# Stock Market Analyses - Terend

Saeid Rezaei - STU - 205812010

Dec , 11 2021

# R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
################################################################################################
#######################
# Script name: SMP.SaeidRezaei.R
# Porpouse   : This script is developed to analyse the stock market for certain security and
#              provide prediction based on stock price (using time series method)
# Data source: Data source could be off-line (marketPriceHistory.csv) or online SP&500
# R Package usagae:
# quantmod
# ggplot2
# forecast
# plotly
# ggfortify
# tseries
# gridExtra
# docstring
# here
################################################################################################
#######################
#Developer         Date              Version                    Reason
#Saeid Rezaei     2021-12-10           0                      Initial Version
################################################################################################
#######################

# Start program
print ("Start program - Forcaste Stock Marekt")
```

```
## [1] "Start program - Forcaste Stock Marekt"
```

```
print ("STEP 1: Merging data into one file and value missing records")
```

```
## [1] "STEP 1: Merging data into one file and value missing records"
```

```
# If you are using off line market price you would need to execute
# DataClening.pl (Perl) script to merge files and value the secirities
# with missing price, The method is to value the missing price by looking into
# Previous price, if this is first row price would be Zero (0)
# Note: I'm running from my local drive. You would need to specify the path
# if you are running from other location
# Recomandation setup:
# Create subfolder in your local (C) drive call it CHM136
# Create another sub-directory under CHM136 call id StockPriceHist
# Copy all downloaded price .csv files there

system("perl C:/CHM136/DataCleaning.pl")
```

```
## Warning in system("perl C:/CHM136/DataCleaning.pl"): 'perl' not found
```

```
## [1] 127
```

```
print ("STEP 2: Analyse data and train data")
```

```
## [1] "STEP 2: Analyse data and train data"
```

```
print ("STEP 2.1: Install and Load R Packages")
```

```
## [1] "STEP 2.1: Install and Load R Packages"
```

```
#install.packages('quantmod')
#install.packages('binhf')
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
# Load data into Var.
# Load data from local .csv file into var.
#marketPriceHisotry <- read.csv( "C:/CHM136/StockPriceHist/output/secPriceHistory.csv")
#attach(marketPriceHisotry)

# Since Downloading data is not up-t-date, I used R PACKAGE CALLED quantmod to get realtime stoc
k price
# I'll use that source in my project going forward

print ("STEP 2.2: Get stock price from Yahoo and analyse data")
```

```
## [1] "STEP 2.2: Get stock price from Yahoo and analyse data"
```

```
getSymbols('SPY', src='yahoo')
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
## [1] "SPY"
```

```
getSymbols('^GSPC', src='yahoo')
```

```
## [1] "^GSPC"
```

```
getSymbols('^IBEX', src='yahoo')
```

```
## Warning: ^IBEX contains missing values. Some functions will not work if objects
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.
```

```
## [1] "^IBEX"
```

```
getSymbols(c('QQQ'), src='yahoo')
```

```
## [1] "QQQ"
```

```
head(GSPC)
```

```
##            GSPC.Open GSPC.High GSPC.Low GSPC.Close GSPC.Volume GSPC.Adjusted
## 2007-01-03   1418.03   1429.42  1407.86    1416.60  3429160000       1416.60
## 2007-01-04   1416.60   1421.84  1408.43    1418.34  3004460000       1418.34
## 2007-01-05   1418.34   1418.34  1405.75    1409.71  2919400000       1409.71
## 2007-01-08   1409.26   1414.98  1403.97    1412.84  2763340000       1412.84
## 2007-01-09   1412.84   1415.61  1405.42    1412.11  3038380000       1412.11
## 2007-01-10   1408.70   1415.99  1405.32    1414.85  2764660000       1414.85
```

```
tail(GSPC)
```

```
##            GSPC.Open GSPC.High GSPC.Low GSPC.Close GSPC.Volume GSPC.Adjusted
## 2021-12-08   4690.86   4705.06  4674.52    4701.21  3061550000       4701.21
## 2021-12-09   4691.00   4695.26  4665.98    4667.45  2851660000       4667.45
## 2021-12-10   4687.64   4713.57  4670.24    4712.02  2858310000       4712.02
## 2021-12-13   4710.30   4710.30  4667.60    4668.97  3322050000       4668.97
## 2021-12-14   4642.99   4660.47  4606.52    4634.09  3292740000       4634.09
## 2021-12-15   4636.46   4712.60  4611.22    4709.85  3367580000       4709.85
```

```
head(SPY)
```

```
##            SPY.Open SPY.High SPY.Low SPY.Close SPY.Volume SPY.Adjusted
## 2007-01-03   142.25   142.86  140.57    141.37   94807600     105.4467
## 2007-01-04   141.23   142.05  140.61    141.67   69620600     105.6705
## 2007-01-05   141.33   141.40  140.38    140.54   76645300     104.8277
## 2007-01-08   140.82   141.41  140.25    141.19   71655000     105.3125
## 2007-01-09   141.31   141.60  140.40    141.07   75680100     105.2230
## 2007-01-10   140.58   141.57  140.30    141.54   72428000     105.5735
```

```
tail(SPY)
```

```
##            SPY.Open SPY.High SPY.Low SPY.Close SPY.Volume SPY.Adjusted
## 2021-12-08   468.70   470.00  466.83    469.52   72238800       469.52
## 2021-12-09   468.15   469.63  466.14    466.35   61272600       466.35
## 2021-12-10   469.23   470.90  466.51    470.74   76949400       470.74
## 2021-12-13   470.19   470.56  466.27    466.57   87724700       466.57
## 2021-12-14   463.09   465.74  460.25    463.36   97264100       463.36
## 2021-12-15   463.42   470.86  460.74    470.60  116899300       470.60
```

```
# Remove the null values
QQQ <- QQQ[!(rowSums(is.na(QQQ))),]
SPY <- SPY[!(rowSums(is.na(SPY))),]

GSPC <- GSPC[!(rowSums(is.na(GSPC))),]
IBEX <- IBEX[!(rowSums(is.na(IBEX))),]



# GSPC and SPY are Time sereies data, Let's find the class
class(GSPC)
```

```
## [1] "xts" "zoo"
```

```
# Create a vector and put more than one symbol into that
# This VAR will being used to compare more than one symbol
# and analyse the market
basketSymbols <-(c('YELP','AAPL','AMZN'))
getSymbols(basketSymbols, src='yahoo')
```

```
## [1] "YELP" "AAPL" "AMZN"
```

```
# Analyse the Data
summary(YELP)
```

```
##      Index              YELP.Open       YELP.High        YELP.Low
##  Min.   :2012-03-02   Min.   :14.49   Min.   : 15.26   Min.   :12.89
##  1st Qu.:2014-08-14   1st Qu.:26.45   1st Qu.: 27.22   1st Qu.:25.71
##  Median :2017-01-25   Median :35.65   Median : 36.16   Median :35.09
##  Mean   :2017-01-24   Mean   :38.25   Mean   : 39.03   Mean   :37.47
##  3rd Qu.:2019-07-09   3rd Qu.:43.83   3rd Qu.: 44.36   3rd Qu.:43.25
##  Max.   :2021-12-15   Max.   :99.80   Max.   :101.75   Max.   :97.25
##    YELP.Close     YELP.Volume       YELP.Adjusted
##  Min.   :14.46   Min.   :  179800   Min.   :14.46
##  1st Qu.:26.40   1st Qu.:  913300   1st Qu.:26.40
##  Median :35.59   Median : 1507000   Median :35.59
##  Mean   :38.24   Mean   : 2166734   Mean   :38.24
##  3rd Qu.:43.81   3rd Qu.: 2498600   3rd Qu.:43.81
##  Max.   :98.04   Max.   :47155000   Max.   :98.04
```

```
summary(AAPL)
```

```
##      Index              AAPL.Open         AAPL.High          AAPL.Low      
##  Min.   :2007-01-03   Min.   :  2.835   Min.   :  2.929   Min.   :  2.793  
##  1st Qu.:2010-09-28   1st Qu.: 10.234   1st Qu.: 10.352   1st Qu.: 10.089  
##  Median :2014-06-25   Median : 23.591   Median : 23.835   Median : 23.361  
##  Mean   :2014-06-25   Mean   : 34.736   Mean   : 35.102   Mean   : 34.372  
##  3rd Qu.:2018-03-21   3rd Qu.: 42.839   3rd Qu.: 43.228   3rd Qu.: 42.456  
##  Max.   :2021-12-15   Max.   :181.120   Max.   :182.130   Max.   :175.530  
##    AAPL.Close       AAPL.Volume        AAPL.Adjusted    
##  Min.   :  2.793   Min.   :4.100e+07   Min.   :  2.394  
##  1st Qu.: 10.267   1st Qu.:1.248e+08   1st Qu.:  8.802  
##  Median : 23.596   Median :2.611e+08   Median : 21.072  
##  Mean   : 34.753   Mean   :3.961e+08   Mean   : 33.222  
##  3rd Qu.: 42.804   3rd Qu.:5.423e+08   3rd Qu.: 41.198  
##  Max.   :179.450   Max.   :3.373e+09   Max.   :179.450  
```

```
summary(AMZN)
```

```
##      Index              AMZN.Open         AMZN.High          AMZN.Low      
##  Min.   :2007-01-03   Min.   :  35.29   Min.   :  37.07   Min.   :  34.68  
##  1st Qu.:2010-09-28   1st Qu.: 154.96   1st Qu.: 156.67   1st Qu.: 152.45  
##  Median :2014-06-25   Median : 333.15   Median : 335.52   Median : 327.12  
##  Mean   :2014-06-25   Mean   : 860.95   Mean   : 870.06   Mean   : 850.89  
##  3rd Qu.:2018-03-21   3rd Qu.:1472.29   3rd Qu.:1503.25   3rd Qu.:1449.78  
##  Max.   :2021-12-15   Max.   :3744.00   Max.   :3773.08   Max.   :3696.79  
##    AMZN.Close       AMZN.Volume        AMZN.Adjusted    
##  Min.   :  35.03   Min.   :   881300   Min.   :  35.03  
##  1st Qu.: 155.22   1st Qu.:  3027925   1st Qu.: 155.22  
##  Median : 332.30   Median :  4353650   Median : 332.30  
##  Mean   : 860.74   Mean   :  5464611   Mean   : 860.74  
##  3rd Qu.:1481.69   3rd Qu.:  6547300   3rd Qu.:1481.69  
##  Max.   :3731.41   Max.   :104329200   Max.   :3731.41  
```

```
# Merge all there symbol data into one data frame
basket <- data.frame(as.xts(merge(YELP,AAPL,AMZN)))
# N/A respresents when Symbol does not have have price
head(basket)
```

```
##            YELP.Open YELP.High YELP.Low YELP.Close YELP.Volume YELP.Adjusted
## 2007-01-03        NA        NA       NA         NA          NA            NA
## 2007-01-04        NA        NA       NA         NA          NA            NA
## 2007-01-05        NA        NA       NA         NA          NA            NA
## 2007-01-08        NA        NA       NA         NA          NA            NA
## 2007-01-09        NA        NA       NA         NA          NA            NA
## 2007-01-10        NA        NA       NA         NA          NA            NA
##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
## 2007-01-03  3.081786  3.092143 2.925000   2.992857  1238319600      2.565971
## 2007-01-04  3.001786  3.069643 2.993571   3.059286   847260400      2.622925
## 2007-01-05  3.063214  3.078571 3.014286   3.037500   834741600      2.604247
## 2007-01-08  3.070000  3.090357 3.045714   3.052500   797106800      2.617107
## 2007-01-09  3.087500  3.320714 3.041071   3.306071  3349298400      2.834510
## 2007-01-10  3.383929  3.492857 3.337500   3.464286  2952880000      2.970158
##            AMZN.Open AMZN.High AMZN.Low AMZN.Close AMZN.Volume AMZN.Adjusted
## 2007-01-03     38.68     39.06    38.05      38.70    12405100         38.70
## 2007-01-04     38.59     39.14    38.26      38.90     6318400         38.90
## 2007-01-05     38.72     38.79    37.60      38.37     6619700         38.37
## 2007-01-08     38.22     38.31    37.17      37.50     6783000         37.50
## 2007-01-09     37.60     38.06    37.34      37.78     5703000         37.78
## 2007-01-10     37.49     37.70    37.07      37.15     6527500         37.15
```

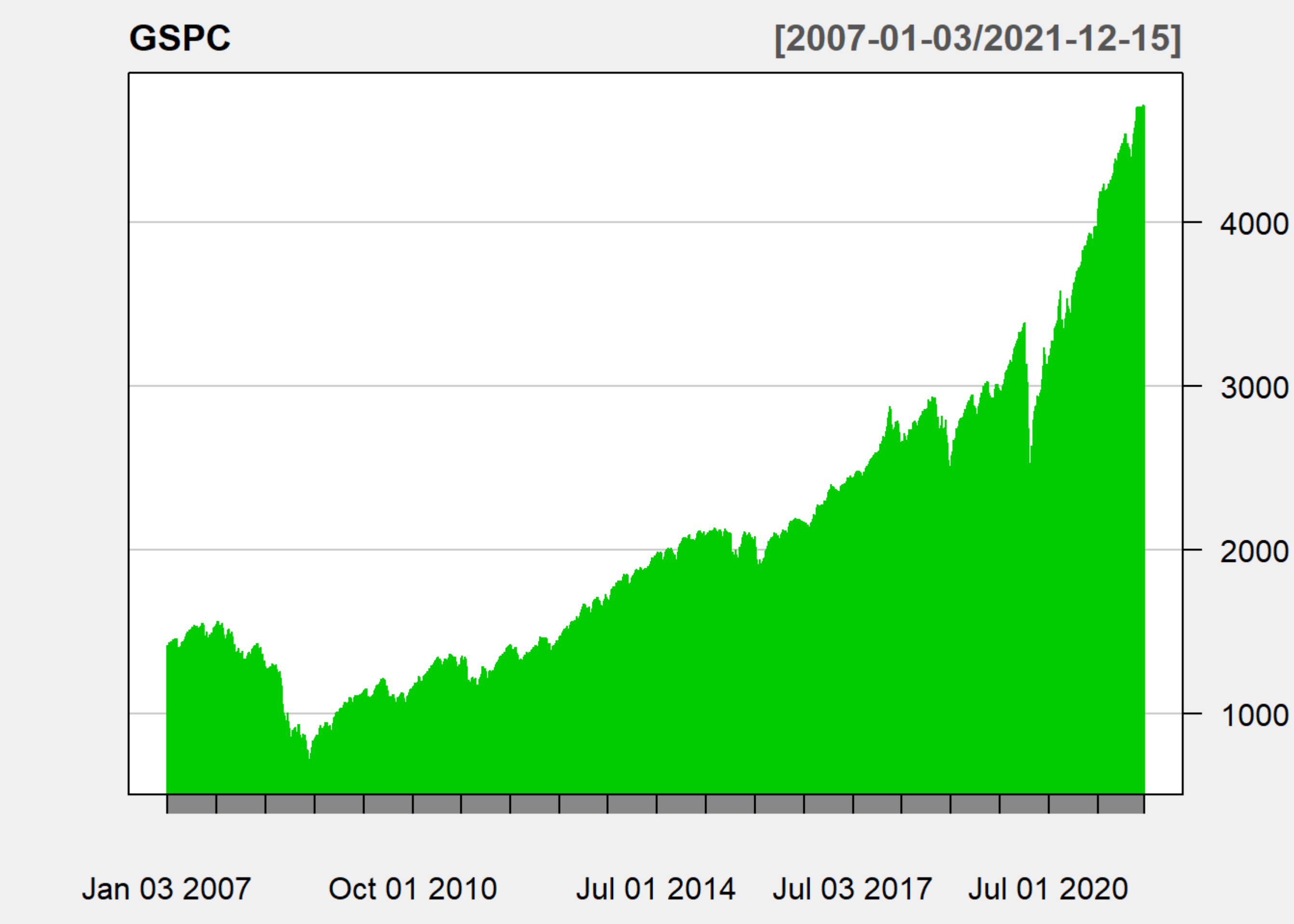
```
tail(basket)
```

```
##            YELP.Open YELP.High YELP.Low YELP.Close YELP.Volume YELP.Adjusted
## 2021-12-08     36.75     37.67    36.51      37.27      390800         37.27
## 2021-12-09     37.13     37.74    36.51      36.54      463000         36.54
## 2021-12-10     36.83     37.25    35.44      36.04      987300         36.04
## 2021-12-13     35.79     36.13    34.92      35.53      550700         35.53
## 2021-12-14     35.18     35.70    34.99      35.34      482200         35.34
## 2021-12-15     34.96     35.59    33.89      35.49      896300         35.49
##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
## 2021-12-08    172.13    175.96   170.70     175.08   116998900        175.08
## 2021-12-09    174.91    176.75   173.92     174.56   108923700        174.56
## 2021-12-10    175.21    179.63   174.69     179.45   115228100        179.45
## 2021-12-13    181.12    182.13   175.53     175.74   153237000        175.74
## 2021-12-14    175.25    177.74   172.21     174.33   139380400        174.33
## 2021-12-15    175.11    179.50   172.31     179.30   131063300        179.30
##            AMZN.Open AMZN.High AMZN.Low AMZN.Close AMZN.Volume AMZN.Adjusted
## 2021-12-08   3523.01   3543.60  3495.01    3523.16     2262700       3523.16
## 2021-12-09   3515.00   3539.39  3482.79    3483.42     2303100       3483.42
## 2021-12-10   3508.34   3518.54  3410.00    3444.24     3031400       3444.24
## 2021-12-13   3440.00   3442.00  3382.60    3391.35     3108500       3391.35
## 2021-12-14   3351.00   3389.98  3328.80    3381.83     2798800       3381.83
## 2021-12-15   3371.96   3472.00  3303.90    3466.30     3789700       3466.30
```

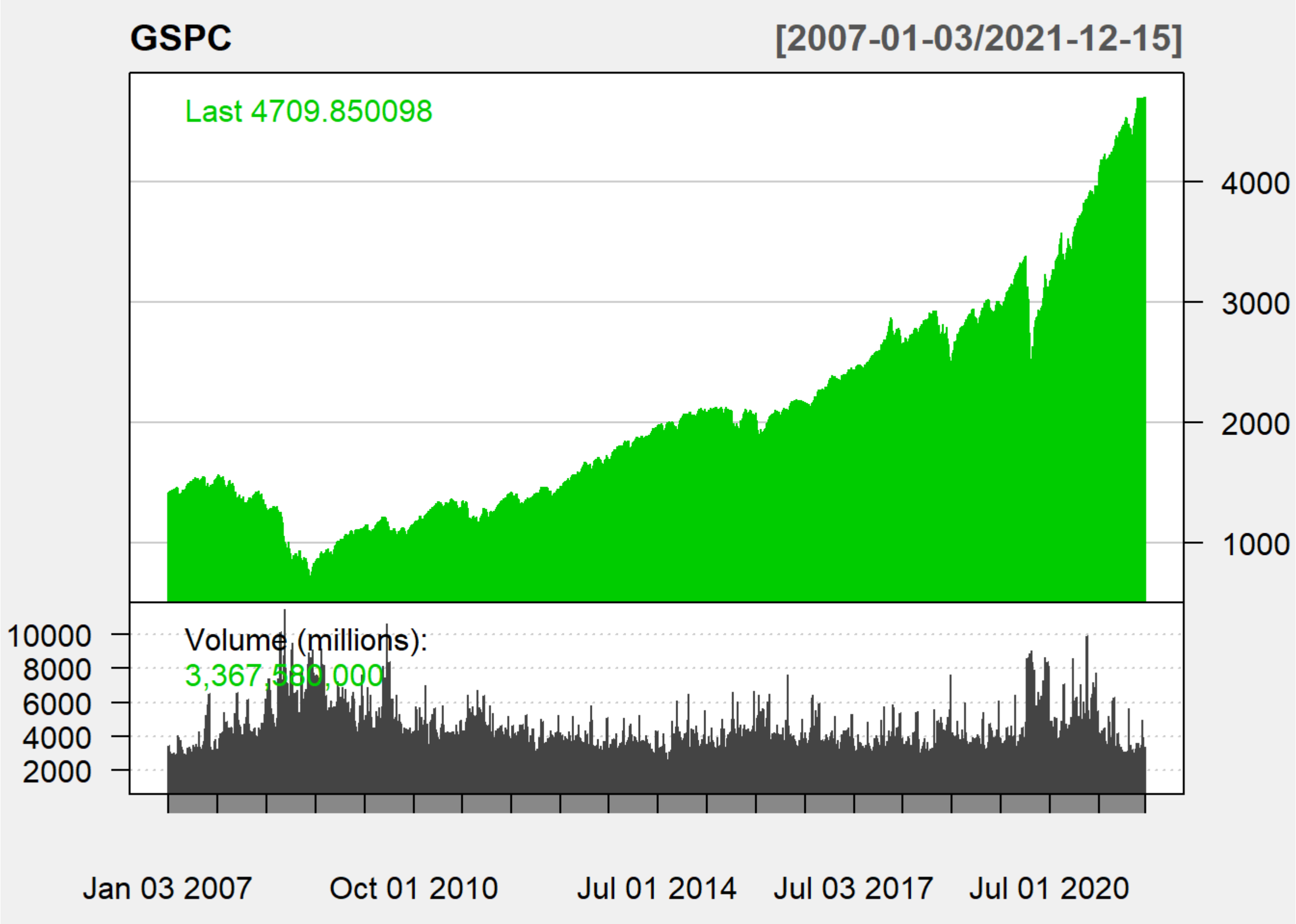
```
# Draw few charts to do basid analyses
print ("STEP 2.3: Draw few charts and analyse them")
```

```
## [1] "STEP 2.3: Draw few charts and analyse them"
```

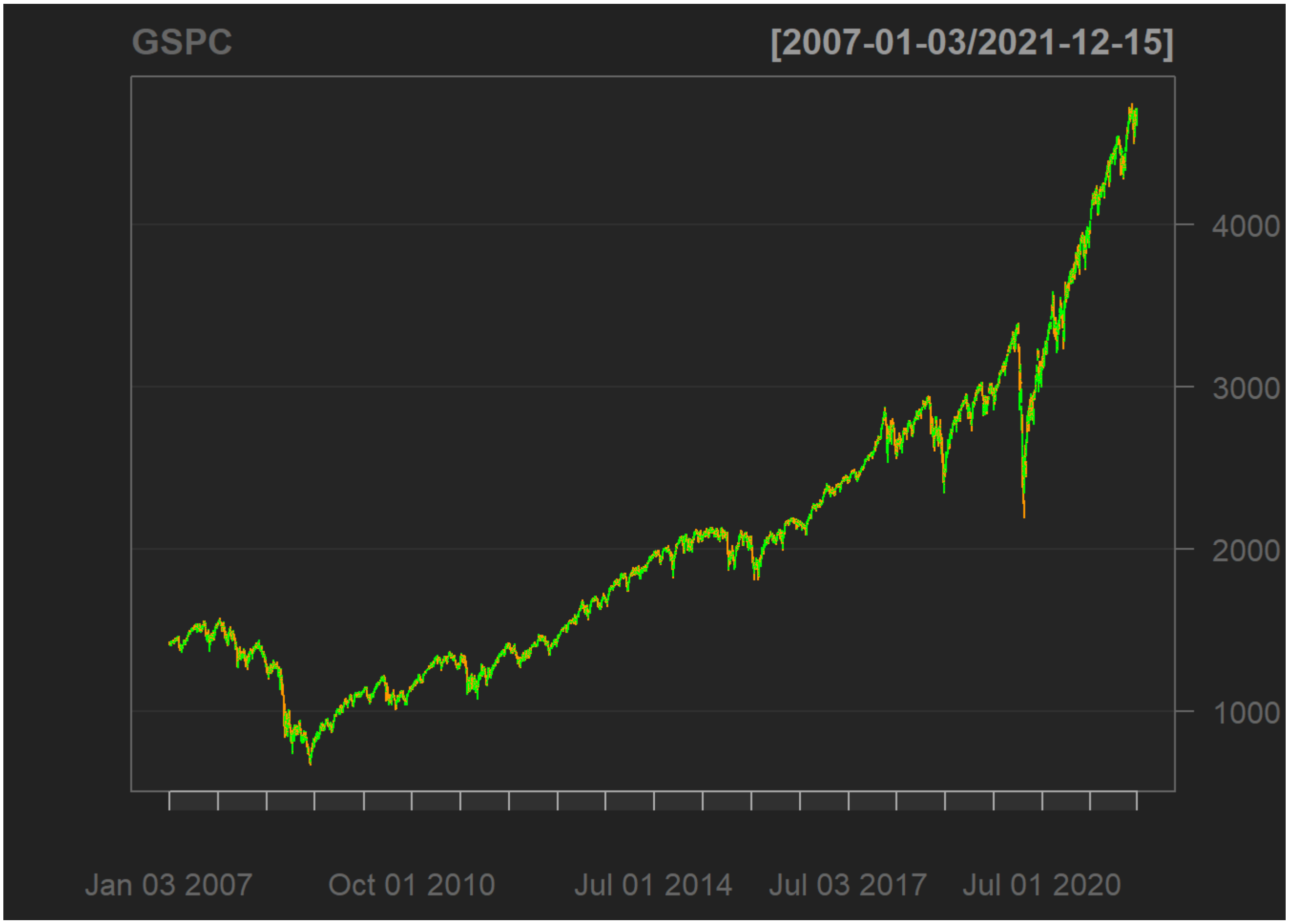
```
lineChart(GSPC,line.type = 'h',theme = 'white',TA=NULL)
```



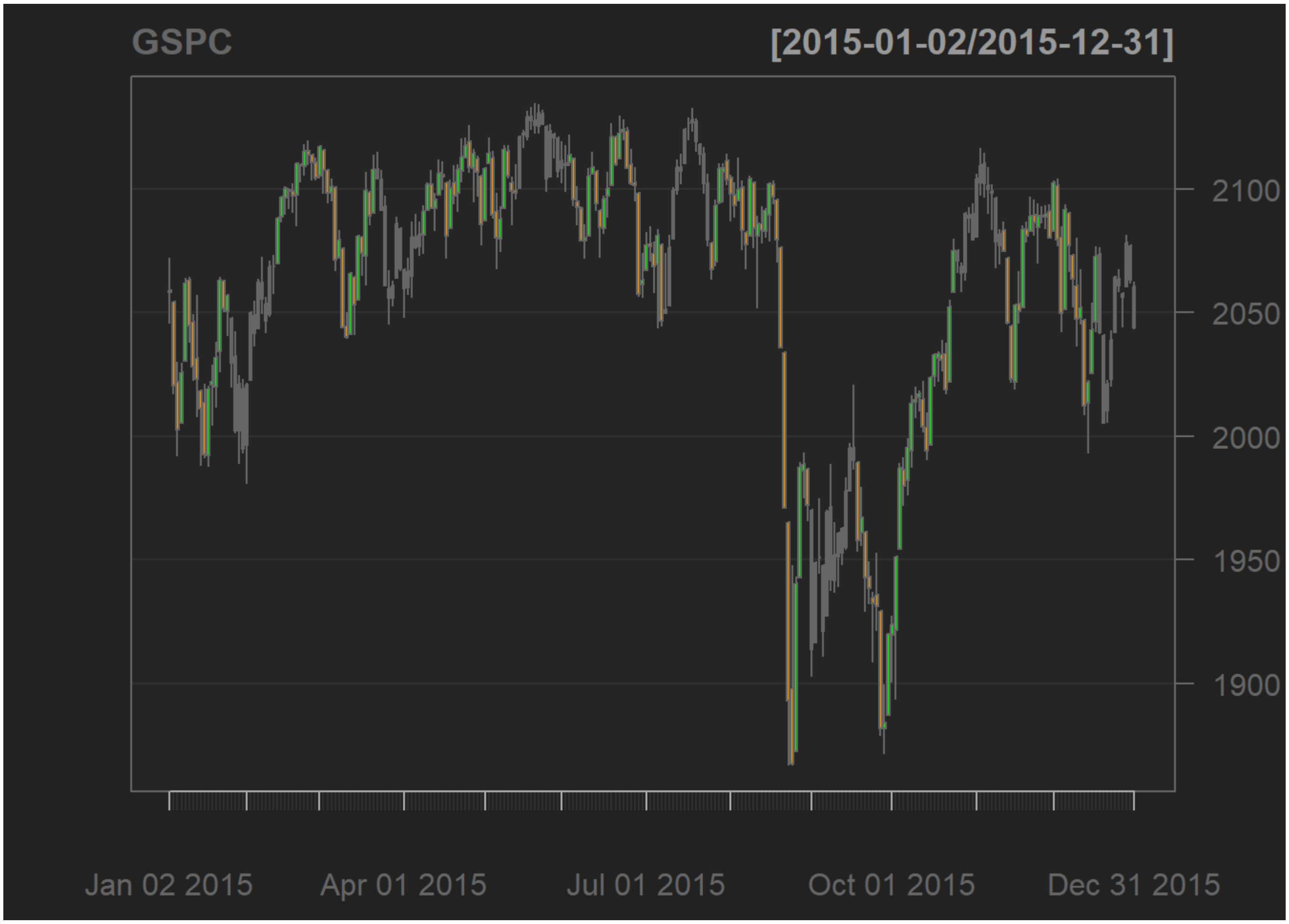
```
# put the volumn
lineChart(GSPC,line.type = 'h',theme = 'white')
```



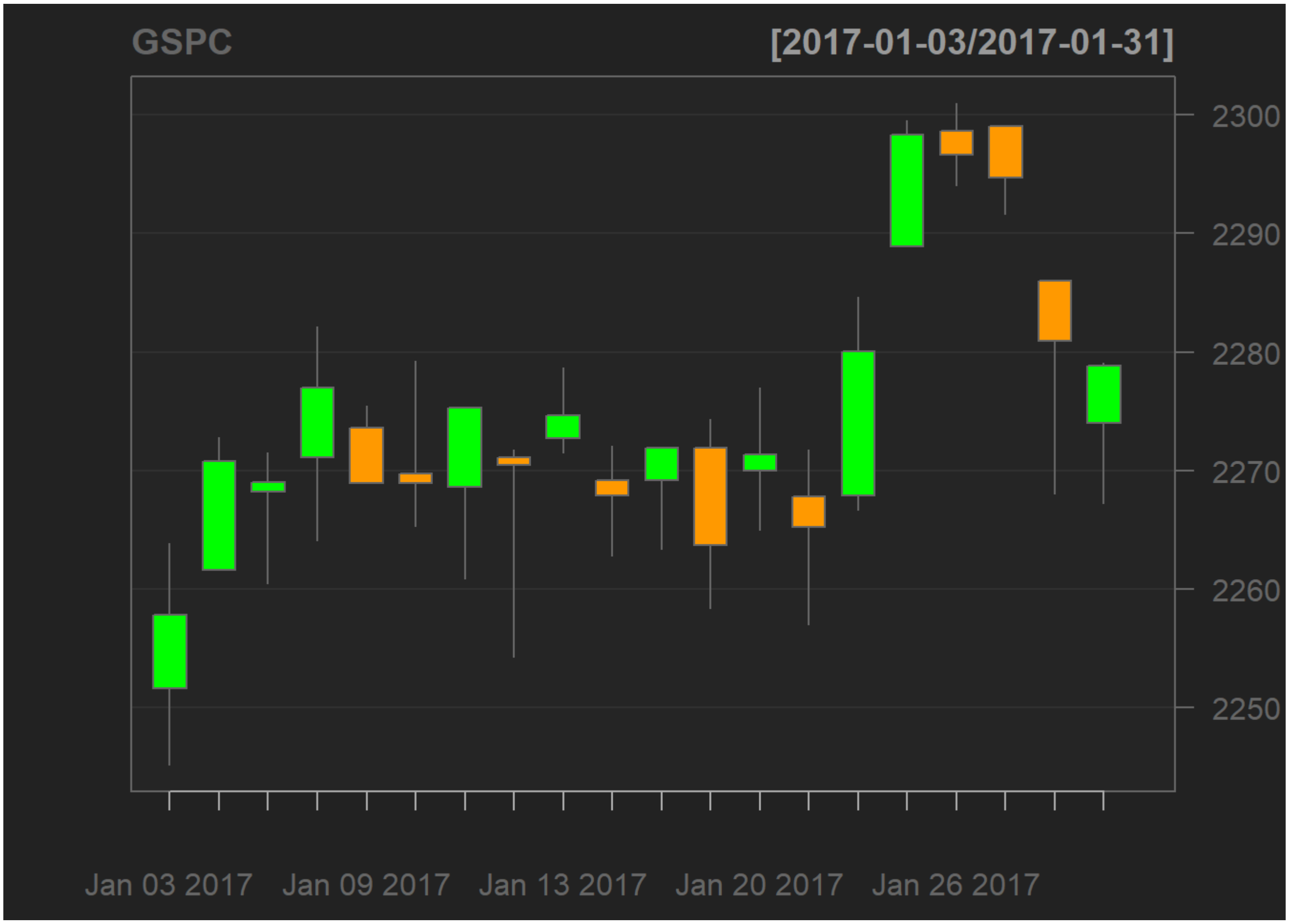
```
barChart(GSPC,bar.type = 'hcl',TA=NULL)
```



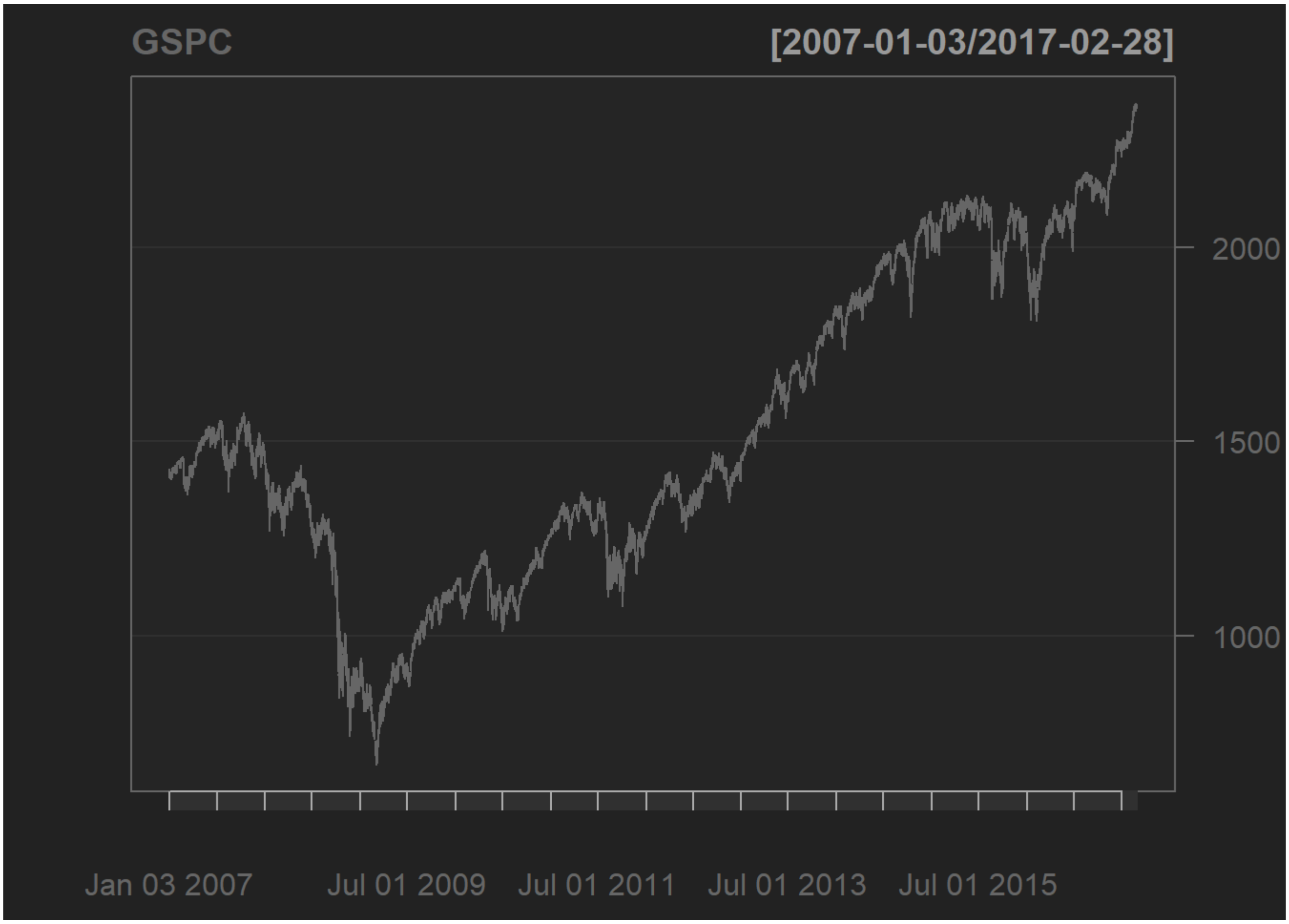
```
candleChart(GSPC,TA=NULL,subset = '2015')
```



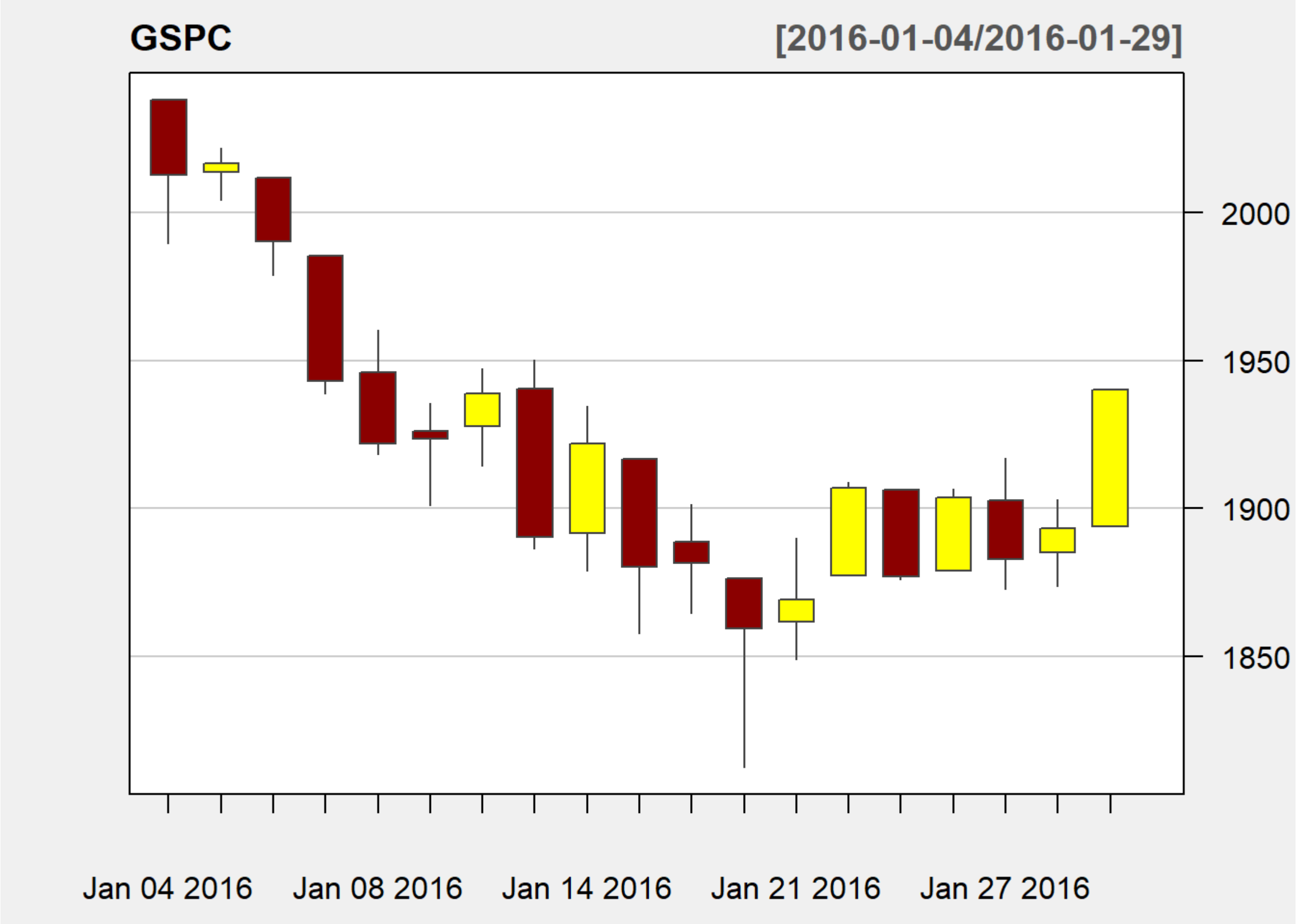
```
# Fucase on Jan 2017
candleChart(GSPC,TA=NULL,subset = '2017-01')
```



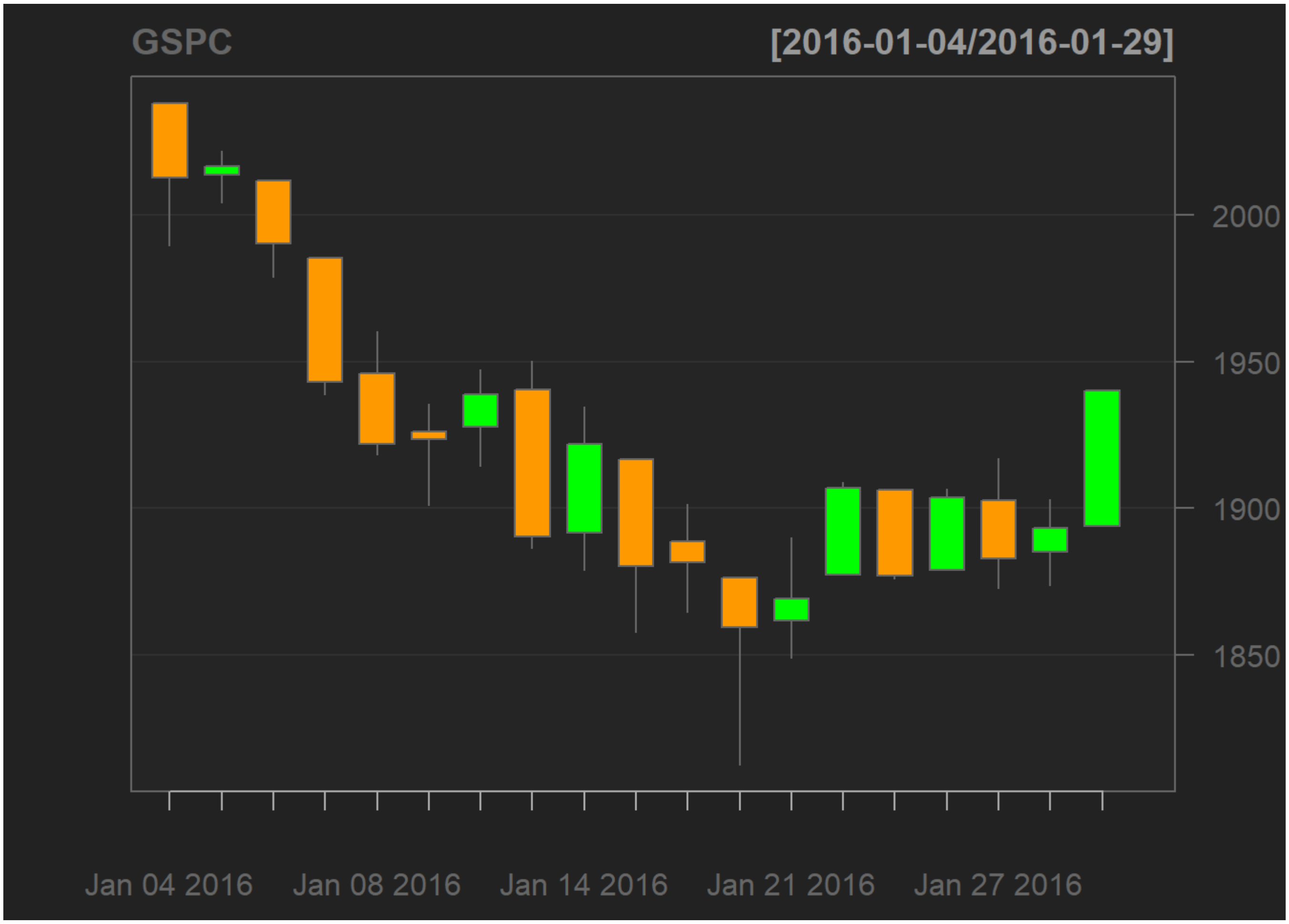
```
# Review the price changes from Feb 2017 and backward to 1st day
candleChart(GSPC,TA=NULL,subset = '::2017-02')
```



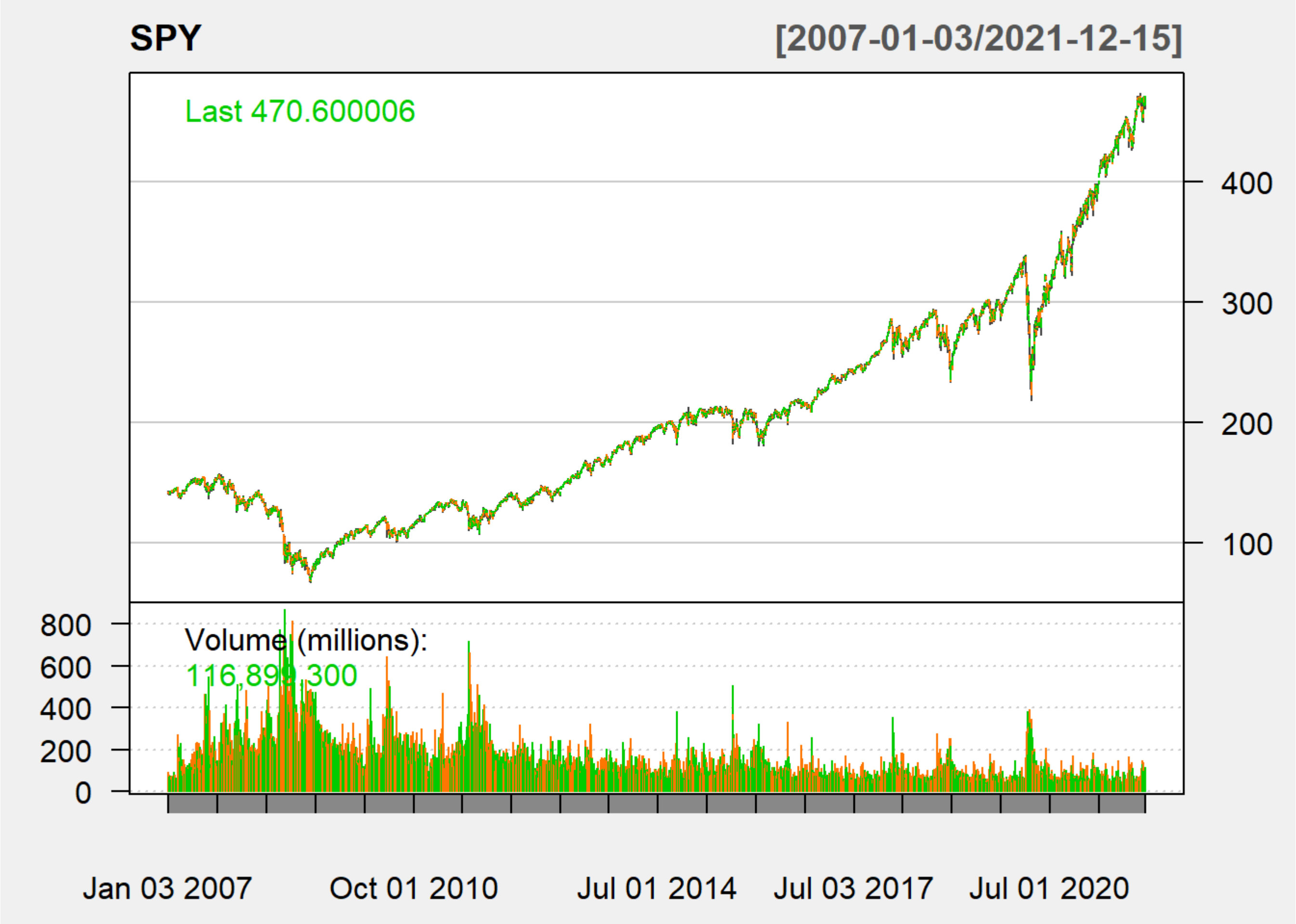
```
candleChart(GSPC,theme = chartTheme('white',up.col='yellow',dn.col='darkred'),
            TA=NULL,subset = '2016-01')
```



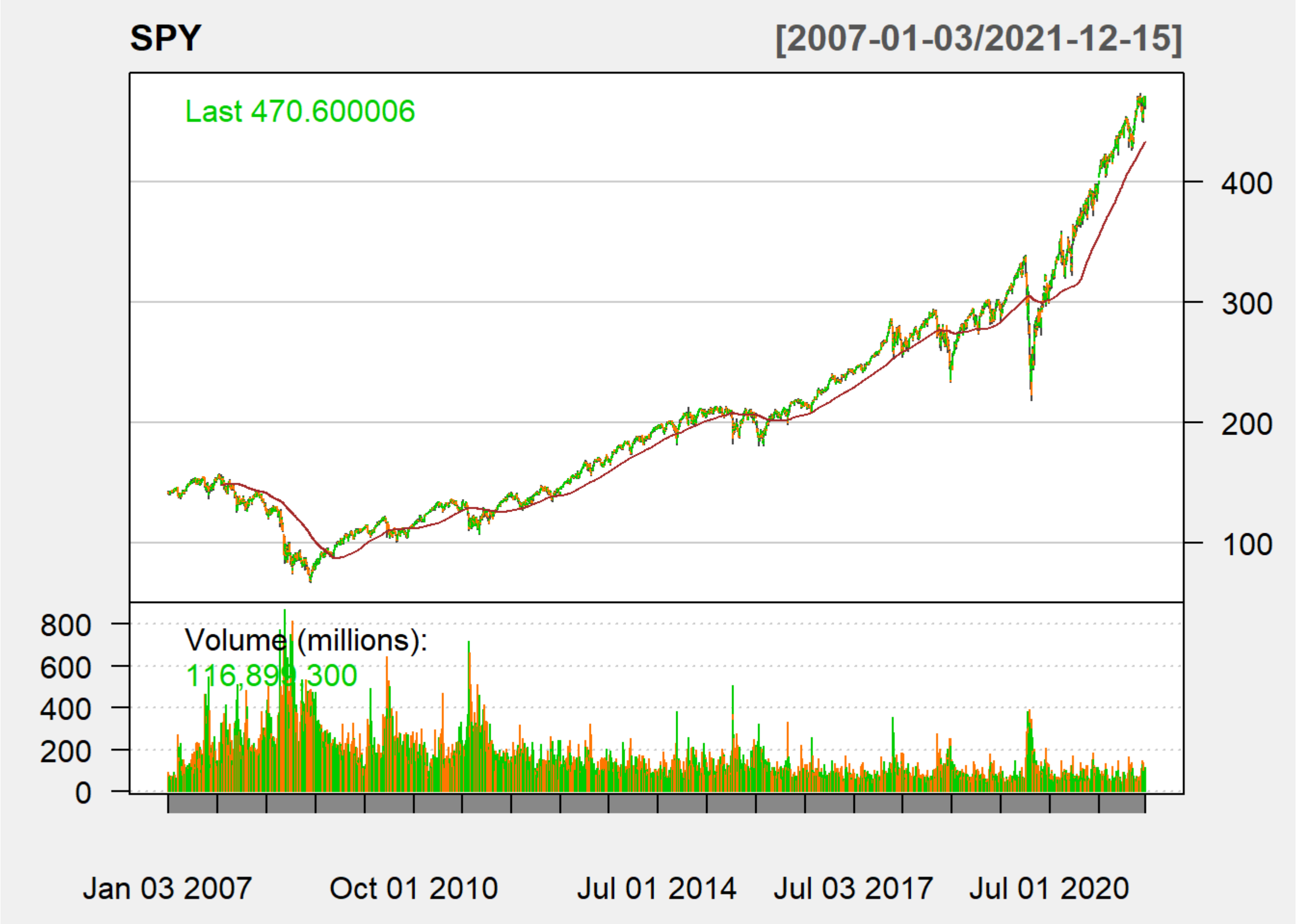
```
chartSeries(GSPC,type =c("candlesticks"),TA=NULL,subset = '2016-01')
```



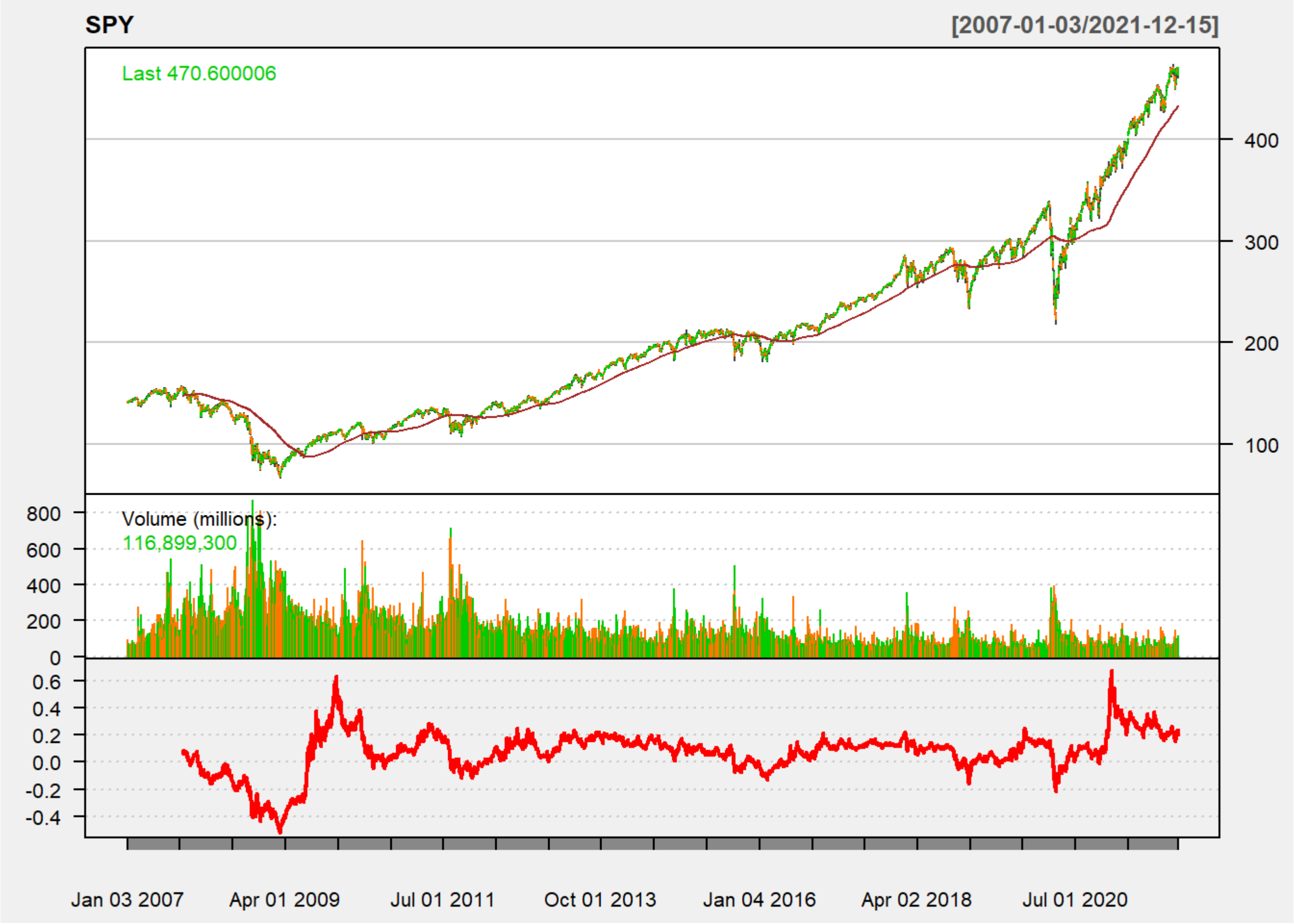
```
chartSeries(SPY, theme='white')
```



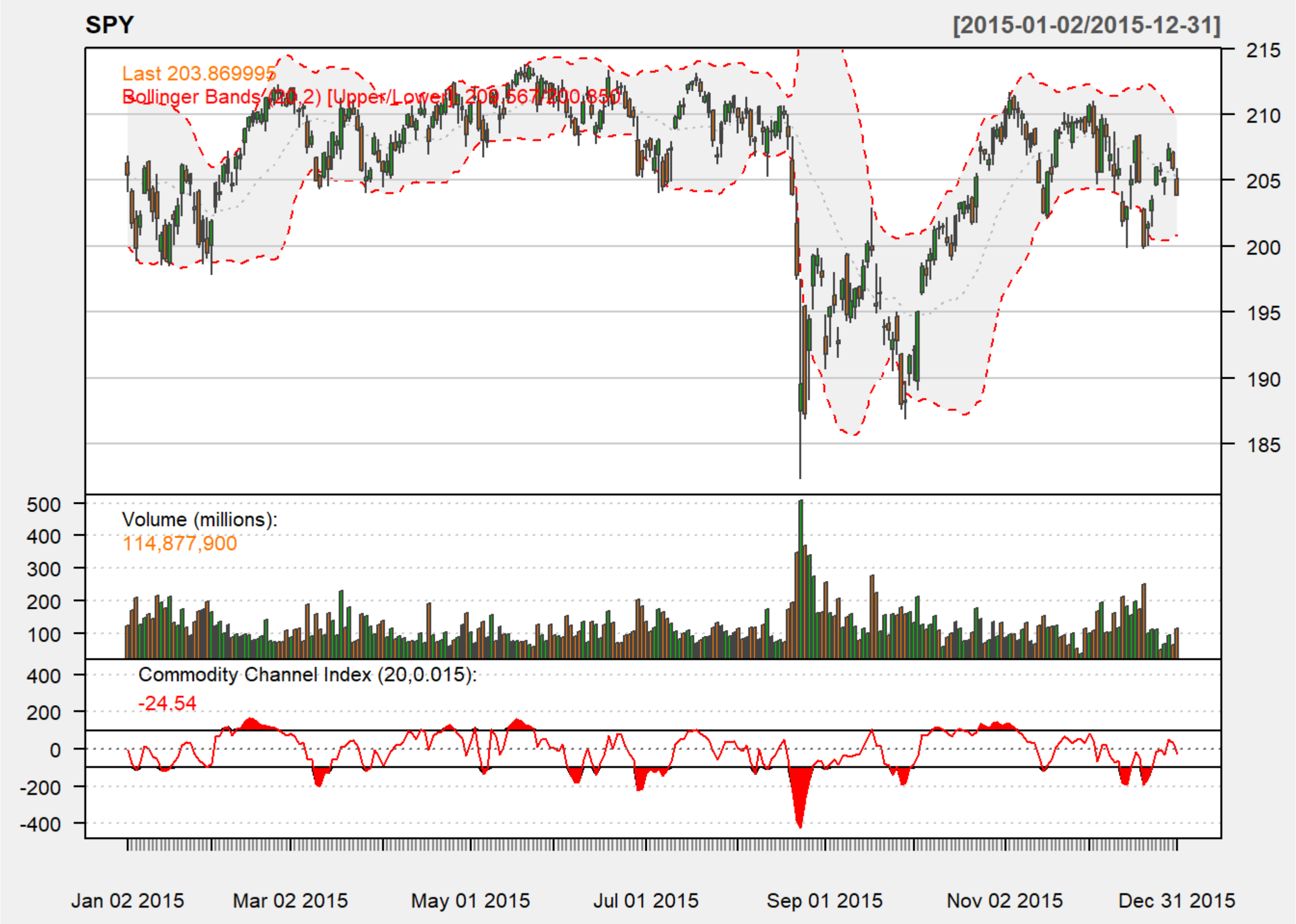
```
# Let's find the Symple moving avarage for period of 200
#{{\mathit  {momentum}} \over N+1}={\mathit  {SMA}}_{{\mathit  {today}}}-{\mathit  {SMA}}_{{\mat
hit  {yesterday}}}


addSMA(n=200)
```



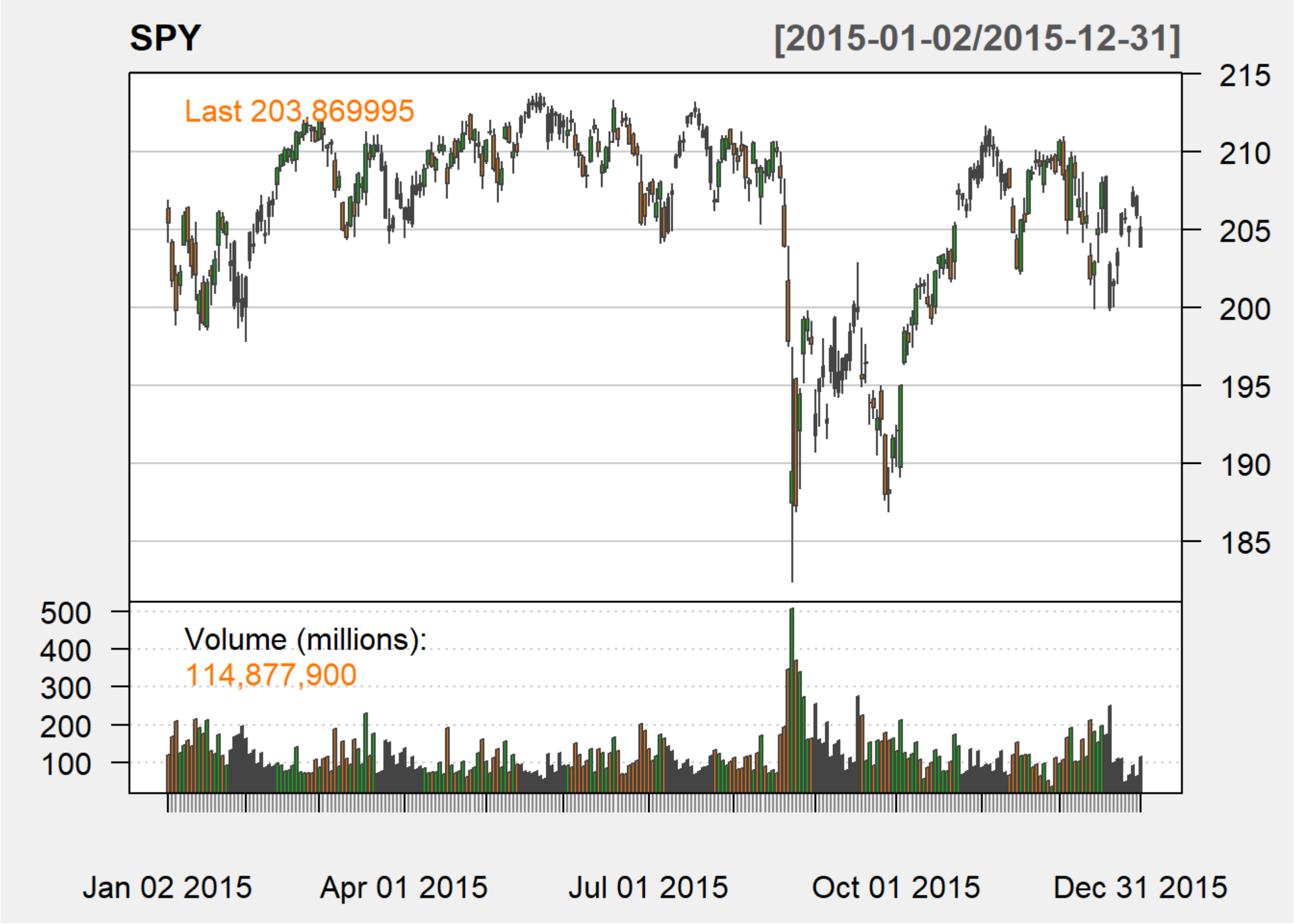
```
#Find the 10 period days of rate of change

addROC(n=200)
```



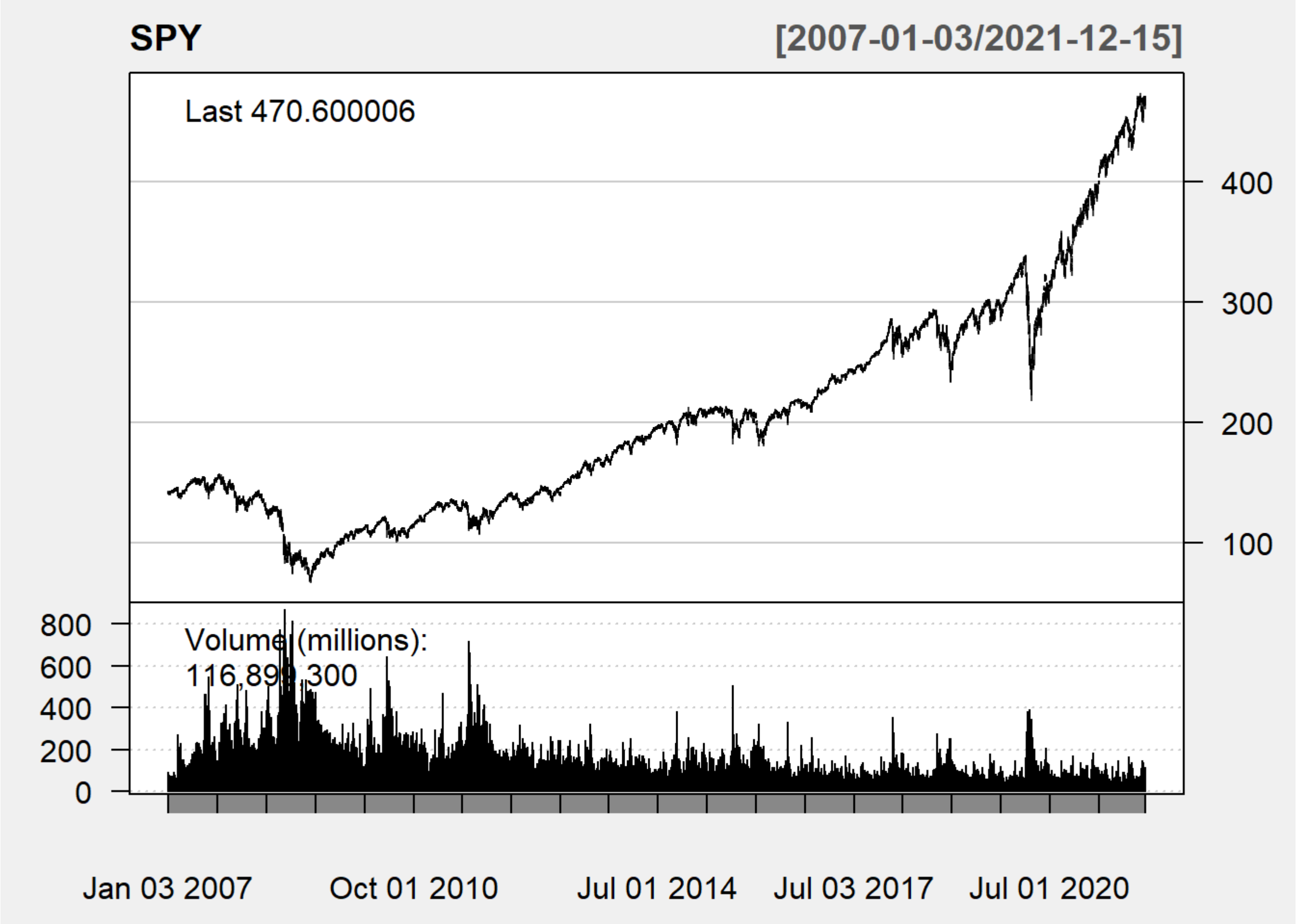
```
chartSeries(SPY, theme="white",
            TA="addVo();addBBands();addCCI()", subset='2015')
```



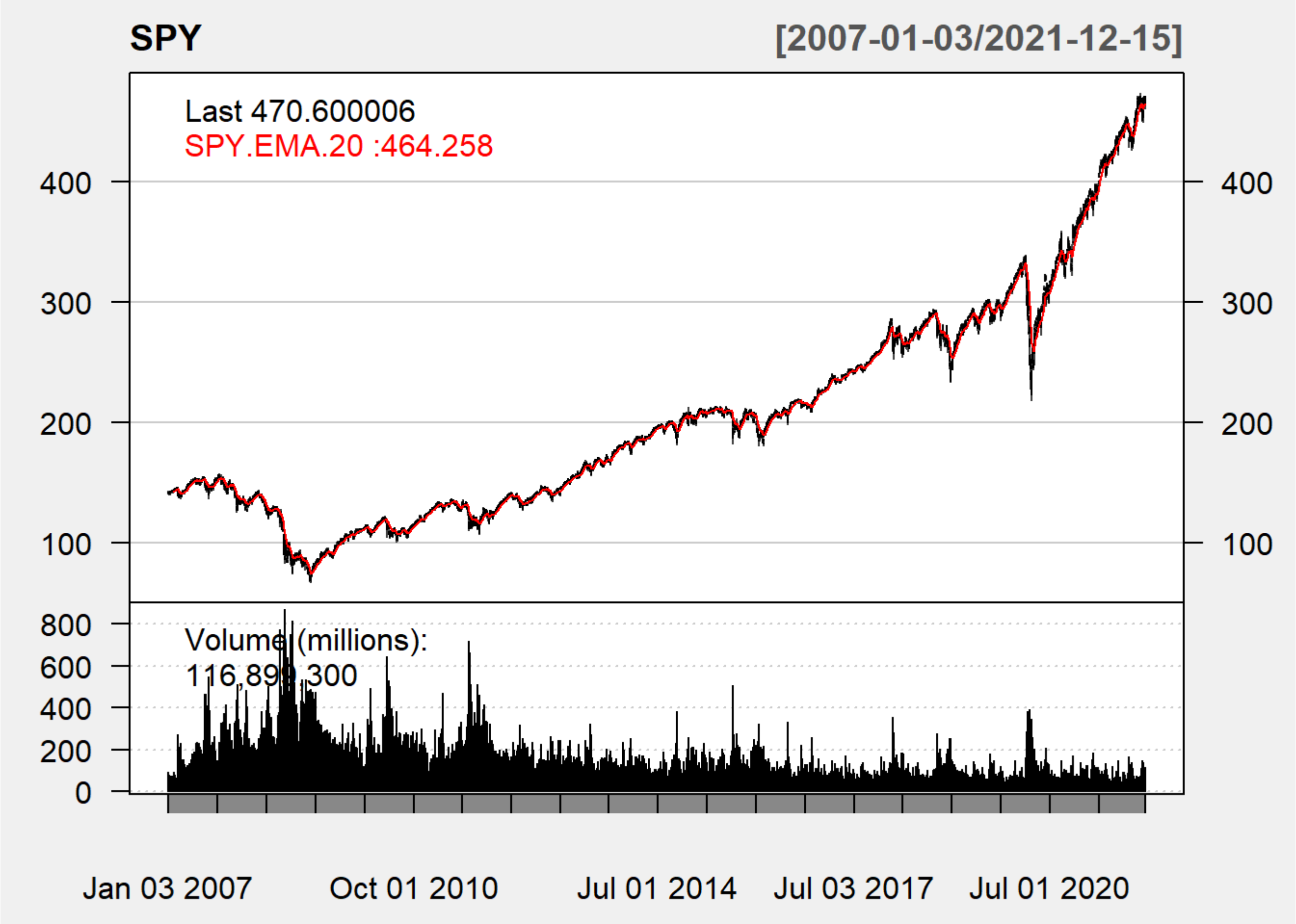
```
chartSeries(SPY, theme="white", subset='2015')
```



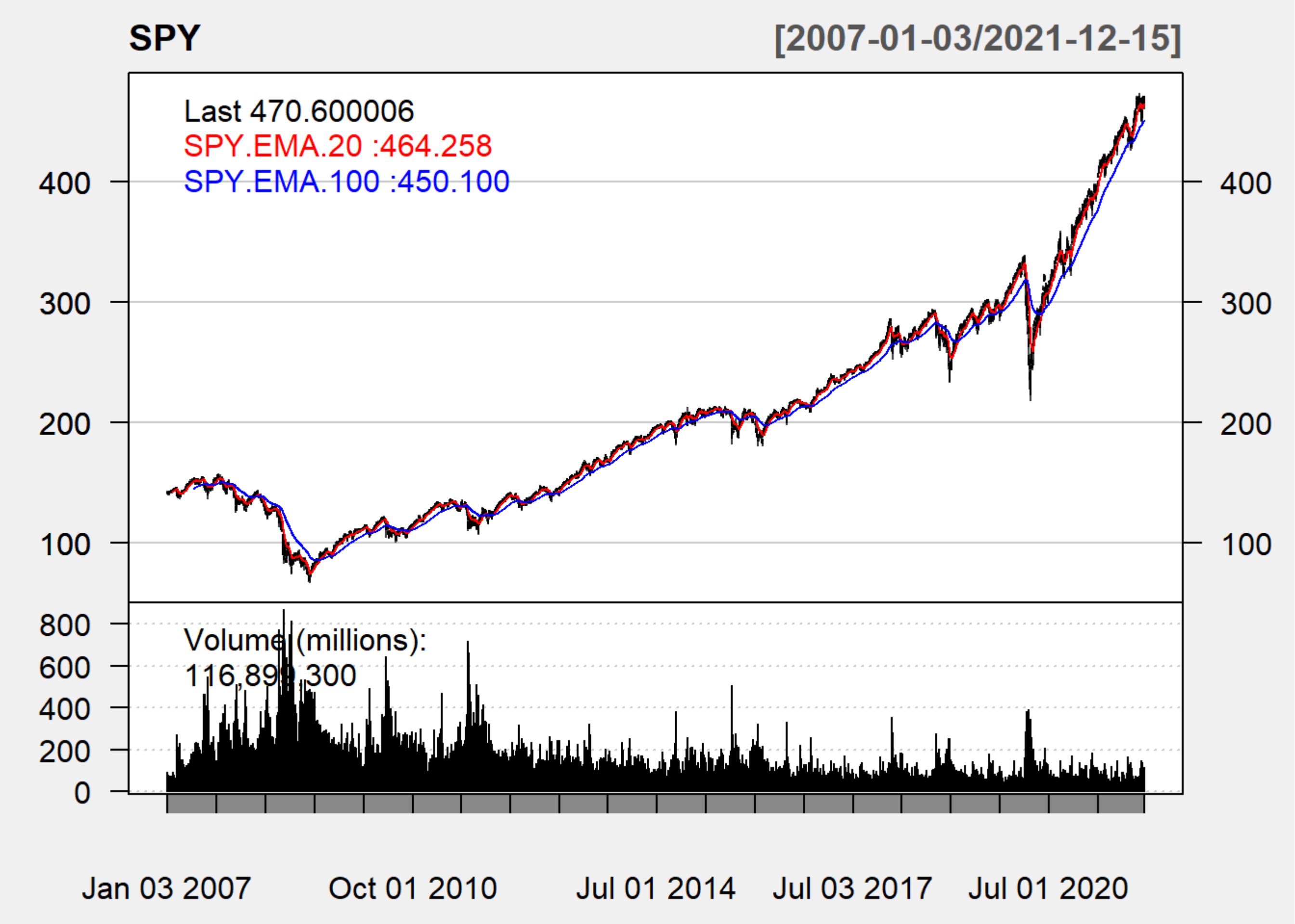
```
chartSeries(SPY, theme=chartTheme('white'), up.col="black",
            dn.col="black")
```



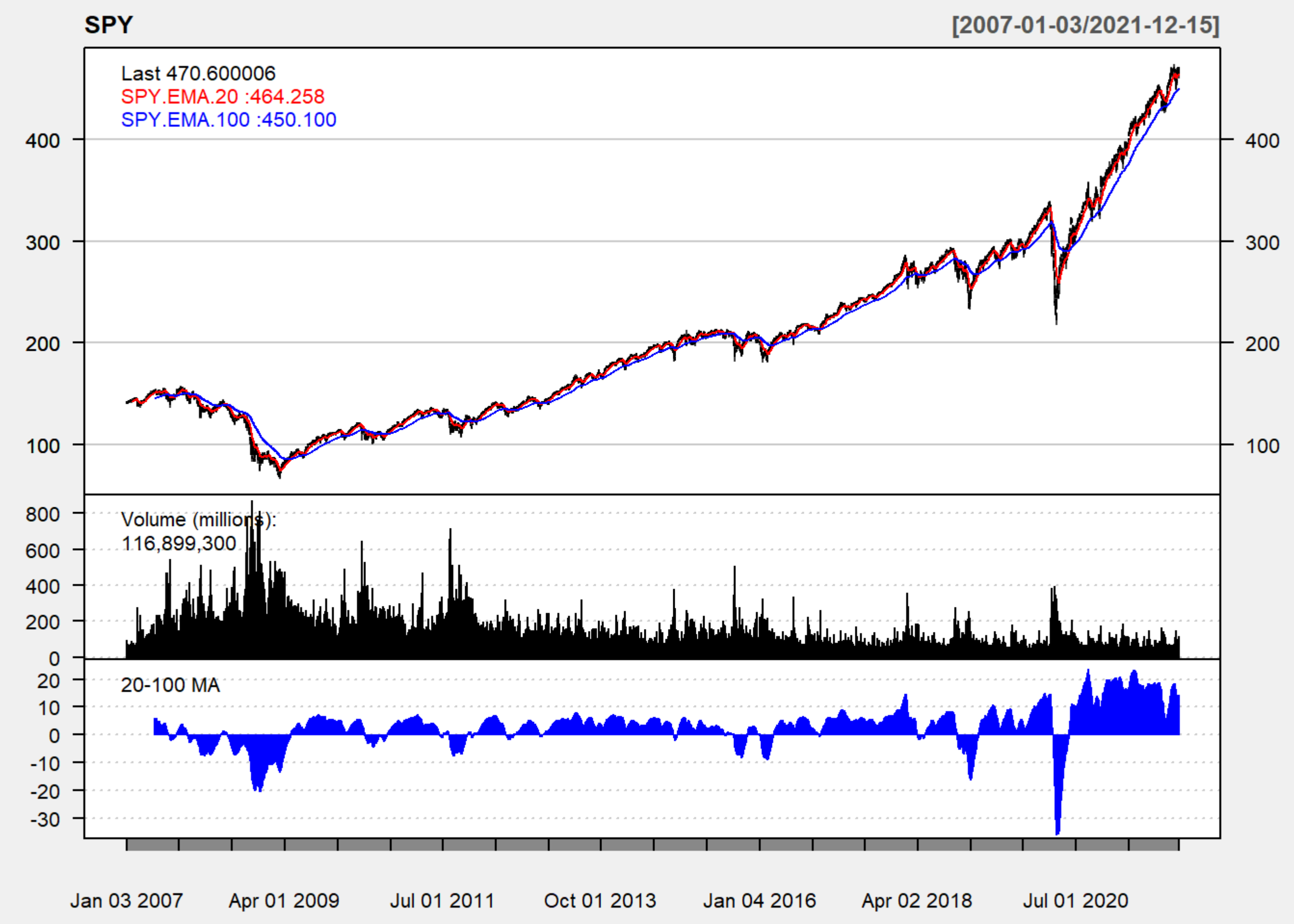
```
SPY.EMA.20<- EMA(SPY$SPY.Close, n=20)
SPY.EMA.100<- EMA(SPY$SPY.Close, n=100)
addTA(SPY.EMA.20, on=1, col = "red")
```



```
addTA(SPY.EMA.100, on=1, col = "blue")
```



```
addTA(SPY.EMA.20 - SPY.EMA.100,col='blue', type='h',legend="20-100 MA")
```



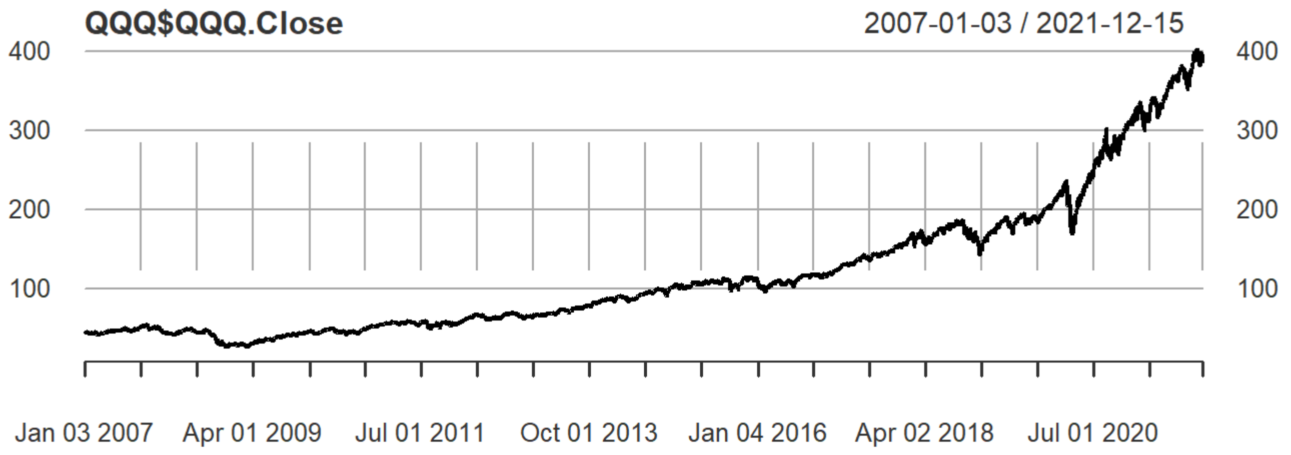
```
# get more inside about Moving Average price
# In the below lines I'm going to explain the SMA
# function that I have used above
print ("STEP 2.4:Creating Moving Average")
```

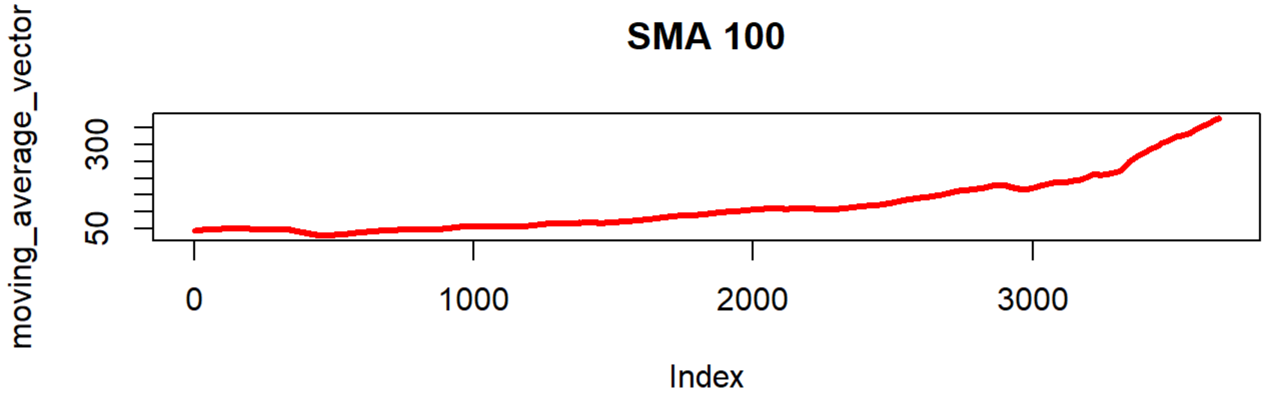
```
## [1] "STEP 2.4:Creating Moving Average"
```

```
getSymbols(c('QQQ'), src='yahoo')
```

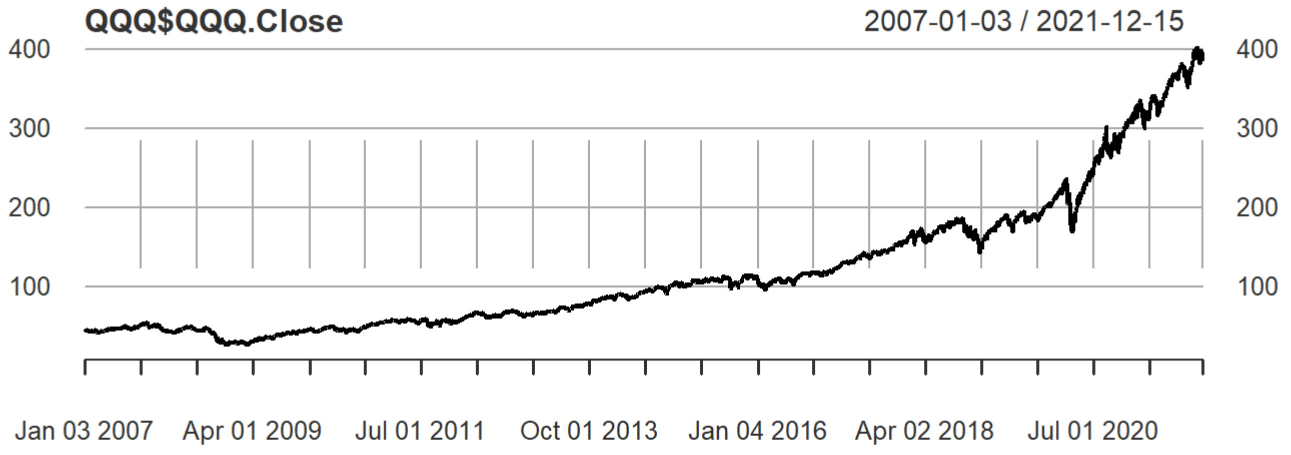
```
## [1] "QQQ"
```

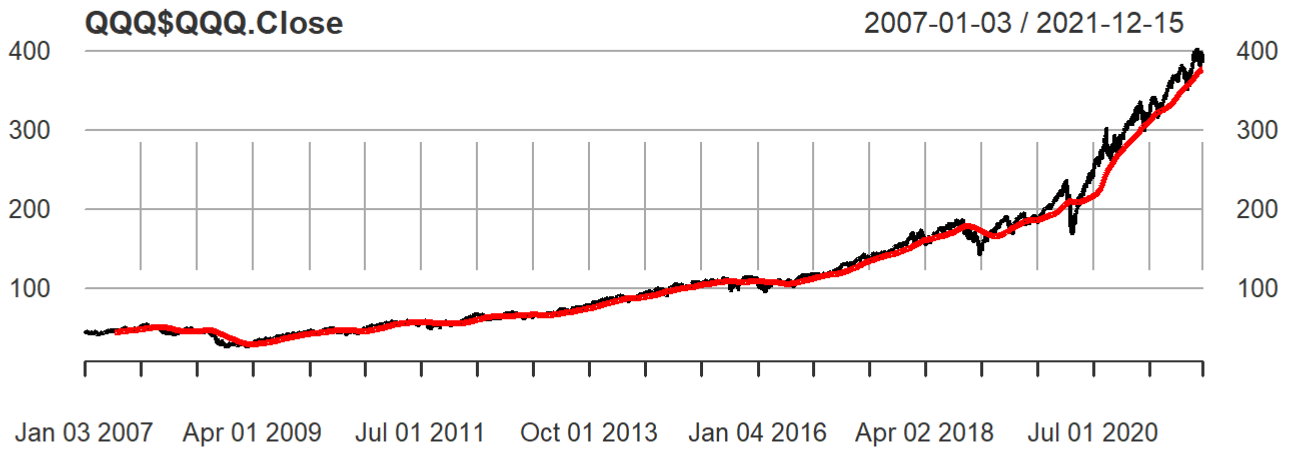
```
#I?ll focus on the Close of the bar (where it closed for the day). Let?s take a quick peek at wh
at we have:
plot(QQQ$QQQ.Close)

#I?ll create a simple function to break down the data and average every price point by x amount
 of points prior to it.
#In this case I?ll use a 100 day smoothing period.

period <- 100
price_vector <- QQQ$QQQ.Close
moving_average_vector <- c()
for (ind in seq((period+1),(length(price_vector))) ){
       moving_average_vector <- c(moving_average_vector, mean(price_vector[(ind-period):ind]))
}

par(mfrow=c(2,1))
plot(QQQ$QQQ.Close)
plot(moving_average_vector, type='l', col='red', lwd=3, main = paste('SMA', period))
```



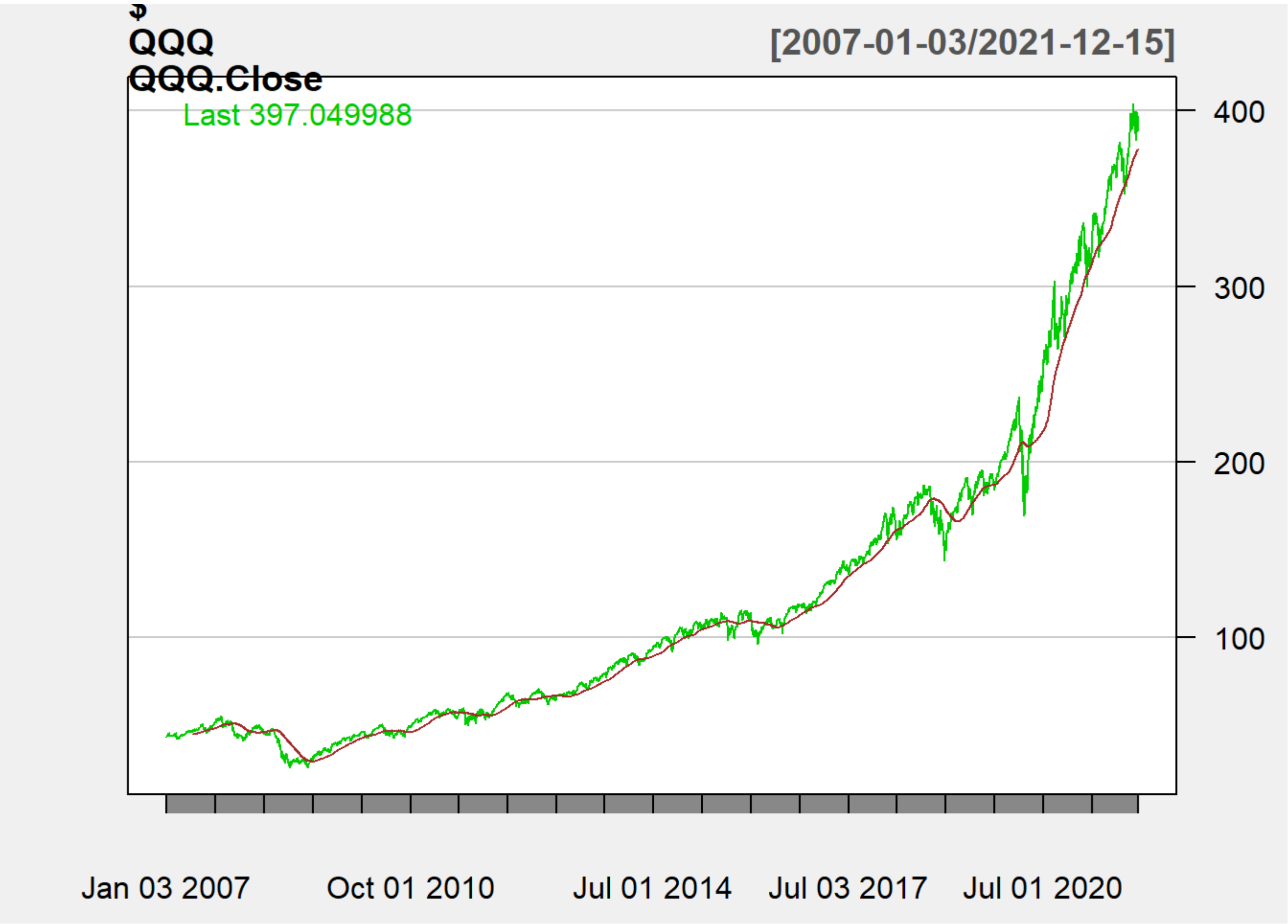


```
#The first plot is the raw QQQ daily closing prices and the second plot, is our smoothed versio
n. Keep in mind that the first 100 days of price data
#can?t be used as that is the minimum data we need to create a 100 period average.
#The issue we have is our new SMA vector contains 2065 entries, while our the QQQ market downloa
d, has 2165 entries.
#This should be easy to understand as it takes 100 entries to calculate an SMA.
#This is going to make it difficult to overlay our SMA onto the raw market data.
#One way around this is to buffer our SMA with 100 NA?s.

period <- 100
price_vector <- QQQ$QQQ.Close
moving_average_vector <- c(rep(NA, period))
# moving_average_vector <- c(rep(as.numeric(QQQ$QQQ.Close[period]), period))
for (ind in seq((period+1),(length(price_vector))) ){
       moving_average_vector <- c(moving_average_vector, mean(price_vector[(ind-period):ind]))
}

# pass it back to our time series object
QQQ$QQQ.Close.SMA <- moving_average_vector

plot(QQQ$QQQ.Close)
lines(QQQ$QQQ.Close.SMA, type='l', col='red', lwd=3)
```





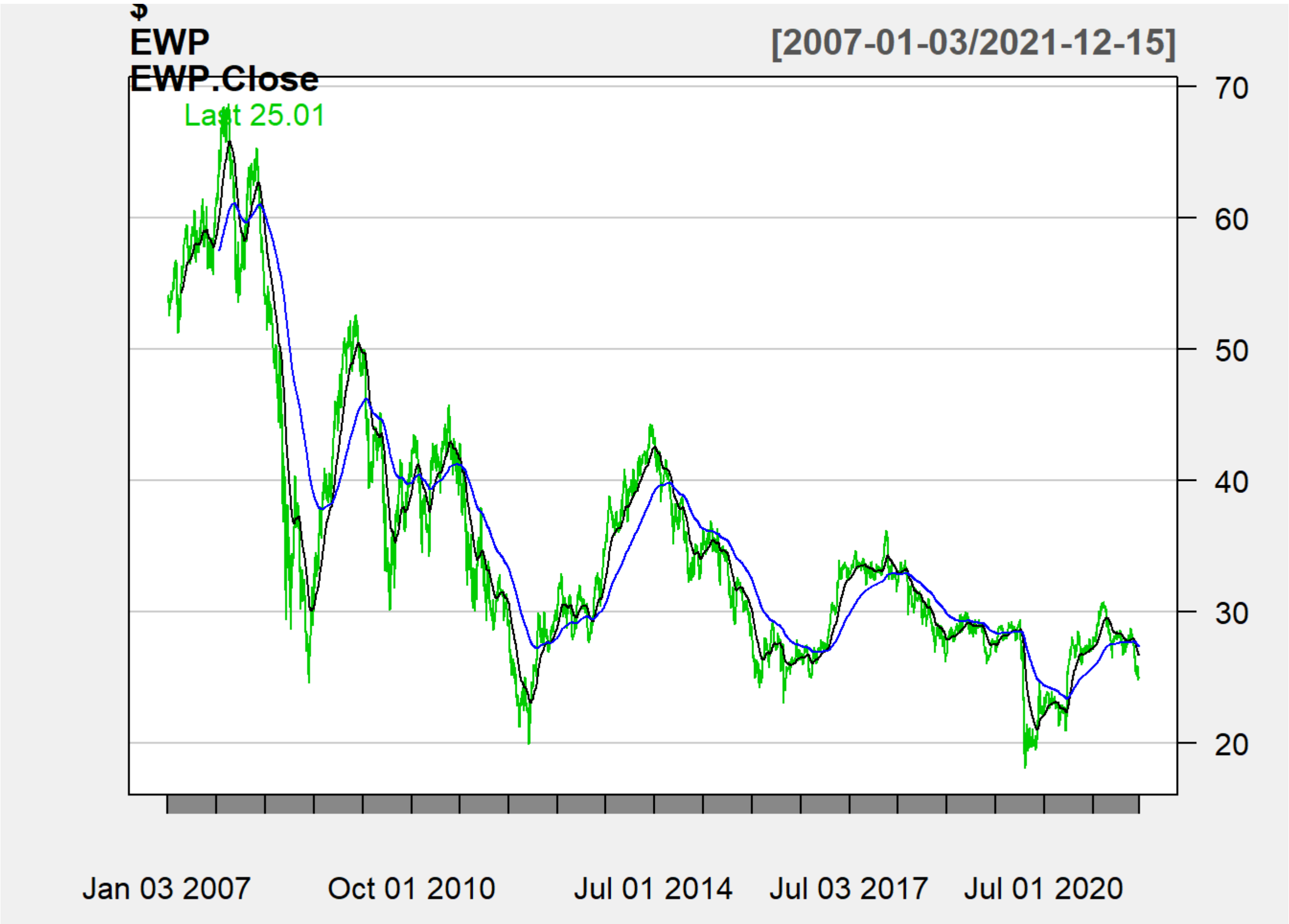
```
# All above action could be simplified by using TTA package same as below:
chartSeries(QQQ$QQQ.Close, theme="white", TA="addSMA(100)")
```



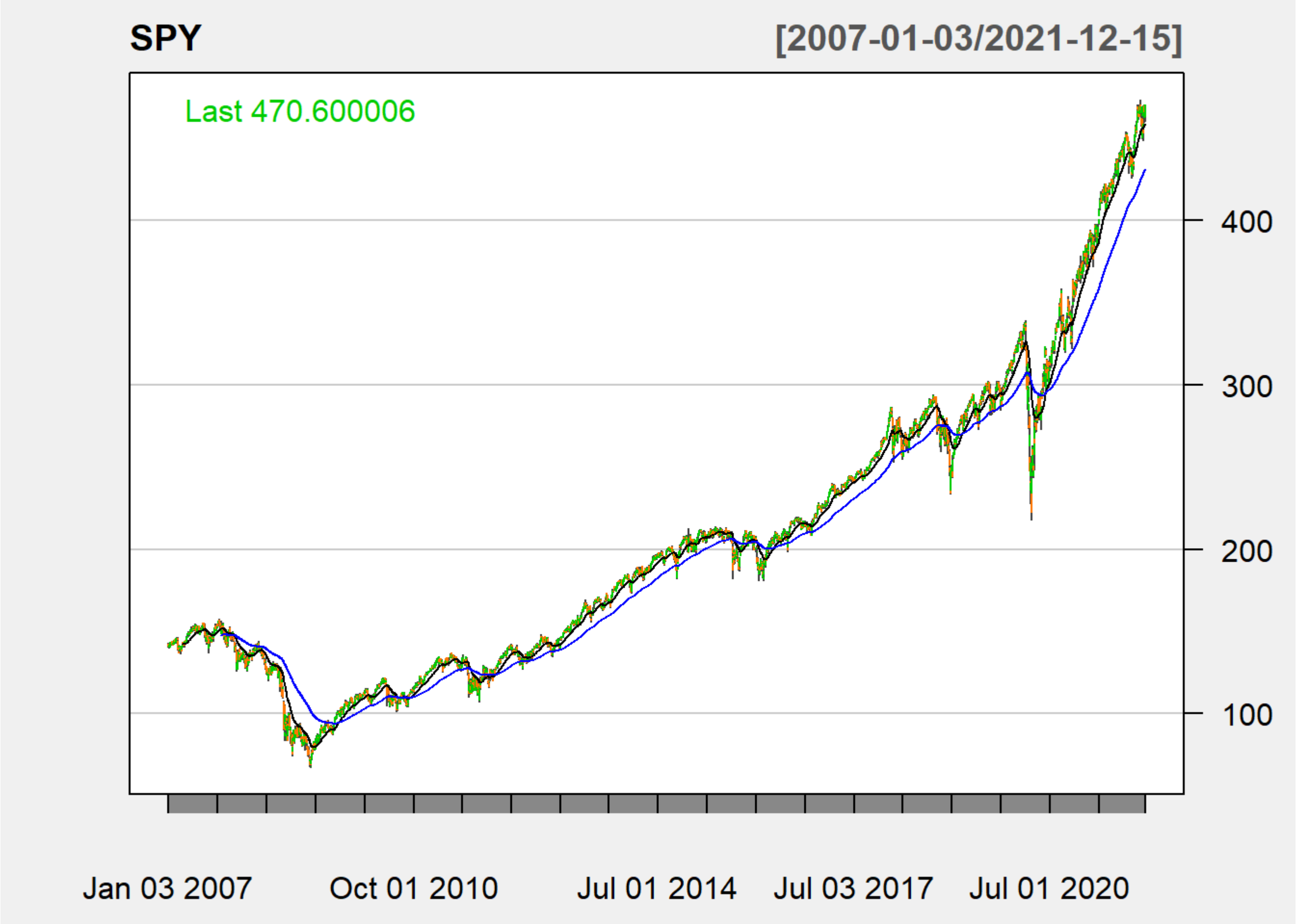
```
# Following the trend with multiple moving avarge
# Looking at multiple moving averages, the 10, 50 & 200 MAs * Detrending market action
getSymbols(c('EWP', 'SPY'), src='yahoo')
```

```
## [1] "EWP" "SPY"
```

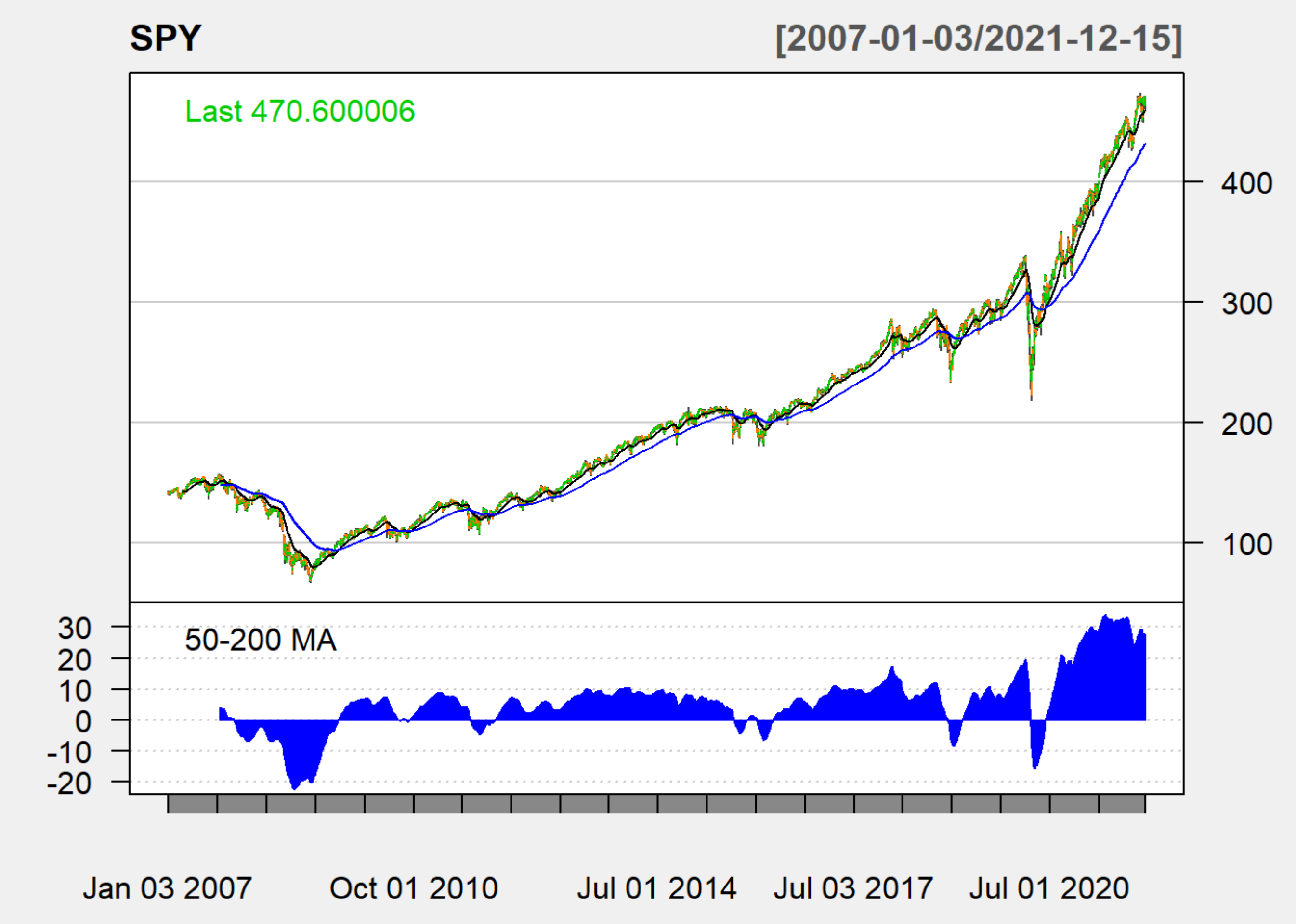
```
#Let?s chart the data using a 50 and 200-period moving average.
#These are common periods often used as benchmarks to indicate a strengthening or weakening stoc
k.
chartSeries(EWP$EWP.Close, theme="white", TA="addEMA(50, col='black');addEMA(200, col='blue')")
```



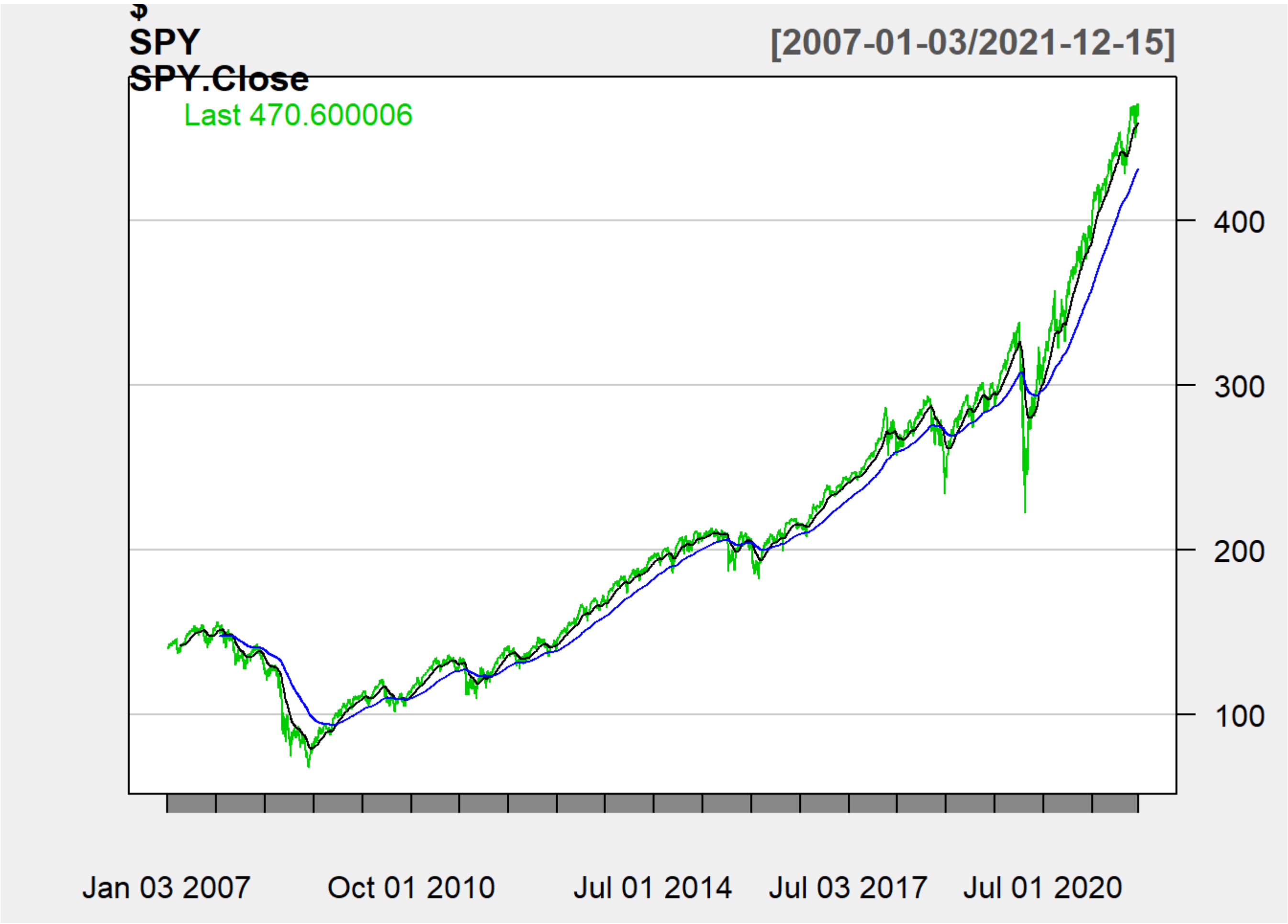
```
chartSeries(SPY, theme="white", TA="addEMA(50, col='black');addEMA(200, col='blue')")
```



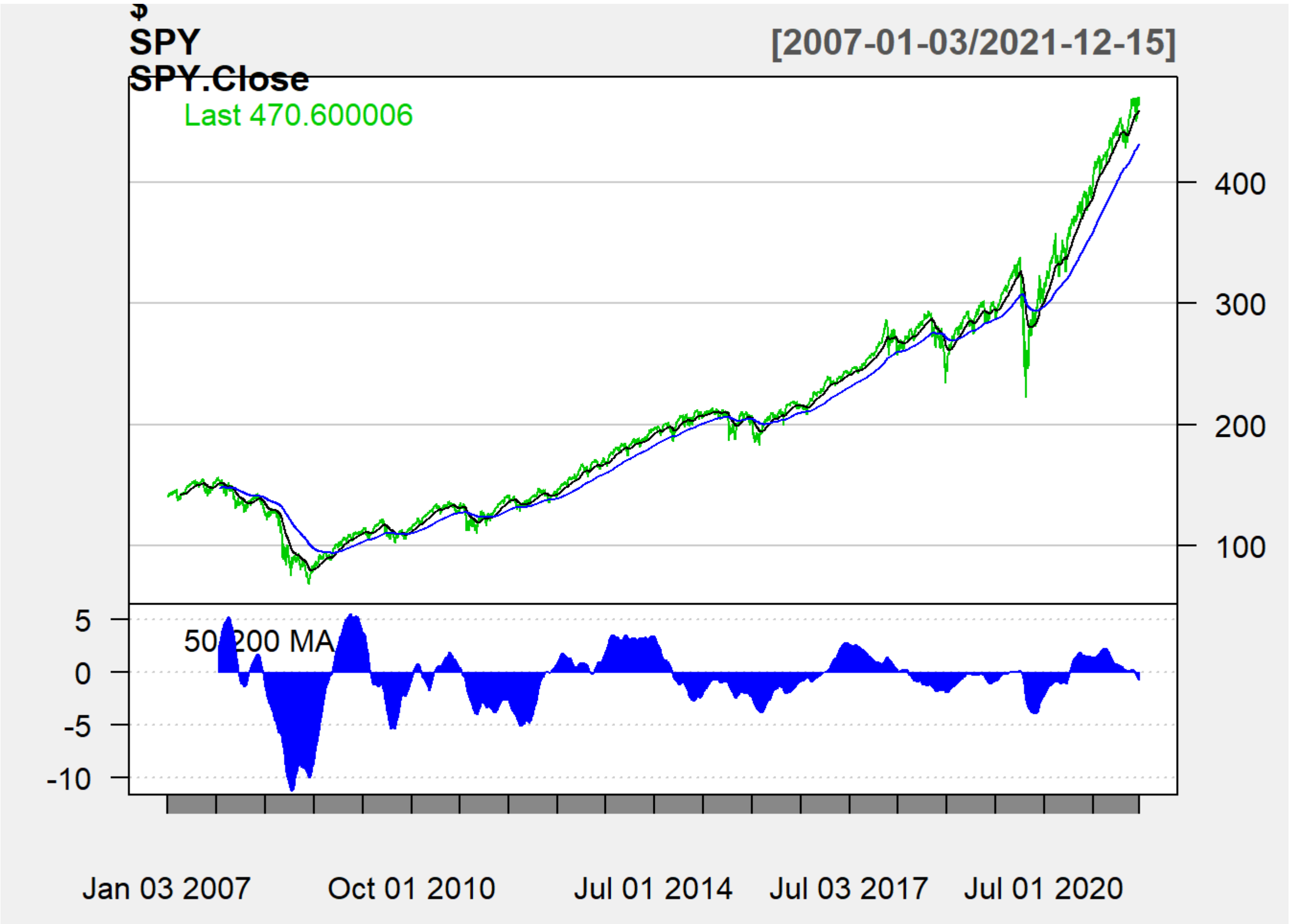
```
#Having two moving averages of different periods removes a lot of the noise.
#When the fast moving average is above the slow one, the market is moving upwards,
#and when the fast is below the slow, it is going down. Some traders will look at the
#crossing of these moving averages to take a directional position


SPY.EMA.50<- EMA(SPY$SPY.Close, n=50, )
SPY.EMA.200<- EMA(SPY$SPY.Close, n=200, )
#SPY.EMA.50 fast change
#SPY.EMA.200 slow change
addTA(SPY.EMA.50 - SPY.EMA.200,col='blue', type='h',legend="50-200 MA")
```



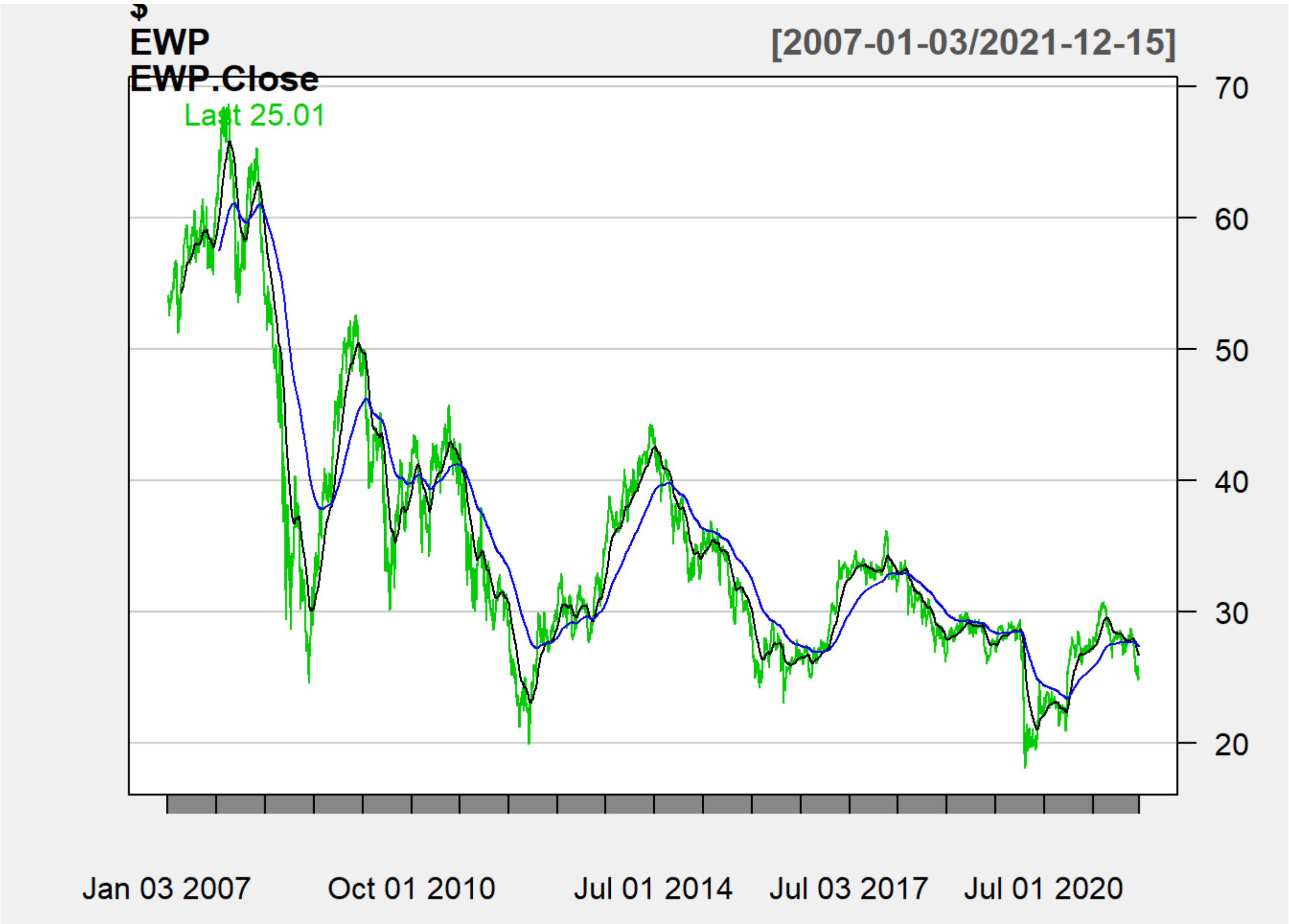
```
chartSeries(SPY$SPY.Close, theme="white", TA="addEMA(50, col='black');addEMA(200, col='blue')")
```



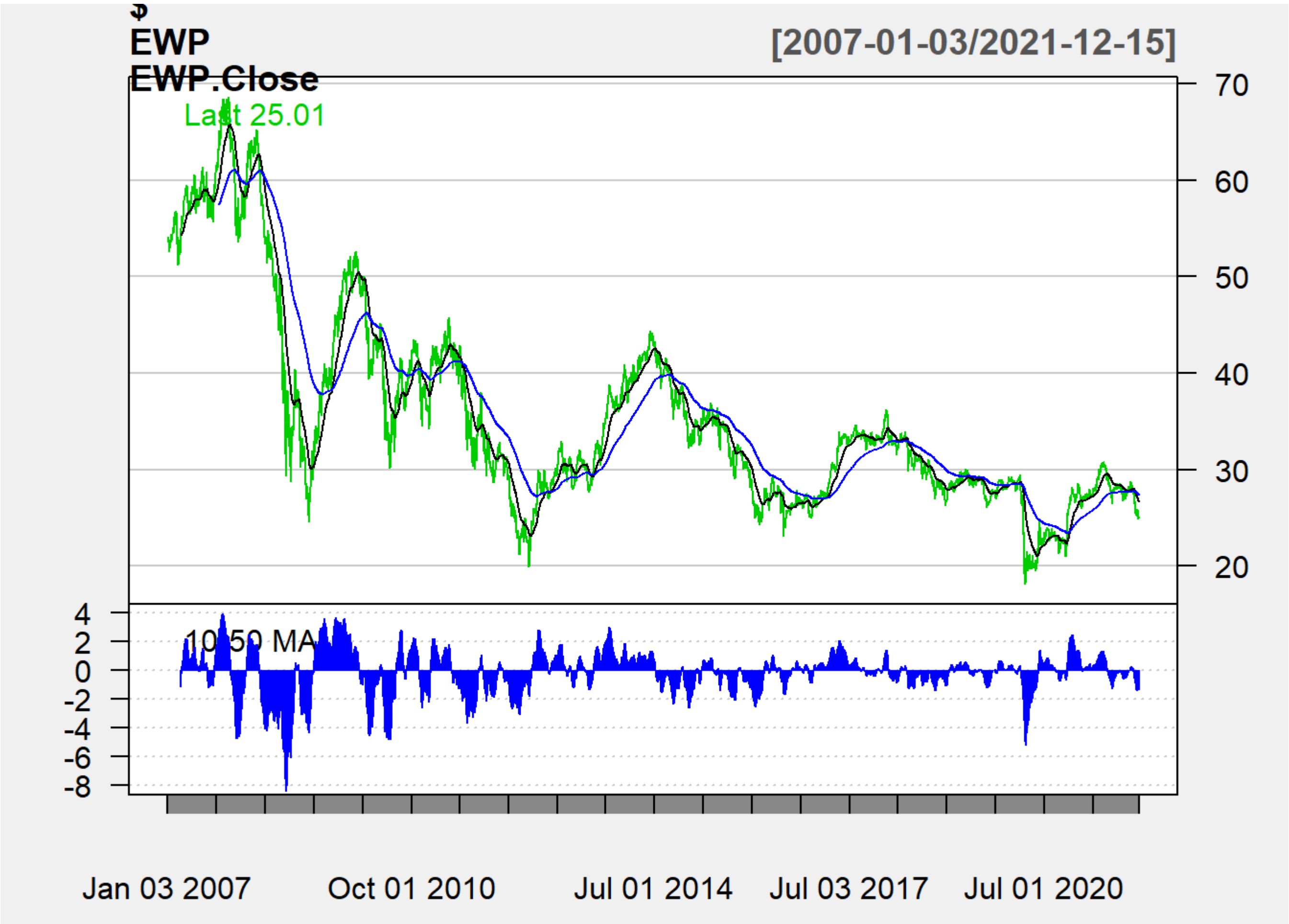
```
EWP.EMA.50 <- EMA(EWP$EWP.Close, n=50, )
EWP.EMA.200 <- EMA(EWP$EWP.Close, n=200, )
addTA(EWP.EMA.50 - EWP.EMA.200, col='blue', type='h',legend="50-200 MA")
```



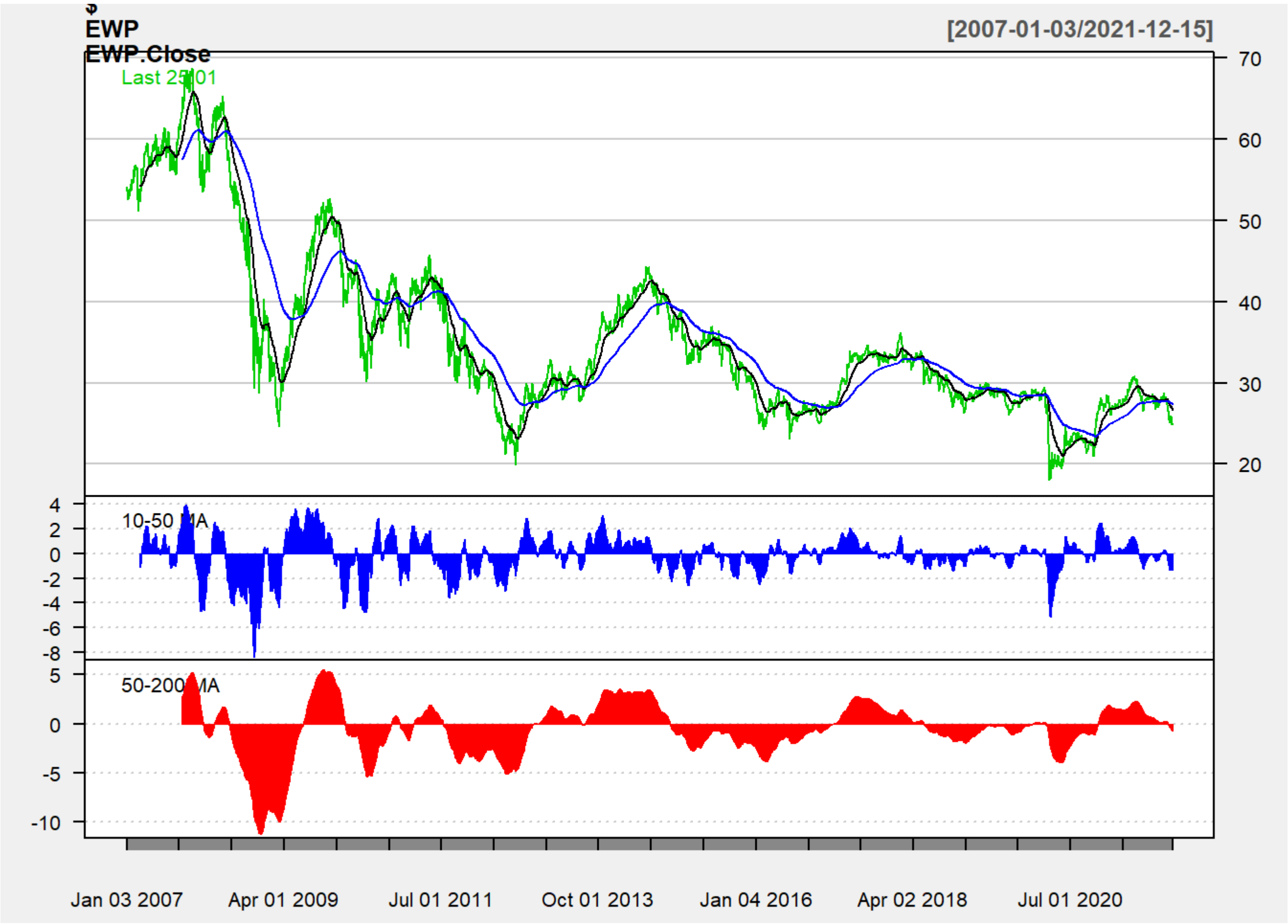
```
chartSeries(EWP$EWP.Close, theme="white", TA="addEMA(50, col='black');addEMA(200, col='blue')")
```



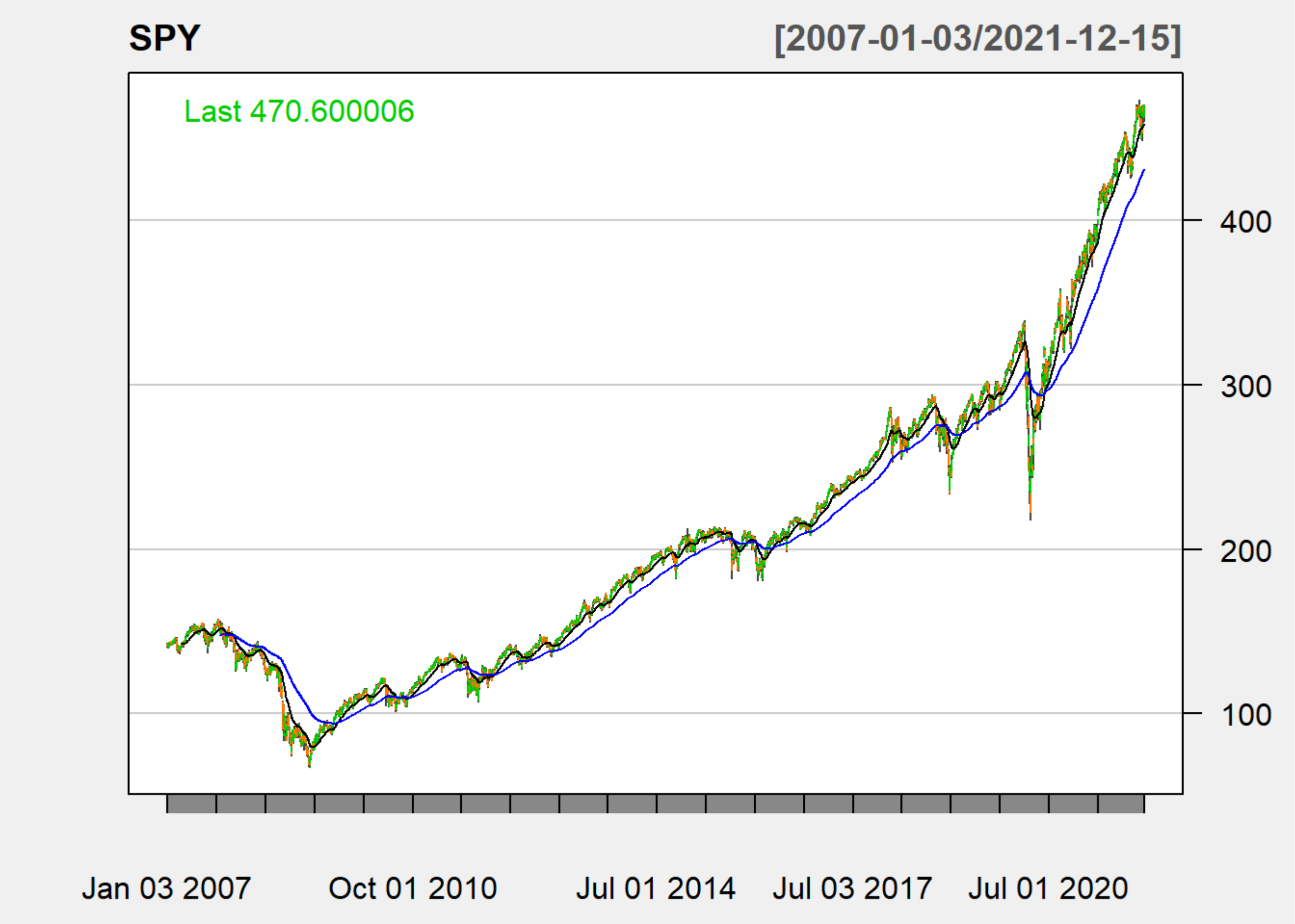
```
# everyting below Zero - You should not be long - and keep the Index , Holding
# everything above Zero - You should not be short - and sell the Index , Holding
# Let's look into three avarage moving , I'm adding 10 period

EWP.EMA.10 <- EMA(EWP$EWP.Close, n=10, )
EWP.EMA.50 <- EMA(EWP$EWP.Close, n=50, )
EWP.EMA.200 <- EMA(EWP$EWP.Close, n=200, )
Fast.Diff <- EWP.EMA.10 - EWP.EMA.50
Slow.Diff <- EWP.EMA.50 - EWP.EMA.200
addTA(Fast.Diff, col='blue', type='h',legend="10-50 MA")
```



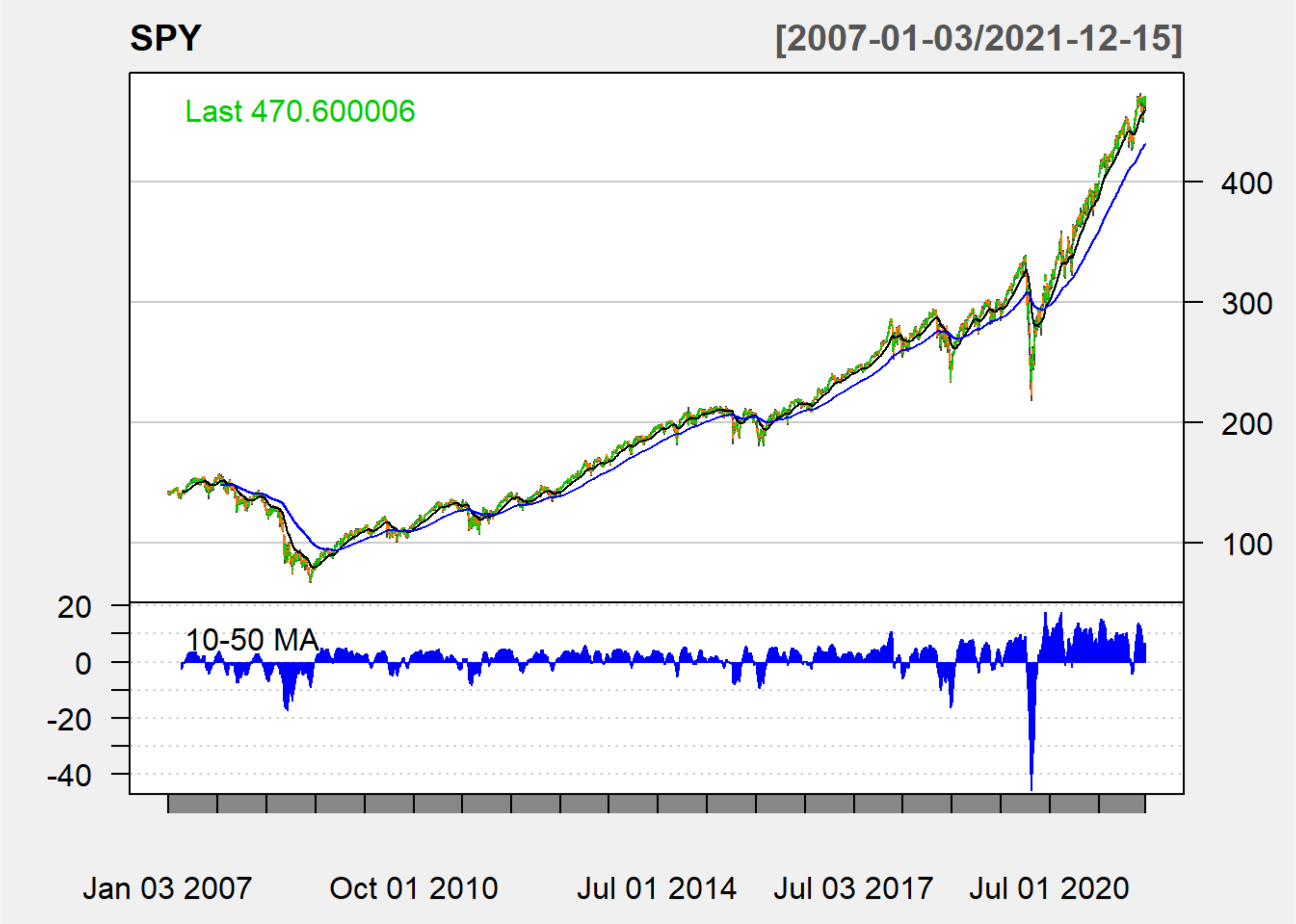
```
addTA(Slow.Diff, col='red', type='h',legend="50-200 MA")
```



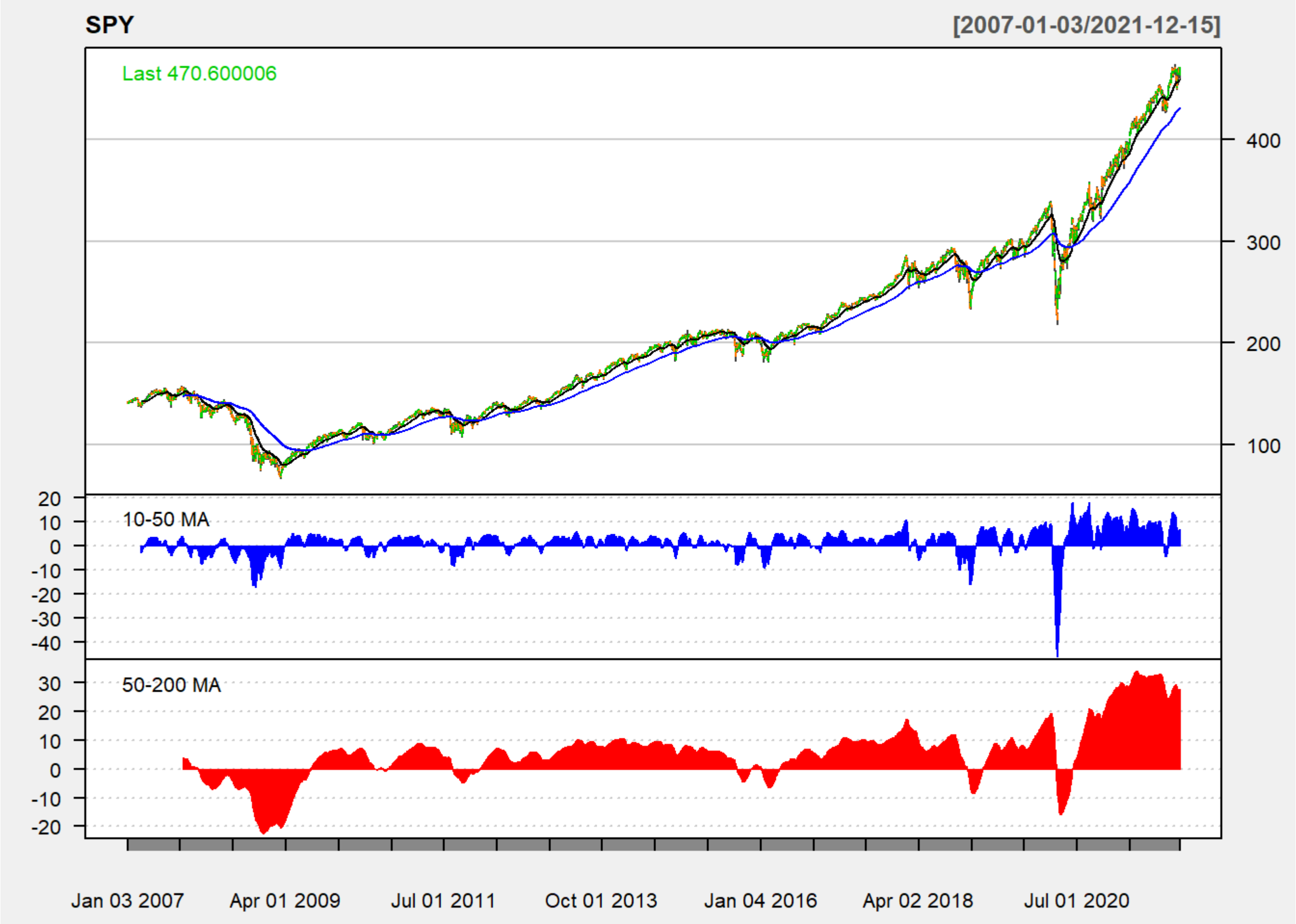
```
chartSeries(SPY, theme="white", TA="addEMA(50, col='black');addEMA(200, col='blue')")
```



```
SPY.EMA.10 <- EMA(SPY$SPY.Close, n=10, )
SPY.EMA.50 <- EMA(SPY$SPY.Close, n=50, )
SPY.EMA.200 <- EMA(SPY$SPY.Close, n=200, )
Fast.Diff <- SPY.EMA.10 - SPY.EMA.50
Slow.Diff <- SPY.EMA.50 - SPY.EMA.200
addTA(Fast.Diff, col='blue', type='h',legend="10-50 MA")
```



```
addTA(Slow.Diff, col='red', type='h',legend="50-200 MA")
```



```
#Trading With The Trend

#You can only enter in the direction of the red Slow.Diff indicator,
#if its above zero you can take long signals, if its below zero,
#you can take short signals. The Fast.Diff indicator dictates the entries.
#When the blue line goes from negative to positive, its a long trade (and the slower red Slow.Diff indicator is above zero).
#Same thing for shorts. This is also referred to as a moving average crossover trading system.

#To run this system, we need to build rules to hunt them down.

#The rules are:

#    if no position: red > 0 and blue-1 < 0 and blue > 0 go long
#    if long: blue < 0 exit long

#    if no position: red < 0 and blue-1 > 0 and blue < 0 go short
#    if short: blue > 0 exit short
# New chalange would to find the blue -1 means, meaning lag of blue, Pre. price .
print ("STEP 2.5:Trading With The Trend")
```

```
## [1] "STEP 2.5:Trading With The Trend"
```

```
library(binhf)
```

```
## Loading required package: wavethresh
```

```
## Loading required package: MASS
```

```
## WaveThresh: R wavelet software, release 4.6.8, installed
```

```
## Copyright Guy Nason and others 1993-2016
```

```
## Note: nlevels has been renamed to nlevelsWT
```

```
## Loading required package: adlift
```

```
## Loading required package: EbayesThresh
```

```
## 
##  **********************************************
##  adlift: a package to perform wavelet lifting schemes
## 
##  --- Written by Matt Nunes and Marina Knight ---
##    Current package version:  1.4-1  ( 2018-07-09 )
## 
##              -+ packaged by MAN +-
##  **********************************************
## 
##  adlift 1.4-1 loaded
```

```
## 
## Attaching package: 'adlift'
```

```
## The following object is masked from 'package:EbayesThresh':
## 
##     postmean.cauchy
```

```
##
##  **********************************************
##  binhf: Haar-Fisz functions for binomial data
##
##  --- Written by Matt Nunes ---
##    Current package version:  1.0-3  ( 2018-07-18 )
##
##
##  **********************************************
##
##  binhf 1.0-3 loaded
```

```
##
## Attaching package: 'binhf'
```

```
## The following object is masked from 'package:EbayesThresh':
##
##     negloglik.laplace
```

```
## The following object is masked from 'package:base':
##
##     norm
```

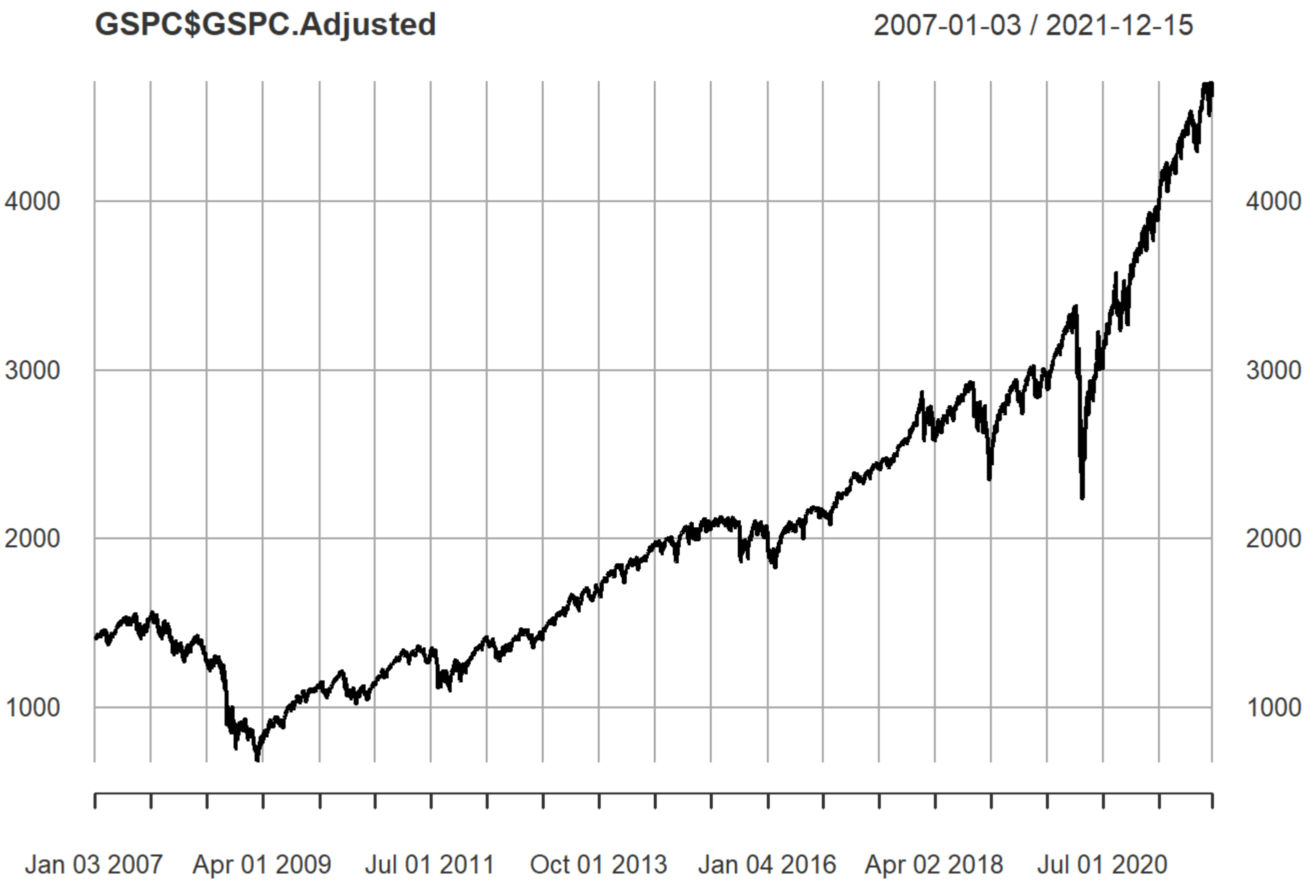
```
tail(as.numeric(Fast.Diff))
```

```
## [1] 5.364051 5.700954 6.551166 6.578154 6.094709 6.703664
```

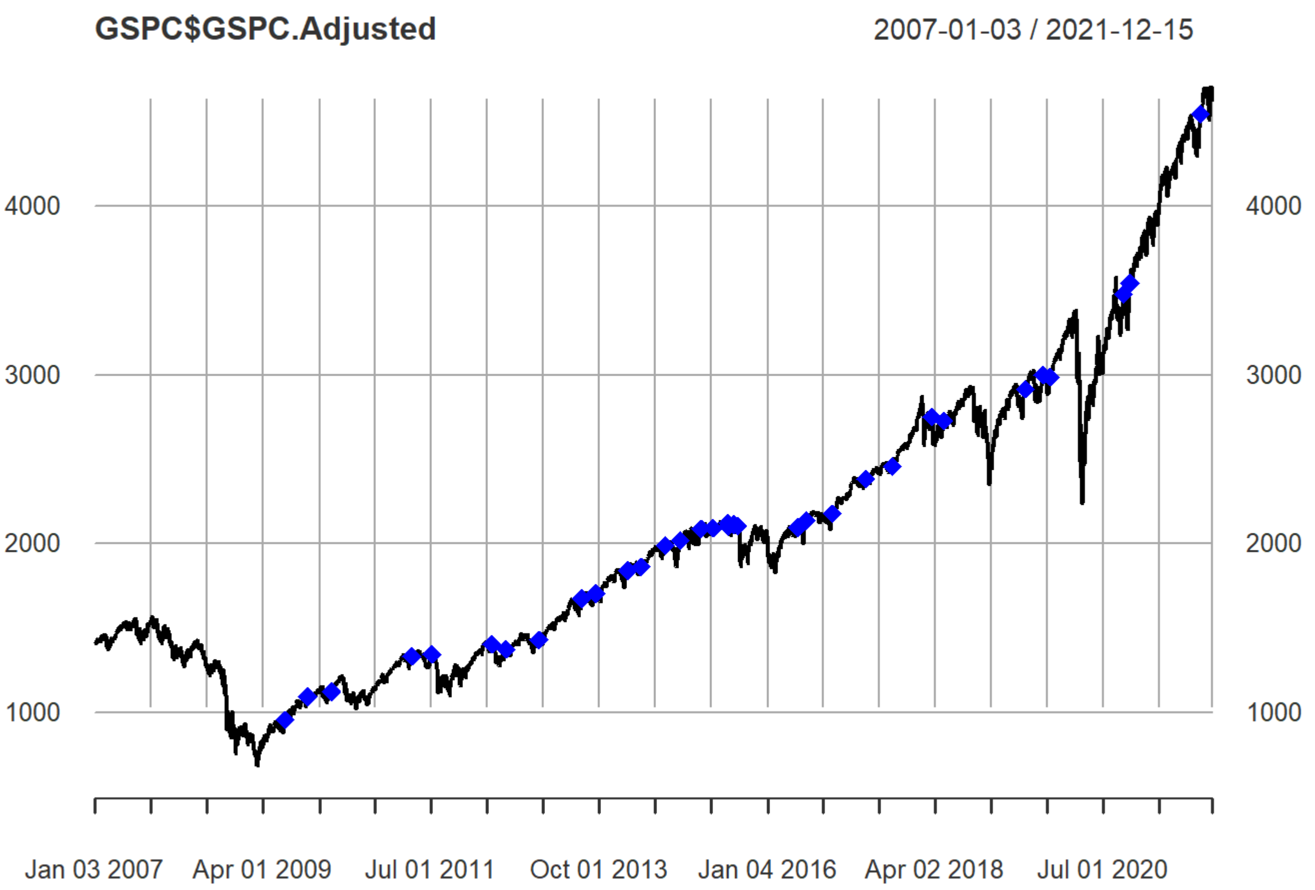
```
# return prev. data
tail(shift(v=as.numeric(Fast.Diff), places=1, dir="right"))
```

```
## [1] 4.311767 5.364051 5.700954 6.551166 6.578154 6.094709
```

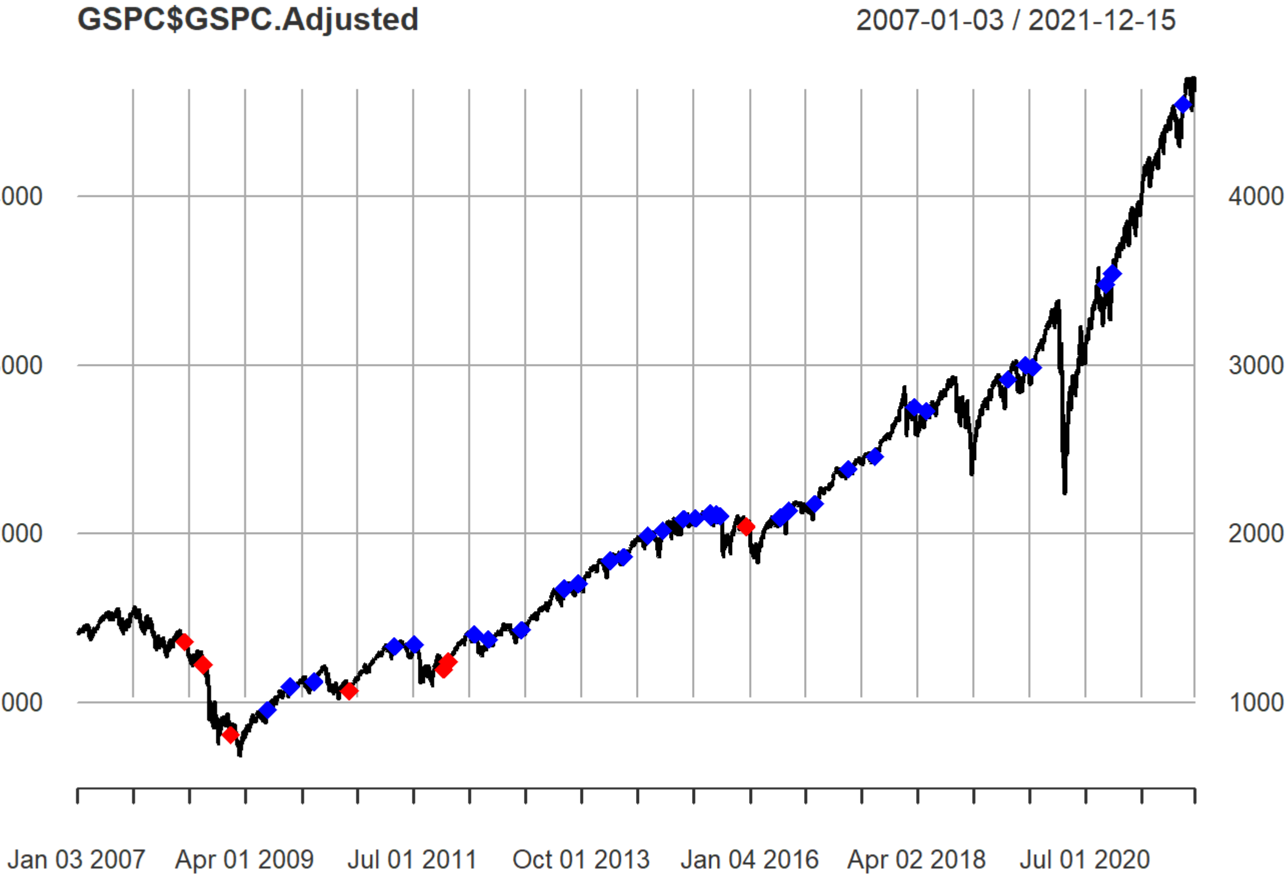
```
#This allows us to compare the values of two different rows on the same row.
#We still have our indicator value of today, but we now can compare it with yesterday?s value on
the same row.
#Sure, we could have just easily created a loop and run through each value but by doing it this
 way we stick to vector comparison in its simplest form.

#Now, let?s translate our trend trading system pseudo code into R code:
#Note: Closing price won't give us best price since compay pays dividend / interest and this pri
ce is not accure ah the end of the
# month, Hence I have used Adjusted price.

GSPC.SMA.10 <- SMA(GSPC$GSPC.Adjusted, n=10, )
GSPC.SMA.50 <- SMA(GSPC$GSPC.Adjusted, n=50, )
GSPC.SMA.200 <- SMA(GSPC$GSPC.Adjusted, n=200, )
Fast.Diff <- GSPC.SMA.10 - GSPC.SMA.50
Slow.Diff <- GSPC.SMA.50 - GSPC.SMA.200

# look for long entries
Long_Trades <- ifelse(
Slow.Diff  > 0 &
Fast.Diff  > 0 &
shift(v=as.numeric(Fast.Diff), places=1, dir="right") < 0, GSPC$GSPC.Adjusted, NA)

# look for long exits (same thing but inverse signts)
Short_Trades <- ifelse(
Slow.Diff  < 0 &
Fast.Diff  < 0 &
shift(v=as.numeric(Fast.Diff), places=1, dir="right") > 0, GSPC$GSPC.Adjusted, NA)
plot(GSPC$GSPC.Adjusted)
```



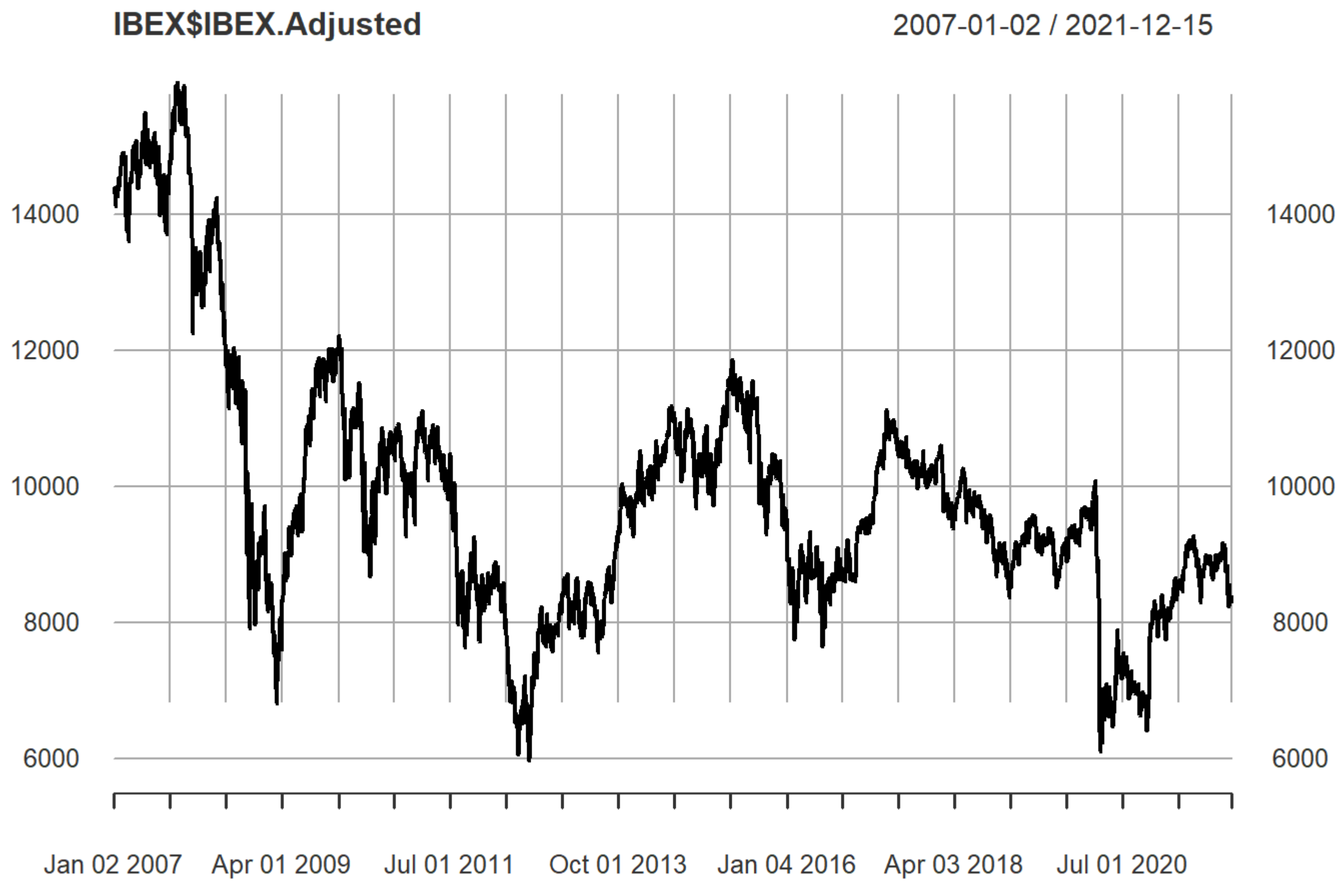
```
## Warning in plot.xts(EWP): only the univariate series will be plotted
points(Long_Trades, col='blue', cex=1.5, pch=18)
```



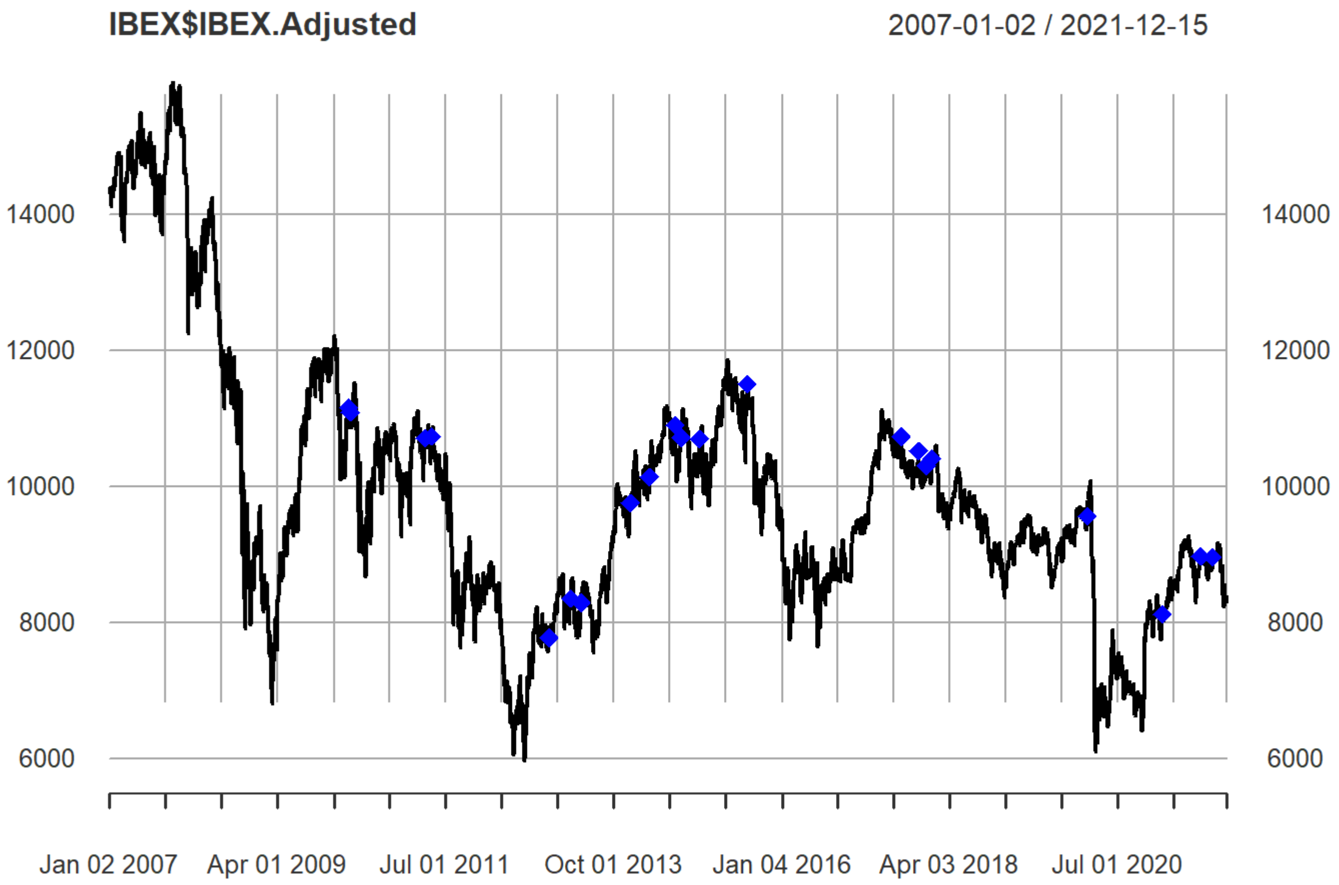
```
points(Short_Trades, col='red', cex=1.5, pch=18)
```


GSPC$GSPC.Adjusted
2007-01-03 / 2021-12-15
4000
3000
2000
1000
4000
3000
2000
1000
Jan 03 2007
Apr 01 2009
Jul 01 2011
Oct 01 2013
Jan 04 2016
Apr 02 2018
Jul 01 2020

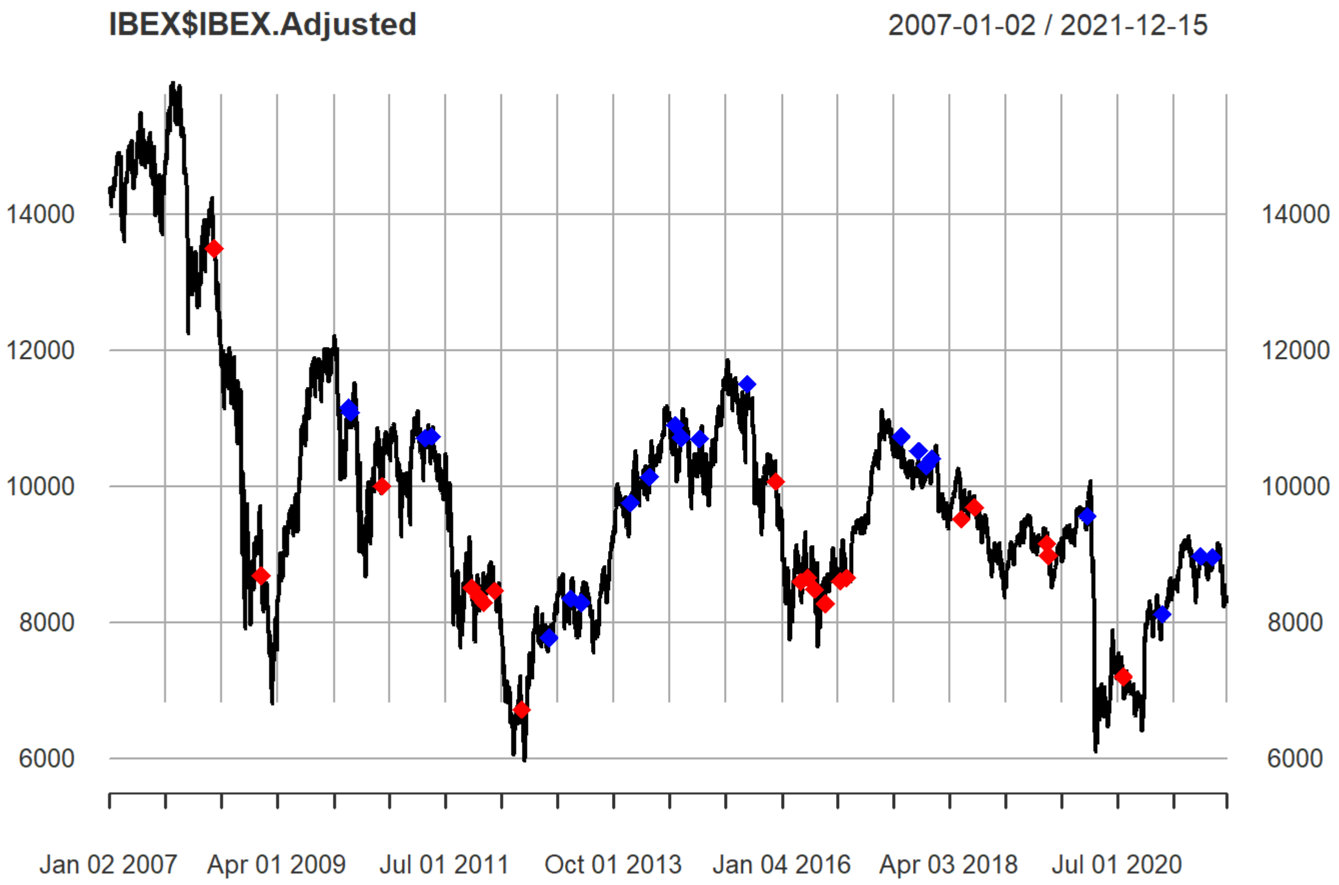
```
#Mixture of entry points and that is usually how it works on a trading, bouncing trend.
#Though we aren?t going to design full trending systems here, a stop-loss exit order is key to any directional
#trading so you don?t lose everything! Let?s see what it does on trending market:



IBEX.EMA.10 <- EMA(IBEX$IBEX.Adjusted, n=10 )
IBEX.EMA.50 <- EMA(IBEX$IBEX.Adjusted, n=50, )
IBEX.EMA.200 <- EMA(IBEX$IBEX.Adjusted, n=200, )
Fast.Diff <- IBEX.EMA.10 - IBEX.EMA.50
Slow.Diff <- IBEX.EMA.50 - IBEX.EMA.200

# look for long entries
Long_Trades <- ifelse(
  Slow.Diff  > 0 &
    Fast.Diff  > 0 &
    shift(v=as.numeric(Fast.Diff), places=1, dir="right") < 0, IBEX$IBEX.Adjusted, NA)

# look for long exits (same thing but inverse signts)
Short_Trades <- ifelse(
  Slow.Diff  < 0 &
    Fast.Diff  < 0 &
    shift(v=as.numeric(Fast.Diff), places=1, dir="right") > 0, IBEX$IBEX.Adjusted, NA)

plot(IBEX$IBEX.Adjusted)
```



```
points(Long_Trades, col='blue', cex=1.5, pch=18)
```



```
points(Short_Trades, col='red', cex=1.5, pch=18)
```



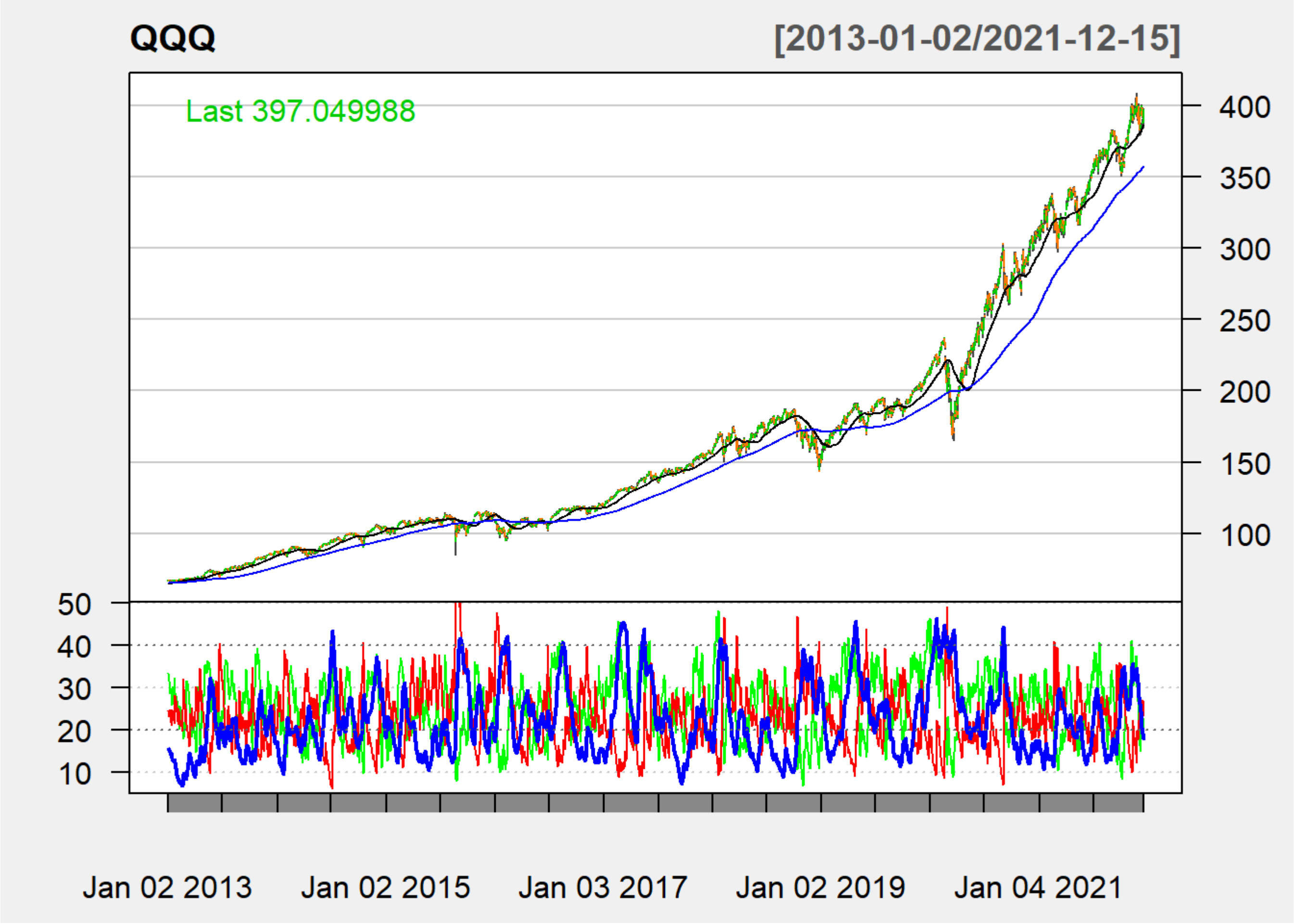
```
print ("STEP 2.6:Volume-based indicators")
```

```
## [1] "STEP 2.6:Volume-based indicators"
```

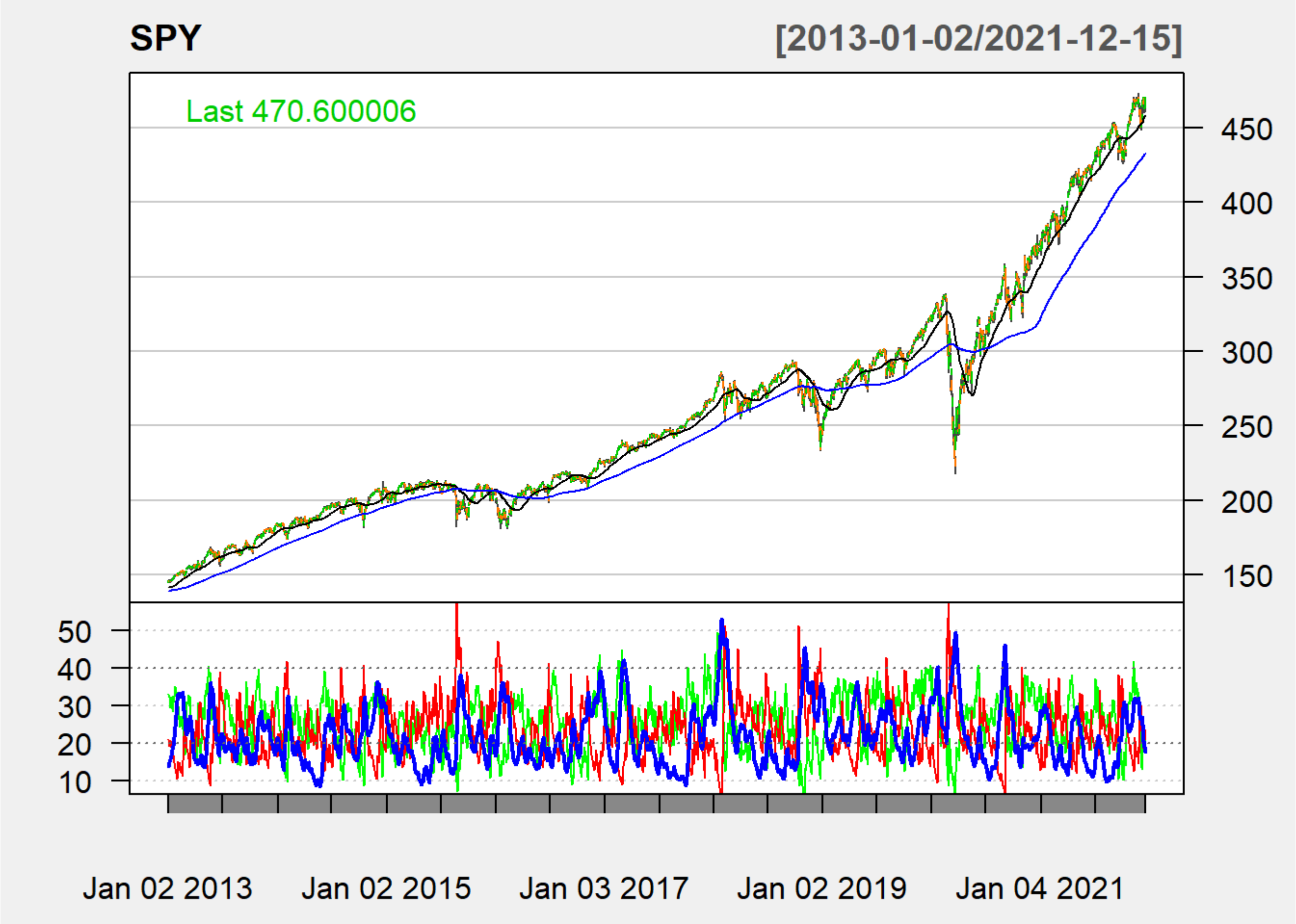
```
library(quantmod)
getSymbols(c('QQQ', 'SPY'), src='yahoo')
```

```
## [1] "QQQ" "SPY"
```

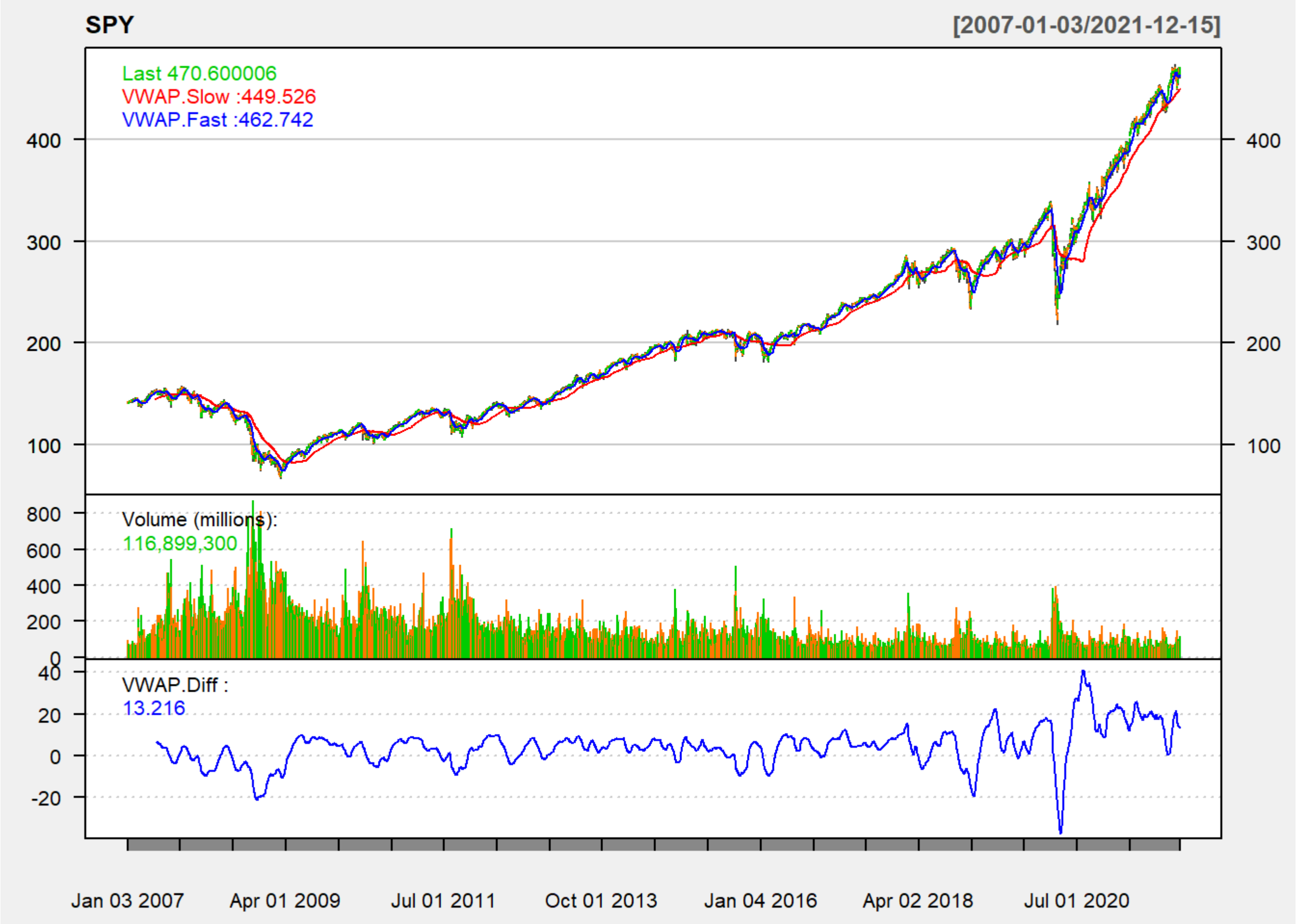
```
# remove any NAs
QQQ <- QQQ[!(rowSums(is.na(QQQ))),]
SPY <- SPY[!(rowSums(is.na(SPY))),]

library(TTR)

#The ADX is Welles Wilder?s Directional Movement Indicator. It is used by lots of people to dete
rmine if the market is trending or range bound.
# Refrence: https://en.wikipedia.org/wiki/Average_directional_movement_index
chartSeries(QQQ, theme="white", TA="addSMA(50, col='black');addSMA(200, col='blue');addADX(n = 1
4, maType='EMA', wilder=TRUE)", subset='2013::')
```



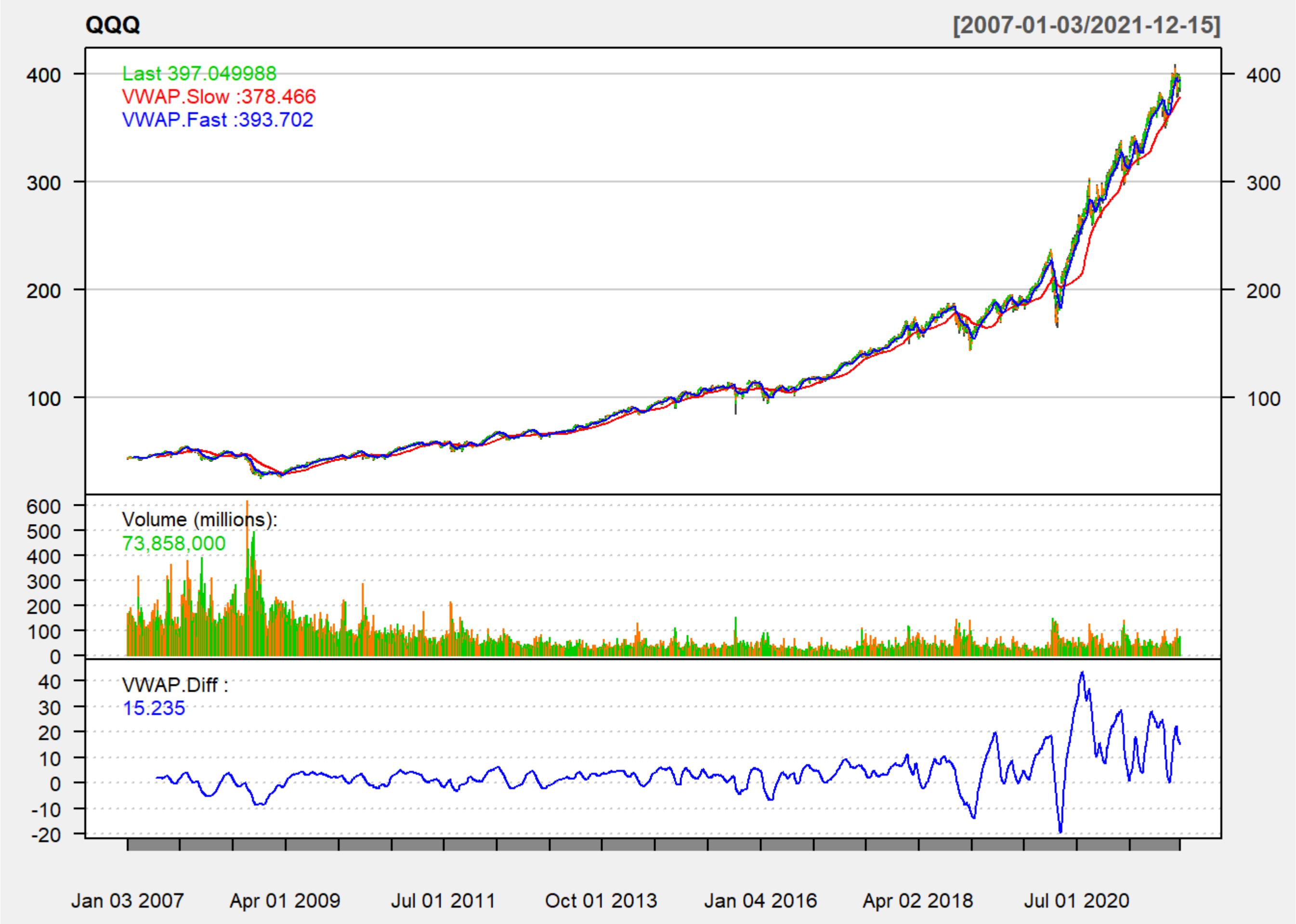
```
# Look into price as of 2013 and onward
chartSeries(SPY, theme="white", TA="addSMA(50, col='black');addSMA(200, col='blue');addADX(n = 1
4, maType='EMA', wilder=TRUE)", subset='2013::')
```


SPY
[2013-01-02/2021-12-15]
Last 470.600006
450
400
350
300
250
200
150
50
40
30
20
10
Jan 02 2013
Jan 02 2015
Jan 03 2017
Jan 02 2019
Jan 04 2021

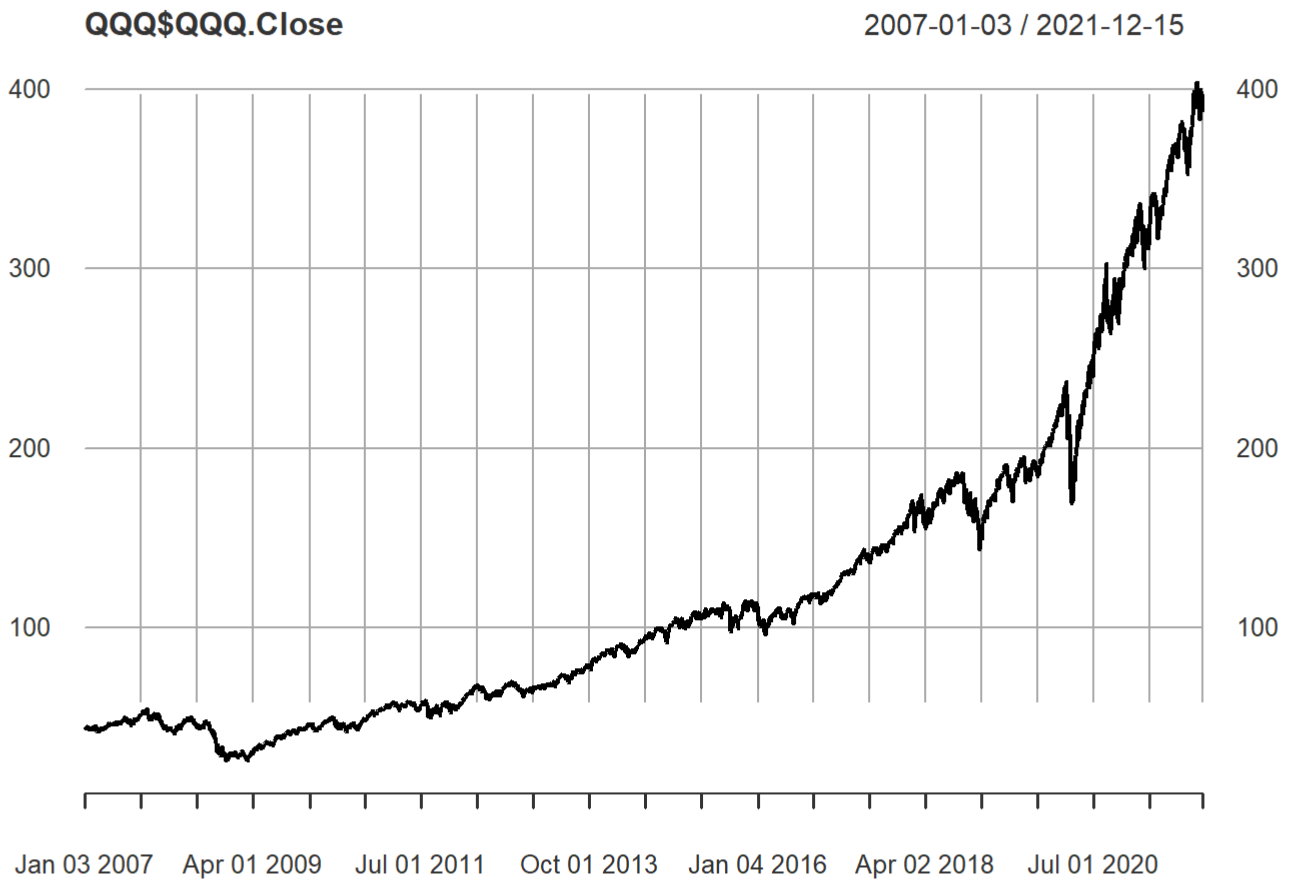
```
#In a nutshell, Welles recommends using the ADX with a 14-day period. When the main blue line is above 20, it is considered a strong,
#trending market, when it is below, it is considered a weak one.
#Volume

#As this is an introductory course, we?re mostly using the closing price but it is important to note that there are a lot of other market variables available.
#You can design systems with the open price, the high or low, the difference between the open and close, etc. And there is also the volume.

#This an important indicator. A falling stack on rising volume or a rising stock on falling volume may mean the move is about to
#reverse. Whatever the reason for abnormal volume, it should be a warning to keep a vigilant eye on the stock.

#There are plenty of indicators that include the volume price such as the Volume-weighted average price (VWAP).
#The VWAP is a guide more than a trading indicator as to where the market is trading compared to the volume adjusted price.
#It divides dollars traded by volume (see above link for more details).

VWAP.Slow <- VWAP(price=SPY$SPY.Close, volume=SPY$SPY.Volume, n=100)
VWAP.Fast <- VWAP(price=SPY$SPY.Close, volume=SPY$SPY.Volume, n=20)
VWAP.Diff <- VWAP.Fast- VWAP.Slow

chartSeries(SPY, theme="white", TA="addVo();addTA(VWAP.Slow, on=1, col='red');addTA(VWAP.Fast, on=1, col='blue');addTA(VWAP.Diff, col='blue')")
```



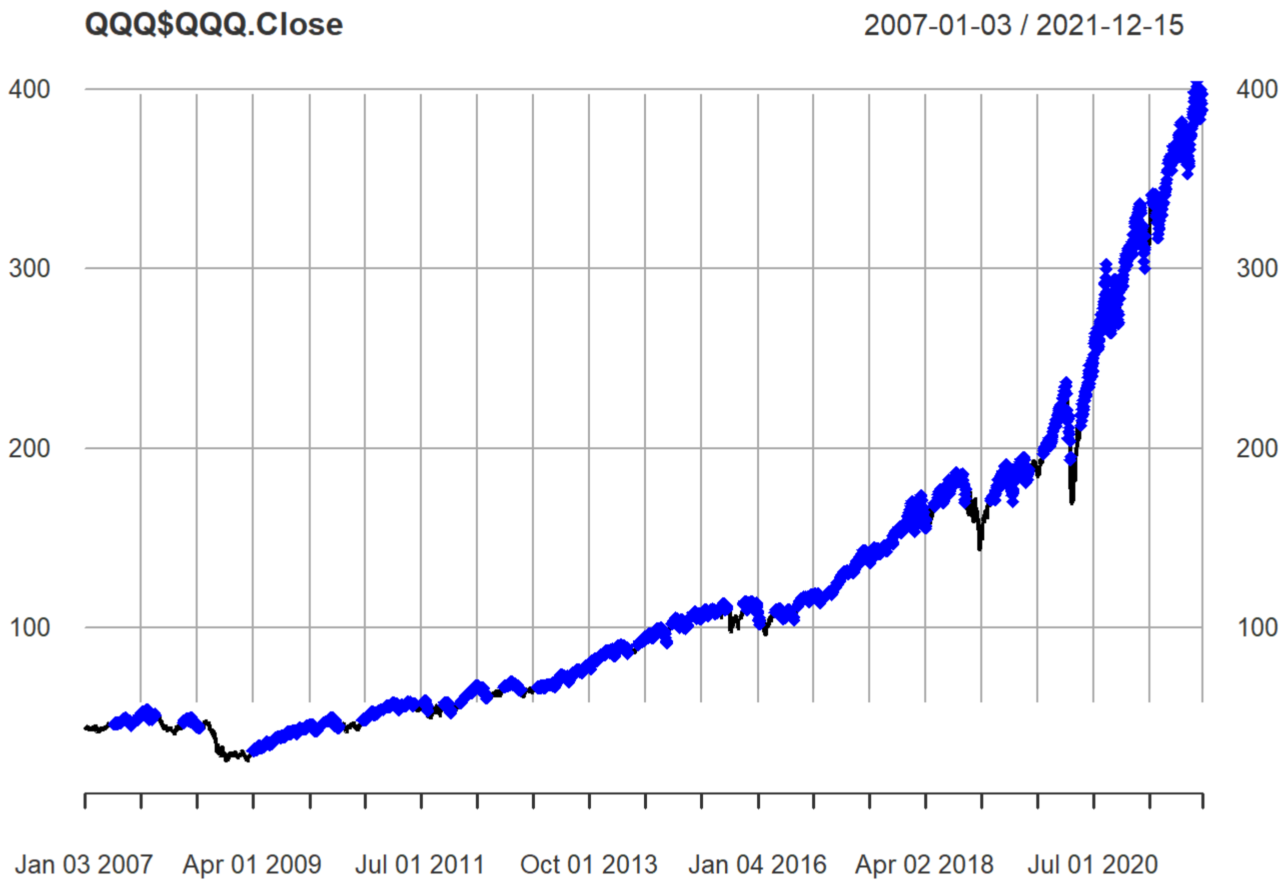
```
# QQQ
VWAP.Slow <- VWAP(price=QQQ$QQQ.Close, volume=QQQ$QQQ.Volume, n=100)
VWAP.Fast <- VWAP(price=QQQ$QQQ.Close, volume=QQQ$QQQ.Volume, n=20)
VWAP.Diff <- VWAP.Fast- VWAP.Slow

chartSeries(QQQ, theme="white", TA="addVo();addTA(VWAP.Slow, on=1, col='red');addTA(VWAP.Fast, on=1, col='blue');addTA(VWAP.Diff, col='blue')")
```



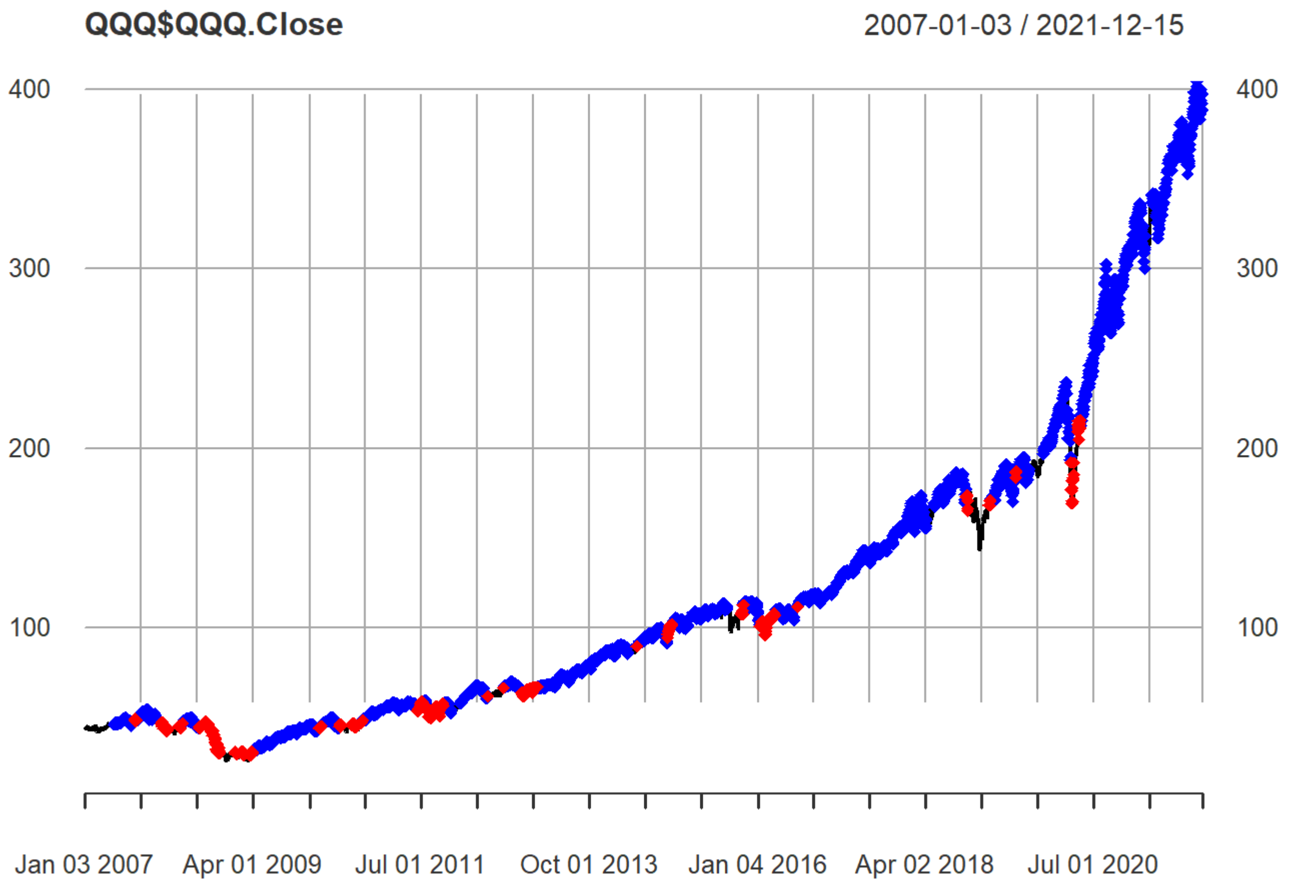
```
ADX.20 <- ADX(QQQ,n=14)

# look for long entries
Long_Trades <- ifelse(
  ADX.20$ADX > 20 &
    VWAP.Diff> 0, QQQ$QQQ.Close, NA)

# look for long entries
Short_Trades <- ifelse(
  ADX.20$ADX > 20 &
    VWAP.Diff < 0, QQQ$QQQ.Close, NA)

plot(QQQ$QQQ.Close)
```



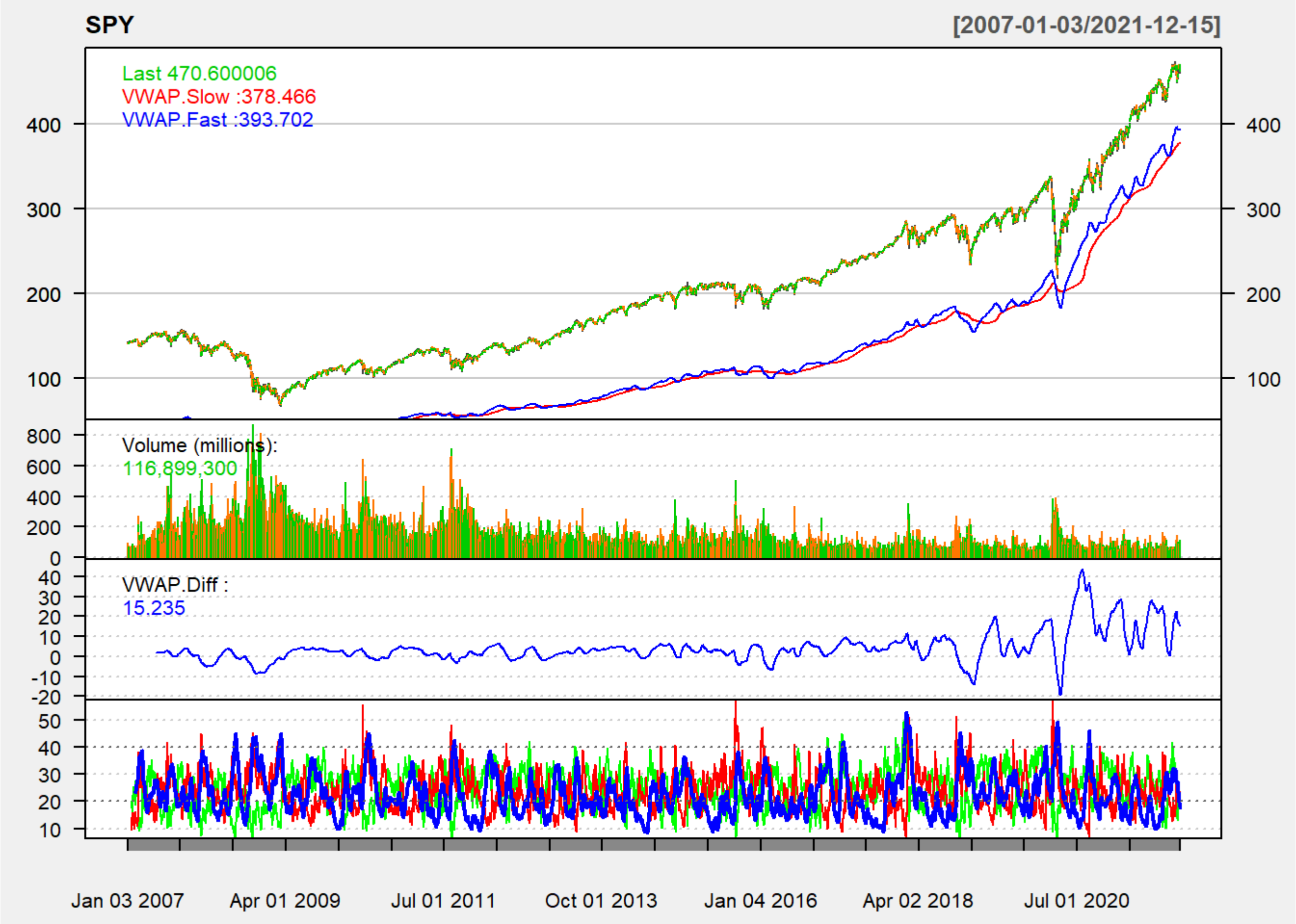
```
points(Long_Trades, col='blue', cex=1, pch=18)
```



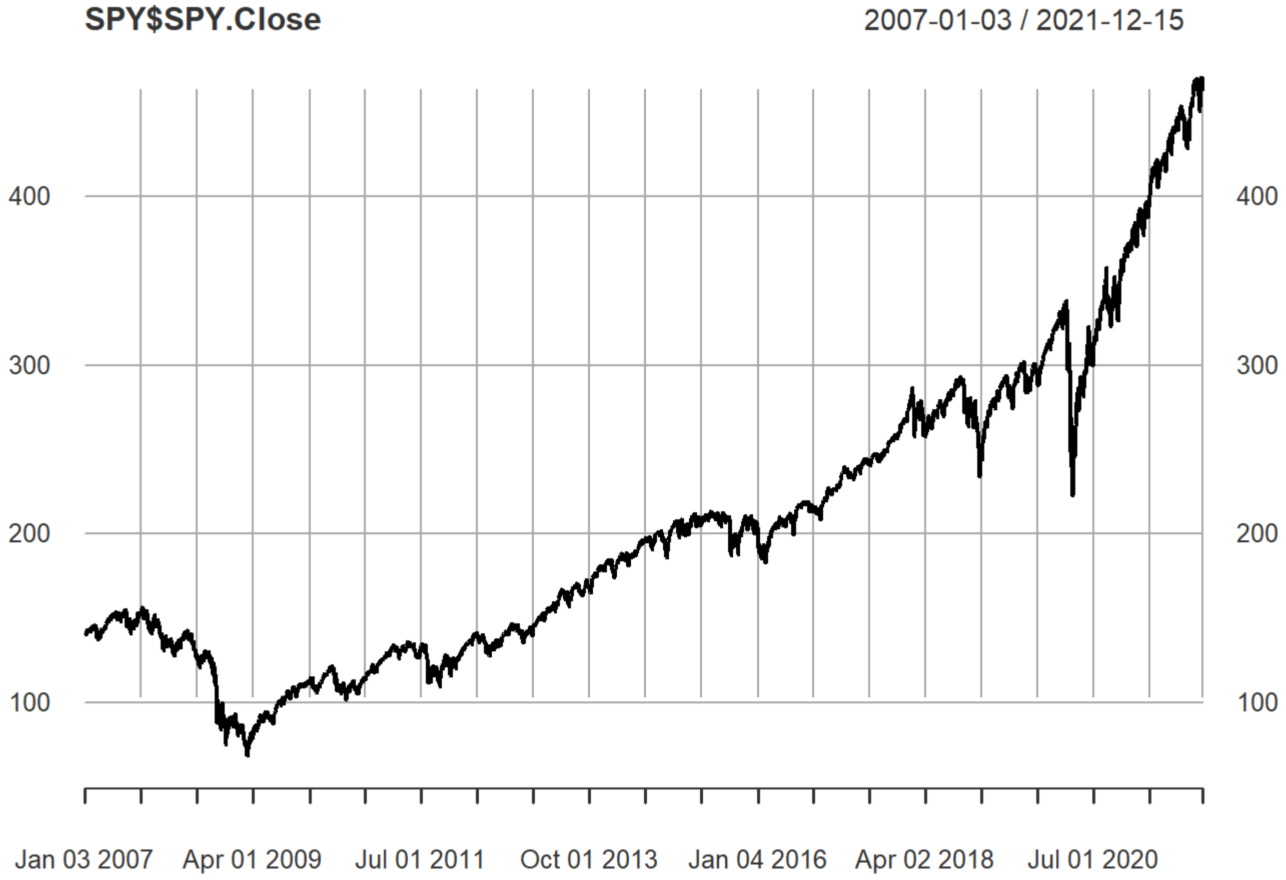
```
points(Short_Trades, col='red', cex=1, pch=18)
```



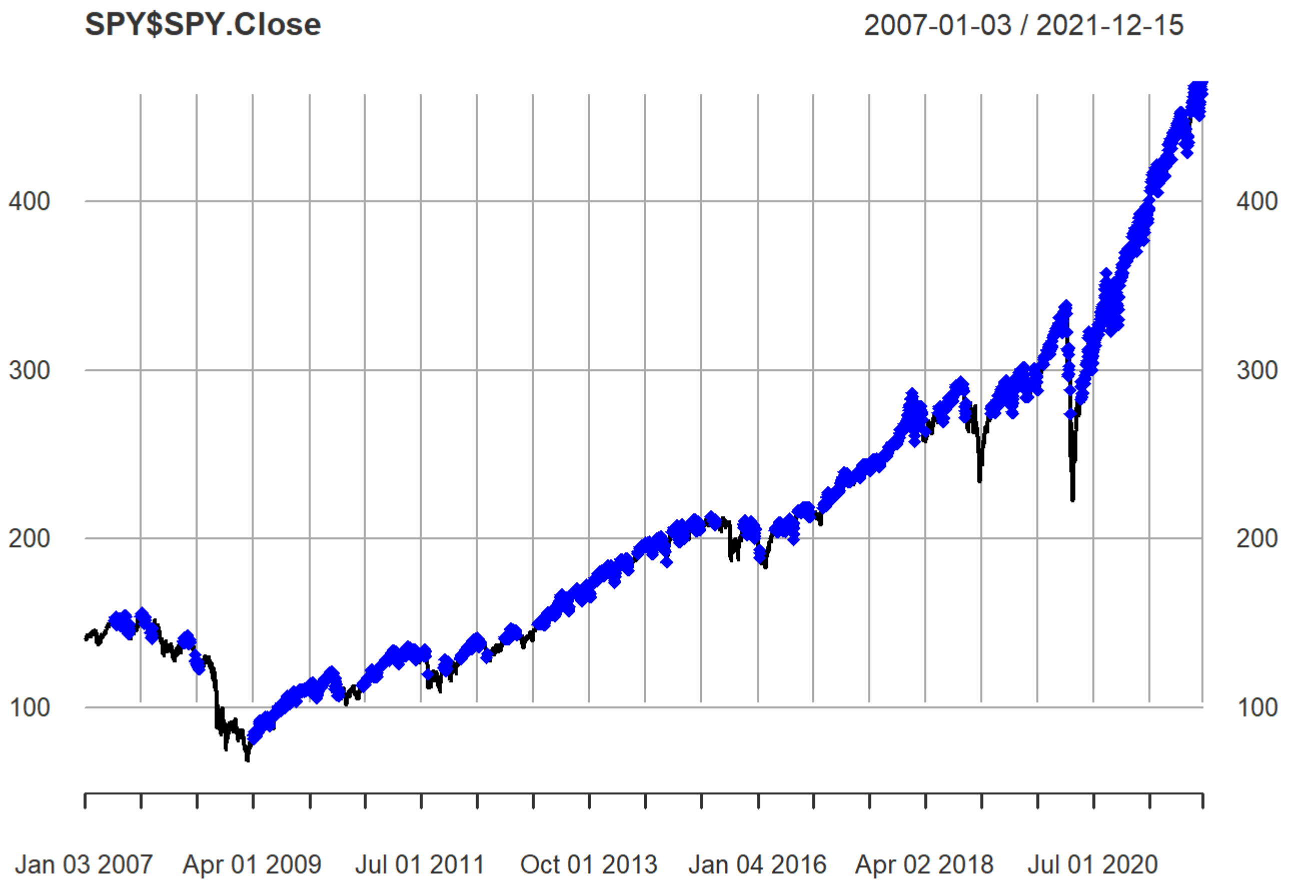
```
chartSeries(SPY, theme="white", TA="addVo();addTA(VWAP.Slow, on=1, col='red');addTA(VWAP.Fast, on=1, col='blue');addTA(VWAP.Diff, col='blue');
            addADX(n = 14, maType='EMA', wilder=TRUE)")
```



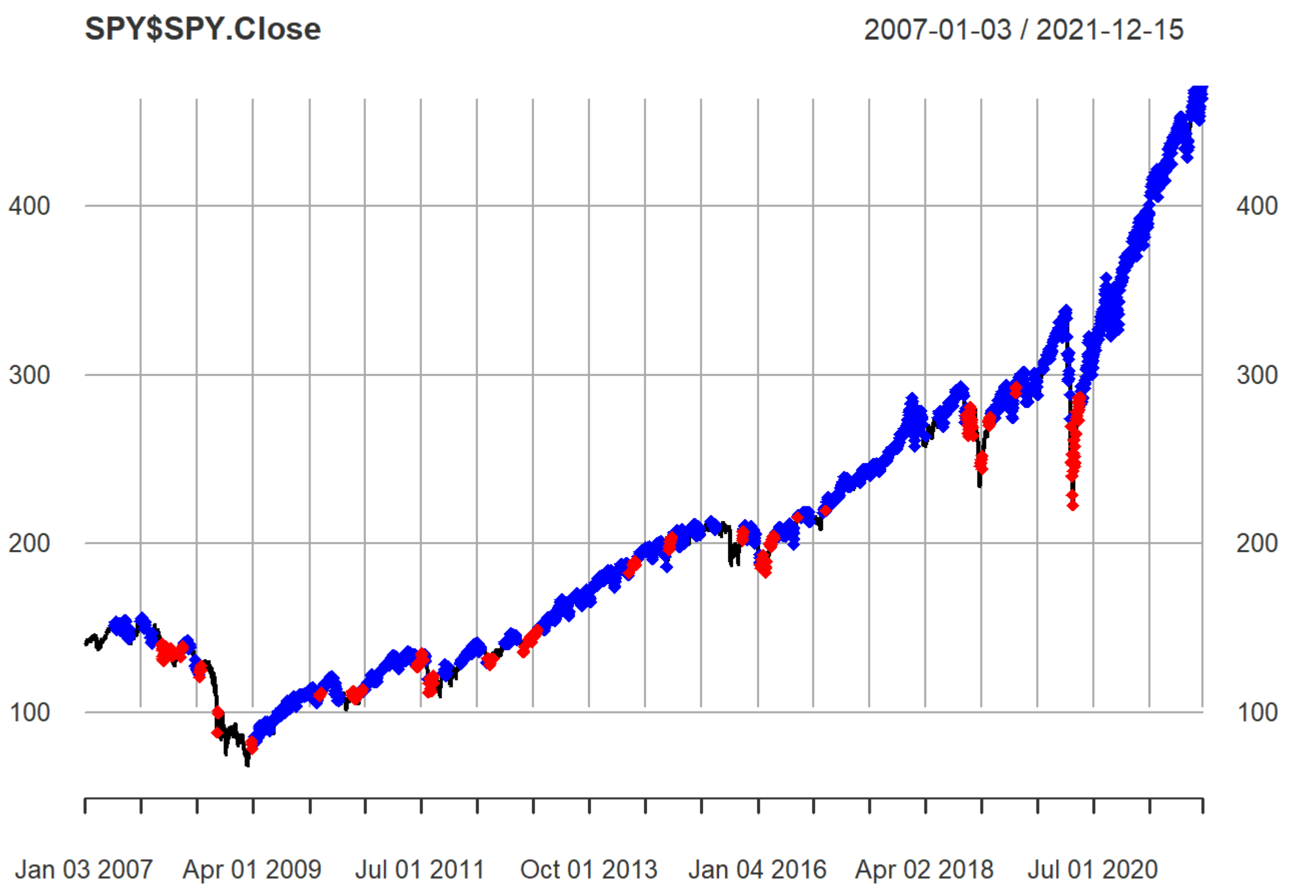
```
ADX.20 <- ADX(SPY,n=14)

# look for long entries
Long_Trades <- ifelse(
        ADX.20$ADX > 20 &
        VWAP.Diff> 0, SPY$SPY.Close, NA)

# look for long entries
Short_Trades <- ifelse(
        ADX.20$ADX > 20 &
        VWAP.Diff < 0, SPY$SPY.Close, NA)

plot(SPY$SPY.Close)
```



```
## Warning in plot.xts(SPY): only the univariate series will be plotted
points(Long_Trades, col='blue', cex=1, pch=18)
```



```
points(Short_Trades, col='red', cex=1, pch=18)
```



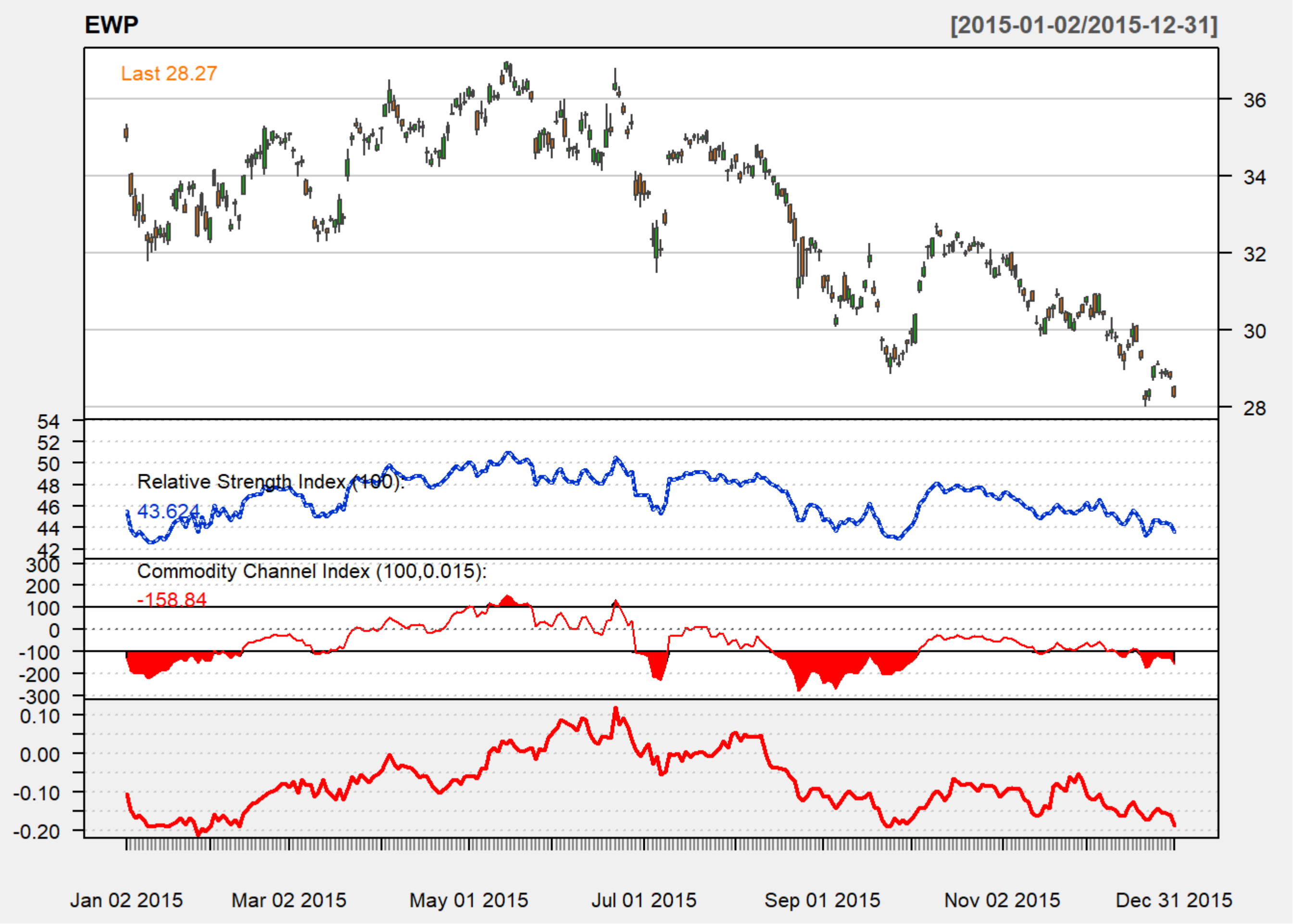
```
print ("STEP 2.7: Counter-Trend Systems including * Momentum Indicators * Volatility Indicator * Counter-Trend Systems")
```

```
## [1] "STEP 2.7: Counter-Trend Systems including * Momentum Indicators * Volatility Indicator * Counter-Trend Systems"
```

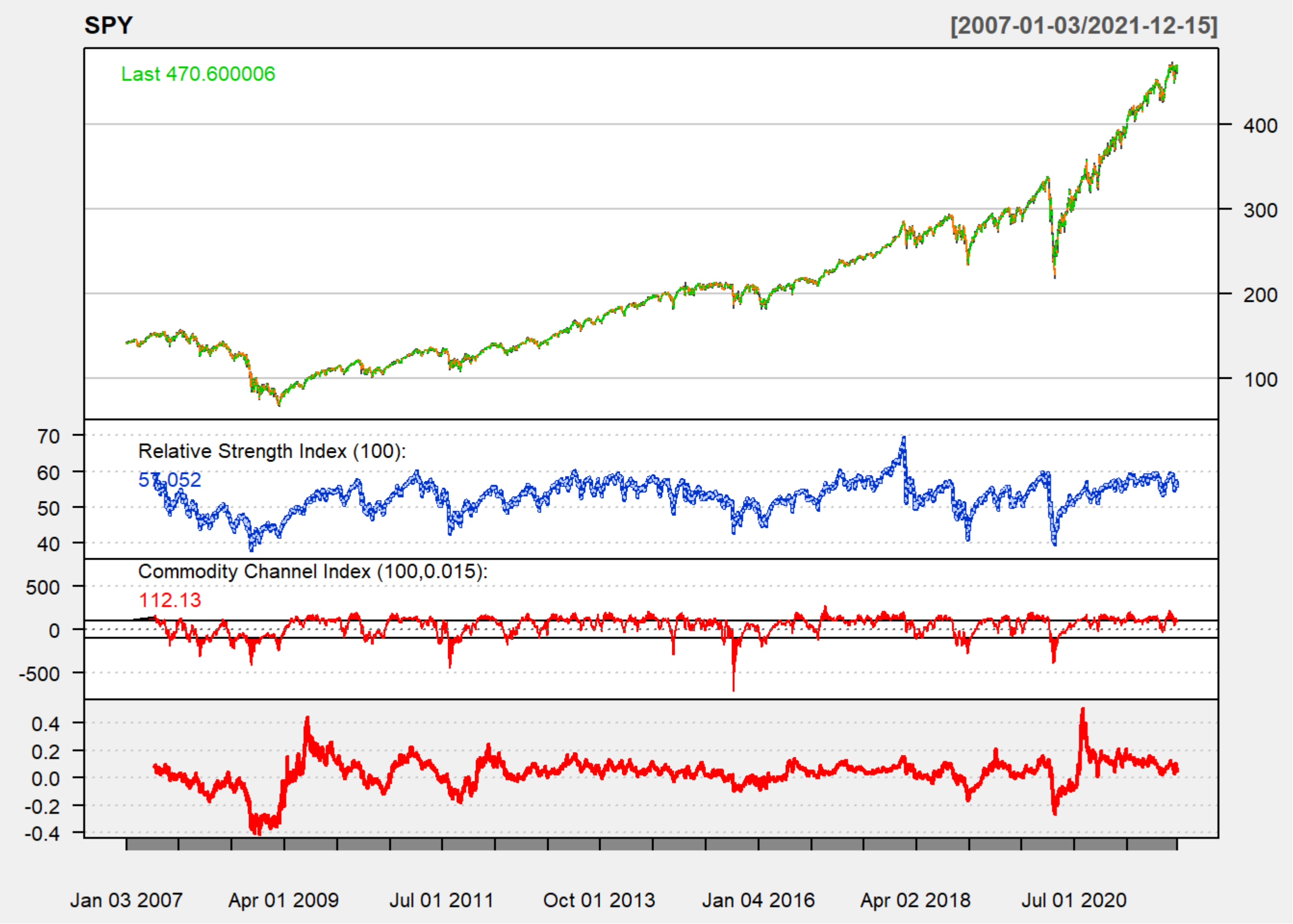
```
#Counter-trend systems are tricky. You trade raw counter trends when you?re sure you?re in a range-bound market
#and are trading at the extremes otherwise you use added indicators to stay aligned with longer-term trends.
#Raw counter-trend trading feels like picking tops and bottoms, and those rarely work out.
#Here we?ll focus on trading the short-term counter trend, while following the long-term trend.


library(binhf)
library(quantmod)
getSymbols(c('EWP', 'SPY'), src='yahoo')
```

```
## [1] "EWP" "SPY"
```

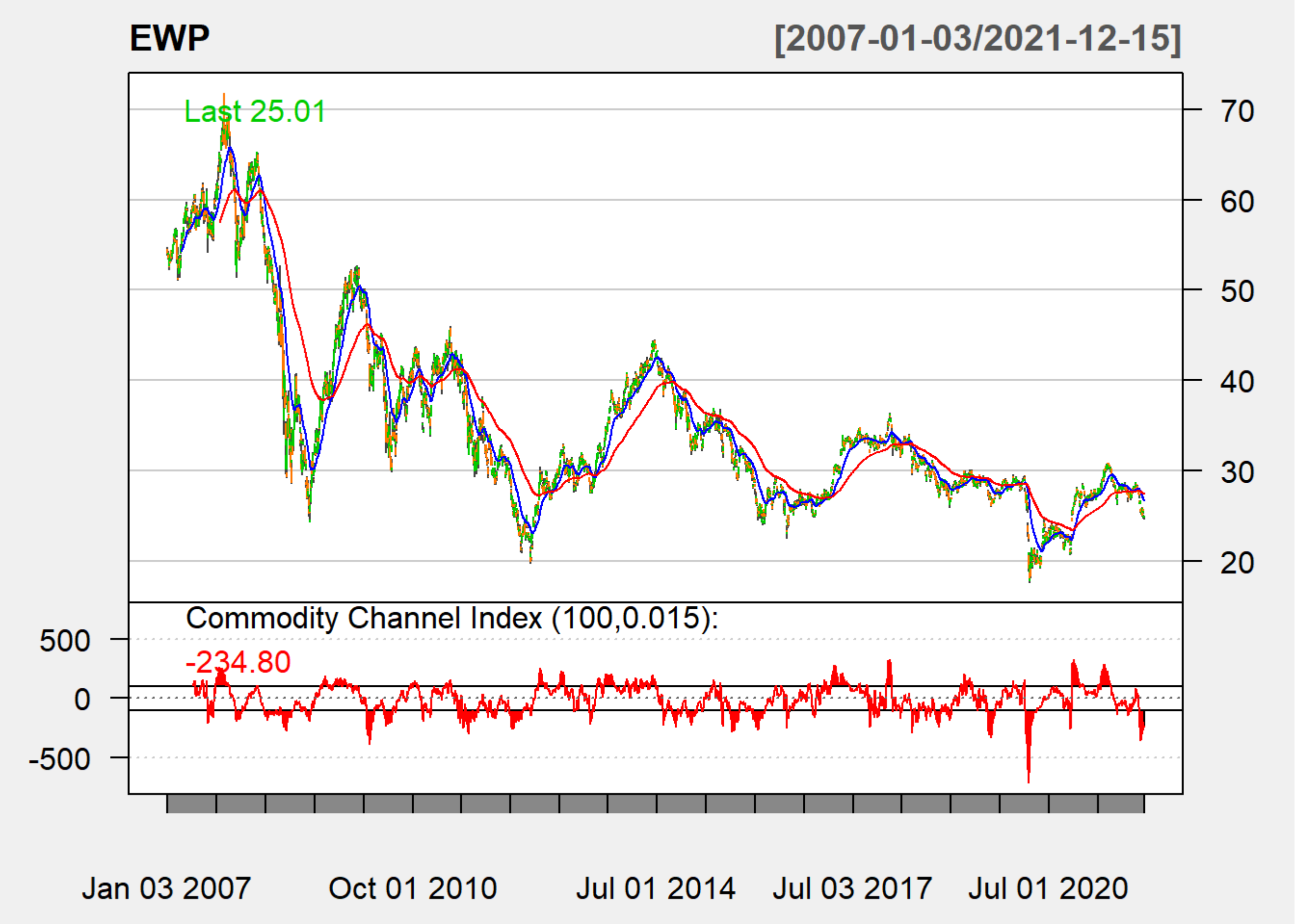
```
# remove any NAs
EWP <- EWP[!(rowSums(is.na(EWP))),]
SPY <- SPY[!(rowSums(is.na(SPY))),]



#Momentum Indicators

#We?re going to look at 3 interesting momentum indicators that capture short-term cycles:

#Relative Strength Index (RSI), is an momentum indicator that measures movement. Its author, J.
 Welles Wilder, recommends using a period of 14 and when it is over 70, it is strongly bought (o
r overbought) and under 30, it is strongly sold (or oversold).
#REF: https://en.wikipedia.org/wiki/Relative_strength_index
#Commodity Channel Index (CCI) by Donald Lambert, is a price-derived indicator revolving around
 0, where 100 is usually considered overbought and -100, oversold.
#REF:https://en.wikipedia.org/wiki/Commodity_channel_index
#Rate of Change (ROC), also a momentum indicator, looks at accelerating and decelerating market
 moves.
#REF:https://en.wikipedia.org/wiki/Momentum_(technical_analysis)



#Let?s look at all 3 of them with a 20-period setting:

chartSeries(EWP, theme="white", TA="addRSI(n=100);addCCI(n=100);addROC(n=100)", subset='2015')
```



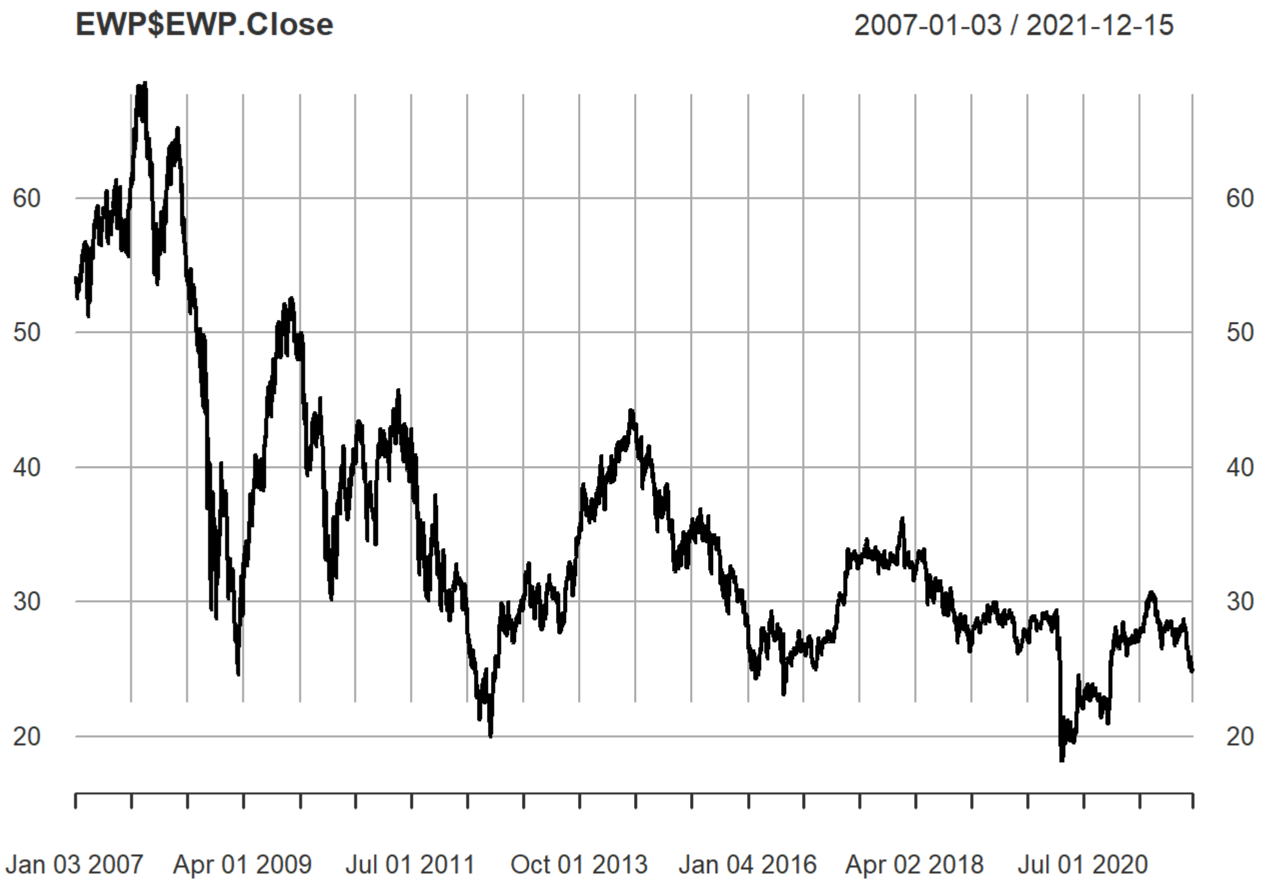
```
chartSeries(SPY, theme="white", TA="addRSI(n=100);addCCI(n=100);addROC(n=100)")
```



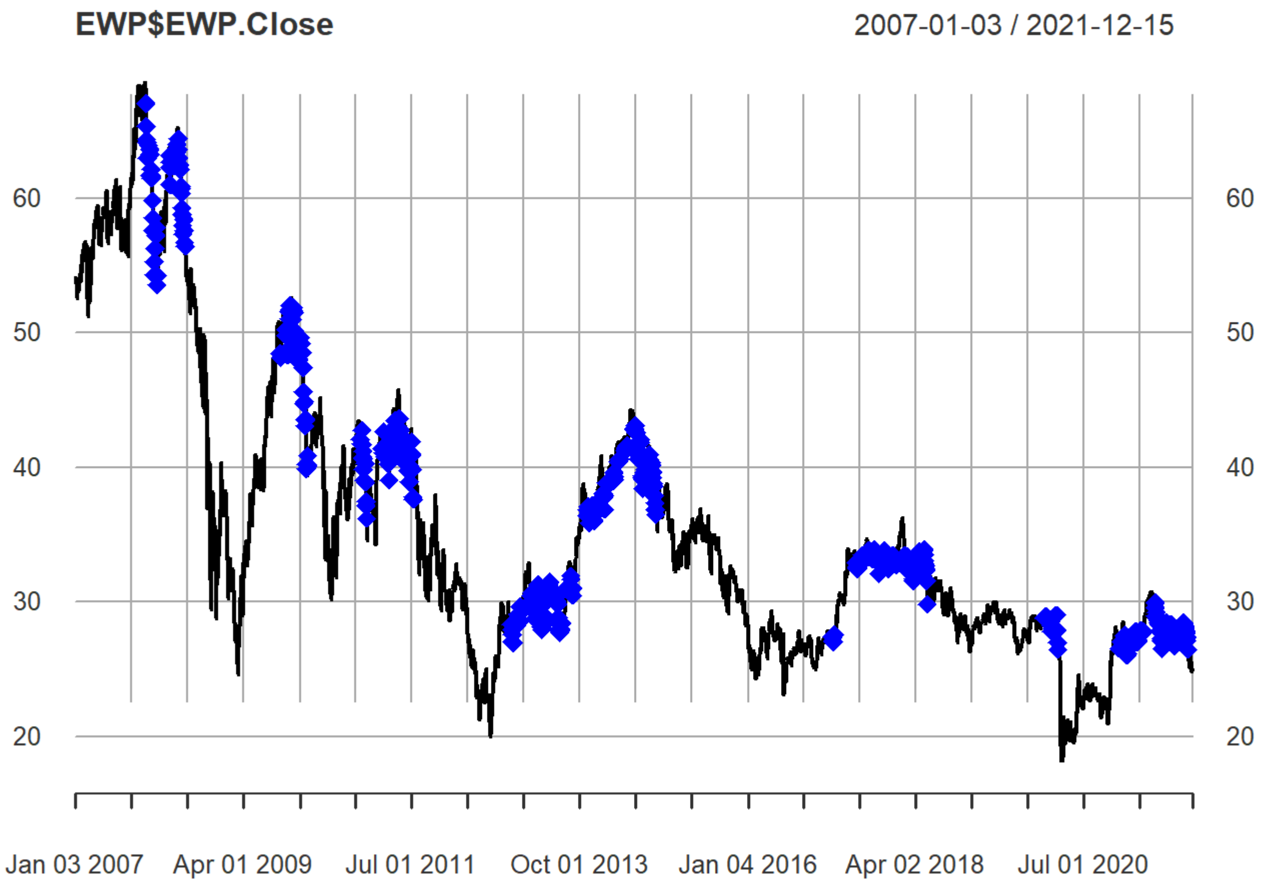
```
#Counter-Trend Systems

#For our counter-trend system, we will counter a faster cycle but stay in the direction of the s
lower one. In essence, we?re trading with the slow trend but against the fast one. While in the
 previous systems, we only took a trade while both directions aligned in the direction of the lo
ng-term trend.

#The key is to use one of the derived indicators that best signals overbought/oversold signals.

#We?ll try each one of them with a long-term EMA.

chartSeries(EWP, theme="white", TA="addCCI(n=100);addEMA(n=50,col='blue');addEMA(n=200,col='re
d')")
```



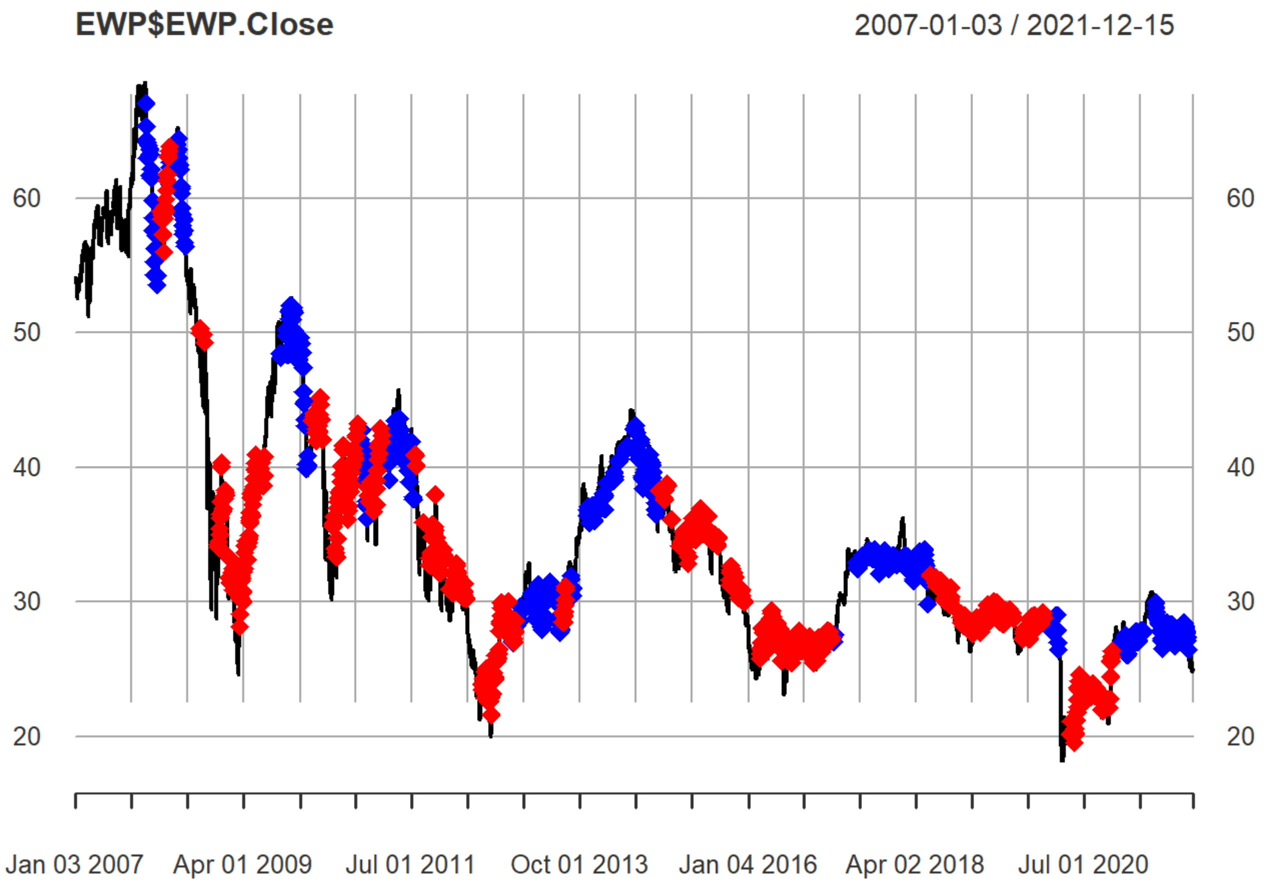
```
# create a slow ema difference
EWP.EMA.50 <- EMA(EWP$EWP.Close, n=50)
EWP.EMA.200 <- EMA(EWP$EWP.Close, n=200)
Slow.Diff <- EWP.EMA.50 - EWP.EMA.200
CCI.IND <- CCI(HLC=EWP[,c("EWP.High","EWP.Low","EWP.Close")],n=100)

# look for long entries
Long_Trades <- ifelse(
  shift(v=as.numeric(CCI.IND), places=1, dir="right") > CCI.IND &
    CCI.IND < 100 &
    Slow.Diff > 0, EWP$EWP.Close, NA)

# look for short entries
Short_Trades <- ifelse(
  shift(v=as.numeric(CCI.IND), places=1, dir="right") < CCI.IND &
    CCI.IND > -100 &
    Slow.Diff < 0, EWP$EWP.Close, NA)

plot(EWP$EWP.Close)
```



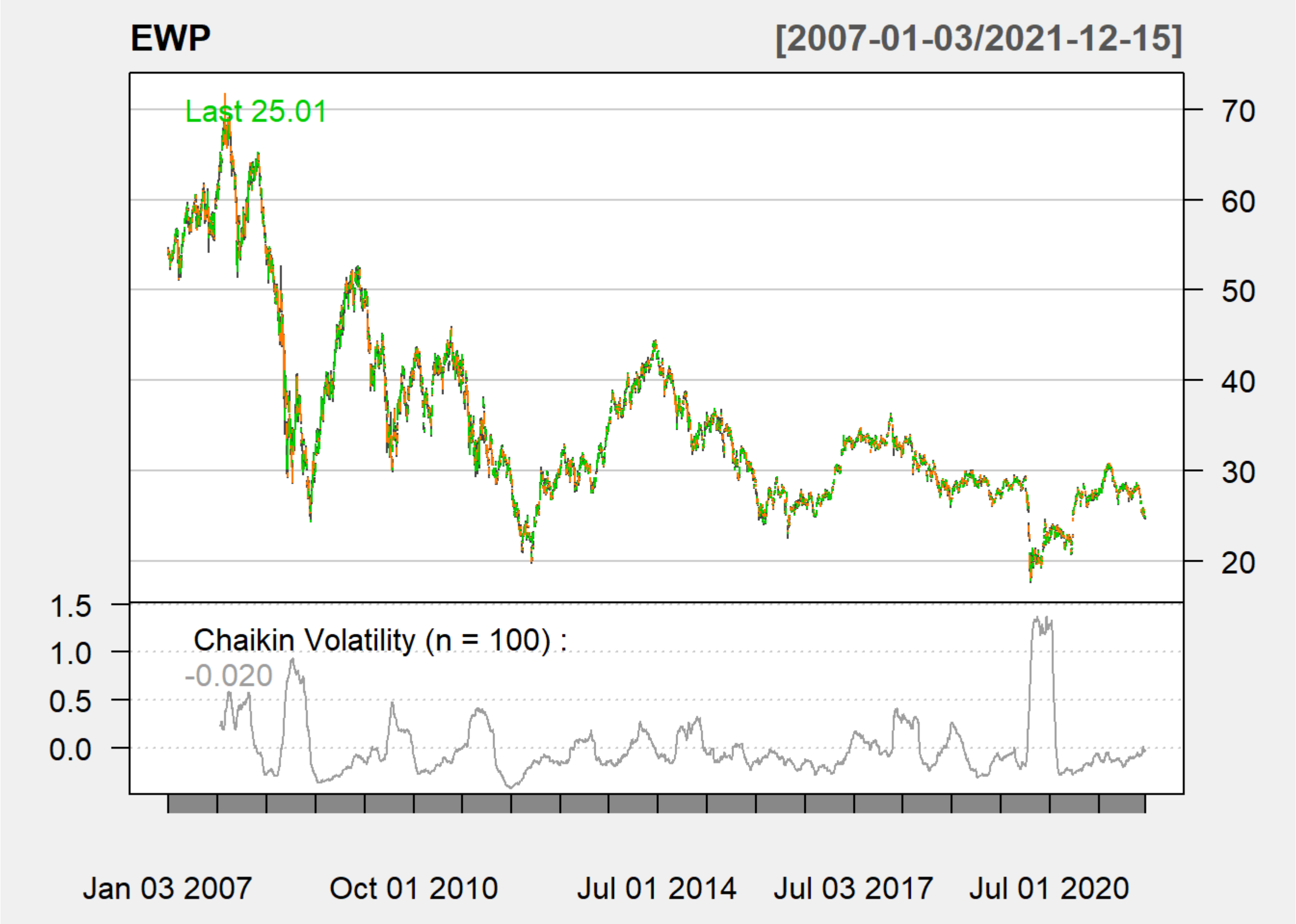
```
## Warning in plot.xts(EWP): only the univariate series will be plotted
points(Long_Trades, col='blue', cex=1.5, pch=18)
```



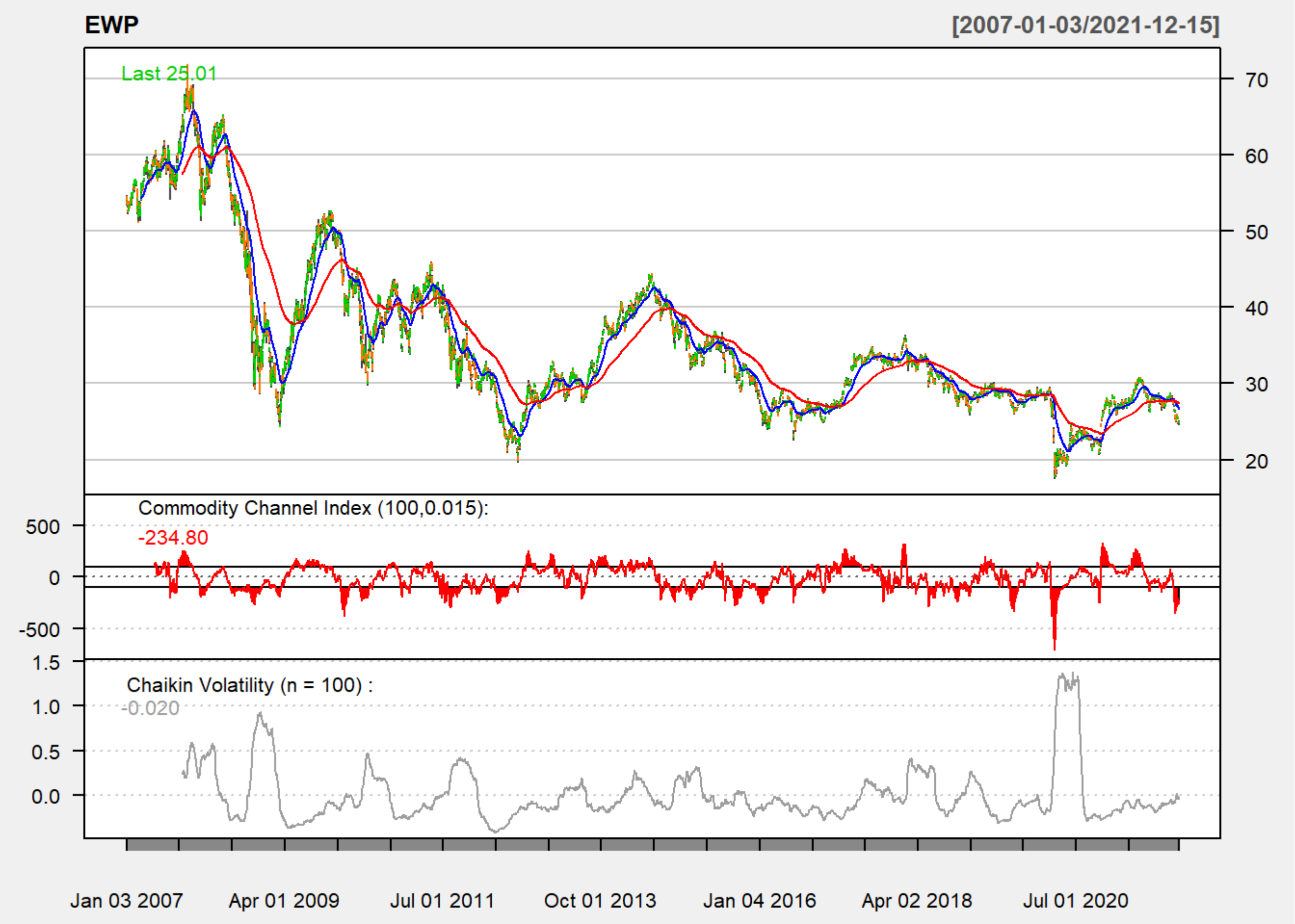
```
points(Short_Trades, col='red', cex=1.5, pch=18)
```



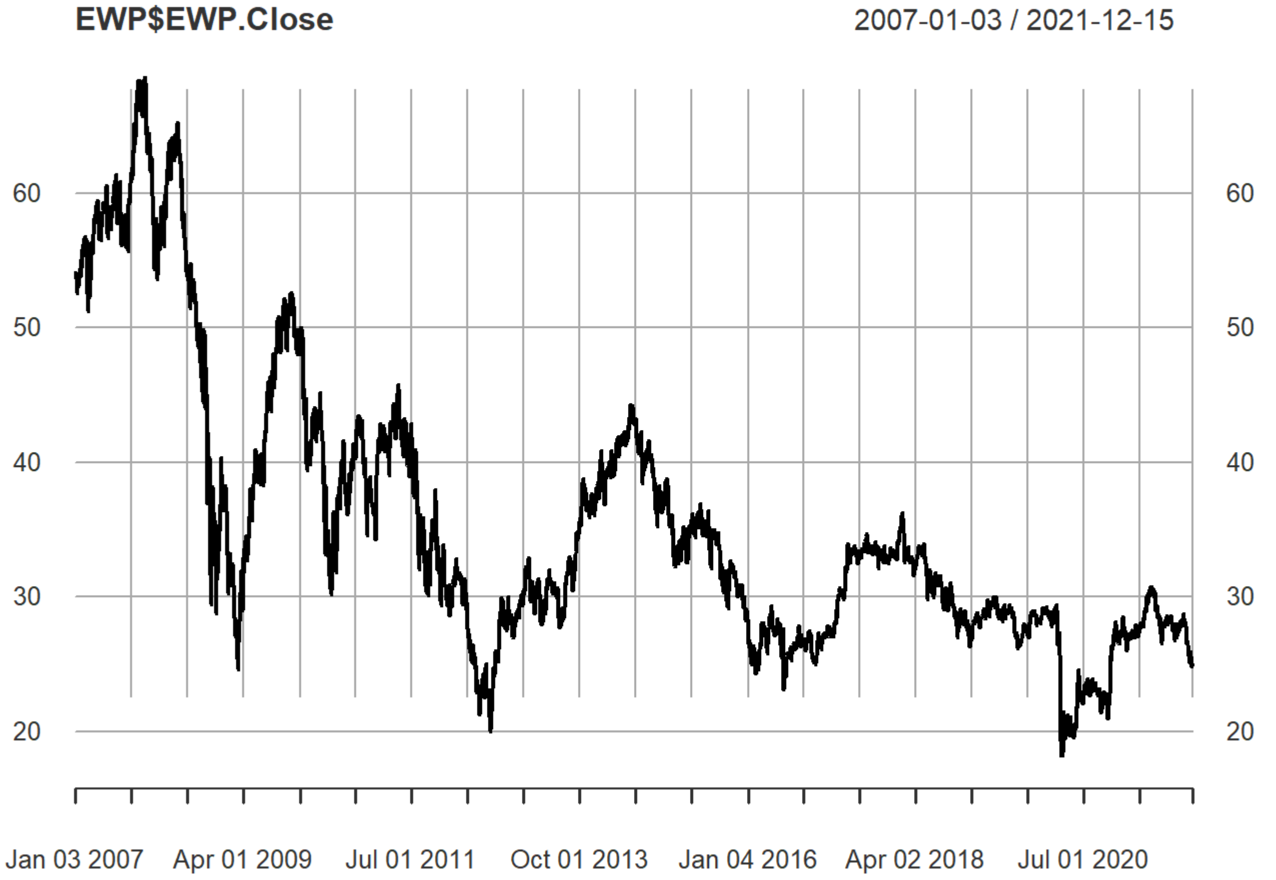
```
#Volatility indicator

#Chaikin Volatility, uses the high, low, close for its accumulation/distribution and subtracts t
wo moving averages of different
#periods of the AD.

chartSeries(EWP, theme="white", TA="addChVol(n=100);")
```



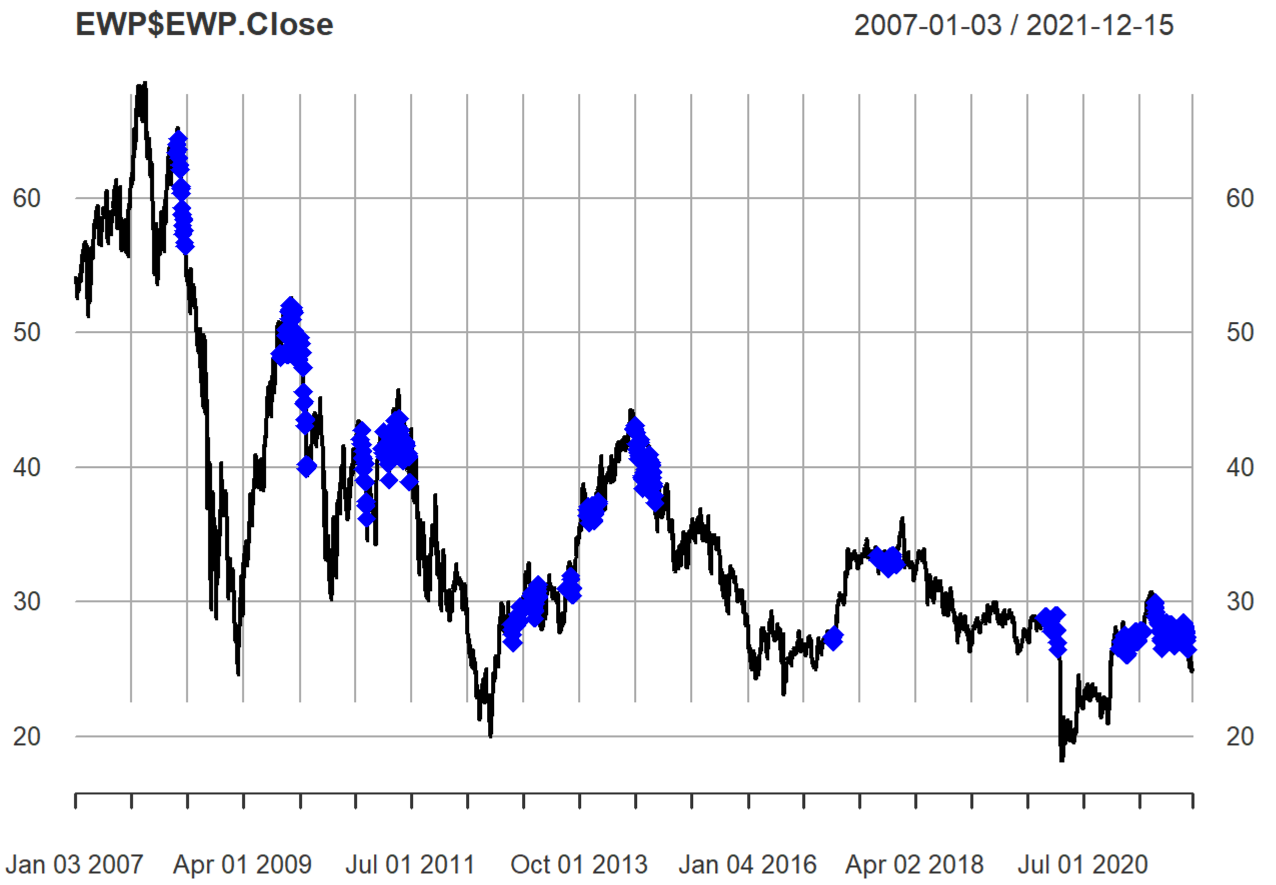
```
chartSeries(EWP, theme="white", TA="addCCI(n=100);addEMA(n=50,col='blue');addEMA(n=200,col='re
d');addChVol(n=100);")
```



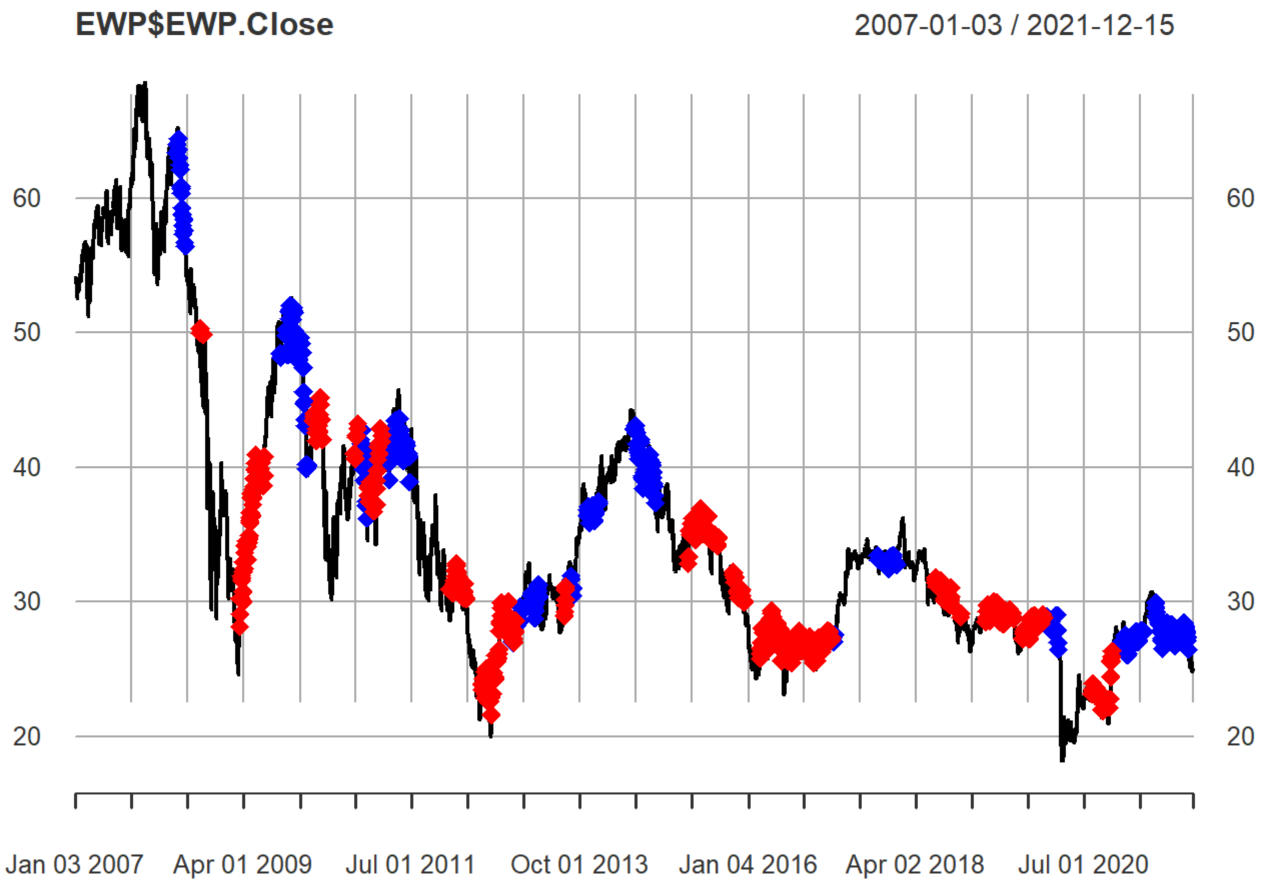
```
# create a slow ema difference
EWP.EMA.50 <- EMA(EWP$EWP.Close, n=50)
EWP.EMA.200 <- EMA(EWP$EWP.Close, n=200)
Slow.Diff <- EWP.EMA.50 - EWP.EMA.200
CCI.IND <- CCI(HLC=EWP[,c("EWP.High","EWP.Low","EWP.Close")],n=100)
CV.IND <- chaikinVolatility(HL=EWP[,c("EWP.High","EWP.Low")], n=100)

# look for long entries
Long_Trades <- ifelse(
     shift(v=as.numeric(CCI.IND), places=1, dir="right") > CCI.IND &
        CCI.IND < 100 &
        CV.IND < 0 &
        Slow.Diff > 0, EWP$EWP.Close, NA)

# look for short entries
Short_Trades <- ifelse(
       shift(v=as.numeric(CCI.IND), places=1, dir="right") < CCI.IND &
        CCI.IND > -100 &
        CV.IND < 0 &
        Slow.Diff < 0, EWP$EWP.Close, NA)

plot(EWP$EWP.Close)
```



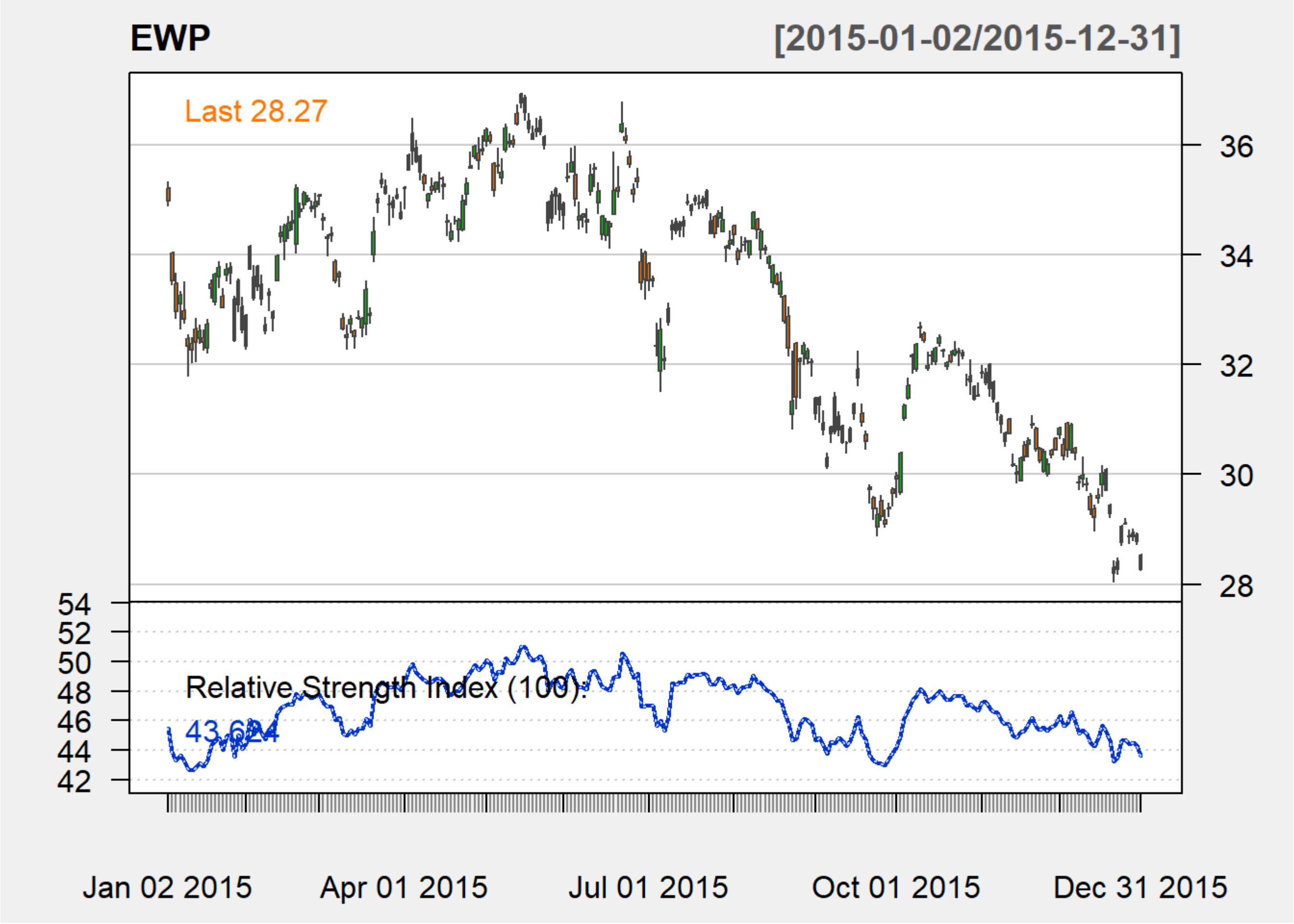
```
## Warning in plot.xts(EWP): only the univariate series will be plotted
points(Long_Trades, col='blue', cex=1.5, pch=18)
```



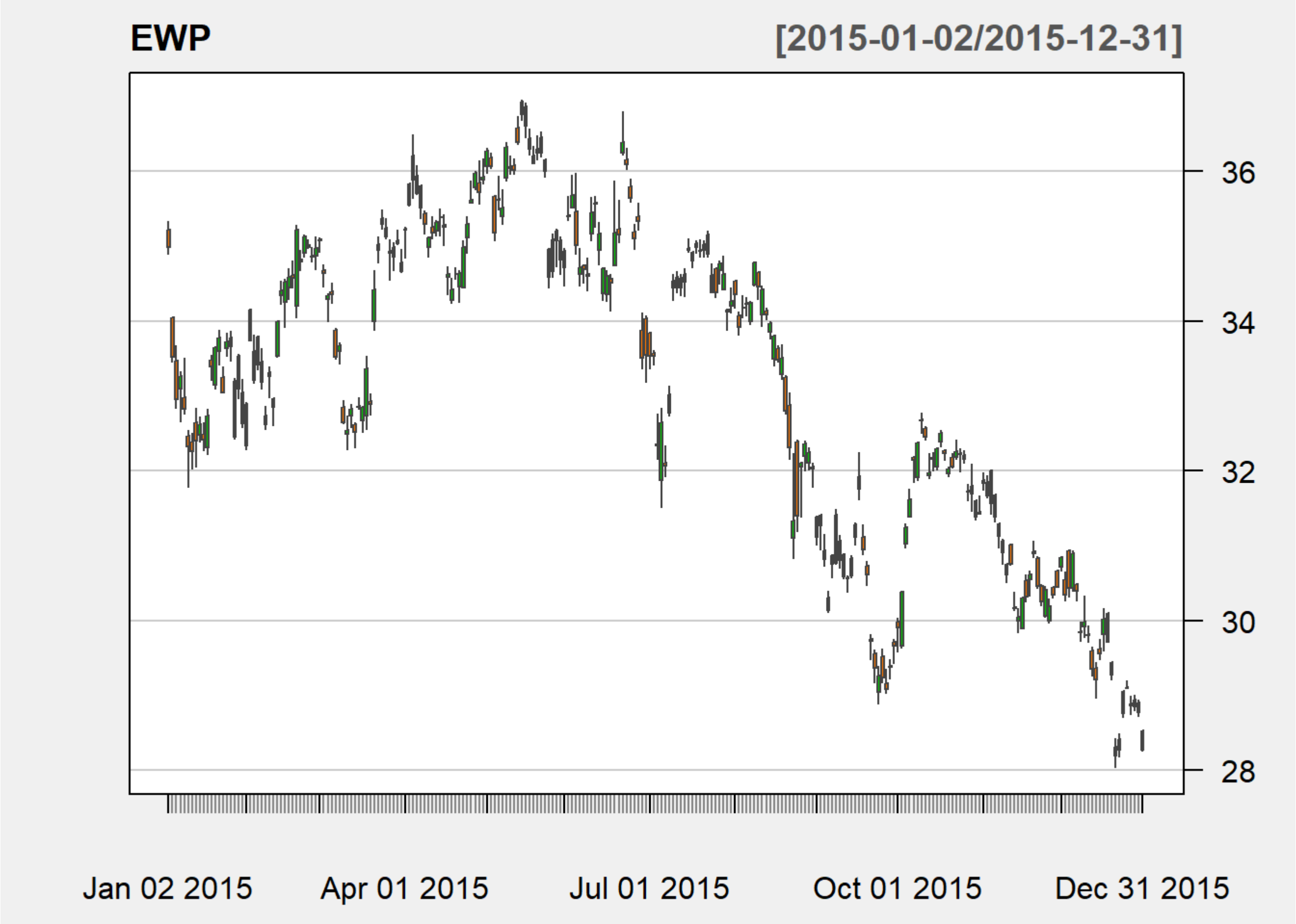
```
points(Short_Trades, col='red', cex=1.5, pch=18)
```



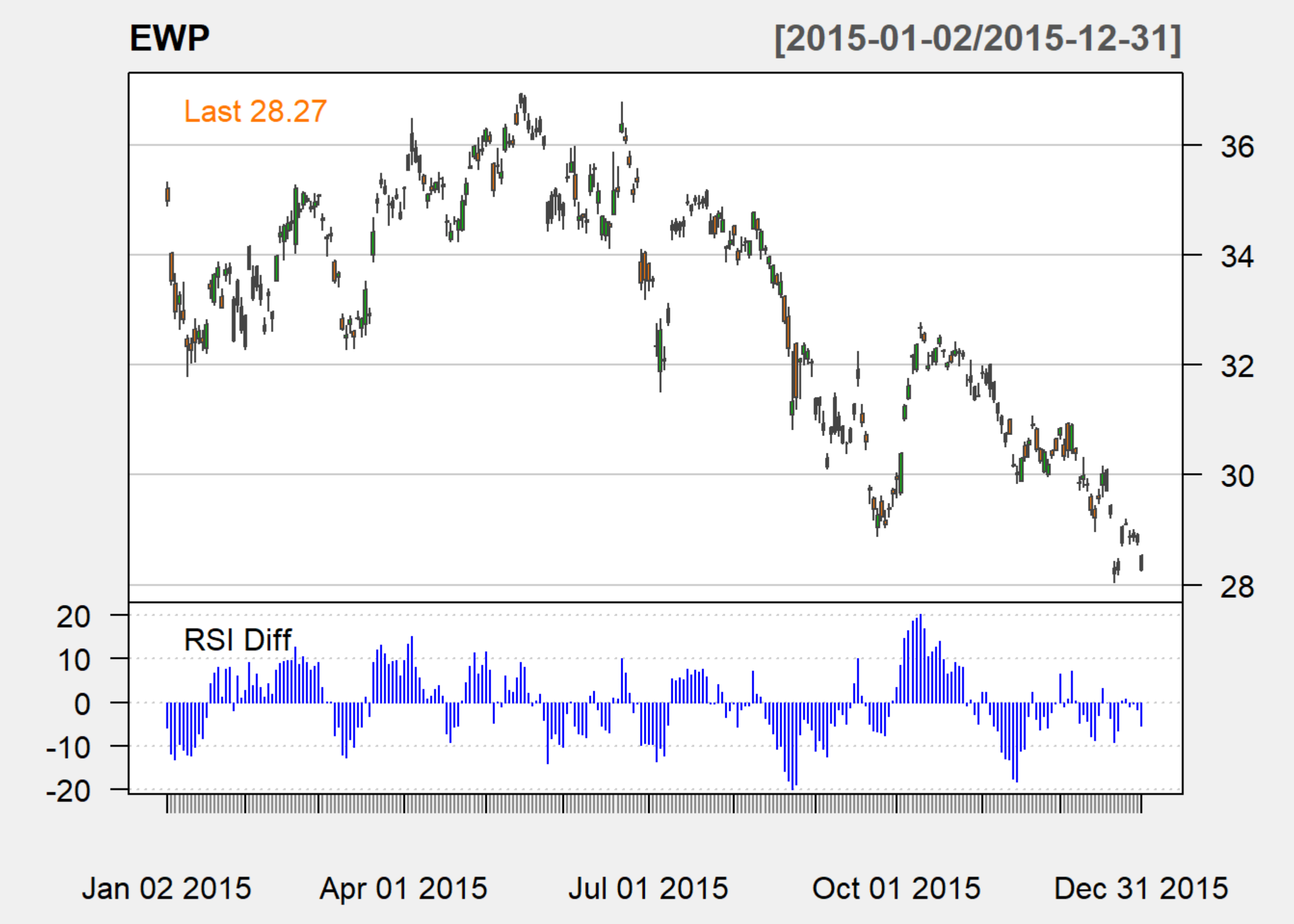
```
#What about shifting further back on the CCI, this ensures that it is a retracement and not a ra
ndom bump?

chartSeries(EWP, theme="white", TA="addRSI(n=100);", subset='2015')
```



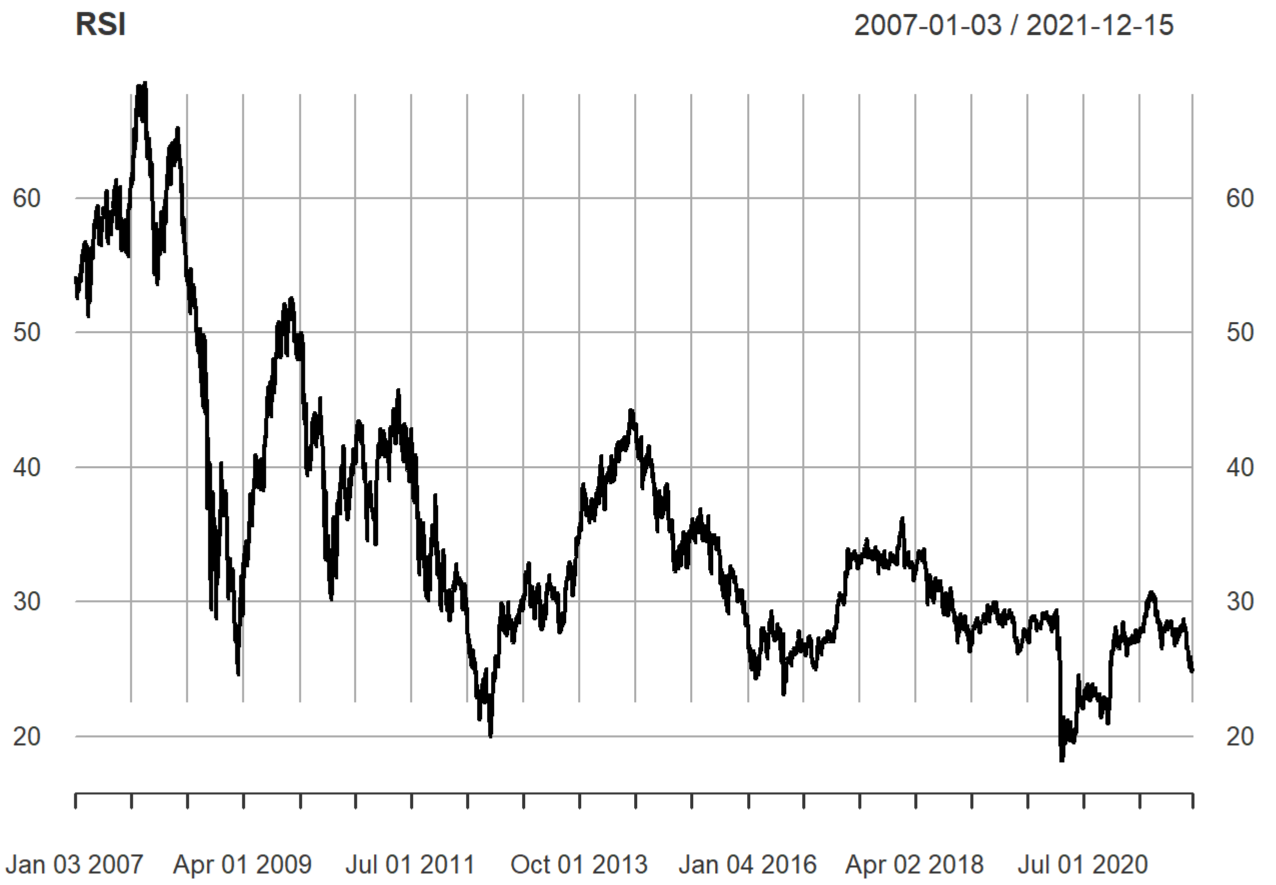
```
chartSeries(EWP, theme="white", TA=NULL, subset='2015')
```



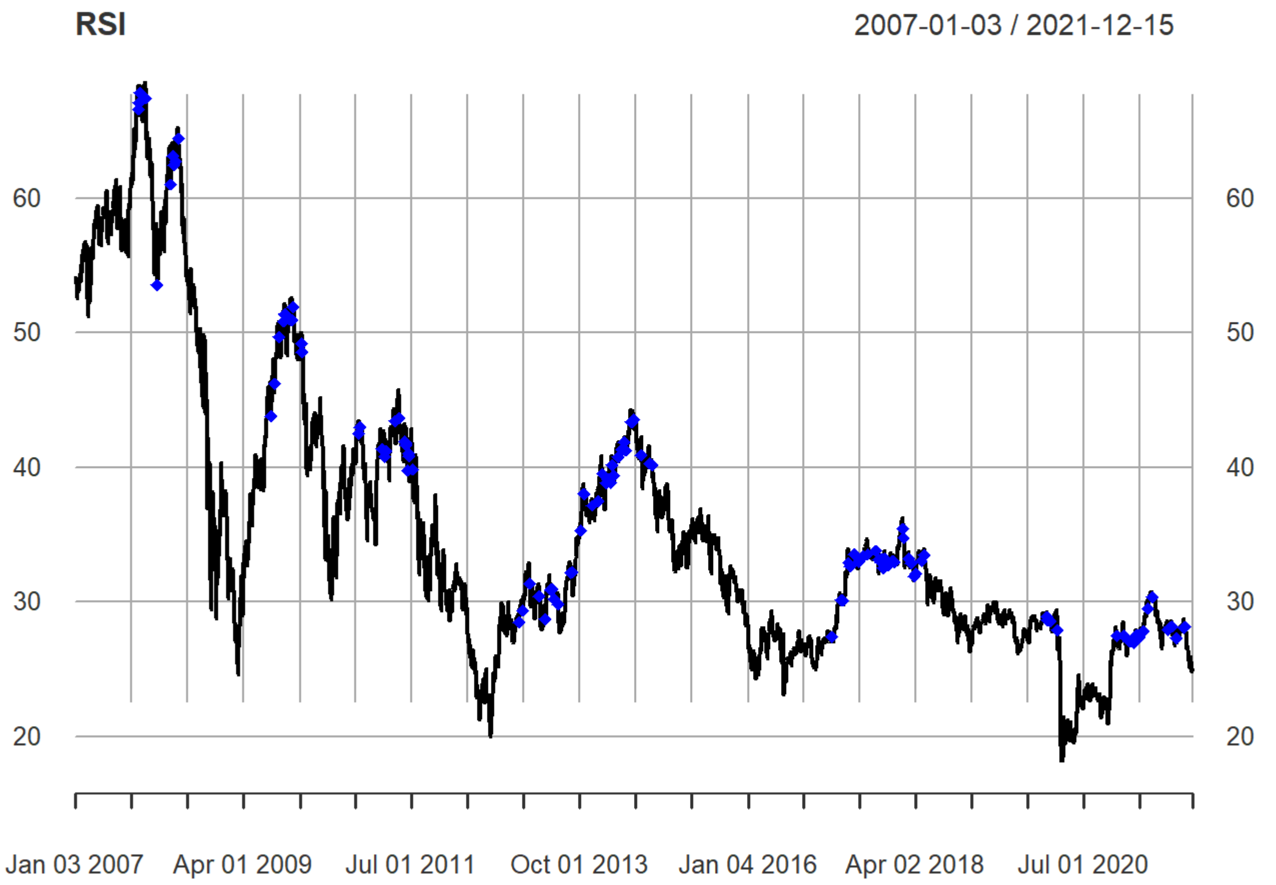
```
RSI.Fast <- RSI(price=EWP$EWP.Close,n=10)
RSI.Slow <- RSI(price=EWP$EWP.Close,n=30)
RSI.Diff <- RSI.Fast-RSI.Slow
addTA(RSI.Diff, col='blue', type='h',legend="RSI Diff")
```



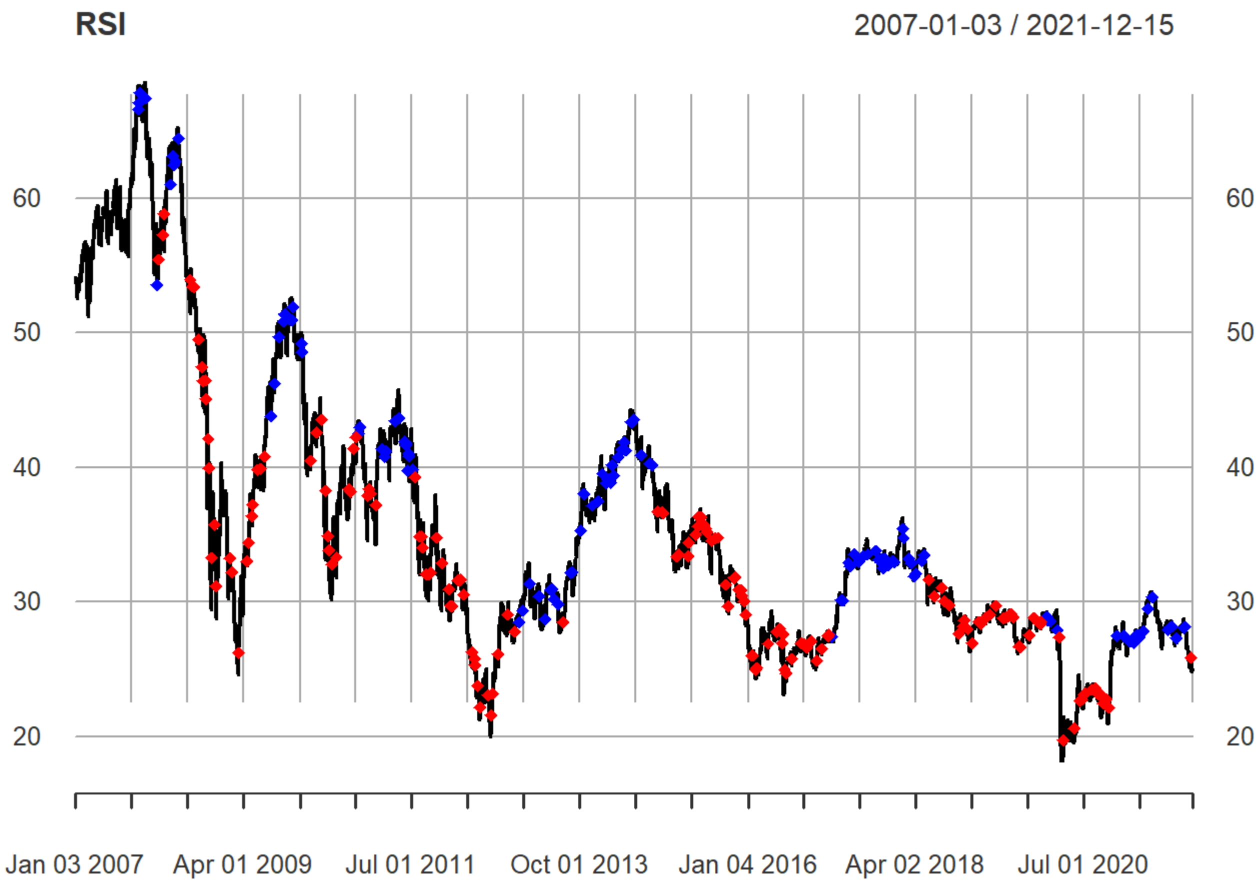
```
# create a slow ema difference
EWP.EMA.50 <- EMA(EWP$EWP.Close, n=50)
EWP.EMA.200 <- EMA(EWP$EWP.Close, n=200)
Slow.Diff <- EWP.EMA.50 - EWP.EMA.200

RSI.IND <- RSI(price=EWP$EWP.Close,n=30)

# look for long entries
Long_Trades <- ifelse(
  RSI.Diff  < 0 &
    shift(v=as.numeric(RSI.Diff ), places=1, dir="right") > 0  &
    Slow.Diff > 0, EWP$EWP.Close, NA)

# look for short entries
Short_Trades <- ifelse(
  RSI.Diff  > 0 &
    shift(v=as.numeric(RSI.Diff ), places=1, dir="right") < 0  &
    Slow.Diff < 0, EWP$EWP.Close, NA)

plot(EWP$EWP.Close, main='RSI')
```



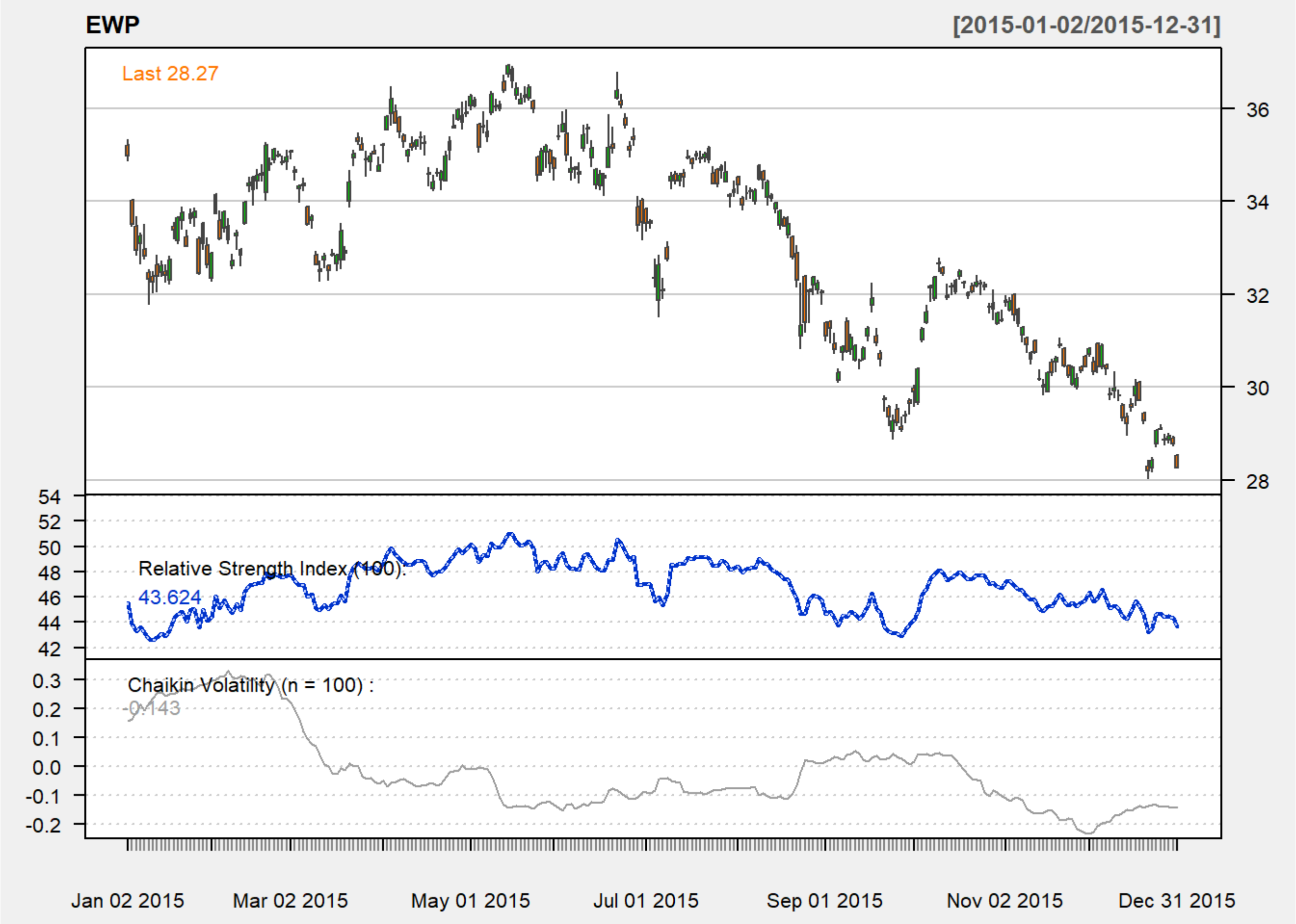
```
## Warning in plot.xts(EWP, main = "RSI"): only the univariate series will be
## plotted
points(Long_Trades, col='blue', cex=1, pch=18)
```



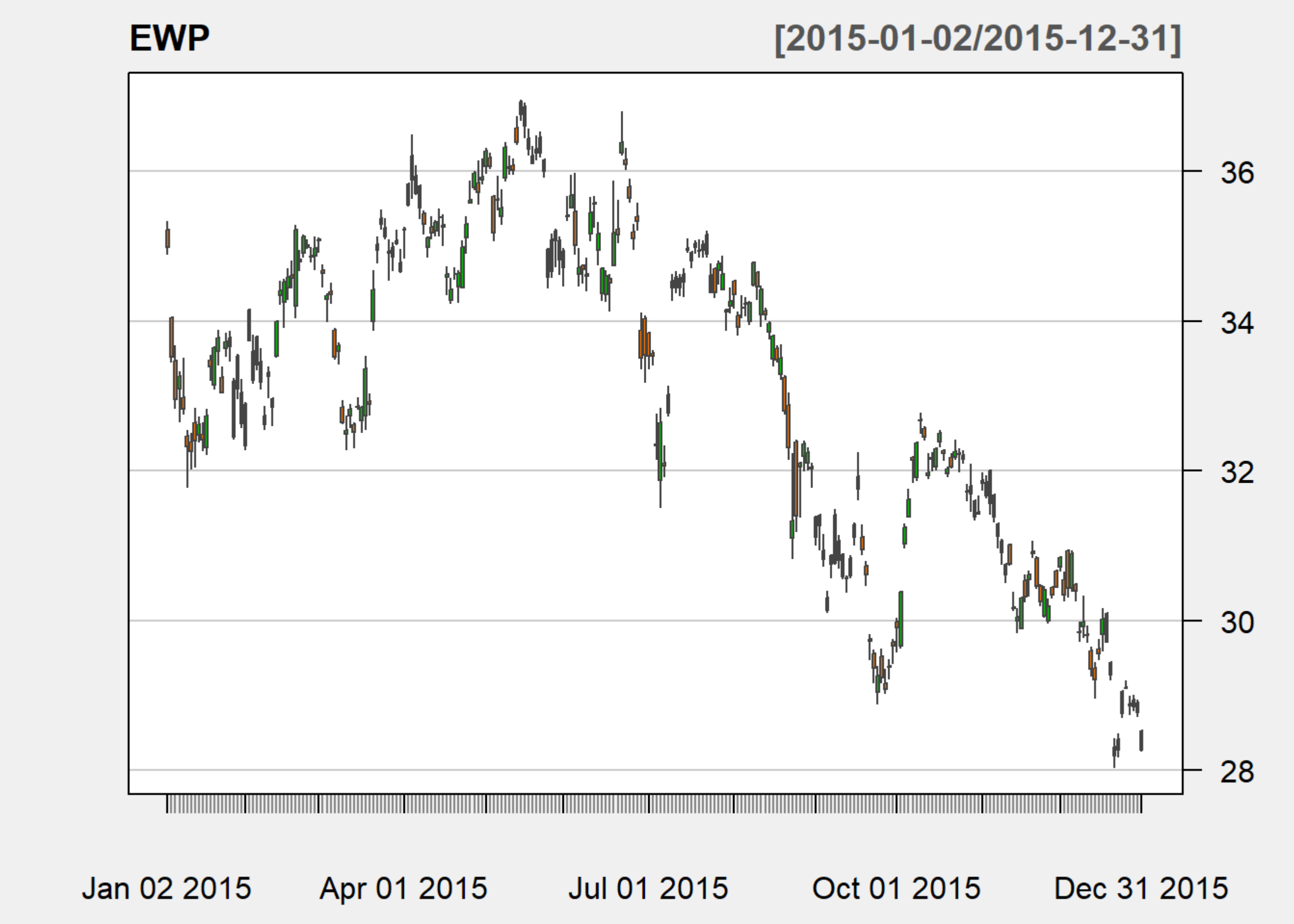
```
points(Short_Trades, col='red', cex=1, pch=18)
```



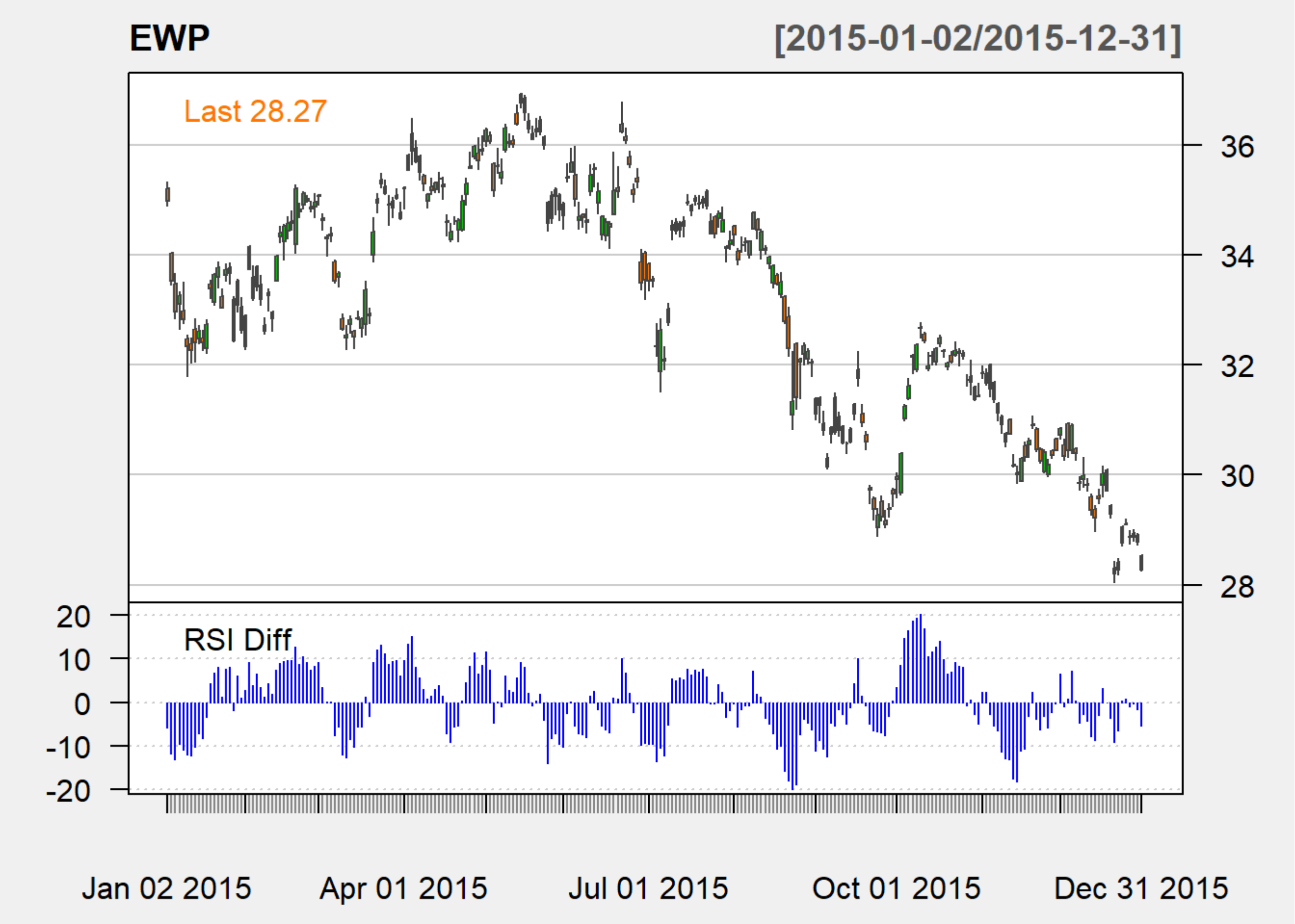
```
#Lets see if we can improve this by adding the Chaikin Volatility to the RSI like we did earlier
with the CCI counter-trading system.

chartSeries(EWP, theme="white", TA="addRSI(n=100);addChVol(n=100);", subset='2015')
```



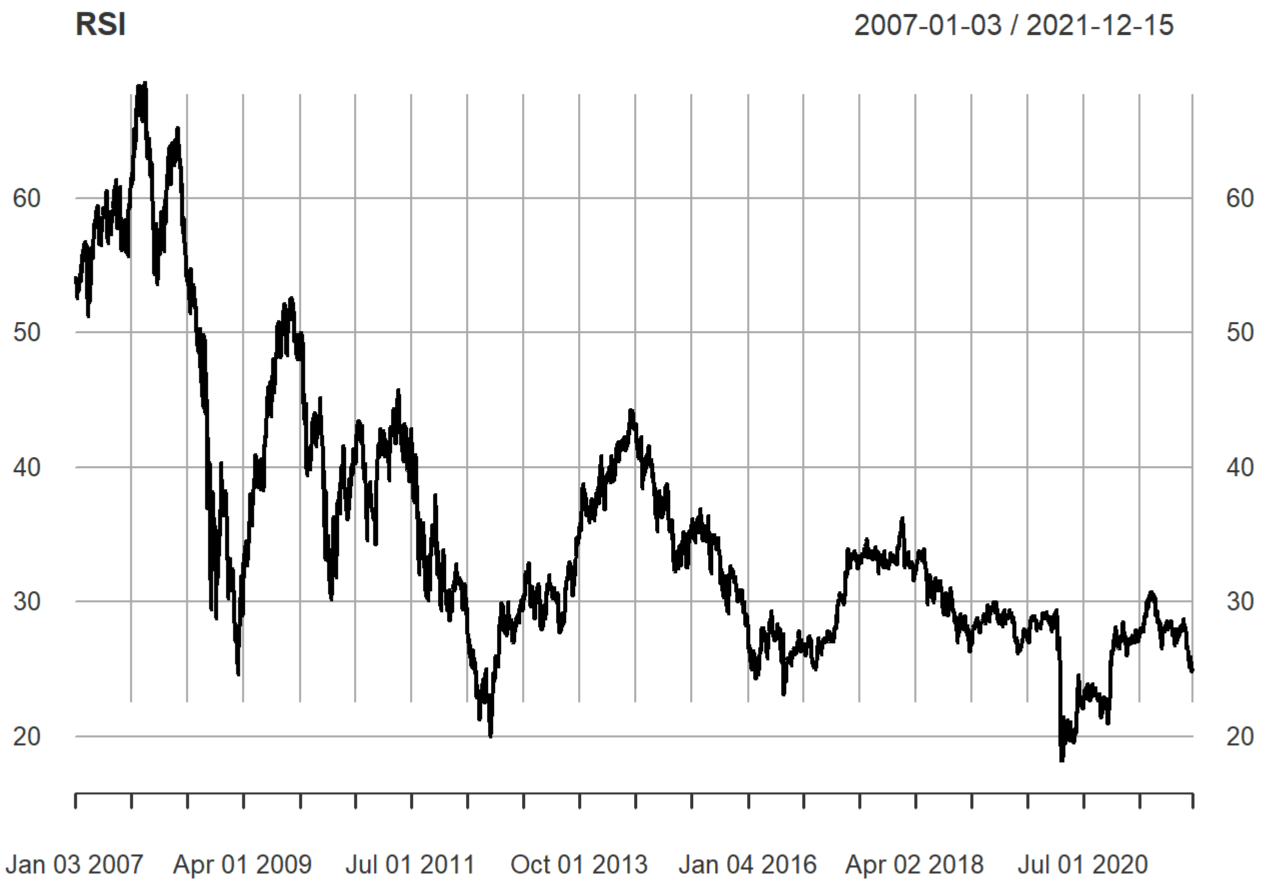
```
chartSeries(EWP, theme="white", TA=NULL, subset='2015')
```



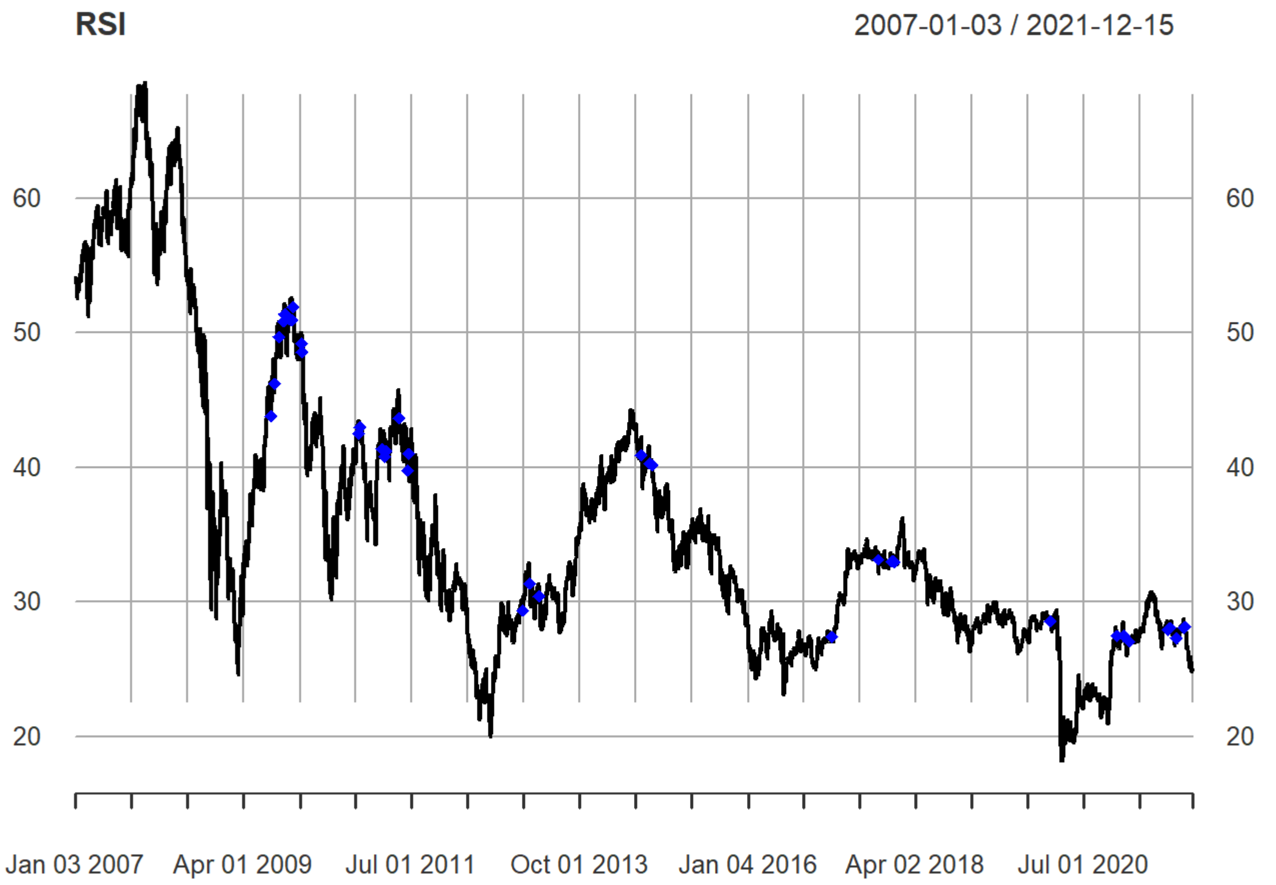
```
RSI.Fast <- RSI(price=EWP$EWP.Close,n=10)
RSI.Slow <- RSI(price=EWP$EWP.Close,n=30)
RSI.Diff <- RSI.Fast-RSI.Slow
addTA(RSI.Diff, col='blue', type='h',legend="RSI Diff")
```



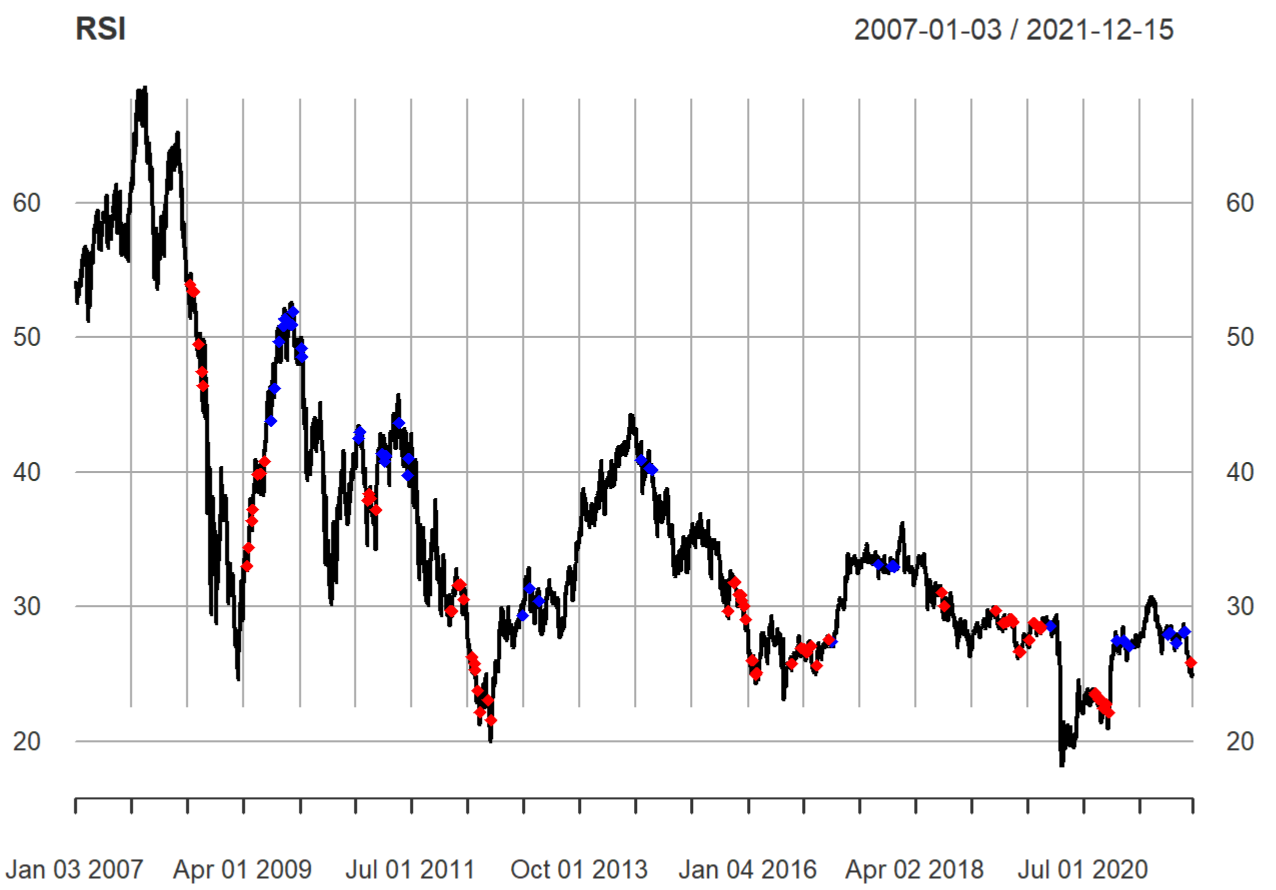
```
# create a slow ema difference
EWP.EMA.50 <- EMA(EWP$EWP.Close, n=50)
EWP.EMA.200 <- EMA(EWP$EWP.Close, n=200)
Slow.Diff <- EWP.EMA.50 - EWP.EMA.200
CV.IND <- chaikinVolatility(HL=EWP, n=100)
RSI.IND <- RSI(price=EWP$EWP.Close,n=30)

# look for long entries
Long_Trades <- ifelse(
  RSI.Diff  < 0 &
    shift(v=as.numeric(RSI.Diff ), places=1, dir="right") > 0  &
    CV.IND < -0.1 &
    Slow.Diff > 0, EWP$EWP.Close, NA)

# look for short entries
Short_Trades <- ifelse(
  RSI.Diff  > 0 &
    shift(v=as.numeric(RSI.Diff ), places=1, dir="right") < 0  &
    CV.IND < -0.1 &
    Slow.Diff < 0, EWP$EWP.Close, NA)

plot(EWP$EWP.Close, main='RSI')
```



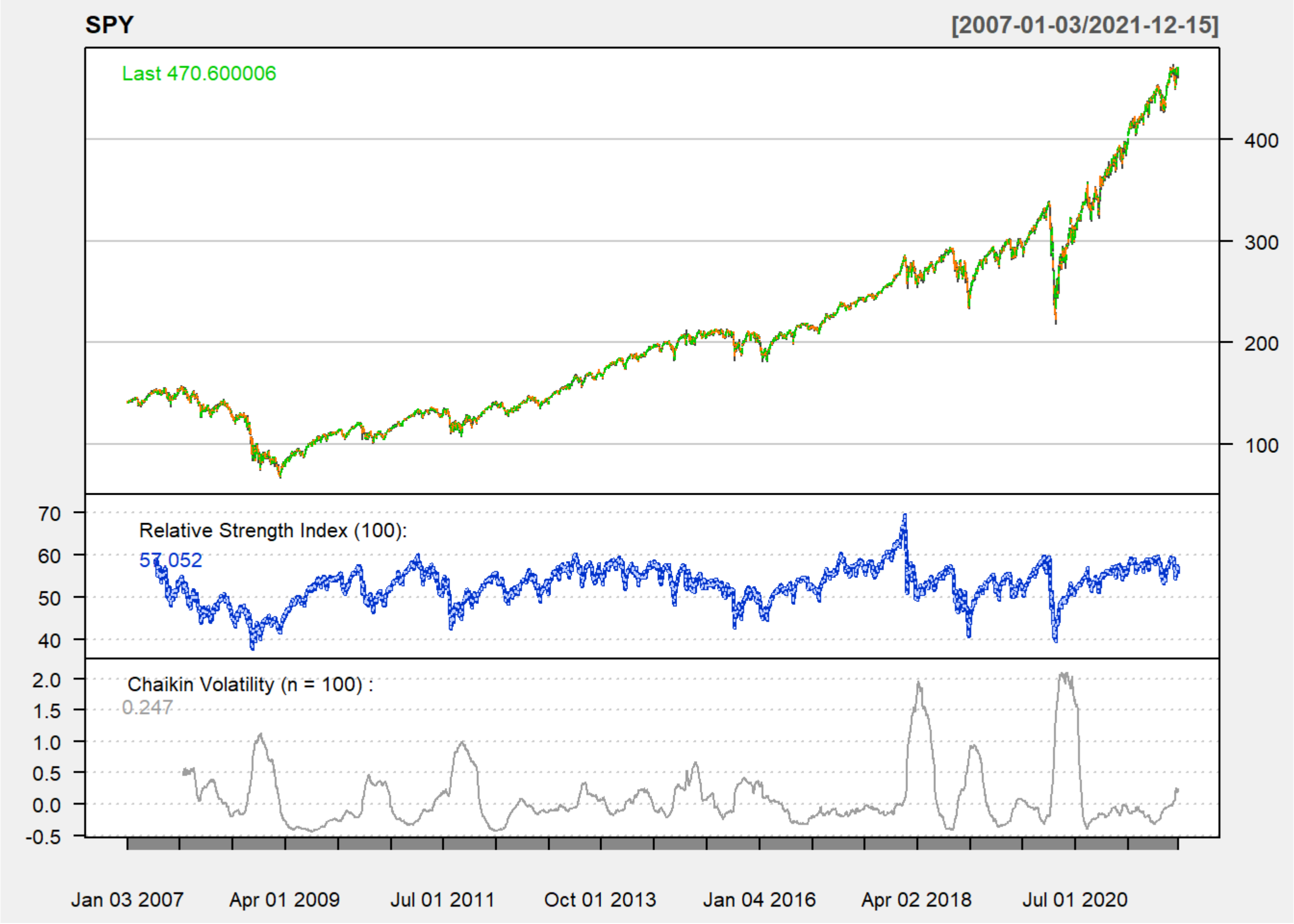
```
## Warning in plot.xts(EWP, main = "RSI"): only the univariate series will be
## plotted
points(Long_Trades, col='blue', cex=1, pch=18)
```



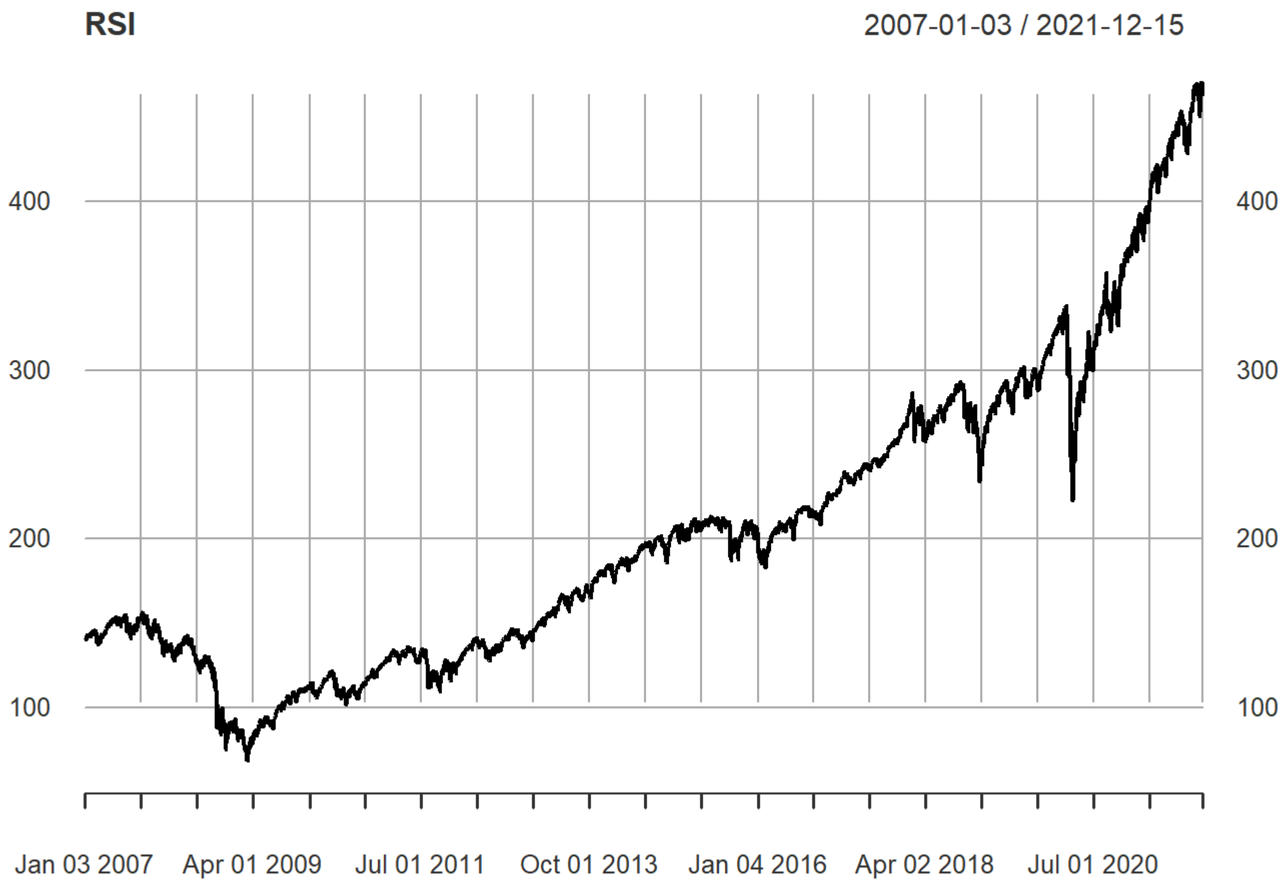
```
points(Short_Trades, col='red', cex=1, pch=18)
```



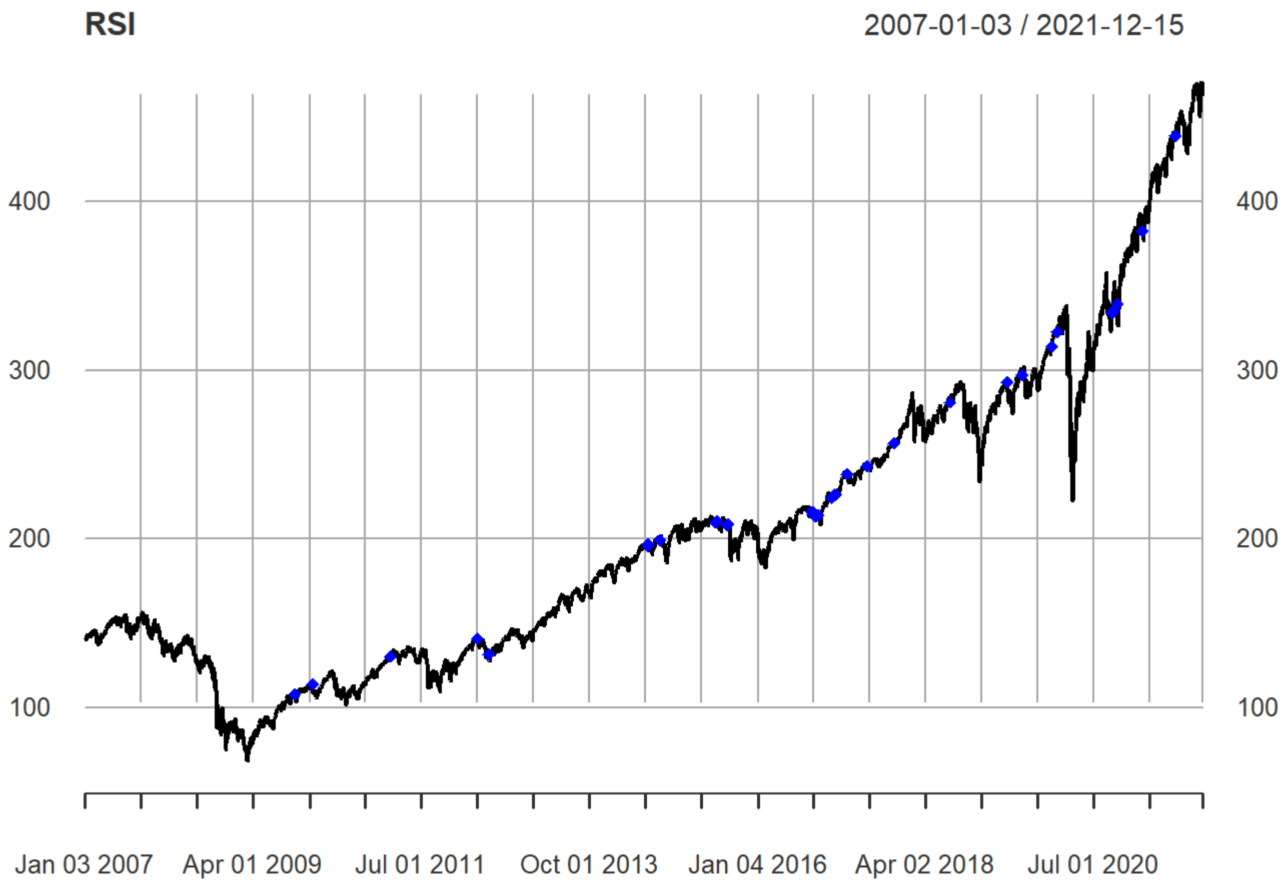
```
#Let?s try this final system on the S&P 500
chartSeries(SPY, theme="white", TA="addRSI(n=100);addChVol(n=100);")
```


SPY
[2007-01-03/2021-12-15]
Last 470.600006
400
300
200
100
70
60
50
40
Relative Strength Index (100):
57.052
2.0
1.5
1.0
0.5
0.0
-0.5
Chaikin Volatility (n = 100) :
0.247
Jan 03 2007
Apr 01 2009
Jul 01 2011
Oct 01 2013
Jan 04 2016
Apr 02 2018
Jul 01 2020

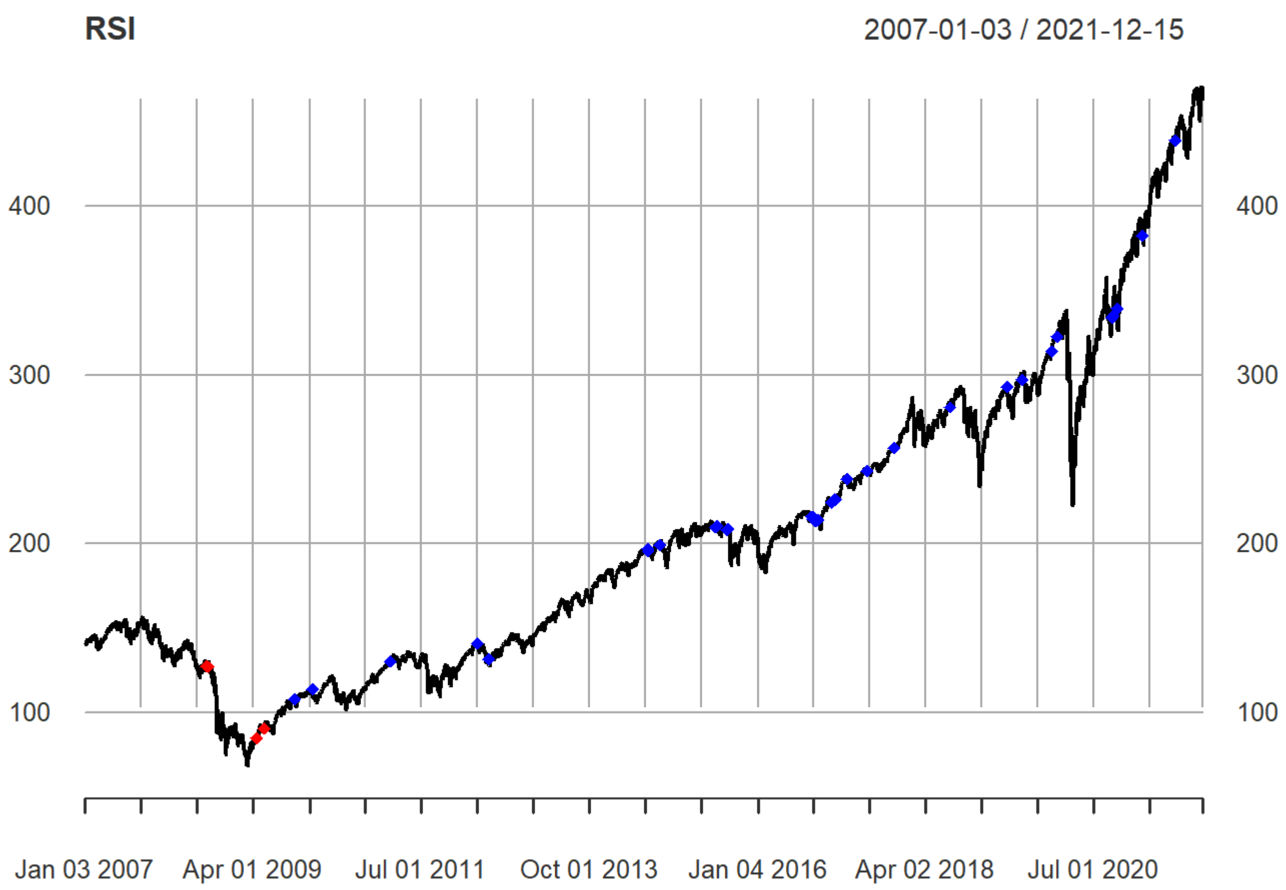
```
# create a slow ema difference
SPY.EMA.50 <- EMA(SPY$SPY.Close, n=50)
SPY.EMA.200 <- EMA(SPY$SPY.Close, n=200)
Slow.Diff <- SPY.EMA.50 - SPY.EMA.200

RSI.Fast <- RSI(price=SPY$SPY.Close,n=10)
RSI.Slow <- RSI(price=SPY$SPY.Close,n=30)
RSI.Diff <- RSI.Fast-RSI.Slow

CV.IND <- chaikinVolatility(HL=SPY, n=100)

# look for long entries
Long_Trades <- ifelse(
  CV.IND < -0.1 &
    RSI.Diff  < 0 &
    shift(v=as.numeric(RSI.Diff ), places=1, dir="right") > 0  &
    shift(v=as.numeric(RSI.Diff ), places=2, dir="right") < 0  &
    Slow.Diff > 0, SPY$SPY.Close, NA)

# look for short entries
Short_Trades <- ifelse(
  CV.IND < -0.1 &
    RSI.Diff  > 0 &
    shift(v=as.numeric(RSI.Diff ), places=1, dir="right") < 0  &
    shift(v=as.numeric(RSI.Diff ), places=2, dir="right") > 0  &
    Slow.Diff < 0, SPY$SPY.Close, NA)

plot(SPY$SPY.Close, main='RSI')
```



```
## Warning in plot.xts(SPY, main = "RSI"): only the univariate series will be
## plotted
points(Long_Trades, col='blue', cex=1, pch=18)
```



```
points(Short_Trades, col='red', cex=1, pch=18)
```



```
# Basket Analysis
#Basket of stocks related to the QQQ

#We'll use a few member stocks of the QQQ Index. This makes things easy for us, but the concepts discussed here can be
#AAPLied to any other financial product and index as long they are related in some way.

#We'll focus on the following tech stocks:

#CSCO, INTC, MSFT, AAPL, TXN. They're fairly related, of similar size, and we can donwload 10+ years of data for each.

print ("STEP 2.8: Basket of stocks related to the QQQ Index")
```

```
## [1] "STEP 2.8: Basket of stocks related to the QQQ Index"
```

```
library(quantmod)
basket_symbols <- c('MSFT', 'INTC', 'AAPL', 'CSCO', 'TXN', 'QQQ')
getSymbols(basket_symbols, src='yahoo')
```

```
## pausing 1 second between requests for more than 5 symbols
## pausing 1 second between requests for more than 5 symbols
## [1] "MSFT" "INTC" "AAPL" "CSCO" "TXN"  "QQQ"
```

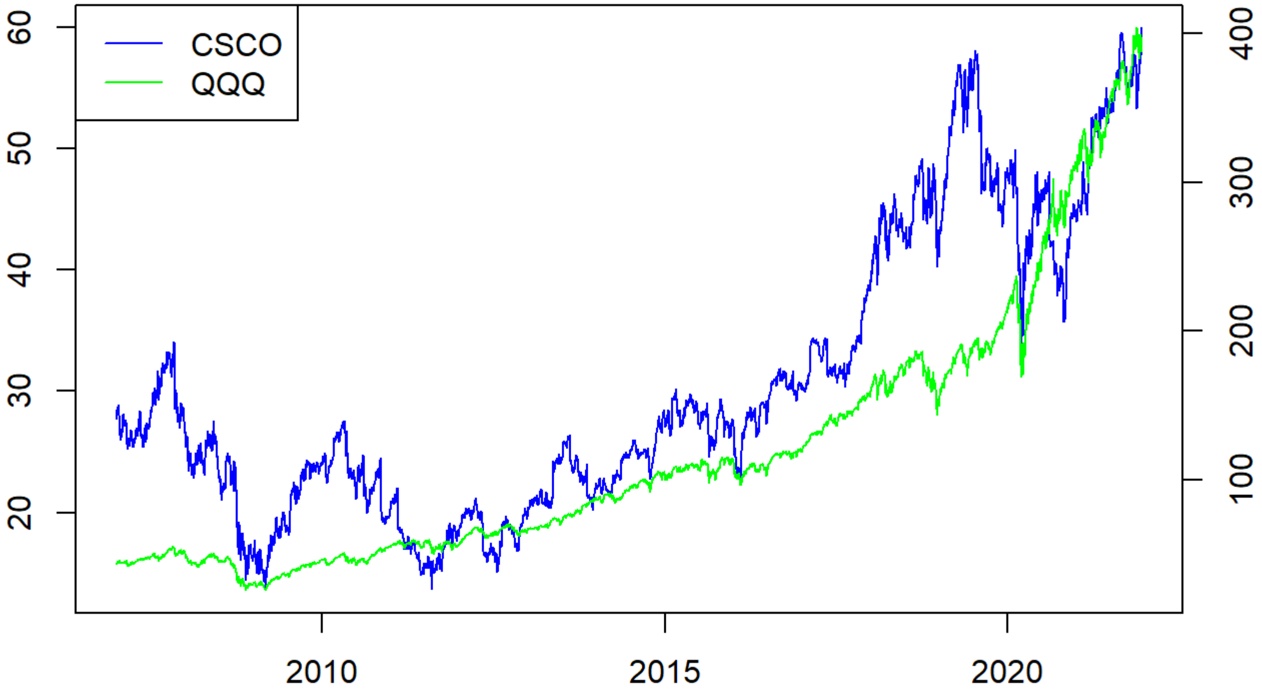
```
#We need to merge all the stocks into one data.frame. We'll use as.xts that converts objects to
 xts class,
#this will merge by time all our columns into one data frame:
basket <- data.frame(as.xts(merge(MSFT, INTC, AAPL, CSCO, TXN, QQQ)))
head(basket,2)
```

```
##            MSFT.Open MSFT.High MSFT.Low MSFT.Close MSFT.Volume MSFT.Adjusted
## 2007-01-03     29.91     30.25    29.40      29.86    76935100      21.82972
## 2007-01-04     29.70     29.97    29.44      29.81    45774500      21.79317
##            INTC.Open INTC.High INTC.Low INTC.Close INTC.Volume INTC.Adjusted
## 2007-01-03     20.45     20.88    20.14      20.35    69001200      13.11656
## 2007-01-04     20.63     21.33    20.56      21.17    88902300      13.64508
##            AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
## 2007-01-03  3.081786  3.092143 2.925000   2.992857  1238319600      2.565971
## 2007-01-04  3.001786  3.069643 2.993571   3.059286   847260400      2.622925
##            CSCO.Open CSCO.High CSCO.Low CSCO.Close CSCO.Volume CSCO.Adjusted
## 2007-01-03     27.46     27.98    27.33      27.73    64226000      20.31001
## 2007-01-04     27.68     28.49    27.54      28.46    73012100      20.84467
##            TXN.Open TXN.High TXN.Low TXN.Close TXN.Volume TXN.Adjusted QQQ.Open
## 2007-01-03    29.12    29.22   28.35     28.56   20786800     20.18442    43.46
## 2007-01-04    28.50    29.11   28.41     29.10   20417600     20.56606    43.30
##            QQQ.High QQQ.Low QQQ.Close QQQ.Volume QQQ.Adjusted
## 2007-01-03    44.06   42.52     43.24  167689500     38.12289
## 2007-01-04    44.21   43.15     44.06  136853500     38.84586
```

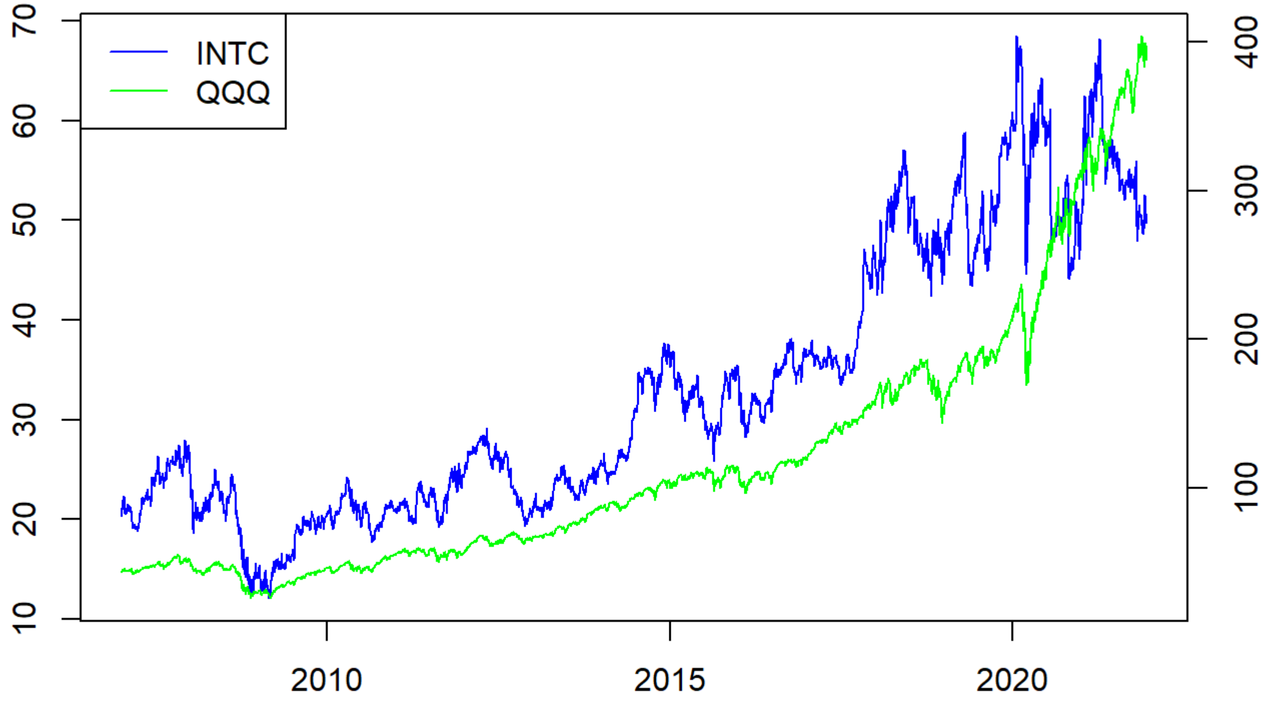
```
#To keep things simple, we'll only keep the Close column for all symbols:
basket <- basket[,names(basket)[grepl(x=names(basket), pattern='Close')]]
head(basket)
```

```
##            MSFT.Close INTC.Close AAPL.Close CSCO.Close TXN.Close QQQ.Close
## 2007-01-03      29.86      20.35   2.992857      27.73     28.56     43.24
## 2007-01-04      29.81      21.17   3.059286      28.46     29.10     44.06
## 2007-01-05      29.64      21.10   3.037500      28.47     28.76     43.85
## 2007-01-08      29.93      21.01   3.052500      28.63     28.90     43.88
## 2007-01-09      29.96      21.03   3.306071      28.47     28.84     44.10
## 2007-01-10      29.66      21.52   3.464286      28.68     29.33     44.62
```

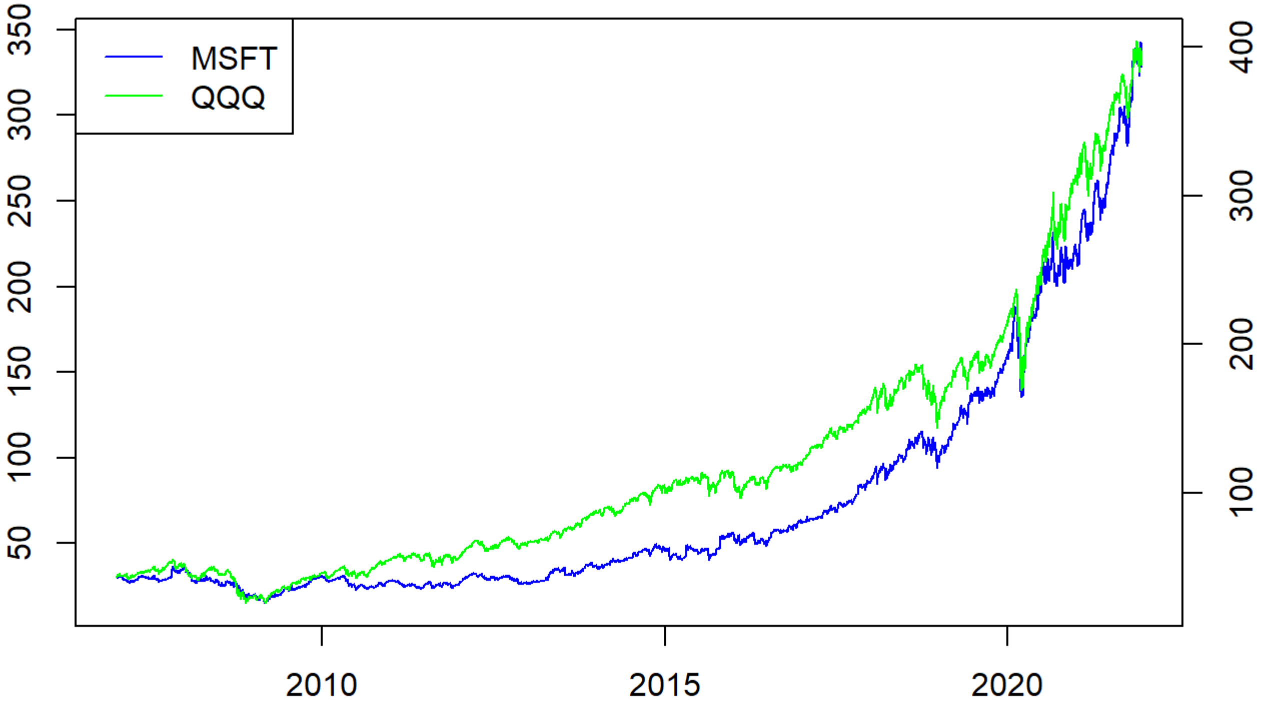
```
#Let's pair every stock with the QQQ in a chart. We'll overlay them together, and, even though t
hey won't share the same price scale,
#it should still give us an idea of how they both move:
plot(as.Date(row.names(basket)), basket$CSCO.Close, col="blue", type='l', ylab="", xlab="")
par(new=TRUE)
plot(as.Date(row.names(basket)), basket$QQQ.Close, col='green', type='l',
     xaxt="n", yaxt='n', xlab="",ylab="")
axis(4)
legend("topleft",col=c("blue","green"),lty=1,legend=c("CSCO","QQQ"))
```



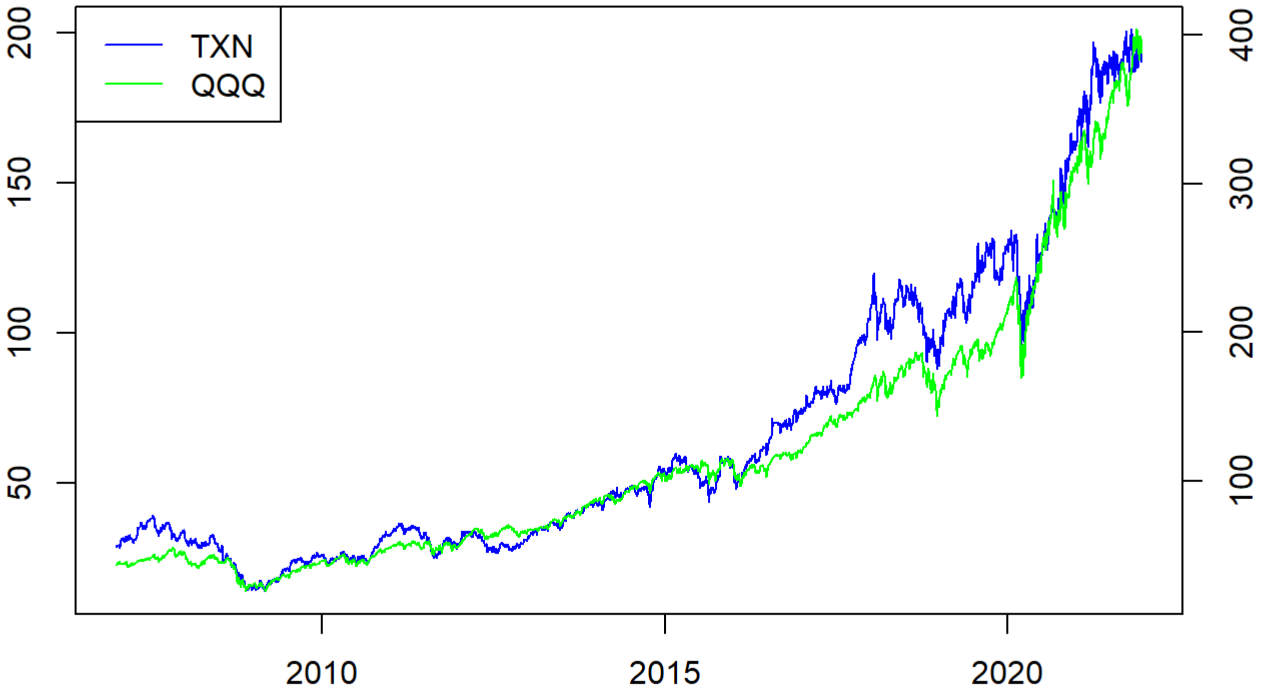
```
plot(as.Date(row.names(basket)), basket$INTC.Close, col="blue", type='l', ylab="", xlab="")
par(new=TRUE)
plot(as.Date(row.names(basket)), basket$QQQ.Close, col='green', type='l',
     xaxt="n", yaxt='n', xlab="",ylab="")
axis(4)
legend("topleft",col=c("blue","green"),lty=1,legend=c("INTC","QQQ"))
```



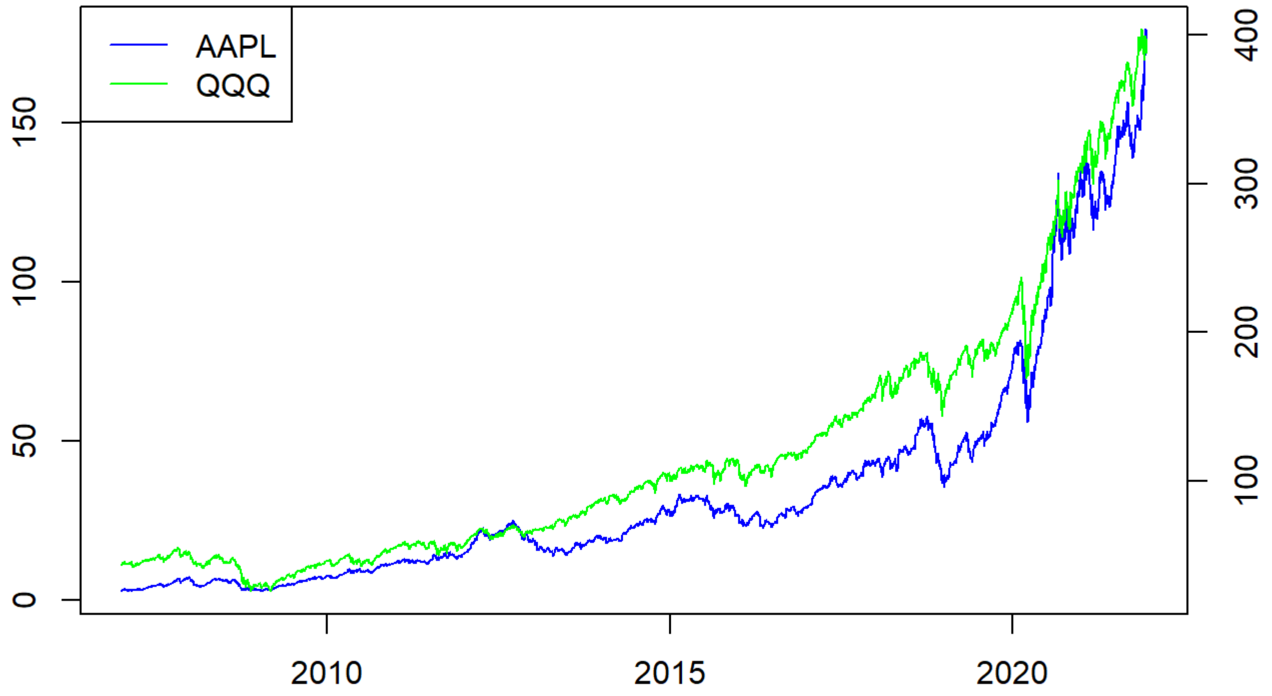
```
plot(as.Date(row.names(basket)), basket$MSFT.Close, col="blue", type='l', ylab="", xlab="")
par(new=TRUE)
plot(as.Date(row.names(basket)), basket$QQQ.Close, col='green', type='l',
     xaxt="n", yaxt='n', xlab="",ylab="")
axis(4)
legend("topleft",col=c("blue","green"),lty=1,legend=c("MSFT","QQQ"))
```



```
plot(as.Date(row.names(basket)), basket$TXN.Close, col="blue", type='l', ylab="", xlab="")
par(new=TRUE)
plot(as.Date(row.names(basket)), basket$QQQ.Close, col='green', type='l',
     xaxt="n", yaxt='n', xlab="",ylab="")
axis(4)
legend("topleft",col=c("blue","green"),lty=1,legend=c("TXN","QQQ"))
```



```
plot(as.Date(row.names(basket)), basket$AAPL.Close, col='blue', type='l', ylab="", xlab="")
par(new=TRUE)
plot(as.Date(row.names(basket)), basket$QQQ.Close, col='green', type='l',
     xaxt="n", yaxt='n', xlab="",ylab="")
axis(4)
legend("topleft",col=c("blue","green"),lty=1,legend=c("AAPL","QQQ"))
```



```
#All the stocks in our basket have followed the QQQ relatively well with the exception of CISCO.
#The point here, is that there may be arbitrage opportunities with stocks that deviate from their group or index but
#it's important to be cautious. Stocks deviate from their peers for a reason and may want to investigate before jumping in -
#whether its just a perception or a serious change.

#Looking at direction

#There is a handy function in quantmod called OHLC.Transformations.
#This allows you to quickly tranform and compare time-series data.
#We'll use the ClCl function that will calculate the difference between the current and previous close.
#We will use the difference between closes to determine if it is an up or down day bar
#(if yesterday's close is lower than today's, then its an up day).

movement_MSFT <- ifelse(ClCl(MSFT)[-1] > 0, 1, -1)
movement_QQQ <- ifelse(ClCl(QQQ)[-1] > 0, 1, -1)
# use a table to see what matched and what didn't
table(movement_MSFT, movement_QQQ)
```

```
##              movement_QQQ
## movement_MSFT   -1    1
##            -1 1310  507
##            1   358 1590
```

```
# Or a simpler way:
sum(movement_MSFT == movement_QQQ) / length(movement_QQQ)
```

```
## [1] 0.7702523
```

```
#The resulting table matrix tells us that out of the 2167 trading days recorded,
#they both had the same down days 762 times and the same up days 843 times. They basically were
 in sync 74% of the time.

#Let's compare our other symbols:

movement_INTC <- ifelse(ClCl(INTC)[-1] > 0, 1, -1)
sum(movement_INTC[-1] == movement_QQQ) / length(movement_QQQ)
```

```
## [1] 0.7373174
```

```
movement_AAPL <- ifelse(ClCl(AAPL)[-1] > 0, 1, -1)
sum(movement_AAPL[-1] == movement_QQQ[-1]) / length(movement_QQQ)
```

```
## [1] 0.764409
```

```
movement_CSCO <- ifelse(ClCl(CSCO)[-1] > 0, 1, -1)
sum(movement_CSCO == movement_QQQ[-1]) / length(movement_QQQ)
```

```
## [1] 0.736255
```

```
movement_TXN <- ifelse(ClCl(TXN)[-1] > 0, 1, -1)
sum(movement_TXN == movement_QQQ[-1]) / length(movement_QQQ)
```

```
## [1] 0.750332
```

```
print ("STEP 2.9:Basket Analysis * Overall correlation * Time-split correlations")
```

```
## [1] "STEP 2.9:Basket Analysis * Overall correlation * Time-split correlations"
```

```
library(quantmod)
basket_symbols <- c('MSFT', 'INTC', 'AAPL', 'CSCO', 'TXN', 'QQQ')
getSymbols(basket_symbols, src='yahoo')
```

```
## pausing 1 second between requests for more than 5 symbols
## pausing 1 second between requests for more than 5 symbols
```

```
## [1] "MSFT" "INTC" "AAPL" "CSCO" "TXN"  "QQQ"
```

```
basket <- data.frame(as.xts(merge(MSFT, INTC, AAPL, CSCO, TXN, QQQ)))
basket <- basket[,names(basket)[grepl(x=names(basket), pattern='Close')]]

#Overall correlation

#So, how correlated are our stocks in our basket? Let's find out.
#We'll use the base cor function in R. It basically compares two vectors AAPLying covariances an
d standard deviations
# Look at the last column, this shows the QQQ's correlation to each stock:
results <- c()
for (basket_name in names(basket)) {
       result <- round(as.numeric(cor(basket)[,basket_name]),2)
       results <- rbind(results, c(basket_name,result))
}
results <- data.frame(results)
names(results)[-1] <- names(basket)
results
```

```
##           X1 MSFT.Close INTC.Close AAPL.Close CSCO.Close TXN.Close QQQ.Close
## 1 MSFT.Close          1       0.85       0.98       0.86      0.96      0.98
## 2 INTC.Close       0.85          1       0.83       0.91      0.93      0.89
## 3 AAPL.Close       0.98       0.83          1        0.8      0.95      0.98
## 4 CSCO.Close       0.86       0.91        0.8          1      0.92      0.88
## 5  TXN.Close       0.96       0.93       0.95       0.92         1      0.99
## 6  QQQ.Close       0.98       0.89       0.98       0.88      0.99         1
```

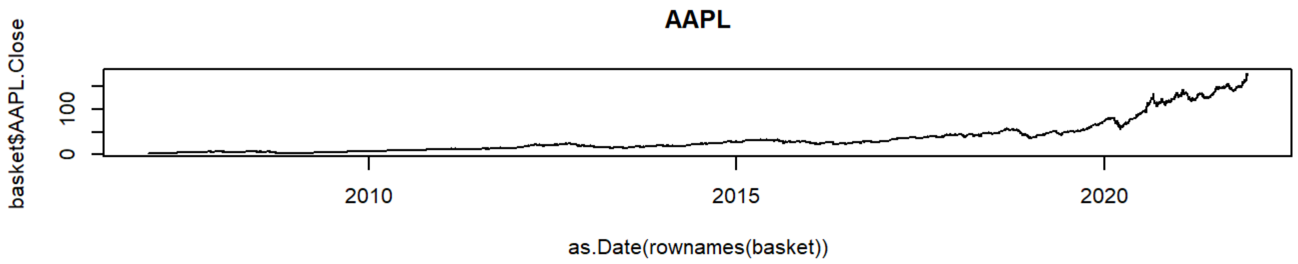
```
#Time-split correlations

#Let's dig deeper and build a function to generelaize the process of getting a correlation tabl
e.
#With this function in hand, we will split the data by time and compare different time periods

# time for a correlation function
Get_Column_Correlations <- function(objDF){
        results <- c()
        for (col_name in names(objDF)) {
               result <- round(as.numeric(cor(objDF)[,col_name]),2)
               results <- rbind(results, c(col_name,result))
        }
        results <- data.frame(results)
        names(results)[-1] <- names(objDF)
        return (results)
}
Get_Column_Correlations(basket[as.Date(rownames(basket)) < '2015-01-01',])[,c('X1','QQQ.Close')]
```

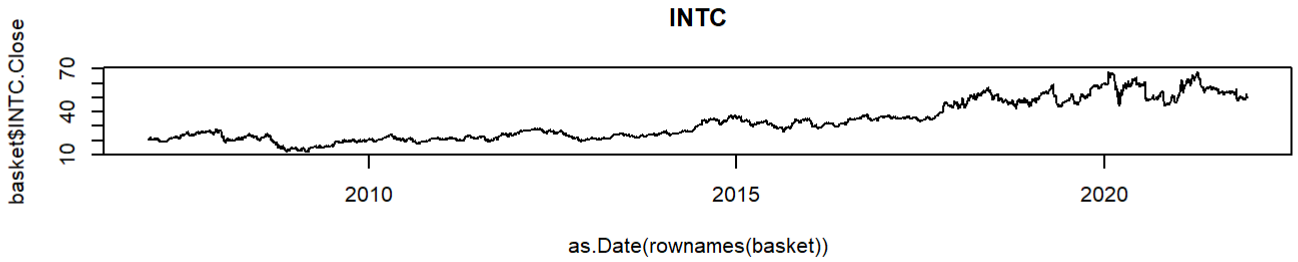
```
##            X1 QQQ.Close
## 1 MSFT.Close      0.88
## 2 INTC.Close      0.82
## 3 AAPL.Close      0.91
## 4 CSCO.Close      0.15
## 5  TXN.Close      0.91
## 6  QQQ.Close         1
```

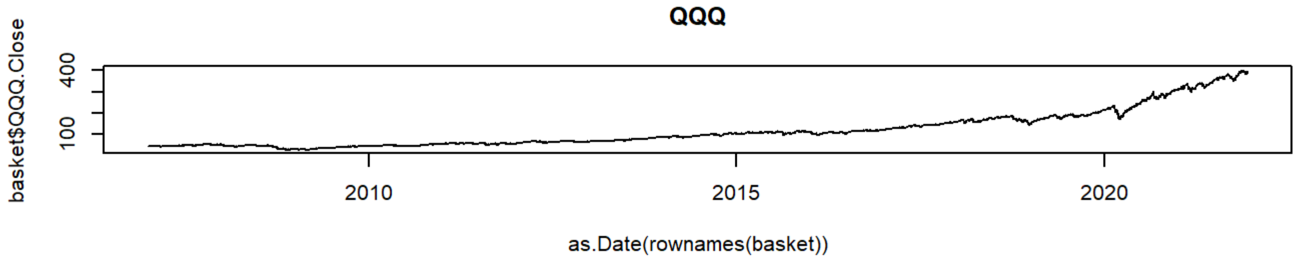
```
Get_Column_Correlations(basket[as.Date(rownames(basket)) >= '2015-01-01',])[,c('X1','QQQ.Close'
)]
```

```
##            X1 QQQ.Close
## 1 MSFT.Close      0.99
## 2 INTC.Close      0.75
## 3 AAPL.Close      0.99
## 4 CSCO.Close      0.78
## 5  TXN.Close      0.97
## 6  QQQ.Close         1
```

```
par(mfrow=c(3,1))
plot(as.Date(rownames(basket)), basket$AAPL.Close, type='l', col='black', main='AAPL')
plot(as.Date(rownames(basket)), basket$INTC.Close, type='l', col='black', main='INTC')
plot(as.Date(rownames(basket)), basket$QQQ.Close, type='l', col='black', main='QQQ')
```



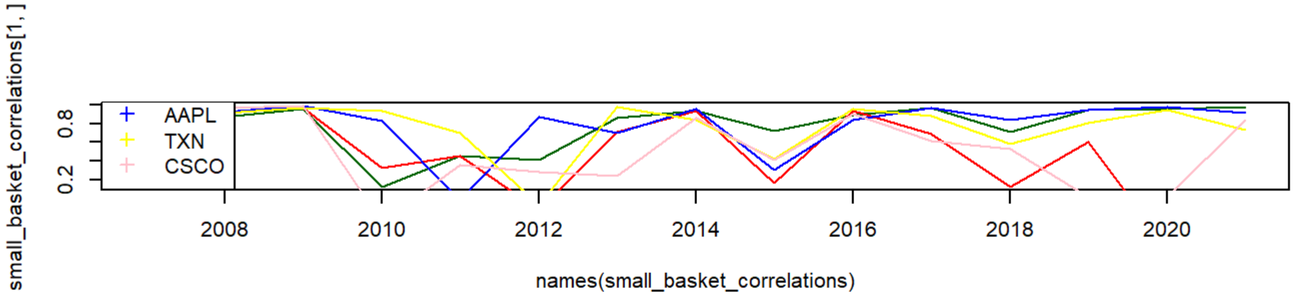


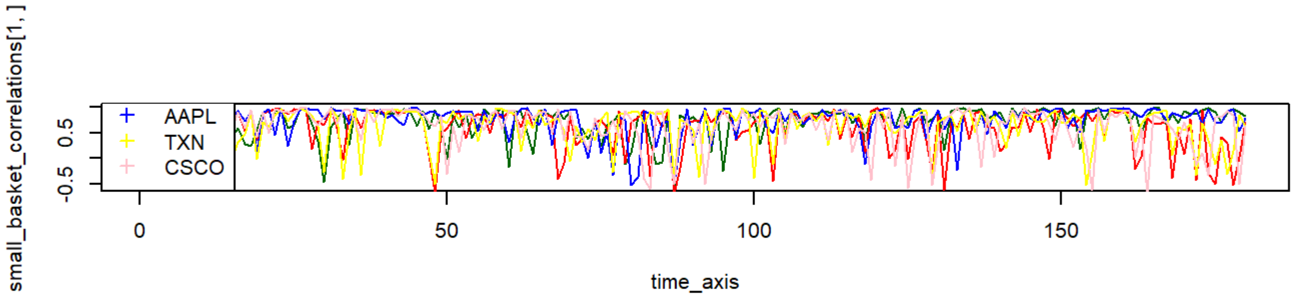


```r
#Let's look at all of these by year and analyze correlations with the QQQ:
basket_years <- unique(substr(rownames(basket), start=1, stop=4))
small_basket <- basket
MSFT_QQQ <- c()
INTC_QQQ <- c()
AAPL_QQQ <- c()
TXN_QQQ <- c()
CSCO_QQQ <- c()
for (year in basket_years) {
        print(year)
        temp_df <- small_basket[substr(rownames(basket), start=1, stop=4)==year,]
        MSFT_QQQ <- cbind(MSFT_QQQ, cor(temp_df$MSFT.Close, temp_df$QQQ.Close))
        INTC_QQQ <- cbind(INTC_QQQ, cor(temp_df$INTC.Close, temp_df$QQQ.Close))
        AAPL_QQQ <- cbind(AAPL_QQQ, cor(temp_df$AAPL.Close, temp_df$QQQ.Close))
        TXN_QQQ <- cbind(TXN_QQQ, cor(temp_df$TXN.Close, temp_df$QQQ.Close))
        CSCO_QQQ <- cbind(CSCO_QQQ, cor(temp_df$CSCO.Close, temp_df$QQQ.Close))
}
```

```
## [1] "2007"
## [1] "2008"
## [1] "2009"
## [1] "2010"
## [1] "2011"
## [1] "2012"
## [1] "2013"
## [1] "2014"
## [1] "2015"
## [1] "2016"
## [1] "2017"
## [1] "2018"
## [1] "2019"
## [1] "2020"
## [1] "2021"
```

```
small_basket_correlations <- data.frame(rbind(MSFT_QQQ, INTC_QQQ, AAPL_QQQ, TXN_QQQ, CSCO_QQQ))
colnames(small_basket_correlations) <- basket_years
plot(names(small_basket_correlations), small_basket_correlations[1,], type='l', col='darkgreen')
lines(names(small_basket_correlations), small_basket_correlations[2,], type='l', col='red')
lines(names(small_basket_correlations), small_basket_correlations[3,], type='l', col='blue')
lines(names(small_basket_correlations), small_basket_correlations[4,], type='l', col='yellow')
lines(names(small_basket_correlations), small_basket_correlations[5,], type='l', col='pink')
legend(x='bottomleft', legend=c("MSFT", "INTC", "AAPL", "TXN", "CSCO"), col=c("darkgreen","red",
"blue", "yellow", "pink"), lwd=1, lty=c(0,0),
        pch=c(3,3))


#This is very revealing how the correlation of both stocks with the index waxes and wanes. Let's
visualize these results.

basket_months <- unique(substr(rownames(basket), start=1, stop=7))
small_basket <- basket #[,names(basket)[grepl(x=names(basket), pattern='MSFT|INTC|QQQ')]]
MSFT_QQQ <- c()
INTC_QQQ <- c()
AAPL_QQQ <- c()
TXN_QQQ <- c()
CSCO_QQQ <- c()
for (yearmonth in basket_months) {
        temp_df <- small_basket[substr(rownames(basket), start=1, stop=7)==yearmonth,]
        MSFT_QQQ <- cbind(MSFT_QQQ, cor(temp_df$MSFT.Close, temp_df$QQQ.Close))
        INTC_QQQ <- cbind(INTC_QQQ, cor(temp_df$INTC.Close, temp_df$QQQ.Close))
        AAPL_QQQ <- cbind(AAPL_QQQ, cor(temp_df$AAPL.Close, temp_df$QQQ.Close))
        TXN_QQQ <- cbind(TXN_QQQ, cor(temp_df$TXN.Close, temp_df$QQQ.Close))
        CSCO_QQQ <- cbind(CSCO_QQQ, cor(temp_df$CSCO.Close, temp_df$QQQ.Close))
}

small_basket_correlations <- data.frame(rbind(MSFT_QQQ, INTC_QQQ, AAPL_QQQ, TXN_QQQ, CSCO_QQQ))
time_axis <- seq(1,ncol(small_basket_correlations))
plot(time_axis, small_basket_correlations[1,], type='l', col='darkgreen')
lines(time_axis, small_basket_correlations[2,], type='l', col='red')
lines(time_axis, small_basket_correlations[3,], type='l', col='blue')
lines(time_axis, small_basket_correlations[4,], type='l', col='yellow')
lines(time_axis, small_basket_correlations[5,], type='l', col='pink')
legend(x='bottomleft', legend=c("MSFT", "INTC", "AAPL", "TXN", "CSCO"), col=c("darkgreen","red",
"blue", "yellow", "pink"), lwd=1, lty=c(0,0),
        pch=c(3,3))
```





```
print ("STEP 2.10:Basket Analysis * AAPLying correlations to entries")
```

```
## [1] "STEP 2.10:Basket Analysis * AAPLying correlations to entries"
```

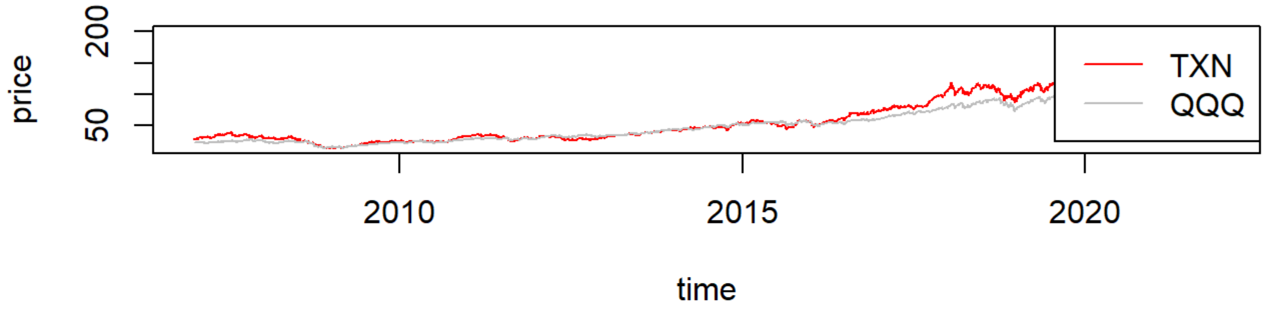
```
 library(quantmod)
 library(binhf)
 basket_symbols <- c('TXN', 'QQQ')
 getSymbols(basket_symbols, src='yahoo')
```

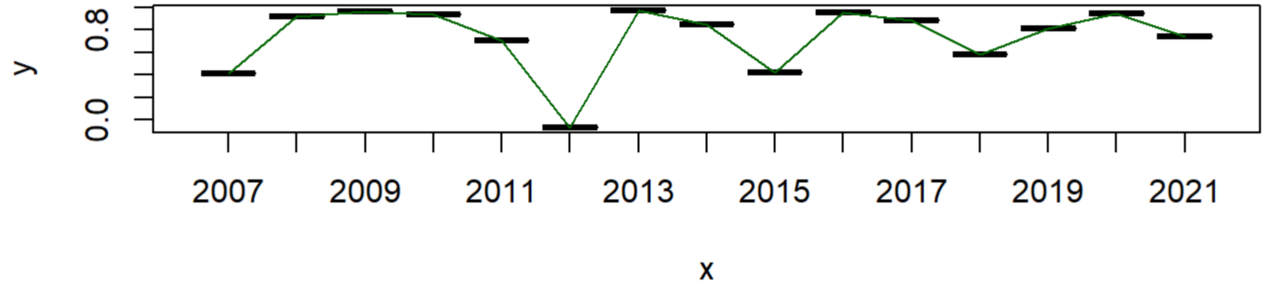
```
## [1] "TXN" "QQQ"
```

```
 basket <- data.frame(as.xts(merge(TXN, QQQ)))
basket <- basket[,names(basket)[grepl(x=names(basket), pattern='Close')]]

#This is a very simplistic arbitrage-type trade.

#So, what if we buy/hold one of these whenever its far from the index?
#So , let's pick a stock that doesn't overly control the index TXN.

getSymbols(c('TXN', 'QQQ'), src='yahoo')
```

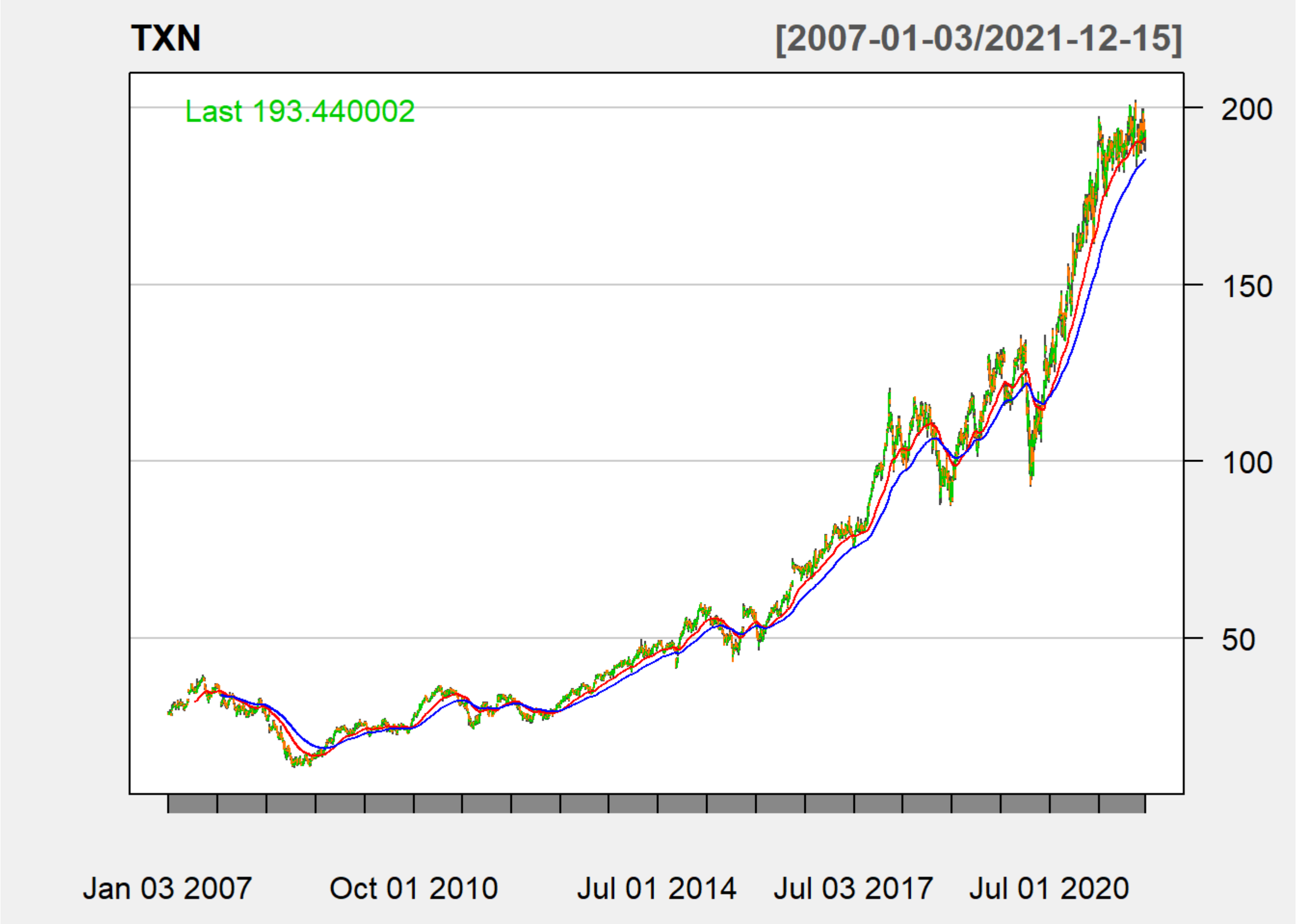
```
## [1] "TXN" "QQQ"
```

```
basket_years <- unique(substr(rownames(basket), start=1, stop=4))
basket_months <- unique(substr(rownames(basket), start=1, stop=7))
small_basket <- basket[,names(basket)[grepl(x=names(basket), pattern='TXN|QQQ')]]
TXN_QQQ <- c()
for (yearmonth in basket_years) {
        temp_df <- small_basket[substr(rownames(basket), start=1, stop=4)==yearmonth,]
        TXN_QQQ <- cbind(TXN_QQQ, cor(temp_df$TXN.Close, temp_df$QQQ.Close))
}

small_basket_correlations <- data.frame(rbind(TXN_QQQ))
colnames(small_basket_correlations) <- basket_years

par(mfrow=c(2,1))
plot(as.Date(row.names(basket)), basket$TXN.Close, col='red',
     type='l', ylab="price", xlab='')
par(new=TRUE)
plot(as.Date(row.names(basket)), basket$QQQ.Close, col='gray', type='l', xaxt="n",yaxt="n",ylab="", xlab='time')
legend("topright",col=c("red","gray"),lty=1,legend=c("TXN","QQQ"))

plot(type='l', col='darkgreen', x=as.factor(names(small_basket_correlations)),  y=as.numeric(small_basket_correlations[1,]))
lines(type='l', col='darkgreen', x=as.factor(names(small_basket_correlations)),
     y=as.numeric(small_basket_correlations[1,]))
```

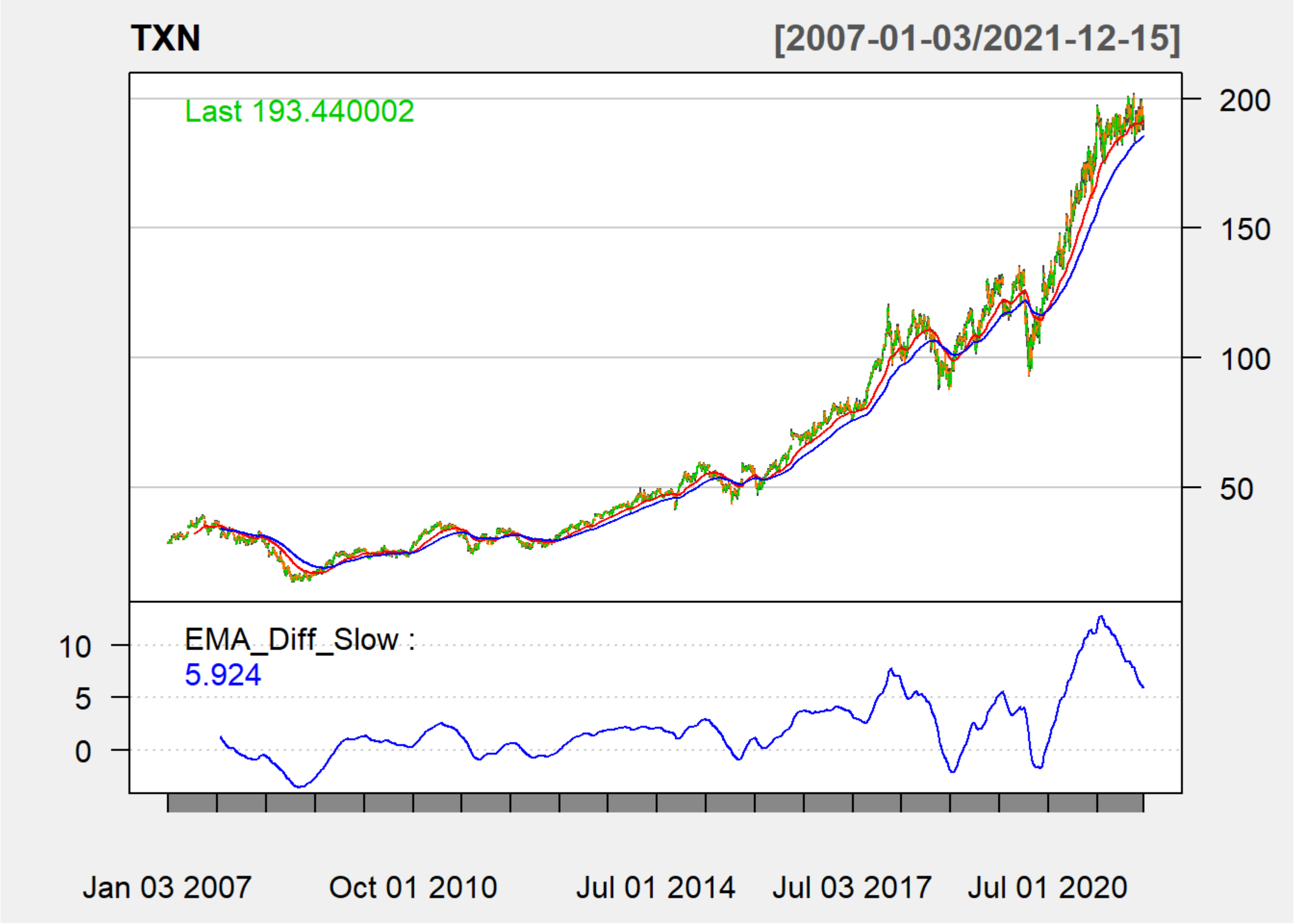




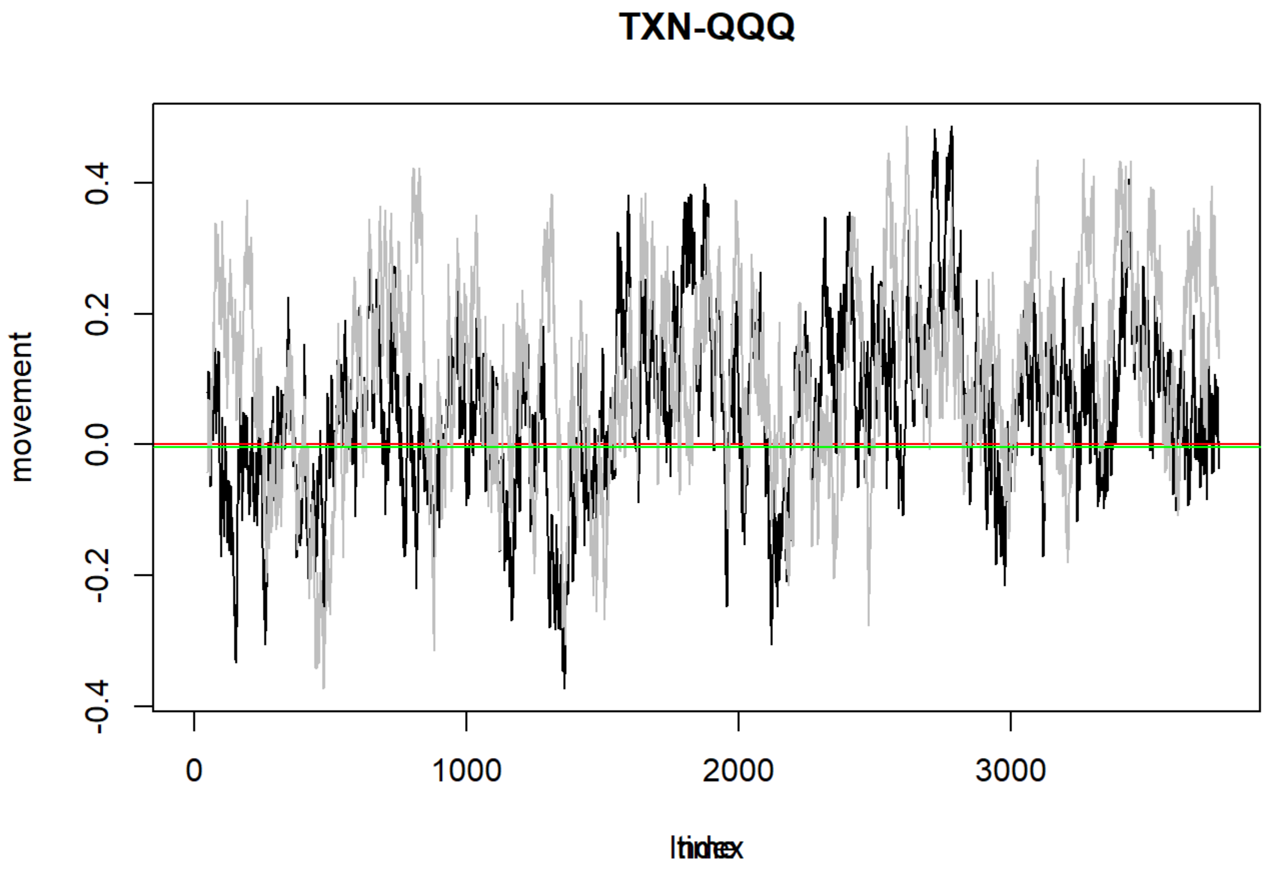
```
#So, let's create moving-average differences like we did in previous lectures to capture trends:

EMA.Fast <- EMA(TXN$TXN.Close, n=30)
EMA.Medium <- EMA(TXN$TXN.Close, n=100)
EMA.Slow <- EMA(TXN$TXN.Close, n=200)
EMA_Diff_Fast <- EMA.Fast - EMA.Medium
EMA_Diff_Slow <- EMA.Medium - EMA.Slow

chartSeries(TXN, theme="white", TA="addEMA(n=100, col='red');addEMA(n=200, col='blue')")
```



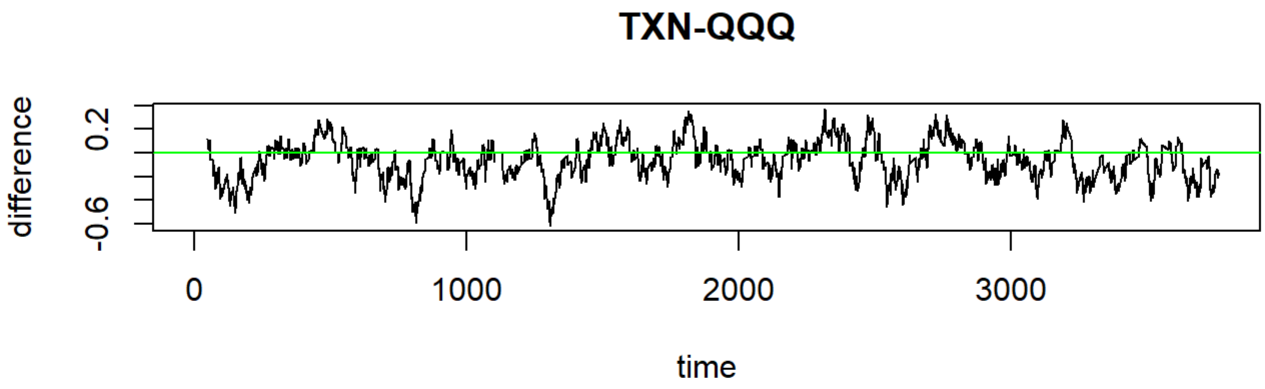
```
addTA(EMA_Diff_Slow, col='blue')
```

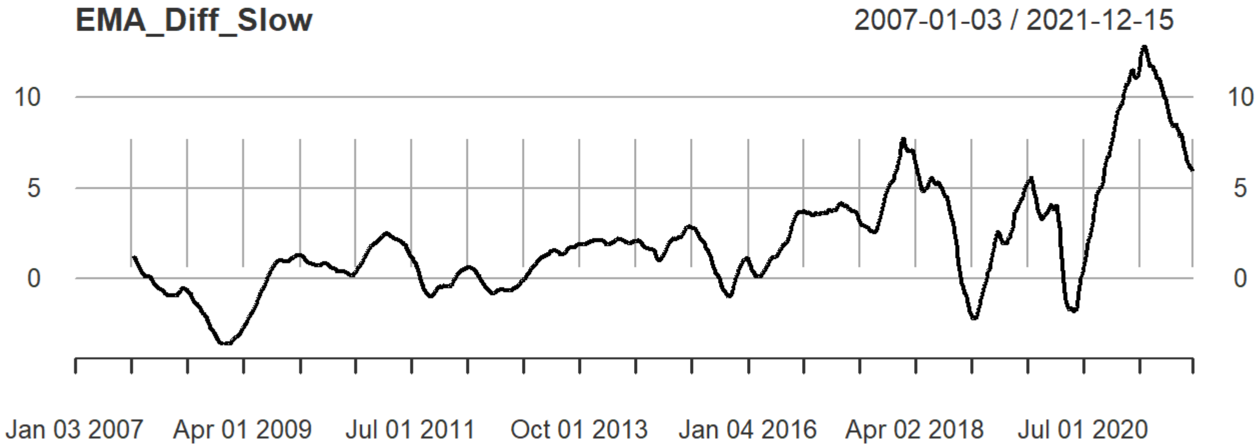


```
QQQ$QQQ.movement <- EMA(ifelse(ClCl(QQQ)  > 0, 1, -1),50)
TXN$TXN.movement <- EMA(ifelse(ClCl(TXN) > 0, 1, -1),50)

plot(as.numeric(TXN$TXN.movement ), col='black', ylab="movement",  main='TXN-QQQ', type = 'l')
abline(h=0, col='red')
par(new=TRUE)
plot(as.numeric(QQQ$QQQ.movement ), col='gray', xaxt="n",yaxt="n",ylab="", xlab='time', type='l'
)
abline(h=0, col='green')
```



```
par(mfrow=c(2,1))
diff <- as.numeric(TXN$TXN.movement)-as.numeric(QQQ$QQQ.movement)
## Warning in as.numeric(TXN$TXN.movement) - as.numeric(QQQ$QQQ.movement):
## longer object length is not a multiple of shorter object length
plot(diff,
     col='black', type='l', xlab='time', ylab='difference',
     main='TXN-QQQ')
abline(h=0, col='green')
plot(EMA_Diff_Slow)
abline(h=0, col='green')
```





```
print ("end of script.")
```

```
## [1] "end of script."
```