Trading Strategy 1: What goes up, goes up…

*June 26, 2013*

By [s3admq](http://www.r-bloggers.com/author/s3admq/)

(This article was first published on [**Trading and travelling and other things » R**](http://tradingposts.wordpress.com/2013/06/26/trading-strategy-1-what-goes-up-goes-up/), and kindly contributed to [R-bloggers)](http://www.r-bloggers.com/)

As I said earlier, my main task at my internship is to hunt for profitable strategies. As you can imagine, strategies can range from the exceedingly simple and easy to implement, to the crazily complex. Let’s start out with one of the simplest trading strategies out there.

The gist of this strategy is you buy the security tomorrow if the closing price today was higher than the opening price today. You hold the security until the end of the day and sell it at close.

This strategy requires buying at open and selling at close so we will calculate Open to Close returns.

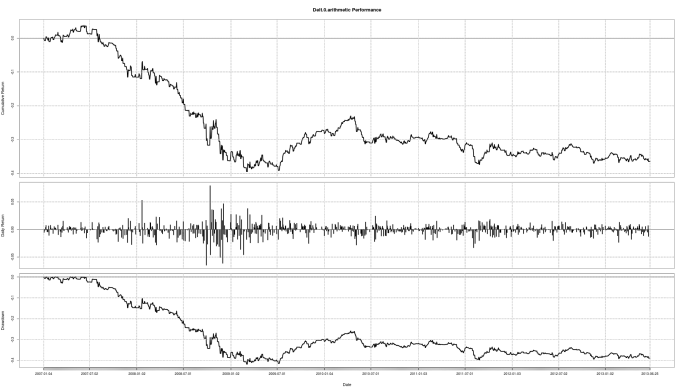
**1**-Get data:  
**getSymbols(‘SPY’)**

**2**-Calculate Open to Close returns:  
**retVec=Delt(Op(SPY),Cl(SPY))**

**3**-Create a trading signal vector:  
**binaryVec=lag(ifelse(Cl(SPY)>Op(SPY),1,0),1)** #This is saying that for any given index in the Opening and Closing price vectors, when the closing price is greater than the closing price, enter it as a ’1′ in the binaryVec vector, otherwise enter a ’0′. Then lag it by one day as you see the signal today but are only able to act on it tomorrow.

**4**-Finally, multiply the trading signal vector with the return vector and run performance analytics on it:  
**stratVec=retVec\*binaryVec**

**5-charts.PerformanceSummary(stratVec)**

[](http://tradingposts.files.wordpress.com/2013/06/strat1summary.png)

Note: Lagging creates NA values in the vector, which you can set to 0 by saying**laggedVector[is.na(laggedVector)]=0.**

This was an easy to run strategy and, not surprisingly, it didn’t do too well (although 2007 has to be one of the worst times historically to enter the market with long-only positions). Our intuition for this strategy was that if a stock ends higher today, it will carry that momentum forward to tomorrow. Several modifications can be made, such as tightening our criteria for what we label as momentum. To avoid capturing just random price fluctuations (to a marginal extent), we can raise the bar for what we judge as momentum by saying that instead of trading tomorrow if today’s price return was positive, we will only trade tomorrow if the last 2 days’ price return was positive as well, or the last 5 days’ price return was positive. There are two ways (that I can think of) something like this can be done in R, which I will cover in the next post.

Strategy 1 Extended (Part 1)

*June 26, 2013*

By [s3admq](http://www.r-bloggers.com/author/s3admq/)

(This article was first published on [**Trading and travelling and other things » R**](http://tradingposts.wordpress.com/2013/06/26/strategy-1-extended-part-1/), and kindly contributed to [R-bloggers)](http://www.r-bloggers.com/)

Like I said in my previous post, there are two ways I could think of, off the top of my head, to implement a 2-day or 5-day extension to the previous strategy. One way would be just a simple extension of the approach above, using multiple conditions in the if-else statement with lagged vectors. This is how I will solve it here.

The second method, and one that I would be more comfortable with, is just coding it using a for loop and if statements. Its pretty straightforward if you know even the very basics of coding, and if you do, you can probably figure it out on your own.

I tried stringing together a bunch of ‘and’ conditions but it didn’t seem to work. This is what I had typed in:  
**binVec3Day=ifelse(((Cl(SPY)>Op(SPY)) && (lag(Cl(SPY),1)>lag(Cl(SPY),1)) && (lag(Cl(SPY),2)>lag(Op(SPY),2))),1,0)**

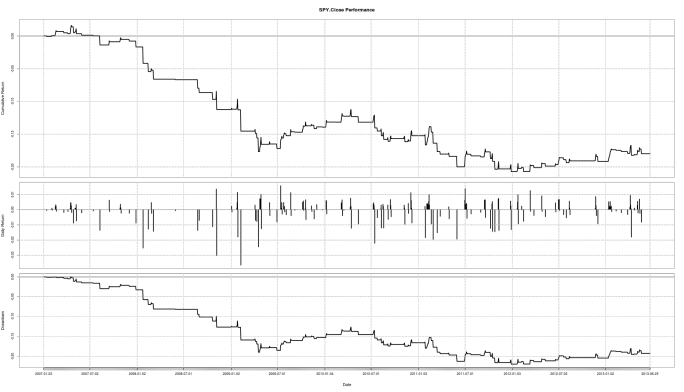
So I just replace the ‘&&’ operator with the multiplication operator, and strung the whole thing together with multiple ifelse statements. Here is what I had:  
**binVec3Day=ifelse(((Cl(SPY)>Op(SPY))),1,0) \* ifelse(lag(Cl(SPY),1)>lag(Op(SPY),1),1,0) \* ifelse(lag(Cl(SPY),2)>lag(Op(SPY),2),1,0)**

Now we need to lag the trading signal vector by a day and get rid of the two NAs which have shown up for our first two days (as it is impossible to know how to trade on the first three days if you need three days of historical data for a trading signal). This is easily done by:

**binVec3Day=lag(binVec3Day,1)**  
**binVec3Day[is.na(binVec3Day)]=0**

Out of curiosity, I ran **sum(binVec3Day)** to see how many trades in total there would be from Jan 1st 2007 to now, June 18 2013, using this strategy. The total came out to 230, which brings me to another very important point. The tests I am conducting now do not include transaction costs or an allowance for slippage. Slippage refers to not getting exactly the price required for the trade. These costs can have a significant impact on strategy returns and I will incorporate them into my strategies later on.

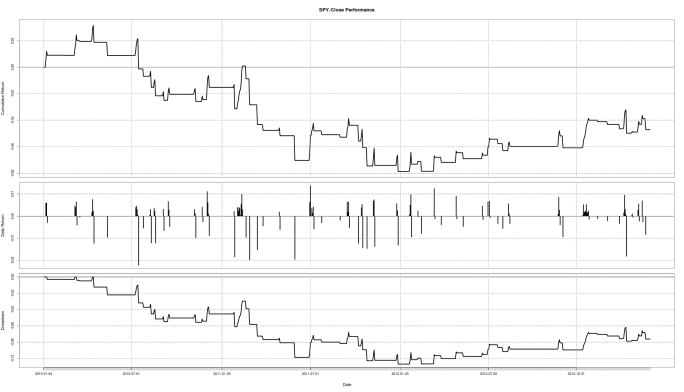
Multiplying by the returns vector and plotting the performance summary, again we see that this strategy did not do too well. This is most likely because our dataset starts from 2007, and thus our strategy has to suffer through the financial crash of 2008. We get a cumulative return from 2007 of approximately -18.00% and an annualized return of approximately -3.00%.

[](http://tradingposts.files.wordpress.com/2013/06/3day.png)

Lets look at a subset of our data, to see if the cumulative returns improve. We can assume an investor decides to enter the market in 2010 using this strategy. To subset our data, we can do this:

**rets2010=strat3Day['2010-01-01/2013-06-01']**

Now we can plot it using performance summary.

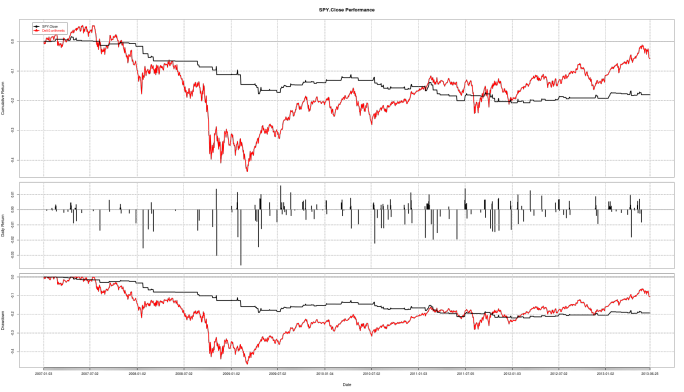
[](http://tradingposts.files.wordpress.com/2013/06/3dayrets2010.png)

We can combine our strategy’s returns with benchmark returns on the same graph for comparison purposes. This is done by:

**combinedReturns=cbind(retVec,Strat3Day)**

(cbind actually stands for column bind, and isn’t short for ‘combined’ which you could mistake from my use of variable names above)

Now when we plot combinedReturns using Performance Summary, we can see that it has underperformed the benchmark so far, in terms of cumulative return. Where our strategy stands out though is the reduced drawdown we suffer in down markets. Annualized volatility is reduced from approximately 18.00% to 4.00%. We can’t really use the Sharpe ratio for comparison here as the excess returns are negative.

[](http://tradingposts.files.wordpress.com/2013/06/combined.png)

You can get the annualized Sharpe ratio and volatility for the S&P by entering:  
**SharpeRatio.annualized(retVec, scale=252)**

**sd.annualized(retVec, scale=252)**

So why are we getting negative cumulative returns? Because the strategy is not in the market enough due to market conditions. Hence the low volatility and negative return. If we can make use of any available big downward moves, we may be able to turn a profit by shorting. Onwards to the next post!

Strategy 1 Extended (Part 2)

*June 26, 2013*

By [s3admq](http://www.r-bloggers.com/author/s3admq/)

(This article was first published on [**Trading and travelling and other things » R**](http://tradingposts.wordpress.com/2013/06/26/strategy-1-extended-part-2/), and kindly contributed to [R-bloggers)](http://www.r-bloggers.com/)

We can extend our strategy and make it more profitable by incorporating short selling. Our annualized volatility will go up, but it will be interesting to see what happens to the annualized return. This is a very simple modification to make.

**1**-First we create a short selling vector:

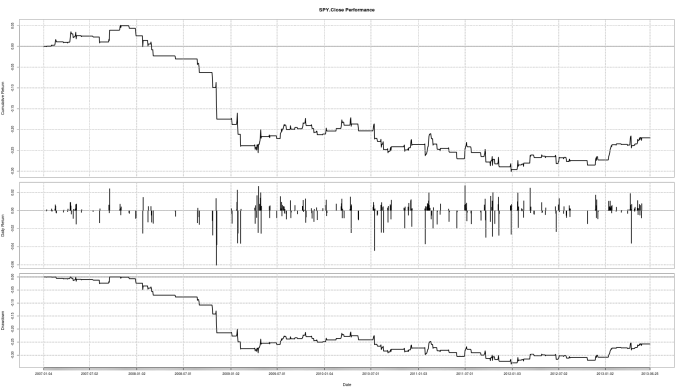
**shortVec=ifelse(((Cl(SPY)<Op(SPY))),-1,0) \* ifelse(lag(Cl(SPY),1)<lag(Op(SPY),1),-1,0) \* ifelse(lag(Cl(SPY),2)<lag(Op(SPY),2),-1,0)** #This is saying that if the stock closes down for three consecutive days, short it.

**2**-As before, we lag it and get rid of the NAs:

**shortVec=lag(binVec3Day,1)**  
**shortVec[is.na(binVec3Day)]=0**

**3**-Now we add the short-signal vector to the lagged and NA-removed long-signal vector we had before:  
**longShortVec=binVec3Day+shortVec**

**4**-And as before, multiply the trading vector with the S&P return vector to get daily strategy returns, then run performance analytics.

[](http://tradingposts.files.wordpress.com/2013/06/lsrets.png)

So with this modification, the annualized volatility rises to 9.40% approximately, and the annualized return falls to -7.10%. Not too good.

The strategy above, and all the subsequent modifications, were momentum based strategies. They rely on large, directed, short-term price movements to be profitable and don’t do too well when the price movements are small and directionless, but the market itself is following an overall trend. The strategies which do well in a trending market are called (!!!) trend strategies. We will look at one in the next post.

Another very important thing I ignored in computing the returns is adjustments for splits and dividends. This can be done using the adjusted price information provided for most equities, and that is what I will be using to calculate returns from now on. By using the adjusted price information however, we are not able to simulate when exactly we enter and exit the market (Open, Close), and that is a tradeoff we’ll have to make for greater convenience. However, I will still be using opening and closing price information to compute the trading signals.

Strategy 2: Riding the SMA Curve

*June 26, 2013*

By [s3admq](http://www.r-bloggers.com/author/s3admq/)

(This article was first published on [**Trading and travelling and other things » R**](http://tradingposts.wordpress.com/2013/06/26/strategy-2-riding-the-sma-curve/), and kindly contributed to [R-bloggers)](http://www.r-bloggers.com/)

This is the least complicated trend strategy in existance. You buy and hold the security as long as the security price is above a XXX-Day Simple Moving Average (SMA), and you can short it if it is below the SMA curve. The important question with this strategy is what the length in days of the SMA should be. We can run a test of different SMAs to see which one is most profitable/ least risky, and then choose accordingly. The intuition in choosing this should be based on an understanding of how long price trends usually last for a given security. This will obviously vary for different markets, different securities and across time periods.

A more technical approach to estimating the optimal variant of the SMA to use could be derived from one of the many parameter optimisation techniques avaiable through R, or through coding it yourself. These run the risk of curve-fitting, but as long as you are aware of the dangers associated with that, this could be one thing for you to try. I’ve found through experience that the 200-Day SMA works best for the S&P 500 as a whole, so I will be running this backtest using that.

Let’s say that if the market closes above its 200-Day Daily High SMA on any given day, we go long the next day and if it closes below it 200-Day Daily Low, we go short.

**1**-Get the data:  
**getSymbols(‘SPY’)**

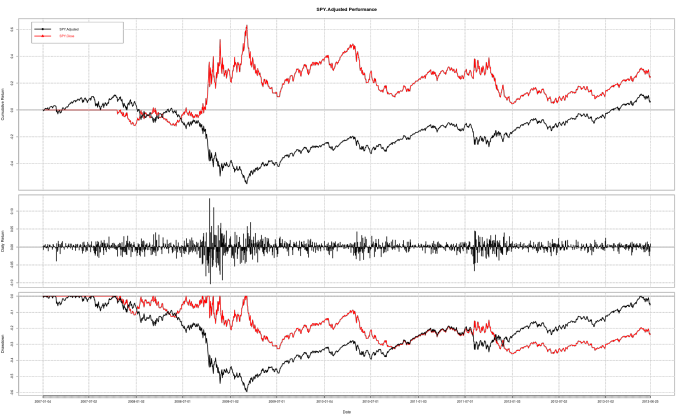
**2**-Calculate the 200-Day SMAs:  
**smaHi200=SMA(Hi(SPY),200)**  
**smaLo200=SMA(Lo(SPY),200)**

**3**-Calculate the lagged trading signal vector:  
**binVec=lag(ifelse(Cl(SPY)>smaHi200,1,0)+ifelse(Cl(SPY)<smaLo200,-1,0),1)**

**4**-Get rid of the NAs:  
**binVec[is.na(binVec)]=0**

**5**-Calculate returns vector and multiply out the trading vector with the returns vector to get the strategy return:  
**rets=diff(log(Ad(SPY)))**  
**stratRets=binVec\*rets**

**6**-Run performance analytics:  
**charts.PerformanceSummary(cbind(rets,stratRets))**  
**Performance(stratRets)**

[](http://tradingposts.files.wordpress.com/2013/06/sma200ls.png)

Note: The Performance function I got from somewhere on the internet. I can’t remember where exactly so unfortunately I can’t directly credit the source. Regardless, I’m thankful to whoever wrote it. Here is the code:

Performance <- [**function**](http://inside-r.org/r-doc/base/function)(x) {

cumRetx = Return.cumulative(x)

annRetx = Return.annualized(x, [**scale**](http://inside-r.org/r-doc/base/scale)=252)

sharpex = SharpeRatio.annualized(x, [**scale**](http://inside-r.org/r-doc/base/scale)=252)

winpctx = [**length**](http://inside-r.org/r-doc/base/length)(x[x > 0])/length(x[x != 0])

annSDx = sd.annualized(x, [**scale**](http://inside-r.org/r-doc/base/scale)=252)

DDs <- findDrawdowns(x)

maxDDx = [**min**](http://inside-r.org/r-doc/base/min)(DDs$return)

maxLx = [**max**](http://inside-r.org/r-doc/base/max)(DDs$length)

Perf = [**c**](http://inside-r.org/r-doc/base/c)(cumRetx, annRetx, sharpex, winpctx, annSDx, maxDDx, maxLx)

[**names**](http://inside-r.org/r-doc/base/names)(Perf) = [**c**](http://inside-r.org/r-doc/base/c)("Cumulative Return", "Annual Return","Annualized Sharpe Ratio",

"Win %", "Annualized Volatility", "Maximum Drawdown", "Max Length Drawdown")

[**return**](http://inside-r.org/r-doc/base/return)(Perf)

}

[Created by Pretty R at inside-R.org](http://www.inside-r.org/pretty-r)

**We get the following results (our sample period is 2007-01-01 to 2013-06-19):**

**Market**  
Cumulative Return: 0.10875991   Annual Return: 0.01613934

Annualized Sharpe Ratio: 0.06700363   Win %: 0.55157505

Annualized Volatility: 0.24087258   Maximum Drawdown: -0.59577736

Max Length Drawdown: 1411.00000000

**Strategy**  
Cumulative Return: 0.30075987  Annual Return: 0.04159395

Annualized Sharpe Ratio: 0.17975121 Win %: 0.53215078

Annualized Volatility: 0.23139732   Maximum Drawdown: -0.35921405

Max Length Drawdown: 1078.00000000

So this strategy performed remarkably well compared to the market. 1-0 for Trend strategies! Looking at the graphs it’s easy to see the points where the strategy mirrors the market returns (and hence is short) and where the strategy follows the market returns (and hence is long). So does that mean we can start trading using this algorithm and make money? Not really- one important point to keep in mind is that backtesting can only be used to reject strategies, not to accept them. It is possible that this strategy can make money going forward, but who really know what the market will do? At least we can’t outright reject this strategy as useless.

What we are essentially saying is that under the conditions we used, our strategy was profitable. If those conditions were to continue, or repeat themselves, than we would have a profitable strategy in our hands. Are those conditions likely to repeat themselves? If market reactions to different stimuli are consistent across time, and if those stimuli re-occur then yes, perhaps. There’s also something to be said here about interactions between the different reactions or stimuli and on their subsequent effects on market outcomes. Is this something we can test for? Maybe, but its not something I’ll be getting into for now.

Parameter Optimization for Strategy 2

*June 27, 2013*

By [s3admq](http://www.r-bloggers.com/author/s3admq/)

(This article was first published on [**Trading and travelling and other things » R**](http://tradingposts.wordpress.com/2013/06/27/parameter-optimization-for-strategy-2/), and kindly contributed to [R-bloggers)](http://www.r-bloggers.com/)

Now, let’s try some parameter optimisation for the SMA strategy! There probably are functions out there on R which I can use to do this, but I figured it would take me as long to actually code it as it would to find something usable on the internet, and I enjoy coding much more than looking stuff up on the internet.

My aim is to find out which SMA is the best to use for going long, and which SMA is the best to use for going short on the S&P 500. Ideally, I should optimize the short SMA for each long SMA (or vice-versa) to find the best combination, but I don’t think optimizing them independently (as I did here) would make much of a difference in this case. This is the code I wrote:

optimizeSMA=[**function**](http://inside-r.org/r-doc/base/function)(mainVector,returnsVector,smaInit=3,smaEnd=200,long=**TRUE**){

bestSMA=0

bestSharpe=0

**for**( i **in** smaInit:smaEnd){

smaVec=[SMA](http://inside-r.org/packages/cran/sma)(mainVector,i)

**if**(long==T){

binVec=[**lag**](http://inside-r.org/r-doc/stats/lag)(**[ifelse](http://inside-r.org/r-doc/base/ifelse" \t "_blank)**(mainVector>smaVec,1,0),1)

binVec[**[is.na](http://inside-r.org/r-doc/base/is.na" \t "_blank)**(binVec)]=0

stratRets=binVec\*returnsVector

sharpe=SharpeRatio.annualized(stratRets, [**scale**](http://inside-r.org/r-doc/base/scale)=252)

**if**(sharpe>bestSharpe){

bestSMA=i

bestSharpe=sharpe

}

}**else**{

binVec=[**lag**](http://inside-r.org/r-doc/stats/lag)(**[ifelse](http://inside-r.org/r-doc/base/ifelse" \t "_blank)**(mainVector<smaVec,-1,0),1)

binVec[**[is.na](http://inside-r.org/r-doc/base/is.na" \t "_blank)**(binVec)]=0

stratRets=binVec\*returnsVector

sharpe=SharpeRatio.annualized(stratRets, [**scale**](http://inside-r.org/r-doc/base/scale)=252)

**if**(sharpe>bestSharpe){

bestSMA=i

bestSharpe=sharpe

}

}

}

[**print**](http://inside-r.org/r-doc/base/print)(**[cbind](http://inside-r.org/r-doc/base/cbind" \t "_blank)**(bestSMA, bestSharpe))

}

[Created by Pretty R at inside-R.org](http://www.inside-r.org/pretty-r)

It is pretty straight-forward and self-explanatory. It initiates a loop which goes through each SMA from smaInit to smaEnd and stores the one with the highest Sharpe ratio. For more complicated strategies, we will need to do a little bit more heavy-lifting when it comes to parameter optimization. This code maximizes the Sharpe ratio, but you can easily modify it to maximize returns, minimize volatility, etc. The highest Sharpe ratio SMA to use for the long position is the 70-Day SMA and for the short position is the 84-day SMA.

After running the strategy with the optimized parameters, these are the performance results:

**Optimized:**

Cumulative Return: 0.31104898   Annual Return: 0.04286661

Annualized Sharpe Ratio: 0.18405777   Win %: 0.53078556

Annualized Volatility: 0.23289757    Maximum Drawdown: -0.28943309

Max Length Drawdown: 1078.00000000

Not a huge difference from what we had before. And the little bit of performance improvement that we achieved is probably more a result of curve-fitting than anything else. If your initial parameter values conform with some market intuition -and thus capture most of the obtainable market return- parameter optimization will not be that helpful in a paper-trading implementation of your strategy, as the improvements will mostly be due to curve-fitting the historical data.

Portfolio Optimization

*June 27, 2013*

By [s3admq](http://www.r-bloggers.com/author/s3admq/)

(This article was first published on [**Trading and travelling and other things » R**](http://tradingposts.wordpress.com/2013/06/27/portfolio-optimization/), and kindly contributed to [R-bloggers)](http://www.r-bloggers.com/)

Changing tracks, I want to now look at portfolio optimization. Although this is very different from developing trading strategies, it is useful to know how to construct minimum-variance portfolios and the like, if only for curiosity’s sake. Also, just a -I hope unnecessary- note, portfolio optimization and parameter optimization (which I covered in the last post) are two completely different things.

Minimum-variance portfolio optimization has a lot of problems associated with it, but it makes for a good starting point as it is the most commonly discussed optimization technique in classroom-finance. One of my biggest issues is with the measurement of risk via volatility. Security out-performance contributes as much to volatility -hence risk- as security under-performance, which ideally shouldn’t be the case.

First, install the package tseries:  
**install.packages(‘tseries’)**

The function of interest is **portfolio.optim()**. I decided to write my own function to enter in a vector of tickers, start and end dates for the dataset, min and max weight constraints and short-selling constraints. This function first processes the data and then passes it to portfolio.optim to determine the minimum variance portfolio for a given level of return. It then cycles through increasingly higher returns to check how high the Sharpe ratio can go.

Here is the code with comments:

minVarPortfolio= [**function**](http://inside-r.org/r-doc/base/function)(tickers,[**start**](http://inside-r.org/r-doc/stats/start)='2000-01-01',[**end**](http://inside-r.org/r-doc/stats/end)=**[Sys.Date](http://inside-r.org/r-doc/base/Sys.Date" \t "_blank)**(),

riskfree=0,short=**TRUE**,lowestWeight=-1,highestWeight=1){

*# Load up the package*

[**require**](http://inside-r.org/r-doc/base/require)([tseries](http://inside-r.org/packages/cran/tseries" \t "_blank))

*#Initialize all the variables we will be using. returnMatrix is*

*#initailized as a vector,with length equal to one of the input*

*#ticker vectors (dependent on the start and end dates).*

*#Sharpe is set to 0. The weights vector is set equal in*

*#length to the number of tickers. The portfolio is set to*

*#NULL. A 'constraint' variable is created to pass on the*

*#short parameter to the portfolio.optim function. And vectors*

*#are created with the low and high weight restrictions, which*

*#are then passed to the portfolio.optim function as well. ##*

returnMatrix=[**vector**](http://inside-r.org/r-doc/base/vector)([**length**](http://inside-r.org/r-doc/base/length)=[**length**](http://inside-r.org/r-doc/base/length)(getSymbols(tickers[1],

auto.assign=**FALSE**,from=**[start](http://inside-r.org/r-doc/stats/start" \t "_blank)**,to=[**end**](http://inside-r.org/r-doc/stats/end))))

sharpe=0

[**weights**](http://inside-r.org/r-doc/stats/weights)=[**vector**](http://inside-r.org/r-doc/base/vector)(,[**length**](http://inside-r.org/r-doc/base/length)(tickers))

port=**NULL**

constraint=short

lowVec=[**rep**](http://inside-r.org/r-doc/base/rep)(lowestWeight,[**length**](http://inside-r.org/r-doc/base/length)(tickers))

hiVec=[**rep**](http://inside-r.org/r-doc/base/rep)(highestWeight,[**length**](http://inside-r.org/r-doc/base/length)(tickers))

*#This is a for-loop which cycles through the tickers, calculates*

*#their return, and stores the returns in a matrix, adding*

*#the return vector for each ticker to the matrix*

**for**(i **in** 1:[**length**](http://inside-r.org/r-doc/base/length)(tickers)){

temp=getSymbols(tickers[i],auto.assign=**FALSE**,from=**[start](http://inside-r.org/r-doc/stats/start" \t "_blank)**,to=[**end**](http://inside-r.org/r-doc/stats/end))

**if**(i==1){

returnMatrix=[**diff**](http://inside-r.org/r-doc/base/diff)([**log**](http://inside-r.org/r-doc/base/log)(Ad(temp)))

}

**else**{

returnMatrix=**[cbind](http://inside-r.org/r-doc/base/cbind" \t "_blank)**(returnMatrix,[**diff**](http://inside-r.org/r-doc/base/diff)([**log**](http://inside-r.org/r-doc/base/log)(Ad(temp))))

}

}

returnMatrix[**[is.na](http://inside-r.org/r-doc/base/is.na" \t "_blank)**(returnMatrix)]=0

it

*#This for-loop cycles through returns to test the portfolio.optim function*

*#for the highest Sharpe ratio.*

**for**(j **in** 1:100){

*#Stores the log of the return in retcalc*

retcalc=[**log**](http://inside-r.org/r-doc/base/log)((1+j/100))

retcalc=retcalc/252

[**print**](http://inside-r.org/r-doc/base/print)([**paste**](http://inside-r.org/r-doc/base/paste)("Ret Calc:",retcalc))

*#Tries to see if the specified return from retcalc can result*

*#in an efficient portfolio*

[**try**](http://inside-r.org/r-doc/base/try)(port<-portfolio.optim(returnMatrix,pm=retcalc,shorts=constraint,

reslow=lowVec,reshigh=hiVec,riskfree=riskfree),silent=T)

*#If the portfolio exists, it is compared against previous portfolios*

*#for different returns using the #Sharpe ratio. If it has the highest*

*#Sharpe ratio, it is stored and the old one is discarded.*

**if**(!**[is.null](http://inside-r.org/r-doc/base/is.null" \t "_blank)**(port)){

[**print**](http://inside-r.org/r-doc/base/print)('Not Null')

[**sd**](http://inside-r.org/r-doc/stats/sd)=port$ps

tSharpe=((retcalc-riskfree)/sd)

[**print**](http://inside-r.org/r-doc/base/print)([**paste**](http://inside-r.org/r-doc/base/paste)("Sharpe",tSharpe))

**if**(tSharpe>sharpe){

sharpe=tSharpe

[**weights**](http://inside-r.org/r-doc/stats/weights)=port$pw

}}

}

[**print**](http://inside-r.org/r-doc/base/print)([**paste**](http://inside-r.org/r-doc/base/paste)('Sharpe:', sharpe))

[**print**](http://inside-r.org/r-doc/base/print)(**[rbind](http://inside-r.org/r-doc/base/rbind" \t "_blank)**(tickers,[**weights**](http://inside-r.org/r-doc/stats/weights)))

[**return**](http://inside-r.org/r-doc/base/return)(returnMatrix)

}

[Created by Pretty R at inside-R.org](http://www.inside-r.org/pretty-r)

This code works fine except for when the restrictions are too strict, the portfolio.optim function can’t find a minimum variance portfolio. This happens if the optimum portfolio has negative returns, which my code doesn’t test for. For this reason, I wanted to try out other ways of finding the highest Sharpe portfolio. There are numerous tutorials out there on how to do this. Some of them are:

1-<http://blog.streeteye.com/blog/2012/01/portfolio-optimization-and-efficient-frontiers-in-r/>  
2-<http://quantivity.wordpress.com/2011/04/17/minimum-variance-portfolios/>  
3-<http://www.rinfinance.com/RinFinance2009/presentations/yollin_slides.pdf>  
4-<http://systematicinvestor.wordpress.com/2013/03/22/maximum-sharpe-portfolio/>  
5-<http://alphaism.wordpress.com/2012/05/04/finding-efficient-frontier-and-optimal-portfolio-in-r/>

After I run my function, with the following tickers and constraints:

**matrix=minVarPortfolio(c(‘NVDA’, ‘YHOO’, ‘GOOG’, ‘CAT’, ‘BNS’, ‘POT’, ‘STO’, ‘MBT’ ,’SNE’),lowestWeight=0,highestWeight=0.2,start=’2000-01-01′, end=’2013-06-01′)**

This is the output I get:

[1] “Sharpe: 0.177751547083007″

tickers                ”NVDA”                                   “YHOO”                        ”GOOG”  
weights “-1.58276161084957e-19″      ”2.02785605793095e-17″           “0.2″  
tickers                 “CAT”                                       “BNS”                           “POT”  
weights “0.104269676769825″                           “0.2″                             “0.2″

tickers                 “STO”                                       “MBT”  
weights “0.189985091184918″             “0.105745232045257″

tickers                 “SNE”  
weights “-2.85654465380669e-17″

The ‘e-XX’ weights basically indicate a weighting of zero on that particular security (NVDA, YHOO and SNE above). In the next post I will look at how all this can be done using a package called ‘fPortfolio’. Happy trading!

Looking out for volatility

*June 26, 2013*

By [s3admq](http://www.r-bloggers.com/author/s3admq/)

(This article was first published on [**Trading and travelling and other things » R**](http://tradingposts.wordpress.com/2013/06/26/looking-out-for-volatility/), and kindly contributed to [R-bloggers)](http://www.r-bloggers.com/)

Let’s do an easy experiment. Lets caluclate the 25-day rolling volatility of the S&P 500 from 2007 onwards.

1-Get the data:  
**getSymbols(‘SPY’,from=’2007/01/01′)**

2-Run the volatility function from the package TTR (comes along with quantmod):  
**vol=volatility(SPY,n=25,N=252,calc=’close’)**  
#n=25 means we want 25 day rolling volatility. N=252 means we are taking a year as 252 days. calc=’close’ indicates that we want to calculate Close to Close volatility. If you look at the help page for the volatility function, there are several different calc=” parameters available.

3-We can now plot this using **chartSeries(vol)**. Notice the huge spike in volatility in 2008.

[](http://tradingposts.files.wordpress.com/2013/06/vol.png)

# Simple Moving Average Strategy with a Volatility Filter

*April 18, 2012*

By [rbresearch](http://www.r-bloggers.com/author/rbresearch/" \o "Posts by rbresearch)

(This article was first published on **[rbresearch » R](http://rbresearch.wordpress.com/2012/04/18/simple-moving-average-strategy-with-a-volatility-filter/)**, and kindly contributed to [R-bloggers)](http://www.r-bloggers.com/)

I would describe my trading approach as systematic long term trend following. A trend following strategy can be difficult mentally to trade after experiencing multiple consecutive losses when a trade reverses due to a volatility spike or the trend reverses. Volatility tends to increase when prices fall. This is not good for a long only trend following strategy, especially when initially entering trades.

Can adding a volatility filter to a simple system improve performance?

SMA System with Volatility Filter Rules

* Buy Rule: Go long if close is greater than the N period SMA and a volatility measure is less than its median over the last N periods.
* Exit Rule: Exit if long and close is less than the N period SMA

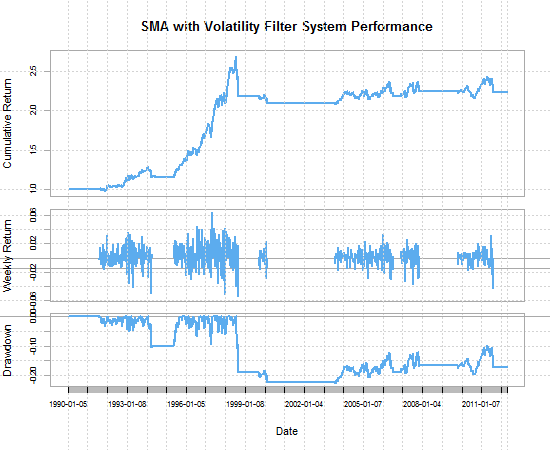
SMA System without Volatility Filter Rules

* Buy Rule: Go long if close is greater than the N period SMA
* Exit Rule: Exit if close is less than the N period SMA

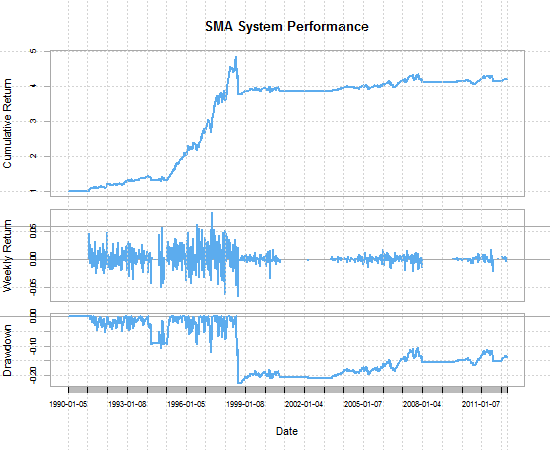
For this test, my volatility measure is the 52 period standard deviation of the 1 period change of close prices and I will use a 52 period SMA.

I will test the strategy on the total return series of the S&P500 using weekly prices from 1/1/1990 to 4/17/2012.

yuck… the equity curves look pretty good up until 1999, then not so good after that.

[](https://rbresearch.files.wordpress.com/2012/04/rplot-sma-vol-filter.png)

rbresearch

[](https://rbresearch.files.wordpress.com/2012/04/rplot-sma-system.png)

rbresearch

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | CAGR | maxDD | MAR | # Trades | Ending Equity | Percent Winning Trades |
| SMA with Volatility Filter | 4.369174 | -22.3993 | 0.195059 | 34 | $239,104.70 | 58.82 |
| SMA System | 7.442673 | -22.2756 | 0.334119 | 57 | $464,198.80 | 53.57 |

This test shows that adding a volatility filter to our entries can actually hinder performance. Keep in mind this is ny no means an exhaustive test on a single instrument. I also chose the 52 period SMA and SDEV somewhat arbitrarily because it represents a year.

Reading through trading forums, it is clear to see that people are in search of the “holy grail” trading system. Some people claim to have found the “holy grail” system, but that system is usually combination of 10+ indicators and rules that say “use indicator A, B, and C when the market is doing X or use indicators D, E, and F when the market is doing Y.” Beware of these “filters” and always test yourself.

Stay tuned for future posts that will look at adding a similar filter on a multiple instrument test.

What have you found with adding entry filters to trading systems?

[**require**](http://inside-r.org/r-doc/base/require)([PerformanceAnalytics](http://inside-r.org/packages/cran/PerformanceAnalytics" \t "_blank))

[**require**](http://inside-r.org/r-doc/base/require)(quantstrat)

sym.st = "GSPC"

currency("USD")

stock(sym.st, currency="USD",multiplier=1)

getSymbols("^GSPC", src='yahoo', index.class=[**c**](http://inside-r.org/r-doc/base/c)("POSIXt","POSIXct"), from='1990-01-01')

GSPC <- to.weekly(GSPC,indexAt='lastof',drop.time=**TRUE**)

*#Custom Order Sizing Function to trade percent of equity based on a stopsize*

osPCTEQ <- [**function**](http://inside-r.org/r-doc/base/function)([**timestamp**](http://inside-r.org/r-doc/utils/timestamp), orderqty, [portfolio](http://inside-r.org/packages/cran/portfolio), [**symbol**](http://inside-r.org/r-doc/grDevices/symbol), ruletype, ...){

tempPortfolio <- getPortfolio(portfolio.st)

dummy <- updatePortf([Portfolio](http://inside-r.org/packages/cran/portfolio)=portfolio.st, [**Dates**](http://inside-r.org/r-doc/base/Dates)=[**paste**](http://inside-r.org/r-doc/base/paste)('::',**[as.Date](http://inside-r.org/r-doc/base/as.Date" \t "_blank)**([**timestamp**](http://inside-r.org/r-doc/utils/timestamp)),sep=''))

trading.pl <- [**sum**](http://inside-r.org/r-doc/base/sum)(getPortfolio(portfolio.st)$summary$Realized.PL) *#change to ..$summary$Net.Trading.PL for Total Equity Position Sizing*

[**assign**](http://inside-r.org/r-doc/base/assign)([**paste**](http://inside-r.org/r-doc/base/paste)("portfolio.",portfolio.st,sep=""),tempPortfolio,pos=.blotter)

total.equity <- initEq+trading.pl

DollarRisk <- total.equity \* trade.percent

ClosePrice <- [**as.numeric**](http://inside-r.org/r-doc/base/as.numeric)(Cl(mktdata[[**timestamp**](http://inside-r.org/r-doc/utils/timestamp),]))

mavg <- [**as.numeric**](http://inside-r.org/r-doc/base/as.numeric)(mktdata$SMA52[[**timestamp**](http://inside-r.org/r-doc/utils/timestamp),])

sign1 <- [**ifelse**](http://inside-r.org/r-doc/base/ifelse)(ClosePrice > mavg, 1, -1)

sign1[**[is.na](http://inside-r.org/r-doc/base/is.na" \t "_blank)**(sign1)] <- 1

Posn = getPosQty([Portfolio](http://inside-r.org/packages/cran/portfolio" \t "_blank) = portfolio.st, Symbol = sym.st, [Date](http://inside-r.org/packages/cran/date) = [**timestamp**](http://inside-r.org/r-doc/utils/timestamp))

StopSize <- [**as.numeric**](http://inside-r.org/r-doc/base/as.numeric)(mktdata$SDEV[[**timestamp**](http://inside-r.org/r-doc/utils/timestamp),]\*StopMult) *#Stop = SDAVG \* StopMult !Must have SDAVG or other indictor to determine stop size*

orderqty <- [**ifelse**](http://inside-r.org/r-doc/base/ifelse)(Posn == 0, sign1\*[**round**](http://inside-r.org/r-doc/base/round)(DollarRisk/StopSize), 0) *# number contracts traded is equal to DollarRisk/StopSize*

[**return**](http://inside-r.org/r-doc/base/return)(orderqty)

}

*#Function that calculates the n period standard deviation of close prices.*

*#This is used in place of ATR so that I can use only close prices.*

SDEV <- [**function**](http://inside-r.org/r-doc/base/function)(x, n){

sdev <- runSD(x, n, [**sample**](http://inside-r.org/r-doc/base/sample) = **FALSE**)

[**colnames**](http://inside-r.org/r-doc/base/colnames)(sdev) <- "SDEV"

reclass(sdev,x)

}

*#Custom indicator function*

RB <- [**function**](http://inside-r.org/r-doc/base/function)(x,n){

x <- x

roc <- ROC(x, n=1, type="discrete")

[**sd**](http://inside-r.org/r-doc/stats/sd) <- runSD(roc,n, [**sample**](http://inside-r.org/r-doc/base/sample)= **FALSE**)

[**sd**](http://inside-r.org/r-doc/stats/sd)[[**is.na**](http://inside-r.org/r-doc/base/is.na)(**[sd](http://inside-r.org/r-doc/stats/sd" \t "_blank)**)] <- 0

med <- runMedian(**[sd](http://inside-r.org/r-doc/stats/sd" \t "_blank)**,n)

med[**[is.na](http://inside-r.org/r-doc/base/is.na" \t "_blank)**(med)] <- 0

mavg <- [SMA](http://inside-r.org/packages/cran/sma)(x,n)

[signal](http://inside-r.org/packages/cran/signal) <- [**ifelse**](http://inside-r.org/r-doc/base/ifelse)(**[sd](http://inside-r.org/r-doc/stats/sd" \t "_blank)** < med & x > mavg,1,0)

[**colnames**](http://inside-r.org/r-doc/base/colnames)([signal](http://inside-r.org/packages/cran/signal)) <- "RB"

*#ret <- cbind(x,roc,sd,med,mavg,signal)*

*#colnames(ret) <- c("close","roc","sd","med","mavg","lowvol")*

reclass([signal](http://inside-r.org/packages/cran/signal" \t "_blank),x)

}

initDate='1900-01-01'

initEq <- 100000

trade.percent <- .05 *#percent risk used in sizing function*

StopMult = 1 *#stop size used in sizing function*

*#Name the portfolio and account*

portfolio.st='RBtest'

account.st='RBtest'

*#Initialization*

initPortf(portfolio.st, [**symbols**](http://inside-r.org/r-doc/graphics/symbols)=sym.st, initPosQty=0, initDate=initDate, currency="USD")

initAcct(account.st,portfolios=portfolio.st, initDate=initDate, initEq=initEq)

initOrders([portfolio](http://inside-r.org/packages/cran/portfolio" \t "_blank)=portfolio.st,initDate=initDate)

*#Name the strategy*

stratRB <- strategy('RBtest')

*#Add indicators*

*#The first indicator is the 52 period SMA*

*#The second indicator is the RB indicator. The RB indicator returns a value of 1 when close > SMA & volatility < runMedian(volatility, n = 52)*

stratRB <- add.indicator(strategy = stratRB, name = "SMA", arguments = [**list**](http://inside-r.org/r-doc/base/list)(x = [**quote**](http://inside-r.org/r-doc/base/quote)(Cl(mktdata)), n=52), label="SMA52")

stratRB <- add.indicator(strategy = stratRB, name = "RB", arguments = [**list**](http://inside-r.org/r-doc/base/list)(x = [**quote**](http://inside-r.org/r-doc/base/quote)(Cl(mktdata)), n=52), label="RB")

stratRB <- add.indicator(strategy = stratRB, name = "SDEV", arguments = [**list**](http://inside-r.org/r-doc/base/list)(x = [**quote**](http://inside-r.org/r-doc/base/quote)(Cl(mktdata)), n=52), label="SDEV")

*#Add signals*

*#The buy signal is when the RB indicator crosses from 0 to 1*

*#The exit signal is when the close crosses below the SMA*

stratRB <- add.signal(strategy = stratRB, name="sigThreshold", arguments = [**list**](http://inside-r.org/r-doc/base/list)(threshold=1, column="RB",relationship="gte", cross=**TRUE**),label="RB.gte.1")

stratRB <- add.signal(strategy = stratRB, name="sigCrossover", arguments = [**list**](http://inside-r.org/r-doc/base/list)(columns=[**c**](http://inside-r.org/r-doc/base/c)("Close","SMA52"),relationship="lt"),label="Cl.lt.SMA")

*#Add rules*

stratRB <- add.rule(strategy = stratRB, name='ruleSignal', arguments = [**list**](http://inside-r.org/r-doc/base/list)(sigcol="RB.gte.1", sigval=**TRUE**, orderqty=1000, ordertype='market', orderside='long', osFUN = 'osPCTEQ', pricemethod='market', [**replace**](http://inside-r.org/r-doc/base/replace)=**FALSE**), type='enter', path.dep=**TRUE**)

stratRB <- add.rule(strategy = stratRB, name='ruleSignal', arguments = [**list**](http://inside-r.org/r-doc/base/list)(sigcol="Cl.lt.SMA", sigval=**TRUE**, orderqty='all', ordertype='market', orderside='long', pricemethod='market',TxnFees=0), type='exit', path.dep=**TRUE**)

*# Process the indicators and generate trades*

start\_t<-**[Sys.time](http://inside-r.org/r-doc/base/Sys.time" \t "_blank)**()

out<-[**try**](http://inside-r.org/r-doc/base/try)(applyStrategy(strategy=stratRB , portfolios=portfolio.st))

end\_t<-**[Sys.time](http://inside-r.org/r-doc/base/Sys.time" \t "_blank)**()

[**print**](http://inside-r.org/r-doc/base/print)("Strategy Loop:")

[**print**](http://inside-r.org/r-doc/base/print)(end\_t-start\_t)

start\_t<-**[Sys.time](http://inside-r.org/r-doc/base/Sys.time" \t "_blank)**()

updatePortf([Portfolio](http://inside-r.org/packages/cran/portfolio)=portfolio.st,[**Dates**](http://inside-r.org/r-doc/base/Dates)=[**paste**](http://inside-r.org/r-doc/base/paste)('::',[**as.Date**](http://inside-r.org/r-doc/base/as.Date)([**Sys.time**](http://inside-r.org/r-doc/base/Sys.time)()),sep=''))

end\_t<-**[Sys.time](http://inside-r.org/r-doc/base/Sys.time" \t "_blank)**()

[**print**](http://inside-r.org/r-doc/base/print)("updatePortf execution time:")

[**print**](http://inside-r.org/r-doc/base/print)(end\_t-start\_t)

chart.Posn([Portfolio](http://inside-r.org/packages/cran/portfolio" \t "_blank)=portfolio.st,Symbol=sym.st)

*#Update Account*

updateAcct(account.st)

*#Update Ending Equity*

updateEndEq(account.st)

*#ending equity*

getEndEq(account.st, [**Sys.Date**](http://inside-r.org/r-doc/base/Sys.Date)()) + initEq

tstats <- tradeStats([Portfolio](http://inside-r.org/packages/cran/portfolio)=portfolio.st, Symbol=sym.st)

*#View order book to confirm trades*

*#getOrderBook(portfolio.st)*

*#Trade Statistics for CAGR, Max DD, and MAR*

*#calculate total equity curve performance Statistics*

ec <- [**tail**](http://inside-r.org/r-doc/utils/tail)([**cumsum**](http://inside-r.org/r-doc/base/cumsum)(getPortfolio(portfolio.st)$summary$Net.Trading.PL),-1)

ec$initEq <- initEq

ec$totalEq <- ec$Net.Trading.PL + ec$initEq

ec$maxDD <- ec$totalEq/cummax(ec$totalEq)-1

ec$logret <- ROC(ec$totalEq, n=1, type="continuous")

ec$logret[[**is.na**](http://inside-r.org/r-doc/base/is.na)(ec$logret)] <- 0

Strat.Wealth.Index <- [**exp**](http://inside-r.org/r-doc/base/exp)(**[cumsum](http://inside-r.org/r-doc/base/cumsum" \t "_blank)**(ec$logret)) *#growth of $1*

period.count <- [**NROW**](http://inside-r.org/r-doc/base/NROW)(ec)-104 *#Use 104 because there is a 104 week lag for the 52 week SD and 52 week median of SD*

year.count <- period.count/52

maxDD <- [**min**](http://inside-r.org/r-doc/base/min)(ec$maxDD)\*100

totret <- [**as.numeric**](http://inside-r.org/r-doc/base/as.numeric)(last(ec$totalEq))/as.numeric(first(ec$totalEq))

CAGR <- (totret^(1/year.count)-1)\*100

[MAR](http://inside-r.org/packages/cran/mAr) <- CAGR/abs(maxDD)

Perf.Stats <- [**c**](http://inside-r.org/r-doc/base/c)(CAGR, maxDD, [MAR](http://inside-r.org/packages/cran/mAr))

[**names**](http://inside-r.org/r-doc/base/names)(Perf.Stats) <- [**c**](http://inside-r.org/r-doc/base/c)("CAGR", "maxDD", "MAR")

Perf.Stats

*#write.zoo(mktdata, file = "E:\\a.csv")*

charts.PerformanceSummary(ec$logret, wealth.index = **TRUE**, colorset = "steelblue2", main = "SMA with Volatility Filter System Performance")

[Created by Pretty R at inside-R.org](http://www.inside-r.org/pretty-r)

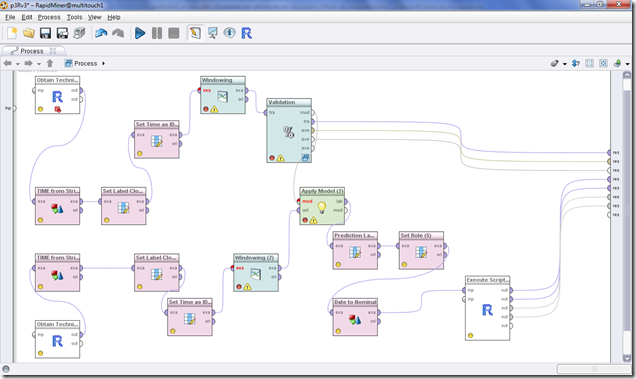
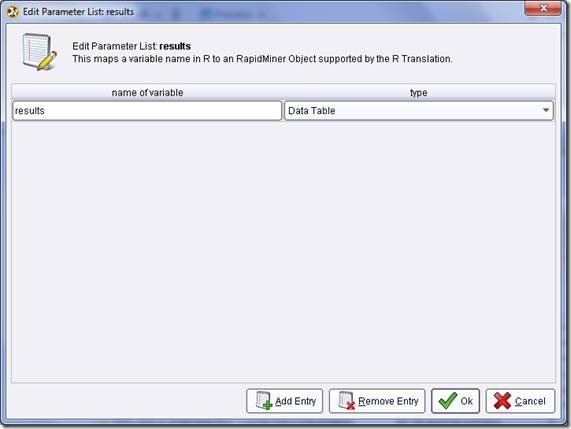
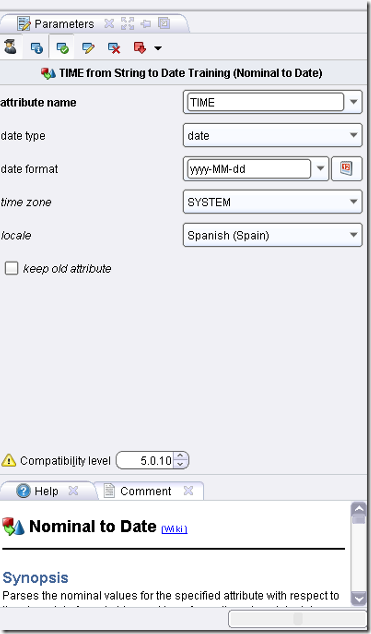
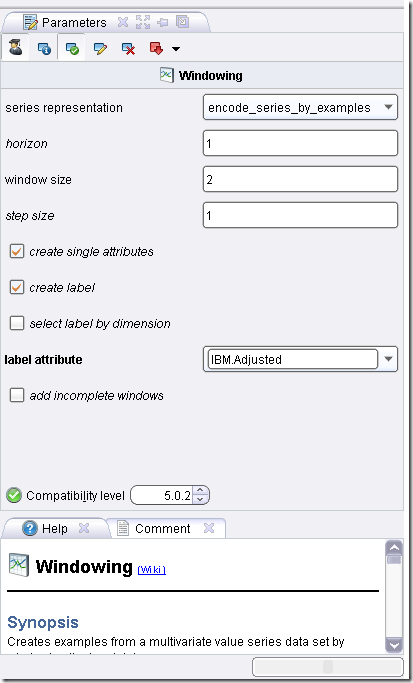
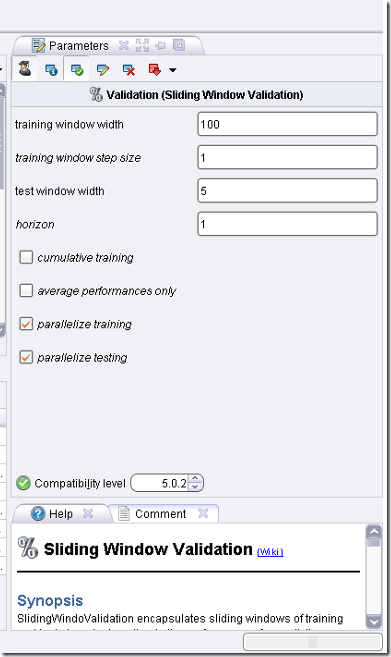
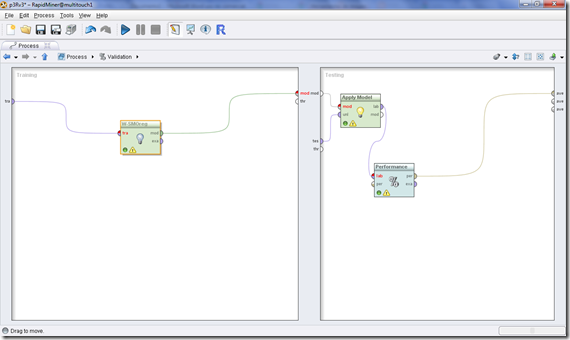
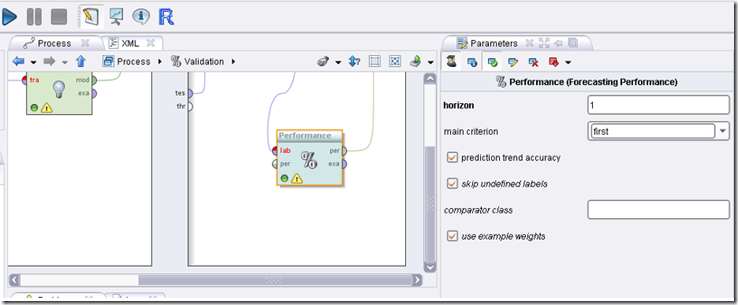
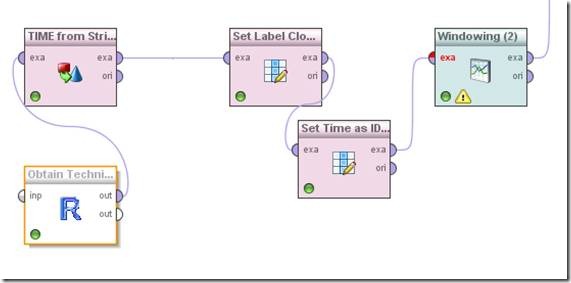
Disclaimer: Past results do not guarantee future returns. Information on this website is for informational purposes only and does not offer advice to buy or sell any securities.

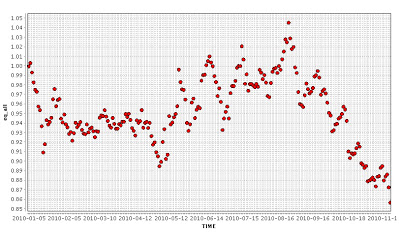
Rapidminer + R Example for Trading

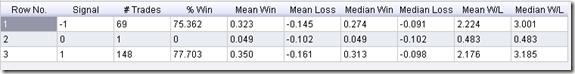
*November 18, 2010*

By [a Physicist](http://www.r-bloggers.com/author/a-physicist/)

(This article was first published on [**a Physicist in Wall Street**](http://www.aphysicistinwallstreet.com/2010/11/example-rapidminer-r-for-trading.html), and kindly contributed to [R-bloggers)](http://www.r-bloggers.com/)

RapidMiner + R is an advanced tool that can be used to analyze trading strategies, In order to check its power I made a simple example using an algorithm based on a support vector machine for predicting the next day's price and based on it I generated buying and selling signals. I have integrated quant indicators, SVM, and inally the strategy is evaluated.   
The requirements needed to build the model are, of course, RapidMiner, Weka extension, time series extension and the R extension. This requires installing R with quantmod, TTR and PerformanceAnalytics packages. There is a thread to solve any problem [here](http://rapid-i.com/rapidforum/index.php/topic,2612.0.html)  
To be able to reproduce my results I will detail each of the modules of the following figure:   
[](http://lh5.ggpht.com/_upmtKBKjH54/TOWpGy1A48I/AAAAAAAAAt8/CsrDhISd_AE/s1600-h/flow%5B4%5D.png)  
**1. R Process.**   
The objective is to process data from Yahoo finance and build the most common indicators to add to the series, these indicators have been taken considering the following article.To this end, [here](http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-63.pdf) is a new paper written by an engineering student at UC Berkeley which uses "support vector machine" together with 10 simple technical indicators to predict the SPX index, purportedly with 60% accuracy   
The content of the process is detailled here:   
    
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
library(quantmod)   
library(TTR)   
library(PerformanceAnalytics)   
# pull IBM data from Yahoo Finance   
getSymbols("IBM",from="2003-01-01")   
# Introduce RSI Indicator   
IBM$RSI2 = RSI(Cl(IBM), 2)   
#Introduce Eponential Moving Average indicator   
IBM$EMA7=EMA(Cl(IBM), n=7, wilder=FALSE, ratio=NULL)   
IBM$EMA50=EMA(Cl(IBM), n=50, wilder=FALSE, ratio=NULL)   
IBM$EMA200=EMA(Cl(IBM), n=200, wilder=FALSE, ratio=NULL)   
#Introduce MACD indicator   
IBM$MACD26=MACD(Cl(IBM), nFast=12, nSlow=26, nSig=9)   
#Introduce ADX indicator   
IBM$ADX14=ADX(IBM, n=14)   
#results <-transform(IBM,RSI.IBM=RSI(Cl(IBM), 2),RETURN=ret ,TIME=as.character(index(IBM)))   
# remove 2003,2004,2005 in order to avoid NaN from EMA indicators   
# To maintain time it is necessary to conver in texts   
results <-transform(IBM["2006-01-01::2009-01-01"],TIME=as.character(index(IBM["2006-01-01::2009-01-01"])))   
    
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
    
The output of the system is:   
[](http://lh6.ggpht.com/_upmtKBKjH54/TOWg4tQmGlI/AAAAAAAAArc/sLUEx7mEpic/s1600-h/clip_image004%5B4%5D.jpg)  
**2. String to Time (Nominal to Date)**   
We convert date string to Date.   
[](http://lh5.ggpht.com/_upmtKBKjH54/TOWpI8yQDvI/AAAAAAAAAuE/UqF_wzA1Qzs/s1600-h/stringtotime%5B4%5D.png)   
**3. Close adjuste to Label**   
We put label the IBM adjusted close value in order to predict one day in advance..   
[](http://lh3.ggpht.com/_upmtKBKjH54/TOWpK6-OxYI/AAAAAAAAAuM/i0HbljDf1zM/s1600-h/ibmlabel%5B4%5D.png)   
**4. set Time to ID (Set Role)**   
  
    
We use the TIME as ID for time serie data.   
**5. Widowing**   
We move one day in the future the variable to predict and add 2 new columns with lagged values in a time window of 2 days.   
[](http://lh5.ggpht.com/_upmtKBKjH54/TOWpMysEUzI/AAAAAAAAAuU/EAL3YbX_bU0/s1600-h/widowing%5B4%5D.png)   
**6. % sliding Window Validation**   
[](http://lh6.ggpht.com/_upmtKBKjH54/TOWpO7mt4-I/AAAAAAAAAuc/TLMQhxepCM4/s1600-h/sliding%5B4%5D.png)   
Time series validation   
[](http://lh6.ggpht.com/_upmtKBKjH54/TOWpRSGLlCI/AAAAAAAAAuk/OzG6ylspaNY/s1600-h/validation%5B4%5D.png)  
We use the Support Vector Machine Weka implementation   
[](http://lh4.ggpht.com/_upmtKBKjH54/TOWpTgF0KYI/AAAAAAAAAus/ciOZSrrecmM/s1600-h/validation2%5B5%5D.png)  
You can improve the accuracy of the prediction algorithm using any parameter optimizer or attribute selection.   
Now Validation process   
**7.. Obtain Technical Test data**  
This module is similar to the first one except we use evaluation data from the last year   
    
    
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
library(quantmod)   
library(TTR)   
library(PerformanceAnalytics)   
# pull IBM data from Yahoo Finance   
getSymbols("IBM",from="2009-01-01")   
# Introduce RSI Indicator   
IBM$RSI2 = RSI(Cl(IBM), 2)   
#Introduce Eponential Moving Average indicator   
IBM$EMA7=EMA(Cl(IBM), n=7, wilder=FALSE, ratio=NULL)   
IBM$EMA50=EMA(Cl(IBM), n=50, wilder=FALSE, ratio=NULL)   
IBM$EMA200=EMA(Cl(IBM), n=200, wilder=FALSE, ratio=NULL)   
#Introduce MACD indicator   
IBM$MACD26=MACD(Cl(IBM), nFast=12, nSlow=26, nSig=9)   
#Introduce ADX indicator   
IBM$ADX14=ADX(IBM, n=14)   
#results <-transform(IBM,RSI.IBM=RSI(Cl(IBM), 2),RETURN=ret ,TIME=as.character(index(IBM)))   
# remove 2009 in order to avoid NaN from EMA indicators 2010 evaluation   
# To maintain time it is necessary to conver in texts   
results <-transform(IBM["2010-01-01::"],TIME=as.character(index(IBM["2010-01-01::"])))   
.   
    
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
We use a similar pre-process Flow..   
[](http://lh5.ggpht.com/_upmtKBKjH54/TOWhE8DIH1I/AAAAAAAAAtM/VxeJCyWa43I/s1600-h/clip_image020%5B4%5D.jpg)  
**11. Apply Model**   
We will apply the model obtained before   
And finally we analyze the trading strategy results   
**12. Prediction Lable as Regular (Set Role)**   
It is modified the predicted label to use inside R process.   
**13. Date to Nominal**  
It is modified the date to nominal to use it in R process.   
**14. Set TIME as Regular (Set Role)**   
It is modified the TIME attributte as a regular to use it in R process..   
**15. Set TIME as Regular (Set Role)**   
This script is inspired in [FOSS trading code](http://blog.fosstrading.com/2009/04/testing-rsi2-with-r.html).   
    
    
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
library(quantmod)   
library(TTR)   
library(PerformanceAnalytics)   
# 31 prediction close\_ROCel  
# 33 close\_ROCel  
close\_ROC <- ROC(data[33])  
dates = as.Date(data$TIME)  
prediction\_ROC <-ROC(data[31])  
close\_ROC[1] <- 0  
prediction\_ROC[1] <- 0   
#generate signals from prediction values   
sigup <- ifelse(prediction\_ROC > 0, 1, 0)   
sigdn <- ifelse(prediction\_ROC < 0, -1, 0)   
# Replace missing signals with no position   
# (generally just at beginning of series)   
sigup[is.na(sigup)] <- 0   
sigdn[is.na(sigdn)] <- 0   
sig <- sigup + sigdn   
# Calculate equity curves   
eq\_up <- cumprod(1+close\_ROC\*sigup)   
eq\_dn <- cumprod(1+close\_ROC\*sigdn)   
eq\_all <- cumprod(1+close\_ROC\*sig)   
# obtain result   
result <-transform(data,sig=sig ,ret=close\_ROC, eq\_up=eq\_up, eq\_dn=eq\_dn, eq\_all=eq\_all)   
# This function gives us some standard summary   
# statistics for our trades.   
tradeStats <- function(signals, returns) {   
# Inputs:   
# signals : trading signals   
# returns : returns corresponding to signals   
# Combine data and convert to data.frame   
sysRet <- signals \* returns \* 100   
posRet <- sysRet > 0 # Positive rule returns   
negRet <- sysRet < 0 # Negative rule returns   
dat <- cbind(signals,posRet\*100,sysRet[posRet],sysRet[negRet],1)   
dat <- as.data.frame(dat)   
# Aggreate data for summary statistics   
means <- aggregate(dat[,2:4], by=list(dat[,1]), mean, na.rm=TRUE)   
medians <- aggregate(dat[,3:4], by=list(dat[,1]), median, na.rm=TRUE)   
sums <- aggregate(dat[,5], by=list(dat[,1]), sum)   
colnames(means) <- c("Signal","% Win","Mean Win","Mean Loss")   
colnames(medians) <- c("Signal","Median Win","Median Loss")   
colnames(sums) <- c("Signal","# Trades")   
all <- merge(sums,means)   
all <- merge(all,medians)   
wl <- cbind( abs(all[,"Mean Win"]/all[,"Mean Loss"]),   
abs(all[,"Median Win"]/all[,"Median Loss"]) )   
colnames(wl) <- c("Mean W/L","Median W/L")   
all <- cbind(all,wl)   
return(all)   
}   
# trade stats   
stats<- as.data.frame(tradeStats(sig,close\_ROC))   
ret\_all<-close\_ROC   
xts.ts <- xts(ret\_all,dates)   
drawdownrport = table.Drawdowns(xts.ts)   
    
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
  
In the following graph you can see the not well ROC of this strategy

[](http://3.bp.blogspot.com/_upmtKBKjH54/TO7NFmM6UYI/AAAAAAAAAu8/tybpz-DB_Hc/s1600/Sin+t%25C3%25ADtulo.jpg)

Return obtained during buy and shell signals   
[](http://lh6.ggpht.com/_upmtKBKjH54/TOWhHvx25VI/AAAAAAAAAto/ZEsjaqfsuo4/s1600-h/clip_image024%5B4%5D.jpg)  
This strategy is a simplification, and that should be understand as a proof of concept.   
All information is in this tutorial, however if you want to [clip_image025](http://uploadnsell.com/buy/pZgaFV) an small quantity of money to improve this web you can obtain the files[here.](http://uploadnsell.com/buy/pZgaFV)

Strategies: borrowing ideas from Engineering Returns

*December 19, 2011*

By [systematicinvestor](http://www.r-bloggers.com/author/systematicinvestor/" \o "Posts by systematicinvestor)

(This article was first published on [**Systematic Investor » R**](http://systematicinvestor.wordpress.com/2011/12/20/rotational-trading-strategies-borrowing-ideas-from-engineering-returns/), and kindly contributed to [R-bloggers)](http://www.r-bloggers.com/)

Frank Hassler at [Engineering Returns](http://engineering-returns.com/) blog wrote an excellent article [Rotational Trading: how to reduce trades and improve returns](http://engineering-returns.com/2011/07/06/rotational-trading-how-to-reducing-trades-and-improve-returns/). The article presents four methods to reduce trades:

* Trade less frequently. I.e. weekly instead of daily rebalancing.
* Different criteria for enter / exit a trade.
* Smooth the rank over the last couple of bars.
* Combination of above.

I want show how to implement these ideas using the backtesting library in the[Systematic Investor Toolbox](http://systematicinvestor.wordpress.com/systematic-investor-toolbox/). I will use the 21 ETFs from the [ETF Sector Strategy](http://www.etfscreen.com/sectorstrategy.php)post as the investment universe.

Following code loads historical prices from Yahoo Fiance and compares performance of the daily versus weekly rebalancing using the backtesting library in the [Systematic Investor Toolbox](http://systematicinvestor.wordpress.com/systematic-investor-toolbox/):

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49 | # Load Systematic Investor Toolbox (SIT)  setInternet2(TRUE)  con = gzcon(url('<https://github.com/systematicinvestor/SIT/raw/master/sit.gz>', 'rb'))      source(con)  close(con)        #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*      # Load historical data      #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*      load.packages('quantmod')      tickers = spl('XLY,XLP,XLE,XLF,XLV,XLI,XLB,XLK,XLU,IWB,IWD,IWF,IWM,IWN,IWO,IWP,IWR,IWS,IWV,IWW,IWZ')        data <- new.env()      getSymbols(tickers, src = 'yahoo', from = '1970-01-01', env = data, auto.assign = T)          for(i in ls(data)) data[[i]] = adjustOHLC(data[[i]], use.Adjusted=T)      bt.prep(data, align='remove.na', dates='1970::2011')        #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*      # Code Strategies : weekly rebalancing      #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*      prices = data$prices      n = len(tickers)        # find week ends      week.ends = endpoints(prices, 'weeks')          week.ends = week.ends[week.ends > 0]          # Rank on ROC 200      position.score = prices / mlag(prices, 200)          position.score.ma = position.score          buy.rule = T        # Select Top 2 funds daily      data$weight[] = NA          data$weight[] = ntop(position.score, 2)          capital = 100000          data$weight[] = (capital / prices) \* bt.exrem(data$weight)      top2.d = bt.run(data, type='share', trade.summary=T, capital=capital)        # Select Top 2 funds weekly      data$weight[] = NA          data$weight[week.ends,] = ntop(position.score[week.ends,], 2)          capital = 100000          data$weight[] = (capital / prices) \* bt.exrem(data$weight)      top2.w = bt.run(data, type='share', trade.summary=T, capital=capital)        # Plot Strategy Metrics Side by Side      plotbt.strategy.sidebyside(top2.d, top2.w, perfromance.fn = 'engineering.returns.kpi') |

The number of trades falls down from 443 to 164 as we switch from daily to weekly rebalancing. The additional bonus is the better returns for the weekly rebalancing.

Next, let’s examine different entry/exit rank. We will buy top 2 ETFs and will keep them till their ranks drop below 4 /6.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  # Code Strategies : different entry/exit rank  #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    # Select Top 2 funds, Keep till they are in 4/6 rank  data$weight[] = NA      data$weight[] = ntop.keep(position.score, 2, 4)      capital = 100000      data$weight[] = (capital / prices) \* bt.exrem(data$weight)  top2.d.keep4 = bt.run(data, type='share', trade.summary=T, capital=capital)    data$weight[] = NA      data$weight[] = ntop.keep(position.score, 2, 6)      capital = 100000      data$weight[] = (capital / prices) \* bt.exrem(data$weight)  top2.d.keep6 = bt.run(data, type='share', trade.summary=T, capital=capital)    # Plot Strategy Metrics Side by Side  plotbt.strategy.sidebyside(top2.d, top2.d.keep4, top2.d.keep6, perfromance.fn = 'engineering.returns.kpi') |

The number of trades falls down from 443 to 95 to 52 as we hold on to our selection for longer periods.

Next, let’s examine rank smoothing. Instead of using the most recent rank, we will use different averages of rank’s recent values.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  # Code Strategies : Rank smoothing  #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    models = list()  models$Bench = top2.d  for( avg in spl('SMA,EMA') ) {      for( i in c(3,5,10,20) ) {          position.score.smooth = bt.apply.matrix(position.score.ma, avg, i)              position.score.smooth[!buy.rule,] = NA            data$weight[] = NA              data$weight[] = ntop(position.score.smooth, 2)              capital = 100000              data$weight[] = (capital / prices) \* bt.exrem(data$weight)          models[[ paste(avg,i) ]] = bt.run(data, type='share', trade.summary=T, capital=capital)      }  }    # Plot Strategy Metrics Side by Side  plotbt.strategy.sidebyside(models, perfromance.fn = 'engineering.returns.kpi') |

The number of trades falls down as we increase the length of period used in averaging. There is no big difference in using simple moving average (SMA) versus exponential smoothing average (EMA).

Next, let’s combine different methods to reduce number of trades.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  # Code Strategies : Combination  #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    # Select Top 2 funds daily, Keep till they are 6 rank, Smooth Rank by 10 day EMA  position.score.smooth = bt.apply.matrix(position.score.ma, 'EMA', 10)      position.score.smooth[!buy.rule,] = NA  data$weight[] = NA      data$weight[] = ntop.keep(position.score.smooth, 2, 6)      capital = 100000      data$weight[] = (capital / prices) \* bt.exrem(data$weight)  top2.d.keep6.EMA10 = bt.run(data, type='share', trade.summary=T, capital=capital)    # Select Top 2 funds weekly, Keep till they are 6 rank  data$weight[] = NA      data$weight[week.ends,] = ntop.keep(position.score[week.ends,], 2, 6)      capital = 100000      data$weight[] = (capital / prices) \* bt.exrem(data$weight)  top2.w.keep6 = bt.run(data, type='share', trade.summary=T, capital=capital)    # Select Top 2 funds weekly, Keep till they are 6 rank, Smooth Rank by 10 week EMA  position.score.smooth[] = NA      position.score.smooth[week.ends,] = bt.apply.matrix(position.score.ma[week.ends,], 'EMA', 10)          position.score.smooth[!buy.rule,] = NA    data$weight[] = NA      data$weight[week.ends,] = ntop.keep(position.score.smooth[week.ends,], 2, 6)      capital = 100000      data$weight[] = (capital / prices) \* bt.exrem(data$weight)  top2.w.keep6.EMA10 = bt.run(data, type='share', trade.summary=T, capital=capital)    # Plot Strategy Metrics Side by Side  plotbt.strategy.sidebyside(top2.d, top2.d.keep6, top2.d.keep6.EMA10, top2.w, top2.w.keep6, top2.w.keep6.EMA10, perfromance.fn = 'engineering.returns.kpi') |

The overall winner is a weekly strategy that buys top 2 ETF’s based on 10 week exponential average rank and keeps them till their ranks drop below 6. The number of trades falls down from 443 to 28 and performance (CAGR) goes up from 2.4% to 7.3%.

The next step, which you can do as a homework, is to find ways to control the strategy’s drawdowns. One solution is discussed in the [Avoiding severe draw downs](http://engineering-returns.com/2010/07/26/rotational-trading-system/) post.

To view the complete source code for this example, please have a look at the[bt.rotational.trading.trades.test() function in bt.test.r at github](https://github.com/systematicinvestor/SIT/blob/master/R/bt.test.r).