



universität
wien

MAGISTERARBEIT / MASTER'S THESIS

Titel der Magisterarbeit / Title of the Master's Thesis

„Evaluation of the performance of the Automatic-1-2-3
trend indicator“

verfasst von / submitted by

Thomas Sinkovics, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of
Magister der Sozial- und Wirtschaftswissenschaften (Mag. rer. soc. oec.)

Wien, 2016 / Vienna 2016

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

A 066 951

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Magisterstudium Statistik

Betreut von / Supervisor:

ao. Univ.-Prof. Dipl.-Ing. Dr. Erhard Reschenhofer

Eidesstattliche Erklärung

Ich erkläre hiermit eidesstattlich, dass ich die Arbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und alle aus ungedruckten Quellen, gedruckter Literatur oder aus dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte gemäß den Richtlinien wissenschaftlicher Arbeiten zitiert, durch Fußnoten gekennzeichnet beziehungsweise mit genauer Quellenangabe kenntlich gemacht habe. Diese schriftliche Arbeit wurde noch an keiner anderen Stelle vorgelegt.

Wien, 2016

A handwritten signature in dark ink, reading "Thomas Sinkovics". The signature is written in a cursive style with a large, stylized 'T' and 'S'.

Unterschrift

Statutory declaration

Hereby, I declare that I have authored this thesis independently, that I have not used other than the declared sources, and that I have explicitly marked all material which has been quoted either literally or by content from other sources. This master thesis has not been previously presented as an examination paper in this or any other form in Austria or abroad.

Vienna 2016

A handwritten signature in cursive script, reading "Thomas Sinkovics". The signature is written in dark ink and is positioned above a horizontal line.

Signature

Table of Contents

1	Introduction	5
1.1	Trend.....	5
1.2	Automatic-1-2-3-algorithm.....	6
1.2.1	MACD	6
1.2.2	Renko	7
1.2.3	Min-Max-process	9
1.2.4	Trend indicator.....	10
1.2.5	Further reading	10
1.3	Simple algorithm	11
1.4	Benchmark	12
2	Materials and Methods.....	13
2.1	Data.....	13
2.1.1	SP500.....	13
2.1.2	FTSE	13
2.1.3	NIKKEI225.....	14
2.1.4	DAX	14
2.1.5	EUR USD	14
2.1.6	USD JPY	14
2.1.7	GOLD HUI	14
2.1.8	OIL XOI.....	14
2.1.9	YAHOO.....	14
2.2	Methods	15
3	Results	17
4	Discussion.....	25
5	Appendix A: References	26
6	Appendix B: Abstract / Zusammenfassung	27
7	Appendix C: R-Code	29
8	Appendix D: CURRICULUM VITAE	37

1 Introduction

Trend-trading describes trading strategies which use trends to make buying- and selling-decisions for securities or commodities. See Dempster and Jones [2] or Silagadze [8] for example. Trends may be defined using the relevant maxima and minima of the corresponding time series. But the decision, which maxima and minima are relevant, is rather subjective. One can construct algorithms doing this task.

1.1 Trend

As said above, given the relevant minima and maxima trends can objectively be defined [7]. A time series is in an up-trend if the last consecutive relevant maxima have risen and the last consecutive relevant minima have not fallen. Analogously, a time series is in a down-trend if the last consecutive relevant minima have fallen and the last consecutive relevant maxima have not risen. These definitions are graphically shown in Figure 1.

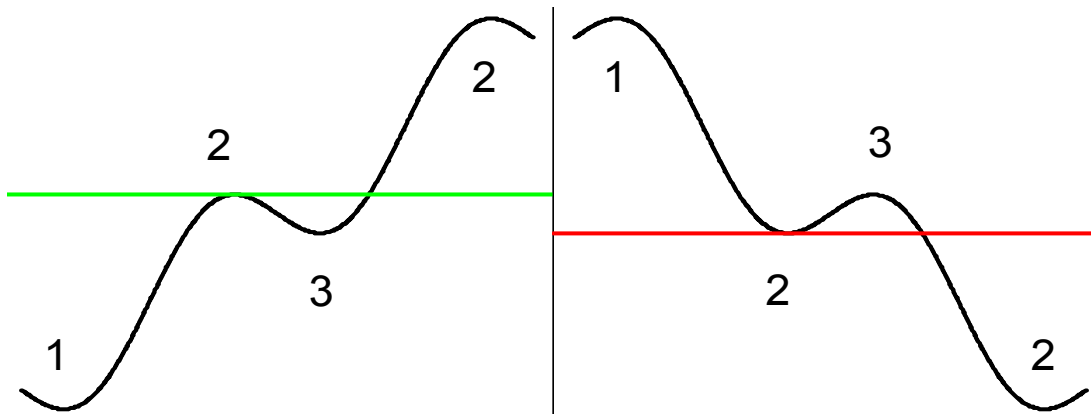


Figure 1: Start of an up-trend (left), start of a down-trend (right)

As seen in Figure 1, the relevant minima and maxima can be numerated. In an up-trend, the starting minimum is numbered 1, the following maxima are numbered 2 and the following minima are numbered 3, as long the trend continues. Vice versa in a down-trend, the starting maximum is numbered 1, the following minima are numbered 2 and the following maxima are numbered 3, as long as the trend continues. The corresponding prices of the minima and maxima are often called point 1, point 2 and point 3.

A forming trend can be recognized, as soon as the price passes the point 2, the relevant minimum or maximum has not to be formed yet. Given this definition, there is still the need for the extraction of the relevant minima and maxima.

1.2 Automatic-1-2-3-algorithm

The Automatic-1-2-3-algorithm (in the following shortly referred to as Auto123) uses a SAR-process (stop-and-reverse) to construct a Min-Max-process (the relevant minima and maxima). In [5] MACD- and Renko-based SAR-processes are described.

1.2.1 MACD

The Moving Average Convergence/Divergence (short MACD) is the difference between a short period (“fast”) exponential moving average (EMA) and a longer period (“slow”) EMA of the price process. The so called signal line is defined as an EMA of the MACD. Obviously, the position of the MACD in regard of the signal line can be used for the recognition of upward (direction = 1) or downward movements (direction = -1). There are different ways using these lines to get a direction process, which is a sequence of -1, 0, and 1’s. It is 0 as long as no direction has appeared so far.

The most intuitive way is to say it’s 1 if the MACD is above the signal line and -1 if it is below. This can be refined by specifying a positive number delta, which is a threshold. If the MACD is more than delta above the signal line you get 1, if it is more than delta below the signal line -1 (see Figure 2).

Since it is also possible, that the MACD is above the signal line for a very long time but not high enough to pass the delta-threshold, it makes sense to cumulate the differences. As long as the MACD is on one side of the signal line, the differences are summed up, you get 1 if the sum exceeds delta or -1 if it falls below - delta. The threshold delta itself can be set up in different ways. It can be either fixed (*delfix*=0) or variable. It can be defined as a multiple of the current price (*delfix*=1) or the average true range (ATR) (*delfix*=2). The ATR is a smoothed moving average of the true range of the *spanne* (arbitrary integer value) last periods, whereby the true range is the maximal absolute difference

between the minimum and maximum, or the previous close value and the maximum or the previous close value and the minimum [11].

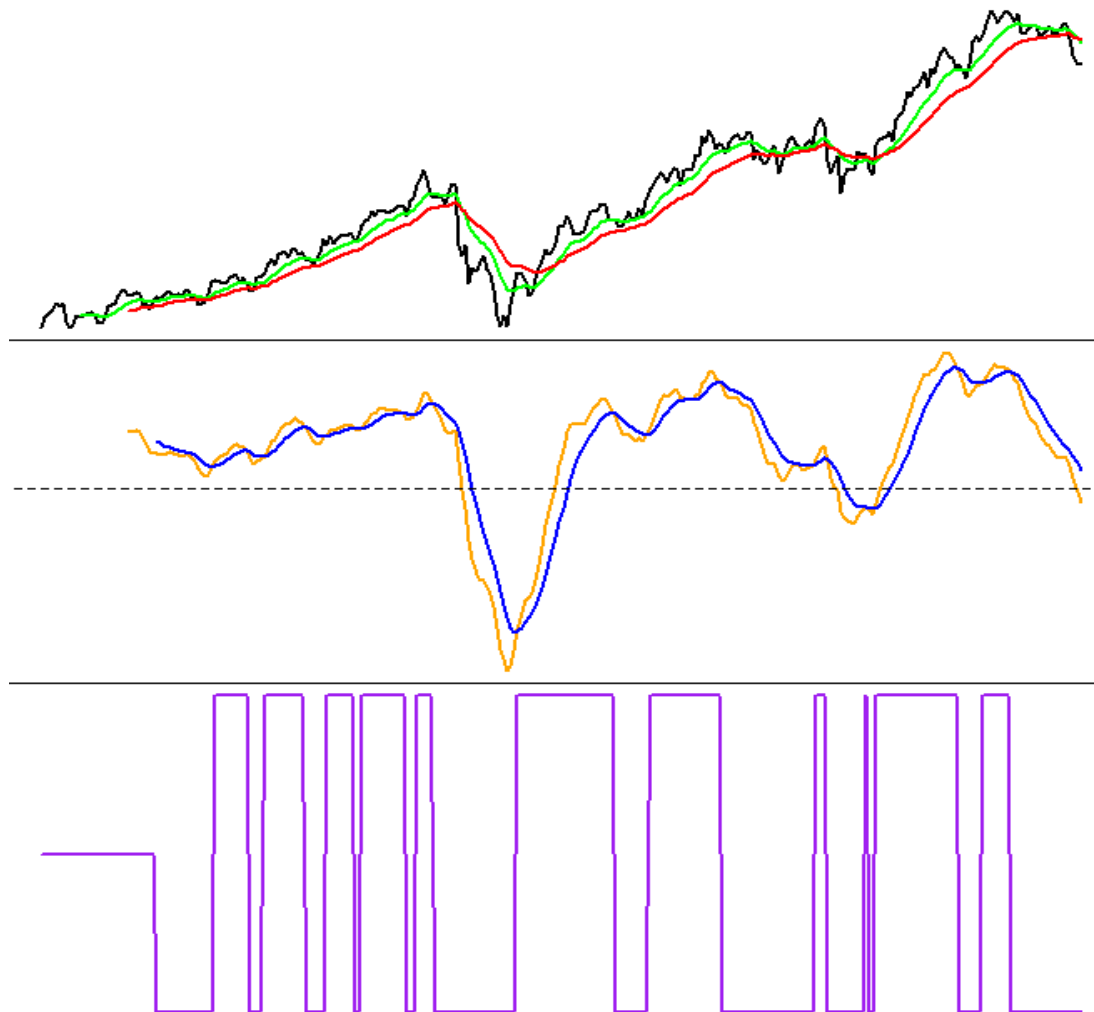


Figure 2: Price, short EMA, longer EMA, MACD, signal line, zero line, directions

All of these possibilities already lead to a huge variety in parameter tuning. You need a period length for the short time EMA (*fastparam*), the longer time EMA (*slowparam*), the signal line (*triggerparam*) and the ATR (*spanne*). A value for the threshold *delta* is needed, as well as choosing the method (*delfix*). The parameters for the EMAs can be tuned simultaneously using a scaling factor (*timescale*), e.g., *fastparam*=12**timescale*, *slowparam*=26**timescale* and *triggerparam*=9**timescale*.

1.2.2 Renko

Renko-charts (from renga, Japanese for bricks) simplify a price process by “discretizing” it into bricks. We distinguish between up-bricks (coloured white) and

down bricks (coloured black). A new up-brick is drawn, if the current price is at least one brick size (a pre specified unit) higher than the top of the last brick. Conversely a new down-brick is drawn, if the current price is at least one brick size below the bottom of the last brick. The date/time axis is usually not interval-scaled anymore since the bricks are forced into the same width.

The brick size can be chosen in different ways. It can be fixed (later called method 0) or variable: as a multiple of the current price (later called method 1) or the average true range (ATR) of some last periods (later called method 2).

To avoid a constant jumping between up- and down-bricks one can specify a minimum number of bricks of the same kind before a change. In case of an up-brick the direction is 1, in case of a down-brick -1 and 0 until first brick is drawn (see Figure 3).

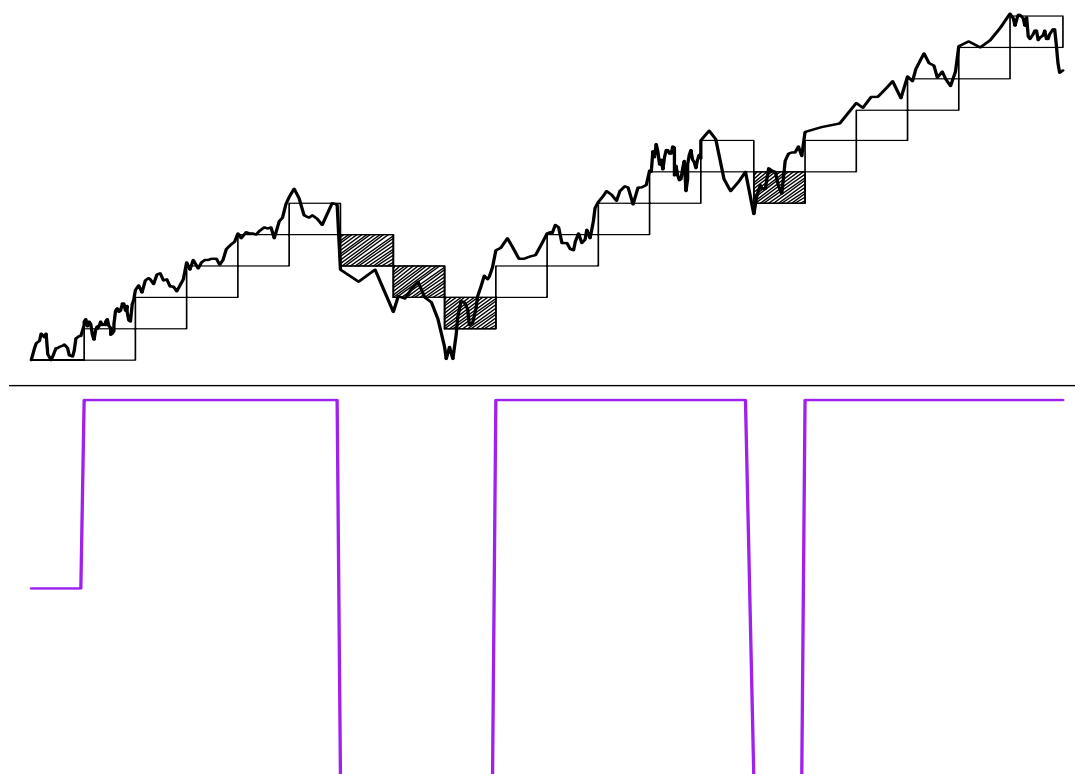


Figure 3: Price process with renko chart, **directions**

So again you get much variation through parameter tuning. A value for the box-size (*bsize*) is needed, as well as choosing the method (*bsfix*). You need a period length for the ATR (*spanne*) and the above mentioned minimum number of same bricks (*revamount*).

1.2.3 Min-Max-process

Based on the price process and the directions (either MACD- or Renko-based) the Min-Max-process can be constructed, which is a sequence of alternating minima and maxima. The basic idea is pretty simple. Whenever the direction is +1 we look for a maximum, when it is -1 we look for a minimum. Figure 4 shows an application of this (note: the used data set does not include HIGH or LOW values, so they are equal to the CLOSE value).

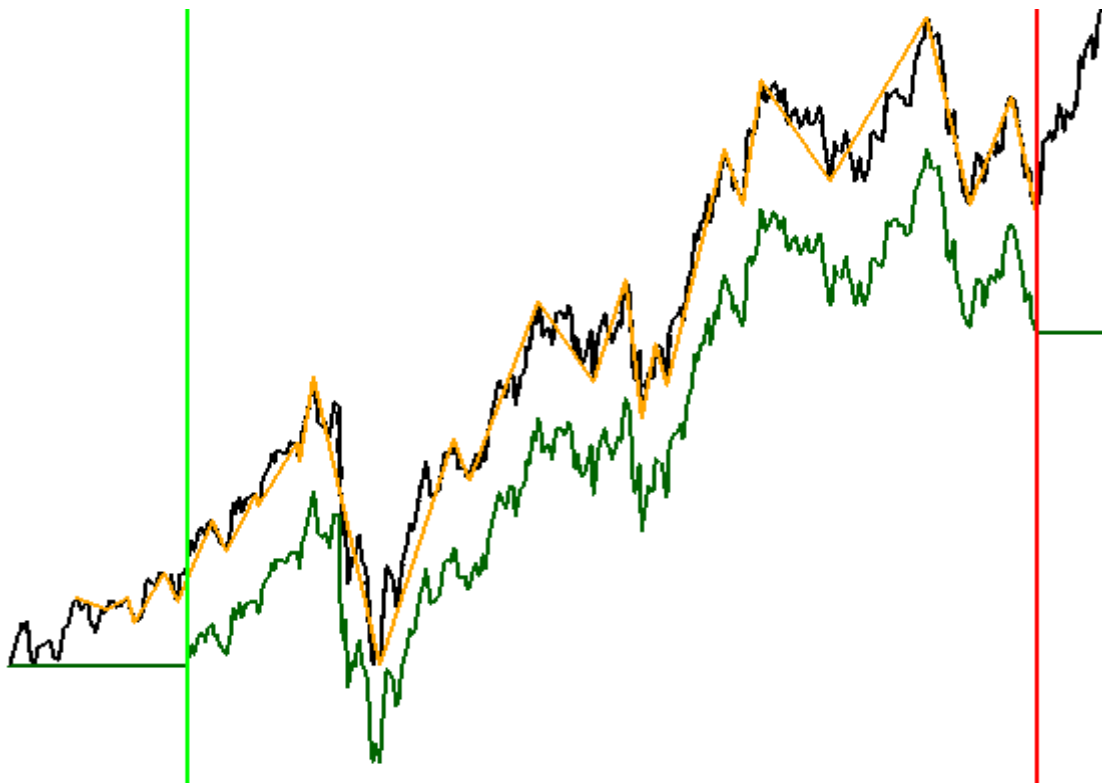


Figure 4: Auto123, price, Min-Max-process, buy-signal, sell-signal, performance

The algorithm, as it is described in [5], uses the HIGH and LOW values of the given time series (usually consisting of DATE, OPEN, HIGH, LOW, CLOSE, VOLUME, ADJUSTED CLOSE).

The direction initially is 0 for both SAR-methods. If at some point it changes to +1 first, the minimum of the LOW values of the previous periods is set as the first fixed minimum, the HIGH value of this point is set as a temporary maximum. Conversely if it changes to -1 first, the maximum of the HIGH values of the previous periods is set as the first fixed maximum, the LOW value of this point is set as a temporary minimum.

As long as the direction stays +1, the temporary maximum is updated if the current HIGH value is higher. If the direction changes to -1, the temporary maximum is fixed, the LOW value of this point is set as a temporary minimum. While the direction stays -1, the temporary minimum is updated if the current LOW value is lower. If the direction changes to +1, the temporary minimum is fixed, the HIGH value of this point is set as a temporary maximum.

A kind of exception situation is also considered. If the direction is +1 (-1) but the LOW (HIGH) value of this point is lower (higher) than the last fixed minimum (maximum), the temporary maximum (minimum) is fixed, and the LOW (HIGH) of this point is set as a new temporary minimum (maximum). This ensures that no relevant minimum/maximum is left out because of the directions.

1.2.4 Trend indicator

The Automatic-1-2-3-algorithm provides a wide variety of possible states. But only indicating whether there is an up-, down-trend or no trend is important to the strategy later used for comparison. As defined in 1.1 it is a simple task to identify the current trend at each time point. In Figure 4 you see the start of an up-trend at the green vertical line and the start of a down-trend at the red vertical line. When using the strategy “buy in up-trend, sell in down-trend” these lines also correspond to the buying- and selling-time-points. The dark green line represents the performance of a portfolio with the use of this strategy. It moves horizontal if the asset is not yet bought or sold (in the money) and proportional to the price process as long the asset is held (in the asset).

1.2.5 Further reading

More details of the Automatic-1-2-3-algorithm are mentioned in [5]. The bachelor's theses [9] and [10] compare the performance of different strategies but do not use many data sets.

The paper [3] describes a way to tune the *timescale* parameter such that the average period length of the Min-Max-process matches the dominant wavelength of the time series, which is computed via cross-correlation (later called Wave).

1.3 Simple algorithm

The goal of this thesis is not only to examine the performance of the Automatic-1-2-3-algorithm but also to compare it to a much simpler algorithm (Simple) which depends only on a single parameter p . This parameter is a kind of exceedance proportion, which has to be achieved, such that a new (relevant) minimum or maximum is added. It ensures, that successive minima and maxima have a difference of at least $100 \cdot p\%$.

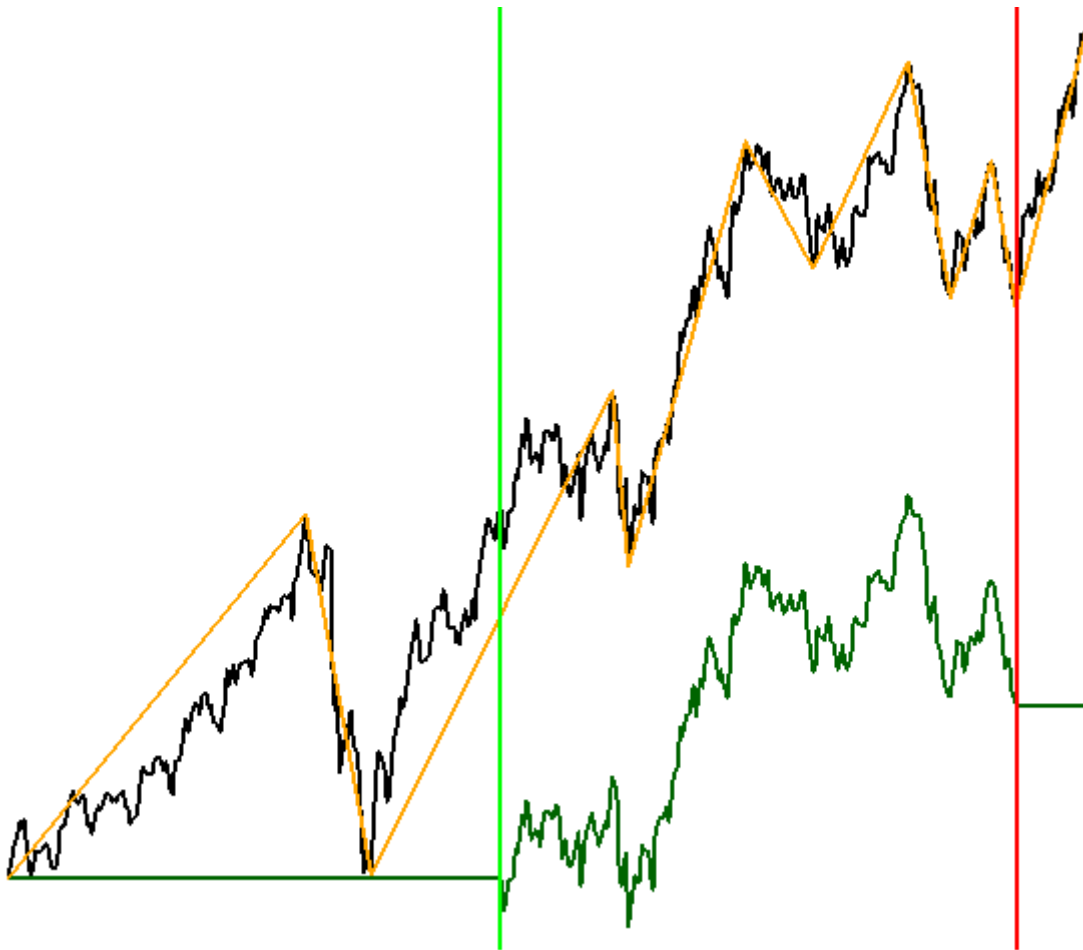


Figure 5: Simple, price, Min-Max-process, buy-signal, sell-signal, performance

At the beginning, as long no minimum or maximum has been found, a running minimum and maximum are calculated. At a given period, if the price is $100 \cdot p\%$ above the running minimum, it becomes a temporary maximum. The other way around, if the price is $100 \cdot p\%$ below the running maximum, it establishes a temporary minimum. The running minimum and maximum are neglected after this initialization phase.

Given the temporary extremum is a maximum, two cases can occur. If the current price is higher than this temporary maximum, it is updated to this point. If it is $100 \cdot p\%$ lower than the temporary maximum, this maximum is fixed and this point becomes a temporary minimum.

Given the temporary extremum is a minimum, two cases can occur. If the current price is lower than this temporary minimum, it is updated to this point. If it is $100 \cdot p\%$ higher than the temporary minimum, this minimum is fixed and this point becomes a temporary maximum.

Now, in the same way as before, trends can be identified using the definition. Figure 5 shows an application of this simple algorithm. The structure is the same way as before.

1.4 Benchmark

To have some kind of benchmark, the simplest strategy possible is also included in this comparison: you buy at the start of the observed period and sell at the end of it. Its performance corresponds to the performance of the price process. It is important to include this strategy since we also have to consider trading fees and recognize strategies where the trading frequencies would consume the benefits.

2 Materials and Methods

In this section, I describe the structure and origin of the used data as well as the methods used to compare the algorithms.

2.1 Data

This thesis covers nine data sets, which have the same variables: DATE/TIME, OPEN, HIGH, LOW, CLOSE, VOLUME, ADJUSTED CLOSE (see Figure 6), though in some sets CLOSE and ADJUSTED CLOSE are the same (e.g. for indices there is no adjustment necessary because there are no splits or dividends) and sometimes HIGH, LOW, and CLOSE are equal (meaning HIGH and LOW were not recorded for these). Descriptions for the data sets were partially taken from Wikipedia [11].

Date	Open	High	Low	Close	Volume	Adj.Close
1950-01-03	16.66	16.66	16.66	16.66	1260000	16.66
1950-01-04	16.85	16.85	16.85	16.85	1890000	16.85
1950-01-05	16.93	16.93	16.93	16.93	2550000	16.93
1950-01-06	16.98	16.98	16.98	16.98	2010000	16.98
1950-01-09	17.08	17.08	17.08	17.08	2520000	17.08
1950-01-10	17.03	17.03	17.03	17.03	2160000	17.03

Figure 6: Data structure

2.1.1 SP500

These are daily data of the Standard&Poor 500 Index, the American stock market index based on the 500 largest companies listed on the New York Stock Exchange or NASDAQ, from 03.01.1950 to 09.12.2015. Source for this time series is Yahoo Finance [12].

2.1.2 FTSE

These are daily data of the Financial Times Stock Exchange 100 Index, the British stock market index based on the 100 largest companies listed on the London Stock Exchange, from 03.01.1984 to 09.12.2015. Source for this time series is Yahoo Finance [12].

2.1.3 NIKKEI225

These are daily data of the NIKKEI 225 Index, the Japanese stock market index based on 225 selected companies listed on the Tokyo Stock Exchange, from 04.01.1984 to 09.12.2015. Source for this time series is Yahoo Finance [12].

2.1.4 DAX

These are daily data of the DAX, the German stock market index based on the 30 major companies listed on the Frankfurt Stock Exchange, from 26.11.1990 to 09.12.2015. Source for this time series is Yahoo Finance [12].

2.1.5 EUR USD

These are daily data of the exchange rate between the Euro and the US-Dollar, from 02.01.2002 to 09.12.2015. Source for this time series is the Frankfurt Exchange [1].

2.1.6 USD JPY

These are daily data of the exchange rate between the US-Dollar and the Japanese Yen, from 02.01.2001 to 09.12.2015. Source for this time series is the Frankfurt Exchange [1].

2.1.7 GOLD HUI

These are daily data of the HUI Gold Index, a weighted index based on gold mining companies, from 04.06.1996 to 14.12.2015. Source for this time series is Yahoo Finance [12].

2.1.8 OIL XOI

These are daily data of the NYSE Arca Oil Index, a weighted index based on companies involved in petroleum production, from 14.08.1991 to 09.12.2015. Source for this time series is Yahoo Finance [12].

2.1.9 YAHOO

These are intraday data (10 minute interval) of the stock value of the technology company Yahoo, from 02.01.2014 9:40 to 31.03.2014 16:00. Source for this time series is LOBSTER [4].

2.2 Methods

To objectively compare the performances of the algorithms, the same assumptions and strategy are applied to them.

The starting capital is set to 1. All assets can be traded without restriction to unit numbers (e.g. one can buy 17.26 pieces of a stock). Buying- and selling-orders can be executed at the close value the day they were placed.

The strategy is simply to buy at the emergence of an up-trend and sell at the appearance of a down-trend. At the end of the observation period a sale is obligatory (if possible), no matter how the situation is.

Comparisons are made for two cases. In the first one trading fees aren't considered at all. The second one regards trading fees of 0.25% of the traded value at each acquisition and each sale.

Since it is preferable to have 2 single lines for comparing the algorithms, a kind of "best of" combination is also included. At each time point the algorithms will evaluate which one of the parameter combinations has performed best so far and will use the best one for the decision at this time point. This can be viewed as a self-learning algorithm (later called BestAuto respectively BestSimple).

Both algorithms were implemented as mentioned before (section 1). Analysis and graphics are made with the statistical programming language and environment R [6]. The used source-code is listed in the appendix of this thesis.

Table 1 shows the parameter constellations which were used. Note that the given values for *fastparam*, *slowparam* and *triggerparam* are most commonly used when working with the MACD in practice (see [5]). The values for WAVE 1 were proposed in [3]. In addition, I tried alternative values to improve the performance (WAVE 2).

The ranges for *p*, *delta*, *bsize* and *revamount* fixed after some pre-testing. Smaller values lead to similar results like the smallest value chosen, which are

pretty poor since the number of trades is way too large in that cases, the performance is bad. Larger values than the chosen ones lead to no trades at all, since the thresholds are (almost) never reached. The remaining parameters (delfix, integrated, bsfix) also lead to the overall best results for the chosen values. All in all for each dataset there are 200 different applications of the simple algorithm compared to 405 of the Auto123 (see Table 1).

Algorithm	# Appl.	Daily data	Intraday
Simple algorithm	200	$p=0.001, 0.002, \dots, 0.2$	$p=0.0002, 0.0004, \dots, 0.04$
Auto123: MACD	100	$\delta=0.2, 0.4, \dots, 20$ $delfix=1$ $integrated= \text{FALSE}$ $fastparam=12$ $slowparam=26$ $triggerparam=9$ $timescale=1$	$\delta=0.04, 0.8, \dots, 4$ $delfix=1$ $integrated= \text{FALSE}$ $fastparam=12$ $slowparam=26$ $triggerparam=9$ $timescale=1$
Auto123: WAVE 1	1	$\delta=0.3$ $delfix=1$ $integrated=\text{FALSE}$ $fastparam=12$ $slowparam=26$ $triggerparam=9$	$\delta=0.3$ $delfix=1$ $integrated=\text{FALSE}$ $fastparam=12$ $slowparam=26$ $triggerparam=9$
Auto123: WAVE 2	4	$\delta=0.5, 1, 2 \text{ or } 5$ $delfix=1$ $integrated= \text{FALSE}$ $fastparam=12$ $slowparam=26$ $triggerparam=9$	$\delta=0.25, 0.5, 0.75 \text{ or } 1$ $delfix=1$ $integrated= \text{FALSE}$ $fastparam=12$ $slowparam=26$ $triggerparam=9$
Auto123: RENKO	300	$bsize=0.2, 0.4, \dots, 20$ $bsfix=1$ $revamount=1, 2 \text{ or } 3$	$bsize=0.04, 0.8, \dots, 4$ $bsfix=1$ $revamount=1, 2 \text{ or } 3$

Table 1: Used Parameters

3 Results

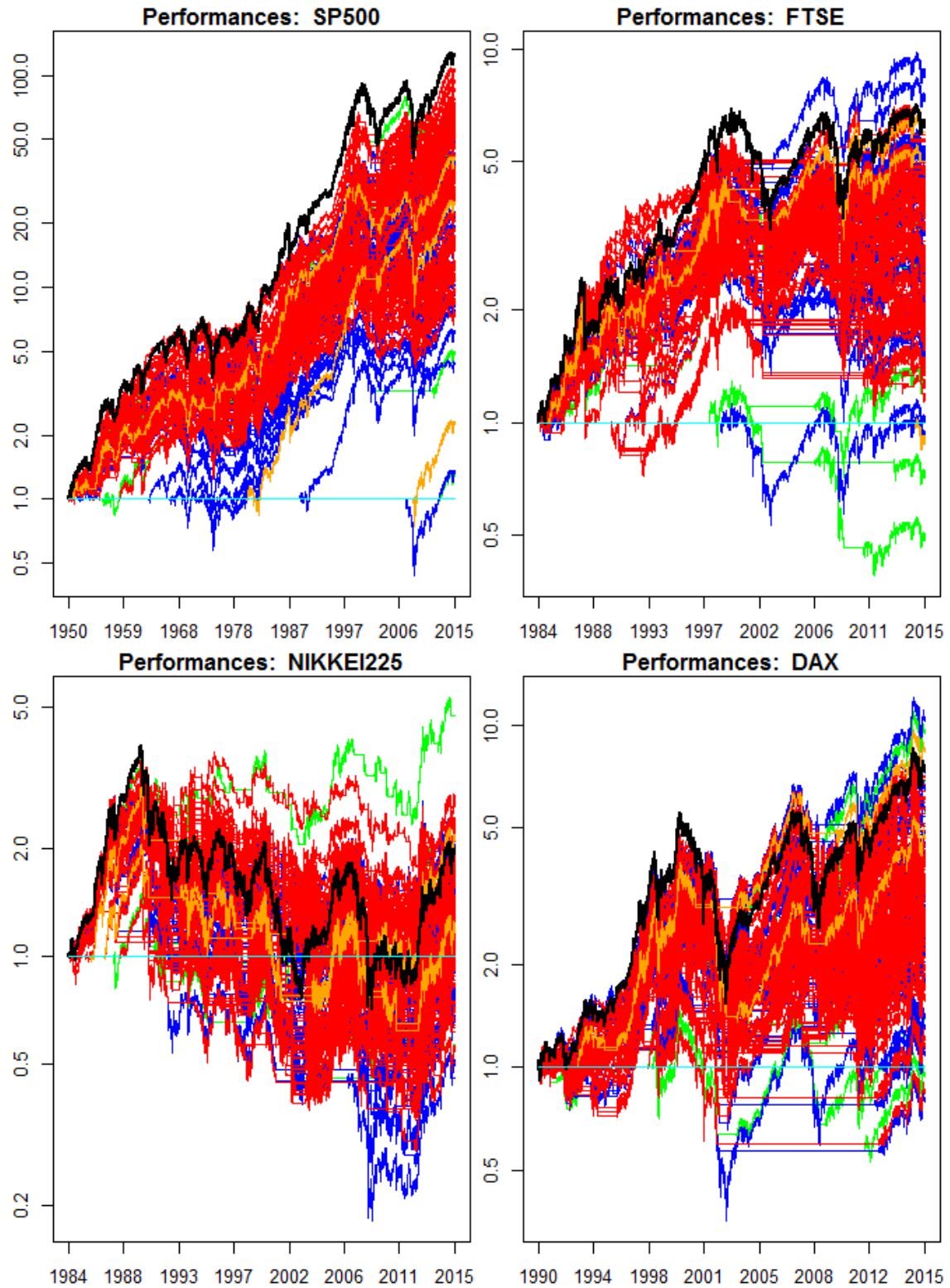


Figure 7: Comparison without fees 1, **Simple**, **Auto123-MACD**, **Auto123-WAVE 1**, **Auto123-WAVE 2**, **Auto123-RENKO**, **Benchmark**

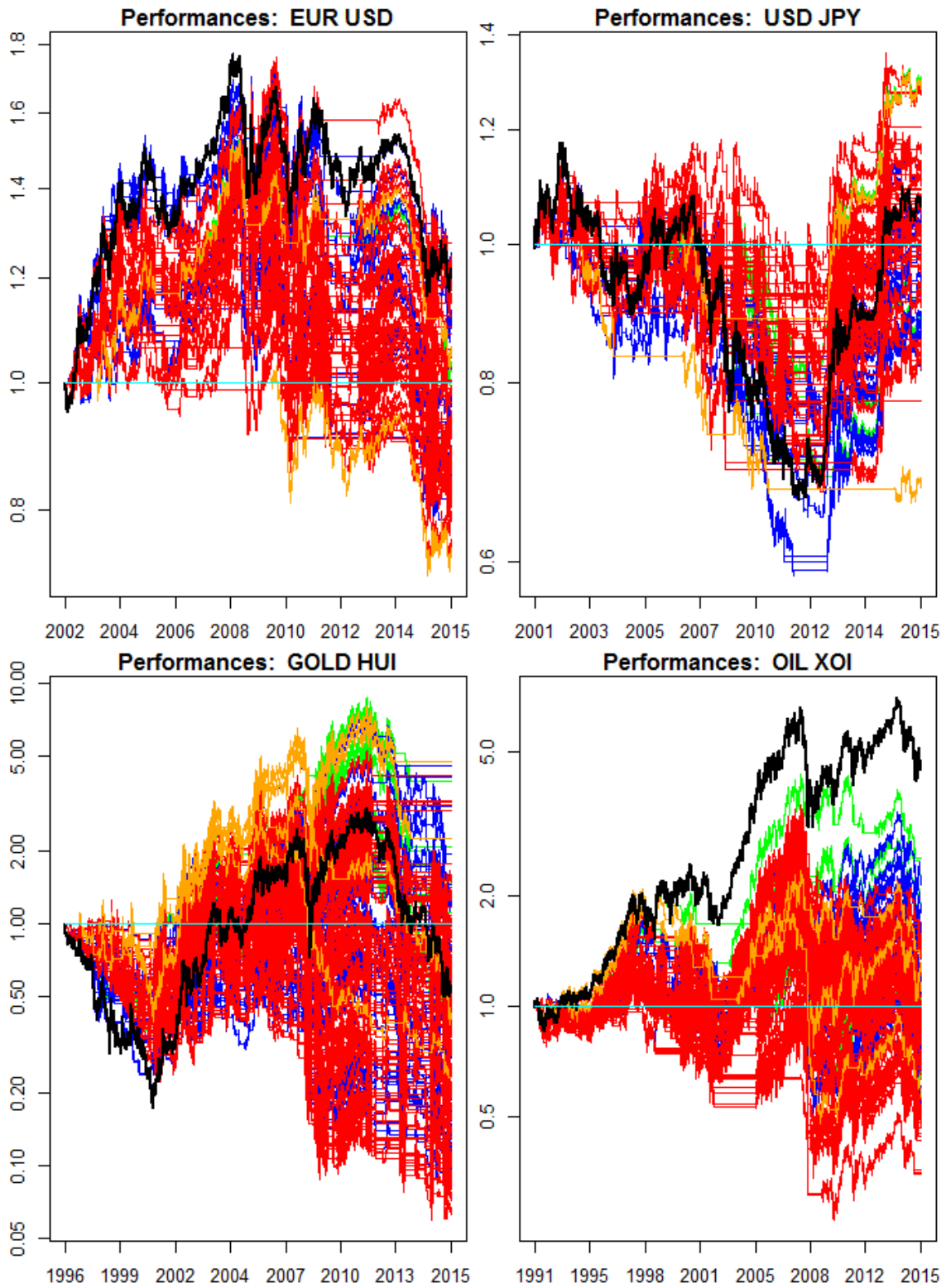


Figure 8: Comparison without fees 2, Simple, Auto123-MACD, Auto123-WAVE 1, Auto123-WAVE 2, Auto123-RENKO, Benchmark

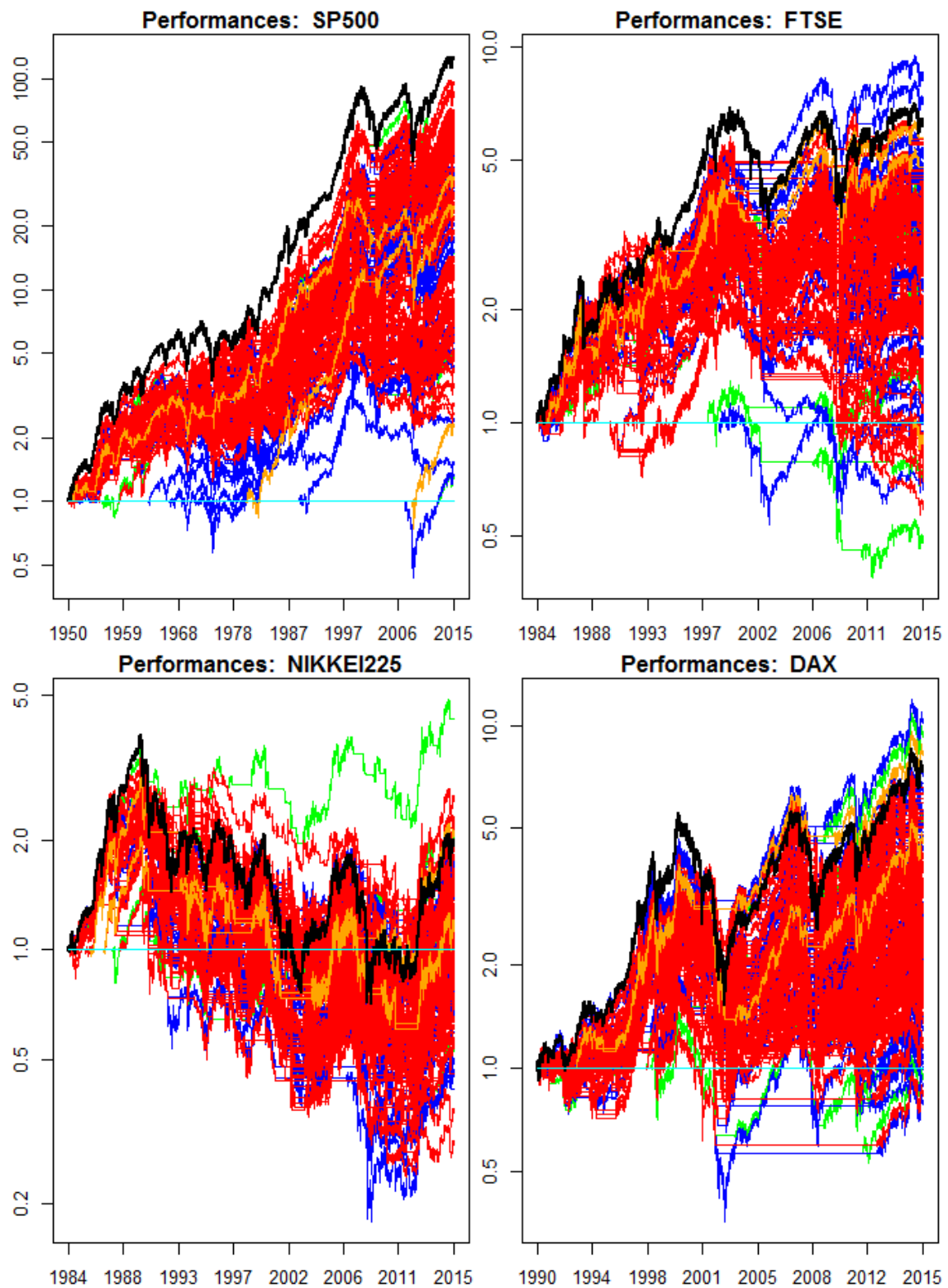


Figure 9: Comparison with fees 1, Simple, Auto123-MACD, Auto123-WAVE 1, Auto123-WAVE 2, Auto123-RENKO, Benchmark

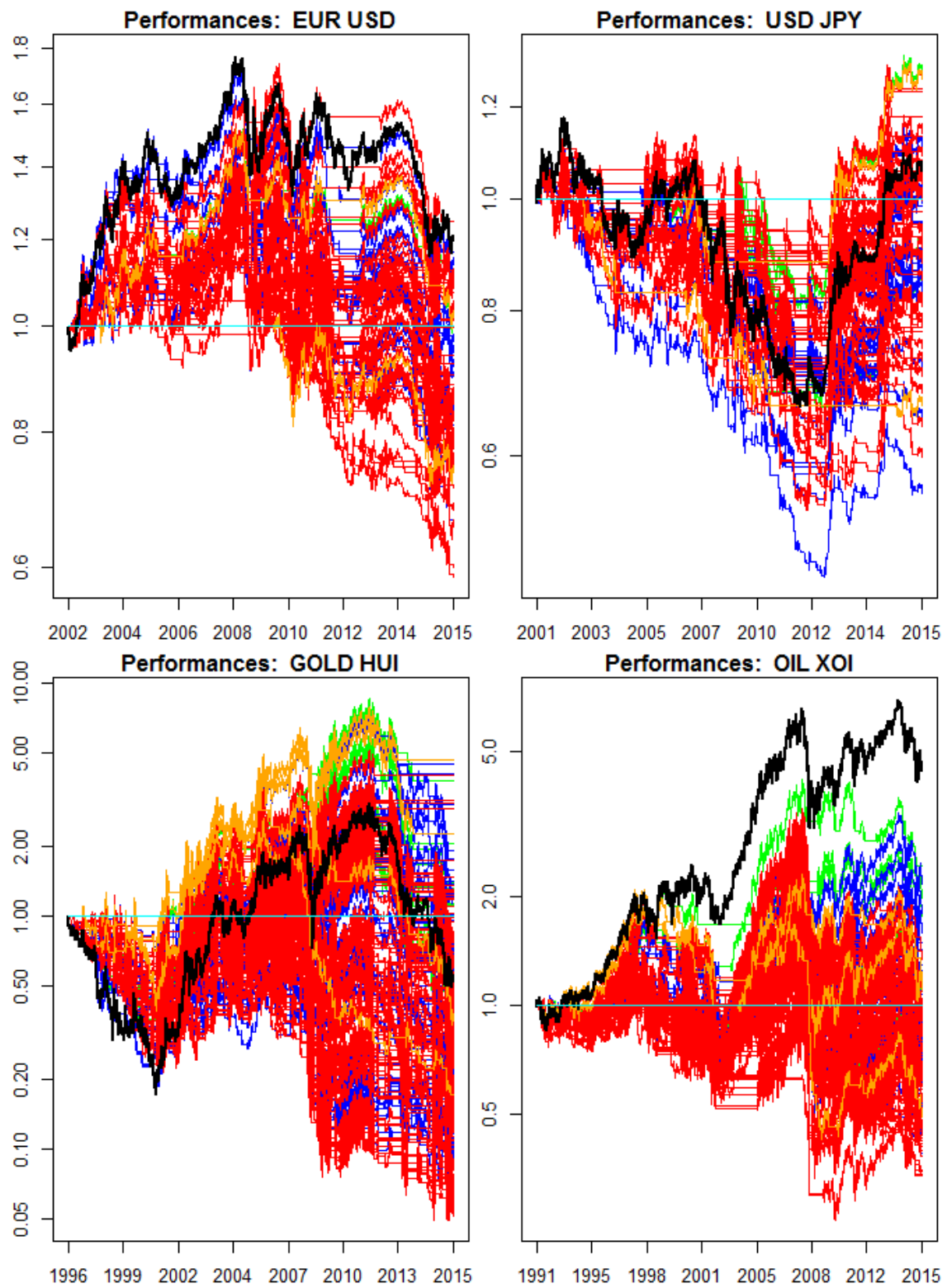


Figure 10: Comparison with fees 2, **Simple**, **Auto123-MACD**, **Auto123-WAVE 1**, **Auto123-WAVE 2**, **Auto123-RENKO**, **Benchmark**

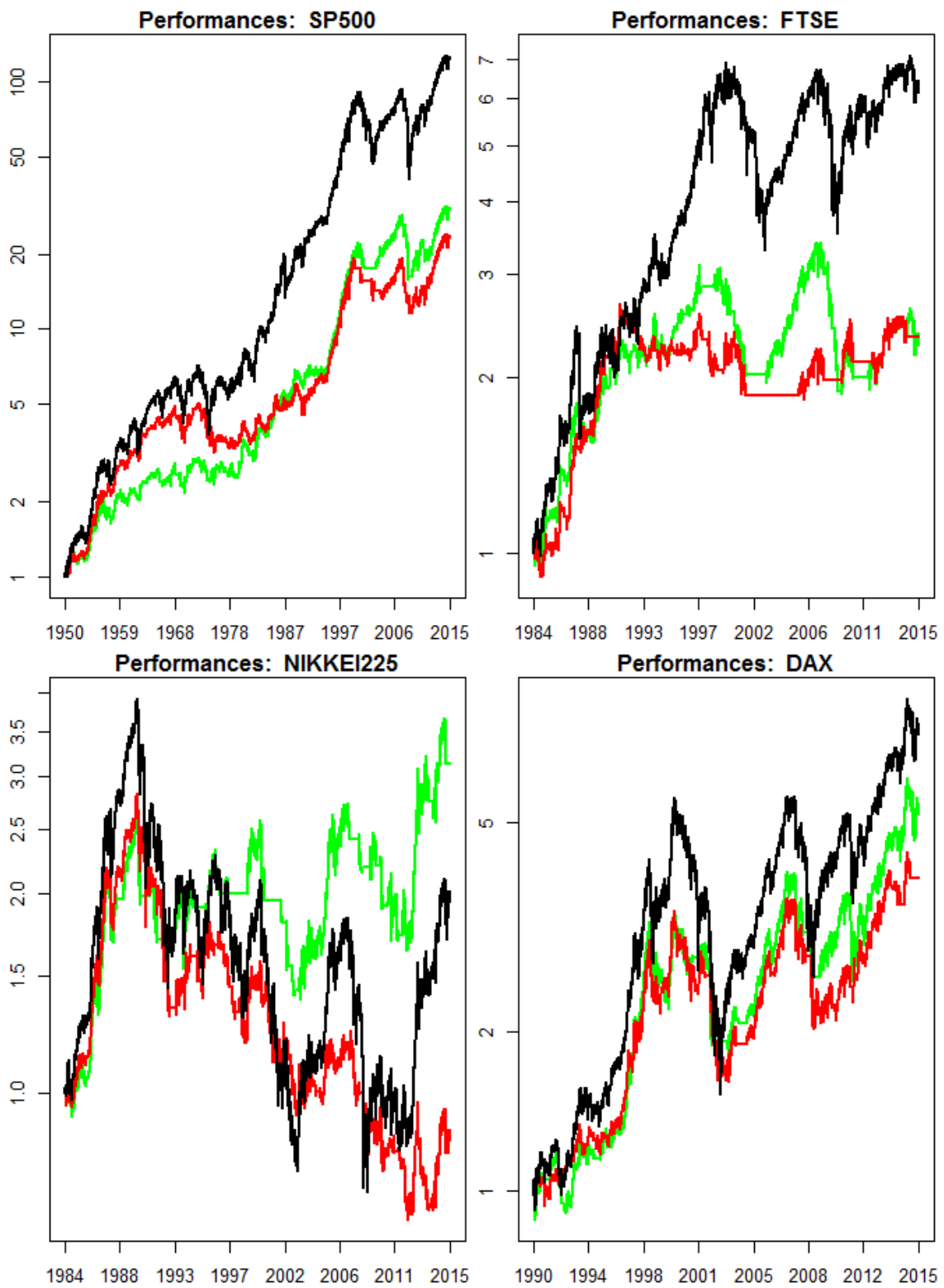


Figure 11: **BestSimple**, **BestAuto**, **Benchmark**

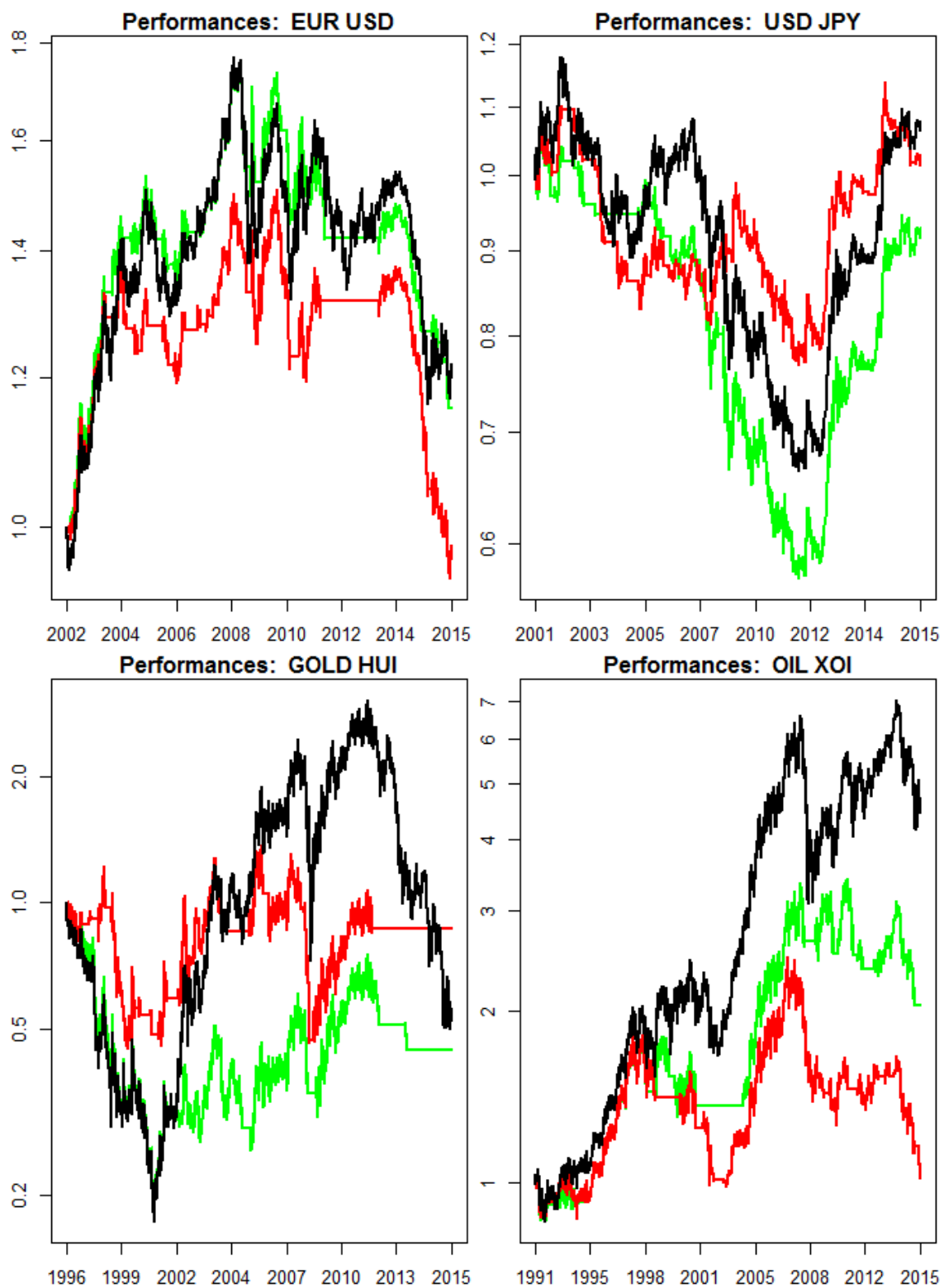


Figure 12: **BestSimple**, **BestAuto**, **Benchmark**

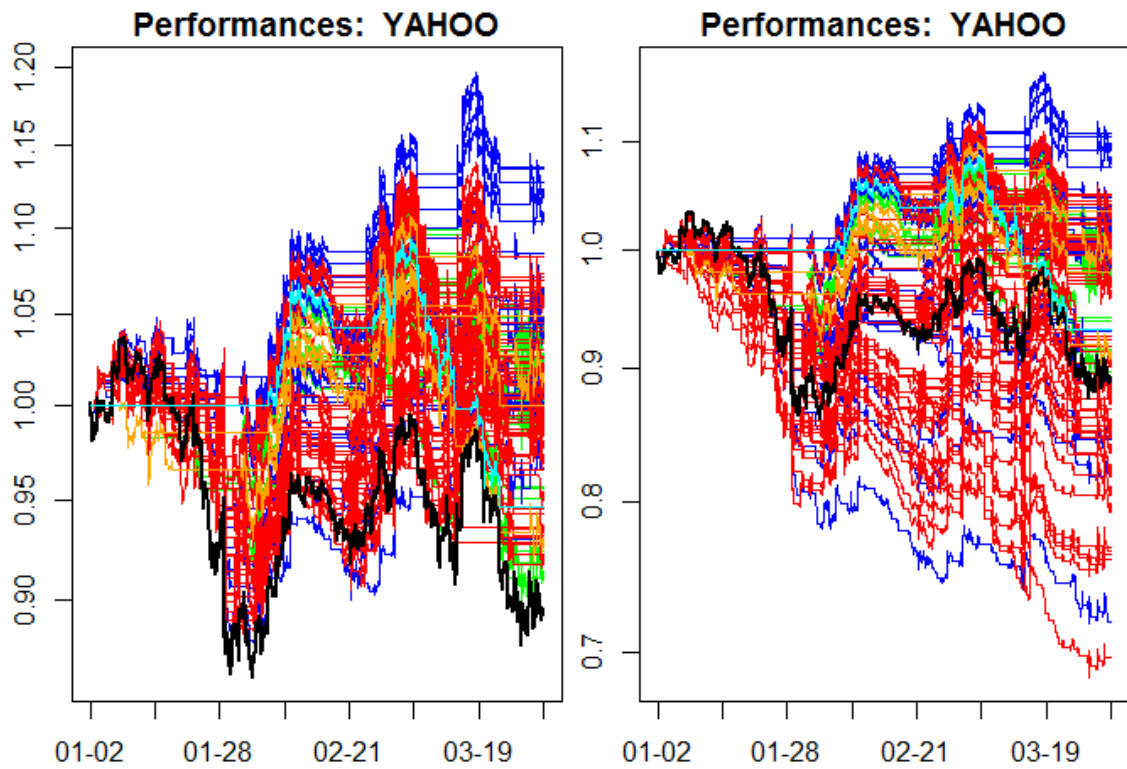


Figure 13: Comparison without (left) and with fees (right), **Simple**, **Auto123-MACD**, **Auto123-WAVE 1**, **Auto123-WAVE 2**, **Auto123-RENKO**, **Benchmark**

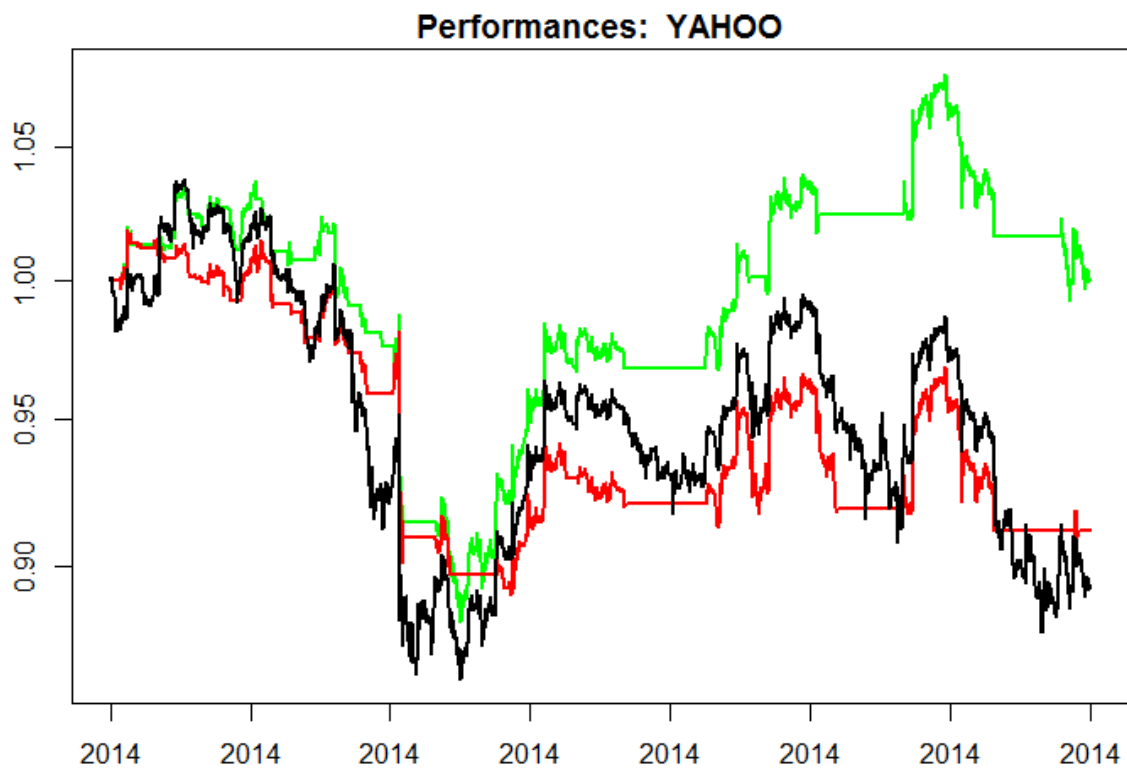


Figure 14: **BestSimple**, **BestAuto**, **Benchmark**

The first thing to notice is that considering trading fees both algorithms perform mostly worse than the benchmark (see Figures 7-14; the y-axis is in logarithmic-scale). This is not surprising since the fees can make up a lot if several trades are done. The comparisons with fees yield a more realistic view.

There are three cases where the Automatic-1-2-3-algorithm performs significantly better than the simple algorithm and the benchmark (FTSE in Figure 9, NIKKEI225 in Figure 9, GOLD HUI in Figure 10) versus six cases where either the simple algorithm or the benchmark perform better or where there is no big difference at all.

Also in the “best of” combinations (Figures 11, 12 and 14) the Automatic-1-2-3-algorithm is not significantly better than the others, again both algorithms are mostly dominated by the benchmark. If you would include trading fees in this cases the superiority of the benchmark would even be clearer.

4 Discussion

As we see, the high complexity of the Automatic-1-2-3-algorithm we get through the variety of the parameters does not really pay off. In most cases, it was either beaten by the simple algorithm or the benchmark.

Of course, this cannot be viewed as a general result since the data sets are only a small selection of available data. Already in this relative small selection we can observe very different outcomes.

This comparison was just made with one trading strategy (buy in up-trend, sell in down-trend). But other strategies can lead to another behaviour of the algorithms.

The results are also highly dependent on the observation period. Even on the same data sets the results can change a lot if a few time points are omitted.

So it remains unclear, if the outcome of this paper can be generalized, but it surely gives strong evidence. Further studies regarding different strategies, other data sets and varying observation times are possible.

5 Appendix A: References

- [1] Börse Frankfurt, *Aktien, Kurse, Charts und Nachrichten*, <http://www.boerse-frankfurt.de/>, 2016.
- [2] M.A.H. Dempster and C.M. Jones, *Can channel pattern trading be profitably automated?* Eur. J. Finance, 2002, 8, 275–301.
- [3] Y. Hafizogullari, S. Maier-Paape and A. Platen, *Empirical study of the 1–2-3-trend indicator, preprint*, <http://www.instmath.rwth-aachen.de/preprints>, 2013.
- [4] LOBSTER, *academic data*, <https://lobsterdata.com/>, 2014.
- [5] S. Maier-Paape, *Automatic One Two Three*, Quantitative Finance, 2013.
- [6] R, *The r project for statistical computing*, <http://www.r-project.org/>, 2016.
- [7] R. Russel, *The Dow Theory Today*, 1981.
- [8] Z.K. Silagadze, *Moving Mini-Max – a new indicator for technical analysis*. IFTA J., 2011, 11, 46–49.
- [9] M. Tronnier, *Automatisierter Korrekturhandel, Bachelor-Arbeit, RWTH Aachen*, <http://www.vtad.de/forschungsarbeiten>, 2010.
- [10] K. Wache, *Automatisierter Bewegungshandel, Bachelor-Arbeit, RWTH Aachen*, <http://www.vtad.de/forschungsarbeiten>, 2010.
- [11] Wikipedia, *The Free Encyclopedia*, <https://en.wikipedia.org/>, 2016.
- [12] Yahoo Finance, *Business Finance, Stock Market, Quotes*, <http://finance.yahoo.com/>, 2016.

6 Appendix B: Abstract / Zusammenfassung

Zusammenfassung

Der Automatic-1-2-3-Algorithmus ist ein Trendindikator, der von S. Maier-Paape vorgestellt wurde [5]. Mit Hilfe eines sogenannten SAR-Prozesses werden die relevanten Minima und Maxima einer Zeitreihe extrahiert und anhand dieser lassen sich Trends klar definieren. Die so erkannten Trends sind dann maßgeblich für Handelsstrategien. Dieser Algorithmus wird jedoch durch mehrere Parameter gesteuert, wodurch sich sehr viele Variationsmöglichkeiten ergeben. Diese Arbeit untersucht die Frage, ob sich diese Komplexität auszahlt, oder ob nicht ein viel simplerer Algorithmus ähnliche Ergebnisse erzielt. Bei der Analyse und den Grafiken wird die Statistiksoftware R verwendet [6]. Am Ende zeigt sich, dass ein Algorithmus mit nur einem Parameter gleich gut abschneidet.

Schlüsselwörter: Automatic-1-2-3, Trendindikator, Evaluation der Performance, Min-Max-Prozess, SAR-Prozess

Abstract

The Automatic-1-2-3-algorithm is a trend indicator which was introduced by S. Maier-Paape [5]. The relevant minima and maxima of a time series are extracted with the aid of a so called SAR-process. Trends can clearly be defined using them. This algorithm is controlled through several parameters, what yields the possibility of large variation. This thesis examines the question whether this complexity pays off or whether a much simpler algorithm produces similar results. For computation and creating figures the statistical software R is used [6]. In the end, it turns out that an algorithm with only one parameter performs as well.

Key words: Automatic-1-2-3, trend indicator, evaluation of performance, Min-Max-process, SAR-process

7 Appendix C: R-Code

```
#MACD function
#Input: time series with Date/Time, Open, High, Low, Close, Volume, Adjusted Close
#Parameters:
#   -fastparam: timeparameter for the short period exponential moving average
#   -slowparam: timeparameter for the long period exponential moving average
#   -triggerparam: timeparameter for the signal line
#   -del: value for delta
#   -delfix: method for delta (0: fixed, 1: multiple of Close, 2: multiple of ATR)
#   -integrated: indicating if the exceedance should be cummulated
#   -timescale: multiplier for fast-,slow- and triggerparam
#   -spanne: timeparameter for ATR
#   -priceindex: index which price of the time series should be used
#Output: vector of directions (-1, 0 or 1)
macd<-function(rd,fastparam=12,slowparam=26,triggerparam=9,del=5,delfix=1,integrated=T,
               priceindex=7,timescale=1,spanne=100) {
  fastparam<-round(fastparam*timescale)
  slowparam<-round(slowparam*timescale)
  triggerparam<-round(triggerparam*timescale)
  M<-nrow(rd)
  slow<-numeric(M)
  fast<-numeric(M)
  MACD<-numeric(M)
  signal<-numeric(M)
  direction<-numeric(M)
  TR<-numeric(M)
  for (i in 1:M) {
    if (delfix==1) {
      delta<-del*rd[i,priceindex]/100
    } else if (delfix==2) {
      if(i>spanne) {
        TR[i]<-max((rd[i,3]-rd[i,4]),abs(rd[i,3]-rd[i-1,5]),abs(rd[i,4]-rd[i-1,5]))
        ATR<-mean(TR[(i-spanne):i])
      } else if (i>1) {
        TR[i]<-max((rd[i,3]-rd[i,4]),abs(rd[i,3]-rd[i-1,5]),abs(rd[i,4]-rd[i-1,5]))
        ATR<-mean(TR[1:i])
      } else {
        TR[i]<-(rd[i,3]-rd[i,4])
        ATR<-TR[i]
      }
      delta<-del*ATR
    } else {delta<-del}
    if (i<=fastparam) {
      fast[i]<-mean(rd[max(1,(i-(fastparam-1))):i,priceindex])
      slow[i]<-mean(rd[max(1,(i-(slowparam-1))):i,priceindex])
    }
    else if (i<=slowparam) {
      fast[i]<-rd[i,priceindex]*2/(fastparam+1)+fast[(i-1)]*(1-2/(fastparam+1))
      slow[i]<-mean(rd[max(1,(i-(slowparam-1))):i,priceindex])
    }
    else {
      fast[i]<-rd[i,priceindex]*2/(fastparam+1)+fast[(i-1)]*(1-2/(fastparam+1))
      slow[i]<-rd[i,priceindex]*2/(slowparam+1)+slow[(i-1)]*(1-2/(slowparam+1))
    }
    MACD[i]<-fast[i]-slow[i]
    if (i<=(slowparam+triggerparam)) {
      signal[i]<-mean(MACD[max(1,(i-(triggerparam-1))):i])
    }
    else {
      signal[i]<-MACD[i]*2/(triggerparam+1)+signal[(i-1)]*(1-2/(triggerparam+1))
    }
  }
  if (i>=(slowparam+triggerparam)) {
    if (integrated) {
      integral<-0
      if (MACD[i]!=signal[i]) {
        j<-i
        sig<-sign(MACD[i]-signal[i])
      }
    }
  }
}
```

```

        while ((sign(MACD[j]-signal[j])==sig)&(j>=1)) {
            integral<-integral+MACD[j]-signal[j]
            j<-(j-1)
        }
    }
    direction[i]<-ifelse(((integral>=delta)|((direction[i-1]==1)&(integral>=
delta))),
                        1,ifelse(((integral<(-delta))|((direction[i-1]==-1)&
                        (integral<delta))),-1,0))
    }
    else {
        direction[i]<-ifelse((MACD[i]-signal[i]>=delta)|((direction[i-1]==1)&
        (MACD[i]-signal[i]>=-delta))),1,
        ifelse((MACD[i]-signal[i]<(-delta))|
        ((direction[i-1]==-1)&(MACD[i]-signal[i]<delta))),-1,0))
    }
}
}
direction
}

```

```

#renko box function
#Input: time series with Date/Time, Open, High, Low, Close, Volume, Adjusted Close
#Parameters:
#   -bsize: boxsize
#   -bsfix: method for boxsize (0: fixed, 1: multiple of Close, 2: multiple of ATR)
#   -revamount: minimum of same boxes before direction change
#   -spanne: timeparameter for ATR
#   -priceindex: index which price of the time series should be used
#Output: vector of directions (-1, 0 or 1)
renko<-function(rd,bsize=5,bsfix=1,revamount=2,priceindex=7,spanne=100) {
    M<-nrow(rd)
    lbrick<-c(0,rd[1,priceindex],rd[1,priceindex])
    direction<-numeric(M)
    TR<-numeric(M)
    for (i in 1:M) {
        if (bsfix==1) {
            boxsize<-(bsize*rd[i,priceindex]/100)
        } else if (bsfix==2) {
            if(i>spanne) {
                TR[i]<-max((rd[i,3]-rd[i,4]),abs(rd[i,3]-rd[i-1,5]),abs(rd[i,4]-rd[i-1,5]))
                ATR<-mean(TR[(i-spanne):i])
            } else if (i>1) {
                TR[i]<-max((rd[i,3]-rd[i,4]),abs(rd[i,3]-rd[i-1,5]),abs(rd[i,4]-rd[i-1,5]))
                ATR<-mean(TR[1:i])
            } else {
                TR[i]<-(rd[i,3]-rd[i,4])
                ATR<-TR[i]
            }
            boxsize<-bsize*ATR
        } else {boxsize<-bsize}
        if (lbrick[1]==1) {
            if(rd[i,priceindex]>lbrick[3]) {lbrick[3]<-rd[i,priceindex]}
            else if((lbrick[3]-rd[i,priceindex])>=boxsize*(1+revamount)){
                lbrick<-c(-1,rd[i,priceindex],lbrick[2])
            }
        }
        else if (lbrick[1]==-1){
            if(rd[i,priceindex]<lbrick[2]) {lbrick[2]<-rd[i,priceindex]}
            else if((rd[i,priceindex]-lbrick[2])>=boxsize*(1+revamount)){
                lbrick<-c(1,lbrick[3],rd[i,priceindex])
            }
        }
        else {
            if ((rd[i,priceindex]-lbrick[2])>=boxsize) {lbrick<-
c(1,lbrick[3],rd[i,priceindex])}
            else if ((lbrick[3]-rd[i,priceindex])>=boxsize) {lbrick<-c(-1,rd[i,priceindex],
lbrick[2])}
        }
        direction[i]<-lbrick[1]
    }
    direction
}

```

```

#crosscorrelation function
#Input: time series with Date/Time, Open, High, Low, Close, Volume, Adjusted Close
#Parameters:
#   -priceindex: index which price of the time series should be used
#Output: dominant wavelength
cross<-function(rd,priceindex=7) {
  M<-nrow(rd)
  nv<-1:min(300,M/3)
  cv<-numeric(length(nv))
  a<-rd[,priceindex]
  for (n in nv[-1]) {
    nf<-floor(n/2)
    b<-numeric(M)
    x<-numeric(M)
    for (t in (nf+1):(M-nf)) {
      b[t]<-1/(2*nf+1)*sum(a[(t-nf):(t+nf)])
      x[t]<-a[t]-b[t]
    }
    cv[n]<-cor(x[(nf+1):(M-n-nf)],x[(nf+n+1):(M-nf)])
  }
  return(which(cv==max(cv)))
}

#Wavelength calibration function
#Input: time series with Date/Time, Open, High, Low, Close, Volume, Adjusted Close
#Parameters:
#   -mi: minimal value for timescale
#   -ma: maximal vlaue for timescale
#   -ste: stepwidth for timescale
#   -for the rest: see functions macd and auto123
#Output: same as function auto123
#Plot: same as function auto123
wave<-function(rd,priceindex=7,mi=0.2,ma=6,ste=0.1,del=0.3,spanne=100,delfix=1,
               integrated=F,a=1,b=nrow(rd)) {
  M<-nrow(rd)
  tsv<-seq(from=mi,to=ma,by=ste)
  nstar<-cross(rd,priceindex)
  distmin<-M
  bestind<-1
  for (i in 1:length(tsv)) {
    invisible(nts<-auto123(rd,priceindex=priceindex,del=del,delfix=delfix,
                          integrated=
ed=integrated,timescale=tsv[i],calib=T,spanne=spanne))
    if ((nts-nstar)^2<distmin) {
      distmin<-(nts-nstar)^2
      bestind<-i
    }
  }
  return(auto123(rd,a=a,b=b,priceindex=priceindex,del=del,delfix=delfix,
                integrated=integrated,timescale=tsv[bestind],calib=F,spanne=spanne))
}

#Automatic-1-2-3 Trend Indicator function
#Input: time series with Date/Time, Open, High, Low, Close, Volume, Adjusted Close
#Parameters:
#   -sar: which sar-process should be used ("macd" or "renko")
#   -a: start date/time for plot
#   -b: end date/time for plot
#   -plotdate: indicating if the date should be plotted on the x-axis
#   -calib: indicator, used to look for the dominant wavelength in function wave
#   -for the rest: see functions macd, renko, wave
#Output: time series with development of the portfoliovalue (uptrend->buy,downtrend->sell)
#Plot: development of the portfoliovalue in comparison to the assets value
auto123<-function(rd,sar="macd",a=1,b=nrow(rd),fastparam=12,slowparam=26,triggerparam=9,
                  del=5,delfix=1,integrated=T,bsize=5,bsfix=1,revamount=2,priceindex=7,
                  plotdate=F,timescale=1,spanne=100,calib=F) {
  M<-nrow(rd)

```

```

if (sar=="renko") {direction<-renko(rd,bsize,bsfix,revamount,priceindex,spanne)}
else {direction<-macd(rd,fastparam,slowparam,triggerparam,del,delfix,integrated,
      priceindex,timescale,spanne)}

status<-direction
excep<-rep(1,M)
trend<-rep(0,M)
uppre<-rep(0,M)
downpre<-rep(0,M)
uptrigger<-rep(0,M)
downtrigger<-rep(0,M)
stoptrigger<-rep(0,M)
movementtrigger<-rep(0,M)
movementnumber<-rep(0,M)
lastminbar<-NULL
lastmaxbar<-NULL
lastmaxbarv<-NULL
lastminbarv<-NULL
tempmaxbar<-NULL
tempminbar<-NULL
bcs<-"Hold"
lastdec<-"None"
pcstock<-0
pcmonev<-1
pcstockd<-numeric(M)
pcmonevd<-numeric(M)
pcmonevd[1]<-1
datecv<-rd[1,1]
valuecv<-rd[1,priceindex]
notr<-numeric(M)
trhap<-0
for (i in 2:M) {
  trhap<-0
  datasf<-rd[(1:i),priceindex]
  if (direction[i-1]==0 & direction[i]==1) {
    lastminbar<-which(datasf==min(datasf))[1]
    lastminbarv<-c(lastminbarv,lastminbar)
    tempmaxbar<-lastminbar
  }
  if (direction[i-1]==0 & direction[i]==-1) {
    lastmaxbar<-which(datasf==max(datasf))[1]
    lastmaxbarv<-c(lastmaxbarv,lastmaxbar)
    tempminbar<-lastmaxbar
  }
  if(excep[i-1]==-1) {
    l1<-(direction[i-1]*direction[i]==-1)
    l2<-F
    l3<-F
    if (length(lastmaxbar)>0) {l2<-(direction[i-1]==1 & rd[lastmaxbar,3]<=rd[i,3])}
    if (length(lastminbar)>0) {l3<-(direction[i-1]==-1 & rd[lastminbar,4]>=rd[i,4])}
    excep[i]<-ifelse(l1|l2|l3,1,-1)
  }
  else if(direction[i-1]==direction[i]) {
    l2<-F
    l3<-F
    if (length(lastmaxbar)>0) {l2<-(direction[i]==-1 & rd[lastmaxbar,3]<=rd[i,3])}
    if (length(lastminbar)>0) {l3<-(direction[i]==1 & rd[lastminbar,4]>=rd[i,4])}
    excep[i]<-ifelse(l2|l3,-1,1)
  }
  status[i]<-excep[i]*direction[i]
  alpha<-ifelse(((length(lastmaxbar)+length(lastminbar))<2),0,
    ((lastmaxbar==lastminbar)*1))
  if (status[i-1]==1) {
    if (rd[tempmaxbar,3]<=rd[i,3]) {tempmaxbar<-i}
    if (status[i]==-1) {
      lastmaxbar<-tempmaxbar
      lastmaxbarv<-c(lastmaxbarv,lastmaxbar)
      tempminbar<-
    }
    (which(rd[(lastmaxbar+alpha):i,4]==min(rd[(lastmaxbar+alpha):i,4]))[1]+
      (lastmaxbar+alpha)-1)
  }
  if (status[i-1]==-1) {
    if (rd[tempminbar,4]>=rd[i,4]) {tempminbar<-i}
    if (status[i]==1) {
      lastminbar<-tempminbar
      lastminbarv<-c(lastminbarv,lastminbar)
    }
  }
}

```

```

tempmaxbar<-
(which(rd[(lastminbar+alpha):i,3]==max(rd[(lastminbar+alpha):i,3]))[1]+
      (lastminbar+alpha)-1)
}
}
if ((length(lastminbarv)+length(lastmaxbarv))>=4) {
  if (status[i]==1) {
    max1<-rd[tempmaxbar,3]
    min1<-rd[lastminbarv[length(lastminbarv)],4]
    max2<-rd[lastmaxbarv[length(lastmaxbarv)],3]
    min2<-rd[lastminbarv[length(lastminbarv)-1],4]
  }
  else if (status[i]==-1) {
    min1<-rd[tempminbar,4]
    max1<-rd[lastmaxbarv[length(lastmaxbarv)],3]
    min2<-rd[lastminbarv[length(lastminbarv)],4]
    max2<-rd[lastmaxbarv[length(lastmaxbarv)-1],3]
  }
  if (max1>max2&min1>=min2&(status[i]==1|trend[i-1]==1)) {trend[i]<-1}
  else if (max1<=max2&min1<min2&(status[i]==(-1)|trend[i-1]==(-1))) {trend[i]<-(-1)}
  else if ((trend[i-1]==1|trend[i-1]==0.5)&(max1<=max2&min1>=min2)) {trend[i]<-0.5}
  else if ((trend[i-1]==(-1)|trend[i-1]==(-0.5))&(max1<=max2&min1>=min2))
    {trend[i]<-(-0.5)}
  if (trend[i]==0) {
    if(status[i]==1&min1>=min2) {uppre[i]<-0.5}
    if(status[i]==(-1)&max1<=max2) {downpre[i]<-(-0.5)}
  }
  if (trend[i]==1|uppre[i]==0.5) {
    uptrigger[i]<-rd[lastmaxbar,4]
    stoptrigger[i]<-rd[lastminbar,5]
  }
  if (trend[i]==(-1)|uppre[i]==(-0.5)) {
    downtrigger[i]<-rd[lastminbar,5]
    stoptrigger[i]<-rd[lastmaxbar,4]
  }
  if (trend[i]==1&datasf[i]>=uptrigger[i]) {movementtrigger[i]<-1}
  if (trend[i]==(-1)&datasf[i]<=downtrigger[i]) {movementtrigger[i]<-(-1)}
  if (trend[i]==1|trend[i]==(-1)) {movementnumber[i]<-ifelse(trend[i]==trend[i-1],
    movementnumber[i-1]+movementtrigger[i],
    movementtrigger[i])}
}
if (trend[i]==1 & trend[(i-1)]!=1 & lastdec!="Buy") {
  bcs[(length(bcs)+1)]<-"Buy"
  lastdec<-"Buy"
  pcstock[(length(pcstock)+1)]<-pcmoneymoney[(length(pcmoneymoney))]/rd[i,priceindex]
  pcmoneymoney[(length(pcmoneymoney)+1)]<-0
  trhap<-1
}
else if (trend[i]==(-1) & trend[(i-1)]!=(-1) & lastdec=="Buy" ) {
  bcs[(length(bcs)+1)]<-"Cash"
  lastdec<-"Cash"
  pcmoneymoney[(length(pcmoneymoney)+1)]<-pcstock[(length(pcstock))]*rd[i,priceindex]
  pcstock[(length(pcstock)+1)]<-0
  trhap<-1
}
else {bcs[(length(bcs)+1)]<-"Hold"}
datecv[(length(datecv)+1)]<-rd[i,1]
valuecv[(length(valuecv)+1)]<-rd[i,priceindex]

pcstockd[i]<-pcstock[length(pcstock)]
pcmoneymoneyd[i]<-pcmoneymoney[length(pcmoneymoney)]
notr[i]<-notr[i-1]+trhap
}
perf<-pcstockd*rd[,priceindex]+pcmoneymoneyd
performance<-data.frame(date=rd[,1],value=perf,trades=notr)
parold<-par() $mar
par(mar=(parold+c(0,0,0,2)))
plot(rd[,priceindex],type="l",xlim=c(a,b),ylim=c(min(perf[a:b]*rd[1,7],
  rd[a:b,priceindex]),max(perf[a:b]*rd[1,7],rd[a:b,priceindex])),
  xlab="time",ylab="price", xaxt='n')
lines(rd[,4],lty=2)
lines(rd[,5],lty=2)
ax-
is(side=4,at=pretty(range(rd[,7]/rd[1,7]))*rd[1,7],labels=pretty(range(rd[,7]/rd[1,7])))
mtext("performance", side = 4, line = 3, cex = par("cex.lab"))
par(mar=parold)
lines(performance$value*rd[1,7],col="darkgreen")

```

```

seqx<-round(seq(from=1,to=M,length=8))
if (plotdate) {axis(1, at=seqx,labels=format(as.Date(rd[seqx,1]),"%Y"))}
else {axis(1, at=seqx)}
title(expression("Price, " * phantom("Min-Max") * ", " * phantom("Buy") * ", " *
phantom("Cash") * ", " * phantom("Strategy") * ""),
col.main = "black")
title(expression(phantom("Price, " * "Min-Max" * phantom(", ") * phantom("Buy") *
phantom(", ") * phantom("Cash") * phantom(", ") *
phantom("Strategy") * ""), col.main = "orange")
title(expression(phantom("Price, " * phantom("Min-Max") * phantom(", ") * "Buy" *
phantom(", ") * phantom("Cash") * phantom(", ") *
phantom("Strategy") * ""), col.main = "green")
title(expression(phantom("Price, " * phantom("Min-Max") * phantom(", ") *
phantom("Buy") * phantom(", ") * "Cash" * phantom(", ") *
phantom("Strategy") * ""), col.main = "red")
title(expression(phantom("Price, " * phantom("Min-Max") * phantom(", ") *
phantom("Buy") * phantom(", ") * phantom("Cash") *
phantom(", ") * "Strategy" * ""), col.main = "darkgreen")
if((length(lastmaxbarv)>0 & length(lastminbarv)>0) {
z<-((lastminbarv[1]<=lastmaxbarv[1])*1)
x<-sort(c(lastminbarv,lastmaxbarv))
y<-numeric(length(x))
for (q in 1:length(x)) {
y[q]<-rd[x[q], (3+z)]
z<-(1-z)
}
lines(x,y,col="orange")
}
for (i in as.numeric(datecv[bcs=="Buy"])) {abline(v=i,col="green")}
for (i in as.numeric(datecv[bcs=="Cash"])) {abline(v=i,col="red")}
if (lastdec=="Buy") {abline(v=as.numeric(rd[M,1]),col="red");
performance$trades[M]<-performance$trades[M]+1}
print(paste("performance :", (pcmoneymoney[length(pcmoneymoney)]+
pcstock[length(pcstock)]*rd[M,priceindex])))
if(!calib) {return(performance)} else {
if (length(lastminbarv)>=2 & length(lastmaxbarv)>=2) {
permin<-mean(diff(sort(lastminbarv)))
permax<-mean(diff(sort(lastmaxbarv)))
return(mean(permin,permax))
} else {
return(M)
}
}
}
}

#Simple algorithm function
#Input: time series with Date/Time, Open, High, Low, Close, Volume, Adjusted Close
#Parameters:
# -p: value for p
# -a: start date/time for plot
# -b: end date/time for plot
# -plotdate: indicating if the date should be plotted on the x-axis
# -priceindex: index which price of the time series should be used
#Output: time series with development of the portfoliovalue (uptrend->buy,downtrend->sell)
#Plot: development of the portfoliovalue in comparison to the assets value
easystrat<-function(rd,p=0.05,a=1,b=nrow(rd),priceindex=7,plotdate=F) {
M<-nrow(rd)
datev<-rd[1,1]
valuev<-rd[1,priceindex]
minmaxv<-c("Start",
trendv<-rep("None",3)
datecv<-rd[1,1]
valuecv<-rd[1,priceindex]
bcs<-c("Hold",
lastdec<-c("None",
pcstock<-0
pcmoneymoney<-1
pcstockd<-numeric(M)
pcmoneymoneyd<-numeric(M)
pcmoneymoneyd[1]<-1
tempmin<-rd[1,priceindex]
tempmax<-rd[1,priceindex]
notr<-numeric(M)

```

```

trhap<-0
for (i in 2:M) {
  trhap<-0
  tempmin<-min(rd[i,priceindex],tempmin)
  tempmax<-max(rd[i,priceindex],tempmax)
  updatetr<-F
  newextr<-F
  if(minmaxv[length(minmaxv)]=="Min") {
    if(valuev[length(minmaxv)]>rd[i,priceindex]) {
      datev[length(minmaxv)]<-rd[i,1]
      valuev[length(minmaxv)]<-rd[i,priceindex]
      updatetr<-T
    }
    else if((rd[i,priceindex]-tempmin)/tempmin>=p) {
      datev[length(minmaxv)+1]<-rd[i,1]
      valuev[length(minmaxv)+1]<-rd[i,priceindex]
      minmaxv[length(minmaxv)+1]<- "Max"
      tempmax<-rd[i,priceindex]
      updatetr<-T
      newextr<-T
    }
  }
  else if(minmaxv[length(minmaxv)]=="Max") {
    if(valuev[length(minmaxv)]<rd[i,priceindex]) {
      datev[length(minmaxv)]<-rd[i,1]
      valuev[length(minmaxv)]<-rd[i,priceindex]
      updatetr<-T
    }
    else if((tempmax-rd[i,priceindex])/tempmax>=p) {
      datev[length(minmaxv)+1]<-rd[i,1]
      valuev[length(minmaxv)+1]<-rd[i,priceindex]
      minmaxv[length(minmaxv)+1]<- "Min"
      tempmin<-rd[i,priceindex]
      updatetr<-T
      newextr<-T
    }
  }
  else {
    if((rd[i,priceindex]-tempmin)/tempmin>=p) {
      datev[length(minmaxv)+1]<-rd[i,1]
      valuev[length(minmaxv)+1]<-rd[i,priceindex]
      minmaxv[length(minmaxv)+1]<- "Max"
      updatetr<-T
      newextr<-T
    }
    else if ((tempmax-rd[i,priceindex])/tempmax>=p) {
      datev[length(minmaxv)+1]<-rd[i,1]
      valuev[length(minmaxv)+1]<-rd[i,priceindex]
      minmaxv[length(minmaxv)+1]<- "Min"
      updatetr<-T
      newextr<-T
    }
  }
}
tempM<-length(valuev)
if(tempM>3 & updatetr==T) {
  if(minmaxv[tempM]=="Min") {
    if(valuev[tempM]>=valuev[(tempM-2)] & valuev[(tempM-1)]>valuev[(tempM-3)] &
      trendv[length(trendv)]=="Up")
    {trendv[(length(trendv)+1)*newextr]<- "Up"}
    else if(valuev[tempM]<valuev[(tempM-2)] & valuev[(tempM-1)]<=valuev[(tempM-3)]) &
      trendv[(length(trendv)+1)*newextr]<- "Down" } else{
      trendv[(length(trendv)+1)*newextr]<-
"None"}
  }
  else if(minmaxv[tempM]=="Max") {
    if(valuev[tempM]>valuev[(tempM-2)] & valuev[(tempM-1)]>=valuev[(tempM-3)])
    {trendv[(length(trendv)+1)*newextr]<- "Up"}
    else if(valuev[tempM]<=valuev[(tempM-2)] & valuev[(tempM-1)]<valuev[(tempM-3)] &
      trendv[(length(trendv)+1)*newextr]=="Down")
    {trendv[(length(trendv)+1)*newextr]<- "Down" } else {
      trendv[(length(trendv)+1)*newextr]<-
"None"}
  }
}
if(updatetr==T) {
  if (trendv[tempM]=="Up" & trendv[(tempM-1)]!="Up" & lastdec!="Buy") {
    bcs[(length(bcs)+1)]<- "Buy"
    lastdec<- "Buy"
  }
}

```

```

pcstock[(length(pcstock)+1)]<-pcmoneymoney[(length(pcmoney))]/rd[i,priceindex]
pcmoneymoney[(length(pcmoney)+1)]<-0
trhap<-1
}
else if (trendv[tempM]=="Down" & trendv[(tempM-1)]!="Down" & lastdec=="Buy" ) {
  bcs[(length(bcs)+1)]<-"Cash"
  lastdec<-"Cash"
  pcmoneymoney[(length(pcmoney)+1)]<-pcstock[(length(pcstock))]*rd[i,priceindex]
  pcstock[(length(pcstock)+1)]<-0
  trhap<-1
} else {bcs[(length(bcs)+1)]<-"Hold"
datecv[(length(datecv)+1)]<-rd[i,1]
valuecv[(length(valuecv)+1)]<-rd[i,priceindex]
}
pcstockd[i]<-pcstock[length(pcstock)]
pcmoneymoneyd[i]<-pcmoneymoney[length(pcmoney)]
notr[i]<-notr[i-1]+trhap
}
print(paste("performance :", (pcmoneymoney[length(pcmoney)]+
pcstock[length(pcstock)]*rd[M,priceindex])))
perf<-pcstockd*rd[,priceindex]+pcmoneymoneyd
performance<-data.frame(date=rd[,1],value=perf,trades=notr)
parold<-par()$mar
par(mar=(parold+c(0,0,0,2)))
plot(rd[,priceindex],type="l",xlim=c(a,b),ylim=c(min(perf[a:b]*rd[1,7],
rd[a:b,priceindex]),max(perf[a:b]*rd[1,7],
rd[a:b,priceindex])),xlab="time",ylab="price",xaxt='n')
ax-
is(side=4,at=pretty(range(rd[,7]/rd[1,7]))*rd[1,7],labels=pretty(range(rd[,7]/rd[1,7])))
mtext("performance", side = 4, line = 3, cex = par("cex.lab"))
par(mar=parold)
lines(performance$value*rd[1,7],col="darkgreen")
seqx<-round(seq(from=1,to=M,length=8))
if (plotdate) {axis(1, at=seqx,labels=format(as.Date(rd[seqx,1]),"%Y"))} else
{axis(1, at=seqx)}
title(expression("Price, " * phantom("Min-Max") * ", " * phantom("Buy") * ", " *
phantom("Cash") * ", " * phantom("Strategy") * ""),
col.main = "black")
title(expression(phantom("Price, " * "Min-Max" * phantom(", " * phantom("Buy") *
phantom(", " * phantom("Cash") * phantom(", " *
phantom("Strategy") * ""), col.main = "orange")
title(expression(phantom("Price, " * phantom("Min-Max") * phantom(", " * "Buy" *
phantom(", " * phantom("Cash") * phantom(", " *
phantom("Strategy") * ""), col.main = "green")
title(expression(phantom("Price, " * phantom("Min-Max") * phantom(", " *
phantom("Buy") * phantom(", " * "Cash" * phantom(", " *
phantom("Strategy") * ""), col.main = "red")
title(expression(phantom("Price, " * phantom("Min-Max") * phantom(", " *
phantom("Buy") * phantom(", " * phantom("Cash") *
phantom(", " * "Strategy" * ""), col.main = "darkgreen")
lines(as.numeric(datev),valuev,type="l",col="orange")
for (i in as.numeric(datecv[bcs=="Buy"])) {abline(v=i,col="green")}
for (i in as.numeric(datecv[bcs=="Cash"])) {abline(v=i,col="red")}
if (lastdec=="Buy") {abline(v=as.numeric(rd[M,1]),col="red");
performance$trades[M]<-performance$trades[M]+1}
return(performance)
}

```

8 Appendix D: CURRICULUM VITAE

Personal details:

Name: Thomas Sinkovics
Degree: Bachelor of Science
Date of birth: 21.05.1991
Place of birth: Oberwart, Austria
Citizenship: Austria
Address: Landstraßer Hauptstraße 112/2/16
1030 Wien, Austria
Phone: +43 664 48 69 551
E-Mail: Thomas.Sinkovics@gmail.com

Education:

2014-present University of Vienna, Master's Degree Program in Statistics
2011-2014 University of Vienna, Bachelor's Degree Program in Statistics
2005-2010 Business High School Stegersbach, specialization on IT-
Management

Professional work experience:

03/2016-present Tutor for Mathematical Optimization, University of Vienna
09/2015 Internship in Department for Data Mining and Business Analysis,
Erste Group Bank AG
03/2015-06/2015 Tutor for Mathematical Optimization, University of Vienna
09/2014 Internship at the WIFI Vienna, Vienna Economic Chamber
08/2014 Internship in the Risk-Management Department, UNIQA Insurance
Group
08/2013 Internship in the Department of Statistics, Austrian Economic Cham-
ber

Skills and competences:

Languages: Native language German; fluent in English; basics in Croatian
IT: Expert in Windows, MS Office, VBA, R; advanced in SPSS and SAS
Other: Statistics, mathematics, driving license in classes A,B,C and E

