Michele - test bench

Michele D'Ambrosio

19/12/2019

Abstract

A file where to run my own section.

Contents

1 Introduction **2**

2 Data **2**

2.1 Stocks 2

2.2 Other economic data 2

3 Technical analysis: implementation of a naive strategy in R **3**

3.1 Technical analysis indicators 4

3.2 quantmod, blotter and quantstrat packages 6

3.3 Portfolio management with quantstrat 7

3.3.1 Preliminary setup 8

3.3.2 Create indicators 8

3.3.3 Combining indicators to create trading signals 9

3.3.4 Executing tradings 10

3.3.5 Running the model 11

3.3.6 Performance evaluation 13

3.3.7 Benchmarking with buy-and-hold strategy 15

3.4 Final comments and future steps 16

3.5 knitr and Rmarkdown 17

1 Introduction

What motivates us to do this work. Touch on important finding in literature.

2 Data

Present data used in the analysis

2.1 Stocks

2.2 Other economic data

3 Technical analysis: implementation of a naive strategy in R

As Brock and Lakonishok [1992] explain, *technical analysis* is an "attempt to forecast prices by the study of past prices and a few other related summary statistics about security trading" (pag 2). This however clashes with the Efficient Market Hypotesis lied out by Fama [1970] , according to which market prices incorporate all available information on the securities at any times. Nevertheless, Fama himself highlights the existance of a positive correlation between price changes and returns, which potentially one could try and exploit, if it wasn't that, Fama claims, there are only marginal gains to exploit. Hence, the likely high number of operations needed to reach a significative profit would imply too high transaction costs. In short, there is no chance of profitably forecasting prices.

Despite Fama's paper, which soon got traction among academia, the work by Taylor and Allen [1992] reports that practioners in the world largest financial centres declare making use of Technical analysis, especially for intraday and short-term period trading. This revamped a large stream of research on the topic, to the point where complex optimization algorithms (namely, of the class of evolutionary algorithms) were introduced to get better predictions in terms of global optima search. For example, genetic programming is implemented to avoid the human bias in creating trading rules, letting the algorithms evaluate a series of rules implemented at each search stage, picking the best features and discarding the least performing ones. For a lengthier treatment on the topic, the reader can refer to Neely et al. [1997], or the more recent Mousavi et al. [2014] .

As Lo et al. [2000] put it, *chartism* pertains to a visual analisis of prices charts (hence, not surprisingly, the name), as opposed to the numerical approach of quantitative finance. Their attempt to reconcile the two views leads to the analysis of some *indicators* - namely, the so-called "head and shoulders" - that however may not have desirable mathematical properties. It is Neftci [1991] who had given earlier a proper formal charaterization of the main classes of technical analysis indicators, and it is in fact this paper that inspired the choice of the TA indicators to be used in the present section. Neftci claims that "any well-defined technical analysis rule has to pass the test of being a Markov time", otherwise "the procedure would be using future information in order to issue such signals" (pag. 8). Hence, relying heavily on chart analysis can be misleading. A class of signals that appear to be stopping times are those generated by moving average indicators.

One caveat one should always keep in mind when talking about technical analysis is that its predictive power can be biased upward by its intrinsic self-fulfilling feature: if agents believe that TA can have some predictive power, then following the trends will indeed let the predicted events happen, as all agents will implement similar decisions, given that they base their analysis on similar indicators.

In what follows, we will use some R packages to track the performance of an actively managed portfolio of stocks over six-year period. Trading operations are triggered by signals generated by the usage of technical analysis indicators on past prices. The main purpose is to create a plausible setup that, upon further improvements, can help an agent to infer some information on future prices movements out of the analysis of past prices. We will deliberately leave this model at a simplistic and unsophisticated stage, for the sake of outlying its main basic features with clarity. Nevertheless, several improvements would be needed before a proper implementation for real-life usage. We will mention them in the the remaining of this chapter.

3.1 Technical analysis indicators

In developing our toy model, we considered indicators of intuitive interpretation, widespread use and featuring a 'good' mathematical characterization, as illustrated by Neftci [1991]. We will offer both a formal definition and a visual representation of each indicator. In the following charts, prices are represented by the so-called 'candlesticks'. The rectangular area will show the day open and close, whereas the wicks represent the day high and low. The green filling indicates that the security closed at a higher price than the opening, whereas orange indicates a negative performance on a day-basis. Trading volumes are also shown in the bottom part of the first chart. Moreover, R code chunks used to generate the charts and the results are also shown.

The first indicator is the *simple moving average*. It is defined as

$$SMA = \sum_{t=1}^{n} \frac{x_t}{n} \tag{1}$$

where x_t = the t-th price observation, for $t \in \{1, ..., n\}$.

```
chartSeries(AIR.PA,# need '.PA' suffix for Paris CAC40
            type="matchsticks",
            name = "HLOC Candle chart and SMA, Air France",
            theme=chartTheme('white')) # chart setting

quantmod::addSMA(n=50,on=1,col = 'blue') # to overimpose SMA(50), in blue
quantmod::addSMA(n=200,on=1,col = 'purple') # TA functions from 'TTR'
```

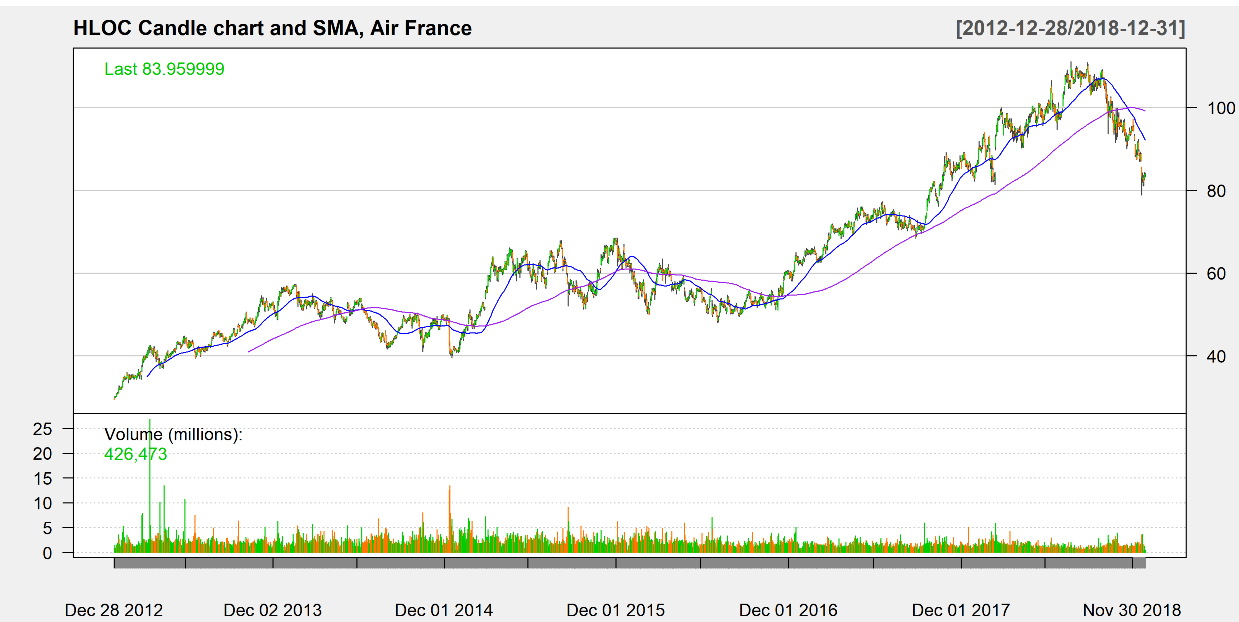

Figure 1: Daily HLOC Chart price, with SMA(50) in blu, and SMA(200) in purple.

The SMA is used to smooth out the price trends from the short-term fluctuations. Usually, (at least) two SMA's of different time-lags are used together to generate trading signals. The short-term average crossing from below the long-term average is considered a buy signal, whereas a crossing in the other direction is taken as an indication to sell. One common choice for the lags are the 50- and the 200-days averages.

The second indicator is the **Relative Strength Index**, or RSI . It is computed as

$$RSI = 100 - \left(\frac{100}{(1 + \frac{\Delta_u}{\Delta_d}} \right) \tag{2}$$

where Δ_u = Average of Upward Price Change, and Δ_d = Average of Downward Price Change[1]. (See also Hsu et al. [2016] for an example of different parametrizations.)

The RSI is an oscillator, since its value is in the range [0,100]. It gives indications on the **momentum**, that is, the "the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset."[2] Generally, $RSI > 70$ is taken as an indication for a security to be overbought, whereas $RSI < 30$ has the opposite meaning.

```
quantmod::addRSI(n=14, maType= 'SMA')
# the RSI is shown, albeit not overimposed to the price pattern
```

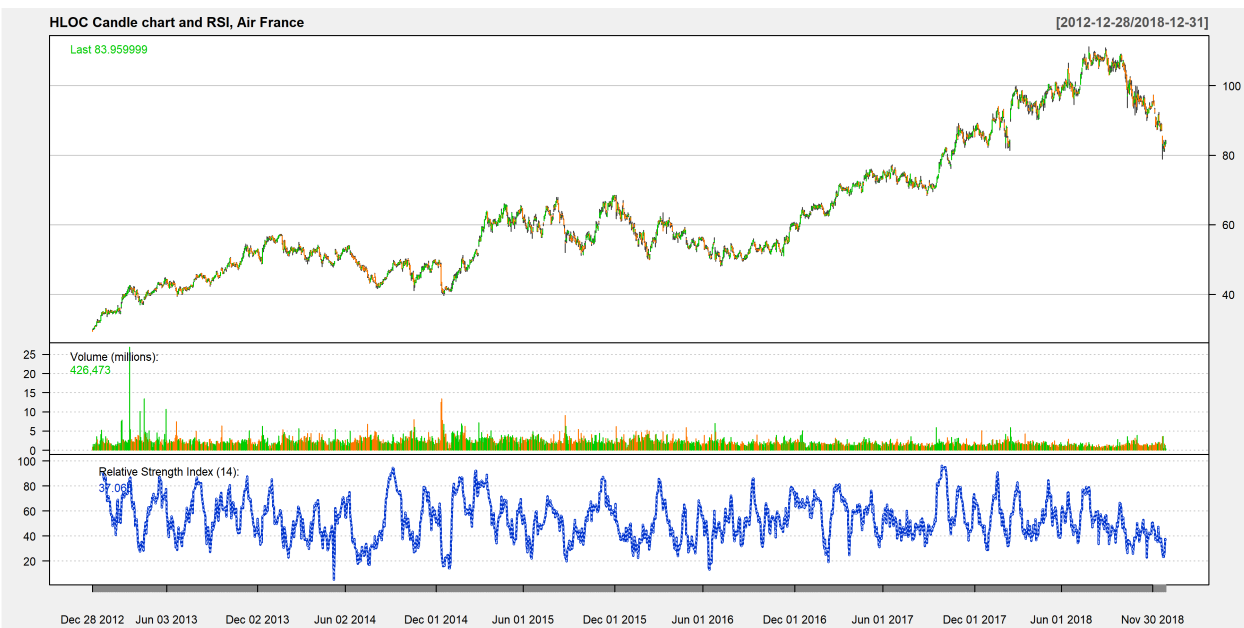

Figure 2: Daily HLOC Chart price with RSI(14)

The third indicator are the **Bollinger Bands** [3]. These are defined by the plot of a simple moving average, as defined below by the μ, along with its standard deviation around it, defining the upper

[1] https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/RSI
[2] https://www.investopedia.com/terms/r/rsi.asp
[3] https://www.bollingerbands.com/

and lower bands. These are defined respectively as

$$B_u = MA(\mu, n) + m \cdot \sigma(\mu, n) \tag{3}$$

and

$$B_d = MA(\mu, n) - m \cdot \sigma(\mu, n) \tag{4}$$

with

$HLC_t := High_t + Low_t + Close_t$;

$\mu = \frac{HLC_t}{3}$;

$\sigma_n = \sqrt{\frac{\sum_{t=1}^{n} (HLC_t - \mu)^2}{n}}$ the standard deviation of HLC over n days;

$n \in \mathbb{N}^*$ the number of days over which the SMA is computed;

$m \in \mathbb{N}^*$ the number of standard deviations used to create the width of the bands.

A common parametrization of this indicator is one with two-standard-deviation-wide bands around a 20-day simple moving average (i.e. $m = 2$,$n = 20$).

```
addBBands(n=20, sd=2, maType= "SMA")
```

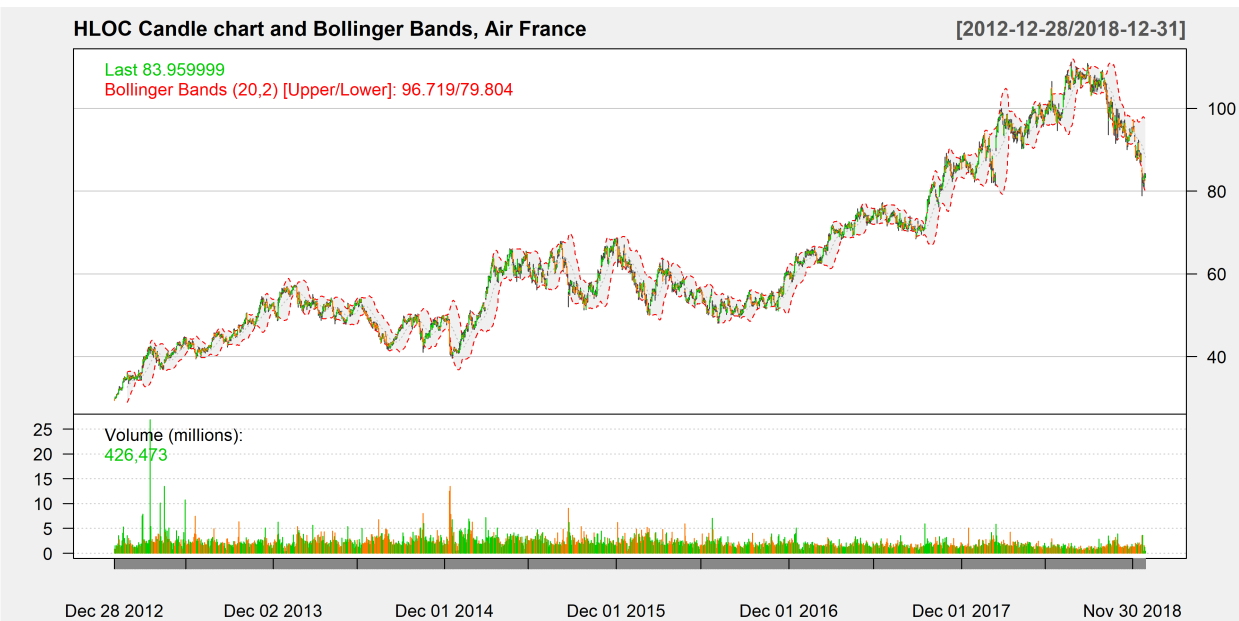

Figure 3: Daily HLOC Chart price with Bollinger Bands (20,2)

3.2 quantmod, blotter and quantstrat packages

One important building block towards the implementation of Technical Analysis in R is the package quantmod[4], which allows the user to download stock prices data from Yahoo! Finance, and

[4] https://github.com/joshuaulrich/quantmod

to visualize different types of price charts, which are of great support in a first, rough analysis of the evolution of price trends over time. For the sake of offering a visual understanding of TA indicators, these functionalities were already exploited in the previuous paragraph, using the functions `quantmod::getSymbols()` and `quantmod::chartSeries`, along with `quantmod::addSMA`, `quantmod::addRSI`, and `quantmod::addBBands`.

Moreover, R supports a package named `quantstrat`, a "transaction-oriented infrastructure for constructing trading systems and simulation[5]". However, this package counts several dependencies, and therefore we'll need our machine to be equipped with all these packages, as brilliantly shown in Yu [2019].

```
install.packages("quantmod")
install.packages("FinancialInstrument")
install.packages("PerformanceAnalytics")
install.packages("foreach")
install.packages("devtools")
devtools::install_github("braverock/blotter")
devtools::install_github("braverock/quanstrat")
```

3.3 Portfolio management with quantstrat

We build an actilvely managed portfolio made of five of the top ten firms of the CAC40, as of September 2019[6], using six years of data, from 2012-12-28 to 2018-12-31, for a total of 1534 trading days. We highlight that the choice of the stocks felt simply on some of the most liquid Starting with an initial capital of € 100,000, we will buy or sell 100 stocks of each equity, whenever some conditions on historic prices are met. These conditions are later referred to as *signals*, and are linked to the TA indicators illustrated in their mathematical essence in the previuos part. We then offer an overview on how to translate these mathematical conditions into a 'quantstrat' action in the sections that follow.

Table 1: :
Top ten firms in CAC40 by market cap, September 2019

Company	MNEMO	Sector	Weight%
TOTAL	FP	Oil and Gas	9,54
LVMH	MC	Personal and Household Goods	7,97
SANOFI	SAN	Health Care	7,54
AIRBUS	AIR	Industrial Goods and Services	5,47
L'OREAL	OR	Personal and Household Goods	5,10
AIR LIQUIDE	AI	Chemicals	4,40
DANONE	BN	Food and Beverage	4,14
VINCI	DG	Construction and Materials	3,97
BNP PARIBAS ACT.A	BNP	Banks	3,95
SAFRAN	SAF	Industrial Goods and Services	3,72

[5] https://www.rdocumentation.org/packages/quantstrat/versions/0.16.6

[6] The full list here: https://live.euronext.com/en/product/indices/FR0003500008-XPAR/market-information

3.3.1 Preliminary setup

We first retrieve data from Yahoo to the R global environment:

```
from ="2012-12-28"
to ="2018-12-31"
symbols = c("MC.PA","BN.PA","AIR.PA","BNP.PA","DG.PA" )
# the .PA label is needed to specify the Paris CAC40 stock exchange
quantmod::getSymbols(symbols,
                     from=from, to=to)
```

We then set up the strategy, the portfolio and the account objects:

```
initEq=100000 # capital to invest

strategy.st <- "strat.full.limit" # name the objects
portfolio.st <- "portf.full.limit"
account.st <- "acct.full.limit"

blotter::initPortf(portfolio.st, symbols) #initialize portfolio
blotter::initAcct(account.st, # also need specifing linked portfolio
         portfolios=portfolio.st,
         initEq = initEq)
quantstrat::initOrders(portfolio.st) # the orders 'container'
quantstrat::strategy(strategy.st, store=TRUE) # we save our strategy in
# the global environement
```

3.3.2 Create indicators

As illustrated earlier, we make use of three simple indicators: simple moving average, relative strength index and Bollinger Bands. The indicators will be linked to the startegy object previoulsy defined. The input dataframe is called *mktdata*, an object that will be created authomatically when we validate the strategy.

```
quantstrat::add.indicator(
  strategy.st, name="SMA", # "name" must correspond to an R function
  arguments= list(x= quote(Cl(mktdata)), n=50), # input data are closing prices
  # set input data (mktdata) and indicator parameters through function 'arguments'
  label="sma50") ## unique reference 'mktdata' column name

add.indicator(strategy = strategy.st,
              name = 'RSI', # normally, we'll be using functions
              # from the TTR package
              arguments = list(price = quote(Cl(mktdata)), n=14),
              # among the 'arguments', we need to set the
              # parameter (number of days of the SMA) for the RSI
              label = 'rsi.14')
```

```
add.indicator(strategy = strategy.st,
              name= "BBands",
              arguments =
                list(HLC = quote(HLC(mktdata)),n=20, maType= "SMA", sd=2),
              # BBands need High,Low and Closing Prices as an Input
              label = "bb.20.2")
```

3.3.3 Combining indicators to create trading signals

The third step after setting up the blotter objects and creating the indicators, is creating buy and sell signals out of an appropriate blend of indicators. Below we offer a summary of the signals that will be used here.

Table 2: Summary of signals and related trading executions

Type of Signal (quantmod)	Signal	Execution	Label
Crossover	SMA(50)>SMA(200)	NA	smabuy
Crossover	SMA(50)<SMA(200)	NA	smasell
Threshold	RSI<30	NA	rsi14buy
Threshold	RSI>70	NA	rsi14sell
Crossover	Closing > Upper BBand	NA	Cl.gt.UpperBand
Crossover	Closing < Lower BBand	NA	Cl.lt.LowerBand
Crossover and Threshold	SMA(50)>SMA(200) and RSI<30	Buy	entry1
Crossover and Threshold	SMA(50)<SMA(200) and RSI>70	Sell	exit1
Crossover and Threshold	Closing < Lower BBand and RSI<30	Buy	entry2
Crossover and Threshold	Closing > Upper BBand and RSI>70	Sell	exit2

Despite having defined ten signals, only four of these will trigger actual execution orders. This is because it looks safer to rely on a mixed signal - i.e. one making use of more than one indicator - rather than only on a simple one.

Here is some code showing how a mixed signal is created, based on the evaluation of two simple ones, which in turn may have different level of complexities, depending on the kind of indicators used to build them. Notice that the function `sigFormula` can take more than two arguments, evaluating therefore several conditions at once.

```
## 'Bull market' (i.e. market considered underpriced --> indication to buy)
# if SMA50>SMA200
quantstrat::add.signal(
  strategy.st, name="sigCrossover", ## "name" is a function taking on
           # specific arguments
           arguments= list(columns=c("sma50","sma200"),relationship="gt"),
            # 'columns' refers to the indicators defined above
           label="smabuy") # 'label' will be the reference for
# the execution phase hereafter.

## Specifies all instance when RSI is below 30
```

```r
## (indication of asset being oversold)
add.signal(strategy.st, name = "sigThreshold",
           arguments =
             list(column = "rsi.14", threshold = 30, relationship = "lt",
                  cross = TRUE), # signals
           label = "rsi14buy")

## sigFormula which indicates that both smabuy and rsi14buy must evaluate TRUE.
add.signal(strategy.st, name = "sigFormula",
           arguments = list(formula = "smabuy & rsi14buy",cross = TRUE),
           # notice that the formula is evaluating two booleans
           label = "entry1")
```

3.3.4 Executing tradings

The very last step of our strategy will be defining the trading operations to execute, once the relevant signals are observed. As shown in the table above, only four signals trigger an order execution. We define accordingly four trading execution rules.

```r
## Enter the position when SMA(50)>SMA(200), and the RSI<30
quantstrat::add.rule(
  strategy.st, name = "ruleSignal",
         arguments = list(sigcol = "entry1", sigval = TRUE,
                          orderqty = 1000, ordertype = "market",
                          orderside = "long", replace = FALSE,
                          prefer = "Open",
                          tradeSize = tradesize, maxSize = tradesize),
         type = "enter")


## Exit the position when SMA(50)<SMA(200), and the RSI>70
add.rule(strategy.st, name = "ruleSignal",
         arguments = list(sigcol = "exit1", sigval = TRUE,
                          orderqty = -1000 , ordertype = "market",
                          orderside = "long", replace = FALSE,
                          prefer = "Open",
                          tradeSize = tradesize,
                          maxSize = tradesize),
         type = "exit")


## Enter the position when (Closing price < Lower BBand) and RSI<30
add.rule(strategy = strategy.st, name= 'ruleSignal',
         arguments = list(sigcol = "entry2", sigval= TRUE,
                          orderqty = 1000, ordertype= 'market',
                          orderside= NULL,threshold=NULL ),
```

```
        type= "enter")


## Exit the position when (Closing price > Upper BBand) and RSI>70
add.rule(strategy = strategy.st, name= 'ruleSignal',
         arguments = list(sigcol = "exit2", sigval= TRUE,
                          orderqty = -1000, ordertype= 'market', orderside= NULL,
                          threshold=NULL ),
         type= "exit")
```

3.3.5 Running the model

At this stage, we are ready to run our model. It is safer to check first whether the code for the strategy we seek to implement is 'bug-free'. To do this, we run the `quantstrat::applyStrategy` command within the `base::try` function, which allows to "run an expression that might fail and allow the user's code to handle error-recovery"[7].

```
testing_strat <- base::try( quantstrat::applyStrategy(strategy.st,
                       portfolios=portfolio.st ))
## try is a function in one of the basis R packages

## We test whether our entire startegy was coded correctly
```

```
## [1] "Set capital to 100,000 Euros "
## initalize portfolio: portf.fullinitalize account: acct.fullinitializie strategy: strat.full
## [1] "2013-08-01 00:00:00 AIR.PA -100 @ 45.275002"
## [1] "2013-10-07 00:00:00 AIR.PA -100 @ 50.299999"
## [1] "2014-07-17 00:00:00 AIR.PA 100 @ 44.98"
## [1] "2014-09-05 00:00:00 AIR.PA -100 @ 49.189999"
## [1] "2015-01-26 00:00:00 AIR.PA -100 @ 50.139999"
## [1] "2015-02-19 00:00:00 AIR.PA -100 @ 51.990002"
## [1] "2015-03-02 00:00:00 AIR.PA -100 @ 56.09"
## [1] "2016-06-14 00:00:00 AIR.PA 100 @ 50.98"
## [1] "2016-06-17 00:00:00 AIR.PA 100 @ 50.830002"
## [1] "2016-12-14 00:00:00 AIR.PA -100 @ 62.400002"
## [1] "2016-12-16 00:00:00 AIR.PA -100 @ 64.18"
## [1] "2017-02-24 00:00:00 AIR.PA -100 @ 68.349998"
## [1] "2017-10-25 00:00:00 AIR.PA -100 @ 83.470001"
## [1] "2017-11-01 00:00:00 AIR.PA -100 @ 87.379997"
## [1] "2018-06-15 00:00:00 AIR.PA -100 @ 103.599998"
## [1] "2013-06-25 00:00:00 FP.PA 100 @ 36.294998"
## [1] "2013-08-26 00:00:00 FP.PA -100 @ 42"
## [1] "2013-08-29 00:00:00 FP.PA -100 @ 42.509998"
## [1] "2014-05-09 00:00:00 FP.PA -100 @ 52.259998"
```

[7] https://stat.ethz.ch/R-manual/R-devel/library/base/html/try.html

```
## [1] "2014-10-16 00:00:00 FP.PA 100 @ 42.555"
## [1] "2017-07-10 00:00:00 FP.PA 100 @ 42.764999"
## [1] "2018-05-18 00:00:00 FP.PA -100 @ 54.48"
## [1] "2014-01-09 00:00:00 MC.PA 100 @ 122.949997"
## [1] "2014-04-11 00:00:00 MC.PA -100 @ 141.199997"
## [1] "2014-07-28 00:00:00 MC.PA 100 @ 131.550003"
## [1] "2014-11-24 00:00:00 MC.PA -100 @ 144.300003"
## [1] "2014-12-18 00:00:00 MC.PA 100 @ 130.25"
## [1] "2015-02-05 00:00:00 MC.PA -100 @ 153.5"
## [1] "2015-03-12 00:00:00 MC.PA -100 @ 170.600006"
## [1] "2016-07-28 00:00:00 MC.PA -100 @ 153.350006"
## [1] "2016-10-12 00:00:00 MC.PA -100 @ 164.350006"
## [1] "2016-10-17 00:00:00 MC.PA -100 @ 165.399994"
## [1] "2017-01-20 00:00:00 MC.PA -100 @ 190.949997"
## [1] "2017-03-20 00:00:00 MC.PA -100 @ 200"
## [1] "2017-04-03 00:00:00 MC.PA -100 @ 204.050003"
## [1] "2017-04-25 00:00:00 MC.PA -100 @ 223.149994"
## [1] "2017-10-27 00:00:00 MC.PA -100 @ 253.949997"
## [1] "2018-04-11 00:00:00 MC.PA -100 @ 278.25"
## [1] "2018-10-11 00:00:00 MC.PA 100 @ 261.950012"
## [1] "2013-01-30 00:00:00 OR.PA -100 @ 113.550003"
## [1] "2013-04-03 00:00:00 OR.PA -100 @ 127.300003"
## [1] "2018-02-09 00:00:00 OR.PA 100 @ 172.350006"
## [1] "2018-10-11 00:00:00 OR.PA 100 @ 187.600006"
## [1] "2013-04-03 00:00:00 SAN.PA -100 @ 80.050003"
## [1] "2013-11-07 00:00:00 SAN.PA -100 @ 78.660004"
## [1] "2014-10-16 00:00:00 SAN.PA 100 @ 78.800003"
## [1] "2014-10-29 00:00:00 SAN.PA 100 @ 71.150002"
## [1] "2015-02-16 00:00:00 SAN.PA -100 @ 85.720001"
## [1] "2015-02-20 00:00:00 SAN.PA -100 @ 87.389999"
## [1] "2015-02-24 00:00:00 SAN.PA -100 @ 88.75"
## [1] "2015-04-10 00:00:00 SAN.PA -100 @ 98.75"
## [1] "2016-06-15 00:00:00 SAN.PA 100 @ 68.110001"
## [1] "2016-11-10 00:00:00 SAN.PA -100 @ 76.980003"
## [1] "2017-04-05 00:00:00 SAN.PA -100 @ 85.43"
## [1] "2017-05-05 00:00:00 SAN.PA -100 @ 89.690002"
## [1] "2017-12-06 00:00:00 SAN.PA 100 @ 73.25"
## [1] "2018-07-05 00:00:00 SAN.PA -100 @ 72.440002"
## [1] "2018-08-01 00:00:00 SAN.PA -100 @ 75.599998"
```

Since the strategy appears to run smoothly, we can update our portfolio and account to register the transactions and variations in the level of capital owned. We are now ready to extract the data on the performance, in the form of summary statistics and charts. Unfortunately, it seems that, due to an apparently still unfixed bug in the quantstrat package[8], it is not possible to export the strategy log. This is why we kept it printed to screen, as a full output of the command that runs the relevant strategy implementation script.

[8] https://github.com/braverock/quantstrat/issues/57

3.3.6 Performance evaluation

As summarized in the first line of the table below, our strategy tested on a five-year panel with five stocks seems 'conservative', in that the maximum number of tradings on a stock is 17 (for MC.PA), with an average of roughly 12 across the five stocks. As the table below shows, for three out of five stocks, the strategy net performance is negative. In particular, trading on LVHM register a loss above €50,000. On the other hand, the strategy yields a positive return for Total and Sanofi. The lines "Percent.Positive" and "Percent.Negative", combined with "Num.Trades", gives an idea of how many open positions were closed during the observed time frame, and if the operation carried a positive or negative yield. By looking at the transaction log, one can see that the strategy on both Air France was particualry unsuccessful because several short positions were opened (hence, with the 'expectation' of a fall in the prices), and were (forcibly) closed only 'at the end of the world' i.e. when we close the account and the portfolio at the end of our observed period, after prices had rather steadly surged. For LVHM we see a slightly different story, in which the first three couples of operations are three profitable buy&sell , after which the algorithms becomes inefficient and determines the same patter as AIR.PA to be observed. Sanofi looks like the example of a successful implementation of our algorithm.

There are now some issues regarding our strategy that we would like to higlight, and that would require further inspection and study:

- the first n days of observations (where n is the highest parameter value across all indicators making use of moving averages) , should be left out. This is because during those n days only one out of two rules is implemented. This can clearly be source of biases. However, in this R framework , it does not seem possible to implement this option;
- the algorithm parameters were set by a 'thumb-rule' type of choice, i.e. no action was taken to determine whether any further restrictons were to be put in place to obtain a better performance. For a better treatment on the code-related side of the matter, we refer the reader to Trice [2016] for a reference to parameter optimization and backtesting ;
- no in-depth study on the market conditions over the six years observed was made, neither on the specific stocks cases;
- there is clearly the need for some constraints that prevent opening an excessive number of short positions - especially when the algorithm returns signals contrasting with price trend.

One caveat about the table that follows: it is generated through `blotter::tradeStats`, and its output cannot be truncated. Moreover, the function developement seems unfinished, as one can notice from the uncomplete description of the statistics[9] .

Table 3: Startegy trading statistics

Measure	AIR.PA	FP.PA	MC.PA	OR.PA	SAN.PA
Num.Txns	16	7	17	4	15
Num.Trades	1	3	4	1	2
Net.Trading.PL	-17778	2346	-52745	-11910	9853

[9]Full account of the statistics here: https://rdrr.io/rforge/blotter/man/tradeStats.html

Measure	AIR.PA	FP.PA	MC.PA	OR.PA	SAN.PA
Avg.Trade.PL	-17778	781.8	-13186	-11910	4926
Med.Trade.PL	-17778	830	1550	-11910	4926
Largest.Winner	0	570.5	2325	0	2204
Largest.Loser	-176.4	0	-6154	-6718	0
Gross.Profits	0	2346	5425	0	9853
Gross.Losses	-17778	0	-58170	-11910	0
Std.Dev.Trade.PL	NA	191.8	29992	NA	5728
Std.Err.Trade.PL	NA	110.8	14996	NA	4050
Percent.Positive	0	100	75	0	100
Percent.Negative	100	0	25	100	0
Profit.Factor	0	NA	0.09	0	NA
Avg.Win.Trade	NA	781.8	1808	NA	4926
Med.Win.Trade	NA	830	1825	NA	4926
Avg.Losing.Trade	-17778	NA	-58170	-11910	NA
Med.Losing.Trade	-17778	NA	-58170	-11910	NA
Avg.Daily.PL	-147.2	505.2	-182.4	-5955	1116
Med.Daily.PL	-161.4	483	1550	-5955	1102
Std.Dev.Daily.PL	38.33	57.55	4004	1078	907.1
Std.Err.Daily.PL	22.13	33.22	2002	762.5	453.5
Ann.Sharpe	-60.98	139.3	-0.72	-87.67	19.53
Max.Drawdown	-45318	-1522	-115215	-16030	-10475
Profit.To.Max.Draw	-0.39	1.54	-0.46	-0.74	0.94
Avg.WinLoss.Ratio	NA	NA	0.03	NA	NA
Med.WinLoss.Ratio	NA	NA	0.03	NA	NA
Max.Equity	1440	2440	9345	1045	16406
Min.Equity	-43878	-856.5	-105870	-14985	-3323
End.Equity	-17778	2346	-52745	-11910	9853

As one could have speculated by looking at the strategy log, only FP.PA and SAN.PA yield a positive return, under the active strategy we implemented. On the other hand, the three other stocks yield a negative return, with MC.PA being the worst among the three.

A quick glance at the risk measure indicator seems to suggest that a high volatility may have an implication on the accuracy of our automated trading strategy - indeed, AIR.PA and MC.PA were traded unprofitably, and their standard deviation is the highest. One may guess that the algorithm was thrown off by the frequent changes in prices. There is clearly the need for a more thourugh analysis of these statistics, which however goes beyond the scope of our analysis.

Table 4: Active strategy returns' statistics

Measure	AIR.PA DailyEqPL	FP.PA DailyEqPL	MC.PA DailyEqPL	OR.PA DailyEqPL	SAN.PA DailyEqPL
Cumulative Return	-0.202	0.023	-0.55	-0.121	0.09
Annualized Return	-0.036	0.004	-0.123	-0.021	0.014

Measure	AIR.PA DailyEqPL	FP.PA DailyEqPL	MC.PA DailyEqPL	OR.PA DailyEqPL	SAN.PA DailyEqPL
Annualized Sharpe Ratio	-0.291	0.35	-0.411	-0.372	0.228
Annualized StdDev	0.125	0.011	0.299	0.056	0.063

Finally, one very last measure of "fitness" of our startegy is by far the simplest: the level of capital left in our account, once all positions have been closed. We see that the initial capital of € 100,000 has decreased to below $5e+04$, that is, below € 50,000. Moreover, there seems to be the need for a constraint on the capital invested, since towards the end of the observed period, we would actually get indebted. To our knowledge, one should hard-code an appropriate function to prevent more short positions being taken once the level of capital goes below zero.

Figure 4: Account Capital level at closing of active positions

3.3.7 Benchmarking with buy-and-hold strategy

The negative performance of our active portfolio strategy prompts the comparison with a passive strategy. We now implement a buy and hold strategy, over the same timeframe as before, with the same stocks we considered for our active strategy. Here we will allocate equal parts of the initial capital to each of the five stocks, and execute a buy order for each on the first day of observations. On the last day, we sell all the stocks we had bought. Thus, the increase(reduction) in the stock price will determine our gains(losses).

```
## [1] "Set capital to 100,000 Euros "
## [1] "2012-12-28 00:00:00 FP.PA 514 @ 38.91"
```

```
## [1] "2012-12-28 00:00:00 MC.PA 145 @ 137.800003"
## [1] "2012-12-28 00:00:00 SAN.PA 282 @ 70.730003"
## [1] "2012-12-28 00:00:00 AIR.PA 680 @ 29.395"
## [1] "2012-12-28 00:00:00 OR.PA 190 @ 104.800003"
## [1] "2018-12-31 00:00:00 FP.PA -514 @ 46.18"
## [1] "2018-12-31 00:00:00 MC.PA -145 @ 258.200012"
## [1] "2018-12-31 00:00:00 SAN.PA -282 @ 75.660004"
## [1] "2018-12-31 00:00:00 AIR.PA -680 @ 83.959999"
## [1] "2018-12-31 00:00:00 OR.PA -190 @ 201.199997"
```

Since each stock price has risen during the five years examined, it can be easily noted that this buy and hold strategy appears rather profitable: our initial € 100,000 investment stake has increased by about 80%. At this stage, it is difficult to think that an active strategy could have beaten the market. On the other hand, an in-depth study is required to examine what factors determined such increases in prices. This in turn highlights the importance of an appropriate market study that should accompany the creation of the trading strategy. Clearly, investing is far from gambling, and coding or mathemathical skills cannot make up for a poor understanding of the markets.

Figure 5: Account Capital level at closing of long positions in B&H strategy

3.4 Final comments and future steps

It is clear that our strategy implementation lacks of several important feautres, on which we already touched above. We left one important point untackled - namely, the absence of a proper testing strategy. As Guy Yollin illustrates[10], implementing a trading strategy would first require a the so-called *walk-forward analysis*, which allows for a dynamic parametrization of the model, with a lower impact of overfitting[11]. Nevertheless, we would rather highlight the importance of having created

[10] http://www.r-programming.org/files

[11] https://algotrading101.com/learn/what-is-walk-forward-optimization/

a realistic model, which can constitute a first, solid building block towards a proper in-depth study of TA optimized applications to trading.

3.5 knitr and Rmarkdown

As a final remark, we would like to add that this report was created in Rmarkdown[12] and compiled with with `knitr`[13] .

References

William Brock and Josef Lakonishok. Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. *The Journal of Finance*, 47(5):1731–1764, 1992. URL http://www.jstor.org/stable/2328994.

Eugene F Fama. Efficient Capital Markets: A Review of Theory and Empirical Work Author(s):. *The Journal of Finance*, 25(2):383–417, 1970. URL https://www.jstor.org/stable/2325486Accessed:.

Po Hsuan Hsu, Mark P. Taylor, and Zigan Wang. Technical trading: Is it still beating the foreign exchange market? *Journal of International Economics*, 102:188–208, 2016. ISSN 18730353. doi: 10.1016/j.jinteco.2016.03.012. URL http://dx.doi.org/10.1016/j.jinteco.2016.03.012.

Andrew W. Lo, Harry Mamaysky, and Jiang Wang. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *Journal of Finance*, 55 (4):1705–1765, 2000. ISSN 00221082. doi: 10.1111/0022-1082.00265.

Somayeh Mousavi, Akbar Esfahanipour, and Mohammad Hossein Fazel Zarandi. A novel approach to dynamic portfolio trading system using multitree genetic programming. *Knowledge-Based Systems*, 66:68–81, 2014. ISSN 09507051. doi: 10.1016/j.knosys.2014.04.018. URL http://dx.doi.org/10.1016/j.knosys.2014.04.018.

Christopher Neely, Paul Weller, and Rob Dittmar. Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach. *The Journal of Financial and Quantitative Analysis*, 32(4):405, 1997. ISSN 00221090. doi: 10.2307/2331231.

Salih N. Neftci. Naive Trading Rules in Financial Markets and Wiener-Kolmogorov Prediction Theory: A Study of "Technical Analysis". *The Journal of Business*, 64(4):549, 1991. ISSN 0021-9398. doi: 10.1086/296551.

Mark P. Taylor and Helen Allen. The use of technical analysis in the foreign exchange market. *Journal of International Money and Finance*, 11(3):304–314, 1992. ISSN 02615606. doi: 10.1016/0261-5606(92)90048-3.

Tim Trice. Backtesting Strategies with R, 2016. URL https://timtrice.github.io/backtesting-strategies/basic-strategy.html.

Ko Chiu Yu. Techincal Analysis with R, 2019. URL https://bookdown.org/kochiuyu/Technical-Analysis-with-R/.

[12] https://bookdown.org/yihui/rmarkdown/

[13] https://yihui.org/knitr/