Few words about Trading Strategies

One of the biggest challenges is to predict the Market. Many people have developed their own trading strategies, some of them are advanced based on Machine Learning and Artificial Intelligence algorithms like LSTM, xgBoost, Random Forest etc, some others are based on Statistical models like ARIMA, and some others are based on technical analysis.

Whatever Trading Strategy we are about to apply we need to backtest it, meaning to simulate it and finally to evaluate it. Today, we will provide an example of how you can easily backtest your own trading strategy in R.

Define the Trading Strategy

For illustrative purposes, we defined an arbitrary Trading Strategy which does not make much sense, but it is good to work on it as an example. Let's define the rules:

When the **close price** of the Cryptocurrency is **X consecutive** days in the same direction (i.e. 7 consecutive days "up" or 7 consecutive days "down") then we **Open a Position** as follows:

- When the close price is X consecutive days down, then the next day we buy (long) the cryptocurrency at the "open price"
- When the close price is X consecutive days up, then the next day we sell (short) the cryptocurrency at the "open price". We assume that short selling is allowed with CFD

Once we have Opened our positions we use the following alerts:

- TP: Take profit when your P/L is above X%
- SL: Stop loss when you P/L is below -Y%

Every position is closed at the open price. Of course, we can extend this assumption by considering hourly or even per minute data. Every trade is **1 unit** but we can change this to a multiplier or to a fraction. Notice that the assumption of the 1-unit does not affect our outcome, since we communicate the ROI which is the (P/L) as a percentage.

R Code for to backtest the Trading Strategy

You can have a look at how we can get the Cryptocurrency prices in R and how to count the consecutive events in R. Below we build a function which takes as parameters:

- **symbol**: The cryptocurrency symbol. For example, `BTC` is for the Bitcoin.
- **consecutive**: The consecutive count of the signs of the closing prices.
- **SL**: The percentage that we stop loss.
- **TP**: The percentage that we take profit.
- **start_date**: The date that we want to start the backtesting strategy.

Notice that the open positions that have not met the alert criteria of SL and TP and still "Active" and we return them with an "Active" status and as "Profit" we return their current "Profit".

```
library(tidyverse)
library(crypto)


back_testing<-function(symbol="BTC", consecutive=7, SL=0.1, TP=0.1, start_date = "20180101") {



  df<-crypto_history(coin = symbol, start_date = start_date)



  df<-df%>%mutate(Sign = ifelse(close>lag(close),"up", "down"))%>%
    mutate(Streak=sequence(rle(Sign)$lengths))


  df<-df%>%select(symbol, date, open, high, low, close, Sign, Streak)%>%na.omit()%>%
    mutate(Signal = case_when(lag(Sign)=="up" & lag(Streak)%%consecutive==0~'short',
                              lag(Sign)=="down" & lag(Streak)%%consecutive==0~'long',
                              TRUE~""), Dummy=TRUE
    )


  Trades<-df%>%filter(Signal!="")%>%select(Open_Position_Date=date, Open_Position_Price=open,
Dummy, Signal)
  Trades


  Portfolios<-Trades%>%inner_join(df%>%select(-Signal), by="Dummy")%>%filter(date>
Open_Position_Date)%>%select(-Dummy)%>%mutate(Pct_Change=open/Open_Position_Price-1)%>%
    mutate(Alert = case_when(Signal=='long'& Pct_Change>TP~'TP',
                             Signal=='long'& Pct_Change< -SL~'SL',
                             Signal=='short'& Pct_Change>TP~'SL',
                             Signal=='short'& Pct_Change< -SL~'TP'
    )
    )%>%group_by(Open_Position_Date)%>%mutate(Status=ifelse(sum(!is.na(Alert))>0, 'Closed',
'Active'))


  Active<-Portfolios%>%filter(Status=='Active')%>%group_by(Open_Position_Date)%>%arrange(
date)%>%slice(n())%>%
    mutate(Profit=case_when(Signal=='short'~Open_Position_Price-open,
                            Signal=='long'~open-Open_Position_Price))%>%
    select(symbol, Status, Signal, date, Open_Position_Date, Open_Position_Price, open, Profit)

  Closed<-Portfolios%>%filter(Status=='Closed')%>%na.omit()%>%group_by(Open_Position_Date)
%>%arrange(date)%>%slice(1)%>%
    mutate(Profit=case_when(Signal=='short'~Open_Position_Price-open,
                            Signal=='long'~open-Open_Position_Price))%>%
    select(symbol, Status, Signal, date, Open_Position_Date, Open_Position_Price, open, Profit)

  final<-bind_rows(Closed,Active)%>%ungroup()%>%arrange(date)%>%mutate(ROI=Profit/
Open_Position_Price, Running_ROI=cumsum(Profit)/cumsum(Open_Position_Price))


  return(final)


}
```

Results of the Backtest

Let’s assume that we want to backtest the trading strategy that we described earlier with the following parameters:

```
symbol="BTC", consecutive=5, SL=0.1, TP=0.5, start_date = "20180101"
```

```
ttt<-back_testing(symbol="BTC", consecutive=5, SL=0.1, TP=0.15, start_date = "20180101")
```

symbol	Status	Signal	Closing_Date	Open_Position_Date	Open_Position_Price	Closing_Price	Profit	**ROI**	**Running_ROI**
BTC	Closed	short	1/9/2018	1/7/2018	17527	15124	2404	13.70%	13.70%
BTC	Closed	short	3/8/2018	3/6/2018	11500	9951	1549	13.50%	13.60%
BTC	Closed	long	3/18/2018	3/11/2018	8853	7891	-962	-10.90%	7.89%
BTC	Closed	short	4/25/2018	4/15/2018	7999	9701	-1702	-21.30%	2.81%
BTC	Closed	short	8/9/2018	7/18/2018	7315	6306	1010	13.80%	4.32%
BTC	Closed	long	8/9/2018	8/4/2018	7439	6306	-1133	-15.20%	1.92%
BTC	Closed	long	11/20/2018	11/17/2018	5579	4864	-715	-12.80%	0.68%
BTC	Closed	short	12/28/2018	12/21/2018	4134	3653	481	11.60%	1.32%
BTC	Closed	short	4/3/2019	2/20/2019	3947	4880	-933	-23.60%	0.00%
BTC	Closed	short	5/10/2019	4/21/2019	5336	6176	-840	-15.70%	-1.06%
BTC	Closed	short	5/12/2019	5/5/2019	5831	7204	-1372	-23.50%	-2.59%

BTC	Closed short	5/27/2019	5/12/2019	7204	8674	-1471	-20.40%	-3.98%
BTC	Closed short	6/5/2019	5/28/2019	8803	7704	1098	12.50%	-2.55%
BTC	Closed short	6/23/2019	6/17/2019	8989	10697	-1708	-19.00%	-3.89%
BTC	Closed short	6/27/2019	6/24/2019	10854	13017	-2163	-19.90%	-5.32%
BTC	Closed short	8/30/2019	8/4/2019	10822	9515	1307	12.10%	-3.90%
BTC	Closed short	9/25/2019	9/4/2019	10621	8603	2018	19.00%	-2.20%
BTC	Closed long	9/25/2019	9/18/2019	10248	8603	-1644	-16.00%	-3.12%
BTC	Closed long	1/15/2020	11/23/2019	7296	8825	1529	21.00%	-2.03%
BTC	Closed long	1/15/2020	12/5/2019	7253	8825	1572	21.70%	-1.00%
BTC	Closed short	1/31/2020	1/8/2020	8162	9508	-1346	-16.50%	-1.72%
BTC	Closed short	2/27/2020	2/10/2020	10116	8825	1290	12.80%	-0.93%
BTC	Closed long	3/13/2020	2/29/2020	8671	5018	-3653	-42.10%	-2.77%
BTC	Closed short	5/2/2020	4/27/2020	7679	8869	-1190	-15.50%	-3.25%
BTC	Closed long	7/28/2020	6/28/2020	9048	11017	1969	21.80%	-2.18%

Discussion

As we can see the Trading Strategy ended with **-2.18% P/L** without taking into account the transaction fees. You can see also that in some periods was profitable (2018) and in some other periods was not (2019), that is why is very important to backtest a trading strategy for many different periods.

Notice that with this function we can backtest hundred of backtest strategies with some small tweaks. Instead of this simple trading strategy could be an advanced machine learning model. Once we define the trading signals, then the backtest is the same.