

# Assignment 1

*Philipp Epstein*

*10/14/2018*

## Introduction

Blabla Blabla

## Trading Idea

Explain here the trading idea, where I found it. What the theoretical foundation is -> Long Term positive autocorrelation -> Short term negative autocorrelation

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 <- 7

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=1))

# 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) {
    if(TRUE){
      # 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 today's
# low price. (Buy tomorrow for 'price <= today's 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)

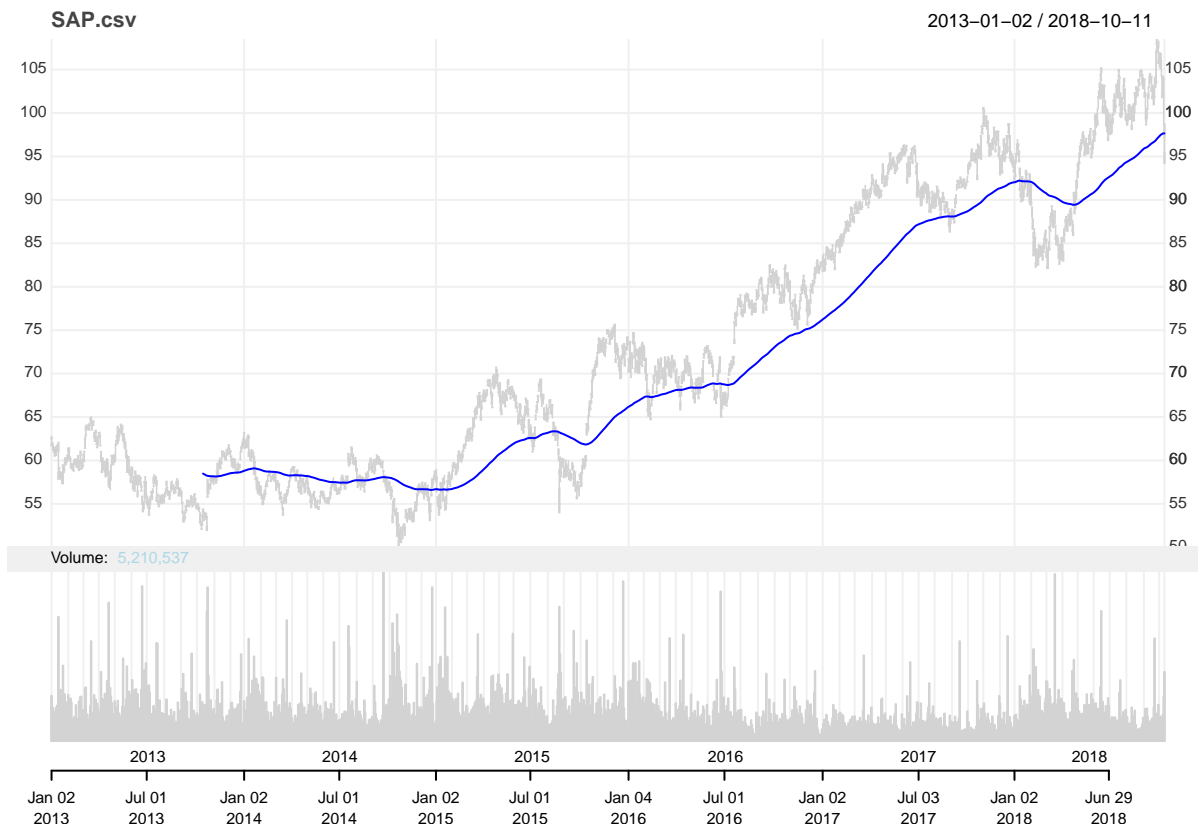
} # End Bar-by-bar processing
} # End for loop for multiple instruments

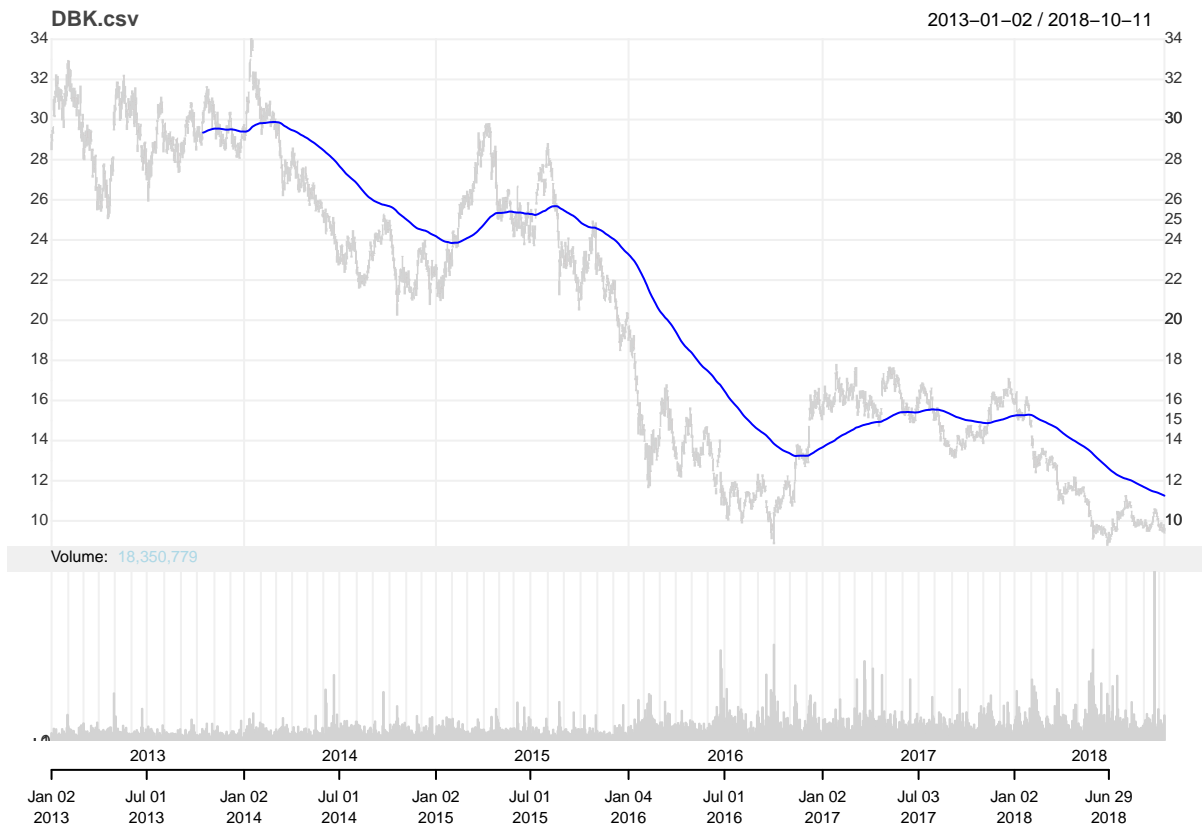
```

## Part C: Analysis and Reporting

### Visualize original data

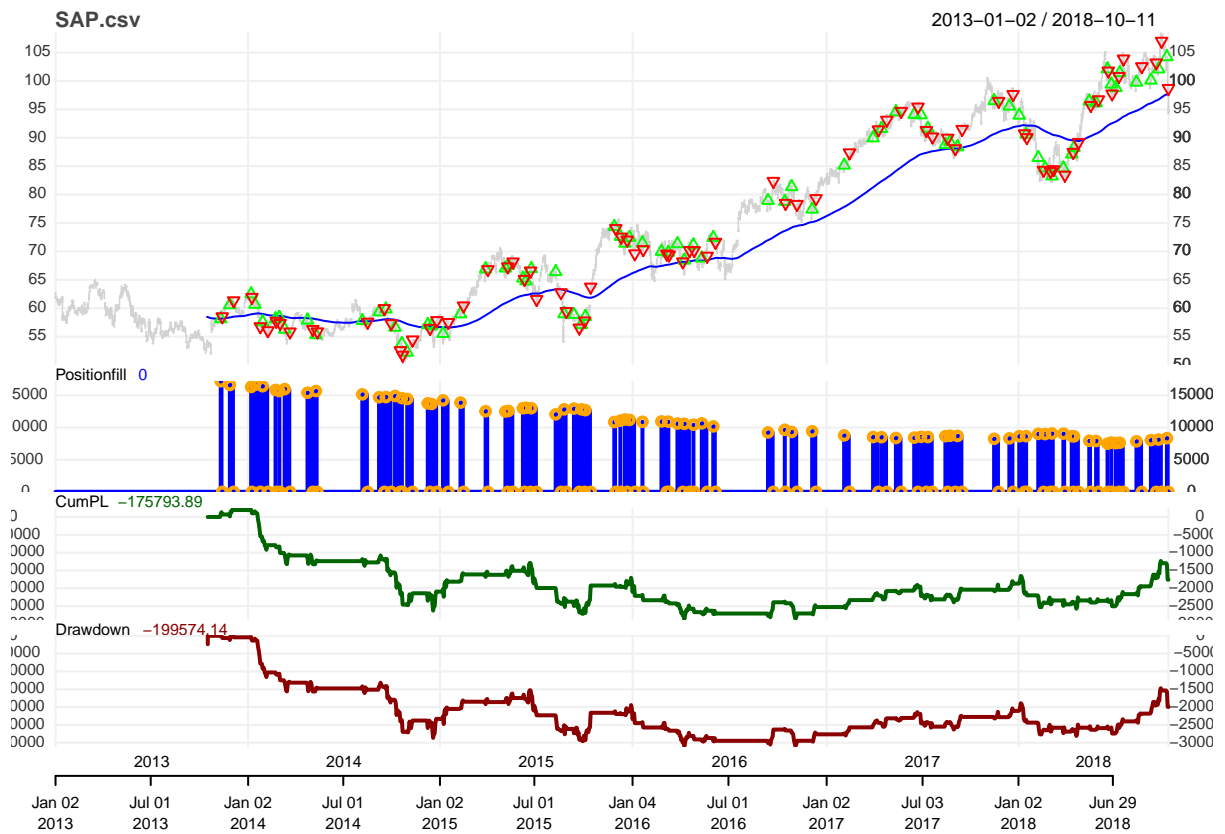
Plot of the instrument with the EMA line which indicates the general trend of the stock exponentially smoothed for the last 200 days. Moreover, the trade volume is added below the graph.



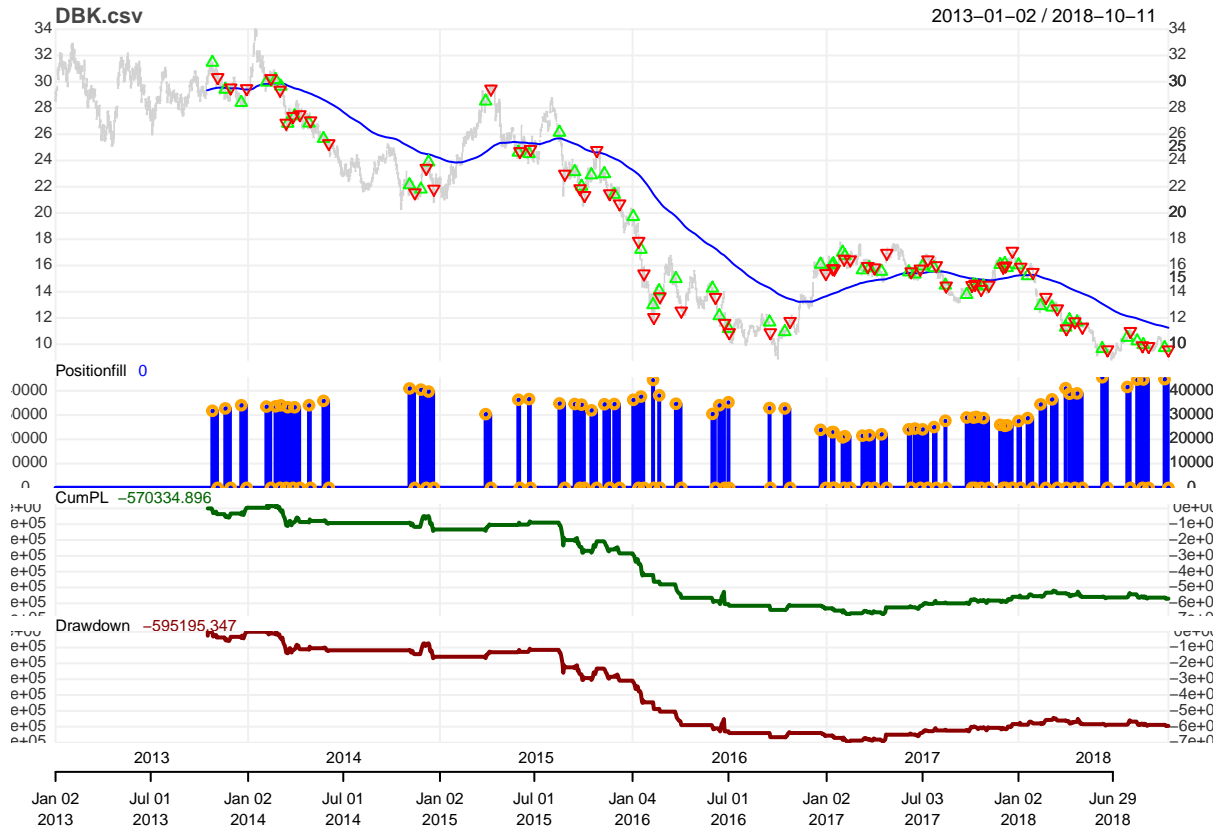


### Graph which visualize the transactions

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 cumulative net profit curve, while the red curve indicates the drawdown on each day compared to the last reached high.







### 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
##	2013-11-06	17205	58.12	-20	999954.6	58.12
##	2013-11-08	-17205	58.53	-20	-1007008.7	58.53
##	2013-11-22	16620	60.59	-20	1007005.8	60.59
##	2013-11-29	-16620	61.34	-20	-1019470.8	61.34
##	2014-01-08	16311	62.50	-20	1019437.5	62.50
##	2014-01-09	-16311	61.94	-20	-1010303.3	61.94
##	2014-01-15	16654	60.66	-20	1010231.6	60.66
##	2014-01-24	-16654	56.78	-20	-945614.1	56.78
##	2014-01-29	16425	57.57	-20	945587.2	57.57
##	2014-02-07	-16425	56.07	-20	-920949.8	56.07
##	2014-02-21	15850	58.10	-20	920885.0	58.10
##	2014-02-24	-15850	57.73	-20	-915020.5	57.73
##	2014-02-28	15676	58.37	-20	915008.1	58.37
##	2014-03-03	-15676	57.35	-20	-899018.6	57.35
##	2014-03-11	15976	56.27	-20	898969.5	56.27
##	2014-03-20	-15976	55.85	-20	-892259.6	55.85
##	2014-04-24	15412	57.89	-20	892200.7	57.89
##	2014-05-06	-15412	56.29	-20	-867541.5	56.29
##	2014-05-12	15682	55.32	-20	867528.2	55.32

##	2014-05-14	-15682	55.87	-20	-876153.3	55.87
##	2014-08-05	15157	57.80	-20	876074.6	57.80
##	2014-08-14	-15157	57.57	-20	-872588.5	57.57
##	2014-09-04	14707	59.33	-20	872566.3	59.33
##	2014-09-15	-14707	59.96	-20	-881831.7	59.96
##	2014-09-17	14736	59.84	-20	881802.2	59.84
##	2014-09-26	-14736	57.31	-20	-844520.2	57.31
##	2014-10-06	14912	56.63	-20	844466.6	56.63
##	2014-10-15	-14912	52.56	-20	-783774.7	52.56
##	2014-10-17	14573	53.78	-20	783735.9	53.78
##	2014-10-20	-14573	51.76	-20	-754298.5	51.76
##	2014-10-28	14439	52.24	-20	754293.4	52.24
##	2014-11-06	-14439	54.45	-20	-786203.6	54.45
##	2014-12-04	13765	57.11	-20	786119.2	57.11
##	2014-12-09	-13765	56.48	-20	-777447.2	56.48
##	2014-12-11	13668	56.88	-20	777435.8	56.88
##	2014-12-22	-13668	57.85	-20	-790693.8	57.85
##	2015-01-08	14236	55.54	-20	790667.4	55.54
##	2015-01-19	-14236	57.51	-20	-818712.4	57.51
##	2015-02-10	13880	58.98	-20	818642.4	58.98
##	2015-02-16	-13880	60.44	-20	-838907.2	60.44
##	2015-03-27	12539	66.90	-20	838859.1	66.90
##	2015-04-01	-12539	66.82	-20	-837856.0	66.82
##	2015-05-07	12497	67.04	-20	837798.9	67.04
##	2015-05-12	-12497	67.31	-20	-841173.1	67.31
##	2015-05-13	12582	67.53	-20	849662.5	67.53
##	2015-05-21	-12582	68.17	-20	-857714.9	68.17
##	2015-06-10	13008	65.28	-20	849162.2	65.28
##	2015-06-12	-13008	65.11	-20	-846950.9	65.11
##	2015-06-16	13074	64.78	-20	846933.7	64.78
##	2015-06-24	-13074	66.59	-20	-870597.7	66.59
##	2015-06-25	13015	66.94	-20	871224.1	66.94
##	2015-07-06	-13015	61.58	-20	-801463.7	61.58
##	2015-08-10	12056	66.42	-20	800759.5	66.42
##	2015-08-19	-12056	62.76	-20	-756634.6	62.76
##	2015-08-25	12826	58.99	-20	756605.7	58.99
##	2015-08-28	-12826	59.46	-20	-762634.0	59.46
##	2015-09-11	12951	58.88	-20	762554.9	58.88
##	2015-09-22	-12951	56.55	-20	-732379.0	56.55
##	2015-09-25	12830	57.08	-20	732336.4	57.08
##	2015-10-01	-12830	57.79	-20	-741445.7	57.79
##	2015-10-02	12705	58.52	-20	743496.6	58.52
##	2015-10-13	-12705	63.75	-20	-809943.8	63.75
##	2015-11-25	10865	74.35	-20	807812.7	74.35
##	2015-11-27	-10865	74.05	-20	-804553.2	74.05
##	2015-12-07	11080	72.61	-20	804518.8	72.61
##	2015-12-08	-11080	72.55	-20	-803854.0	72.55
##	2015-12-15	11255	71.42	-20	803832.1	71.42
##	2015-12-18	-11255	72.03	-20	-810697.7	72.03
##	2015-12-23	11191	72.44	-20	810676.0	72.44
##	2016-01-07	-11191	69.60	-20	-778893.6	69.60
##	2016-01-21	10899	71.46	-20	778842.5	71.46
##	2016-01-22	-10899	70.34	-20	-766635.7	70.34
##	2016-02-25	10954	69.98	-20	766560.9	69.98

##	2016-03-07	-10954	69.57	-20	-762069.8	69.57
##	2016-03-09	10918	69.80	-20	762076.4	69.80
##	2016-03-10	-10918	69.40	-20	-757709.2	69.40
##	2016-03-29	10622	71.33	-20	757667.3	71.33
##	2016-04-07	-10622	68.19	-20	-724314.2	68.19
##	2016-04-11	10584	68.43	-20	724263.1	68.43
##	2016-04-20	-10584	70.19	-20	-742891.0	70.19
##	2016-04-27	10448	71.10	-20	742852.8	71.10
##	2016-04-28	-10448	70.18	-20	-733240.6	70.18
##	2016-05-12	10655	68.81	-20	733170.6	68.81
##	2016-05-24	-10655	69.19	-20	-737219.4	69.19
##	2016-06-03	10181	72.41	-20	737206.2	72.41
##	2016-06-08	-10181	71.64	-20	-729366.8	71.64
##	2016-09-13	9239	78.94	-20	729326.7	78.94
##	2016-09-22	-9239	82.36	-20	-760924.0	82.36
##	2016-10-14	9659	78.77	-20	760839.4	78.77
##	2016-10-17	-9659	78.49	-20	-758134.9	78.49
##	2016-10-27	9317	81.37	-20	758124.3	81.37
##	2016-11-07	-9317	78.30	-20	-729521.1	78.30
##	2016-12-05	9424	77.41	-20	729511.8	77.41
##	2016-12-12	-9424	79.32	-20	-747511.7	79.32
##	2017-02-02	8778	85.15	-20	747446.7	85.15
##	2017-02-13	-8778	87.38	-20	-767021.6	87.38
##	2017-03-28	8526	89.96	-20	766999.0	89.96
##	2017-04-06	-8526	91.43	-20	-779532.2	91.43
##	2017-04-12	8508	91.62	-20	779503.0	91.62
##	2017-04-25	-8508	93.13	-20	-792350.0	93.13
##	2017-05-12	8380	94.54	-20	792245.2	94.54
##	2017-05-23	-8380	94.68	-20	-793418.4	94.68
##	2017-06-16	8433	94.08	-20	793376.6	94.08
##	2017-06-23	-8433	95.44	-20	-804845.5	95.44
##	2017-06-29	8561	94.01	-20	804819.6	94.01
##	2017-07-10	-8561	91.31	-20	-781704.9	91.31
##	2017-07-12	8528	91.66	-20	781676.5	91.66
##	2017-07-20	-8528	90.18	-20	-769055.0	90.18
##	2017-08-14	8661	88.79	-20	769010.2	88.79
##	2017-08-17	-8661	89.96	-20	-779143.6	89.96
##	2017-08-22	8730	89.25	-20	779152.5	89.25
##	2017-08-31	-8730	88.10	-20	-769113.0	88.10
##	2017-09-05	8702	88.37	-20	768995.7	88.37
##	2017-09-13	-8702	91.50	-20	-796233.0	91.50
##	2017-11-14	8250	96.51	-20	796207.5	96.51
##	2017-11-22	-8250	96.44	-20	-795630.0	96.44
##	2017-12-12	8327	95.55	-20	795644.8	95.55
##	2017-12-19	-8327	97.65	-20	-813131.6	97.65
##	2018-01-03	8654	93.95	-20	813043.3	93.95
##	2018-01-12	-8654	90.73	-20	-785177.4	90.73
##	2018-01-16	8665	90.61	-20	785135.7	90.61
##	2018-01-17	-8665	90.05	-20	-780283.2	90.05
##	2018-02-07	9020	86.50	-20	780230.0	86.50
##	2018-02-16	-9020	84.31	-20	-760476.2	84.31
##	2018-02-20	8982	84.67	-20	760505.9	84.67
##	2018-03-01	-8982	84.10	-20	-755386.2	84.10
##	2018-03-05	9073	83.25	-20	755327.2	83.25

##	2018-03-07	-9073	84.42	-20	-765942.7	84.42
##	2018-03-26	9048	84.65	-20	765913.2	84.65
##	2018-03-28	-9048	83.45	-20	-755055.6	83.45
##	2018-04-12	8666	87.12	-20	754981.9	87.12
##	2018-04-16	-8666	87.46	-20	-757928.4	87.46
##	2018-04-17	8637	88.26	-20	762301.6	88.26
##	2018-04-25	-8637	89.20	-20	-770420.4	89.20
##	2018-05-16	7942	96.44	-20	765926.5	96.44
##	2018-05-18	-7942	95.72	-20	-760208.2	95.72
##	2018-05-30	7909	96.12	-20	760213.1	96.12
##	2018-06-04	-7909	96.72	-20	-764958.5	96.72
##	2018-06-20	7492	102.10	-20	764933.2	102.10
##	2018-06-21	-7492	101.78	-20	-762535.8	101.78
##	2018-06-27	7669	99.42	-20	762452.0	99.42
##	2018-06-28	-7669	97.75	-20	-749644.8	97.75
##	2018-07-05	7587	98.80	-20	749595.6	98.80
##	2018-07-11	-7587	100.82	-20	-764921.3	100.82
##	2018-07-12	7642	101.44	-20	775204.5	101.44
##	2018-07-19	-7642	103.90	-20	-794003.8	103.90
##	2018-08-13	7856	99.76	-20	783714.6	99.76
##	2018-08-22	-7856	102.54	-20	-805554.2	102.54
##	2018-09-07	8045	100.12	-20	805465.4	100.12
##	2018-09-18	-8045	103.20	-20	-830244.0	103.20
##	2018-09-21	8133	102.08	-20	830216.6	102.08
##	2018-09-27	-8133	107.02	-20	-870393.7	107.02
##	2018-10-09	8349	104.24	-20	870299.8	104.24
##	2018-10-11	-8349	98.71	-20	-824129.8	98.71
##	Net.Txn.Realized.PL					
##	1999-12-31		0.00			
##	2013-11-06		-20.00			
##	2013-11-08		7034.05			
##	2013-11-22		-20.00			
##	2013-11-29		12445.00			
##	2014-01-08		-20.00			
##	2014-01-09		-9154.16			
##	2014-01-15		-20.00			
##	2014-01-24		-64637.52			
##	2014-01-29		-20.00			
##	2014-02-07		-24657.50			
##	2014-02-21		-20.00			
##	2014-02-24		-5884.50			
##	2014-02-28		-20.00			
##	2014-03-03		-16009.52			
##	2014-03-11		-20.00			
##	2014-03-20		-6729.92			
##	2014-04-24		-20.00			
##	2014-05-06		-24679.20			
##	2014-05-12		-20.00			
##	2014-05-14		8605.10			
##	2014-08-05		-20.00			
##	2014-08-14		-3506.11			
##	2014-09-04		-20.00			
##	2014-09-15		9245.41			
##	2014-09-17		-20.00			

## 2014-09-26	-37302.08
## 2014-10-06	-20.00
## 2014-10-15	-60711.84
## 2014-10-17	-20.00
## 2014-10-20	-29457.46
## 2014-10-28	-20.00
## 2014-11-06	31890.19
## 2014-12-04	-20.00
## 2014-12-09	-8691.95
## 2014-12-11	-20.00
## 2014-12-22	13237.96
## 2015-01-08	-20.00
## 2015-01-19	28024.92
## 2015-02-10	-20.00
## 2015-02-16	20244.80
## 2015-03-27	-20.00
## 2015-04-01	-1023.12
## 2015-05-07	-20.00
## 2015-05-12	3354.19
## 2015-05-13	-20.00
## 2015-05-21	8032.48
## 2015-06-10	-20.00
## 2015-06-12	-2231.36
## 2015-06-16	-20.00
## 2015-06-24	23643.94
## 2015-06-25	-20.00
## 2015-07-06	-69780.40
## 2015-08-10	-20.00
## 2015-08-19	-44144.96
## 2015-08-25	-20.00
## 2015-08-28	6008.22
## 2015-09-11	-20.00
## 2015-09-22	-30195.83
## 2015-09-25	-20.00
## 2015-10-01	9089.30
## 2015-10-02	-20.00
## 2015-10-13	66427.15
## 2015-11-25	-20.00
## 2015-11-27	-3279.50
## 2015-12-07	-20.00
## 2015-12-08	-684.80
## 2015-12-15	-20.00
## 2015-12-18	6845.55
## 2015-12-23	-20.00
## 2016-01-07	-31802.44
## 2016-01-21	-20.00
## 2016-01-22	-12226.88
## 2016-02-25	-20.00
## 2016-03-07	-4511.14
## 2016-03-09	-20.00
## 2016-03-10	-4387.20
## 2016-03-29	-20.00
## 2016-04-07	-33373.08
## 2016-04-11	-20.00

## 2016-04-20	18607.84
## 2016-04-27	-20.00
## 2016-04-28	-9632.16
## 2016-05-12	-20.00
## 2016-05-24	4028.90
## 2016-06-03	-20.00
## 2016-06-08	-7859.37
## 2016-09-13	-20.00
## 2016-09-22	31577.38
## 2016-10-14	-20.00
## 2016-10-17	-2724.52
## 2016-10-27	-20.00
## 2016-11-07	-28623.19
## 2016-12-05	-20.00
## 2016-12-12	17979.84
## 2017-02-02	-20.00
## 2017-02-13	19554.94
## 2017-03-28	-20.00
## 2017-04-06	12513.22
## 2017-04-12	-20.00
## 2017-04-25	12827.08
## 2017-05-12	-20.00
## 2017-05-23	1153.20
## 2017-06-16	-20.00
## 2017-06-23	11448.88
## 2017-06-29	-20.00
## 2017-07-10	-23134.70
## 2017-07-12	-20.00
## 2017-07-20	-12641.44
## 2017-08-14	-20.00
## 2017-08-17	10113.37
## 2017-08-22	-20.00
## 2017-08-31	-10059.50
## 2017-09-05	-20.00
## 2017-09-13	27217.26
## 2017-11-14	-20.00
## 2017-11-22	-597.50
## 2017-12-12	-20.00
## 2017-12-19	17466.70
## 2018-01-03	-20.00
## 2018-01-12	-27885.88
## 2018-01-16	-20.00
## 2018-01-17	-4872.40
## 2018-02-07	-20.00
## 2018-02-16	-19773.80
## 2018-02-20	-20.00
## 2018-03-01	-5139.74
## 2018-03-05	-20.00
## 2018-03-07	10595.41
## 2018-03-26	-20.00
## 2018-03-28	-10877.60
## 2018-04-12	-20.00
## 2018-04-16	2926.44
## 2018-04-17	-20.00

```

## 2018-04-25      8098.78
## 2018-05-16      -20.00
## 2018-05-18     -5738.24
## 2018-05-30      -20.00
## 2018-06-04      4725.40
## 2018-06-20      -20.00
## 2018-06-21     -2417.44
## 2018-06-27      -20.00
## 2018-06-28    -12827.23
## 2018-07-05      -20.00
## 2018-07-11     15305.74
## 2018-07-12      -20.00
## 2018-07-19     18779.32
## 2018-08-13      -20.00
## 2018-08-22     21819.68
## 2018-09-07      -20.00
## 2018-09-18     24758.60
## 2018-09-21      -20.00
## 2018-09-27     40157.02
## 2018-10-09      -20.00
## 2018-10-11    -46189.97

```

```
## [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
## 2013-10-31	-31779	30.330	-20	-963857.1	30.330
## 2013-11-14	32766	29.415	-20	963811.9	29.415
## 2013-11-25	-32766	29.539	-20	-967874.9	29.539
## 2013-12-13	34066	28.411	-20	967849.1	28.411
## 2013-12-27	-34066	29.483	-20	-1004367.9	29.483
## 2014-02-05	33526	29.956	-20	1004304.9	29.956
## 2014-02-13	-33526	30.249	-20	-1014128.0	30.249
## 2014-02-21	33619	30.164	-20	1014083.5	30.164
## 2014-03-03	-33619	29.368	-20	-987322.8	29.368
## 2014-03-04	34019	29.645	-20	1008493.3	29.645
## 2014-03-13	-34019	26.853	-20	-913512.2	26.853
## 2014-03-17	33265	26.823	-20	892267.1	26.823
## 2014-03-26	-33265	27.368	-20	-910396.5	27.368
## 2014-03-28	33243	27.385	-20	910359.6	27.385
## 2014-04-08	-33243	27.496	-20	-914049.5	27.496
## 2014-04-28	34054	26.840	-20	914009.4	26.840
## 2014-04-30	-34054	27.036	-20	-920683.9	27.036
## 2014-05-26	35877	25.661	-20	920639.7	25.661
## 2014-06-04	-35877	25.295	-20	-907508.7	25.295
## 2014-10-31	40999	22.134	-20	907471.9	22.134
## 2014-11-11	-40999	21.545	-20	-883323.5	21.545
## 2014-11-21	40503	21.808	-20	883289.4	21.808
## 2014-12-02	-40503	23.415	-20	-948377.7	23.415
## 2014-12-05	39721	23.875	-20	948338.9	23.875
## 2014-12-16	-39721	21.826	-20	-866950.5	21.826
## 2015-03-27	30401	28.516	-20	866914.9	28.516
## 2015-04-09	-30401	29.453	-20	-895400.7	29.453
## 2015-06-02	36367	24.620	-20	895355.5	24.620
## 2015-06-04	-36367	24.687	-20	-897792.1	24.687

##	2015-06-22	36624	24.513	-20	897764.1	24.513
##	2015-06-24	-36624	24.856	-20	-910326.1	24.856
##	2015-08-17	34827	26.137	-20	910273.3	26.137
##	2015-08-26	-34827	22.969	-20	-799941.4	22.969
##	2015-09-14	34571	23.138	-20	799903.8	23.138
##	2015-09-23	-34571	21.858	-20	-755652.9	21.858
##	2015-09-25	34304	22.027	-20	755614.2	22.027
##	2015-10-01	-34304	21.344	-20	-732184.6	21.344
##	2015-10-14	31981	22.893	-20	732141.0	22.893
##	2015-10-23	-31981	24.772	-20	-792233.3	24.772
##	2015-11-06	34470	22.982	-20	792189.5	22.982
##	2015-11-17	-34470	21.478	-20	-740346.7	21.478
##	2015-11-25	34568	21.416	-20	740308.3	21.416
##	2015-12-04	-34568	20.706	-20	-715765.0	20.706
##	2016-01-05	36302	19.716	-20	715730.2	19.716
##	2016-01-14	-36302	17.855	-20	-648172.2	17.855
##	2016-01-19	37647	17.216	-20	648130.8	17.216
##	2016-01-25	-37647	15.373	-20	-578747.3	15.373
##	2016-02-10	44520	12.999	-20	578715.5	12.999
##	2016-02-11	-44520	12.049	-20	-536421.5	12.049
##	2016-02-22	38158	14.057	-20	536387.0	14.057
##	2016-02-23	-38158	13.629	-20	-520055.4	13.629
##	2016-03-23	34690	14.990	-20	520003.1	14.990
##	2016-04-05	-34690	12.540	-20	-435012.6	12.540
##	2016-06-02	30499	14.262	-20	434976.7	14.262
##	2016-06-08	-30499	13.580	-20	-414176.4	13.580
##	2016-06-15	34069	12.156	-20	414142.8	12.156
##	2016-06-24	-34069	11.603	-20	-395302.6	11.603
##	2016-07-01	35316	11.192	-20	395256.7	11.192
##	2016-07-04	-35316	10.893	-20	-384697.2	10.893
##	2016-09-15	32939	11.678	-20	384661.6	11.678
##	2016-09-16	-32939	10.889	-20	-358672.8	10.889
##	2016-10-14	32761	10.947	-20	358634.7	10.947
##	2016-10-25	-32761	11.750	-20	-384941.8	11.750
##	2016-12-20	23932	16.083	-20	384898.4	16.083
##	2016-12-30	-23932	15.396	-20	-368457.1	15.396
##	2017-01-11	22996	16.021	-20	368418.9	16.021
##	2017-01-12	-22996	15.771	-20	-362669.9	15.771
##	2017-01-13	23020	16.110	-20	370852.2	16.110
##	2017-01-16	-23020	15.744	-20	-362426.9	15.744
##	2017-01-31	20830	17.002	-20	354151.7	17.002
##	2017-02-02	-20830	16.494	-20	-343570.0	16.494
##	2017-02-03	21306	16.735	-20	356555.9	16.735
##	2017-02-14	-21306	16.440	-20	-350270.6	16.440
##	2017-03-08	21535	15.659	-20	337216.6	15.659
##	2017-03-17	-21535	15.940	-20	-343267.9	15.940
##	2017-03-21	21702	15.815	-20	343217.1	15.815
##	2017-03-30	-21702	15.830	-20	-343542.7	15.830
##	2017-04-12	22098	15.545	-20	343513.4	15.545
##	2017-04-25	-22098	16.940	-20	-374340.1	16.940
##	2017-06-07	24156	15.495	-20	374297.2	15.495
##	2017-06-12	-24156	15.530	-20	-375142.7	15.530
##	2017-06-19	24453	15.340	-20	375109.0	15.340
##	2017-06-28	-24453	15.740	-20	-384890.2	15.740



##	2017-07-03	24128	15.950	-20	384841.6	15.950
##	2017-07-12	-24128	16.440	-20	-396664.3	16.440
##	2017-07-24	25071	15.820	-20	396623.2	15.820
##	2017-07-27	-25071	16.000	-20	-401136.0	16.000
##	2017-08-14	27681	14.490	-20	401097.7	14.490
##	2017-08-15	-27681	14.450	-20	-399990.4	14.450
##	2017-09-21	29024	13.780	-20	399950.7	13.780
##	2017-10-02	-29024	14.490	-20	-420557.8	14.490
##	2017-10-05	28981	14.510	-20	420514.3	14.510
##	2017-10-09	-28981	14.590	-20	-422832.8	14.590
##	2017-10-11	29239	14.460	-20	422795.9	14.460
##	2017-10-19	-29239	14.195	-20	-415047.6	14.195
##	2017-10-24	28790	14.415	-20	415007.8	14.415
##	2017-11-03	-28790	14.510	-20	-417742.9	14.510
##	2017-11-24	26001	16.065	-20	417706.1	16.065
##	2017-11-30	-26001	15.885	-20	-413025.9	15.885
##	2017-12-04	25580	16.145	-20	412989.1	16.145
##	2017-12-05	-25580	15.980	-20	-408768.4	15.980
##	2017-12-07	25828	15.825	-20	408728.1	15.825
##	2017-12-18	-25828	17.100	-20	-441658.8	17.100
##	2018-01-02	27542	16.034	-20	441608.4	16.034
##	2018-01-05	-27542	15.884	-20	-437477.1	15.884
##	2018-01-18	28764	15.208	-20	437442.9	15.208
##	2018-01-29	-28764	15.486	-20	-445439.3	15.486
##	2018-02-12	34442	12.932	-20	445403.9	12.932
##	2018-02-21	-34442	13.582	-20	-467791.2	13.582
##	2018-03-05	36491	12.818	-20	467741.6	12.818
##	2018-03-14	-36491	12.710	-20	-463800.6	12.710
##	2018-03-29	41100	11.284	-20	463772.4	11.284
##	2018-04-03	-41100	11.192	-20	-459991.2	11.192
##	2018-04-09	38827	11.846	-20	459944.6	11.846
##	2018-04-18	-38827	11.726	-20	-455285.4	11.726
##	2018-04-23	38943	11.690	-20	455243.7	11.690
##	2018-05-03	-38943	11.314	-20	-440601.1	11.314
##	2018-06-11	45621	9.657	-20	440562.0	9.657
##	2018-06-20	-45621	9.589	-20	-437459.8	9.589
##	2018-07-26	41604	10.514	-20	437424.5	10.514
##	2018-08-01	-41604	10.986	-20	-457061.5	10.986
##	2018-08-14	44631	10.240	-20	457021.4	10.240
##	2018-08-23	-44631	9.895	-20	-441623.7	9.895
##	2018-08-24	44669	9.957	-20	444769.2	9.957
##	2018-09-04	-44669	9.839	-20	-439498.3	9.839
##	2018-10-04	44838	9.730	-20	436273.7	9.730
##	2018-10-11	-44838	9.583	-20	-429682.6	9.583
##	Net.Txn.Realized.PL					
##	1999-12-31		0.000			
##	2013-10-22		-20.000			
##	2013-10-31		-36152.723			
##	2013-11-14		-20.000			
##	2013-11-25		4042.984			
##	2013-12-13		-20.000			
##	2013-12-27		36498.752			
##	2014-02-05		-20.000			
##	2014-02-13		9803.118			

## 2014-02-21	-20.000
## 2014-03-03	-26780.724
## 2014-03-04	-20.000
## 2014-03-13	-95001.048
## 2014-03-17	-20.000
## 2014-03-26	18109.425
## 2014-03-28	-20.000
## 2014-04-08	3669.973
## 2014-04-28	-20.000
## 2014-04-30	6654.584
## 2014-05-26	-20.000
## 2014-06-04	-13150.982
## 2014-10-31	-20.000
## 2014-11-11	-24168.411
## 2014-11-21	-20.000
## 2014-12-02	65068.321
## 2014-12-05	-20.000
## 2014-12-16	-81408.329
## 2015-03-27	-20.000
## 2015-04-09	28465.737
## 2015-06-02	-20.000
## 2015-06-04	2416.589
## 2015-06-22	-20.000
## 2015-06-24	12542.032
## 2015-08-17	-20.000
## 2015-08-26	-110351.936
## 2015-09-14	-20.000
## 2015-09-23	-44270.880
## 2015-09-25	-20.000
## 2015-10-01	-23449.632
## 2015-10-14	-20.000
## 2015-10-23	60072.299
## 2015-11-06	-20.000
## 2015-11-17	-51862.880
## 2015-11-25	-20.000
## 2015-12-04	-24563.280
## 2016-01-05	-20.000
## 2016-01-14	-67578.022
## 2016-01-19	-20.000
## 2016-01-25	-69403.421
## 2016-02-10	-20.000
## 2016-02-11	-42314.000
## 2016-02-22	-20.000
## 2016-02-23	-16351.624
## 2016-03-23	-20.000
## 2016-04-05	-85010.500
## 2016-06-02	-20.000
## 2016-06-08	-20820.318
## 2016-06-15	-20.000
## 2016-06-24	-18860.157
## 2016-07-01	-20.000
## 2016-07-04	-10579.484
## 2016-09-15	-20.000
## 2016-09-16	-26008.871

## 2016-10-14	-20.000
## 2016-10-25	26287.083
## 2016-12-20	-20.000
## 2016-12-30	-16461.284
## 2017-01-11	-20.000
## 2017-01-12	-5769.000
## 2017-01-13	-20.000
## 2017-01-16	-8445.320
## 2017-01-31	-20.000
## 2017-02-02	-10601.640
## 2017-02-03	-20.000
## 2017-02-14	-6305.270
## 2017-03-08	-20.000
## 2017-03-17	6031.335
## 2017-03-21	-20.000
## 2017-03-30	305.530
## 2017-04-12	-20.000
## 2017-04-25	30806.710
## 2017-06-07	-20.000
## 2017-06-12	825.460
## 2017-06-19	-20.000
## 2017-06-28	9761.200
## 2017-07-03	-20.000
## 2017-07-12	11802.720
## 2017-07-24	-20.000
## 2017-07-27	4492.780
## 2017-08-14	-20.000
## 2017-08-15	-1127.240
## 2017-09-21	-20.000
## 2017-10-02	20587.040
## 2017-10-05	-20.000
## 2017-10-09	2298.480
## 2017-10-11	-20.000
## 2017-10-19	-7768.335
## 2017-10-24	-20.000
## 2017-11-03	2715.050
## 2017-11-24	-20.000
## 2017-11-30	-4700.180
## 2017-12-04	-20.000
## 2017-12-05	-4240.700
## 2017-12-07	-20.000
## 2017-12-18	32910.700
## 2018-01-02	-20.000
## 2018-01-05	-4151.300
## 2018-01-18	-20.000
## 2018-01-29	7976.392
## 2018-02-12	-20.000
## 2018-02-21	22367.300
## 2018-03-05	-20.000
## 2018-03-14	-3961.028
## 2018-03-29	-20.000
## 2018-04-03	-3801.200
## 2018-04-09	-20.000
## 2018-04-18	-4679.240

```
## 2018-04-23          -20.000
## 2018-05-03        -14662.568
## 2018-06-11          -20.000
## 2018-06-20        -3122.228
## 2018-07-26          -20.000
## 2018-08-01        19617.088
## 2018-08-14          -20.000
## 2018-08-23       -15417.695
## 2018-08-24          -20.000
## 2018-09-04        -5290.942
## 2018-10-04          -20.000
## 2018-10-11       -6611.186
```

## 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, "Percent.Loss"]),
    print(row.names(tstats[i, ]))
  )
  print(trades.tab)
}
```

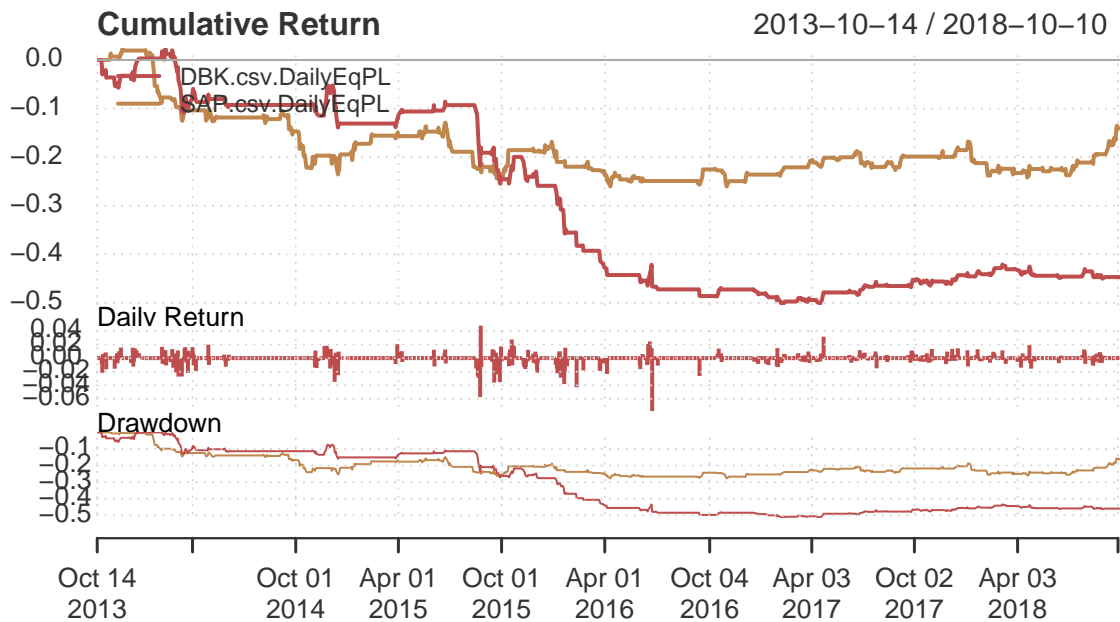
```
## [1] "SAP.csv"
##      [,1]      [,2]
## [1,] "Trades"  "77"
## [2,] "Win Percent" "46.7532467532467"
## [3,] "Loss Percent" "53.2467532467532"
## [4,] "W/L Ratio"  "0.878048780487805"
## [1] "DBK.csv"
##      [,1]      [,2]
## [1,] "Trades"  "64"
## [2,] "Win Percent" "40.625"
## [3,] "Loss Percent" "59.375"
## [4,] "W/L Ratio"  "0.684210526315789"
```

## 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)
```

## SAP.csv DBK.csv



### Calculate statistics of the Portfolio

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

# 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(somestats) <- c("Performance Metric", "Performance Value", "Risk Metric", "Risk Value")
  print(somestats)
}
```

	Performance Metric	Performance Value	Risk Metric	Risk Value
## 1	Cumulative Return	-0.4505548	Annualized StdDev	0.1058642
## 2	Annualized Return	-0.4505548	Max Drawdown	0.1058642
## 3	Annualized Sharp Ratio	-0.4505548	Value-at-Risk	0.1058642
## 4	Calmar Ratio	-0.4505548	Conditional VaR	0.1058642
## 1	Cumulative Return	-0.1127065	Annualized StdDev	0.5145028
## 2	Annualized Return	-0.1127065	Max Drawdown	0.5145028
## 3	Annualized Sharp Ratio	-0.1127065	Value-at-Risk	0.5145028
## 4	Calmar Ratio	-0.1127065	Conditional VaR	0.5145028

## 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, TxnFee=0)

# 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=0)

# 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 calculate 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)
#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	-0.0780	-0.0425	-0.0857
## Quartile 1	0.0000	0.0000	-0.0070
## Median	0.0000	0.0000	0.0010
## Arithmetic Mean	-0.0005	-0.0001	0.0003
## Geometric Mean	-0.0005	-0.0002	0.0002
## Quartile 3	0.0000	0.0000	0.0078
## Maximum	0.0477	0.0411	0.0607

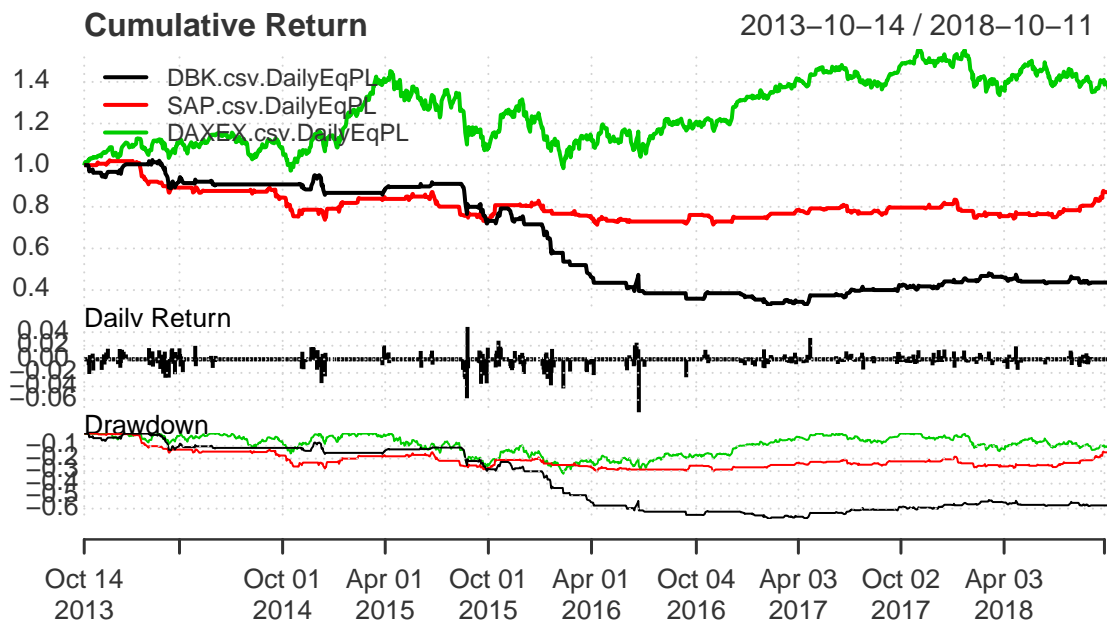
```
## SE Mean          0.0002          0.0002          0.0004
## LCL Mean (0.95)  -0.0008         -0.0005         -0.0005
## UCL Mean (0.95)  -0.0001          0.0002          0.0010
## Variance          0.0000          0.0000          0.0002
## Stdev             0.0067          0.0060          0.0143
## Skewness          -2.3825         -0.4247         -0.3312
## Kurtosis          26.8282         11.5792          2.0578
```

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

```
##                               DBK.csv.DailyEqPL SAP.csv.DailyEqPL
## Annualized Return              -0.1127          -0.0389
## Annualized Std Dev              0.1059           0.0957
## Annualized Sharpe (Rf=0%)       -1.0646          -0.4066
##                               DAXEX.csv.DailyEqPL
## Annualized Return              0.0459
## Annualized Std Dev              0.2269
## Annualized Sharpe (Rf=0%)       0.2024
```

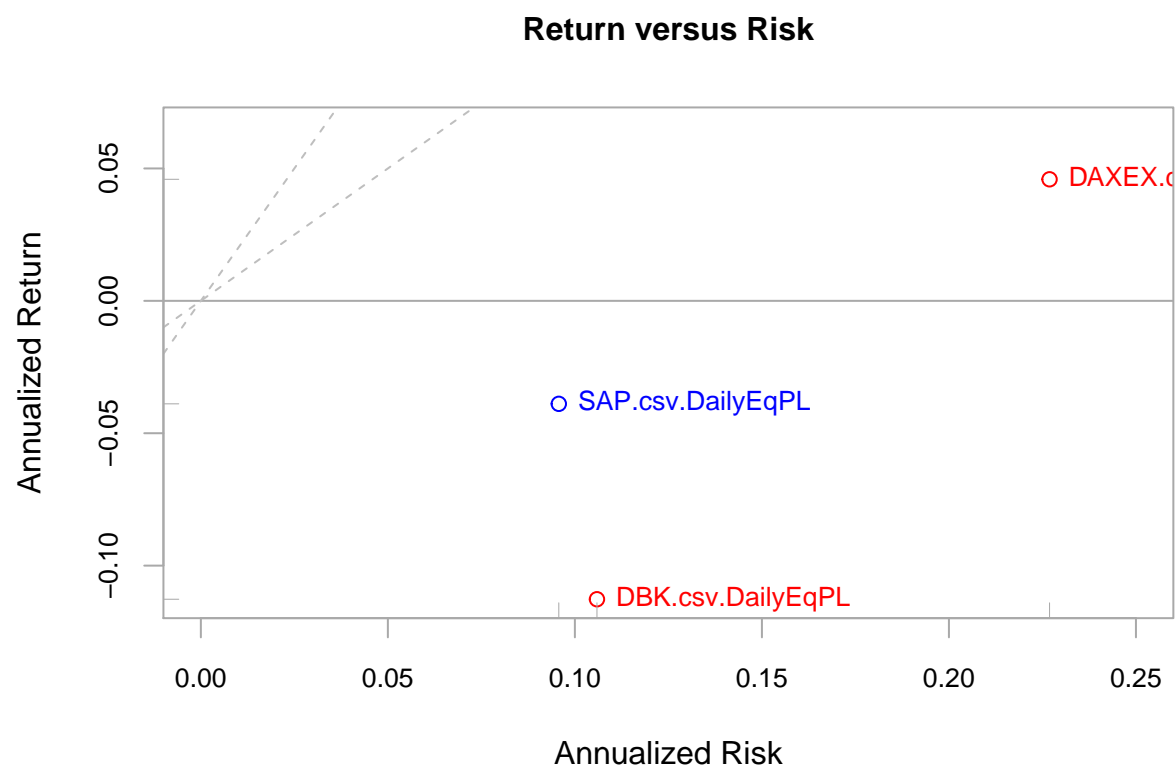
```
charts.PerformanceSummary(returns,geometric=FALSE,wealth.index=TRUE)
```

## DBK.csv.DailyEqPL Performance

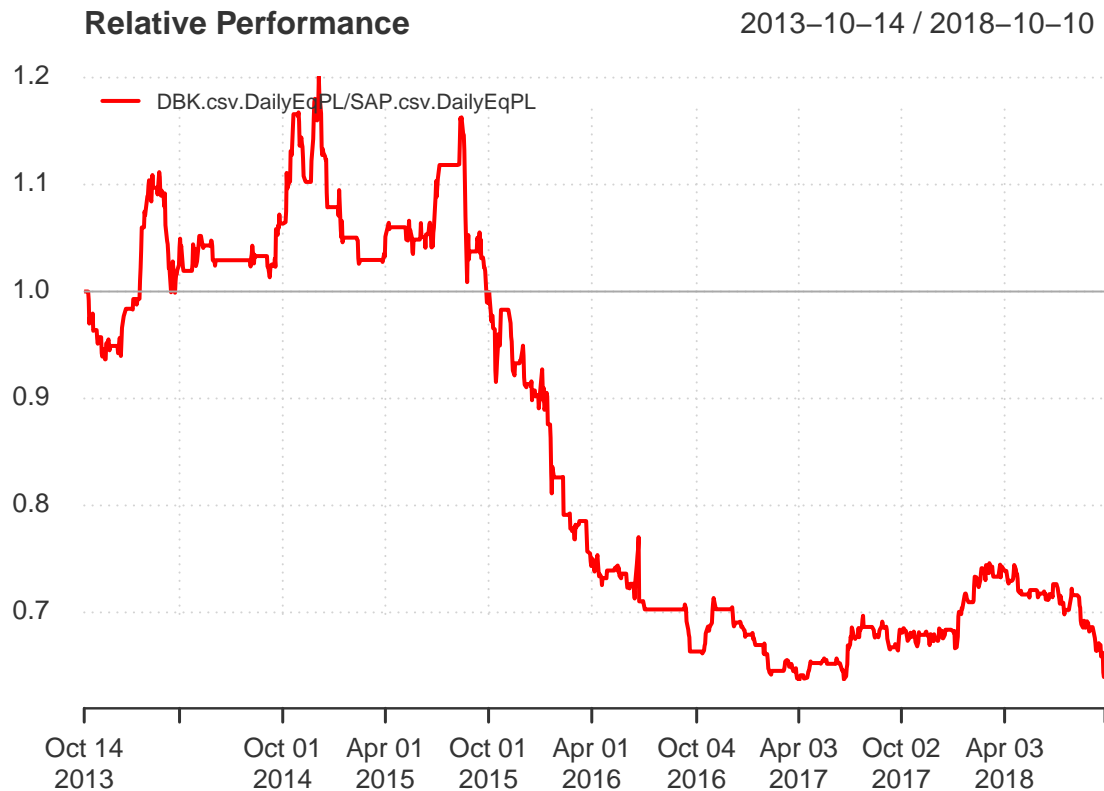


```
chart.RiskReturnScatter(returns,Rf=0,add.sharpe=c(1,2),xlim=c(0,0.25),main="Return versus Risk",colorse
```





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



## Backtest

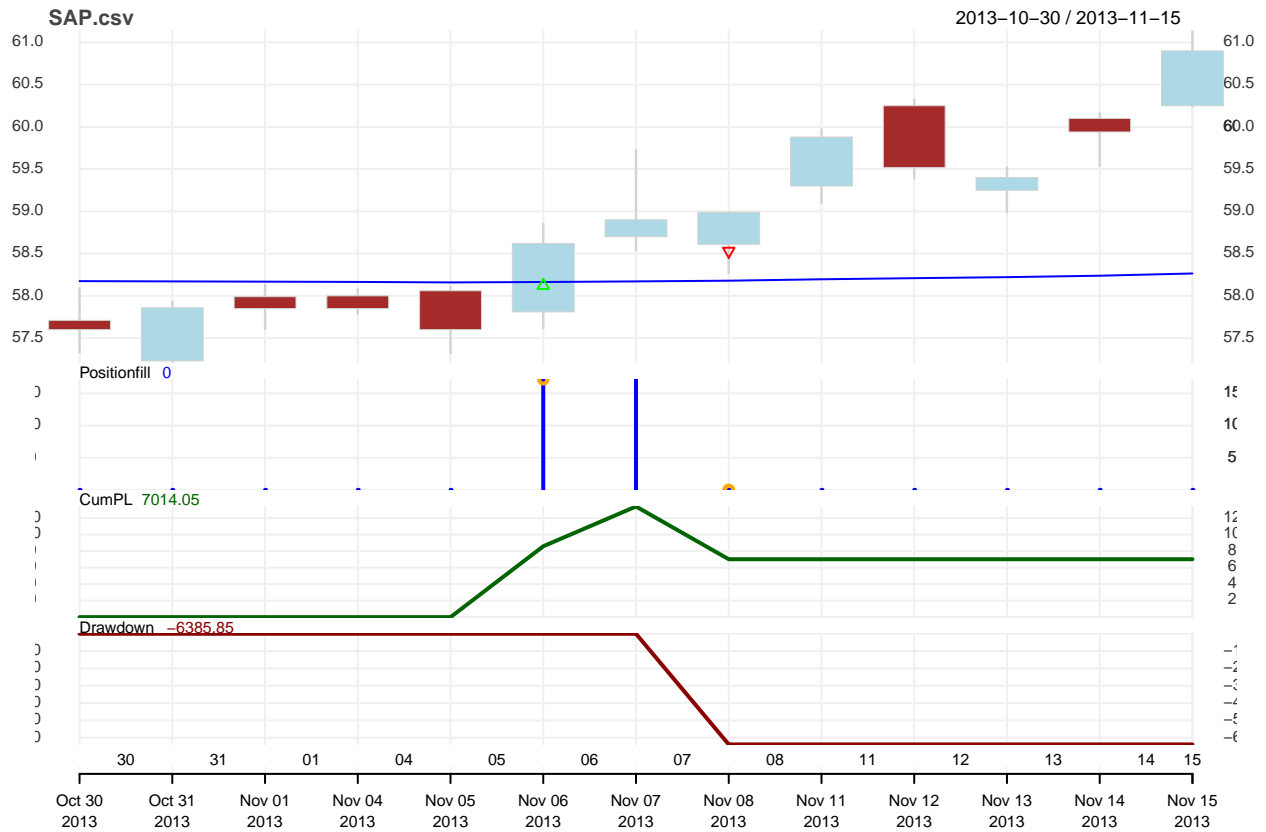
### Approach

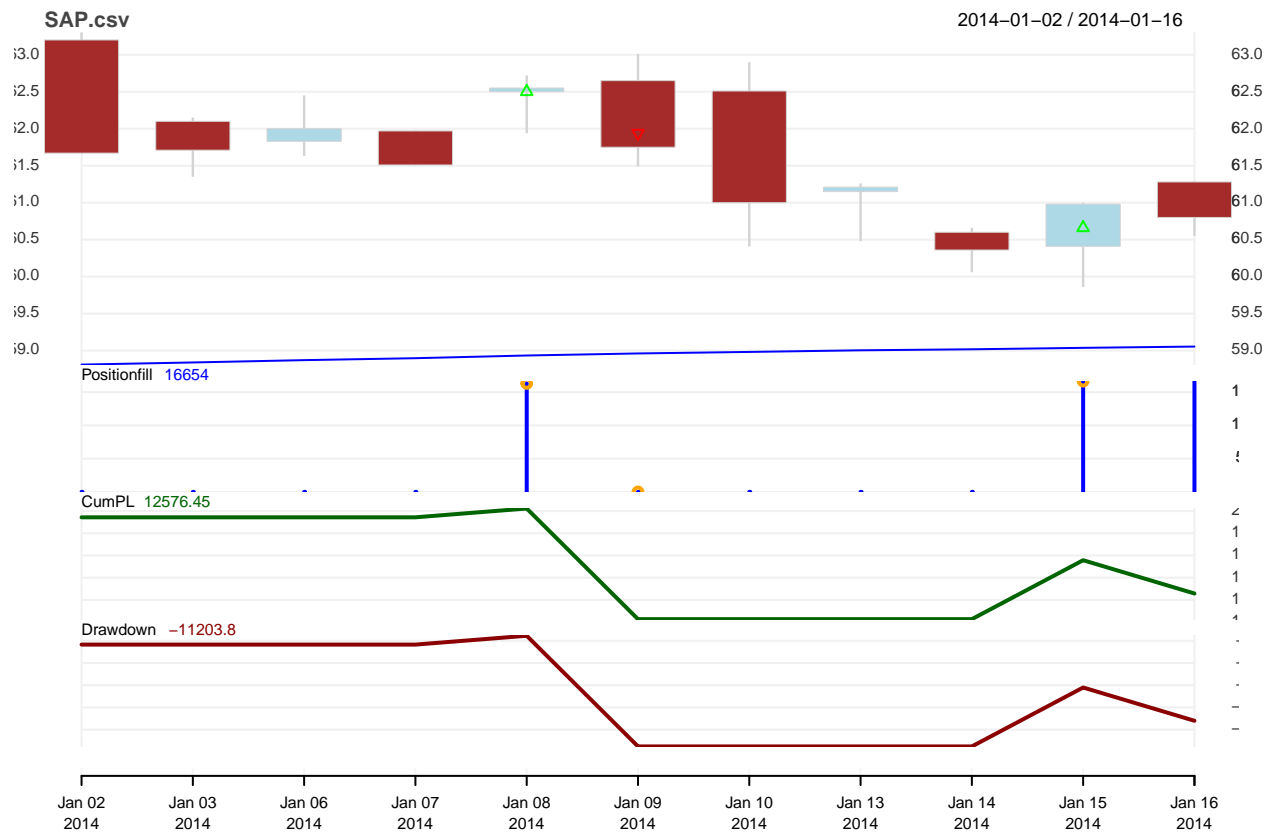
### Implementation

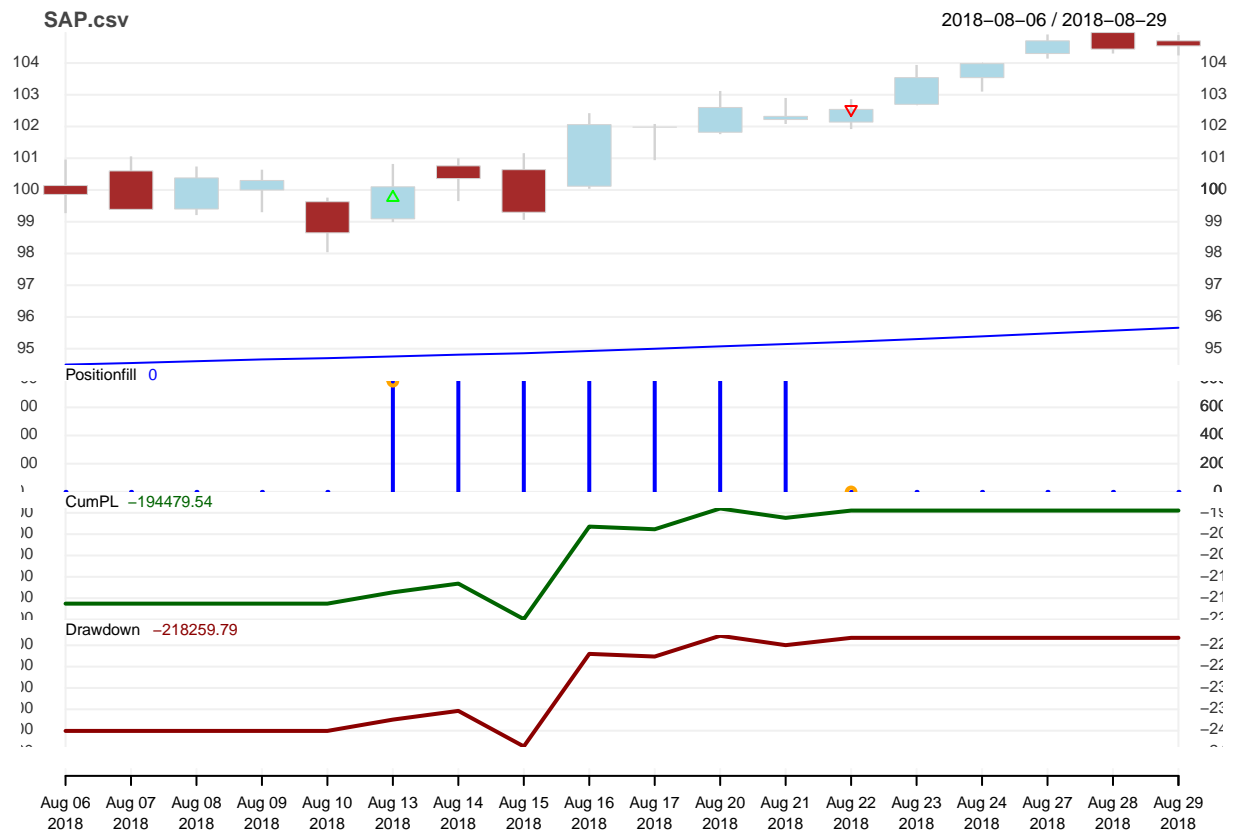
### System Check

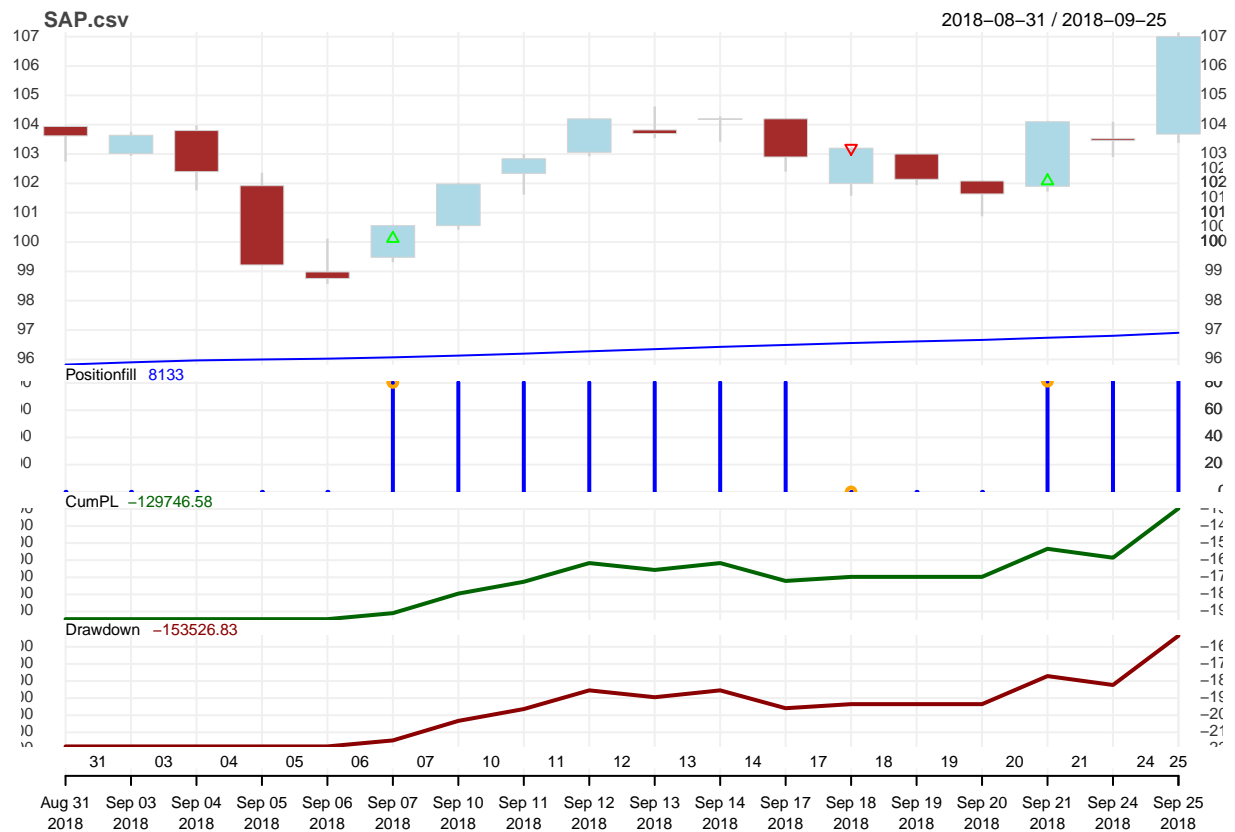
To make sure that our system worked properly some transactions were observed to make sure that the trades were performed according to the defined rules. Prepare the chart. Check 4 transactions. Visualize the charts with the transactions which should be checked.

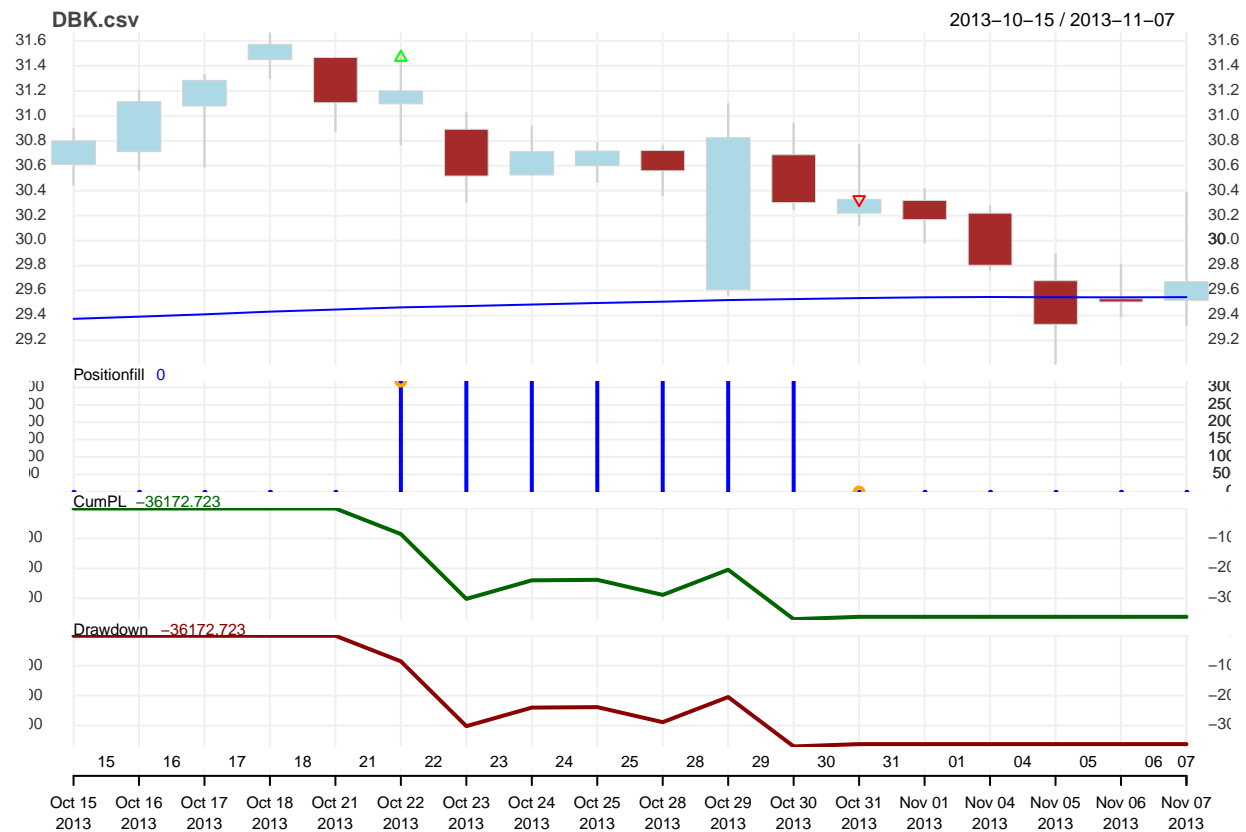
```
for (instrument in instrumentlist){
  rm(daterange_check)
  daterange_check <- c()
  transactionsInstrument <- getTxns(Portfolio=portfolioname,Symbol=instrument)
  for (i in c(2,6, (nrow(transactionsInstrument)-7), (nrow(transactionsInstrument)-5))) {
    from <- as.Date(index(transactionsInstrument[i,1]))-7
    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=daterange_check_i))
  }
}
```

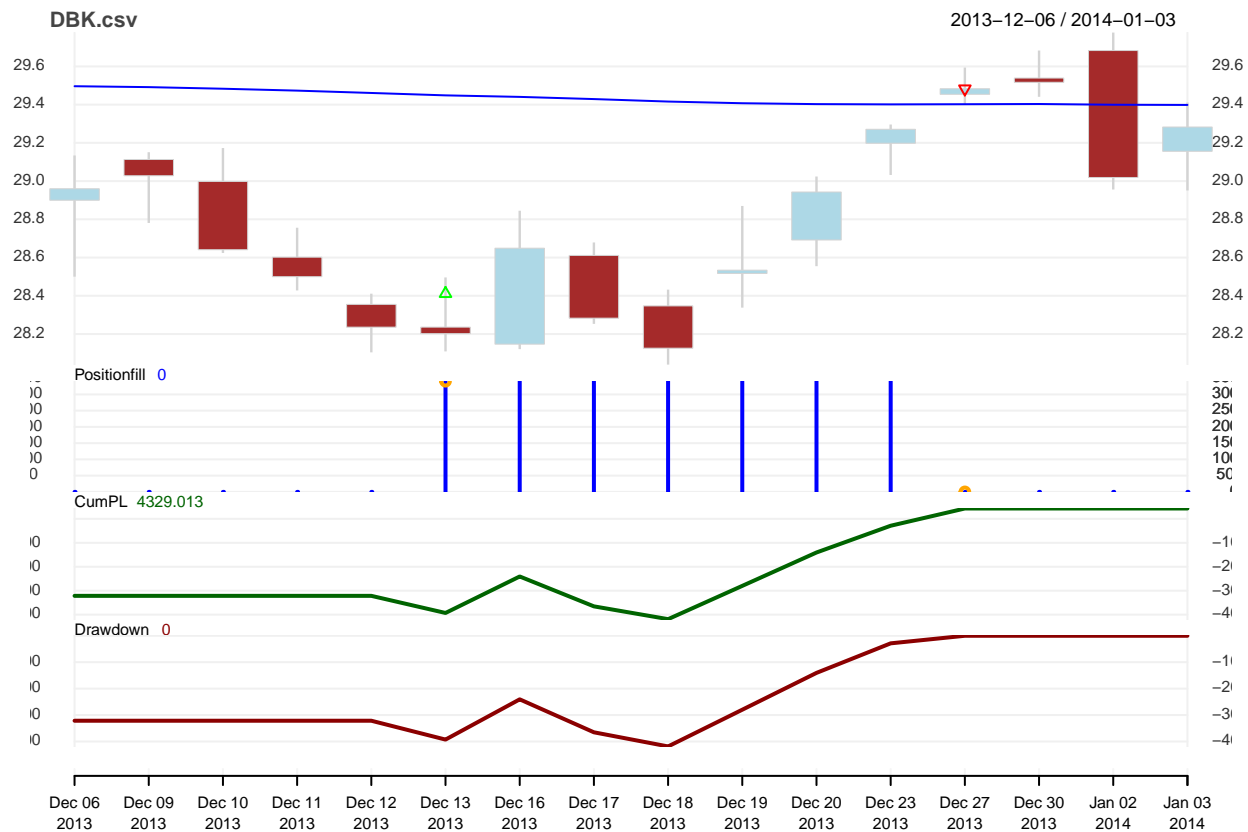




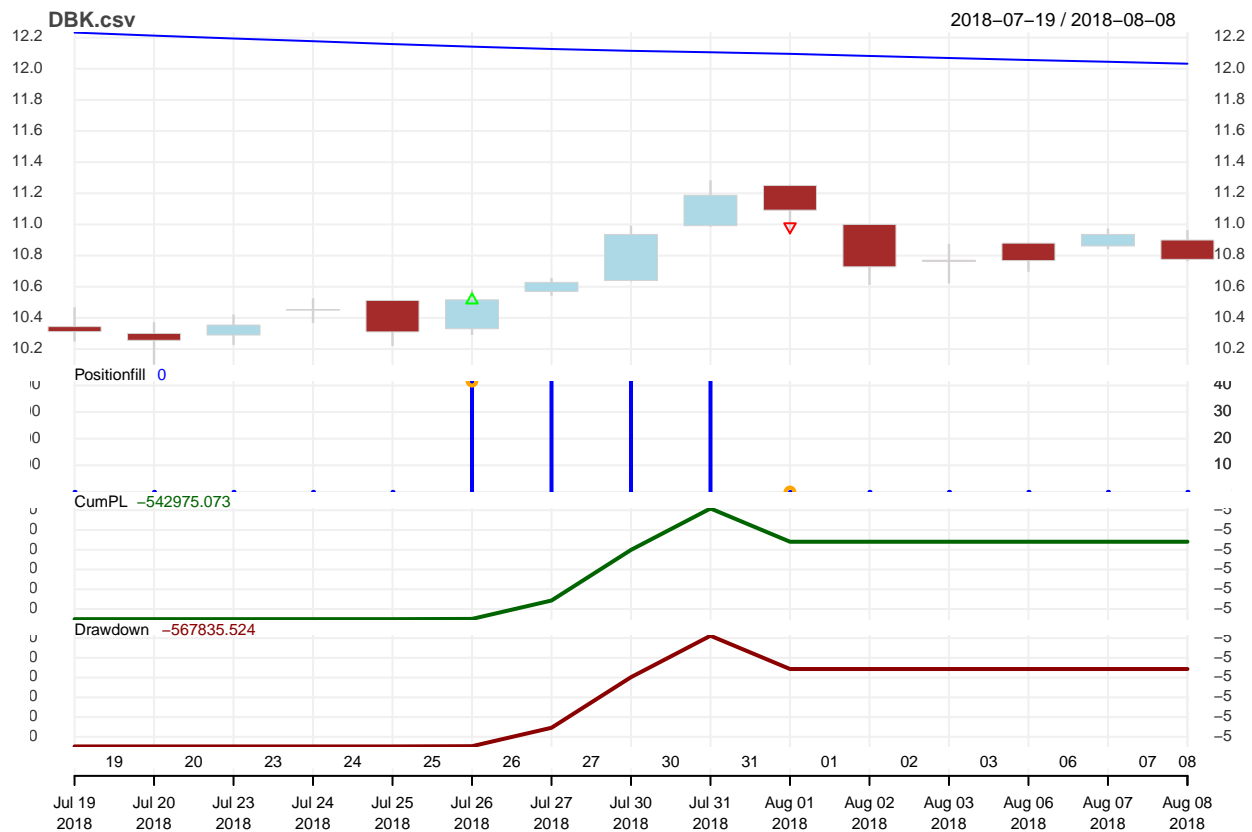


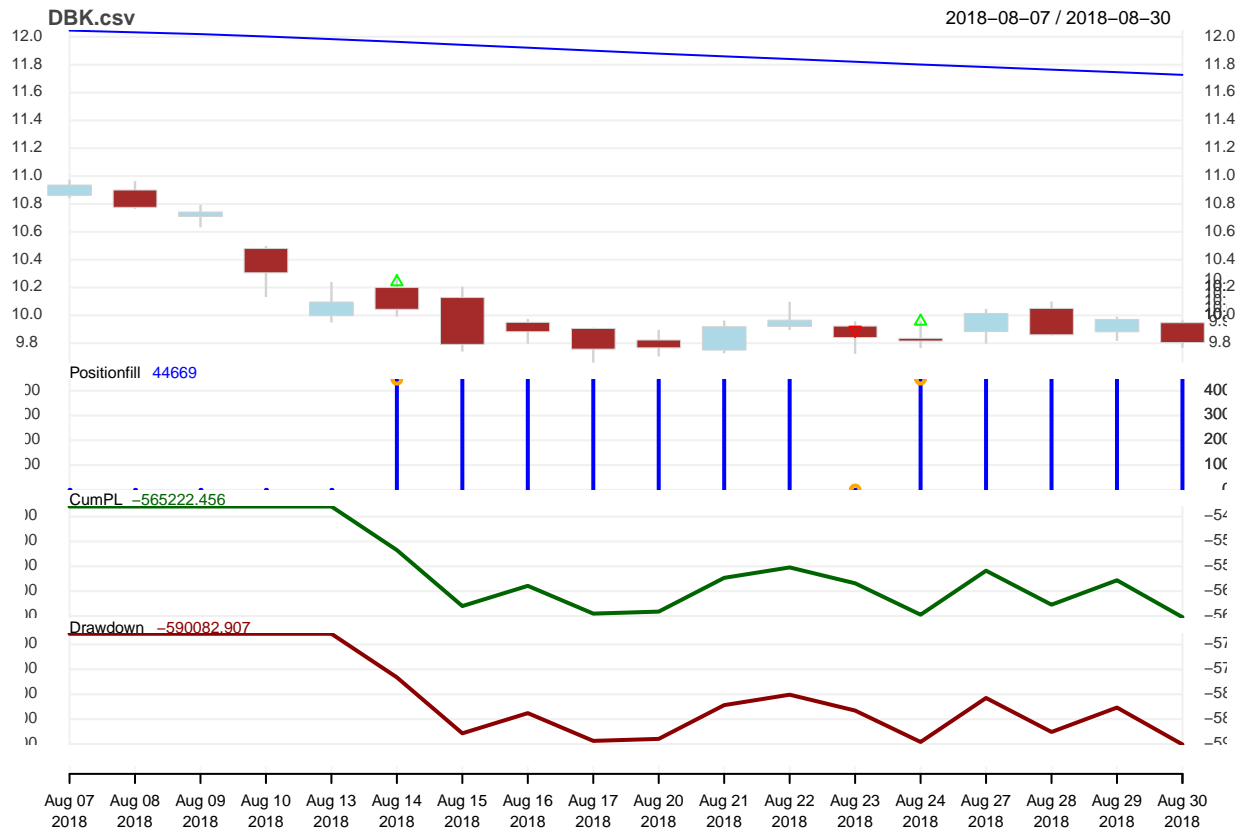












Analysis

Returns

Historical VaR

Equity curve

Conclusion and Suggestions