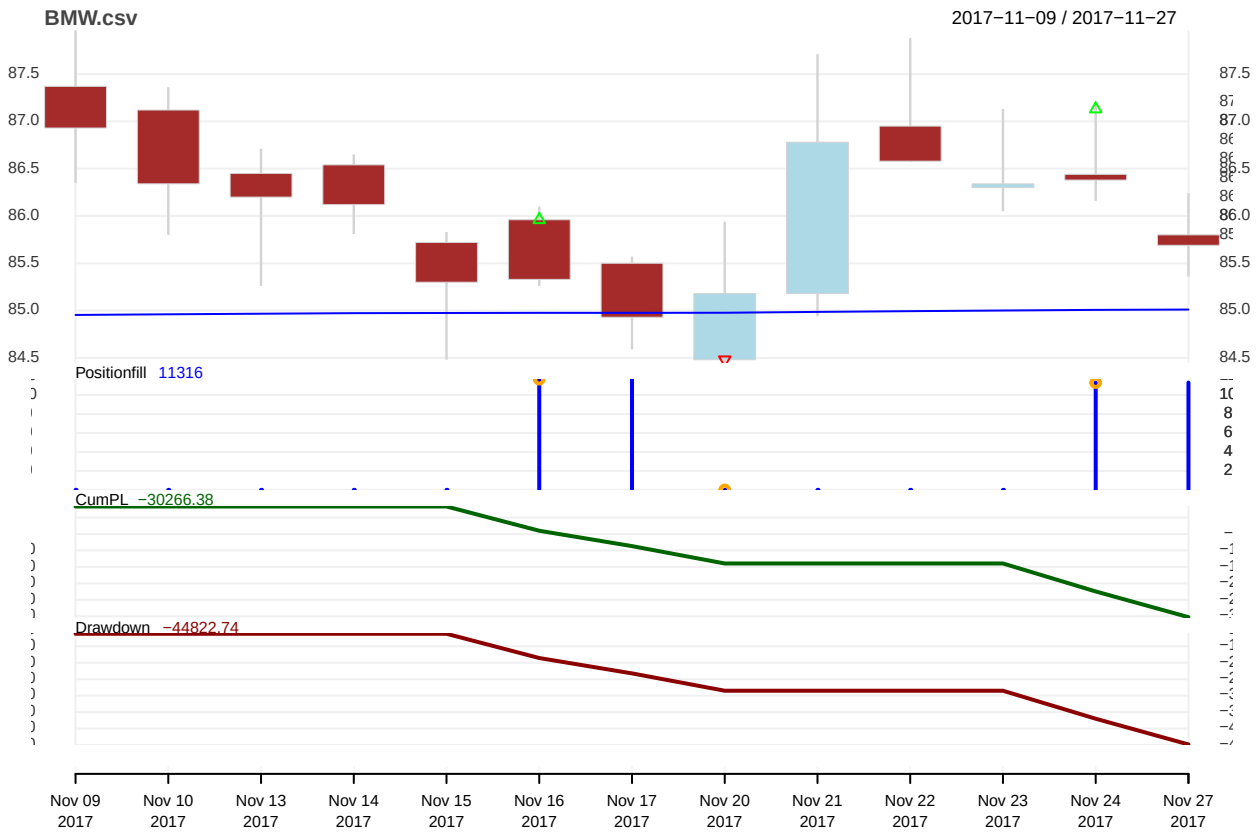
Step 2: System Check

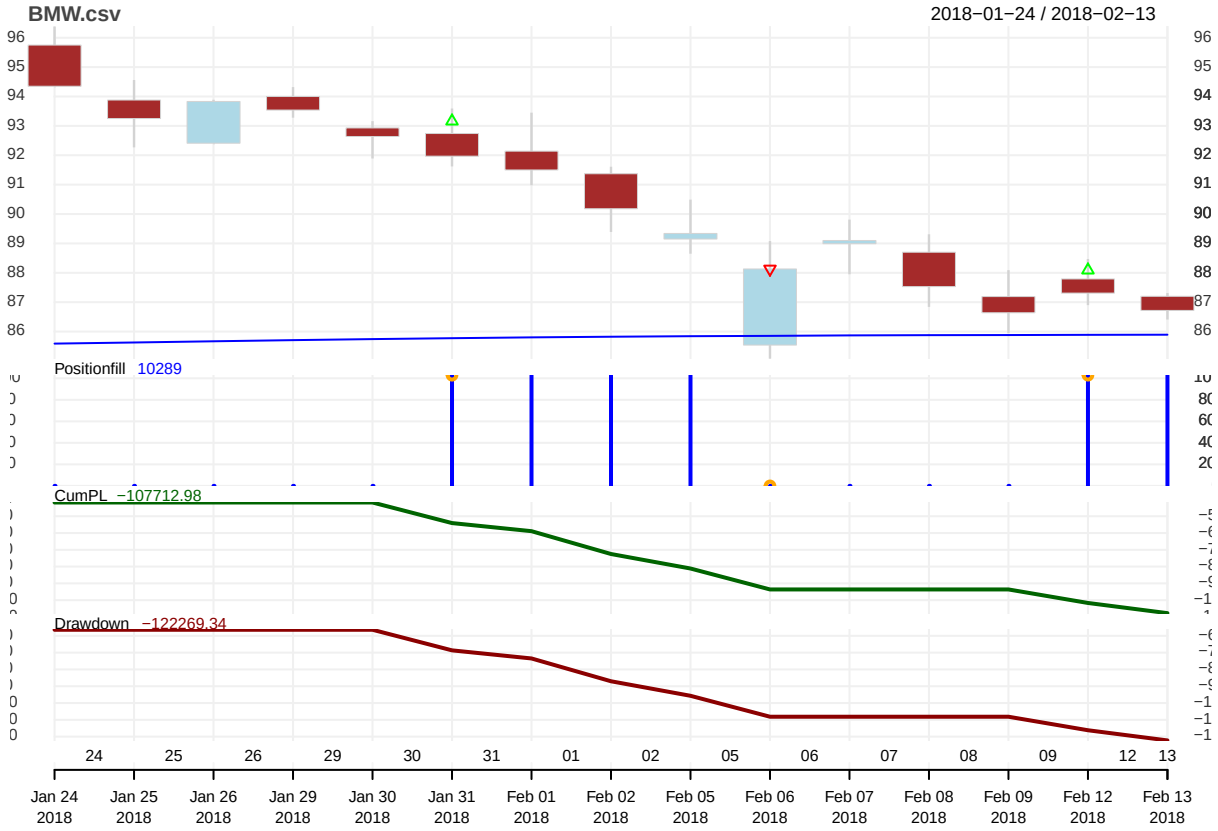
In order to make sure that the system works as designed the plots of some choosen transactions are printed in the following. The plots for the first, third and the 5th last trasnaction are plotted for every instrument in the instrumentlist. The plots can be checked manually and by this it can be ensured that the transactions were performed as expected.

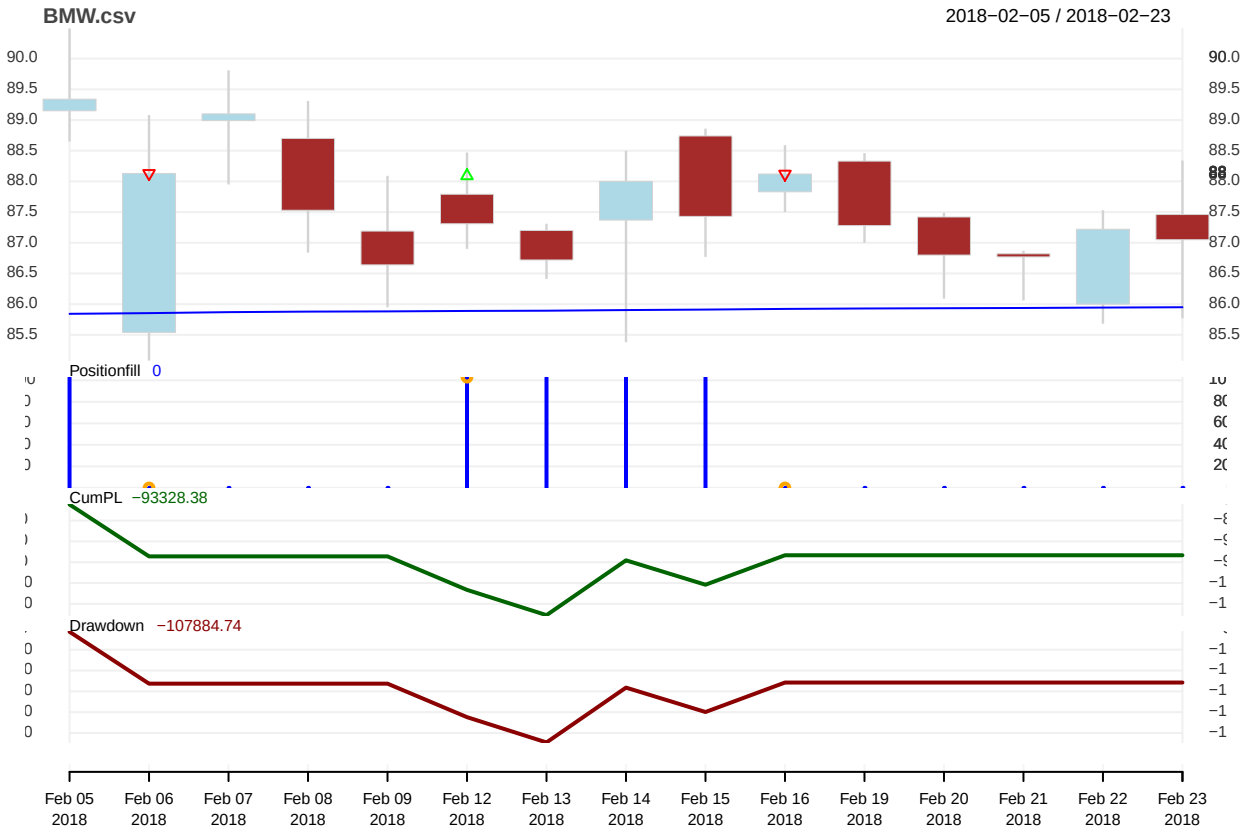
```
# Loop through all instruments in the instrumentlist
for (instrument in instrumentlist){
  rm(daterange_check)
  daterange_check <- c()
  transactionsInstrument <- getTxns(Portfolio=portfolioname,Symbol=instrument)

  # Create a list of transactions to check
  for (i in
```

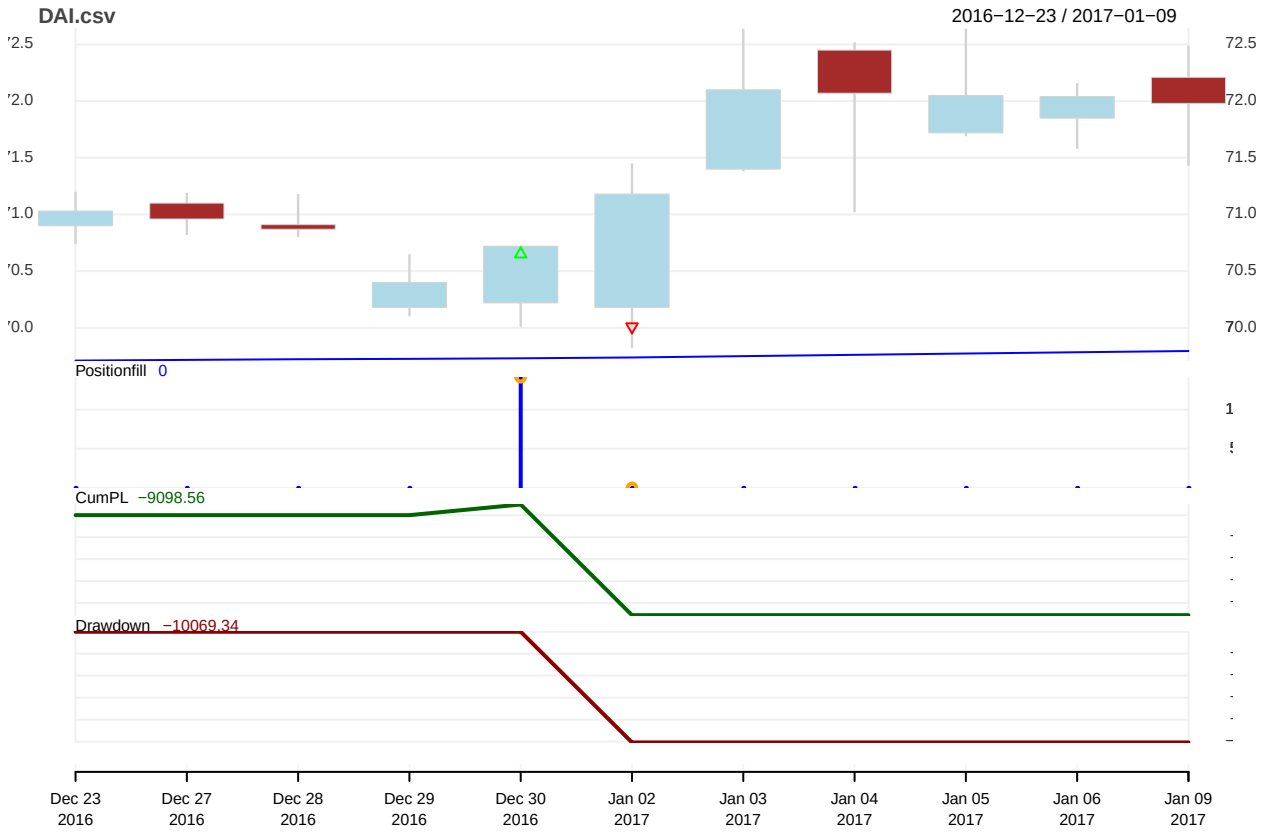
```
      c(2,6, (nrow(transactionsInstrument)-7),(nrow(transactionsInstrument)-5))) {
    from <- as.Date(index(transactionsInstrument[i,1]))-7
    to <- as.Date(index(transactionsInstrument[i+1,1]))+7
    daterange_check <- c(daterange_check, paste(from, "::", to, sep = ""))
  }
  # Plot the transactions and check them manually
  for (daterange_check_i in daterange_check){
    print(chart.Posn(portfolioname,
                     Symbol=instrument,
                     type='candlesticks',
                     theme=myTheme,
                     subset=daterange_check_i,
                     TA=addEMAString))
  }
}
```

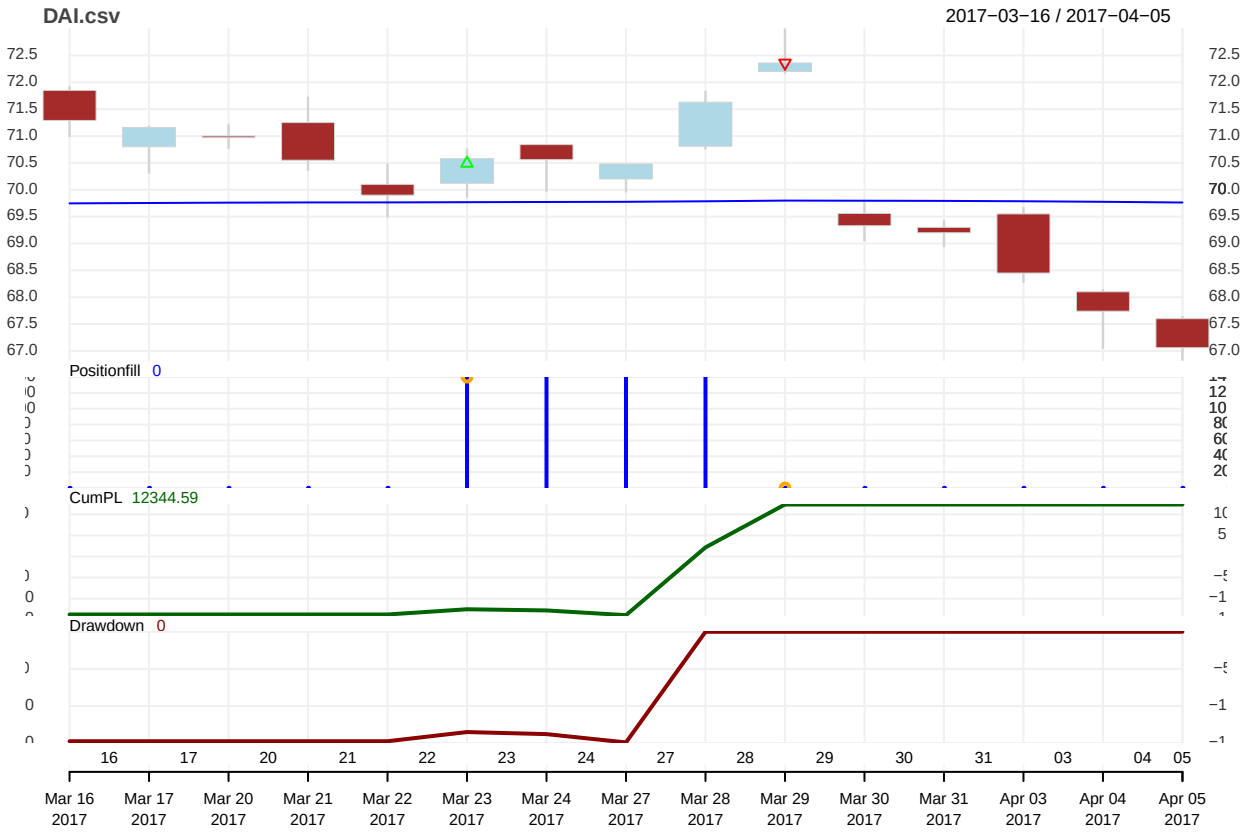


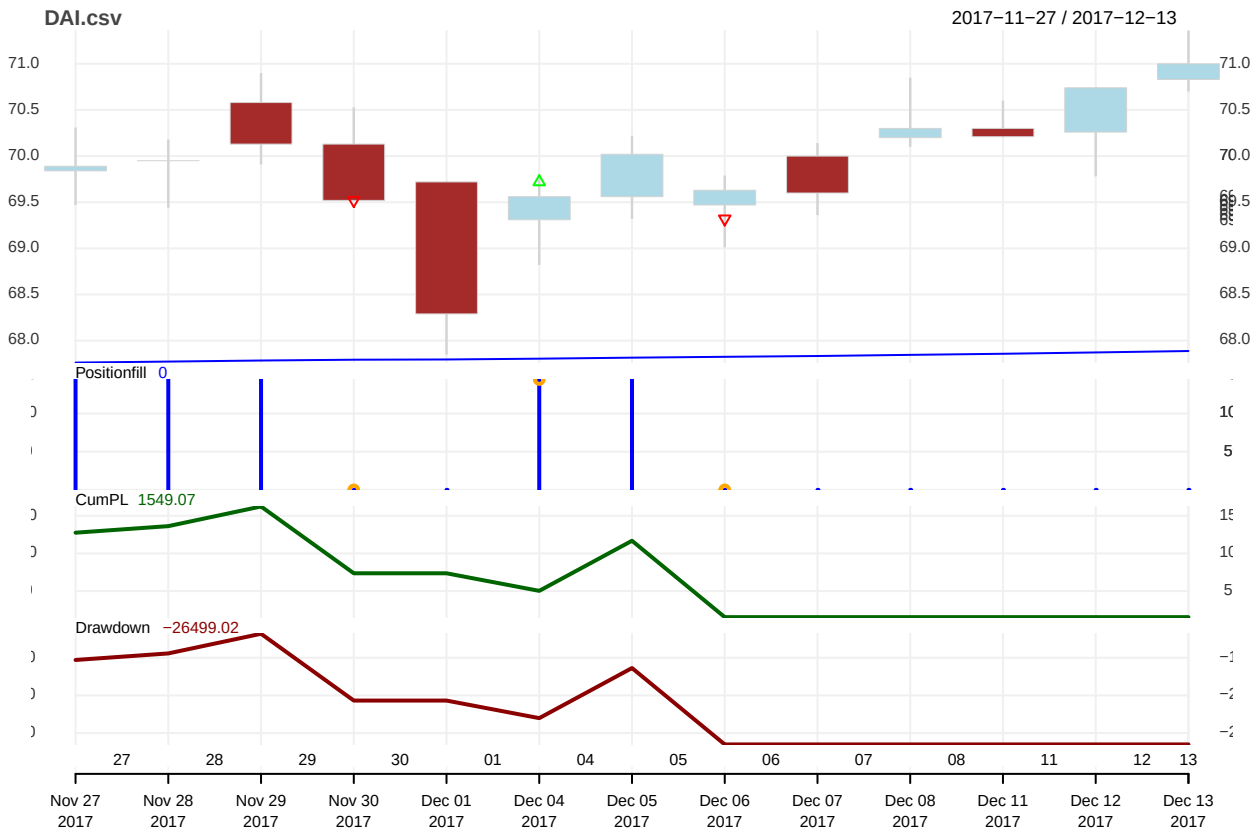

BMW.csv
2017−11−09 / 2017−11−27
87.5
87.0
86.5
86.0
85.5
85.0
84.5
Positionfill 11316
CumPL −30266.38
Drawdown −44822.74
Nov 09 2017
Nov 10 2017
Nov 13 2017
Nov 14 2017
Nov 15 2017
Nov 16 2017
Nov 17 2017
Nov 20 2017
Nov 21 2017
Nov 22 2017
Nov 23 2017
Nov 24 2017
Nov 27 2017

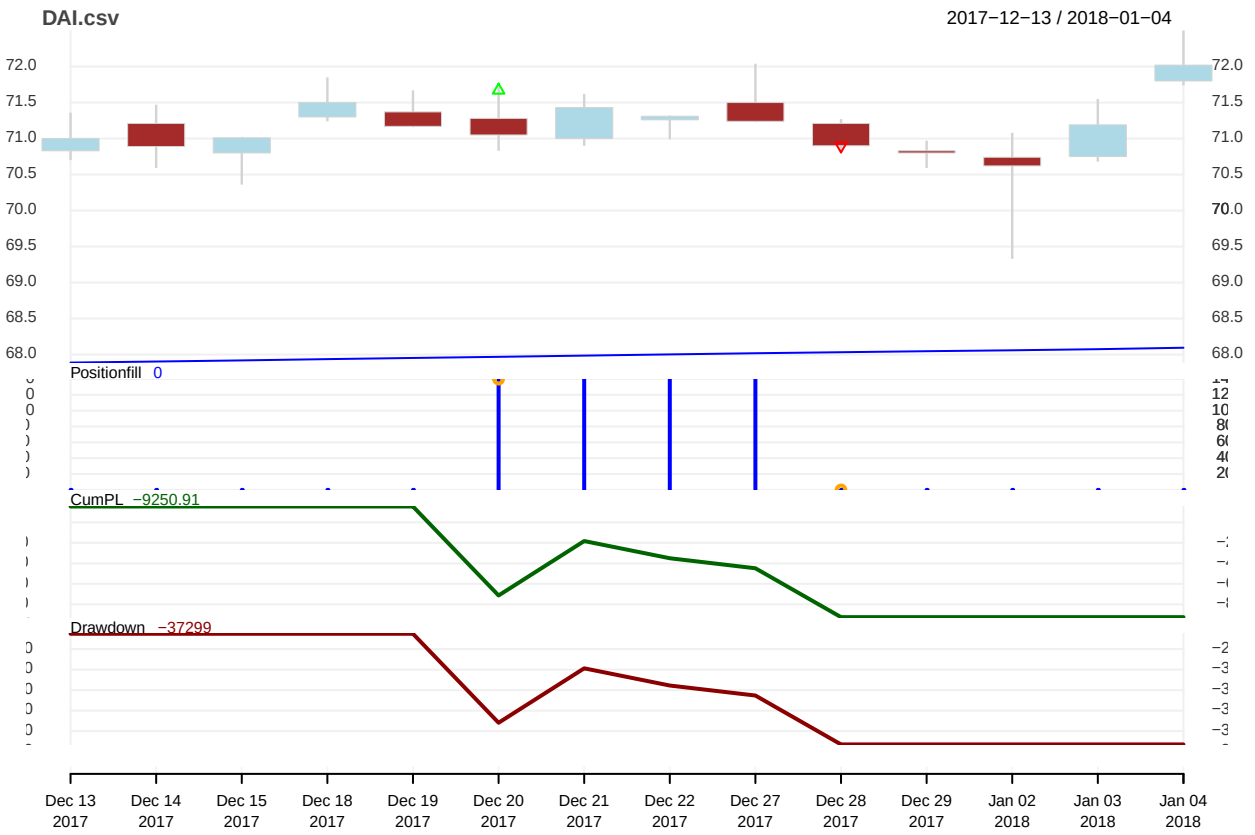

BMW.csv
2018−01−24 / 2018−02−13
Positionfill 10289
CumPL −107712.98
Drawdown −122269.34
24 25 26 29 30 31 01 02 05 06 07 08 09 12 13
Jan 24 2018
Jan 25 2018
Jan 26 2018
Jan 29 2018
Jan 30 2018
Jan 31 2018
Feb 01 2018
Feb 02 2018
Feb 05 2018
Feb 06 2018
Feb 07 2018
Feb 08 2018
Feb 09 2018
Feb 12 2018
Feb 13 2018

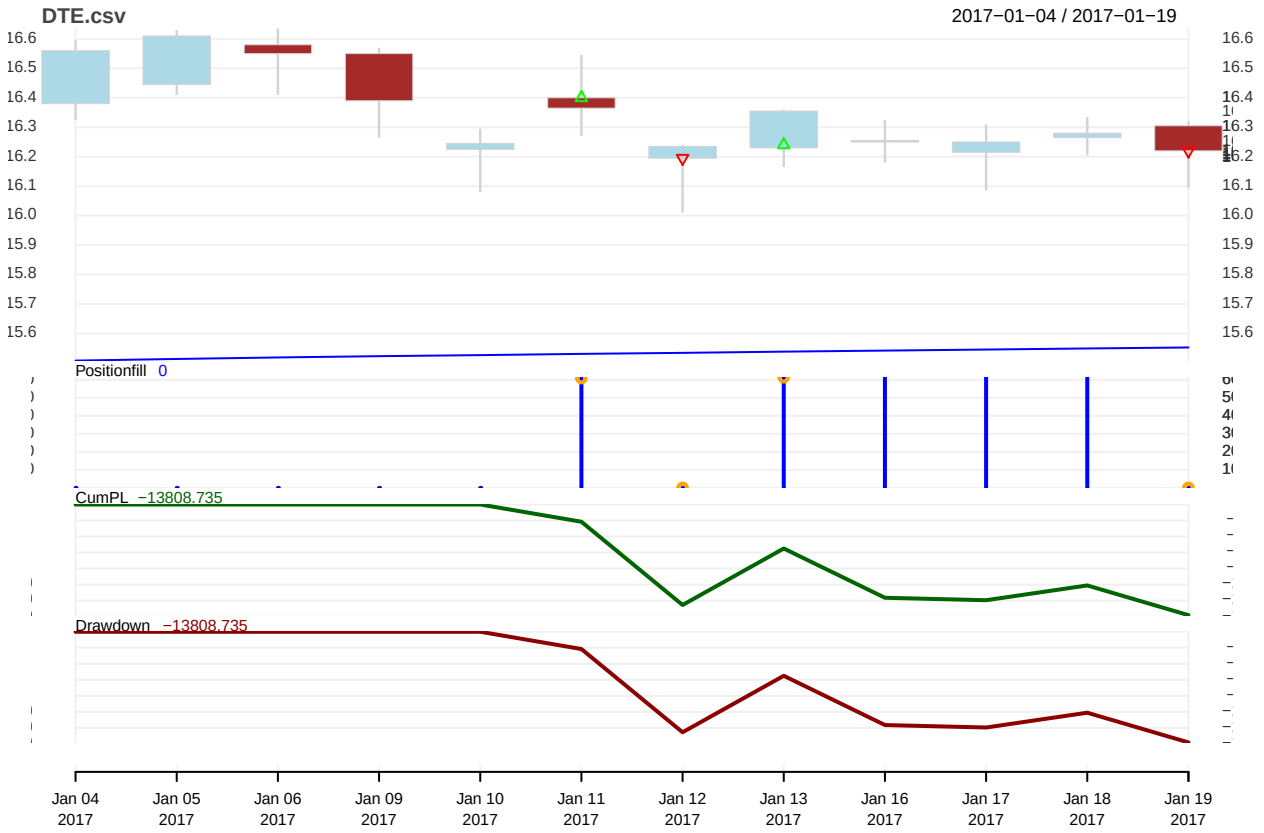

BMW.csv
2018−02−05 / 2018−02−23
90.0
89.5
89.0
88.5
88.0
87.5
87.0
86.5
86.0
85.5
Positionfill 0
CumPL −93328.38
Drawdown −107884.74
Feb 05 2018
Feb 06 2018
Feb 07 2018
Feb 08 2018
Feb 09 2018
Feb 12 2018
Feb 13 2018
Feb 14 2018
Feb 15 2018
Feb 16 2018
Feb 19 2018
Feb 20 2018
Feb 21 2018
Feb 22 2018
Feb 23 2018

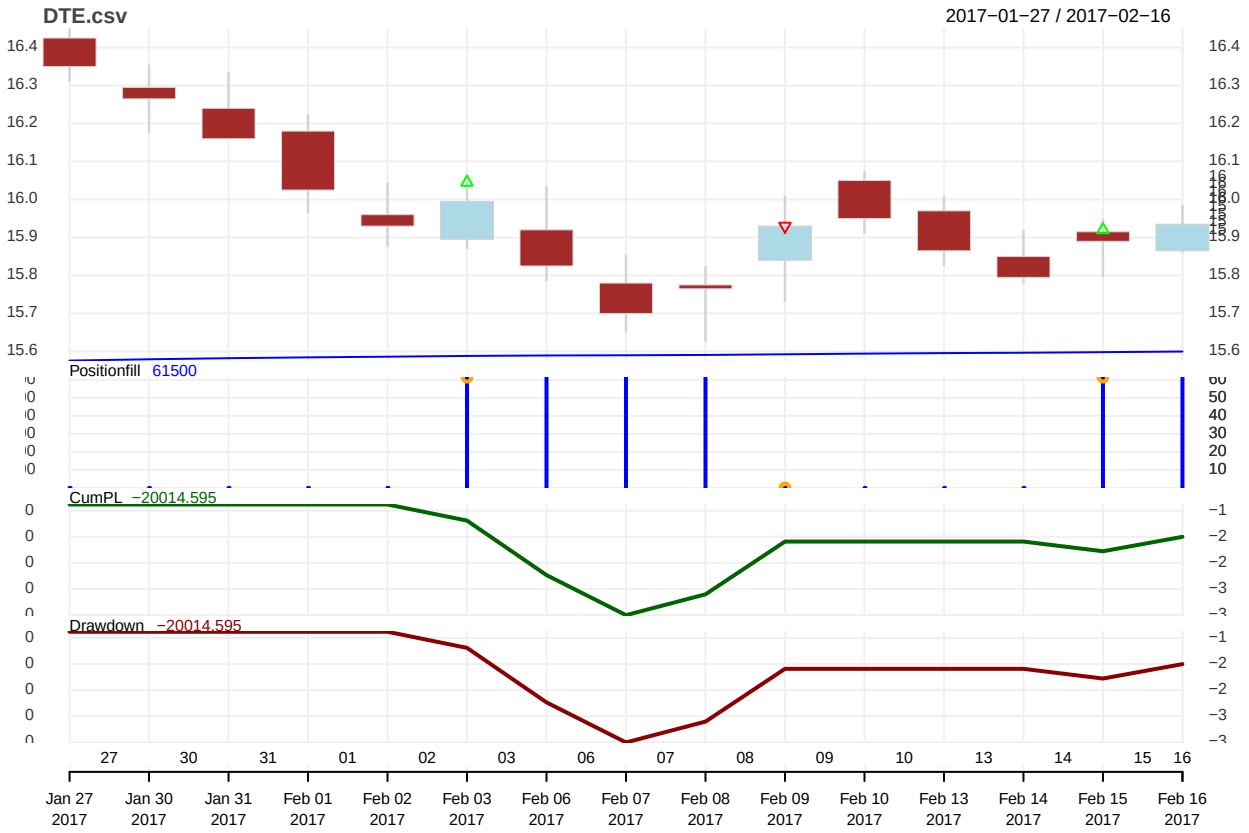

DAI.csv
2016−12−23 / 2017−01−09
72.5
72.0
71.5
71.0
70.5
70.0
Positionfill 0
CumPL −9098.56
Drawdown −10069.34
Dec 23 2016
Dec 27 2016
Dec 28 2016
Dec 29 2016
Dec 30 2016
Jan 02 2017
Jan 03 2017
Jan 04 2017
Jan 05 2017
Jan 06 2017
Jan 09 2017

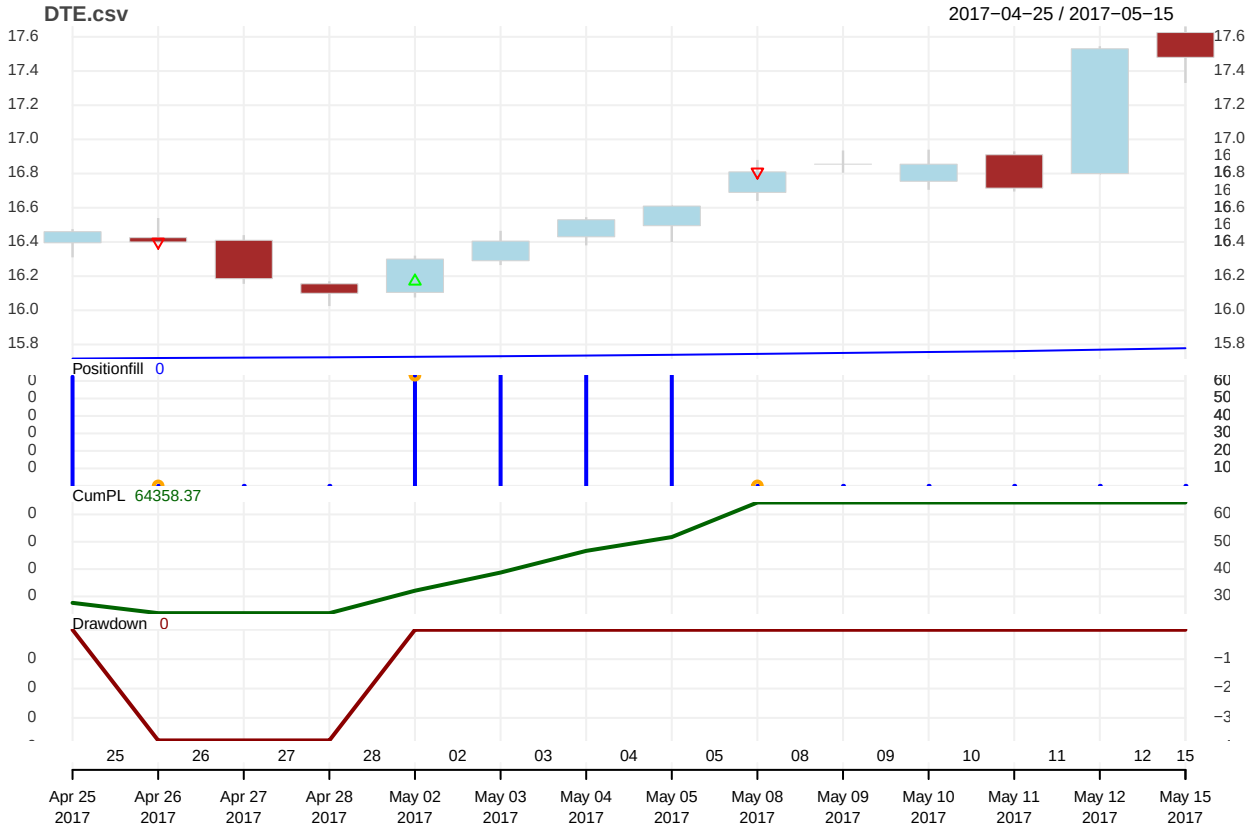

DAI.csv
2017−03−16 / 2017−04−05
72.5
72.0
71.5
71.0
70.5
70.0
69.5
69.0
68.5
68.0
67.5
67.0
Positionfill 0
CumPL 12344.59
Drawdown 0
16
17
20
21
22
23
24
27
28
29
30
31
03
04
05
Mar 16 2017
Mar 17 2017
Mar 20 2017
Mar 21 2017
Mar 22 2017
Mar 23 2017
Mar 24 2017
Mar 27 2017
Mar 28 2017
Mar 29 2017
Mar 30 2017
Mar 31 2017
Apr 03 2017
Apr 04 2017
Apr 05 2017

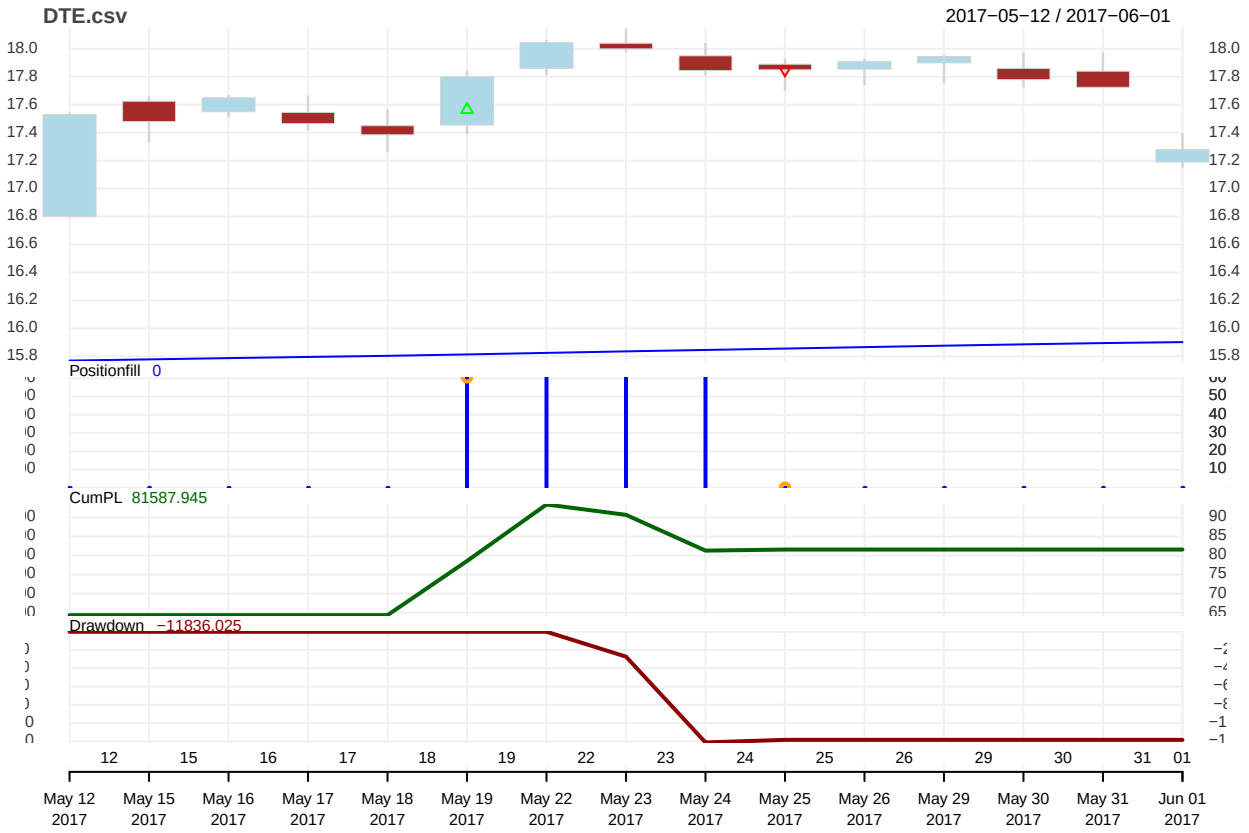

DAI.csv
2017−11−27 / 2017−12−13
71.0
70.5
70.0
69.5
69.0
68.5
68.0
Positionfill 0
CumPL 1549.07
Drawdown −26499.02
27
28
29
30
01
04
05
06
07
08
11
12
13
Nov 27 2017
Nov 28 2017
Nov 29 2017
Nov 30 2017
Dec 01 2017
Dec 04 2017
Dec 05 2017
Dec 06 2017
Dec 07 2017
Dec 08 2017
Dec 11 2017
Dec 12 2017
Dec 13 2017

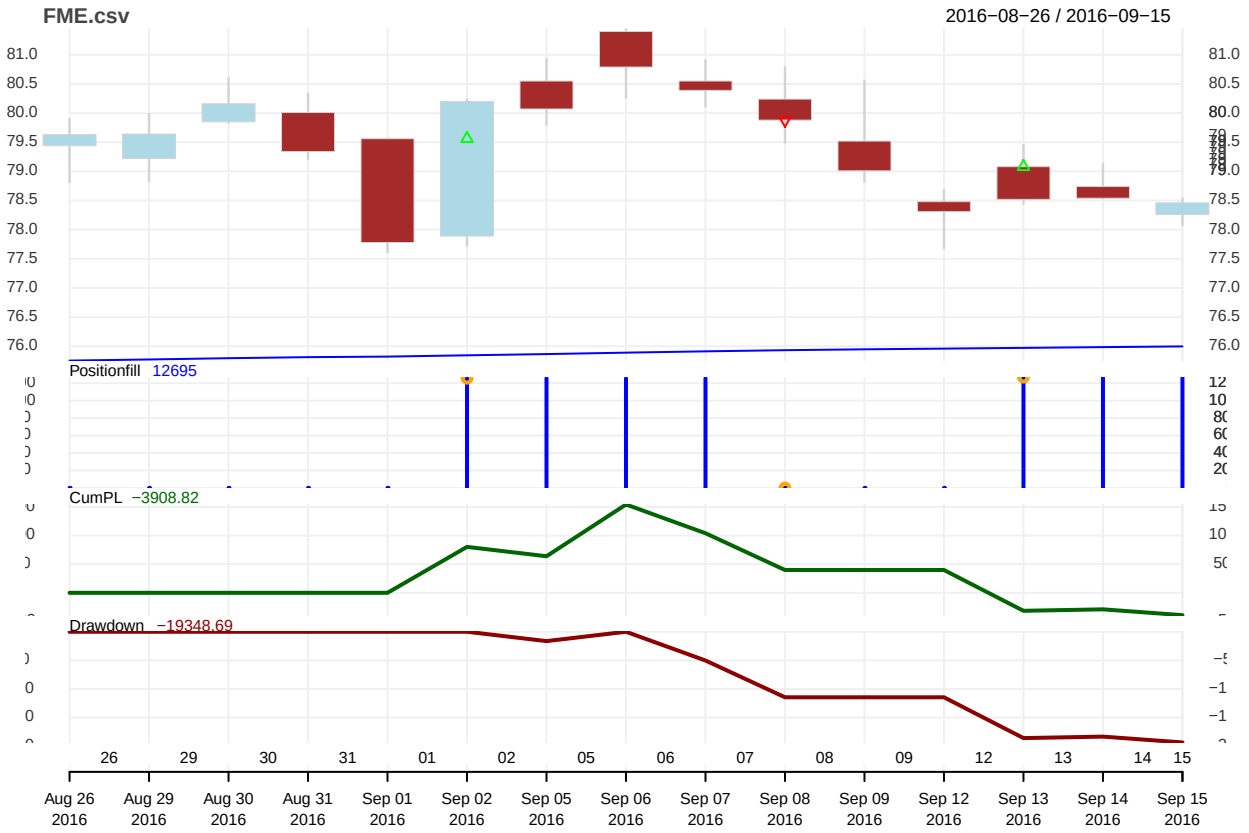

DAI.csv
2017−12−13 / 2018−01−04
72.0
71.5
71.0
70.5
70.0
69.5
69.0
68.5
68.0
Positionfill 0
CumPL −9250.91
Drawdown −37299
Dec 13 2017
Dec 14 2017
Dec 15 2017
Dec 18 2017
Dec 19 2017
Dec 20 2017
Dec 21 2017
Dec 22 2017
Dec 27 2017
Dec 28 2017
Dec 29 2017
Jan 02 2018
Jan 03 2018
Jan 04 2018

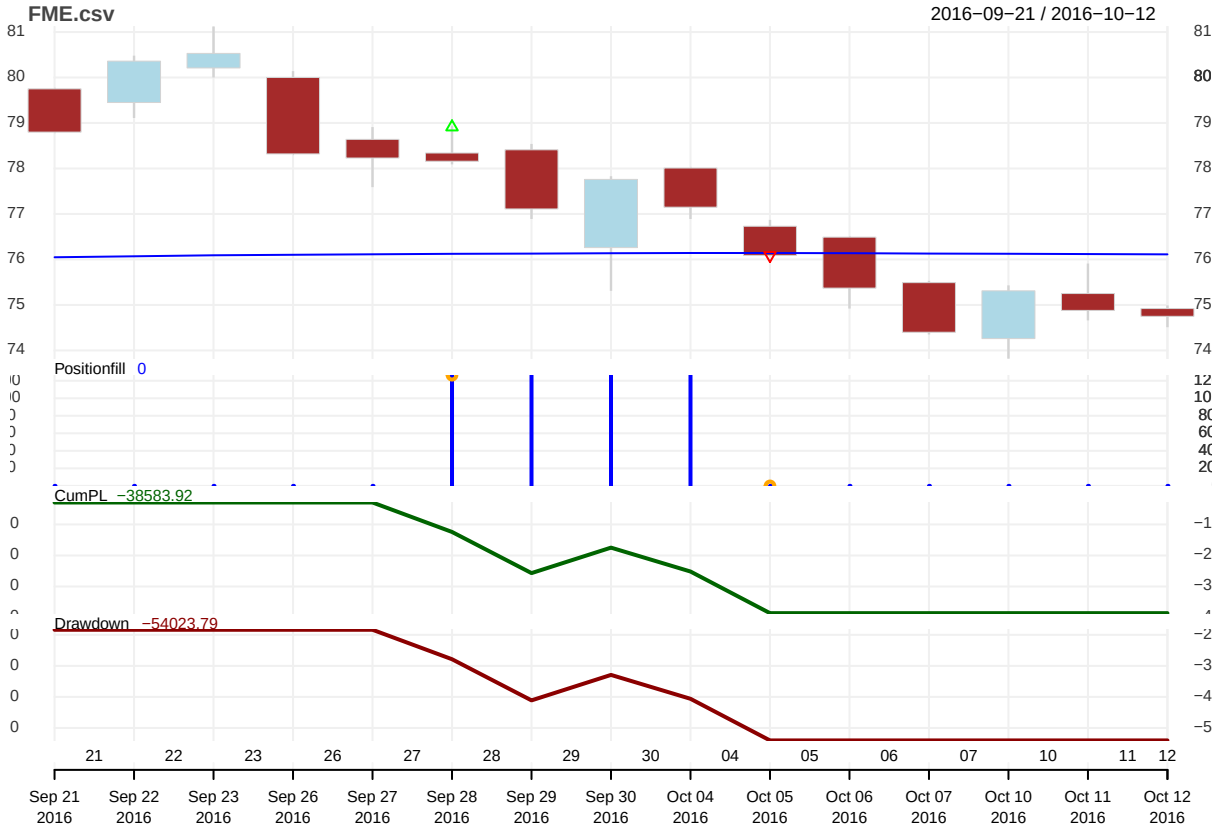

DTE.csv
2017−01−04 / 2017−01−19
16.6
16.5
16.4
16.3
16.2
16.1
16.0
15.9
15.8
15.7
15.6
Positionfill 0
CumPL −13808.735
Drawdown −13808.735
Jan 04 2017
Jan 05 2017
Jan 06 2017
Jan 09 2017
Jan 10 2017
Jan 11 2017
Jan 12 2017
Jan 13 2017
Jan 16 2017
Jan 17 2017
Jan 18 2017
Jan 19 2017

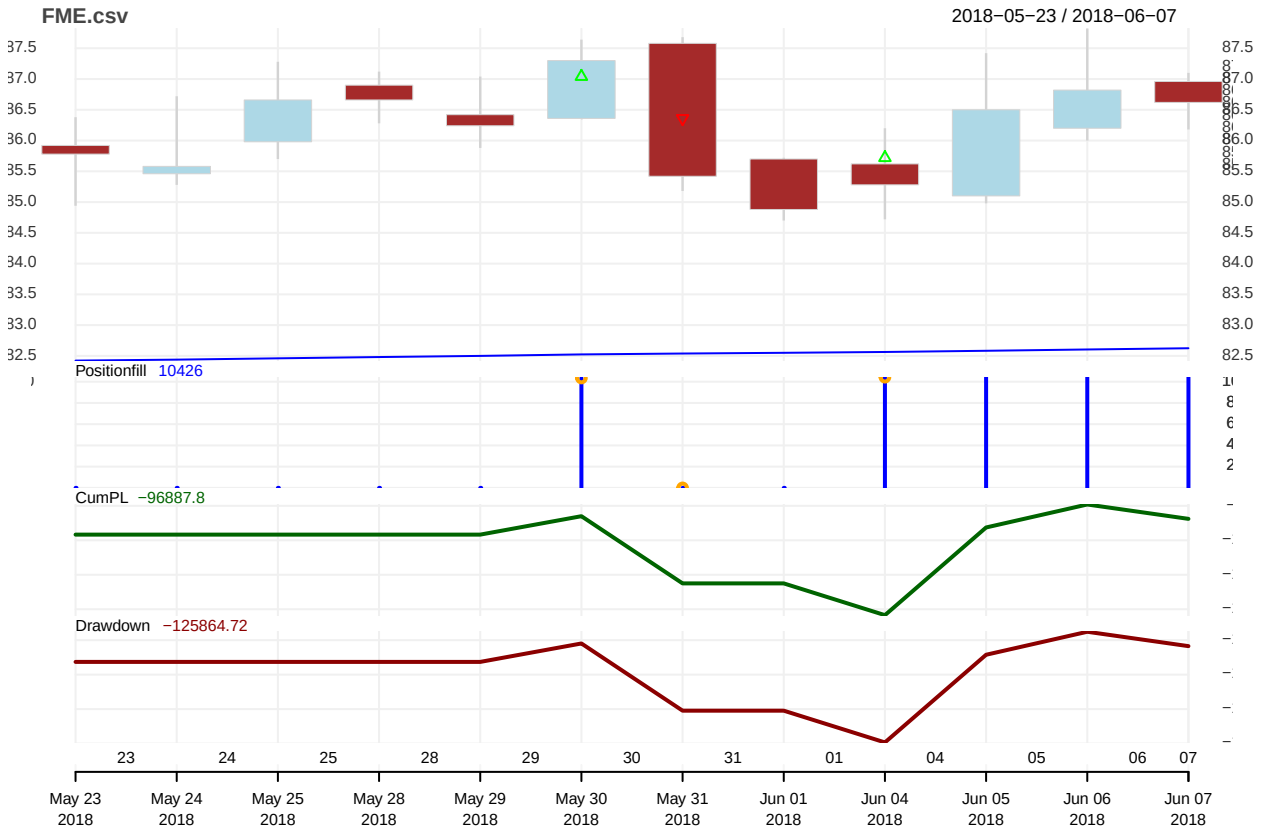

DTE.csv
2017−01−27 / 2017−02−16
16.4
16.3
16.2
16.1
16.0
15.9
15.8
15.7
15.6
Positionfill 61500
60
50
40
30
20
10
CumPL −20014.595
−1
−2
−2
−3
Drawdown −20014.595
−1
−2
−2
−3
27
30
31
01
02
03
06
07
08
09
10
13
14
15
16
Jan 27 2017
Jan 30 2017
Jan 31 2017
Feb 01 2017
Feb 02 2017
Feb 03 2017
Feb 06 2017
Feb 07 2017
Feb 08 2017
Feb 09 2017
Feb 10 2017
Feb 13 2017
Feb 14 2017
Feb 15 2017
Feb 16 2017

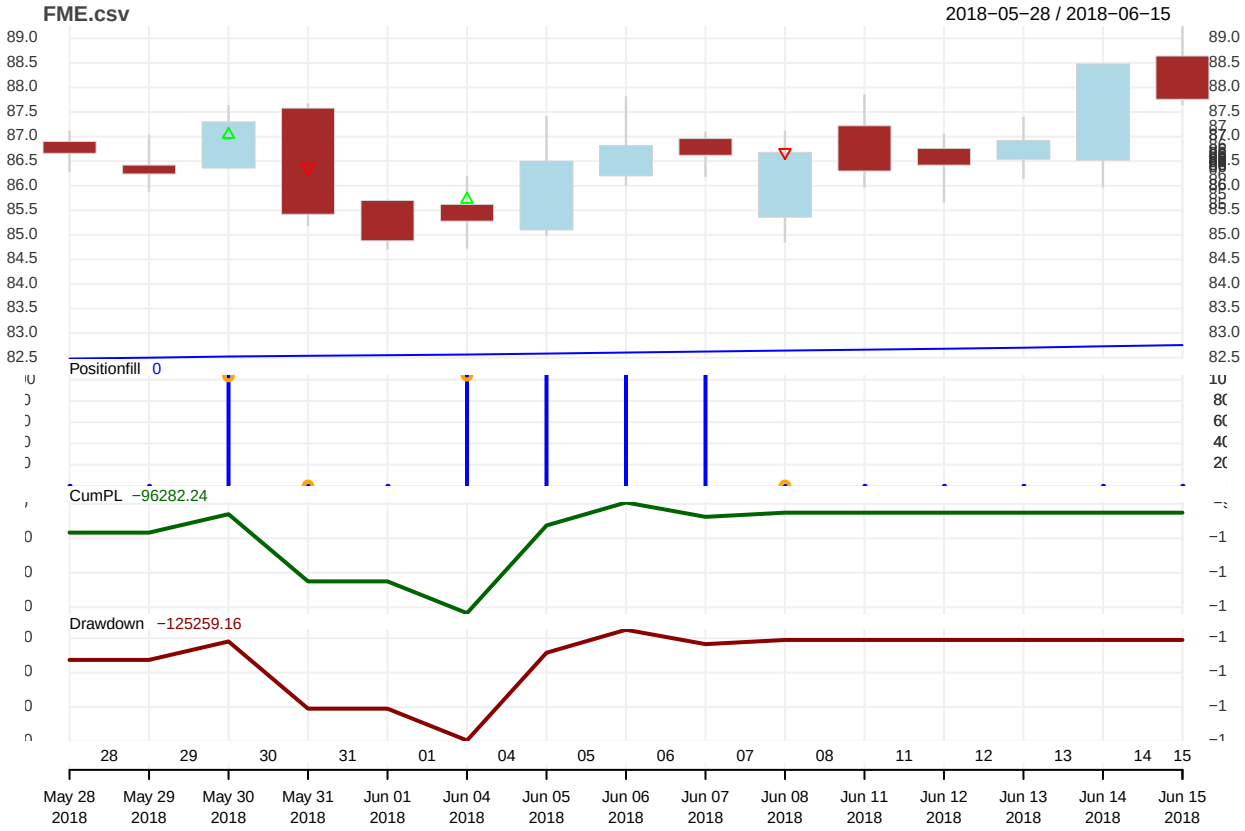

DTE.csv
2017−04−25 / 2017−05−15
17.6
17.4
17.2
17.0
16.8
16.6
16.4
16.2
16.0
15.8
Positionfill 0
CumPL 64358.37
Drawdown 0
25
26
27
28
02
03
04
05
08
09
10
11
12
15
Apr 25 2017
Apr 26 2017
Apr 27 2017
Apr 28 2017
May 02 2017
May 03 2017
May 04 2017
May 05 2017
May 08 2017
May 09 2017
May 10 2017
May 11 2017
May 12 2017
May 15 2017

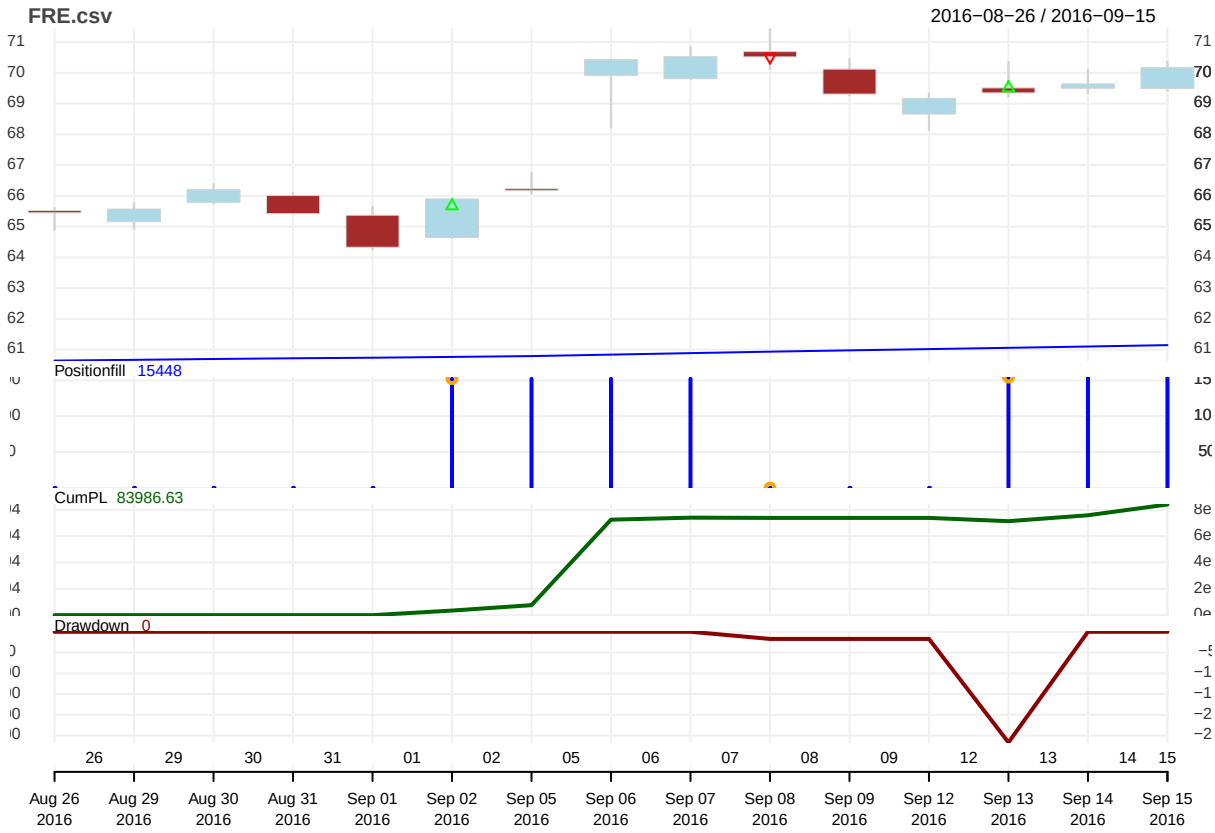

DTE.csv
2017−05−12 / 2017−06−01
Positionfill 0
CumPL 81587.945
Drawdown −11836.025
May 12 2017
May 15 2017
May 16 2017
May 17 2017
May 18 2017
May 19 2017
May 22 2017
May 23 2017
May 24 2017
May 25 2017
May 26 2017
May 29 2017
May 30 2017
May 31 2017
Jun 01 2017

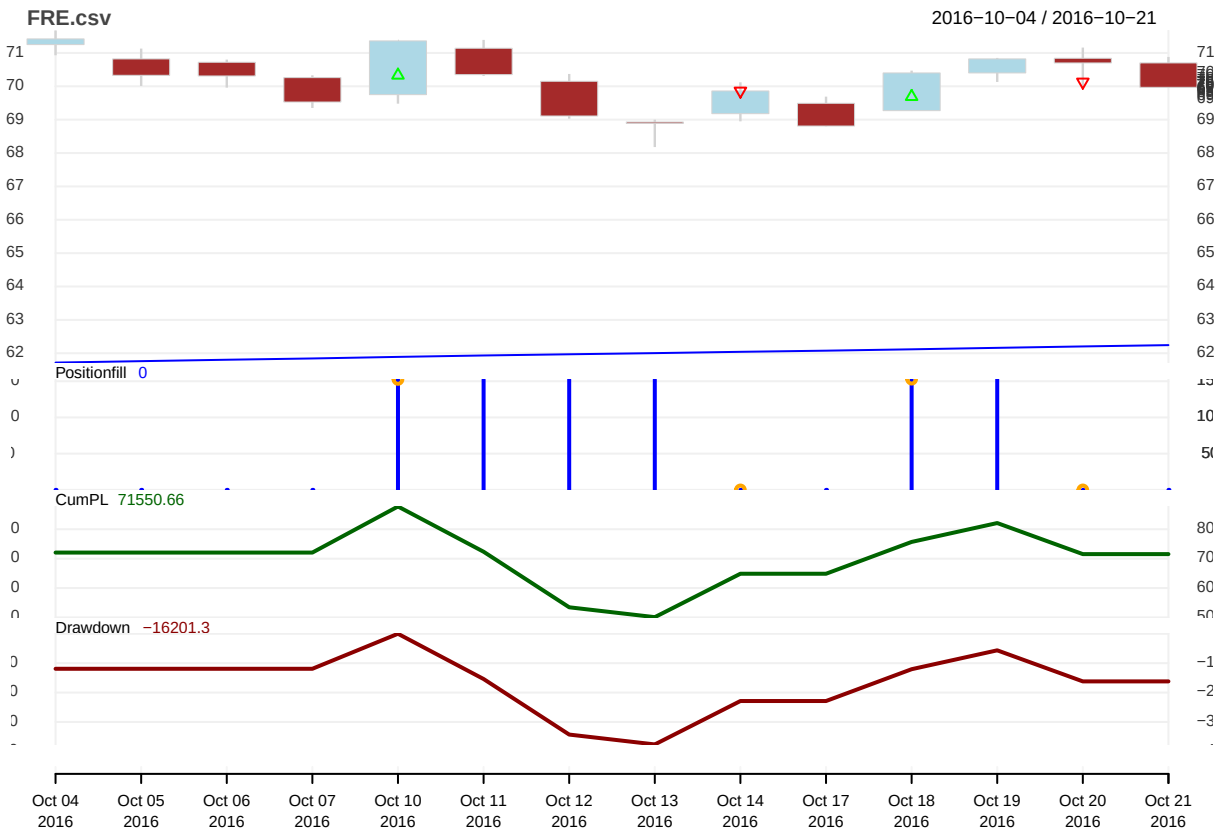

FME.csv
2016−08−26 / 2016−09−15
81.0
80.5
80.0
79.5
79.0
78.5
78.0
77.5
77.0
76.5
76.0
Positionfill 12695
CumPL −3908.82
Drawdown −19348.69
26
29
30
31
01
02
05
06
07
08
09
12
13
14
15
Aug 26 2016
Aug 29 2016
Aug 30 2016
Aug 31 2016
Sep 01 2016
Sep 02 2016
Sep 05 2016
Sep 06 2016
Sep 07 2016
Sep 08 2016
Sep 09 2016
Sep 12 2016
Sep 13 2016
Sep 14 2016
Sep 15 2016

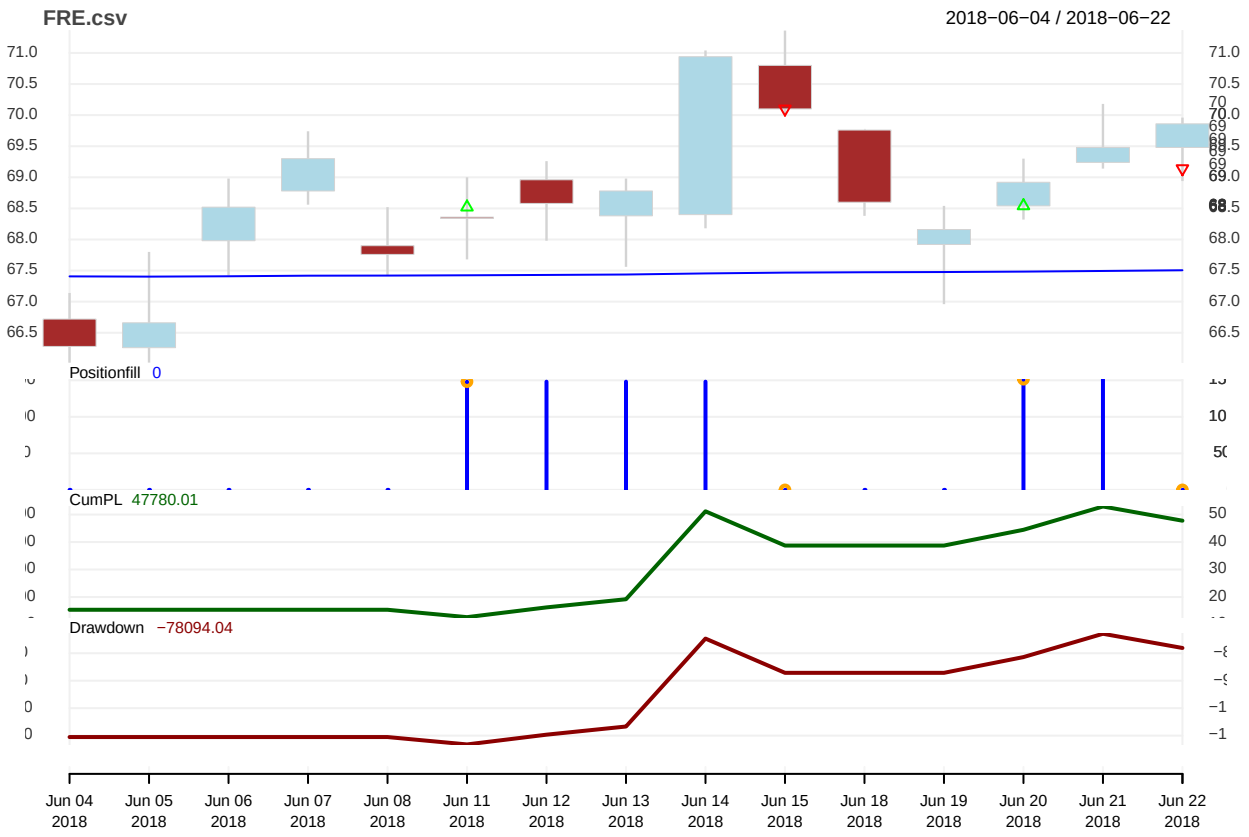

FME.csv
2016−09−21 / 2016−10−12
Positionfill 0
CumPL −38583.92
Drawdown −54023.79
21 22 23 26 27 28 29 30 04 05 06 07 10 11 12
Sep 21 2016
Sep 22 2016
Sep 23 2016
Sep 26 2016
Sep 27 2016
Sep 28 2016
Sep 29 2016
Sep 30 2016
Oct 04 2016
Oct 05 2016
Oct 06 2016
Oct 07 2016
Oct 10 2016
Oct 11 2016
Oct 12 2016

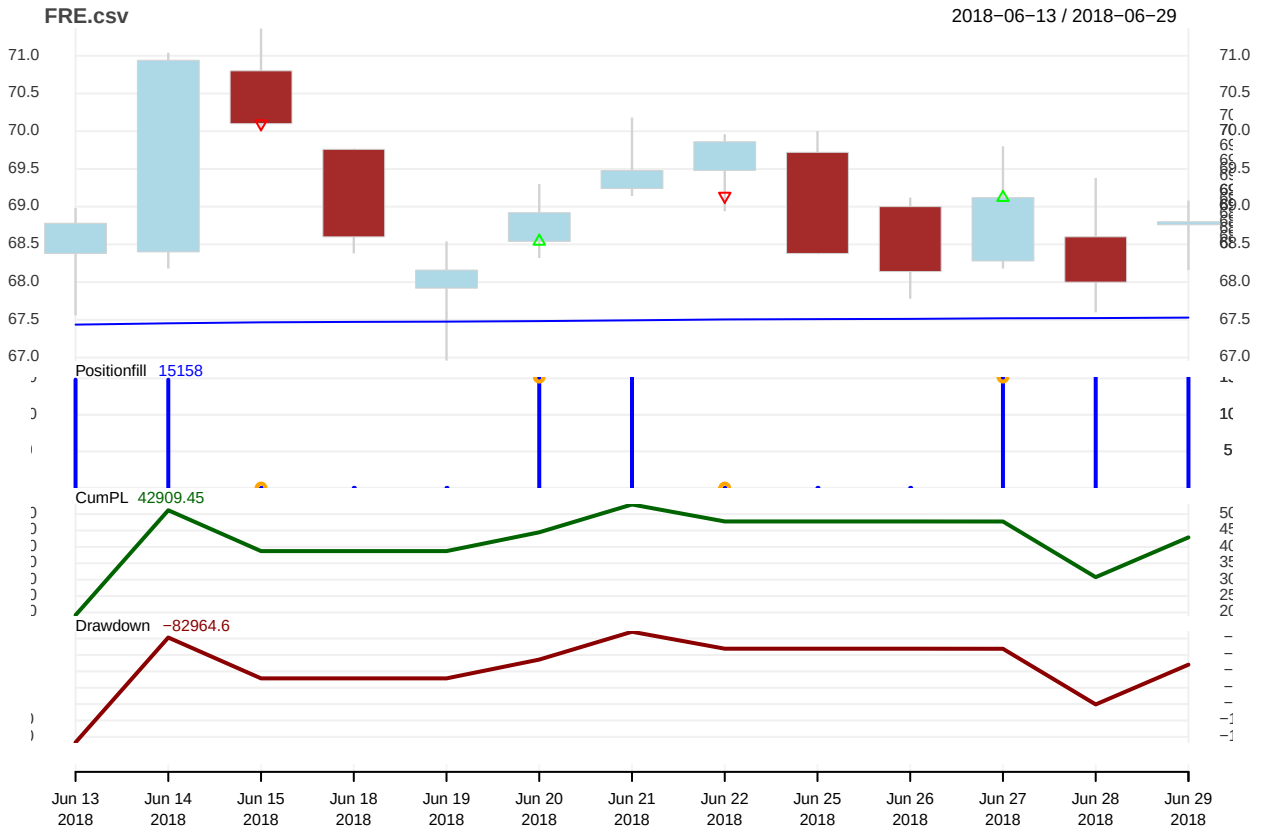

FME.csv
2018−05−23 / 2018−06−07
Positionfill 10426
CumPL −96887.8
Drawdown −125864.72
23
24
25
28
29
30
31
01
04
05
06
07
May 23 2018
May 24 2018
May 25 2018
May 28 2018
May 29 2018
May 30 2018
May 31 2018
Jun 01 2018
Jun 04 2018
Jun 05 2018
Jun 06 2018
Jun 07 2018


FME.csv
2018−05−28 / 2018−06−15
Positionfill 0
CumPL −96282.24
Drawdown −125259.16
May 28 2018
May 29 2018
May 30 2018
May 31 2018
Jun 01 2018
Jun 04 2018
Jun 05 2018
Jun 06 2018
Jun 07 2018
Jun 08 2018
Jun 11 2018
Jun 12 2018
Jun 13 2018
Jun 14 2018
Jun 15 2018


FRE.csv
2016−08−26 / 2016−09−15
Positionfill 15448
CumPL 83986.63
Drawdown 0
Aug 26 2016
Aug 29 2016
Aug 30 2016
Aug 31 2016
Sep 01 2016
Sep 02 2016
Sep 05 2016
Sep 06 2016
Sep 07 2016
Sep 08 2016
Sep 09 2016
Sep 12 2016
Sep 13 2016
Sep 14 2016
Sep 15 2016


FRE.csv
2016−10−04 / 2016−10−21
Positionfill 0
CumPL 71550.66
Drawdown −16201.3
Oct 04 2016
Oct 05 2016
Oct 06 2016
Oct 07 2016
Oct 10 2016
Oct 11 2016
Oct 12 2016
Oct 13 2016
Oct 14 2016
Oct 17 2016
Oct 18 2016
Oct 19 2016
Oct 20 2016
Oct 21 2016


FRE.csv
2018−06−04 / 2018−06−22
71.0
70.5
70.0
69.5
69.0
68.5
68.0
67.5
67.0
66.5
Positionfill 0
CumPL 47780.01
50
40
30
20
Drawdown −78094.04
Jun 04 2018
Jun 05 2018
Jun 06 2018
Jun 07 2018
Jun 08 2018
Jun 11 2018
Jun 12 2018
Jun 13 2018
Jun 14 2018
Jun 15 2018
Jun 18 2018
Jun 19 2018
Jun 20 2018
Jun 21 2018
Jun 22 2018

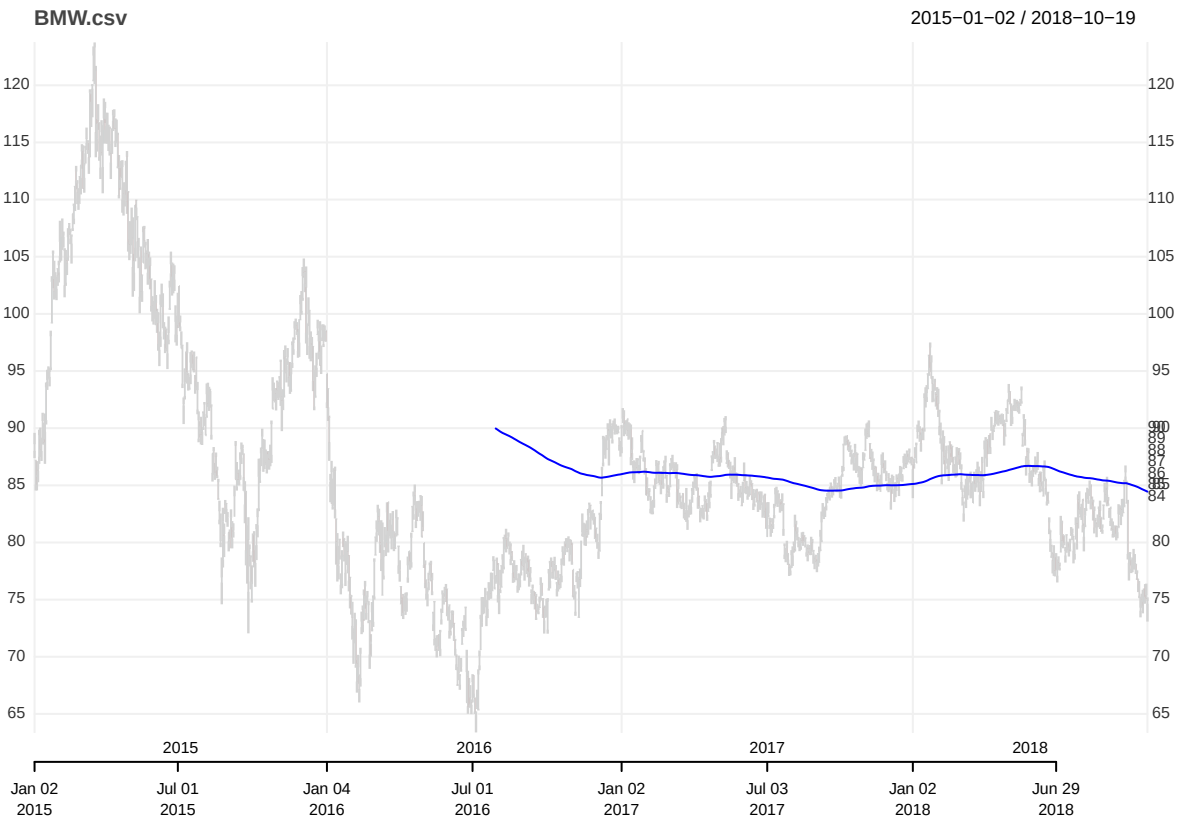


Part C: Analysis and Reporting

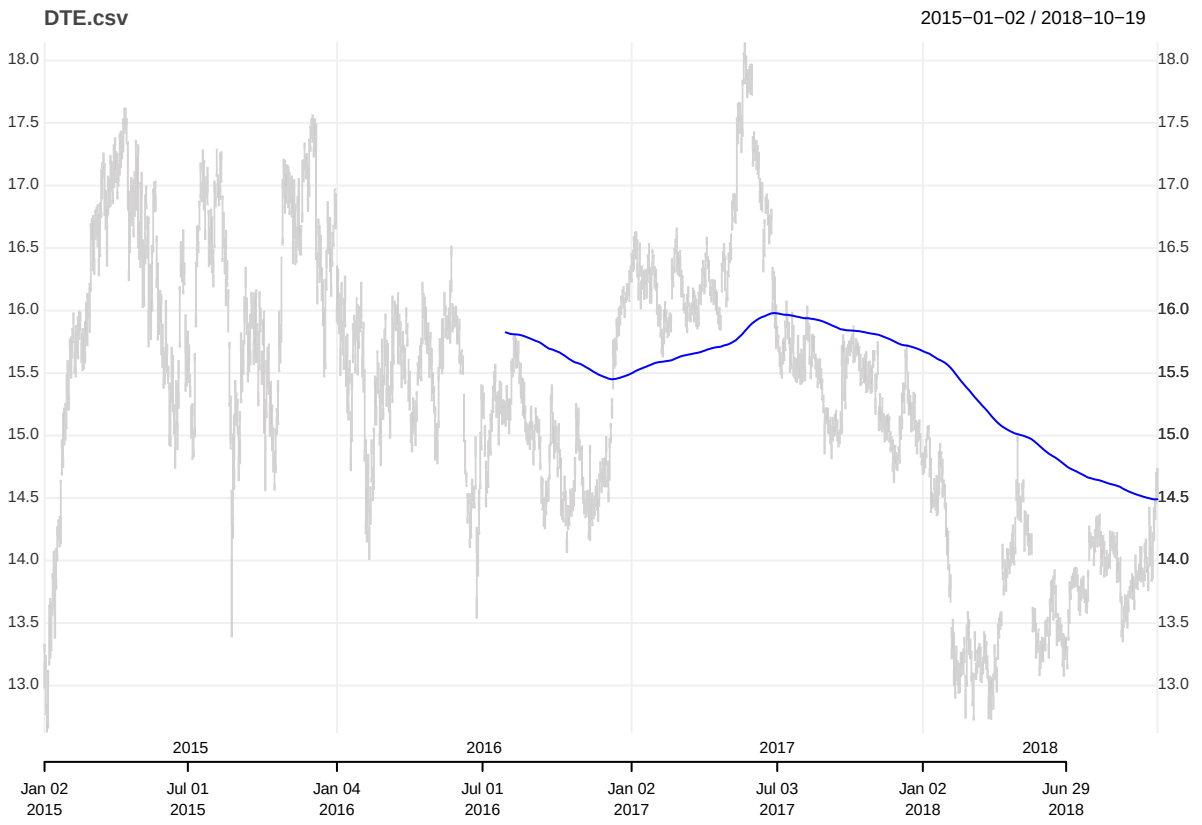
Step 1: Visualize original data

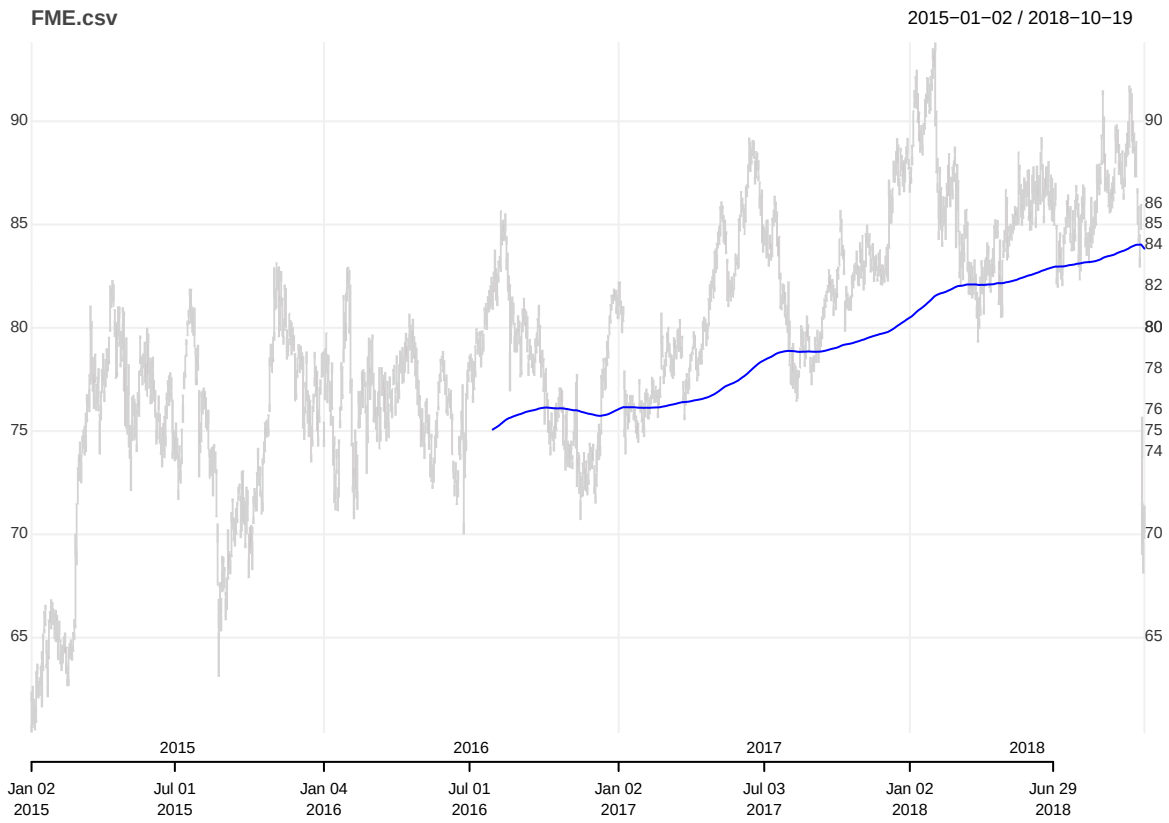
Plot of the instrument with the EMA line which indicates the general trend of the stock exponentially smoothed for the last 200 days. Moreover, the tradevolume is added below the graph. This plots can be used to get a first impression of the long term trend of the instrument and to see which general behaviour of the price can be observed

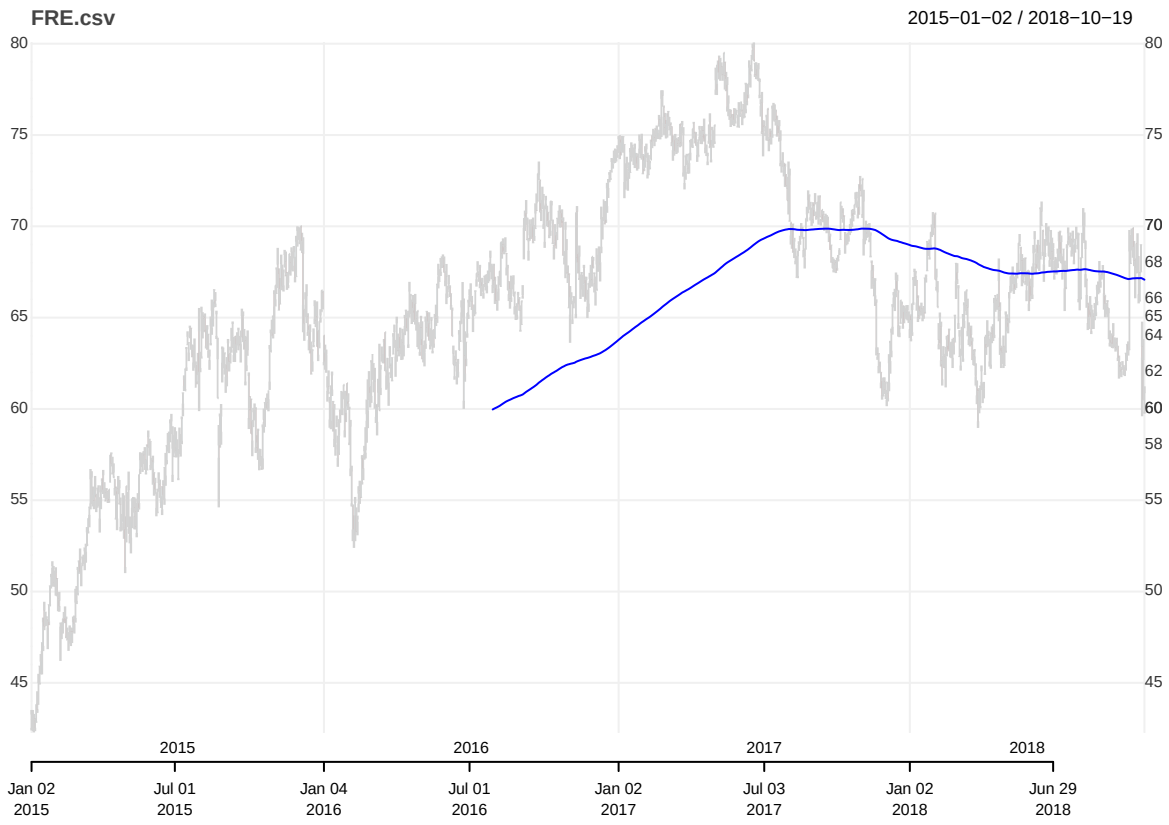
```
for (instrument in instrumentlist){
  chart <- get(instrument)
  print(chart_Series(x=chart,name=instrument,
                     theme=myTheme,
                     subset=daterange,
                     TA="add_TA((chart$EMA), on=1,type='l',col='blue')"))
}
```


BMW.csv
2015−01−02 / 2018−10−19
120
115
110
105
100
95
90
85
80
75
70
65
2015
2016
2017
2018
Jan 02 2015
Jul 01 2015
Jan 04 2016
Jul 01 2016
Jan 02 2017
Jul 03 2017
Jan 02 2018
Jun 29 2018


DAI.csv
2015−01−02 / 2018−10−19
95
90
85
80
75
70
65
60
55
50
74
72
70
68
66
65
64
62
60
55
50
2015
2016
2017
2018
Jan 02 2015
Jul 01 2015
Jan 04 2016
Jul 01 2016
Jan 02 2017
Jul 03 2017
Jan 02 2018
Jun 29 2018


DTE.csv
2015−01−02 / 2018−10−19
18.0
17.5
17.0
16.5
16.0
15.5
15.0
14.5
14.0
13.5
13.0
2015
2016
2017
2018
Jan 02 2015
Jul 01 2015
Jan 04 2016
Jul 01 2016
Jan 02 2017
Jul 03 2017
Jan 02 2018
Jun 29 2018


FME.csv
2015−01−02 / 2018−10−19
90
85
80
75
70
65
90
86
85
84
82
80
78
76
75
74
70
65
2015
2016
2017
2018
Jan 02 2015
Jul 01 2015
Jan 04 2016
Jul 01 2016
Jan 02 2017
Jul 03 2017
Jan 02 2018
Jun 29 2018



Step 2: All transactions performed by the trading system

The following table can be used to get a better overview of the transactions performed and the exact details like the quantity, price and value per transaction.

```
for (instrument in instrumentlist){
  print(sprintf("Transactions for the instrument: %s", instrument))
  txns <- getTxns(Portfolio = portfolioname, Symbol = instrument)
  txns.pr <- txns[,c(1,2,4,6)]
  colnames(txns.pr) <- c("Quantity","Price","Value","Net realized Profit")
  print(txns.pr)
  writeLines("")
}
```

```
## [1] "Transactions for the instrument: BMW.csv"
##            Quantity Price      Value Net realized Profit
## 1999-12-31        0  0.00        0.0                0.00
## 2017-01-17    11473 87.16   999986.7              -20.00
## 2017-01-23   -11473 86.26  -989661.0           -10345.70
## 2017-05-04    11286 87.68   989556.5              -20.00
## 2017-05-09   -11286 88.90 -1003325.4            13748.92
## 2017-11-16    11672 85.96  1003325.1              -20.00
## 2017-11-20   -11672 84.48  -986050.6           -17294.56
## 2017-11-24    11316 87.13   985963.1              -20.00
## 2017-11-29   -11316 85.65  -969215.4           -16767.68
## 2017-12-29    11088 87.41   969202.1              -20.00
## 2018-01-05   -11088 87.00  -964656.0            -4566.08
```

```
## 2018-01-12    10764 89.62   964669.7              -20.00
## 2018-01-15   -10764 89.02  -958211.3             -6478.40
## 2018-01-31    10285 93.16   958150.6              -20.00
## 2018-02-06   -10285 88.13  -906417.0            -51753.55
## 2018-02-12    10289 88.09   906358.0              -20.00
## 2018-02-16   -10289 88.12  -906666.7              288.67
## 2018-05-22    10107 89.70   906597.9              -20.00
## 2018-05-23   -10107 89.28  -902353.0             -4264.94
## 2018-05-25    10262 87.93   902337.7              -20.00
## 2018-05-30   -10262 85.95  -882018.9            -20338.76
##
## [1] "Transactions for the instrument: DAI.csv"
##            Quantity Price      Value Net realized Profit
## 1999-12-31        0  0.00        0.0                0.00
## 2016-12-30    14154 70.65   999980.1              -20.00
## 2017-01-02   -14154 70.01  -990921.5             -9078.56
## 2017-01-17    14055 70.50   990877.5              -20.00
## 2017-01-23   -14055 70.17  -986239.3             -4658.15
## 2017-03-23    13990 70.49   986155.1              -20.00
## 2017-03-29   -13990 72.36 -1012316.4             26141.30
## 2017-10-11    14770 68.54  1012335.8              -20.00
## 2017-10-17   -14770 68.72 -1014994.4              2638.60
## 2017-11-16    14561 69.70  1014901.7              -20.00
## 2017-11-22   -14561 69.98 -1018978.8              4057.08
## 2017-11-24    14490 70.32  1018936.8              -20.00
## 2017-11-30   -14490 69.52 -1007344.8            -11612.00
## 2017-12-04    14448 69.72  1007314.6              -20.00
## 2017-12-06   -14448 69.32 -1001535.4             -5799.20
## 2017-12-20    13974 71.67  1001516.6              -20.00
## 2017-12-28   -13974 70.90  -990756.6            -10779.98
## 2018-01-12    13323 74.36   990698.3              -20.00
## 2018-01-17   -13323 73.77  -982837.7             -7880.57
## 2018-02-07    13929 70.56   982830.2              -20.00
## 2018-02-13   -13929 70.62  -983666.0              815.74
##
## [1] "Transactions for the instrument: DTE.csv"
##            Quantity  Price      Value Net realized Profit
## 1999-12-31        0  0.000        0.0                0.00
## 2017-01-11    60975 16.400   999990.0              -20.00
## 2017-01-12   -60975 16.195  -987490.1            -12519.87
## 2017-01-13    61443 16.240   997834.3              -20.00
## 2017-01-19   -61443 16.220  -996605.5             -1248.86
## 2017-02-03    61464 16.045   986189.9              -20.00
## 2017-02-09   -61464 15.930  -979121.5             -7088.36
## 2017-02-15    61500 15.920   979080.0              -20.00
## 2017-02-21   -61500 16.295 -1002142.5             23042.50
## 2017-03-09    62494 16.035  1002091.3              -20.00
## 2017-03-15   -62494 15.990  -999279.1             -2832.23
## 2017-04-20    62433 16.005   999240.2              -20.00
## 2017-04-26   -62433 16.400 -1023901.2             24641.03
## 2017-05-02    63319 16.170  1023868.2              -20.00
## 2017-05-08   -63319 16.810 -1064392.4             40504.16
## 2017-05-19    60595 17.565  1064351.2              -20.00
## 2017-05-25   -60595 17.850 -1081620.8             17249.58
```

```
## 2017-06-02    62142 17.405  1081581.5             -20.00
## 2017-06-09   -62142 17.125 -1064181.8           -17419.76
## 2017-06-16    63912 16.650  1064134.8             -20.00
## 2017-06-22   -63912 16.695 -1067010.8             2856.04
##
## [1] "Transactions for the instrument: FME.csv"
##            Quantity Price      Value Net realized Profit
## 1999-12-31        0  0.00        0.0                0.00
## 2016-09-02    12569 79.56   999989.6              -20.00
## 2016-09-08   -12569 79.88 -1004011.7             4002.08
## 2016-09-13    12695 79.08  1003920.6              -20.00
## 2016-09-19   -12695 78.54  -997065.3            -6875.30
## 2016-09-28    12635 78.91   997027.8              -20.00
## 2016-10-05   -12635 76.09  -961397.2           -35650.70
## 2016-12-15    12322 78.02   961362.4              -20.00
## 2016-12-21   -12322 81.08  -999067.8            37685.32
## 2017-01-02    12246 81.58   999028.7              -20.00
## 2017-01-06   -12246 80.60  -987027.6           -12021.08
## 2017-02-15    12793 77.15   986980.0              -20.00
## 2017-02-20   -12793 76.42  -977641.1            -9358.89
## 2017-02-27    12502 78.20   977656.4              -20.00
## 2017-03-03   -12502 79.07  -988533.1            10856.74
## 2017-03-20    12501 79.07   988454.1              -20.00
## 2017-03-21   -12501 77.01  -962702.0           -25772.06
## 2017-05-02    11621 82.84   962683.6              -20.00
## 2017-05-08   -11621 84.83  -985809.4            23105.79
## 2017-05-25    12026 81.97   985771.2              -20.00
## 2017-05-31   -12026 85.16 -1024134.2            38342.94
## 2017-06-22    11624 88.10  1024074.4              -20.00
## 2017-06-28   -11624 86.20 -1001988.8           -22105.60
## 2017-07-12    11911 84.12  1001953.3              -20.00
## 2017-07-17   -11911 85.16 -1014340.8            12367.44
## 2017-07-20    12016 84.41  1014270.6              -20.00
## 2017-07-26   -12016 81.41  -978222.6           -36068.00
## 2017-07-31    12123 80.69   978204.9              -20.00
## 2017-08-03   -12123 77.76  -942684.5           -35540.39
## 2017-09-22    11454 82.30   942664.2              -20.00
## 2017-09-28   -11454 82.67  -946902.2             4217.98
## 2017-10-11    11646 81.30   946819.8              -20.00
## 2017-10-17   -11646 80.85  -941579.1            -5260.70
## 2017-11-01    11149 84.45   941533.1              -20.00
## 2017-11-07   -11149 84.19  -938634.3            -2918.74
## 2017-11-09    11115 84.45   938661.8              -20.00
## 2017-11-15   -11115 83.00  -922545.0           -16136.75
## 2017-11-16    11125 83.44   928270.0              -20.00
## 2017-11-22   -11125 82.70  -920037.5            -8252.50
## 2017-11-24    11033 82.86   914194.4              -20.00
## 2017-11-30   -11033 82.60  -911325.8            -2888.58
## 2017-12-21    10358 87.98   911296.8              -20.00
## 2017-12-29   -10358 87.78  -909225.2            -2091.60
## 2018-01-03    10379 87.60   909200.4              -20.00
## 2018-01-09   -10379 92.04  -955283.2            46062.76
## 2018-01-16    10661 89.60   955225.6              -20.00
## 2018-01-22   -10661 90.30  -962688.3             7442.70
```

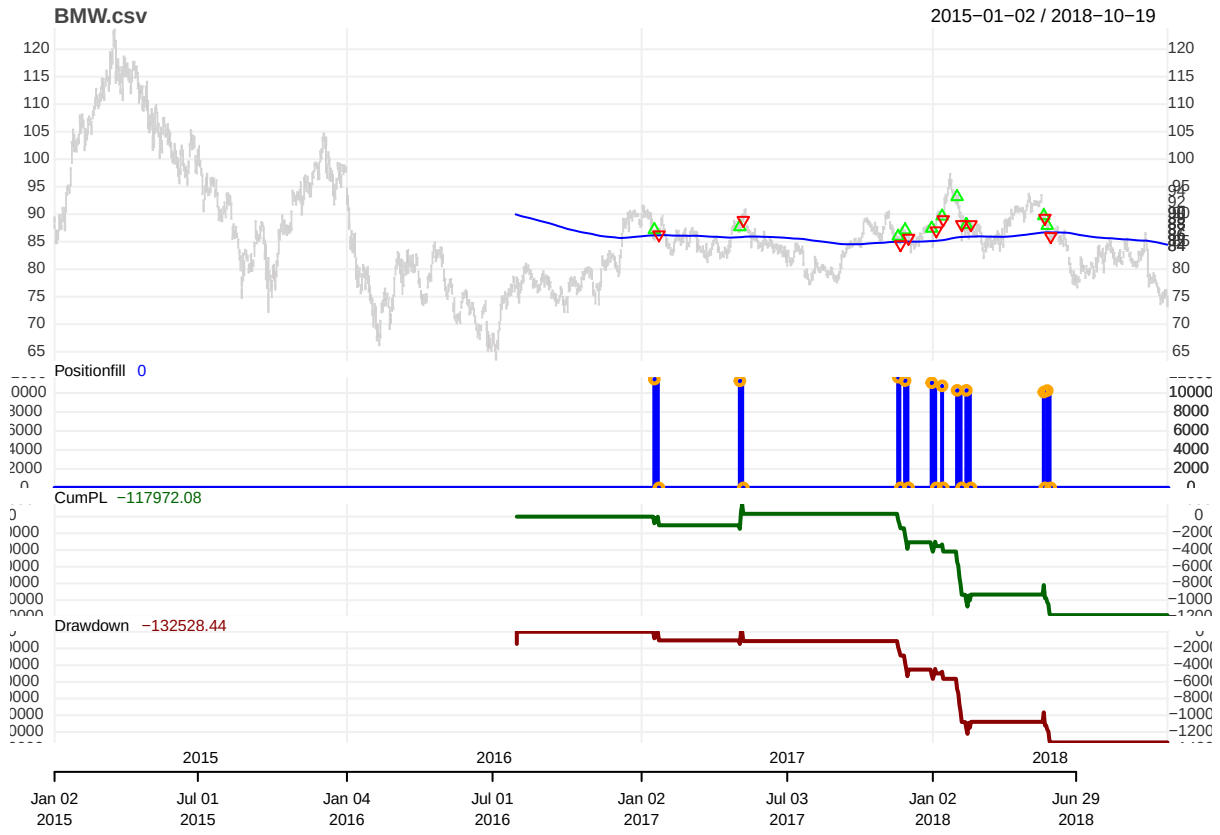
```
## 2018-01-26    10484 91.82   962640.9             -20.00
## 2018-02-01   -10484 90.24  -946076.2           -16584.72
## 2018-02-07    10914 86.68   946025.5             -20.00
## 2018-02-08   -10914 84.96  -927253.4           -18792.08
## 2018-02-12    10807 85.80   927240.6             -20.00
## 2018-02-16   -10807 87.88  -949719.2            22458.56
## 2018-02-28    10853 87.50   949637.5             -20.00
## 2018-03-06   -10853 83.00  -900799.0           -48858.50
## 2018-05-30    10349 87.04   900777.0             -20.00
## 2018-05-31   -10349 86.36  -893739.6            -7057.32
## 2018-06-04    10426 85.72   893716.7             -20.00
## 2018-06-08   -10426 86.68  -903725.7             9988.96
## 2018-07-24    10730 84.22   903680.6             -20.00
## 2018-07-30   -10730 85.72  -919775.6            16075.00
## 2018-09-07    10623 86.58   919739.3             -20.00
## 2018-09-13   -10623 89.04  -945871.9            26112.58
##
## [1] "Transactions for the instrument: FRE.csv"
##            Quantity Price      Value Net realized Profit
## 1999-12-31        0  0.00        0.0                0.00
## 2016-09-02    15227 65.67   999957.1             -20.00
## 2016-09-08   -15227 70.52 -1073808.0            73830.95
## 2016-09-13    15448 69.51  1073790.5             -20.00
## 2016-09-16   -15448 69.40 -1072091.2            -1719.28
## 2016-10-10    15243 70.33  1072040.2             -20.00
## 2016-10-14   -15243 69.86 -1064876.0            -7184.21
## 2016-10-18    15280 69.69  1064863.2             -20.00
## 2016-10-20   -15280 70.13 -1071586.4             6703.20
## 2016-10-27    15611 68.64  1071539.0             -20.00
## 2016-11-02   -15611 66.10 -1031887.1           -39671.94
## 2016-11-07    15653 65.92  1031845.8             -20.00
## 2016-11-11   -15653 68.02 -1064717.1            32851.30
## 2016-11-24    16003 66.53  1064679.6             -20.00
## 2016-11-30   -16003 67.75 -1084203.2            19503.66
## 2017-01-25    14666 73.92  1084110.7             -20.00
## 2017-01-27   -14666 73.98 -1084990.7              859.96
## 2017-03-28    14770 73.46  1085004.2             -20.00
## 2017-04-03   -14770 74.84 -1105386.8            20362.60
## 2017-04-24    14801 74.68  1105338.7             -20.00
## 2017-04-25   -14801 74.52 -1102970.5            -2388.16
## 2017-05-31    14493 76.10  1102917.3             -20.00
## 2017-06-06   -14493 76.29 -1105671.0             2733.67
## 2017-06-07    14570 76.40  1113148.0             -20.00
## 2017-06-13   -14570 77.79 -1133400.3            20232.30
## 2017-06-22    14258 78.96  1125811.7             -20.00
## 2017-06-28   -14258 76.60 -1092162.8           -33668.88
## 2017-07-05    14466 75.50  1092183.0             -20.00
## 2017-07-06   -14466 74.70 -1080610.2           -11592.80
## 2017-08-01    14861 72.71  1080543.3             -20.00
## 2017-08-02   -14861 70.65 -1049929.7           -30633.66
## 2017-09-05    14827 70.81  1049899.9             -20.00
## 2017-09-11   -14827 70.49 -1045155.2            -4764.64
## 2017-10-26    14650 71.34  1045131.0             -20.00
## 2017-11-02   -14650 69.31 -1015391.5           -29759.50
```

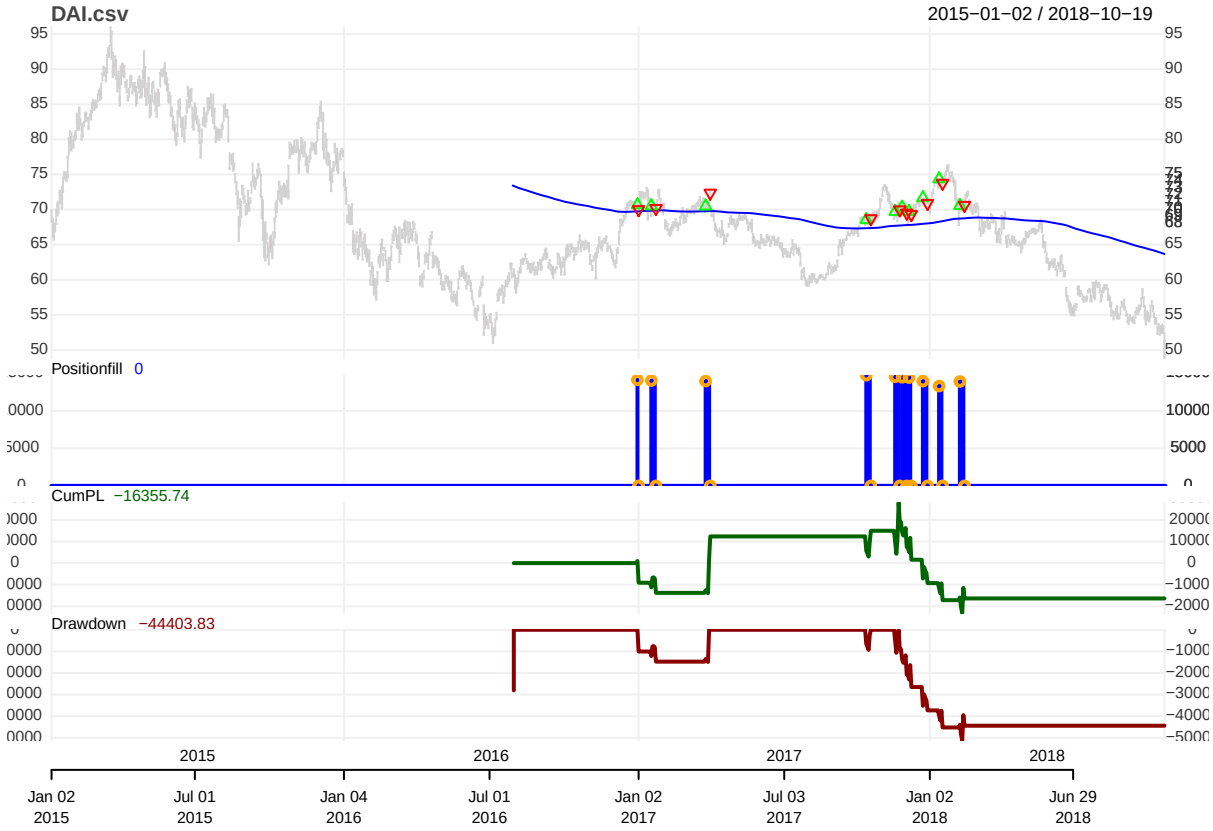
```
## 2018-06-11    14818 68.52  1015329.4            -20.00
## 2018-06-15   -14818 70.10 -1038741.8          23392.44
## 2018-06-20    15155 68.54  1038723.7            -20.00
## 2018-06-22   -15155 69.14 -1047816.7           9073.00
## 2018-06-27    15158 69.12  1047721.0            -20.00
## 2018-07-03   -15158 67.72 -1026499.8          -21241.20
## 2018-07-24    14821 69.26  1026502.5            -20.00
## 2018-07-30   -14821 69.30 -1027095.3            572.84
```

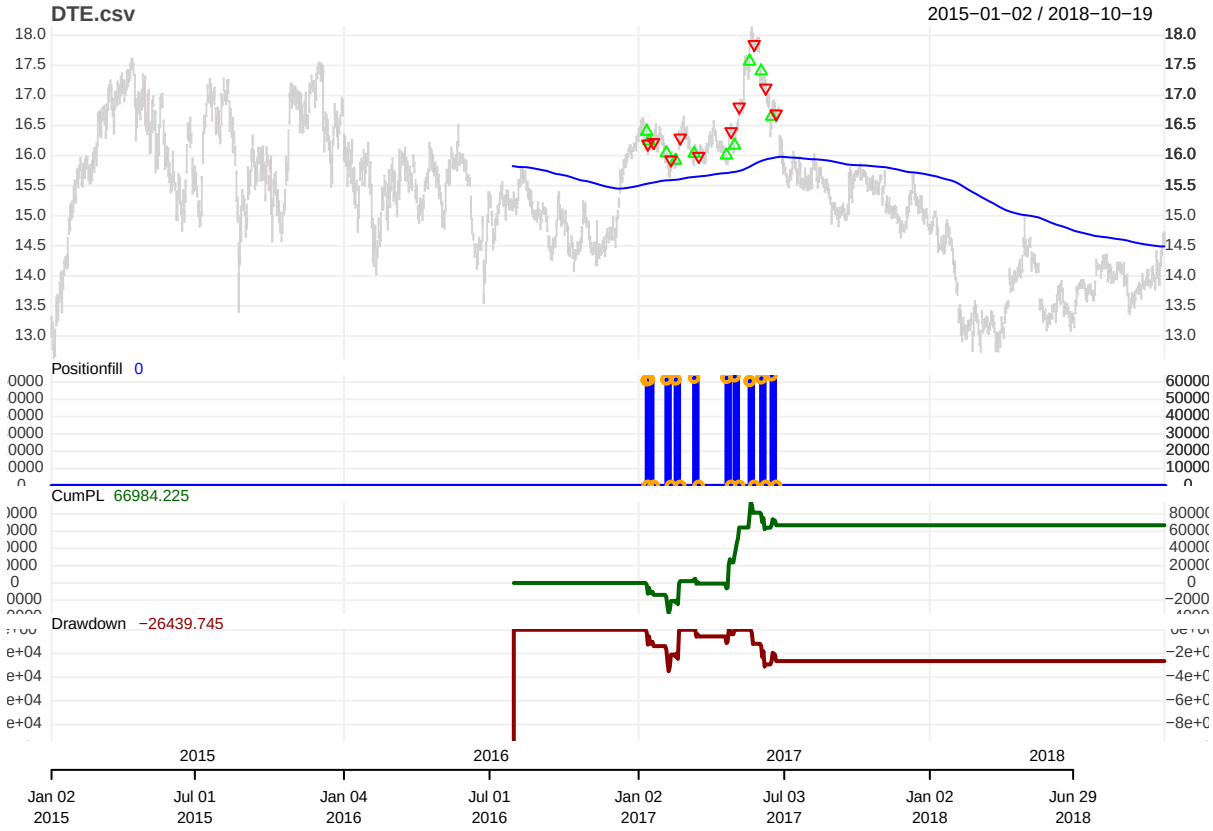
Step 3: Graph which visualize transactions

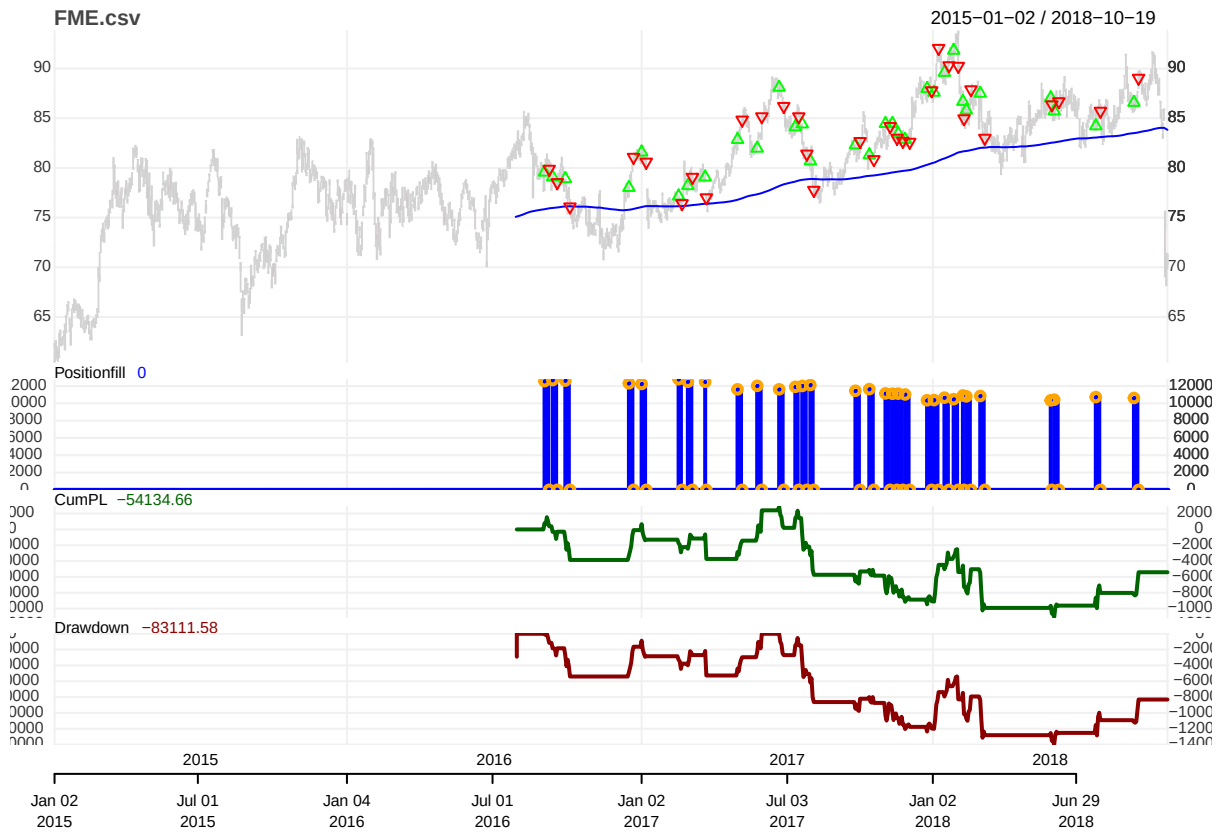
The following graphs show the combined view of the performance of the Smash Day trading system. It visualizes the trades (buy-transactions are visualized in green and sell-transactions are visualized in red). Moreover, the size of the blue squares indicates the size of the position (height) and the holding 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.

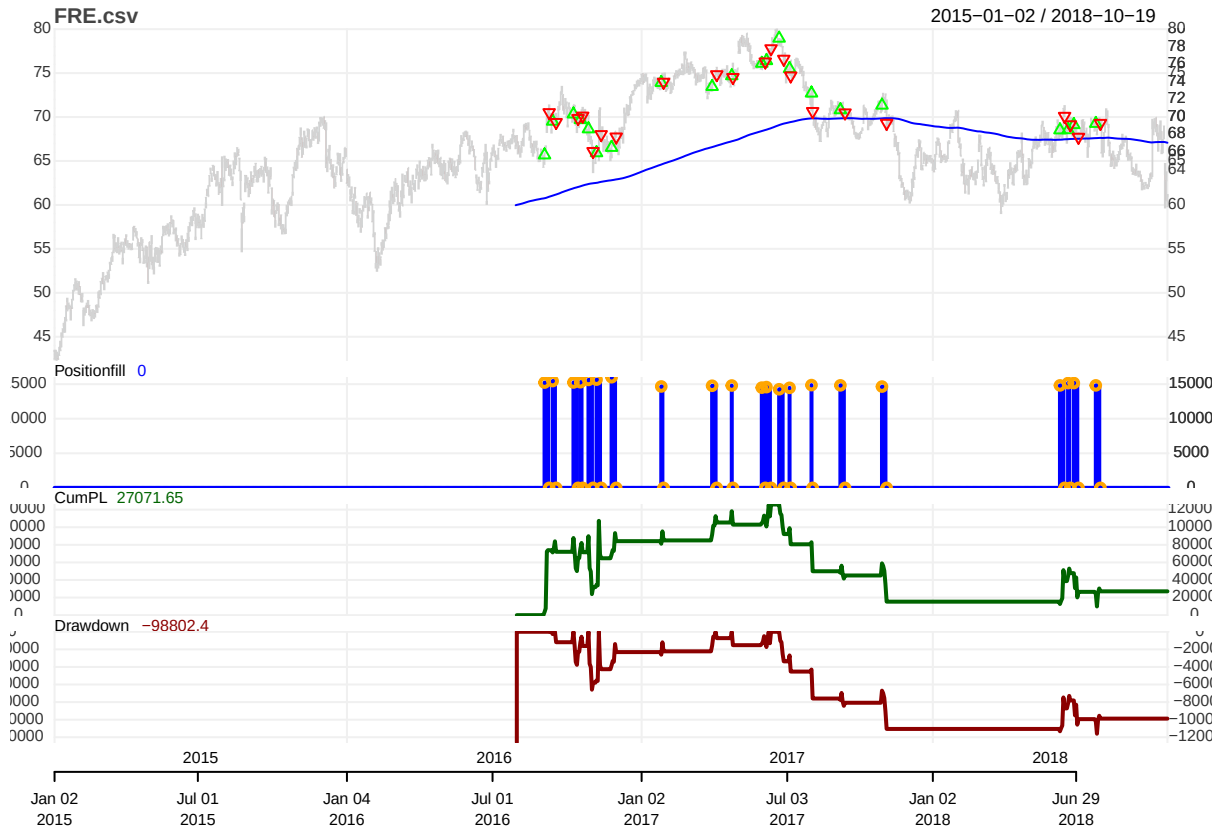
```
# Plot graph with indicators for transaction
for (instrument in instrumentlist){
  chart.Posn(portfolioname,
             Symbol=instrument,
             type='candlesticks',
             theme=myTheme,
             subset=daterange,
             TA=addEMAString)
}
```




DAI.csv
2015−01−02 / 2018−10−19
Positionfill 0
CumPL −16355.74
Drawdown −44403.83
2015
2016
2017
2018
Jan 02 2015
Jul 01 2015
Jan 04 2016
Jul 01 2016
Jan 02 2017
Jul 03 2017
Jan 02 2018
Jun 29 2018


DTE.csv
2015−01−02 / 2018−10−19
Positionfill 0
CumPL 66984.225
Drawdown −26439.745
2015
2016
2017
2018
Jan 02 2015
Jul 01 2015
Jan 04 2016
Jul 01 2016
Jan 02 2017
Jul 03 2017
Jan 02 2018
Jun 29 2018


FME.csv
2015−01−02 / 2018−10−19
Positionfill 0
CumPL −54134.66
Drawdown −83111.58
2015
2016
2017
2018
Jan 02 2015
Jul 01 2015
Jan 04 2016
Jul 01 2016
Jan 02 2017
Jul 03 2017
Jan 02 2018
Jun 29 2018



Step 4: Performance Statistics

The following table summarizes some important trading statistics for all instruments. This statistic is just an excerpt of the overall statistics which can be calculated. It is printed here as it can be assumed that they are interesting for investers to rate the trading strategy.

```
library(PerformanceAnalytics)
# Get returns for the account
rets <- PortfReturns(Account=accountname)
rownames(rets) <- NULL
tstats <- tradeStats(Portfolio=portfolioname, Symbols=instrumentlist)
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] "BMW.csv"
##       Trades  Win Percent Loss Percent    W/L Ratio
##         "10"         "20"         "80"       "0.25"
##
## [1] "DAI.csv"
##       Trades  Win Percent Loss Percent    W/L Ratio
##         "10"         "40"         "60"       "0.67"
##
## [1] "DTE.csv"
##       Trades  Win Percent Loss Percent    W/L Ratio
##         "10"         "50"         "50"          "1"
##
## [1] "FME.csv"
##       Trades  Win Percent Loss Percent    W/L Ratio
##         "31"      "41.94"      "58.06"       "0.72"
##
## [1] "FRE.csv"
##       Trades  Win Percent Loss Percent    W/L Ratio
##         "21"      "52.38"      "47.62"        "1.1"
```

Step 5: Calculate statistics of the Portfolio and all instruments in the portfolio

```
## Performance Metrics
```

```
##                               BMW         DAI        DTE         FME
## Cumulative Return      -0.11261700 -0.01710425 0.06756354 -0.05933989
## Annualized Return      -0.05207404 -0.00769239 0.02969616 -0.02700988
## Annualized Sharp Ratio -1.42351353 -0.27099580 0.78073388 -0.34067096
## Calmar Ratio           -0.41625217 -0.15322338 0.85780598 -0.20131106
##                               FRE
## Cumulative Return      0.015778219
## Annualized Return      0.007031836
## Annualized Sharp Ratio 0.069007518
## Calmar Ratio           0.062823839
```

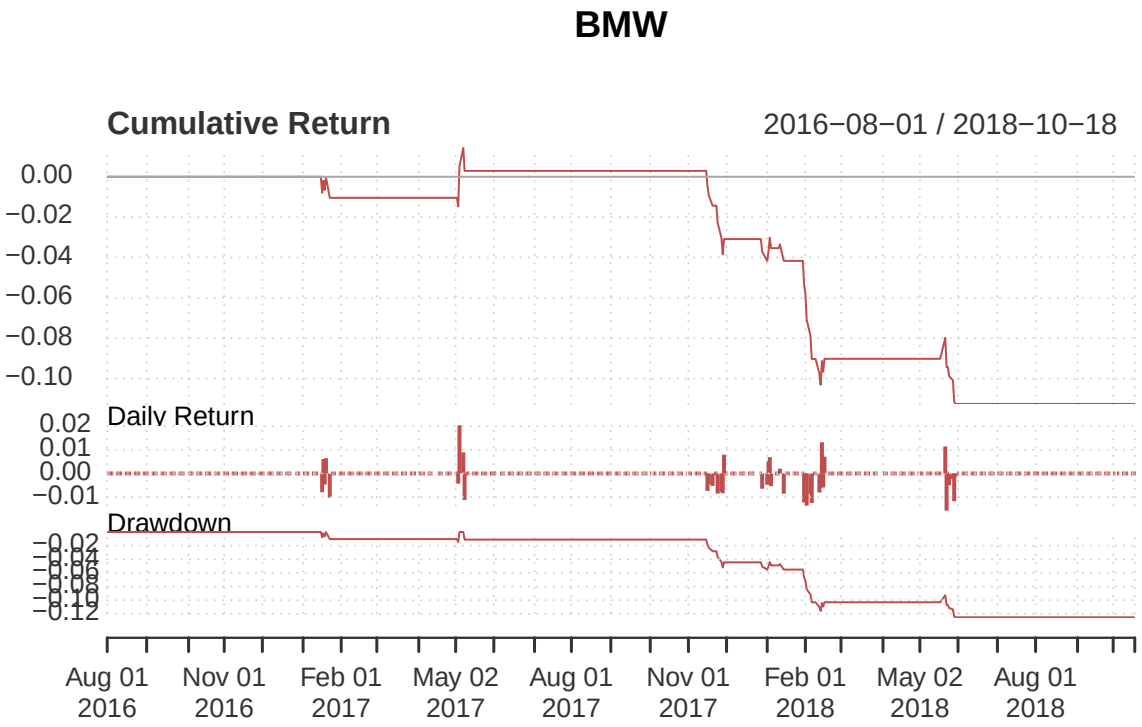
```
## Risk Metrics
```

```
##                            BMW          DAI         DTE          FME
## Annualized StdDev  0.03658134  0.028385642  0.03803622  0.079284348
## Max Drawdown       0.12510214  0.050203760  0.03461874  0.134169850
## Value-at-Risk      0.00000000  0.000000000  0.00000000 -0.007344407
## Conditional VaR   -0.00761643 -0.004372618 -0.00504827 -0.014298697
##                            FRE
## Annualized StdDev  0.10189956
## Max Drawdown       0.11192942
## Value-at-Risk     -0.00471046
## Conditional VaR   -0.01498715
```

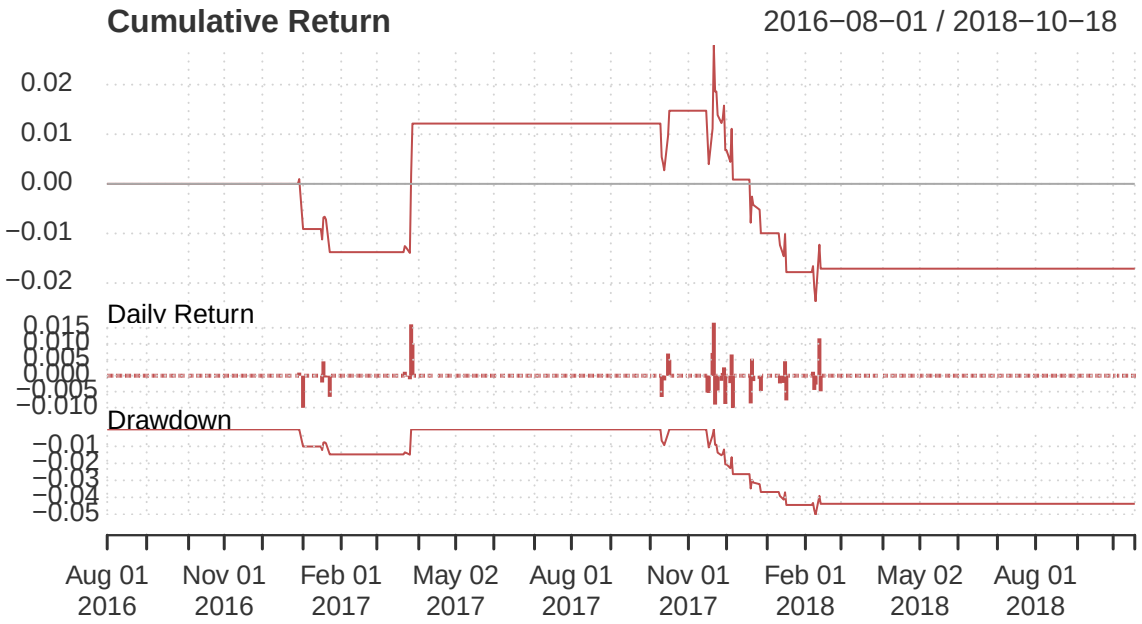
Step 6: Visualize returns of the trading strategy for every instrument

The following plots visualizes some important information about the returns of the strategy applied to the instruments on the instrumentlist. The cummulative return over the entire period, the daily returns and the drawdown for each instrument is illustrated.

```
for (i in 1:ncol(rets)){
  charts.PerformanceSummary(rets[,i],colorset=rainbow12equal,lwd=1,main=substr(colnames(rets[,i]),1,3))
}
```



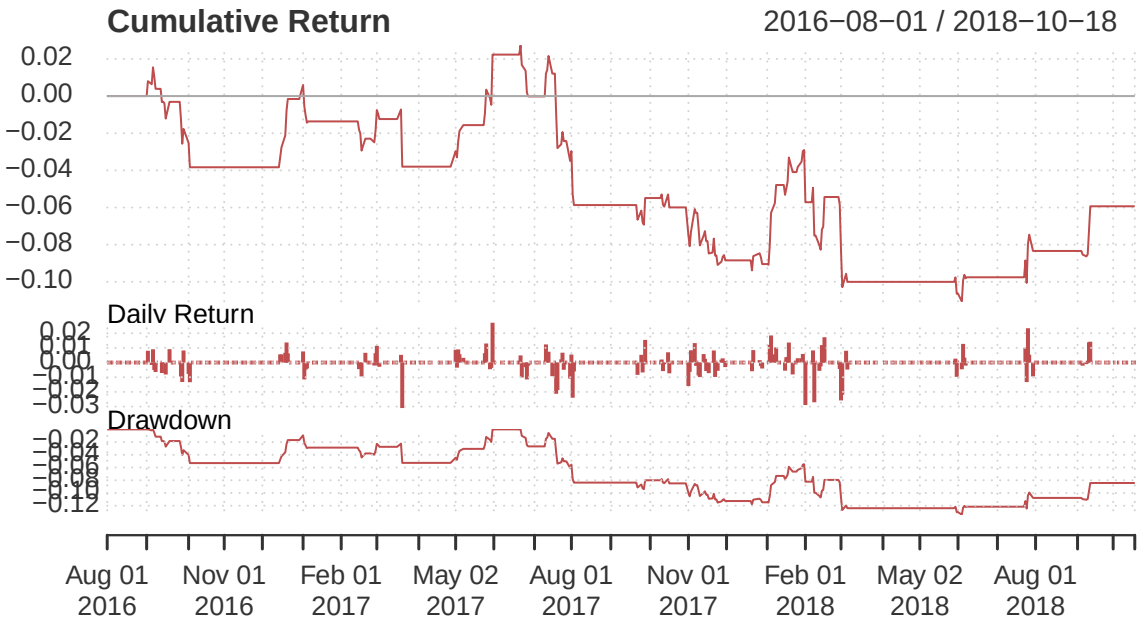
DAI

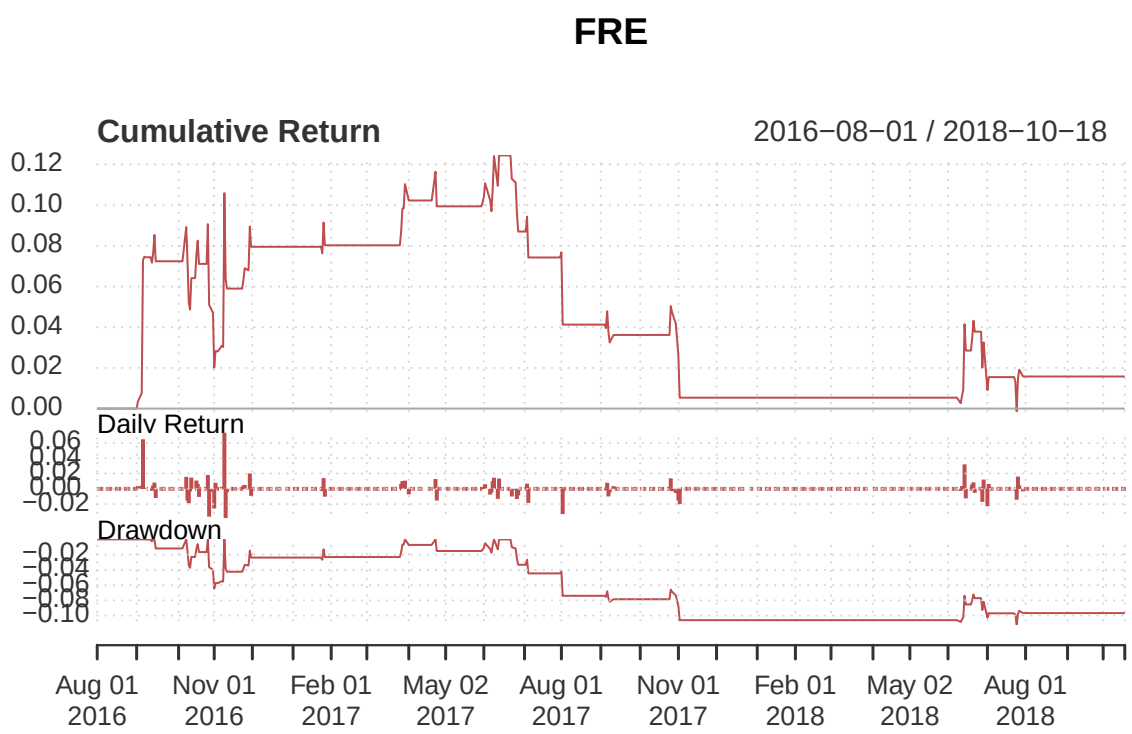

Cumulative Return
2016−08−01 / 2018−10−18
0.02
0.01
0.00
−0.01
−0.02
Dailv Return
0.015
0.010
0.005
0.000
−0.005
−0.010
Drawdown
−0.01
−0.02
−0.03
−0.04
−0.05
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018

DTE


Cumulative Return
2016−08−01 / 2018−10−18
0.08
0.06
0.04
0.02
0.00
−0.02
Dailv Return
0.02
0.01
0.00
−0.01
Drawdown
−0.005
−0.030
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018

FME


Cumulative Return
2016−08−01 / 2018−10−18
0.02
0.00
−0.02
−0.04
−0.06
−0.08
−0.10
Dailv Return
0.02
0.01
0.00
−0.01
−0.02
−0.03
Drawdown
−0.02
−0.12
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018

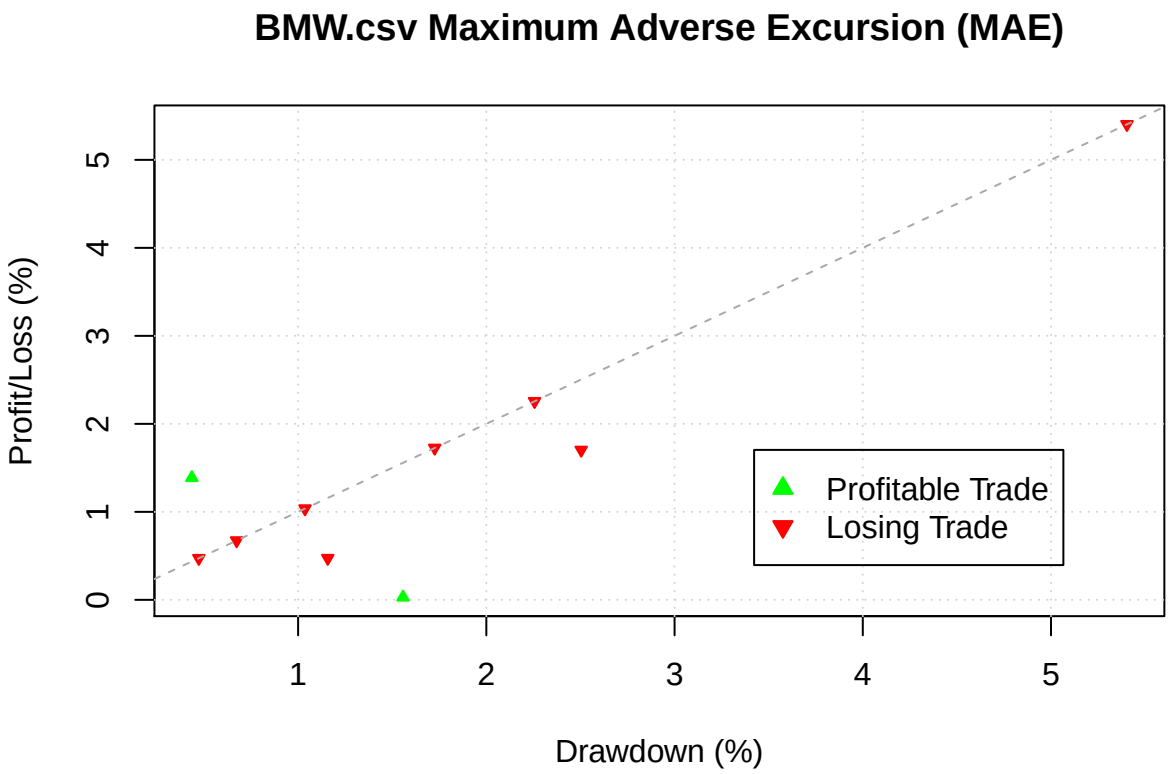


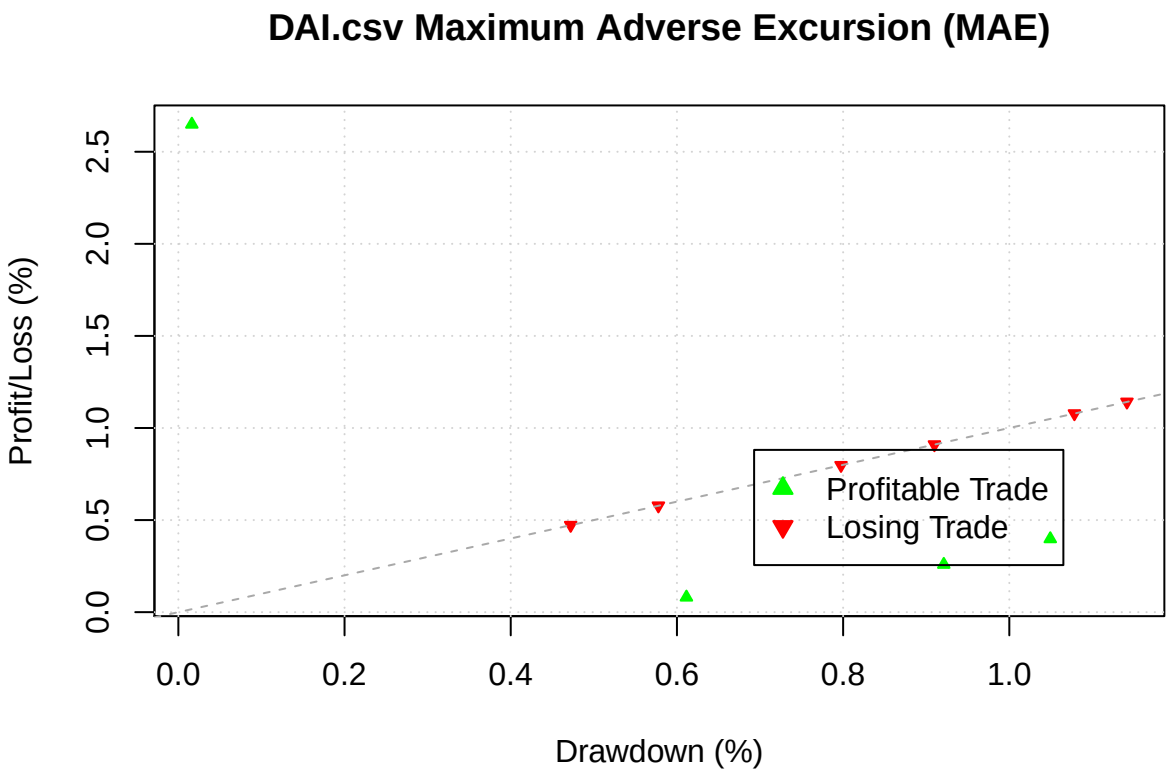
Step 7: Could we have performed better or even worse? MAE / MFE

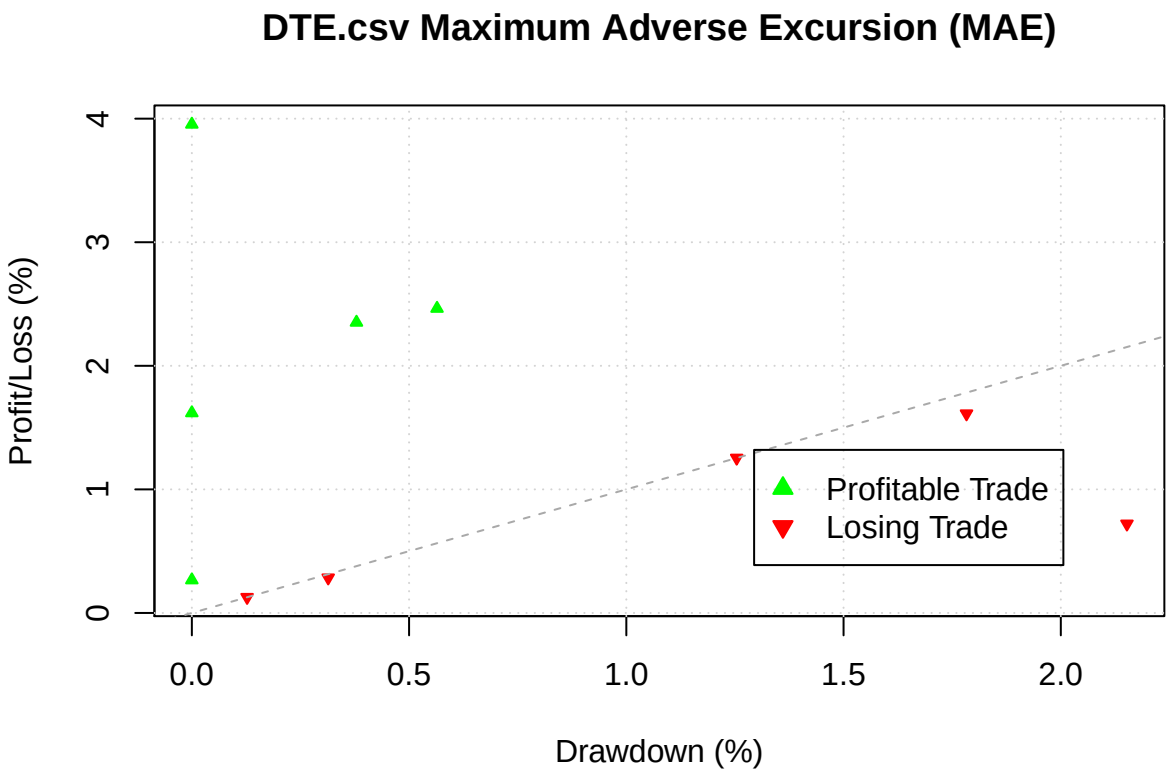
The efficiency of a strategy is also dependent on the question whether we could have performed better or worse. This can be shown with the maximum adverse excursion and the maximum favorable excursion. The Maximum Adverse excursion shows how much we could have lost at most during the trade, while the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

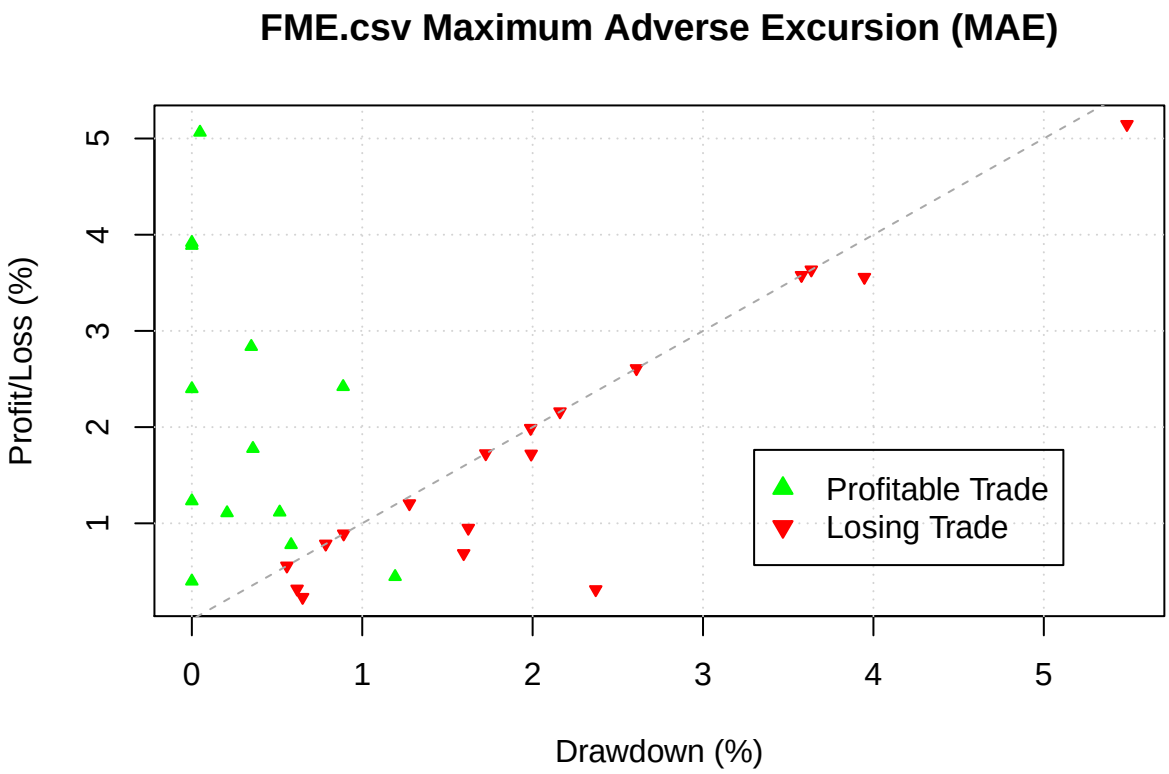
MAE

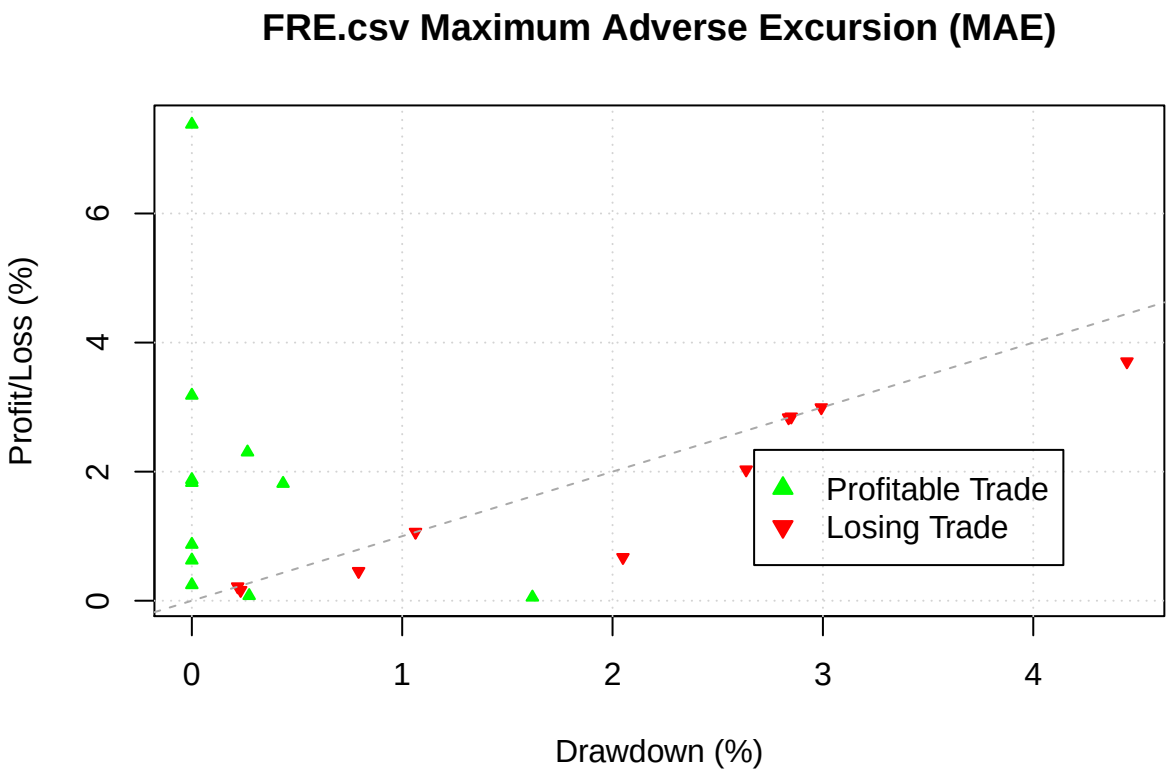
```
for (instrument in instrumentlist){
  chart.ME(portfolioname, instrument, type="MAE", scale="percent")
}
```


BMW.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
0
1
2
3
4
5
1
2
3
4
5
Drawdown (%)
Profitable Trade
Losing Trade


DAI.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.2
0.4
0.6
0.8
1.0
Drawdown (%)
Profitable Trade
Losing Trade

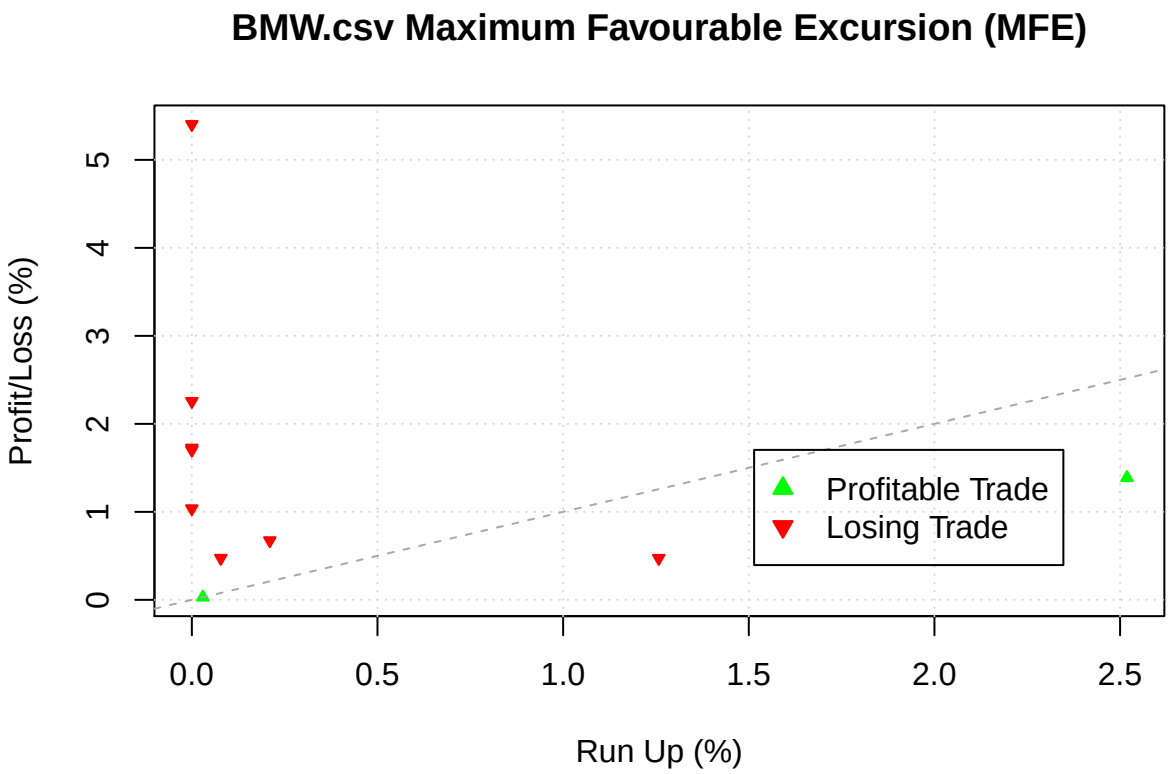

DTE.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
0
1
2
3
4
0.0
0.5
1.0
1.5
2.0
Drawdown (%)
Profitable Trade
Losing Trade

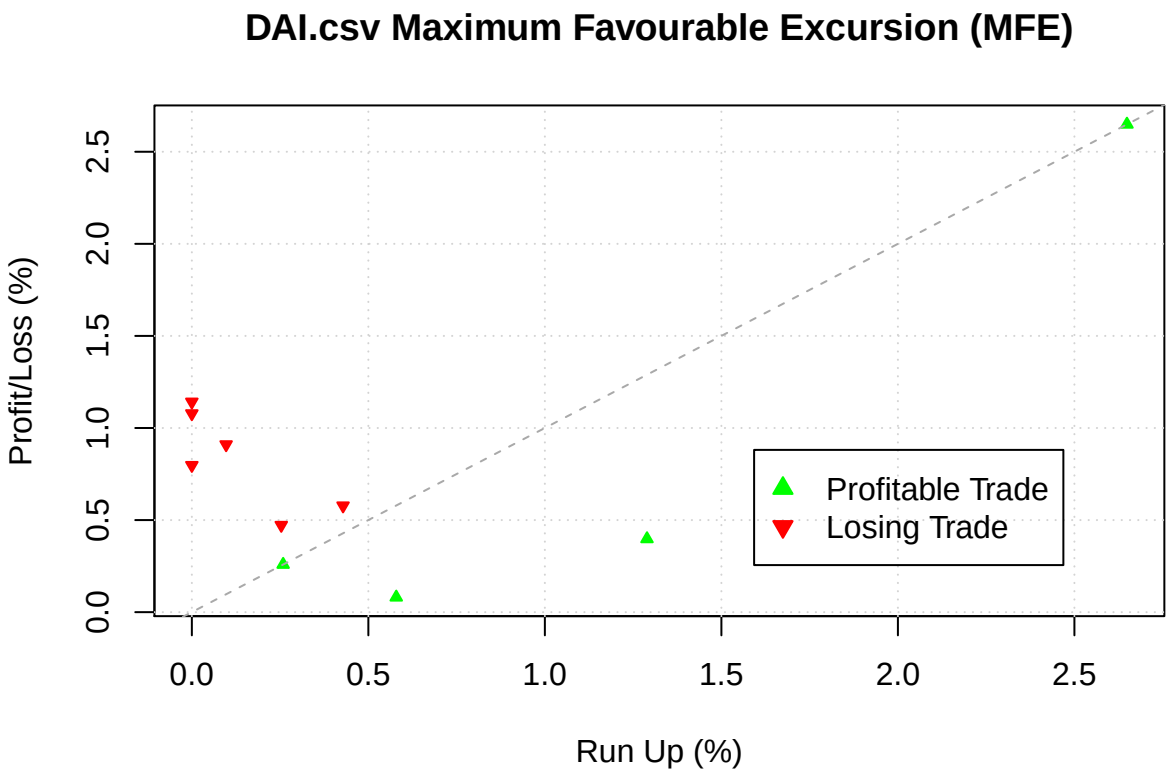

FME.csv Maximum Adverse Excursion (MAE)
Profit/Loss (%)
1
2
3
4
5
0
1
2
3
4
5
Drawdown (%)
Profitable Trade
Losing Trade

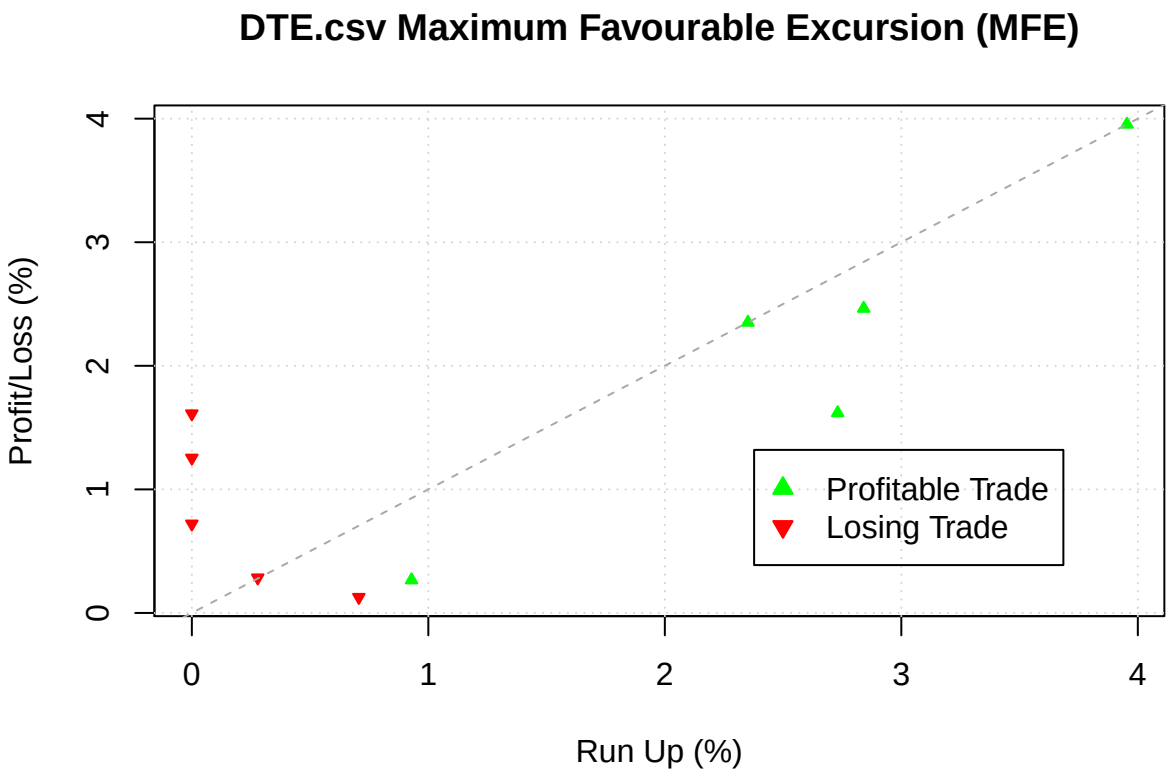


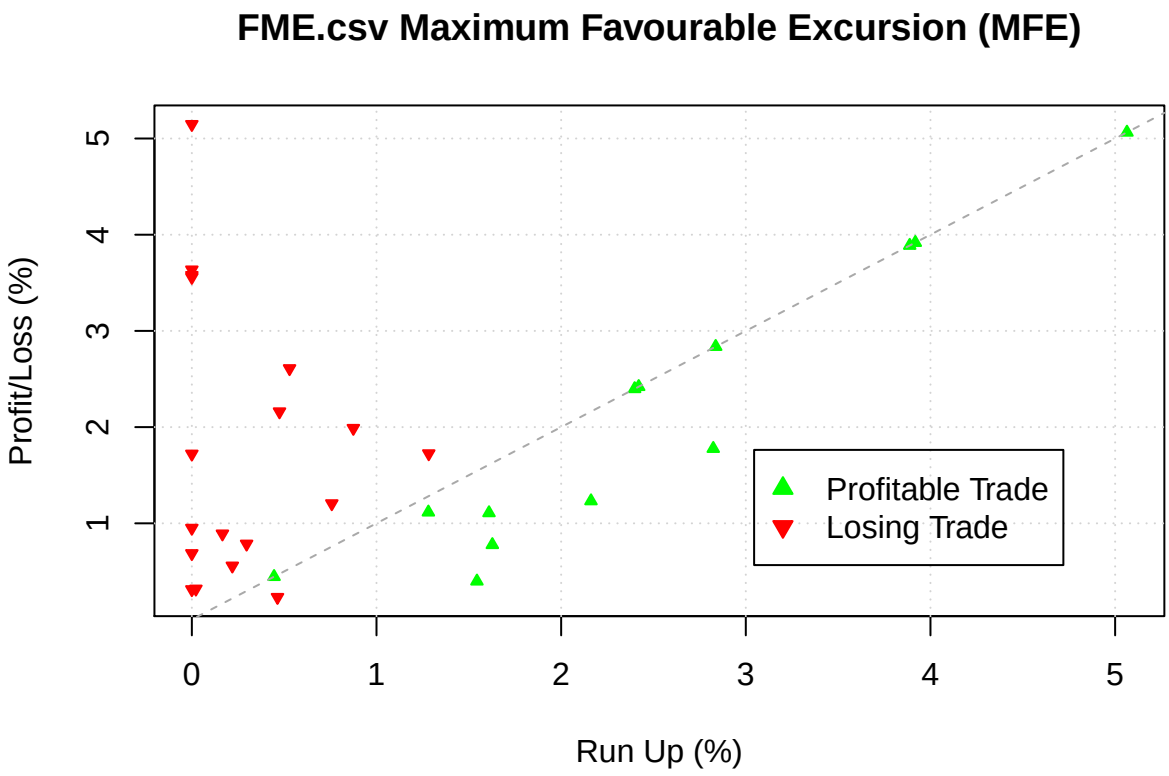
MFE

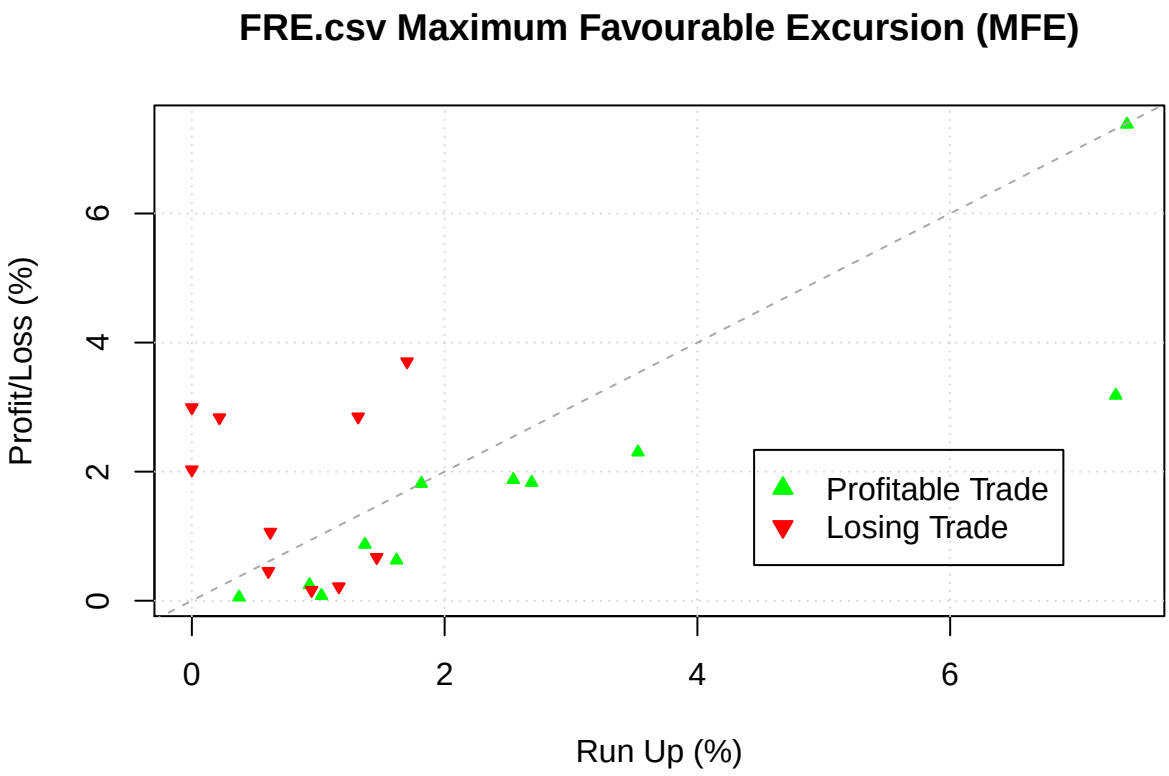
```
for (instrument in instrumentlist){
  chart.ME(portfolioname, instrument, type="MFE", scale="percent")
}
```


BMW.csv Maximum Favourable Excursion (MFE)
Profit/Loss (%)
0
1
2
3
4
5
0.0
0.5
1.0
1.5
2.0
2.5
Run Up (%)
Profitable Trade
Losing Trade


DAI.csv Maximum Favourable Excursion (MFE)
Profit/Loss (%)
0.0
0.5
1.0
1.5
2.0
2.5
Run Up (%)
0.0
0.5
1.0
1.5
2.0
2.5
Profitable Trade
Losing Trade


DTE.csv Maximum Favourable Excursion (MFE)
Profit/Loss (%)
Run Up (%)
0
1
2
3
4
Profitable Trade
Losing Trade


FME.csv Maximum Favourable Excursion (MFE)
Profit/Loss (%)
Run Up (%)
0
1
2
3
4
5
Profitable Trade
Losing Trade



Compare with Buy and Hold Strategy

In order to compare the trading strategy properly we need to define a benchmark against which we can measure the results. 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 stocks from the DAX (German Stocks Market Index) are used in the strategy, the buy and hold strategy is performed on an DAX ETF.

Step 1: Perform the Buy and Hold Strategy

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

# The Buy and hold instrument is loaded
LoadCourseFile(BuyHoldDirectory,BuyHoldInstrument,debugme=TRUE,dates=daterange)
# The Buy and hold instrument is initialized
stock(BuyHoldInstrument,currency="EUR")

BuyHoldSymbol<-get(BuyHoldInstrument)

# The portfolio and account "buyhold" is initialized
initPortf("buyhold",
          BuyHoldInstrument,
```

```
           initDate=initdate,
           currency="EUR")
initAcct("buyhold",
         portfolios="buyhold",
         initDate=initdate,
         initEq=startCapital,
         currency="EUR")

# 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(startCapital/closeprice))

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

# 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=transactionCost)

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

Step 2: Visualize the Buy and Hold strategy

We can see that we hold the position from the first until the last date. The cumulative profits are visualized by the green line.

```
chart.Posn("buyhold",Symbol=BuyHoldInstrument, theme=myTheme)
```

Step 3: Compare the returns of the trading strategy with the buy and hold strategy

In order to compare the results of both strategies, we calculat the returns for the buy and hold strategy and combine them with the returns of the trading strategy which were calculated before.

```
rets.bh <- PortfReturns(Account='buyhold')
returns <- cbind(rets,rets.bh)
colnames(returns) <- substr(colnames(returns),1,3)
```

In order to compare the performance between the strategy investment in one of the stocks and the alternative buy and hold strategy of the DAX ETF, the following graphs visualize the relative performance between each of the stocks used and the DAX ETF.

```
table.Stats(returns)
```

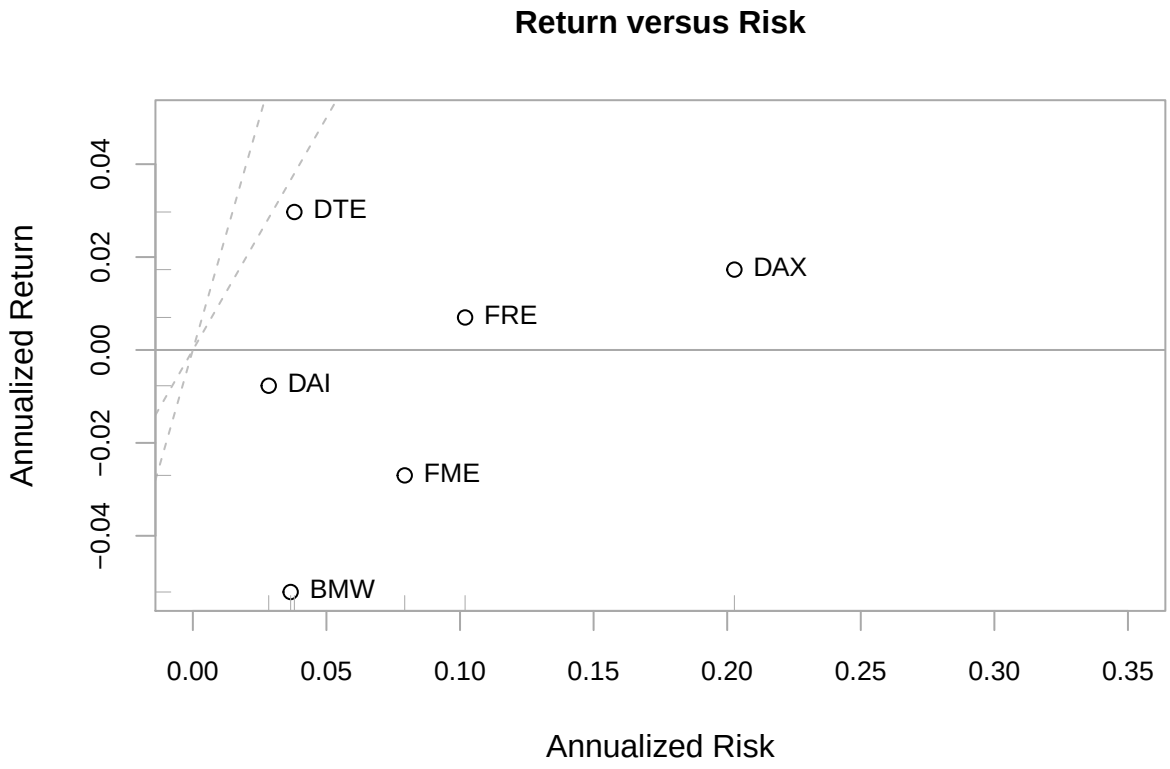
```
##                       BMW      DAI      DTE      FME      FRE      DAX
## Observations     563.0000 563.0000 563.0000 563.0000 563.0000 958.0000
## NAs              400.0000 400.0000 400.0000 400.0000 400.0000   5.0000
## Minimum           -0.0157  -0.0101  -0.0130  -0.0310  -0.0382  -0.0694
## Quartile 1         0.0000   0.0000   0.0000   0.0000   0.0000  -0.0066
## Median             0.0000   0.0000   0.0000   0.0000   0.0000   0.0008
## Arithmetic Mean   -0.0002   0.0000   0.0001  -0.0001   0.0000   0.0001
## Geometric Mean    -0.0002   0.0000   0.0001  -0.0001   0.0000   0.0001
## Quartile 3         0.0000   0.0000   0.0000   0.0000   0.0000   0.0071
## Maximum            0.0203   0.0166   0.0265   0.0272   0.0734   0.0492
## SE Mean            0.0001   0.0001   0.0001   0.0002   0.0003   0.0004
## LCL Mean (0.95)   -0.0004  -0.0002  -0.0001  -0.0005  -0.0005  -0.0007
## UCL Mean (0.95)    0.0000   0.0001   0.0003   0.0003   0.0006   0.0010
```

```
## Variance          0.0000   0.0000   0.0000   0.0000   0.0000   0.0002
## Stdev             0.0023   0.0018   0.0024   0.0050   0.0064   0.0128
## Skewness         -0.5345   2.1870   4.6079  -1.0310   3.5266  -0.3220
## Kurtosis         26.2969  37.6379  52.8325  12.2836  54.8098   1.7271
```

```r
table.AnnualizedReturns(returns)
```

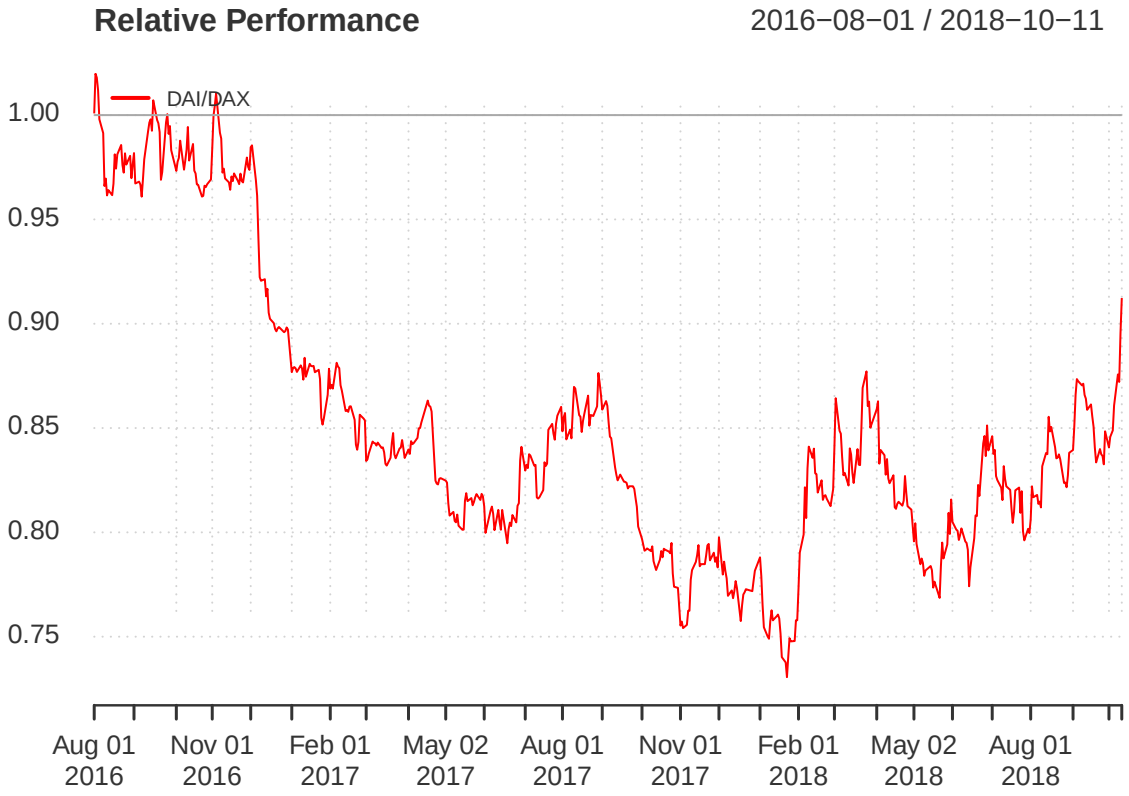
```
##                                BMW     DAI    DTE     FME    FRE    DAX
## Annualized Return          -0.0521 -0.0077 0.0297 -0.0270 0.0070 0.0173
## Annualized Std Dev          0.0366  0.0284 0.0380  0.0793 0.1019 0.2027
## Annualized Sharpe (Rf=0%) -1.4235 -0.2710 0.7807 -0.3407 0.0690 0.0855
```

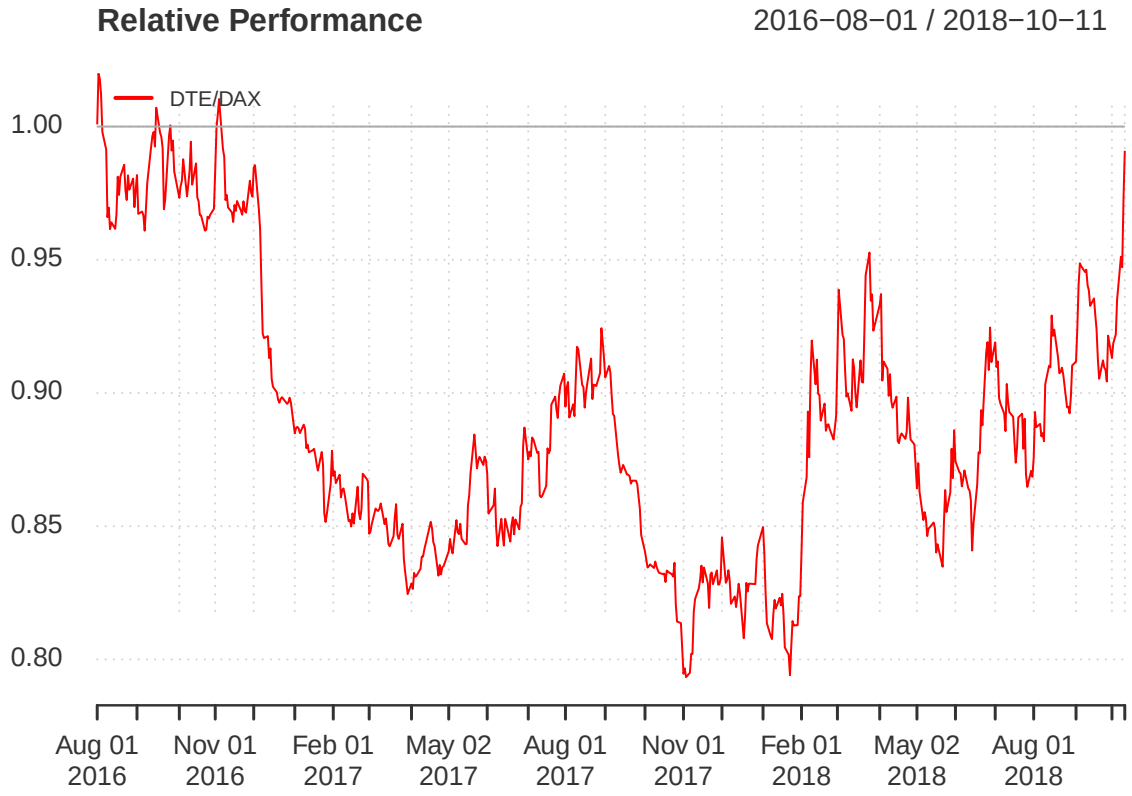
```r
# charts.PerformanceSummary(returns,geometric=FALSE,wealth.index=TRUE)
chart.RiskReturnScatter(returns,Rf=0,
                        add.sharpe=c(1,2),
                        xlim=c(0,0.35),
                        main="Return versus Risk"
                        )
```

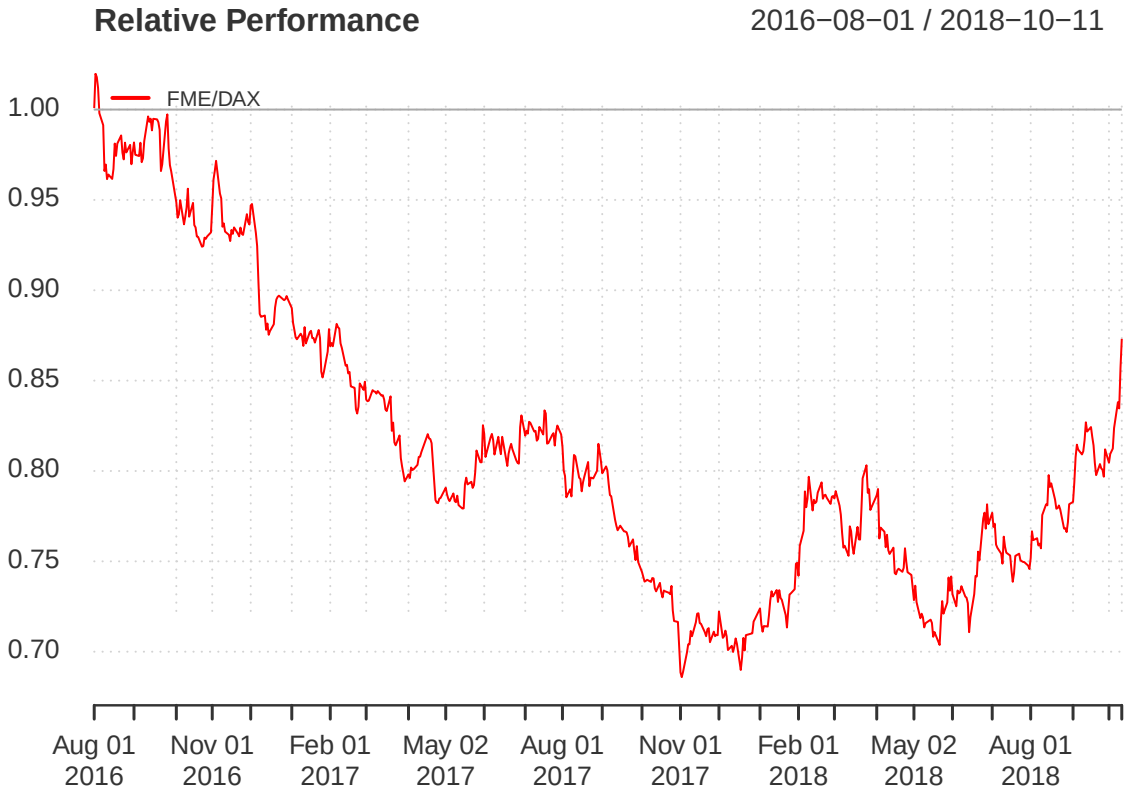


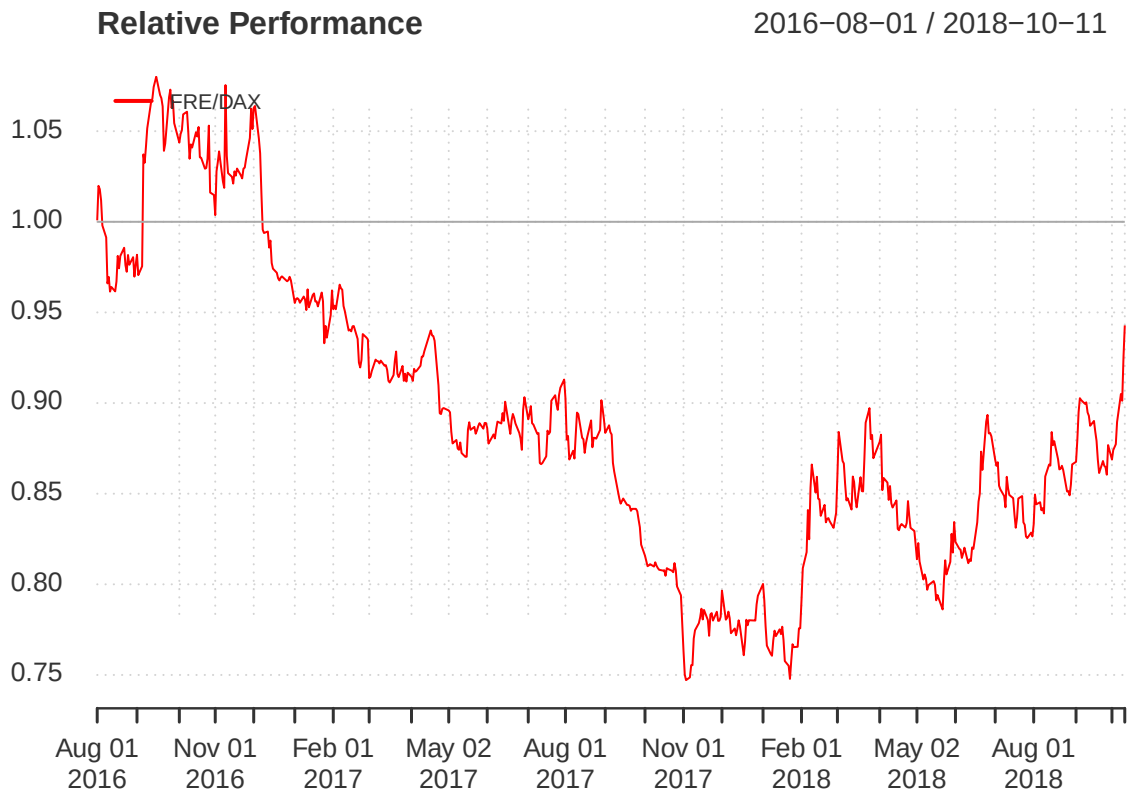

```r
for (i in colnames(returns[,-which(names(returns) == "DAX")])){

  print(chart.RelativePerformance(returns[,i],returns[,"DAX"],
                          colorset=c("red","blue"),
                          lwd=1,
                          legend.loc="topleft"))
}
```


Relative Performance
2016−08−01 / 2018−10−11
BMW/DAX
1.00
0.95
0.90
0.85
0.80
0.75
0.70
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018


Relative Performance
2016−08−01 / 2018−10−11
DAI/DAX
1.00
0.95
0.90
0.85
0.80
0.75
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018


Relative Performance
2016−08−01 / 2018−10−11
DTE/DAX
1.00
0.95
0.90
0.85
0.80
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018

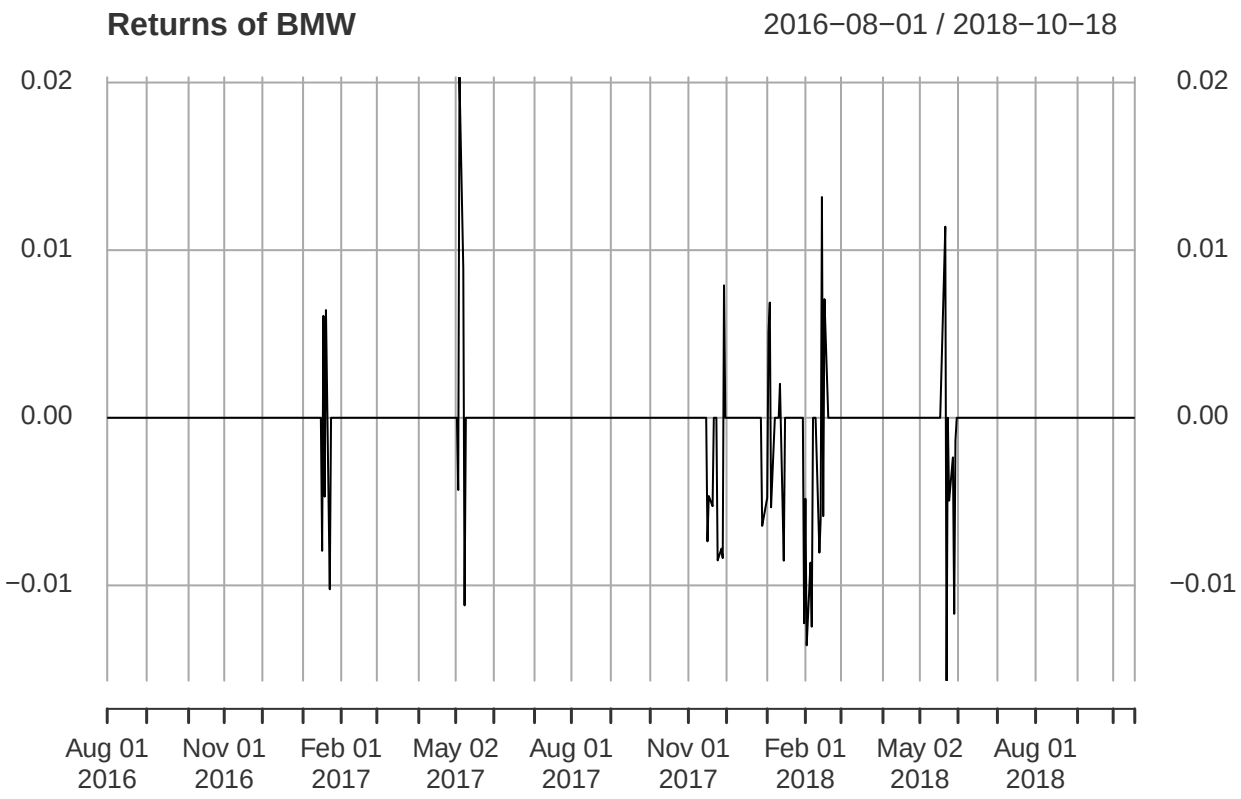

Relative Performance
2016−08−01 / 2018−10−11
FME/DAX
1.00
0.95
0.90
0.85
0.80
0.75
0.70
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018

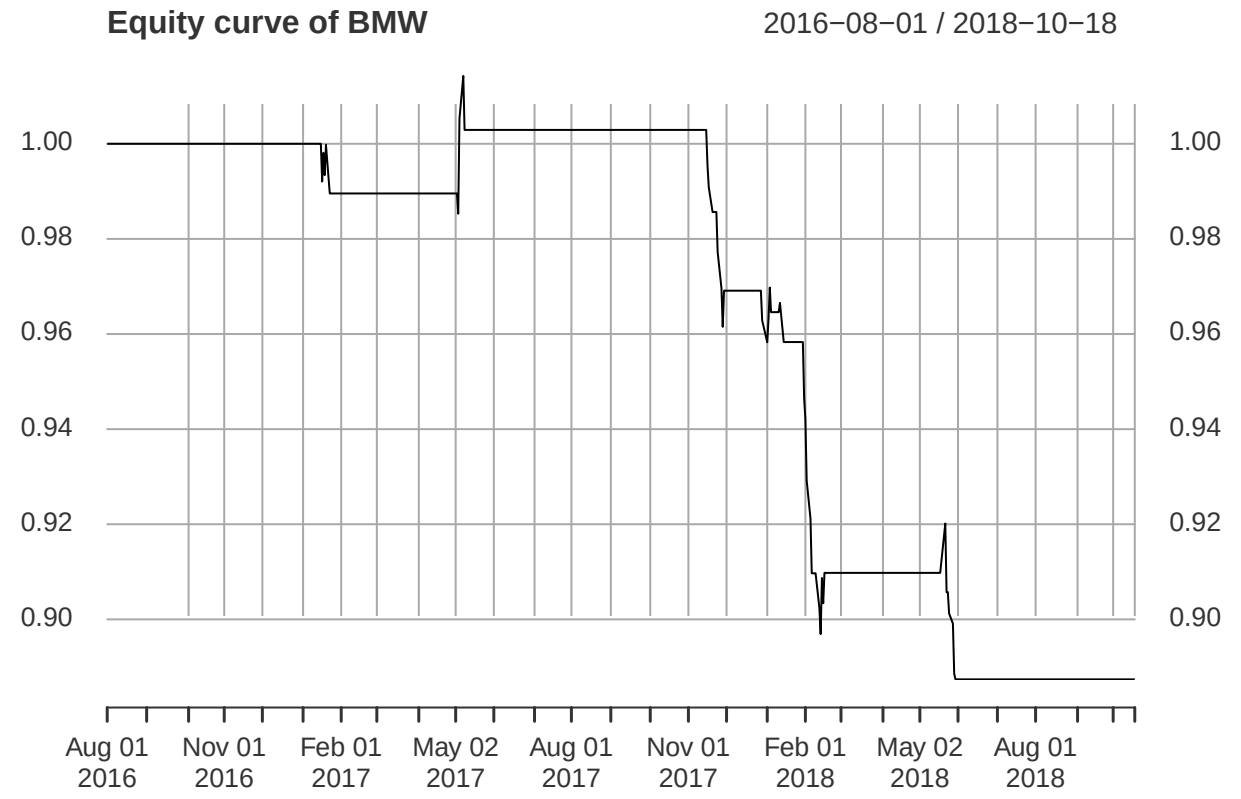


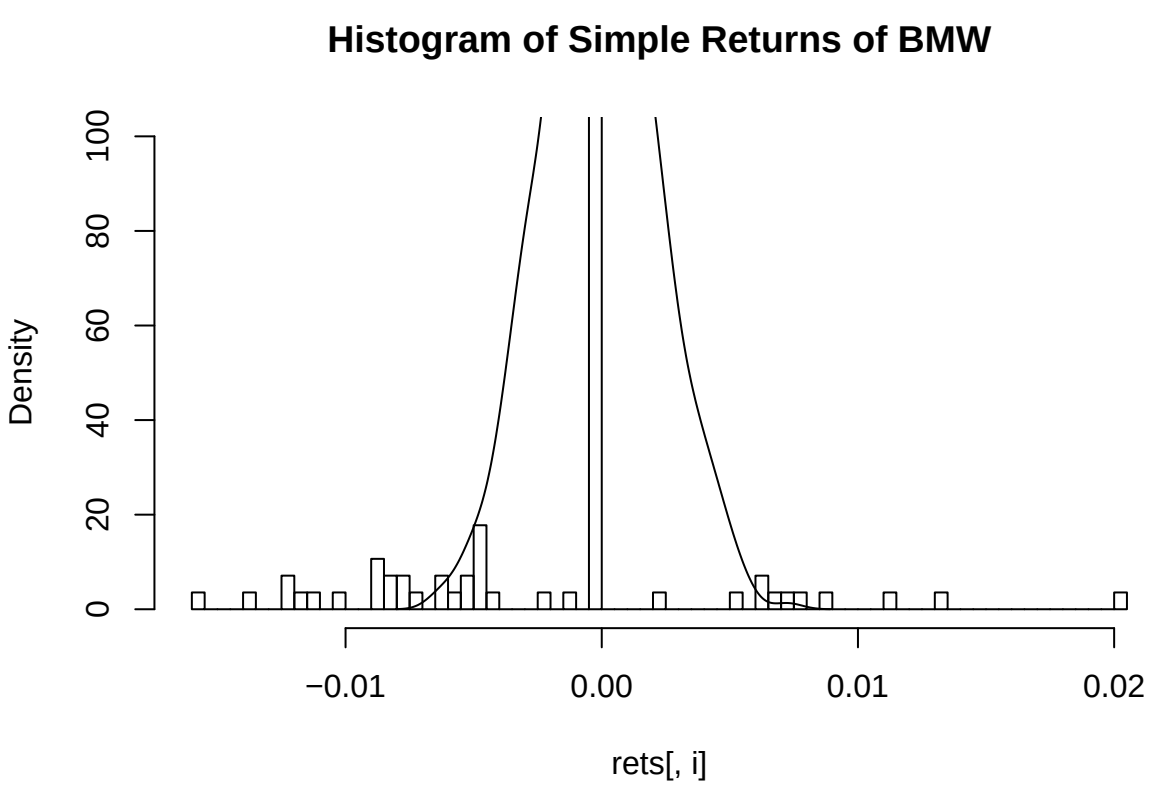
Calculations and visualizations based on returns

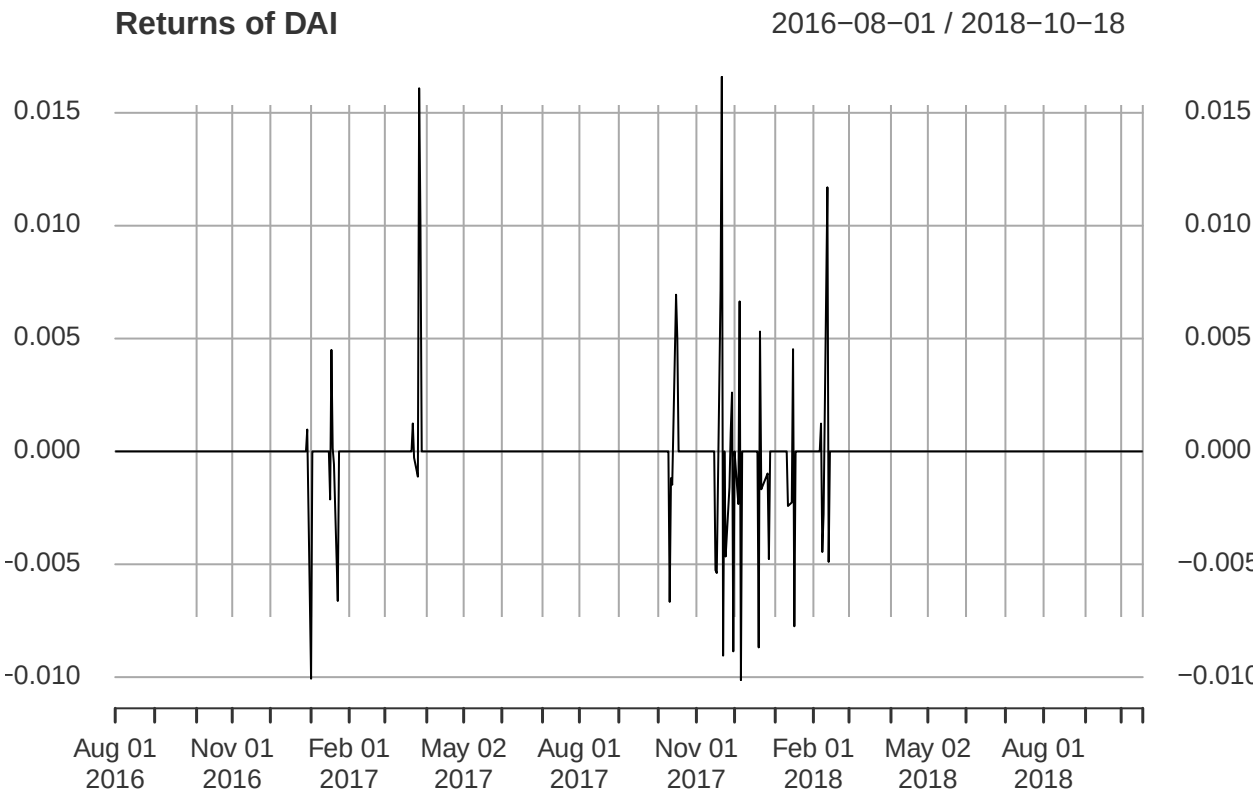
The visualization of returns, equity curve, value at risk and some other measure which can be calculated based on the returns of the trading strategy, can be observed in the following graphs.

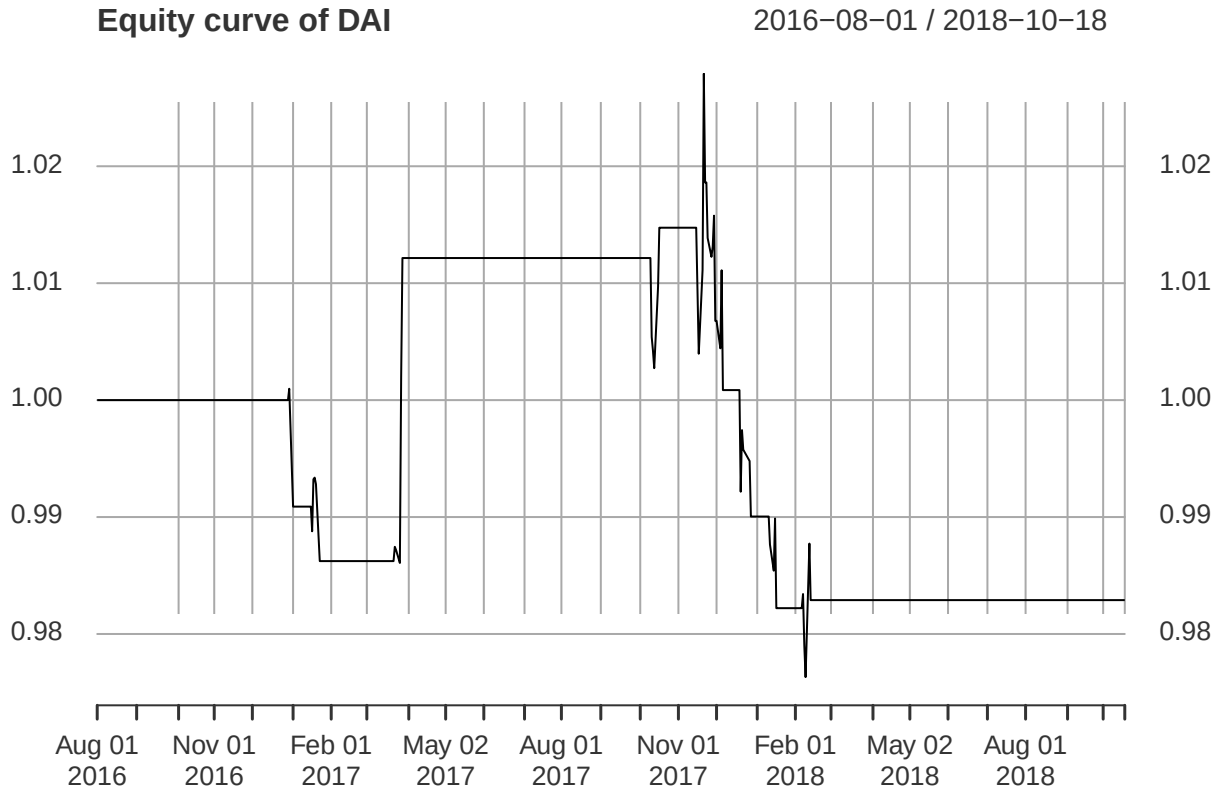
Returns and Equity Curve

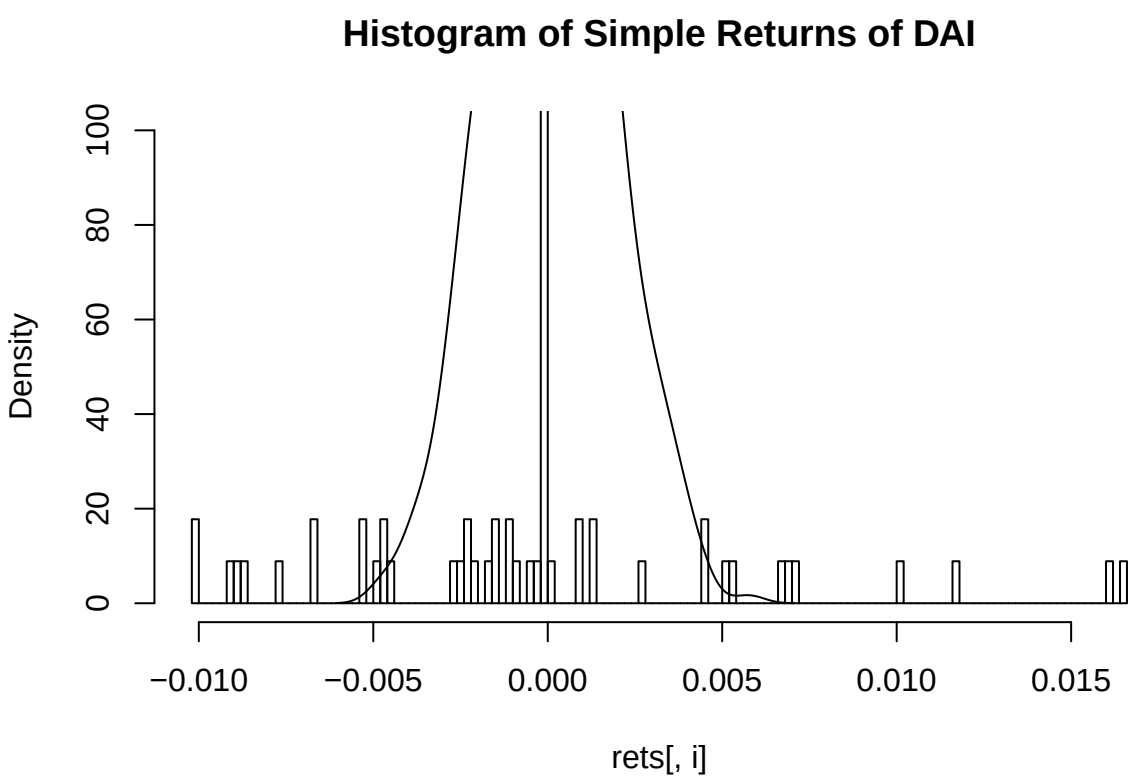


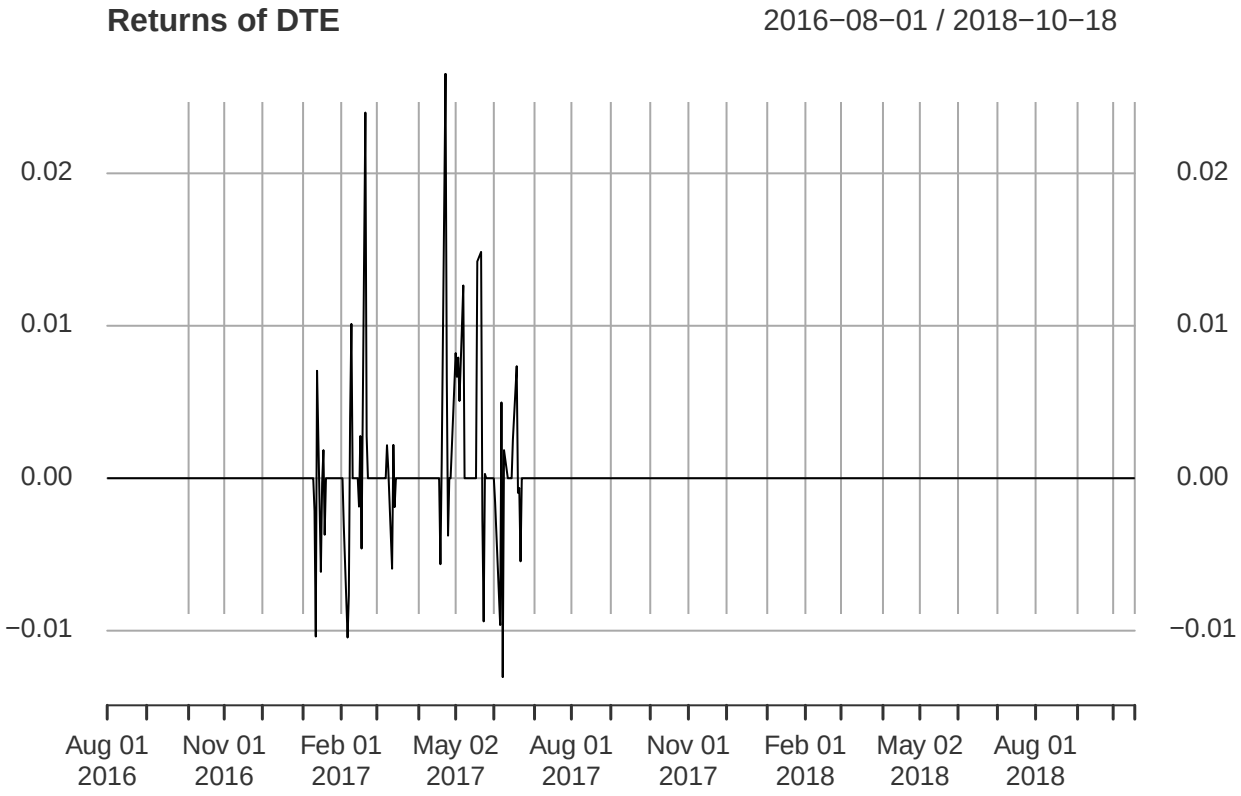


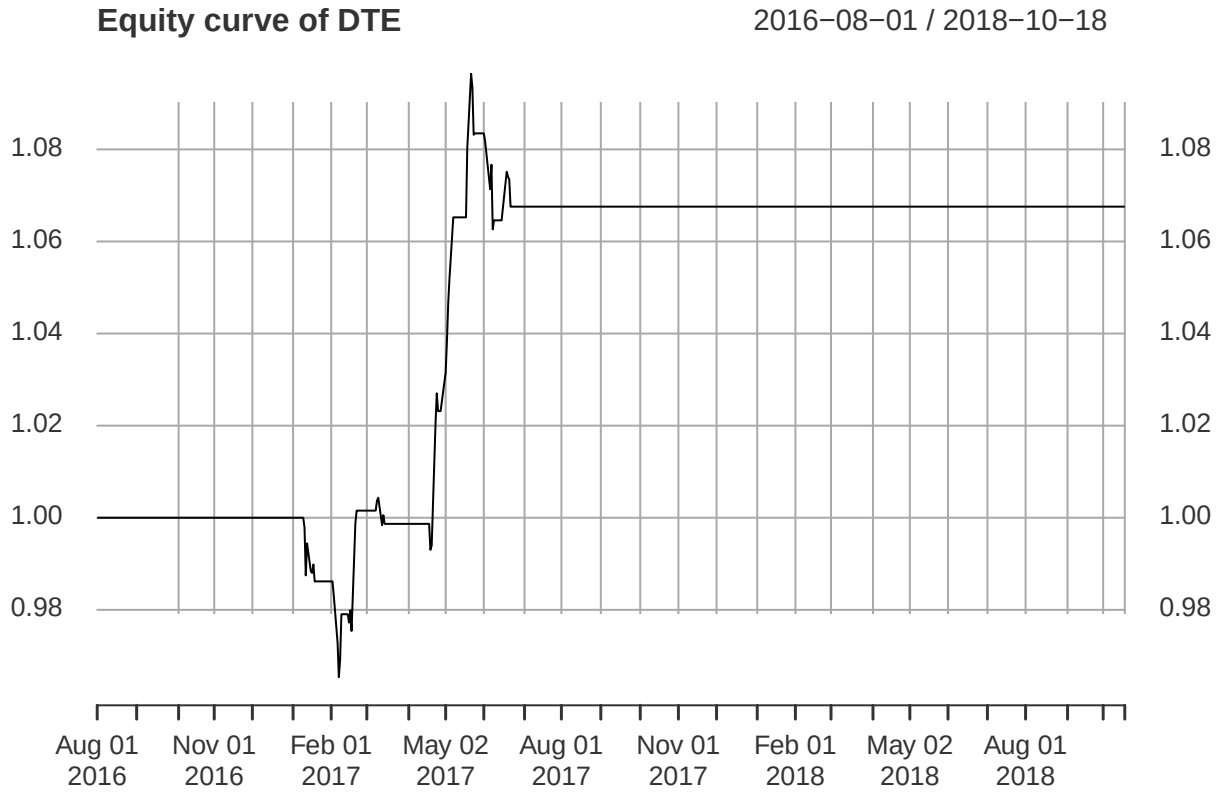

Histogram of Simple Returns of BMW
Density
0
20
40
60
80
100
−0.01
0.00
0.01
0.02
rets[, i]

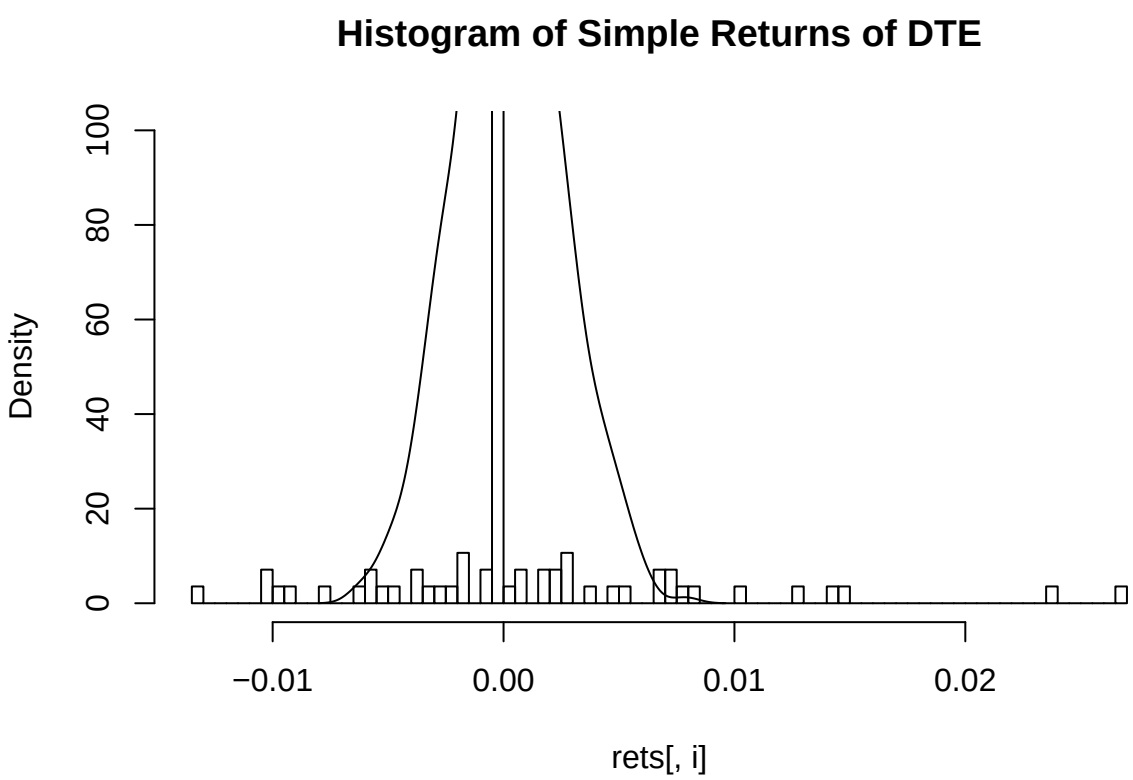

Returns of DAI
2016−08−01 / 2018−10−18
0.015
0.010
0.005
0.000
-0.005
-0.010
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018

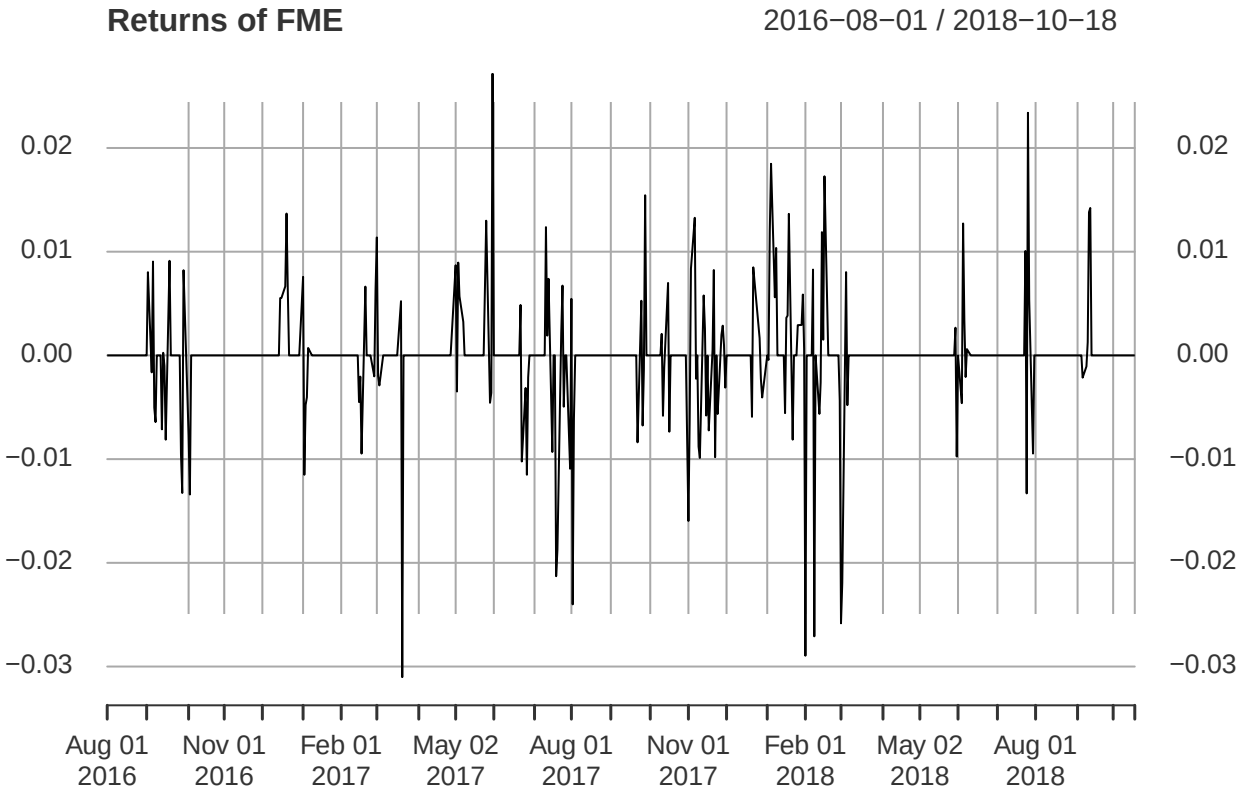

Equity curve of DAI
2016−08−01 / 2018−10−18
1.02
1.01
1.00
0.99
0.98
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018

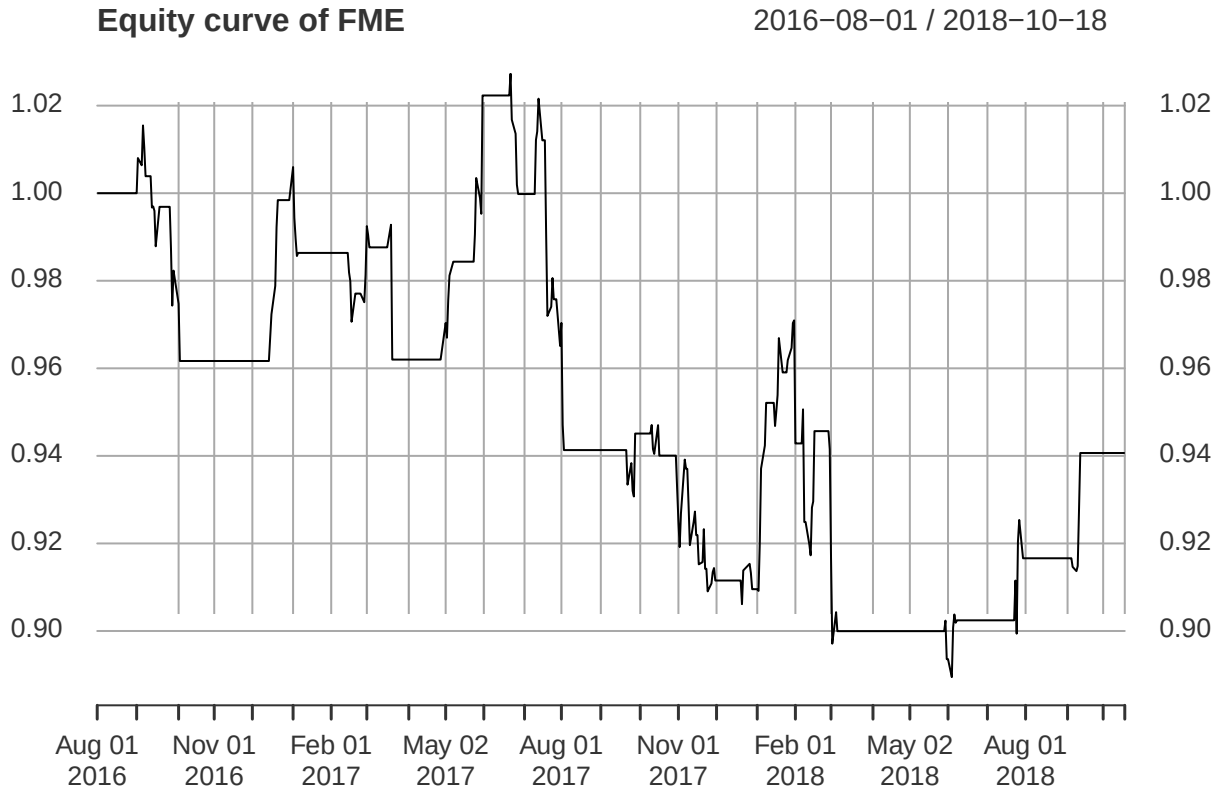

Histogram of Simple Returns of DAI
Density
0
20
40
60
80
100
−0.010
−0.005
0.000
0.005
0.010
0.015
rets[, i]

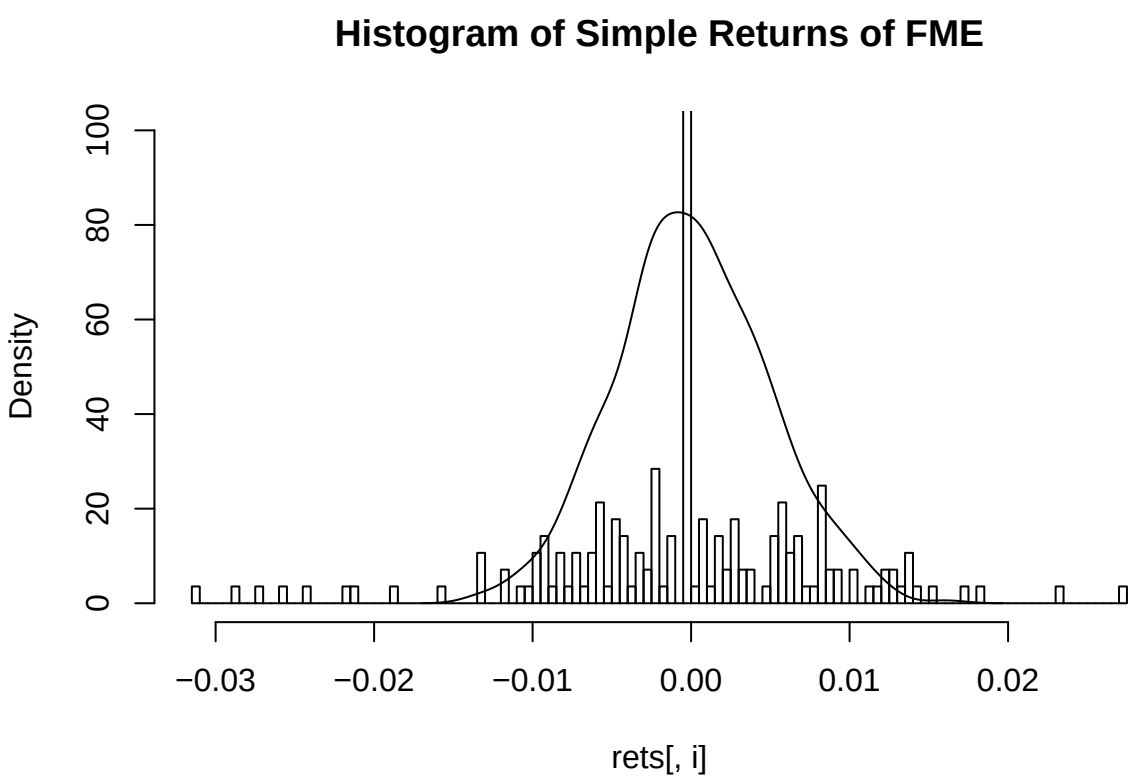

Returns of DTE
2016−08−01 / 2018−10−18
0.02
0.01
0.00
−0.01
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018

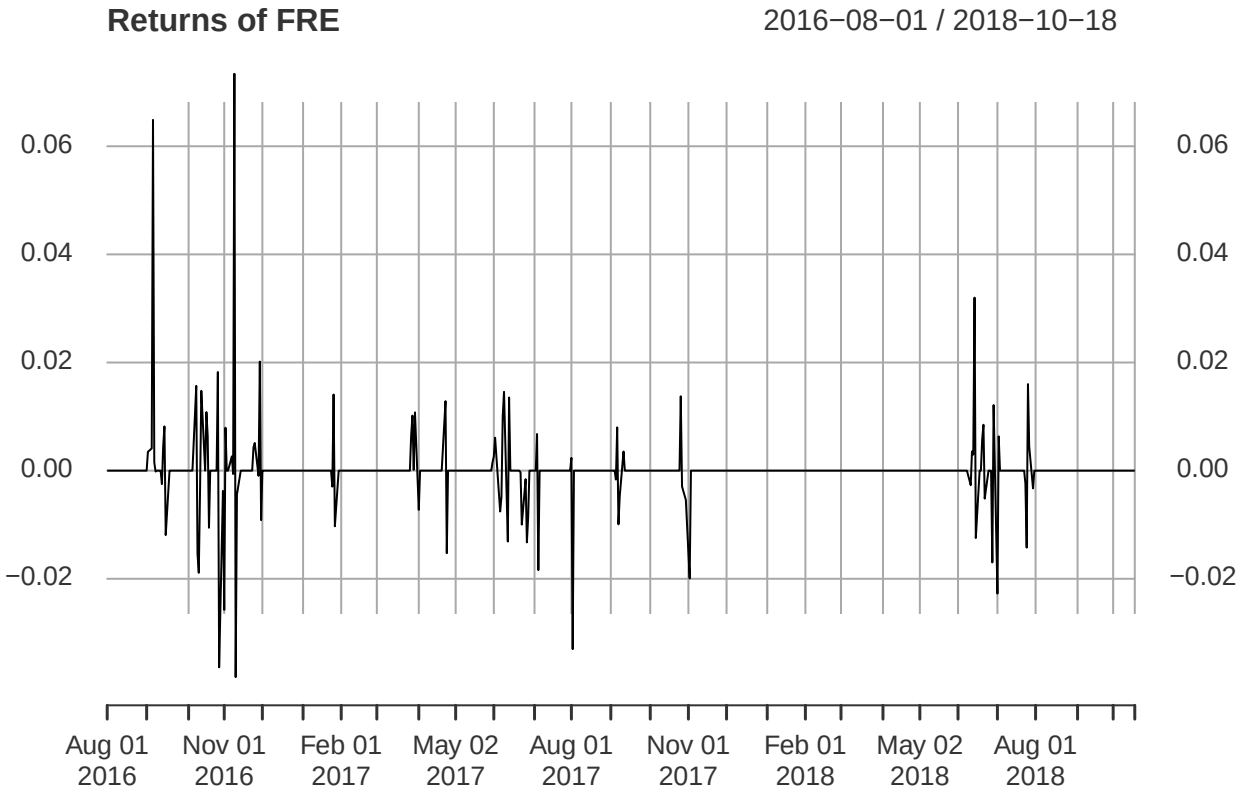

Equity curve of DTE
2016−08−01 / 2018−10−18
1.08
1.06
1.04
1.02
1.00
0.98
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018

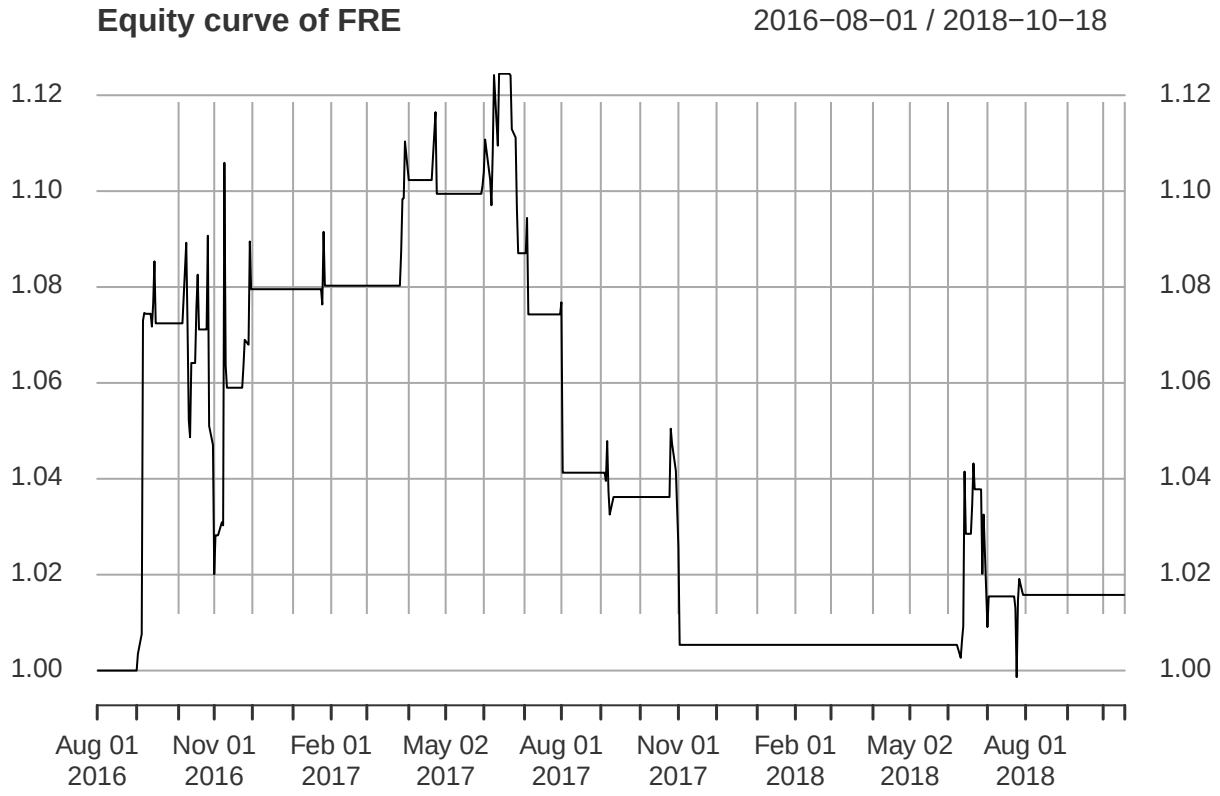

Histogram of Simple Returns of DTE
Density
0
20
40
60
80
100
−0.01
0.00
0.01
0.02
rets[, i]

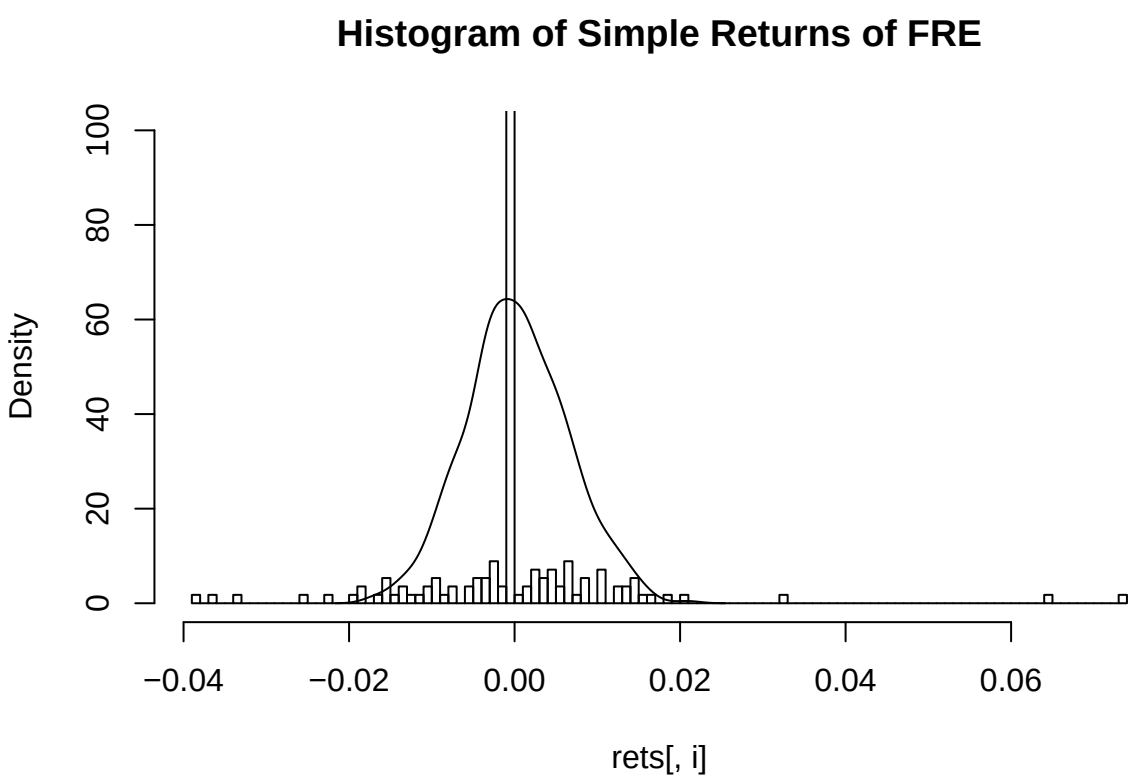

Returns of FME
2016−08−01 / 2018−10−18
0.02
0.01
0.00
−0.01
−0.02
−0.03
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018


Equity curve of FME
2016−08−01 / 2018−10−18
1.02
1.00
0.98
0.96
0.94
0.92
0.90
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018


Histogram of Simple Returns of FME
Density
0
20
40
60
80
100
−0.03
−0.02
−0.01
0.00
0.01
0.02
rets[, i]


Returns of FRE
2016−08−01 / 2018−10−18
0.06
0.04
0.02
0.00
−0.02
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018


Equity curve of FRE
2016−08−01 / 2018−10−18
1.12
1.10
1.08
1.06
1.04
1.02
1.00
Aug 01 2016
Nov 01 2016
Feb 01 2017
May 02 2017
Aug 01 2017
Nov 01 2017
Feb 01 2018
May 02 2018
Aug 01 2018



Value at Risk

The historical "Value at Risk" measures the risk of loss for investments. More specifically, it describes the maximum amount we would expect to lose per day with a confidencelevel of 99%, given an investment of 10000.

```
## [1] "Investment: 10000"
```

```
## [1] "Confidence Level: 0.99"
```

```
## [1] "Given a 10000 investment in BMW we would expect a maximum loss of 105.97 per day."
## [1] "Given a 10000 investment in DAI we would expect a maximum loss of 70.77 per day."
## [1] "Given a 10000 investment in DTE we would expect a maximum loss of 67.29 per day."
## [1] "Given a 10000 investment in FME we would expect a maximum loss of 215.3 per day."
## [1] "Given a 10000 investment in FRE we would expect a maximum loss of 192.98 per day."
```

Conclusion and Suggestions