Final Project - Linear Regression Curves

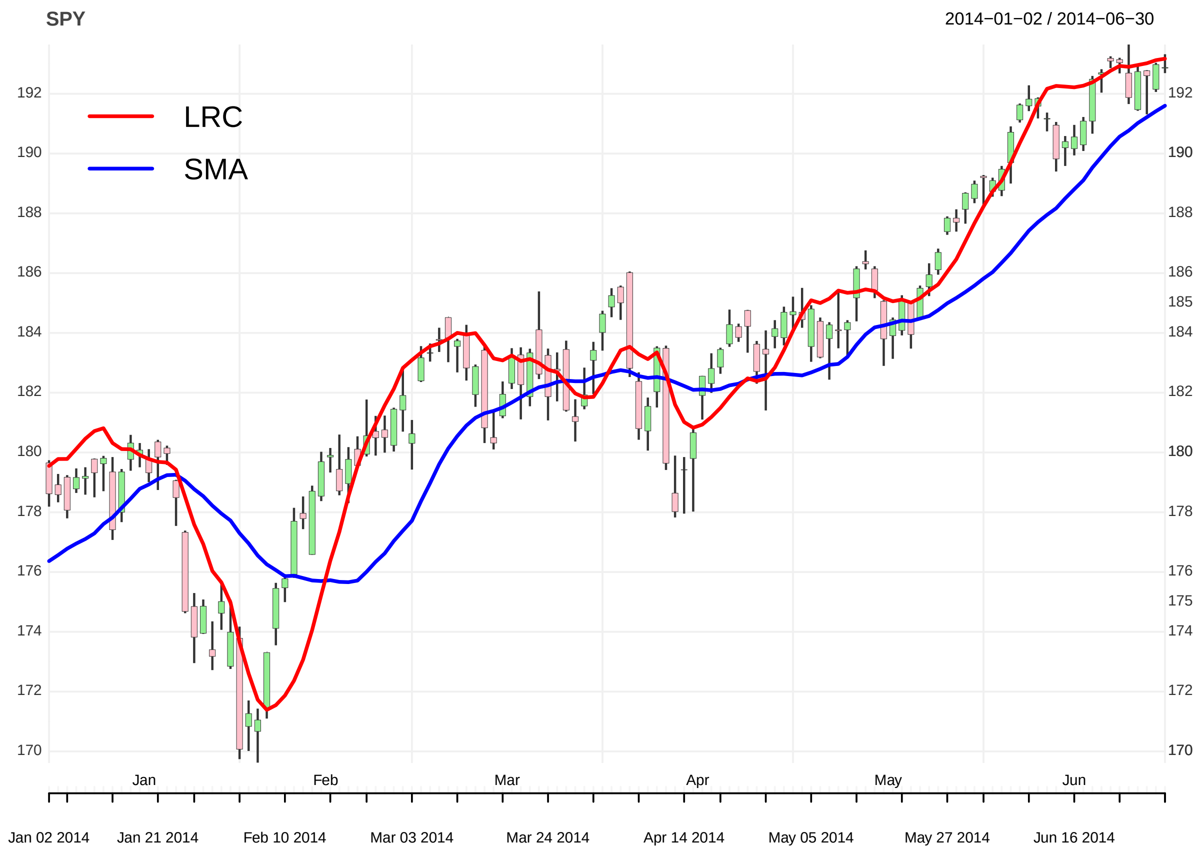
Priyesh Kannan & Michael Sean Sinykin

Tuesday, June 2, 2015

Description

Our project focused upon the tradability of signals generated by linear regression curves (LRC) and there less-sensitive counterparts, the simple moving averages (SMA). To construct a linear regression curve, we choose a period over which to regress, for example n=20. Starting from the beginning of our time series (t=0), we take the first n observations, construct a linear regression line through the data, and define our linear regression curve (LRC) at n=20 as the value of our linear regression line at n=20. We then construct a similar linear regression line starting from t=1 to arrive at the value of our LRC at n=21. Iterating through our time series, we build up our LRC in this way. For more information, see http://gekkoquant.com/2013/09/29/high-probability-credit-spreads-using-linear-regression-curves/.

In constructing the LRC, we compared it to the SMA of the same period, and noticed that the LRC was more responsive to the actual price movement than the SMA. Similar to the above-referenced Gekko Trading post, we decided to exploit this sensitivity by creating a LRC-SMA crossover system, optimizing LRC vs SMA.



For our instrument, we chose SPY, daily data from 01/30/1999 to 06/30/2014. For our training period, we chose 4 years for training and 1 year for testing, conducting WFA through the time series.

Defining Our Trade Rules

We kept our system simple, in order to retain system robustness (generating statistically significant numbers of trades during back tests) and to help avoid curve-fitting traps. Our original plan was to test as a long only as well as long-short system, but found out rather quickly that sequential processing for parameter optimization on our Windows systems can be prohibitively time-consuming, so we settled for a long-only crossover system. Because we found the LRC to be more sensitive to changes in underlying price trends, we set our bullish entries (buy long) to be triggered when the LRC broke above the SMA. Conversely, our sell signal was triggered when the LRC broke down below our SMA.

We also set up stop losses and trailing stops, in addition to profit-taking sells. Optimizing where to take profits was an interesting exercise, as will be reported below, as the robustness was suspect in all but a narrow range. Again, with more time, we would have liked to do further analysis to determine whether taking profits ends up being a net-positive to the system.

Designing and Formatting

We begin with a bit of initialization and historic data-fetching:

```r
rm(list=ls())

library(FinancialInstrument)
library(TTR)
library(blotter)
library(quantstrat)
library(lattice)

.blotter <- new.env()
.strategy <- new.env()
Sys.setenv(TZ="UTC")
STRATEGY<-'LRC'
PORTOLIO.NAME<-'PORTFOLIO.LRC.STOP'
ACCOUNT.NAME<-'AC.LRC.STOP'

try(rm.strat(PORTOLIO.NAME))

symbol<-c('SPY')
currency("USD")
stock(primary_id = symbol,currency = "USD",multiplier = 1)
Sys.setenv(TZ="UTC")

initDate <- '2004-12-31'
startDate <- '2005-01-30'
endDate<- '2014-06-30'

initEq <- 1e6

options("getSymbols.warning4.0"=FALSE)
getSymbols(Symbols = symbol,index.class  = 'POSIXct',
           from =startDate,to = endDate,adjust=TRUE)
```

Next we set up functions to construct our LRC and calculate and account for transaction costs and slippage, in addition to Portfolio, Accounts, and Orders initialization. For transactions costs, we referred to http://www.rinfinance.com/agenda/2013/workshop/Humme+Peterson.pdf, particularly with respect to the fact that transaction costs are borne at exit (if bought long). For our long-only strategy, the takeaway is that our estimate of transaction fees is for a round-trip trade (both the buy to open the position and the sell to close it out).

```r
#-----------------------------------------------------------------------------------------------
# LRC
#-----------------------------------------------------------------------------------------------
LRC <- function(x,n){

  regression <- function(dataBlock){
    fit <-lm(dataBlock~seq(1,length(dataBlock),1))
    return(last(fit$fitted.values))
  }
  return (rollapply(x,width=n,regression,align="right",by.column=FALSE))
}
#-----------------------------------------------------------------------------------------------
# Fees Function
#-----------------------------------------------------------------------------------------------

evalTxnFees<-function(TxnQty, TxnPrice, Symbol,brokerage=0.005,impactCost=0.05)
{
  txnCost<- abs(round(brokerage*(TxnPrice+impactCost)*TxnQty,2))*-1
  return(txnCost)
}
#-----------------------------------------------------------------------------------------------
#  Initialization
#-----------------------------------------------------------------------------------------------

initPortf(name = PORTOLIO.NAME,symbols = symbol,initDate=initDate)
initAcct(name = ACCOUNT.NAME,portfolios = PORTOLIO.NAME,initDate=initDate,initEq=initEq)
initOrders(portfolio = PORTOLIO.NAME,initDate=initDate)
```

Next, taking our trading rules as discussed above, we add our indicators (LRC and SMA, in this case optimized to LRC=20, SMA=30), our signals (LRC crossing above or below SMA), our rules (buy long when LRC crosses above SMA, sell position when LRC crosses below SMA), as well as our stop loss rule, trailing stop rule, and profit taking rule.

```r
strategy(STRATEGY,store=TRUE)

add.indicator(strategy = STRATEGY,name='LRC',
              arguments=list(x=quote(Cl(mktdata)),n=20),
              label='LRC')

add.indicator(strategy = STRATEGY,name='SMA',
              arguments=list(x=quote(Cl(mktdata)),n=30),
              label='SMA')



LONG.ENTRY.SIGNAL<-"LRC_GT_SMA_SIG"
LONG.EXIT.SIGNAL<-"LRC_LT_SMA_SIG"
LONG.ENTRY.RULE<-'L_ENTRY_LRC_SMA_RULE'
LONG.EXIT.RULE<-'L_EXIT_LRC_SMA_RULE'
LONG.ORDERSET.NAME<-'LRCLONGSMA'

add.signal(strategy = STRATEGY,name="sigCrossover",
           arguments=list(columns=c('LRC','SMA'),
                          relationship="gt"),
           label=LONG.ENTRY.SIGNAL)

add.signal(strategy = STRATEGY,name="sigCrossover",
           arguments=list(columns=c('LRC','SMA'),
                          relationship="lt"),
           label=LONG.EXIT.SIGNAL)

add.rule(strategy = STRATEGY,name="ruleSignal",
         arguments=list(sigcol=LONG.ENTRY.SIGNAL,
                        sigval=TRUE,
                        orderqty=100,
                        ordertype="market",
                        TxnFees=0,
                        orderside="long",
                        orderset=LONG.ORDERSET.NAME),
         type="enter",
         label=LONG.ENTRY.RULE)

add.rule(strategy = STRATEGY,name="ruleSignal",
         arguments=list(sigcol=LONG.EXIT.SIGNAL,
                        sigval=TRUE,
                        orderqty='all',
                        ordertype="market",
                        TxnFees="evalTxnFees",
                        orderside="long",
                        orderset=LONG.ORDERSET.NAME),
         type="exit",
         label=LONG.EXIT.RULE)

#-------------------------------------------------------------------------------
# Stop Loss
#------------------------------------------------------------------------------
stopLossPercent<-0.045
LONG.STOPLOSS.RULE<-"LE_LRCSMA_STOPLOSS_EXIT"

add.rule(strategy = STRATEGY,
         name="ruleSignal",
         arguments=list(sigcol=LONG.EXIT.SIGNAL,
                        replace=FALSE,
                        sigval=TRUE,
                        tmult=TRUE,
                        orderqty='all',
                        threshold=quote(stopLossPercent),
                        TxnFees="evalTxnFees",
                        ordertype="stoplimit",
                        orderside="long",
                        orderset=LONG.ORDERSET.NAME),
         type="chain",
         parent=LONG.ENTRY.RULE,
         label=LONG.STOPLOSS.RULE)

#-------------------------------------------------------------------------------
# trailing stop
#------------------------------------------------------------------------------

trailingStopPercent<-0.07
LONG.TRAILINGSTOP.RULE<-"LE_LRCSMA_TRAILINGSTOP_EXIT"


add.rule(strategy = STRATEGY,name="ruleSignal",
         arguments=list(sigcol=LONG.EXIT.SIGNAL,
                        replace=FALSE,
                        sigval=TRUE,
                        orderqty='all',
                        TxnFees='evalTxnFees',
                        ordertype='stoptrailing',
                        tmult=TRUE,
                        threshold=quote(trailingStopPercent),
                        orderside='long',
                        orderset=LONG.ORDERSET.NAME),
         type="chain",parent=LONG.ENTRY.RULE,
         label=LONG.TRAILINGSTOP.RULE)

#-------------------------------------------------------------------------------
# Take Profit
#------------------------------------------------------------------------------

takeProfit<-0.15
LONG.TAKEPROFIT.RULE<-"LE_LRCSMA_TAKEPROFIT_EXIT"

add.rule(strategy = STRATEGY,name="ruleSignal",
         arguments=list(sigcol=LONG.ENTRY.SIGNAL,
                        replace=FALSE,
                        sigval=TRUE,
                        orderqty='all',
                        TxnFees='evalTxnFees',
                        ordertype='limit',
                        tmult=TRUE,
                        threshold=quote(takeProfit),
                        orderside='long',
                        orderset=LONG.ORDERSET.NAME),
         type="chain",parent=LONG.ENTRY.RULE,
         label=LONG.TAKEPROFIT.RULE)
```

Next, we apply our strategy to our data and update our portfolio, account, and our ending equity position

```r
#------------------------------------------------------------------------------

applyStrategy(strategy =STRATEGY,portfolios = PORTOLIO.NAME,debug = TRUE)

#-------------------------------------------------------------------------------
```

```
getTxns(Portfolio = PORTOLIO.NAME,Symbol = symbol)

updatePortf(Portfolio = PORTOLIO.NAME)
updateAcct(name = ACCOUNT.NAME)
updateEndEq(Account = ACCOUNT.NAME)
```

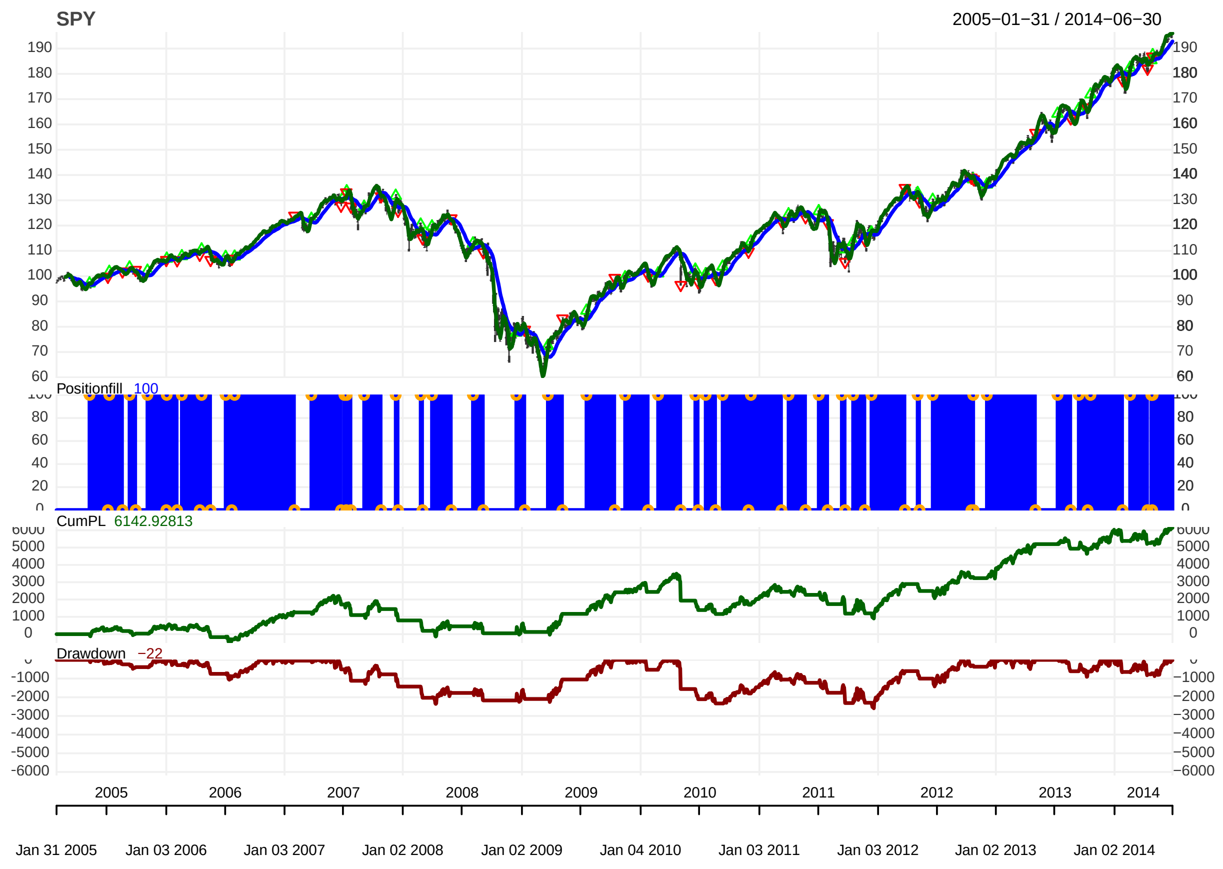
Check data integrity

```
checkBlotterUpdate(PORTOLIO.NAME,ACCOUNT.NAME)
```

```
## [1] TRUE
```

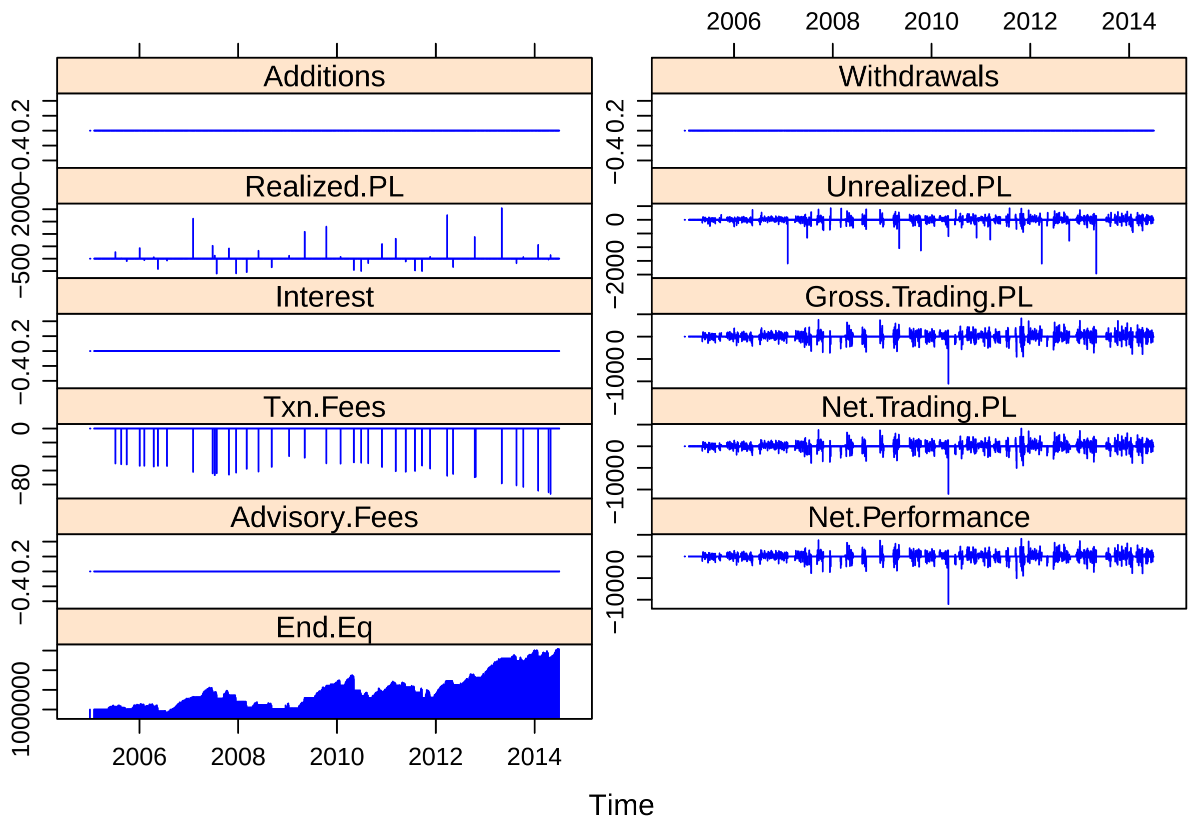
Our optimized strategy yields the following equity curve and drawdowns:

```
chart.Posn(Portfolio = PORTOLIO.NAME,symbols=symbol)
close<-Cl(get(symbol))
add_TA(SMA(close,n=30),col="blue",on=1,lwd=2)
add_TA(LRC(close,n=20),col="darkgreen",on=1,lwd=2)
```



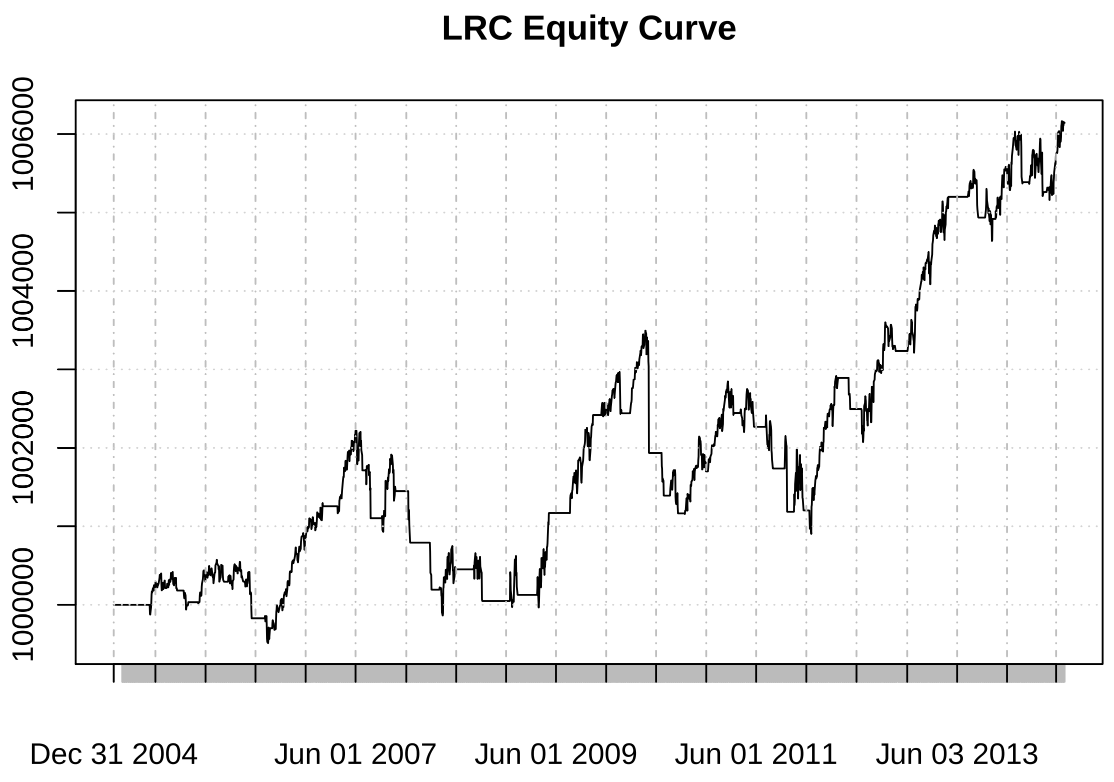
```
accountDetails<-getAccount(Account = ACCOUNT.NAME)
```

```r
xyplot(accountDetails$summary,type="h",col=4)
```



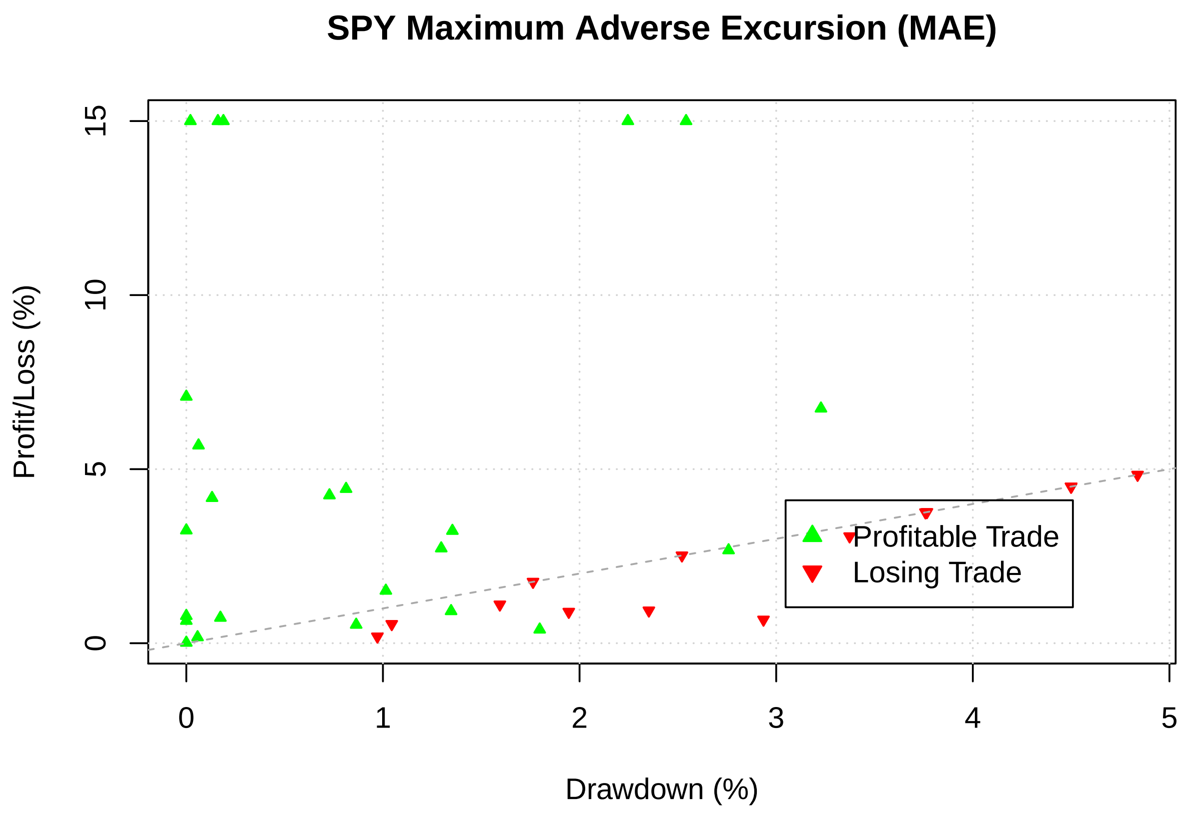
```r
equity<-accountDetails$summary$End.Eq

plot(equity,main="LRC Equity Curve")
```

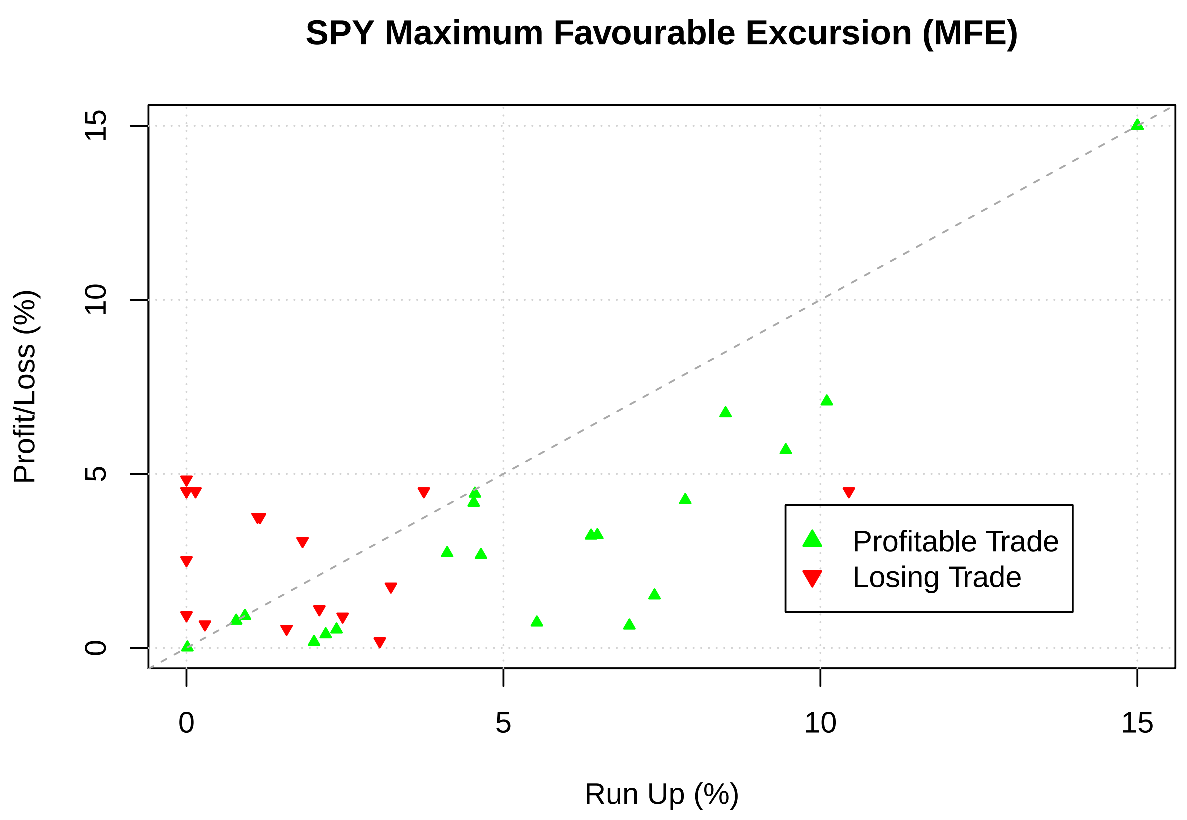


Maximun Adverse Exursion and Maximun Favourable Excursion After Optimization

```
chart.ME(Portfolio=PORTOLIO.NAME, Symbol=symbol, type='MAE', scale='percent')
```



```
chart.ME(Portfolio=PORTOLIO.NAME, Symbol=symbol, type='MAF', scale='percent')
```



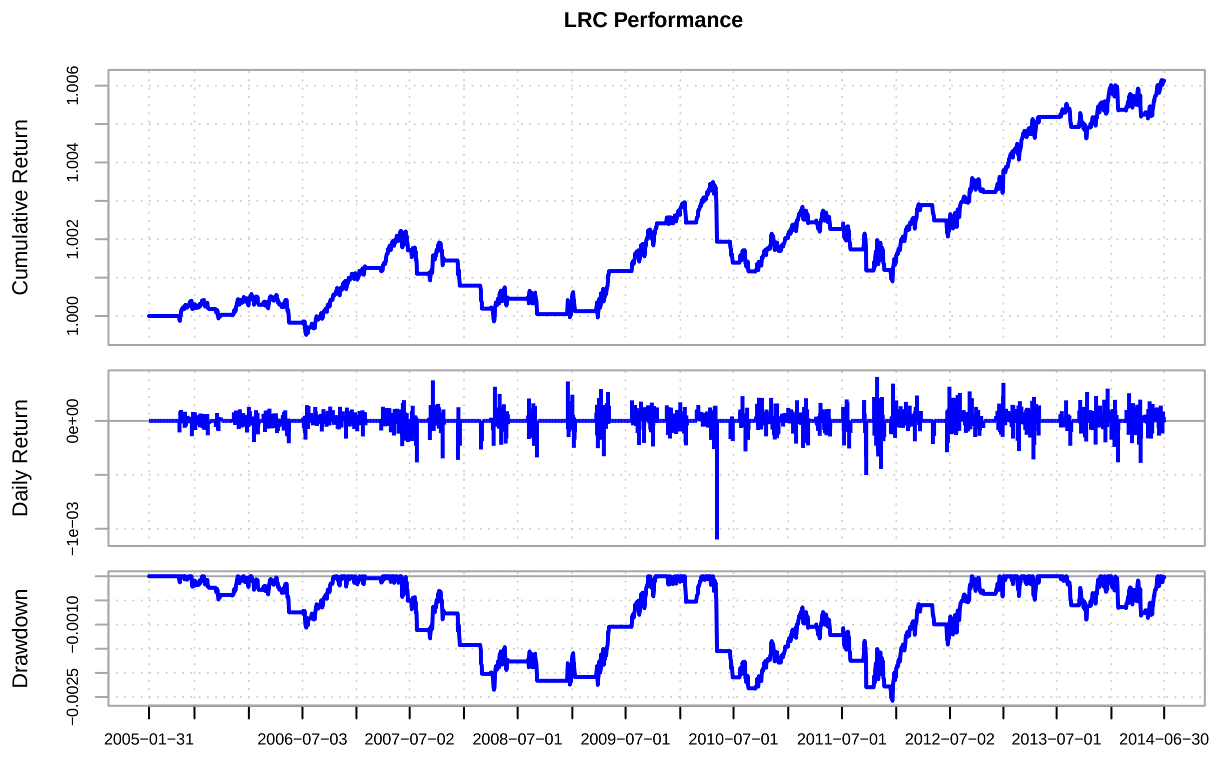
Performance Summary

```
equity.curve <- getAccount(Account = ACCOUNT.NAME)$summary$End.Eq

returns.ns <- Return.calculate(equity.curve,"log")

table.AnnualizedReturns(returns.ns,geometric = FALSE)
table.AnnualizedReturns(returns.ns,geometric = TRUE)


charts.PerformanceSummary(returns.ns,wealth.index=TRUE,geometric = FALSE,
                          colorset="blue",xlab="",
                          main="LRC Performance ",minor.ticks=FALSE)
```


LRC Performance
Cumulative Return
1.000
1.002
1.004
1.006
Daily Return
0e+00
−1e−03
Drawdown
−0.0025
−0.0010
2005−01−31
2006−07−03
2007−07−02
2008−07−01
2009−07−01
2010−07−01
2011−07−01
2012−07−02
2013−07−01
2014−06−30