# Assignment 1

*Philipp Epstein*

*10/14/2018*

## Introduction

Blabla Blabla

## Trading Idea

Explain here the traiding idea, where I found it. What the theoretical foundation is –> Long Term positive autocorreleation –> Short term negative autocorreleation

Trade reverse patterns. When observing a fallback, trade to profit from the upward trend after the fallback. etc. . . .

### Implementation of the trading idea

Short paragraph about what is needed for the implementation

#### Part A: Initialization

#### General Setup

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

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

#### Setting the Variables

The variables set in the next section can be adjusted to test the strategy with different parameters.

```
# Set values:
startCapital <- 1e+6
transactionCost <- -20
daterange <- '2013::2018'
emaPeriod <- 200
maxHoldingPeriod <- 20

InstrumentDirectory <- "~/Desktop/R/DownloadedData/"
instrumentlist <- c("SAP.csv","DBK.csv")
BuyHoldDirectory <- InstrumentDirectory
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
```

### Initializing the portfolio

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

# 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

Loading the instrument, initializing it and adding the ema to the data.

```
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 (Long).
    # Smash Day (Long) is when Todays Close is below Yesterdays Low.
    if (CloseToday < LowYesterday) {
      # Smash Day (Long)

      #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.
        # (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 tomorrows 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 (Short) 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,
```

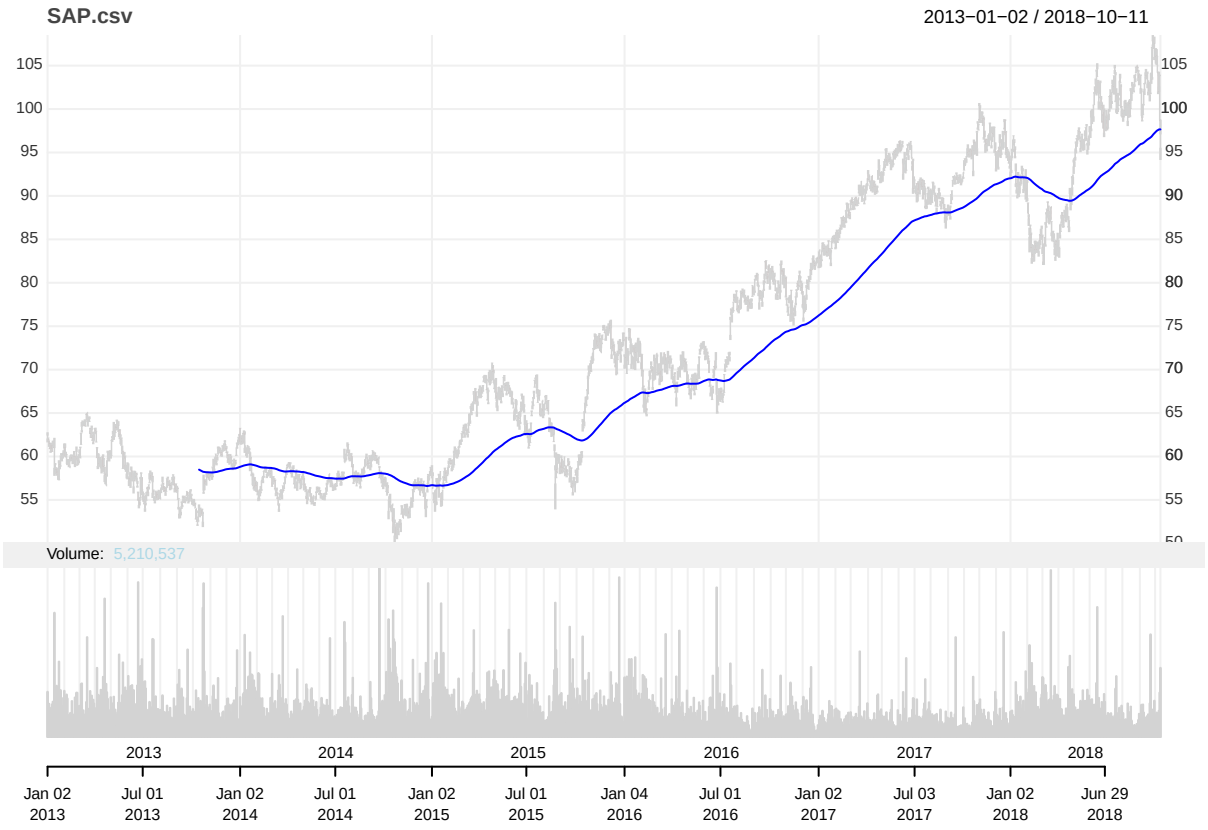
```
      TxnDate = time(symbol[i + 1]),
      TxnPrice = OpenTomorrow,
      TxnQty = -Position,
      TxnFees = transactionCost
    )

  } else {
    # Check Rule 3:

    # Sell Rule 3: If today was a smash day (short) place an order at todays
    # low price. (Buy tomorrow for 'price <= todays low')

    # Simulate this behaviour:

    # Option 1 to check: Check whether the open price tomorrow is below today's
    # low and add the transaction tomorrow at tomorrows 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 (Short)
    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(
```

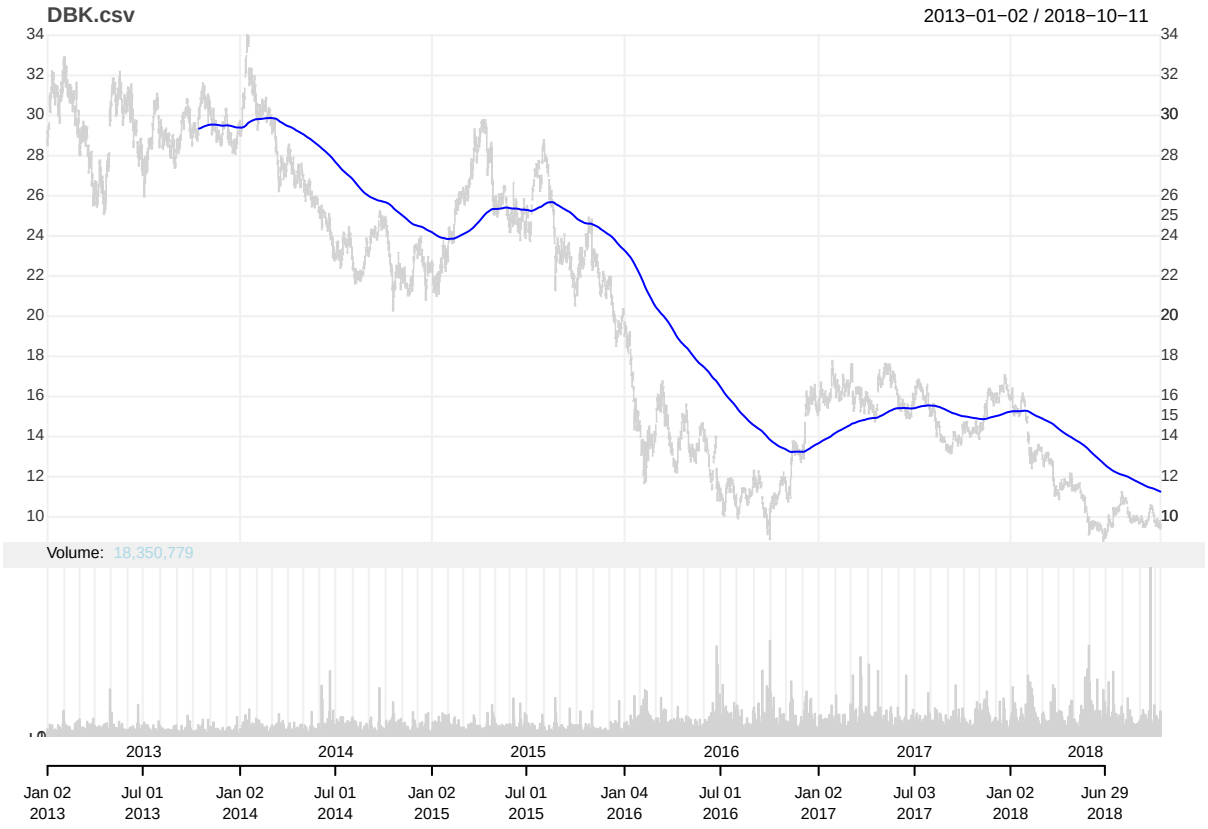
```
                portfolioname,
                Symbol = instrument,
                TxnDate = time(symbol[i + 1]),
                TxnPrice = as.numeric(symbol[i, "Close"]),
                TxnQty = -Position,
                TxnFees = transactionCost
              )
            }
          }
        }
      }
    }

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

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

### Part C: Analysis and Reporting

#### Visualize original data

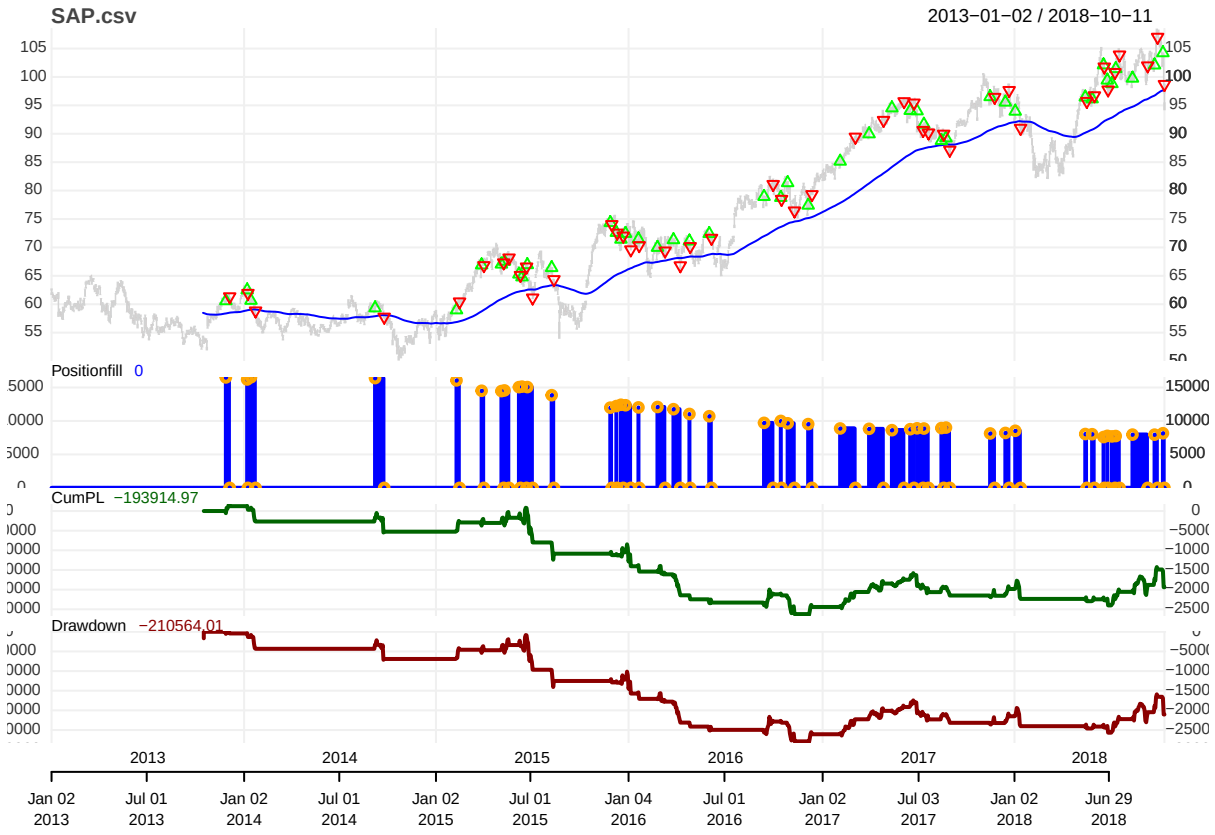
Plot of the instrument with the EMA line which indicates the general trend of the stock exponentially smoothed for the last 200 days. Moreover, the tradevolume is added below the graph.

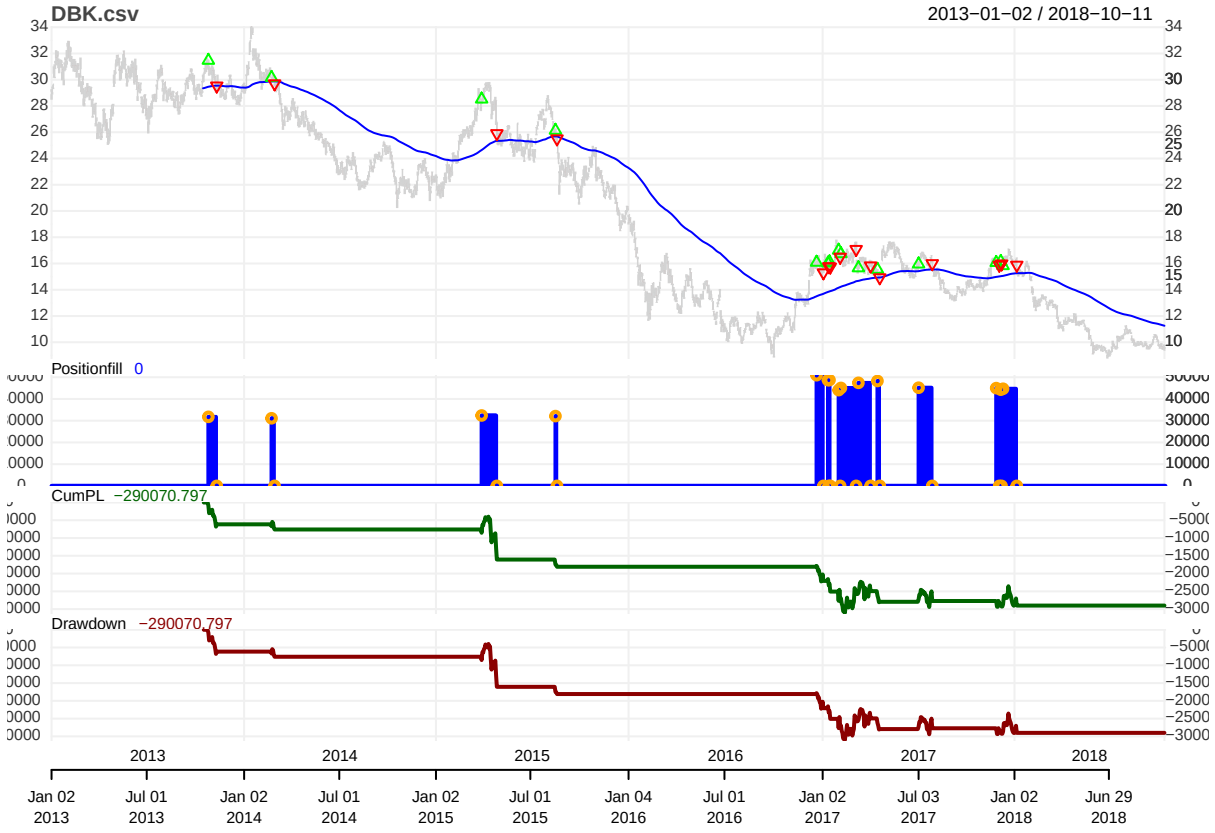





### Graph which visualize the transactions

The following graph shows 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 curve indicates the drawdown on each day compared to the last reached high.


SAP.csv
2013−01−02 / 2018−10−11
Positionfill 0
CumPL −193914.97
Drawdown −210564.01
2013
2014
2015
2016
2017
2018
Jan 02 2013
Jul 01 2013
Jan 02 2014
Jul 01 2014
Jan 02 2015
Jul 01 2015
Jan 04 2016
Jul 01 2016
Jan 02 2017
Jul 03 2017
Jan 02 2018
Jun 29 2018




### 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 per transaction.

```
## [1] "SAP.csv"
##            Txn.Qty Txn.Price Txn.Fees  Txn.Value Txn.Avg.Cost
## 1999-12-31       0      0.00        0        0.0         0.00
## 2013-11-22   16504     60.59      -20   999977.4        60.59
## 2013-11-29  -16504     61.34      -20 -1012355.4        61.34
## 2014-01-08   16197     62.50      -20  1012312.5        62.50
## 2014-01-09  -16197     61.94      -20 -1003242.2        61.94
## 2014-01-15   16538     60.66      -20  1003195.1        60.66
## 2014-01-23  -16538     58.85      -20  -973261.3        58.85
## 2014-09-04   16404     59.33      -20   973249.3        59.33
## 2014-09-22  -16404     57.76      -20  -947495.0        57.76
## 2015-02-10   16064     58.98      -20   947454.7        58.98
## 2015-02-16  -16064     60.44      -20  -970908.2        60.44
## 2015-03-27   14512     66.90      -20   970852.8        66.90
## 2015-04-01  -14512     66.82      -20  -969691.8        66.82
## 2015-05-07   14464     67.04      -20   969666.6        67.04
## 2015-05-12  -14464     67.31      -20  -973571.8        67.31
## 2015-05-13   14562     67.53      -20   983371.9        67.53
## 2015-05-21  -14562     68.17      -20  -992691.5        68.17
## 2015-06-10   15055     65.28      -20   982790.4        65.28
## 2015-06-12  -15055     65.11      -20  -980231.1        65.11
## 2015-06-16   15131     64.78      -20   980186.2        64.78
```

```
## 2015-06-24  -15131     66.59      -20 -1007573.3        66.59
## 2015-06-25   15063     66.94      -20  1008317.2        66.94
## 2015-07-06  -15063     61.13      -20  -920801.2        61.13
## 2015-08-10   13851     66.42      -20   919983.4        66.42
## 2015-08-13  -13851     64.36      -20  -891450.4        64.36
## 2015-11-25   11989     74.35      -20   891382.1        74.35
## 2015-11-27  -11989     74.05      -20  -887785.4        74.05
## 2015-12-07   12226     72.61      -20   887729.9        72.61
## 2015-12-08  -12226     72.55      -20  -886996.3        72.55
## 2015-12-15   12419     71.42      -20   886965.0        71.42
## 2015-12-18  -12419     72.03      -20  -894540.6        72.03
## 2015-12-23   12348     72.44      -20   894489.1        72.44
## 2016-01-07  -12348     69.60      -20  -859420.8        69.60
## 2016-01-21   12027     71.46      -20   859449.4        71.46
## 2016-01-22  -12027     70.34      -20  -845979.2        70.34
## 2016-02-25   12088     69.98      -20   845918.2        69.98
## 2016-03-10  -12088     69.40      -20  -838907.2        69.40
## 2016-03-29   11760     71.33      -20   838840.8        71.33
## 2016-04-11  -11760     66.80      -20  -785568.0        66.80
## 2016-04-27   11048     71.10      -20   785512.8        71.10
## 2016-04-28  -11048     70.18      -20  -775348.6        70.18
## 2016-06-03   10708     72.41      -20   775366.3        72.41
## 2016-06-08  -10708     71.64      -20  -767121.1        71.64
## 2016-09-13    9717     78.94      -20   767060.0        78.94
## 2016-09-29   -9717     81.11      -20  -788145.9        81.11
## 2016-10-14   10005     78.77      -20   788093.8        78.77
## 2016-10-17  -10005     78.49      -20  -785292.4        78.49
## 2016-10-27    9650     81.37      -20   785220.5        81.37
## 2016-11-09   -9650     76.45      -20  -737742.5        76.45
## 2016-12-05    9530     77.41      -20   737717.3        77.41
## 2016-12-12   -9530     79.32      -20  -755919.6        79.32
## 2017-02-02    8877     85.15      -20   755876.6        85.15
## 2017-03-02   -8877     89.44      -20  -793958.9        89.44
## 2017-03-28    8825     89.96      -20   793897.0        89.96
## 2017-04-26   -8825     92.39      -20  -815341.8        92.39
## 2017-05-12    8624     94.54      -20   815313.0        94.54
## 2017-06-06   -8624     95.68      -20  -825144.3        95.68
## 2017-06-16    8771     94.08      -20   825175.7        94.08
## 2017-06-23   -8771     95.44      -20  -837104.2        95.44
## 2017-06-29    8904     94.01      -20   837065.0        94.01
## 2017-07-11   -8904     90.62      -20  -806880.5        90.62
## 2017-07-12    8869     91.66      -20   812932.5        91.66
## 2017-07-20   -8869     90.18      -20  -799806.4        90.18
## 2017-08-14    8938     88.79      -20   793605.0        88.79
## 2017-08-17   -8938     89.96      -20  -804062.5        89.96
## 2017-08-22    9009     89.25      -20   804053.2        89.25
## 2017-08-29   -9009     87.14      -20  -785044.3        87.14
## 2017-11-14    8134     96.51      -20   785012.3        96.51
## 2017-11-22   -8134     96.44      -20  -784443.0        96.44
## 2017-12-12    8209     95.55      -20   784369.9        95.55
## 2017-12-19   -8209     97.65      -20  -801608.9        97.65
## 2018-01-03    8532     93.95      -20   801581.4        93.95
## 2018-01-12   -8532     91.00      -20  -776412.0        91.00
## 2018-05-16    8050     96.44      -20   776342.0        96.44
```

```
## 2018-05-18   -8050     95.72      -20  -770546.0        95.72
## 2018-05-30    8016     96.12      -20   770497.9        96.12
## 2018-06-04   -8016     96.72      -20  -775307.5        96.72
## 2018-06-20    7594    102.10      -20   775347.4       102.10
## 2018-06-21   -7594    101.78      -20  -772917.3       101.78
## 2018-06-27    7773     99.42      -20   772791.7        99.42
## 2018-06-28   -7773     97.75      -20  -759810.8        97.75
## 2018-07-05    7690     98.80      -20   759772.0        98.80
## 2018-07-11   -7690    100.82      -20  -775305.8       100.82
## 2018-07-12    7746    101.44      -20   785754.2       101.44
## 2018-07-19   -7746    103.90      -20  -804809.4       103.90
## 2018-08-13    7962     99.76      -20   794289.1        99.76
## 2018-09-10   -7962    101.98      -20  -811964.8       101.98
## 2018-09-21    7954    102.08      -20   811944.3       102.08
## 2018-09-27   -7954    107.02      -20  -851237.1       107.02
## 2018-10-09    8166    104.24      -20   851223.8       104.24
## 2018-10-11   -8166     98.71      -20  -806065.9        98.71
##            Net.Txn.Realized.PL
## 1999-12-31                0.00
## 2013-11-22              -20.00
## 2013-11-29            12358.00
## 2014-01-08              -20.00
## 2014-01-09            -9090.32
## 2014-01-15              -20.00
## 2014-01-23           -29953.78
## 2014-09-04              -20.00
## 2014-09-22           -25774.28
## 2015-02-10              -20.00
## 2015-02-16            23433.44
## 2015-03-27              -20.00
## 2015-04-01            -1180.96
## 2015-05-07              -20.00
## 2015-05-12             3885.28
## 2015-05-13              -20.00
## 2015-05-21             9299.68
## 2015-06-10              -20.00
## 2015-06-12            -2579.35
## 2015-06-16              -20.00
## 2015-06-24            27367.11
## 2015-06-25              -20.00
## 2015-07-06           -87536.03
## 2015-08-10              -20.00
## 2015-08-13           -28553.06
## 2015-11-25              -20.00
## 2015-11-27            -3616.70
## 2015-12-07              -20.00
## 2015-12-08             -753.56
## 2015-12-15              -20.00
## 2015-12-18             7555.59
## 2015-12-23              -20.00
## 2016-01-07           -35088.32
## 2016-01-21              -20.00
## 2016-01-22           -13490.24
## 2016-02-25              -20.00
```

```
## 2016-03-10           -7031.04
## 2016-03-29             -20.00
## 2016-04-11          -53292.80
## 2016-04-27             -20.00
## 2016-04-28          -10184.16
## 2016-06-03             -20.00
## 2016-06-08           -8265.16
## 2016-09-13             -20.00
## 2016-09-29           21065.89
## 2016-10-14             -20.00
## 2016-10-17           -2821.40
## 2016-10-27             -20.00
## 2016-11-09          -47498.00
## 2016-12-05             -20.00
## 2016-12-12           18182.30
## 2017-02-02             -20.00
## 2017-03-02           38062.33
## 2017-03-28             -20.00
## 2017-04-26           21424.75
## 2017-05-12             -20.00
## 2017-06-06            9811.36
## 2017-06-16             -20.00
## 2017-06-23           11908.56
## 2017-06-29             -20.00
## 2017-07-11          -30204.56
## 2017-07-12             -20.00
## 2017-07-20          -13146.12
## 2017-08-14             -20.00
## 2017-08-17           10437.46
## 2017-08-22             -20.00
## 2017-08-29          -19028.99
## 2017-11-14             -20.00
## 2017-11-22            -589.38
## 2017-12-12             -20.00
## 2017-12-19           17218.90
## 2018-01-03             -20.00
## 2018-01-12          -25189.40
## 2018-05-16             -20.00
## 2018-05-18           -5816.00
## 2018-05-30             -20.00
## 2018-06-04            4789.60
## 2018-06-20             -20.00
## 2018-06-21           -2450.08
## 2018-06-27             -20.00
## 2018-06-28          -13000.91
## 2018-07-05             -20.00
## 2018-07-11           15513.80
## 2018-07-12             -20.00
## 2018-07-19           19035.16
## 2018-08-13             -20.00
## 2018-09-10           17655.64
## 2018-09-21             -20.00
## 2018-09-27           39272.76
## 2018-10-09             -20.00
```

```
## 2018-10-11           -45177.98
## [1] "DBK.csv"
##            Txn.Qty Txn.Price Txn.Fees Txn.Value Txn.Avg.Cost
## 1999-12-31       0     0.000        0       0.0        0.000
## 2013-10-22   31779    31.467      -20  999989.8       31.467
## 2013-11-06  -31779    29.534      -20 -938561.0       29.534
## 2014-02-21   31114    30.164      -20  938522.7       30.164
## 2014-02-27  -31114    29.700      -20 -924085.8       29.700
## 2015-03-27   32404    28.516      -20  924032.5       28.516
## 2015-04-28  -32404    25.905      -20 -839425.6       25.905
## 2015-08-17   32115    26.137      -20  839389.8       26.137
## 2015-08-19  -32115    25.508      -20 -819189.4       25.508
## 2016-12-20   50933    16.083      -20  819155.4       16.083
## 2017-01-03  -50933    15.302      -20 -779376.8       15.302
## 2017-01-11   48645    16.021      -20  779341.5       16.021
## 2017-01-12  -48645    15.771      -20 -767180.3       15.771
## 2017-01-13   48698    16.110      -20  784524.8       16.110
## 2017-01-16  -48698    15.744      -20 -766701.3       15.744
## 2017-01-31   44070    17.002      -20  749278.1       17.002
## 2017-02-02  -44070    16.494      -20 -726890.6       16.494
## 2017-02-03   45077    16.735      -20  754363.6       16.735
## 2017-03-03  -45077    17.083      -20 -770050.4       17.083
## 2017-03-08   47417    15.659      -20  742502.8       15.659
## 2017-03-30  -47417    15.830      -20 -750611.1       15.830
## 2017-04-12   48283    15.545      -20  750559.2       15.545
## 2017-04-19  -48283    14.925      -20 -720623.8       14.925
## 2017-07-03   45178    15.950      -20  720589.1       15.950
## 2017-07-27  -45178    16.000      -20 -722848.0       16.000
## 2017-11-24   44993    16.065      -20  722812.5       16.065
## 2017-11-30  -44993    15.885      -20 -714713.8       15.885
## 2017-12-04   44266    16.145      -20  714674.6       16.145
## 2017-12-05  -44266    15.980      -20 -707370.7       15.980
## 2017-12-07   44697    15.825      -20  707330.0       15.825
## 2018-01-05  -44697    15.884      -20 -709967.1       15.884
##            Net.Txn.Realized.PL
## 1999-12-31               0.000
## 2013-10-22             -20.000
## 2013-11-06          -61448.807
## 2014-02-21             -20.000
## 2014-02-27          -14456.896
## 2015-03-27             -20.000
## 2015-04-28          -84626.844
## 2015-08-17             -20.000
## 2015-08-19          -20220.335
## 2016-12-20             -20.000
## 2017-01-03          -39798.673
## 2017-01-11             -20.000
## 2017-01-12          -12181.250
## 2017-01-13             -20.000
## 2017-01-16          -17843.468
## 2017-01-31             -20.000
## 2017-02-02          -22407.560
## 2017-02-03             -20.000
## 2017-03-03           15666.796
```

```
## 2017-03-08             -20.000
## 2017-03-30            8088.307
## 2017-04-12             -20.000
## 2017-04-19          -29955.460
## 2017-07-03             -20.000
## 2017-07-27            2238.900
## 2017-11-24             -20.000
## 2017-11-30           -8118.740
## 2017-12-04             -20.000
## 2017-12-05           -7323.890
## 2017-12-07             -20.000
## 2018-01-05            2617.123
```

### Performance Statistics

The following table summarizes some important trading statistics for all instruments.

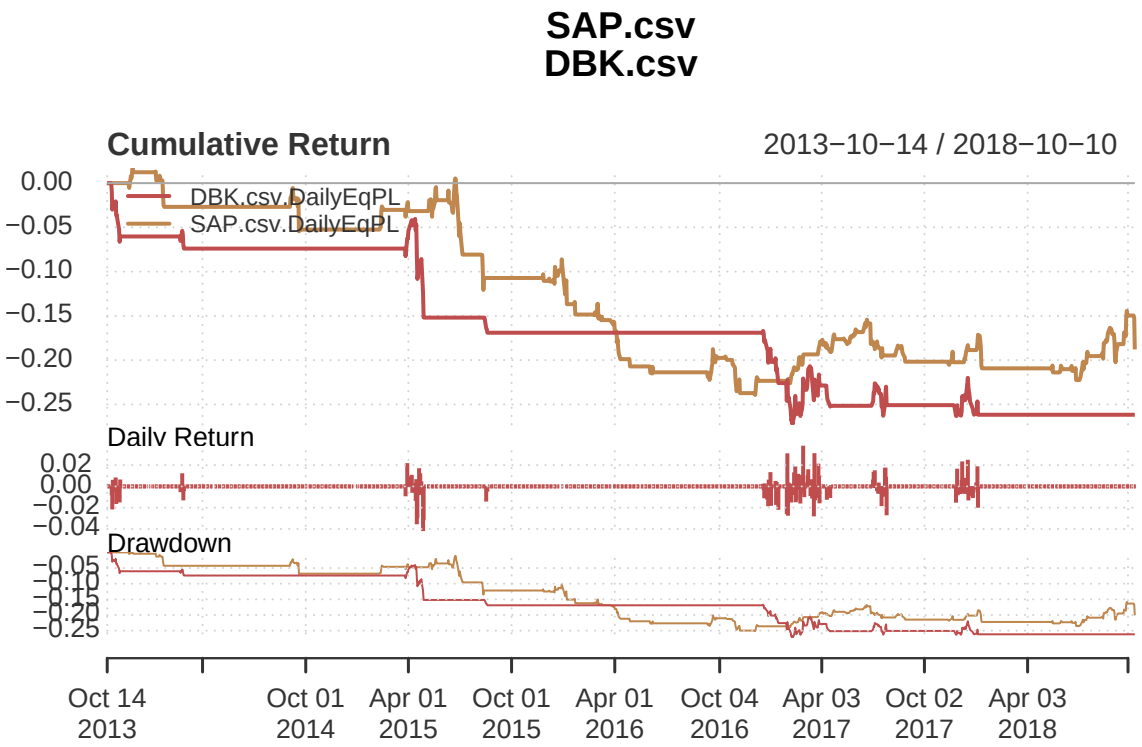
```
tstats <- tradeStats(Portfolio=portfolioname, Symbols=instrumentlist)
for (i in 1:nrow(tstats)) {
  trades.tab <- cbind(
    c("Trades", "Win Percent", "Loss Percent","W/L Ratio"),
    c(tstats[i,"Num.Trades"],tstats[i,"Percent.Positive"],tstats[i,"Percent.Negative"],tstats[i,"Percen
  print(row.names(tstats[i,]))
  print(trades.tab)
  }
```

```
## [1] "SAP.csv"
##      [,1]           [,2]
## [1,] "Trades"       "45"
## [2,] "Win Percent"  "42.2222222222222"
## [3,] "Loss Percent" "57.7777777777778"
## [4,] "W/L Ratio"    "0.730769230769231"
## [1] "DBK.csv"
##      [,1]           [,2]
## [1,] "Trades"       "15"
## [2,] "Win Percent"  "26.6666666666667"
## [3,] "Loss Percent" "73.3333333333333"
## [4,] "W/L Ratio"    "0.363636363636364"
```

### Visualize returns of the trading strategy

```
library(PerformanceAnalytics) # contains lots of methods to investigate performance
# obtain the portfolio returns - with these you can compute virtually any financial metrics you wish
rets <- PortfReturns(Account=accountname)
rownames(rets) <- NULL # this step is important!
```

```
charts.PerformanceSummary(rets,colorset=rainbow12equal,main=instrumentlist)
```




**Calculate statistics of the Portfolio**

```
tab.perf <- table.Arbitrary(rets, metrics=c("Return.cumulative", "Return.annualized","SharpeRatio.annual
                            metricsNames=c("Cumulative Return", "Annualized Return", "Annualized Sharp
tab.risk <- table.Arbitrary(rets, metrics=c("StdDev.annualized", "maxDrawdown","VaR","ES"),
                            metricsNames=c("Annualized StdDev", "Max Drawdown", "Value-at-Risk","Conditi
# present the portfolio statistics
for (i in 1:ncol(tab.perf)) {
somestats <- data.frame(rownames(tab.perf),tab.perf[i,1], rownames(tab.risk),tab.risk[i,1])
colnames(rets[,i])
colnames(somestats) <- c("Performance Metric", "Performance Value", "Risk Metric", "Risk Value")
print(somestats)
}
```

```
##       Performance Metric Performance Value       Risk Metric Risk Value
## 1      Cumulative Return        -0.2616721 Annualized StdDev 0.07266228
## 2      Annualized Return        -0.2616721      Max Drawdown 0.07266228
## 3 Annualized Sharp Ratio        -0.2616721     Value-at-Risk 0.07266228
## 4           Calmar Ratio        -0.2616721   Conditional VaR 0.07266228
##       Performance Metric Performance Value       Risk Metric Risk Value
## 1      Cumulative Return         -0.058779 Annualized StdDev   0.272559
## 2      Annualized Return         -0.058779      Max Drawdown   0.272559
## 3 Annualized Sharp Ratio         -0.058779     Value-at-Risk   0.272559
## 4           Calmar Ratio         -0.058779   Conditional VaR   0.272559
```

## 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.

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

# The Buy and hold symbol 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,Txn

# 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,TxnFee

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

### 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)
```


### 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 calculatet before.

```
rets.bh <- PortfReturns(Account='buyhold')
returns <- cbind(rets,rets.bh)
#rulecol <- paste(portfolioname,instrument,sep="-")
#colnames(returns) <- c(rulecol,"Buy-and-hold")
```

We compare the two strategies by showing some statistical metrics of the returns and plot the returns in one chart to directly compare the performance of the strategies.

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

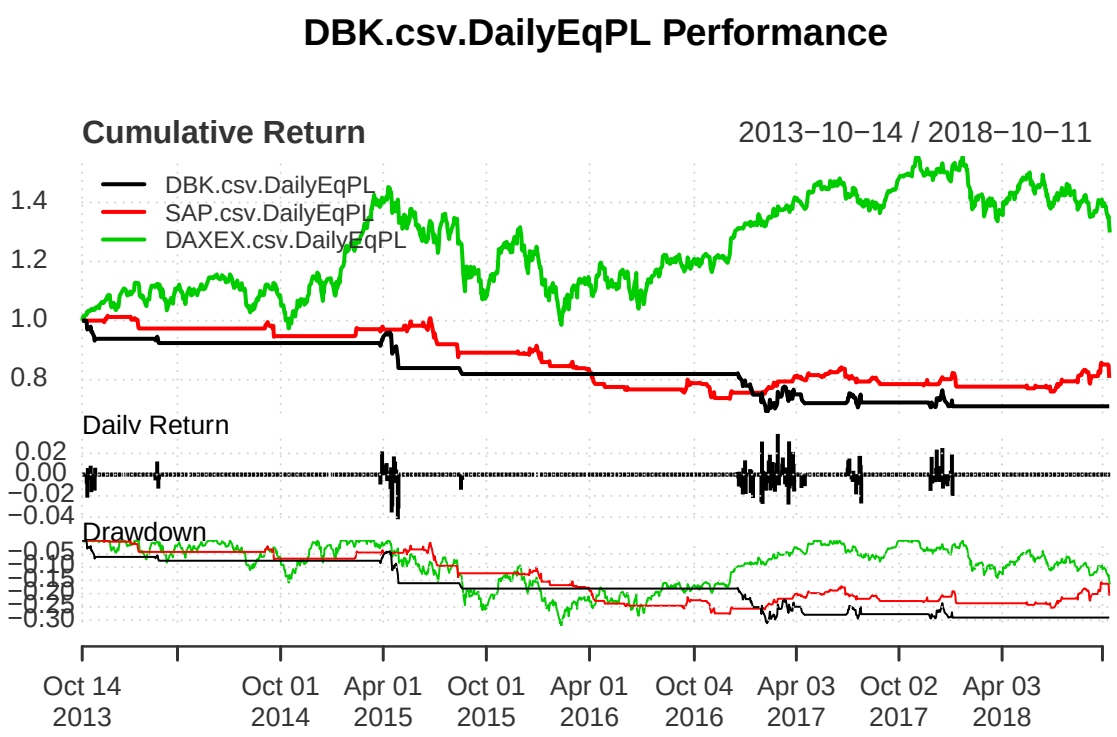
```
##                 DBK.csv.DailyEqPL SAP.csv.DailyEqPL DAXEX.csv.DailyEqPL
## Observations           1262.0000         1262.0000           1463.0000
## NAs                     201.0000          201.0000              0.0000
## Minimum                  -0.0419           -0.0416             -0.0857
## Quartile 1                0.0000            0.0000             -0.0070
## Median                    0.0000            0.0000              0.0010
## Arithmetic Mean          -0.0002           -0.0002              0.0003
## Geometric Mean           -0.0002           -0.0002              0.0002
## Quartile 3                0.0000            0.0000              0.0078
## Maximum                   0.0382            0.0348              0.0607
```

```
## SE Mean                    0.0001            0.0001            0.0004
## LCL Mean (0.95)           -0.0005           -0.0004           -0.0005
## UCL Mean (0.95)            0.0000            0.0001            0.0010
## Variance                   0.0000            0.0000            0.0002
## Stdev                      0.0046            0.0047            0.0143
## Skewness                  -1.0640           -1.5126           -0.3312
## Kurtosis                  27.0980           21.1288            2.0578
```

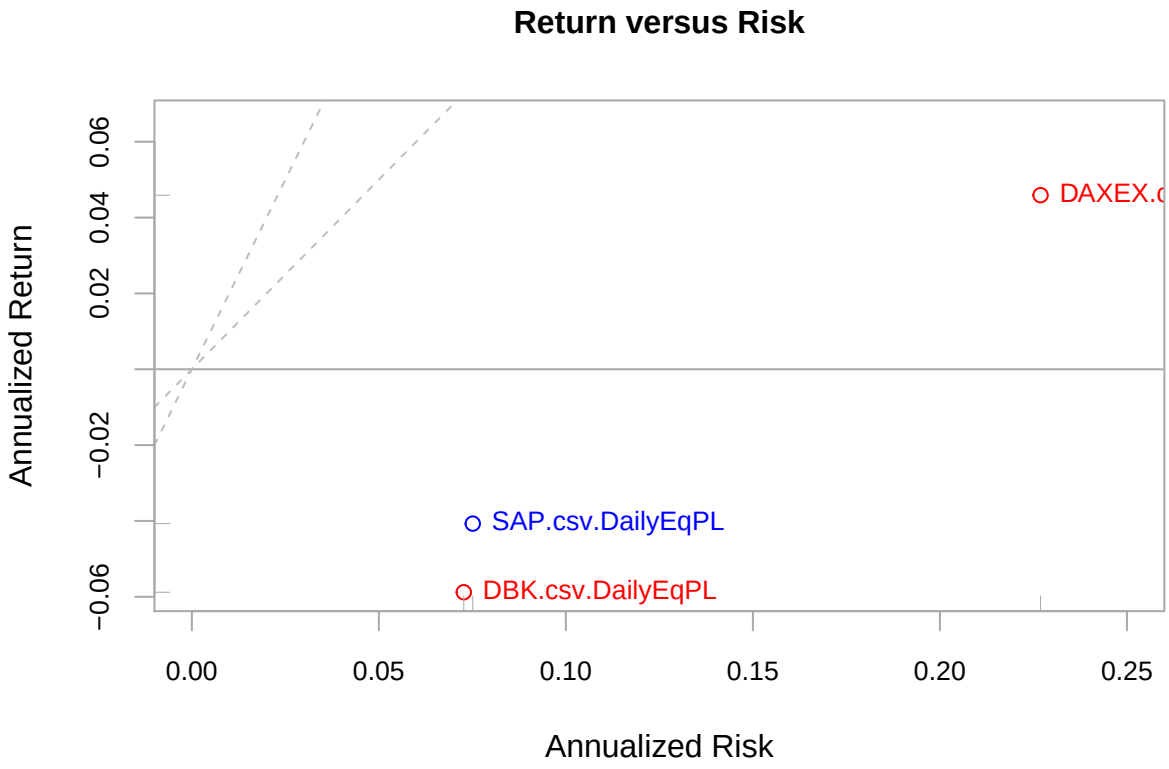
```
table.AnnualizedReturns(returns)
```

```
##                           DBK.csv.DailyEqPL SAP.csv.DailyEqPL
## Annualized Return                   -0.0588           -0.0407
## Annualized Std Dev                   0.0727            0.0751
## Annualized Sharpe (Rf=0%)           -0.8089           -0.5422
##                           DAXEX.csv.DailyEqPL
## Annualized Return                      0.0459
## Annualized Std Dev                     0.2269
## Annualized Sharpe (Rf=0%)              0.2024
```

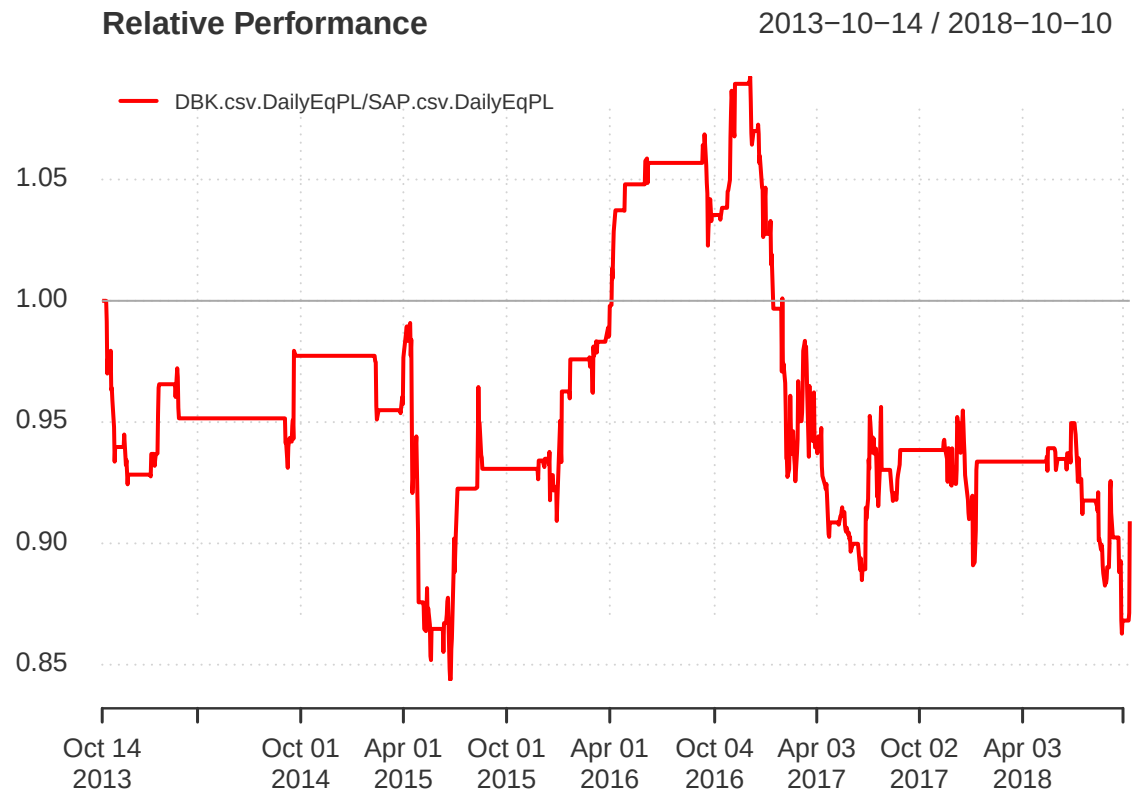
```
charts.PerformanceSummary(returns,geometric=FALSE,wealth.index=TRUE)
```




```
chart.RiskReturnScatter(returns,Rf=0,add.sharpe=c(1,2),xlim=c(0,0.25),main="Return versus Risk",colorse
```




```
chart.RelativePerformance(returns[,1],returns[,2],colorset=c("red","blue"),lwd=2,legend.loc="topleft")
```




## Backtest

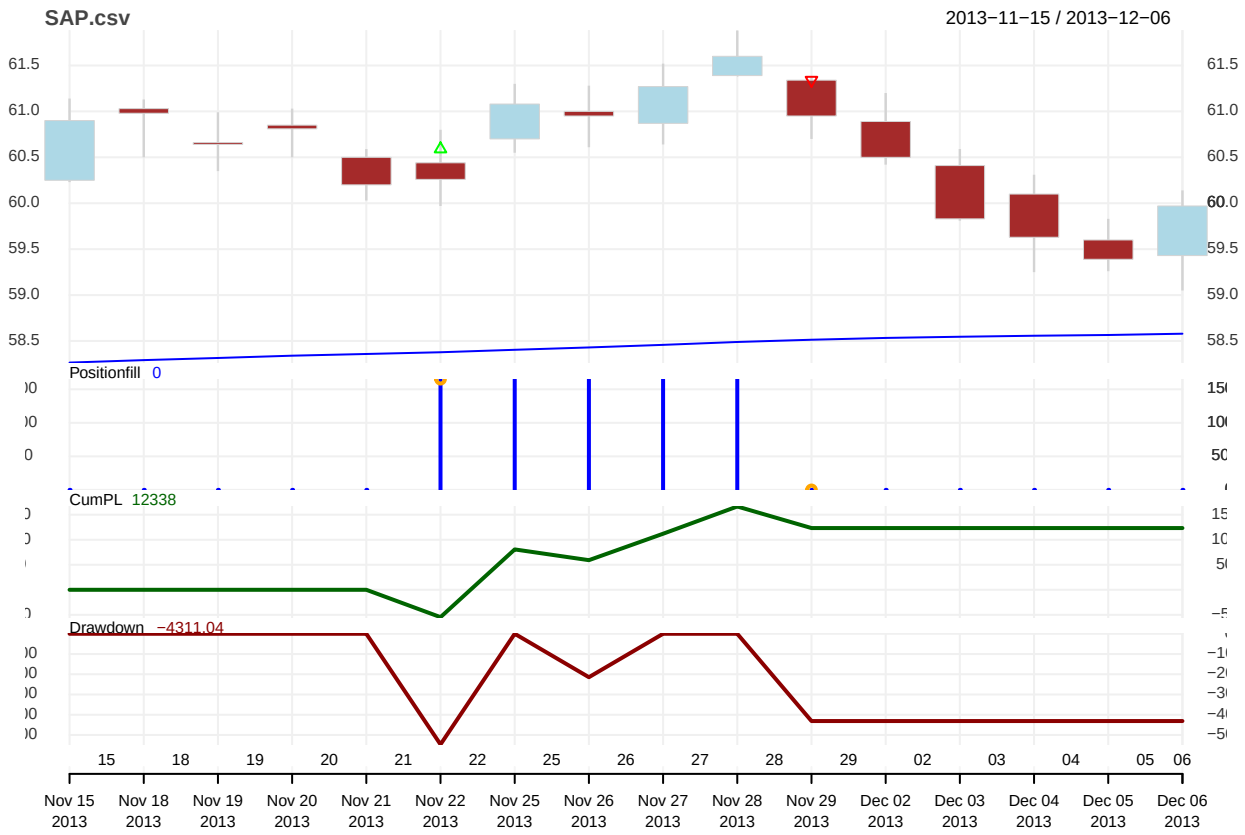
### Approach

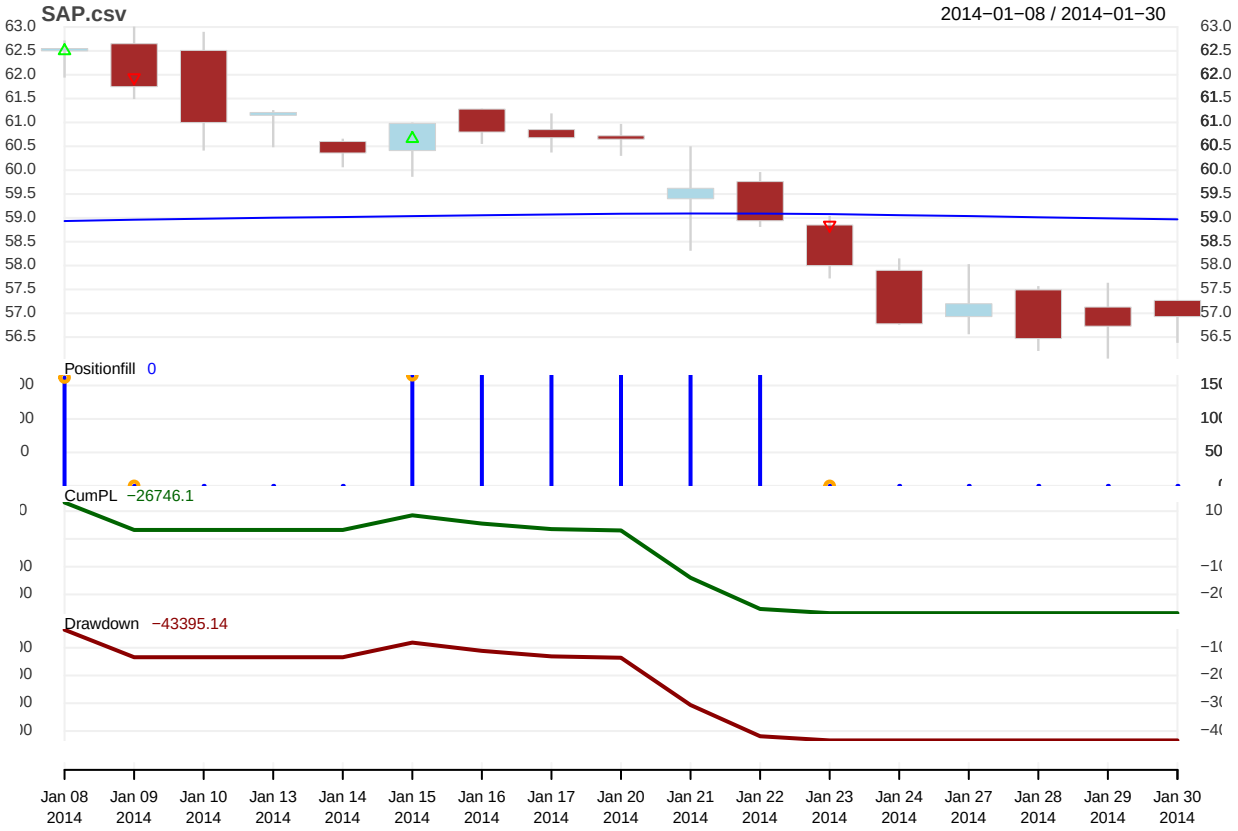
### Implementation

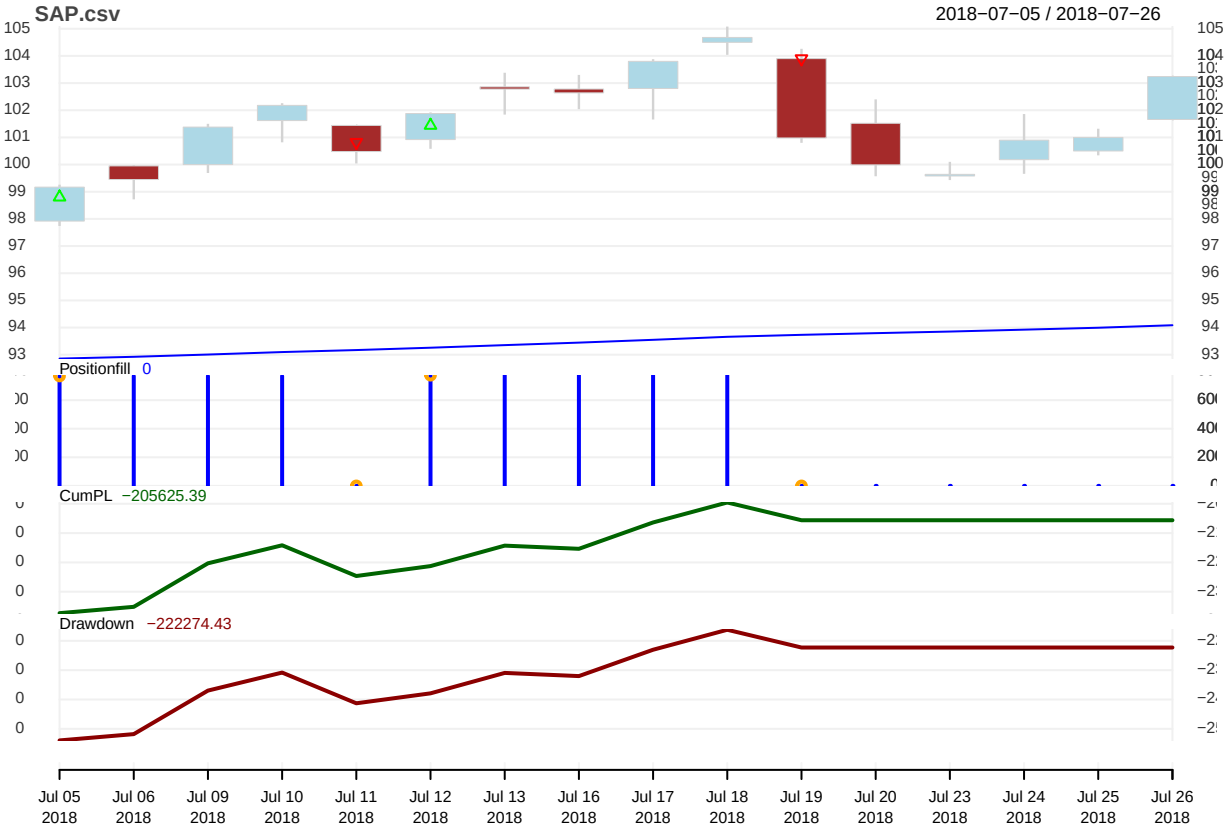
### System Check

To make sure that our system worked properly some transactions were observed to make sure that the trades were performed according to the defined rules. Prepare the chart. Check 4 transactions. Visualize the charts with the transactions which should be checked.

```
for (instrument in instrumentlist){
  rm(daterange_check)
  daterange_check <- c()
  transactionsInstrument <- getTxns(Portfolio=portfolioname,Symbol=instrument)
  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 = ""))
  }
  for (daterange_check_i in daterange_check){
    print(chart.Posn(portfolioname,Symbol=instrument,type='candlesticks', theme=myTheme,subset=daterang
  }
}
```

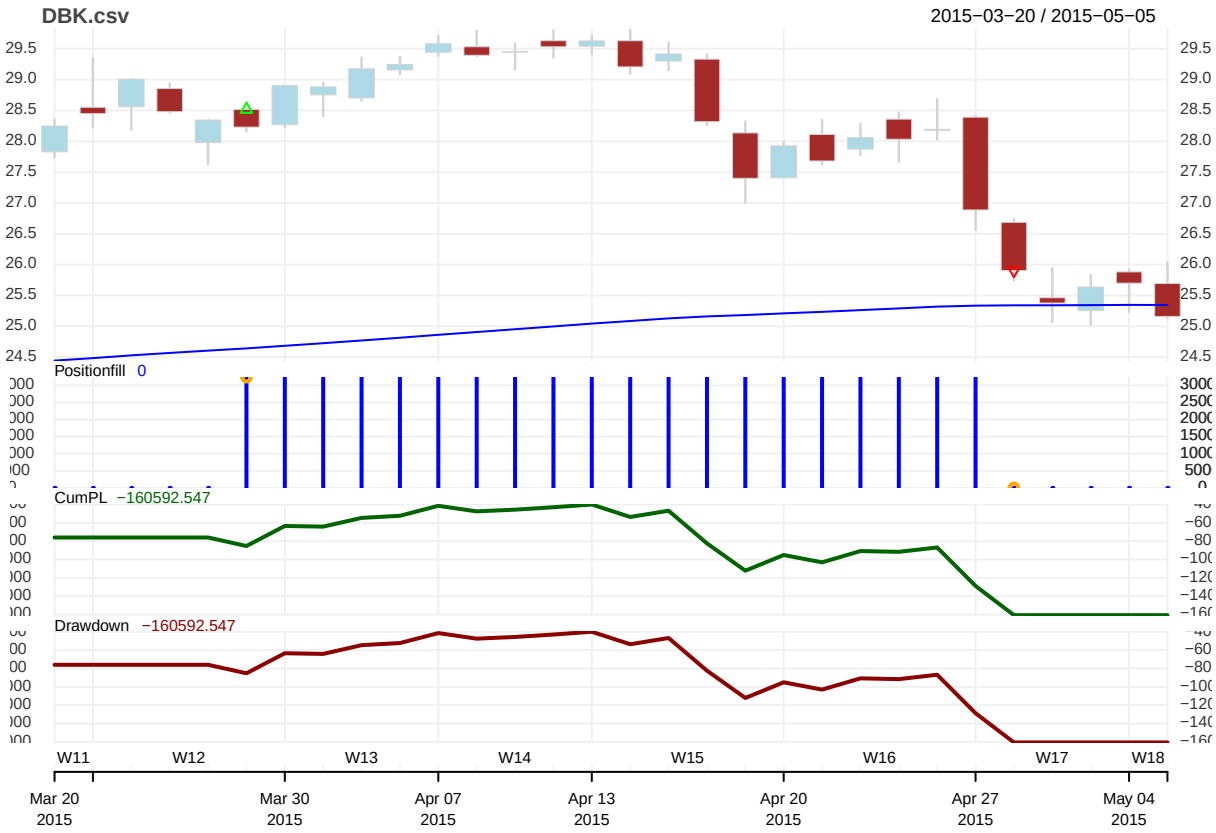

SAP.csv
2013−11−15 / 2013−12−06
61.5
61.0
60.5
60.0
59.5
59.0
58.5
Positionfill 0
CumPL 12338
Drawdown −4311.04
15
18
19
20
21
22
25
26
27
28
29
02
03
04
05
06
Nov 15 2013
Nov 18 2013
Nov 19 2013
Nov 20 2013
Nov 21 2013
Nov 22 2013
Nov 25 2013
Nov 26 2013
Nov 27 2013
Nov 28 2013
Nov 29 2013
Dec 02 2013
Dec 03 2013
Dec 04 2013
Dec 05 2013
Dec 06 2013

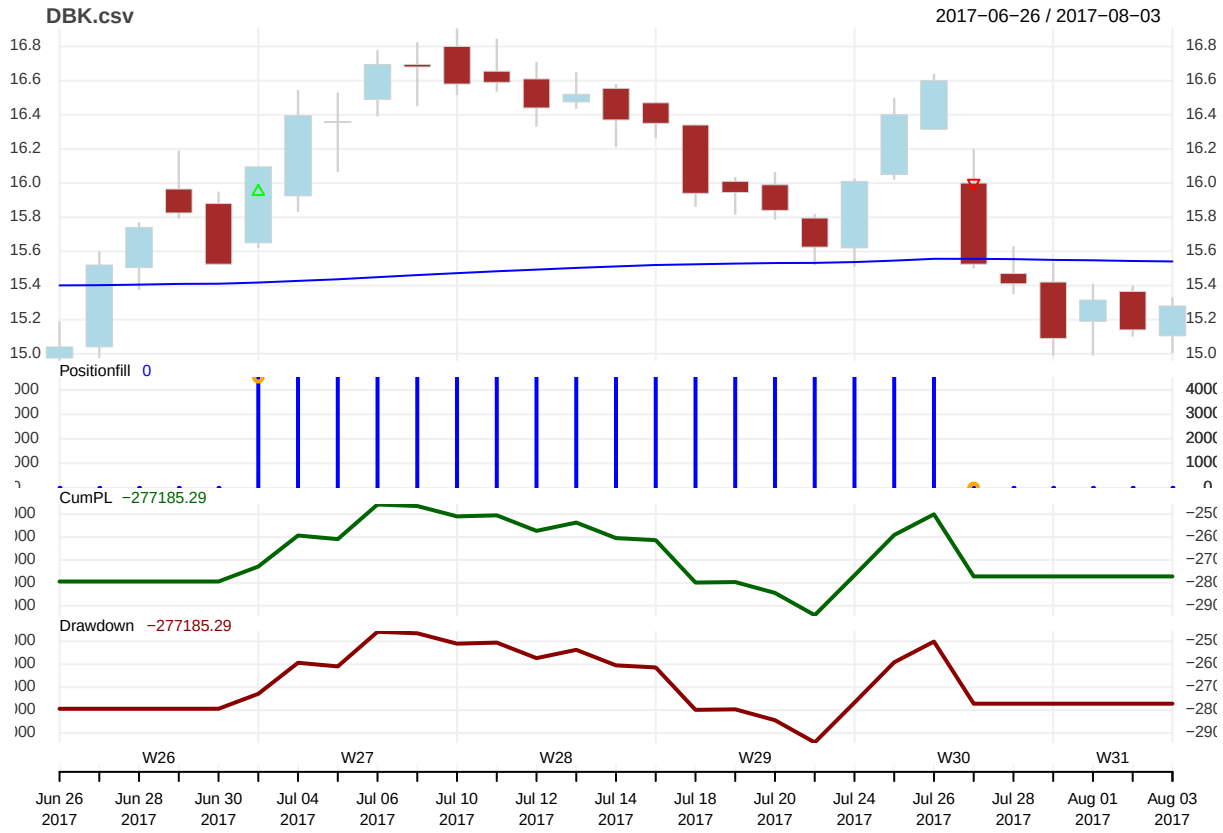

SAP.csv
2014−01−08 / 2014−01−30
Positionfill 0
CumPL −26746.1
Drawdown −43395.14
Jan 08 2014
Jan 09 2014
Jan 10 2014
Jan 13 2014
Jan 14 2014
Jan 15 2014
Jan 16 2014
Jan 17 2014
Jan 20 2014
Jan 21 2014
Jan 22 2014
Jan 23 2014
Jan 24 2014
Jan 27 2014
Jan 28 2014
Jan 29 2014
Jan 30 2014

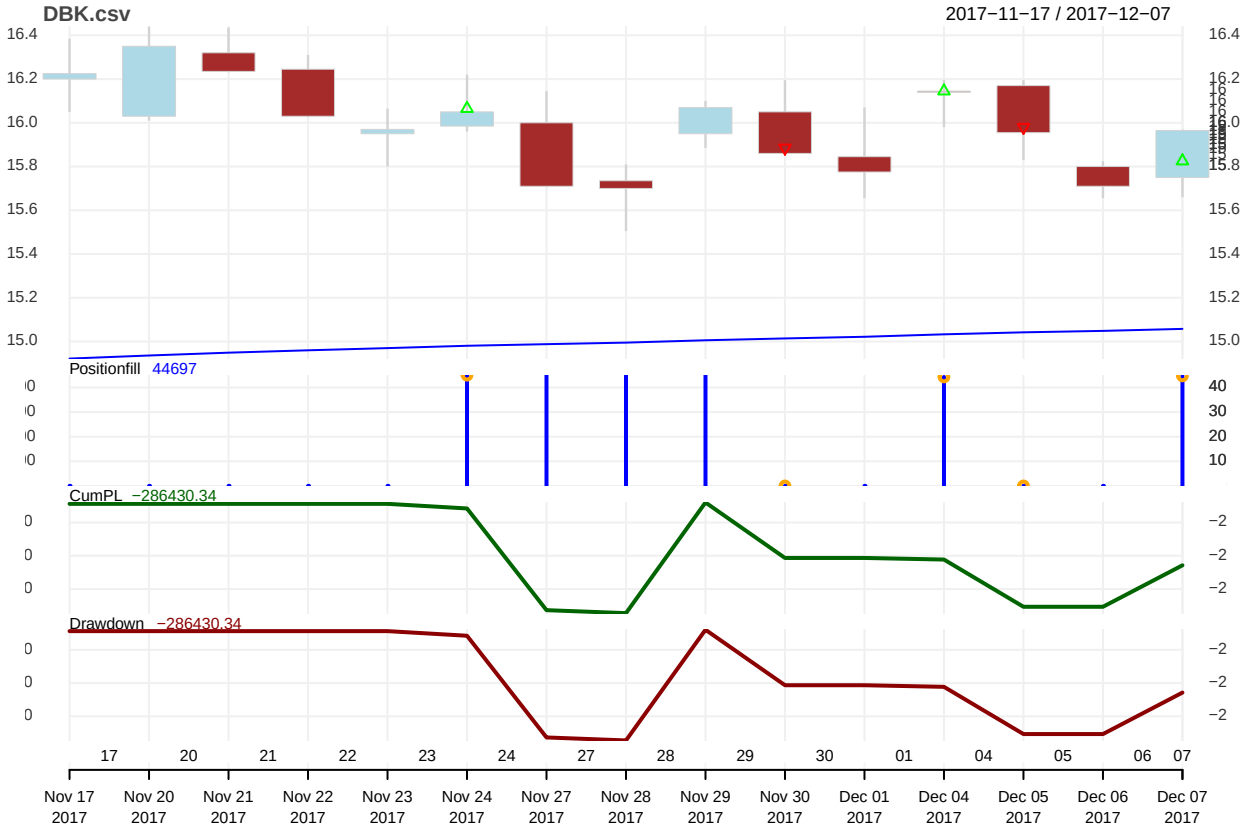

SAP.csv
2018−07−05 / 2018−07−26
Positionfill 0
CumPL −205625.39
Drawdown −222274.43
Jul 05 2018
Jul 06 2018
Jul 09 2018
Jul 10 2018
Jul 11 2018
Jul 12 2018
Jul 13 2018
Jul 16 2018
Jul 17 2018
Jul 18 2018
Jul 19 2018
Jul 20 2018
Jul 23 2018
Jul 24 2018
Jul 25 2018
Jul 26 2018


SAP.csv
2018−08−06 / 2018−09−17
Positionfill 0
CumPL −187989.75
Drawdown −204638.79
W32
W33
W34
W35
W36
W37
W38
Aug 06 2018
Aug 13 2018
Aug 20 2018
Aug 27 2018
Sep 03 2018
Sep 10 2018
Sep 17 2018


DBK.csv
2013−10−15 / 2013−11−13
31.5
31.0
30.5
30.0
29.5
29.0
Positionfill 0
CumPL −61468.807
Drawdown −61468.807
15 16 17 18 21 22 23 24 25 28 29 30 31 01 04 05 06 07 08 11 12 13
Oct 15 2013
Oct 17 2013
Oct 21 2013
Oct 23 2013
Oct 25 2013
Oct 29 2013
Oct 31 2013
Nov 04 2013
Nov 06 2013
Nov 08 2013
Nov 12 2013


DBK.csv
2015−03−20 / 2015−05−05
Positionfill 0
CumPL −160592.547
Drawdown −160592.547
W11
W12
W13
W14
W15
W16
W17
W18
Mar 20 2015
Mar 30 2015
Apr 07 2015
Apr 13 2015
Apr 20 2015
Apr 27 2015
May 04 2015


DBK.csv
2017−06−26 / 2017−08−03
16.8
16.6
16.4
16.2
16.0
15.8
15.6
15.4
15.2
15.0
Positionfill 0
CumPL −277185.29
Drawdown −277185.29
W26
W27
W28
W29
W30
W31
Jun 26 2017
Jun 28 2017
Jun 30 2017
Jul 04 2017
Jul 06 2017
Jul 10 2017
Jul 12 2017
Jul 14 2017
Jul 18 2017
Jul 20 2017
Jul 24 2017
Jul 26 2017
Jul 28 2017
Aug 01 2017
Aug 03 2017




### Analysis

### Returns

### Historical VaR

### Equity curve

## Conclusion and Suggestions