Institute of Technology Blanchardstown

MSc Thesis

Predictions in Financial Time Series Data

Author:
Allan Steel

Supervisor:
Dr. Geraldine Gray

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in the

School of Informatics and Engineering

June 2014

10 June,2014

Declaration of Authorship

I, Allan Steel, declare that this thesis titled, 'Predictions in Financial Time Series Data' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

INSTITUTE OF TECHNOLOGY BLANCHARDSTOWN

Abstract

School of Informatics and Engineering

Master of Science

Predictions in Financial Time Series Data

by Allan Steel

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgements and the people to thank go here, don't forget to include your project advisor…

Thank people for writing the open-source software tools.

Contents

List of Figures

List of Tables

To anyone who has never had anything dedicated to them . . .

Chapter 1

Introduction

1.1 Background

For hundreds of years speculators have tried to make an monetary profit in financial markets by predicting the future price of commodities, stocks, foreign exchange rates and more recently futures and options. Over the last few decades these efforts have increased markedly, using a variety of techniques (Hsu, 2011), which can be broadly classified into three categories:

- fundamental analysis
- technical analysis
- traditional time series forecasting

1.1.1 Fundamental Analysis

Fundamental analysis makes use of basic market information in order to predict future movements of an asset. If an investor was looking at a particular stock's fundamental data they would consider information such as revenue, profit forecasts, supply, demand and operating margins etc. Speculators looking at commodities might consider weather patterns, political aspects, government legislation and so on. Effectively fundamental analysis is concerned with macro economic and political factors that might affect the future price of a financial asset. Fundamental analysis is not considered further in this study.

1.1.2 Technical Analysis

Technical analysis is the study of historical prices and patterns with the aim of predicting future prices. Practitioners of technical analysis in the past were referred to as chartists, as they believed all that was needed to know about a particular market was contained in its pricing chart. Murphy (1999) defines technical analysis as:

"Technical analysis is the study of market action, primarily through the use of charts for the purpose of forecasting future price trends."

Technical analysis (TA) is interesting as it tends to polarise opinion as to its scientific basis and effectiveness. To many people and particularly scholars in academia it is considered little more than Black Magic. Consider the words of Malkiel (1999):

"Obviously I am biased against the chartist. This is not only a personal predilection, but a professional one as well. Technical Analysis is anathema to the academic world. We love to pick on it. Our bullying tactics are prompted by two considerations: (1) the method is patently false; and (2) it's easy to pick on. And while it may seem a bit unfair to pick on such a sorry target, just remember: it is your money we are trying to save."

However in world of finance technical analysis is ubiquitous and widely used (Menkhoff, 2010). In support of TA a plethora of so-called indicators have been developed over the years from simple moving averages to much more exotic offerings. Today every piece of software or on-line analysis tool provides the ability to place a multitude of technical indicators on a graph of stock, commodity or any financial instrument.

Most technical indicators essentially fall into one of two main categories, ones attempting to detect the start and direction of trends and those trying to identify market reversals generally called oscillators. Trend analysis indicators include Average Direction Index (ADX), Aroon, Moving Averages and Commodity Channel Indexes (CCI). Price oscillator indicators include, Moving Average Convergence Divergence (MACD - (Appel and Dobson, 2007)), Stochastics, Relative Strength Index (RSI) and the Chande Momentum Oscillator (CMO).

1.1.3 Time Series Forecasting

The study of forecasting time series data has been an active area of study for several decades (Gooijer and Hyndman, 2006). Series data is ordered such that the ordering is an important if not critical aspect of the data, with the requirement to maintain this ordering enforcing certain requirements on any processing. Series data can be ordered by factors such as distance or height but typically time is the ordering encountered. Financial data is an important category of series data and a variety of well known time series forecasting methods have been applied to the problem of predicting price movements in the financial markets. These have included, exponential smoothing, auto-regressive moving average (ARMA) and auto-regressive integrated moving average (ARIMA).

A variety of smoothing algorithms have been applied to series data in general and financial data in particular. Moving averages, including simple, weighted and exponential, are widely employed by participants in financial markets to both predict future movements and quantify current conditions. Classical time series analysis such as so-called Holt-Winters exponential smoothing, the auto-regressive moving average (ARMA or Box-Jenkins model) and auto-regressive integrated moving average (ARIMA) methods have been widely employed. In more recent years data mining techniques have been applied to the problem of financial time series prediction, for example with the use of artificial neural networks (ANNs) and support vector machines (SVM) as well as an hybrid approach of combining the classic time series techniques with the data mining methods in an attempt to leverage the strengths of each technique.

1.2 Statement of the Problem

The problem under study in this thesis is that of predicting the movement of financial markets. Financial markets include:

- Indices e.g. Dow Jones Index, FTSE100 etc.
- Commodities e.g. gold, oil etc.
- Foreign exchange rates (also known as Forex or FX) e.g. GBP USD (price of British pounds divided by US dollars).
- Stocks e.g. Google, Apple, Barclays Bank etc.

The goal of financial traders is to detect the movement of the markets and buy instruments expected to rise in price "going long" and sell those predicted to fall in price "going short". The markets are a neutral sum process, for every participant who gains there are those who lose.

1.3 Purpose of Study

The purpose of this study is to investigate and establish the usefulness and accuracy of a selection of technical indicators and time series analysis on the ability to predict future data movements in a group of financial markets including national indices, Forex, commodities and stocks.

1.3.1 Study Objectives

The objective of this study is three fold:

1. Determine if a group of popular and widely used technical indicators can be used to predict the direction of movement in a range of financial markets.
2. Investigate if traditional time series models can predict the direction of movement in a range of financial markets.
3. Use traditional time series models to identify when a financial market moves into the "trending" phase.

1.4 Research Questions or Hypothesis

The hypothesis of the study is that the use of technical indicators or time series analysis can help to predict the future direction and movement of financial markets.

1.5 Methodology

1. Review current research in the field.
2. Collect data, primarily from freely available sources on the internet such as Yahoo and Google.
3. Pre-process the data and perform initial data investigations and analysis.
4. Establish "base line" systems based on initial analysis.
5. Apply Technical Indicators to these "base line" systems to determine if they have a role to play in predicting the movement of a particular financial market.
6. Apply traditional times series modelling methods to evaluate their suitability in predicting future price movements of financial market.

1.6 Limitations of the Study

Limitations in this study include:

1. Choice of Technical Indicators - a small selection of the huge number available was selected. The selected group represent widely used examples and are drawn from the various categories available.

2. Availability of financial data - Daily data in the format of open, high, low and close prices (OHLC) is readily and freely available and is thus used in this study. Data in time frames other than daily are generally only commercially available and beyond the resources of this study.

3. Forex data - Frequently Forex is provided as a single daily value as these markets are traded all 24 hours of the day. This may have impacts on the suitability of this data for various algorithms used in this study.

1.7 Scope of the Study

There are a huge choice of financial data sets from which to choose and likewise many dozens of technical indicators. This study will employ daily data from major national indices such as the German Dax, US Dow and Japanese Nikkei. Commodity data will cover gold and US Crude Oil and forex will include GBP/USD, EUR/USD, EUR/GBP, USD/JPN exchange pairs. Technical Indicators used will include examples from each of the primary categories trend detection and market reversal oscillators.

1.8 Structure of Project

Chapter 2 is a literature review and introduction to time series analysis and financial market trading with systems and technical indicators. The classical time series methods of Holt-Winters exponential smoothing, auto-regressive moving average (ARMA or Box-Jenkins model) and auto-regressive integrated moving average (ARIMA) are introduced and explained. Their adoption and use in predicting financial markets is discussed.

Chapter 3 introduces the methodology used in this study. It includes a description of the data sets employed, software and programming languages levered and the general methodology and approach taken.

Chapter 4 details the implementation and experimentation.

Chapter 5 is an analysis of the results generated and conclusions.

Appendix A

Chapter 2

Literature Review

Speculators, stock market traders, simply traders or market participants are all terms used to describe individuals and organisations who attempt to make a living from buying and selling various financial assets in a huge range of markets around the world. Clearly the ability to forecast the direction of market movements, up or down, is vital to these individuals and entities. To this end a wide variety of techniques and methods have been tried and used by the participants in the market. Further, over the last few decades academics have shown an interest in this field and attempted to quantify and justify the wide variety of techniques used.

Two areas where traders and academics have looked for help in predicting future market direction is time series forecasting, and the use of technical indicators. This chapter is divided into two these general categories, time series modelling and the use of technical indicators.

2.1 Technical Analysis

2.1.1 Trading Systems

A wide variety of techniques have been employed by financial market traders in their attempts to make profits with the term "trading system" being applied generally to the methodology used. Often trading systems are "mechanical" in nature in that traders use a distinct set of rules in order to guide them as when to enter a trade, when to exit and so on. Faith (2007), one of the original and now famous "Turtle Traders" provides an excellent overview of mechanical trading systems (and how they were to become known as the "Turtles"). Richard Dennis and William Eckhardt the sponsors and mentors of the Turtles were trying to settle a debate on whether individuals simply have a natural talent which enables them to become successful traders or if they could be taught using a mechanical trading system. Dennis who believed they could be developed, coined a phrase along the lines of "growing traders like turtles" as they had just visited a turtle farm in Singapore.

Weissman (2005) makes the point that there are several aspects to a trading system. Firstly there are entry and exit signals, these are market events that trigger a speculator to enter into the market and either buy or sell a particular asset. These signals are typically events such as a fast moving average crossing a slower one, the market hitting a certain price or the occurrence of a particular chart pattern (see section 2.1.5). Other elements of a trading system include position sizing rules and money management strategies such that returns are significant, losses are minimised and the entire risk profile is controlled.

Many traders erroneously mistake entry and exit signals as being a full trading system in themselves whereas in actuality they are merely components of a system (Beau and Lucas, 1999). Likewise most, if not all, papers published by academia focus on entry and exit signals alone, which is probably a result of several factors. Firstly, entry and exit signals are important components in trading systems and are a good place to start in system development. Additionally the other aspects of a system are not as well known and their importance is often ignored (Kaufman, 2013). Finally, testing an "entire" system as defined here is far more difficult and time consuming than considering entry and exit signals alone and often it is not practical to extend a study to include a full system. In summary there is value in considering entry and exit signal in isolation but one has to remember it is not the whole story.

Attempting to forecast stock market prices is a complex and challenging endeavour, yet one that is widely encountered. There is a large body of research published in this area which has been reviewed by Atsalakis and Valavanis (2009). Work usually focuses on either individual stocks or more commonly stock indices. Stock indices are the sum movements of many individual equities and therefore reflect the movement of the market as a whole as opposed to any one stock. Many stock market indices have been investigated including those belonging to well-developed countries such as those in Western Europe, North America etc. as well as developing markets such as those in eastern Europe.

In trying to predict stock market movements a variety of input variables have been used. Frequently, the so-called OHLC (open, high, low and closing prices) are used as inputs along with a variety of technical indicators. In addition many authors have used a combination of markets, for example Huang et al. (2005) use both the USD/YEN exchange rate and the S&P 500 to build a prediction model for the Japanese NIKKEI index. A variety of predictive methodologies have been reported in the literature including linear and multi-linear regression, ARMA and ARIMA models, genetic algorithms (GAs), artificial neural networks (ANNs), random walk (RW) and the so-called buy and hold (B & H) strategy.

A variety of performance measures have been reported including both non-statistical and statistical methods. Non-statistical performance measures used include annual return and annual profit of a particular model as well as the hit rate or the number of times a model correctly predicts whether a market will go up or down. Alternatively a variety of statistical measures have also been employed and prominent amongst them are, mean absolute error (MAE), root mean squared (RMSE), mean squared prediction error (MSPE), correlation coefficient and autocorrelation squared correlation and Akaike's minimum final prediction error (FPE).

Two well studied and used methodologies in stock trading are the moving average system and range breakout system reported by (Brock et al., 1992) in one of the very earliest papers published covering technical analysis. In a moving average system (see section 4.3.1) the speculator buys a market when its price is above the moving average and sells in the reverse situation. A large number of variations on this theme can be found, with the use of two moving averages being popular. When using two averages there is normally a "fast" one, usually of the order of 10 to 25 days, and a "slow" one in the 50 to 250 day range. In these circumstances a buy is usually triggered when the fast average crosses above the slower average. The theory is that the moving averages follow the trends in the market and thus allow the market participant to trade in the direction of the trend, which is an advantageous situation for the trader.

A second popular idea is that of breaking out of a range. Often financial markets trade between a range of values in a particular time period, essentially markets are either trending (up or down) or not trending at all but moving within a defined range. While moving in a range the lower price boundary is referred to as support and the upper one as resistance. In a breakout system the analyst buys a market when it moves beyond these resistance levels or sells when it breaks below the support. Brock et al. (1992) analysed both these two ideas and found merit in them. Using daily data from the Dow Jones industrial index they found that these strategies provided better results than those generated with random walk, AR and GARCH models.

2.1.2 Technical Analysis Overview

Technical analysis is the technique of looking at the past history of a financial market, identifying patterns and trends and utilising the information in predicting future price movements (Bulkowski, 2011). A technical indicator is a method used to identify a particular pattern, and there have been a large number developed over the years to predict situations such as the start of a trend or a reversal in price movement. A wide range of papers on technical analysis (TA) indicators and methods can be found in the literature. Likewise technical analysis is prominent in many best selling books including Market Wizards (Schwager, 1988), New Market Wizards (Schwager, 1994) and Covel's Trend Following (Covel, 2009). In the following sections various technical indicators are introduced and their use in predicting market movements are explored. Firstly, the question of whether technical analysis even works is addressed. Although technical analysis is widely used in the market place there is a question mark over the entire concept behind it and many people, especially academics, are highly sceptical about the validity of the entire approach.

2.1.3 Does Technical Analysis Work?

Friesen et al. (2009) have examined various price "patterns" used by traders in their systems such as "head-and-shoulders" and "double-top" patterns. The authors note that although a wide array of patterns have been identified and documented there lacks any convincing explanations for the formation of these patterns and how they can lead to profitable trading systems. The authors report that several studies based on the US equity market have identified distinct behaviours, namely the tendency for short-term momentum over 1 year to 6 months (De Bondt, 1985, Chopra et al., 1992, Jegadeesh and Titman, 1993), longer term mean reversion and finally price reversals over the one to four week period (Jegadeesh, 1990, Lehmann, 1990, Jegadeesh and Titman, 1995, Gutierrez Jr, 2008). These observations lend support to the success of trading systems that purport to detect and follow trends in the market (Sweeney, 1986, Levich and Thomas, 1993, Neely et al., 1997, Dueker and Neely, 2007).

The authors present a model that can explain the profitability of selected trading rules that utilise past chart patterns. One important aspect of this model is the inclusion of confirmation bias, which shows up in a wide range of decision making processes. Their model displays negative autocorrelations over the very short term, positive ones in the mid term and become negative again over the longer horizon, reflecting the documented empirical properties of US stock prices. It is suggested that traders take market positions affected by their original biased view which leads to autocorrelations and price movement patterns resulting in the previously described market behaviour.

Shynkevich (2012) investigated the power of a large selection of technical trading rules to yield profits when applied a selection of small cap and technology portfolios (US stocks) between 1995 and 2010. The author chose technical indicators from four general categories:

1. standard filter rules - for example a buy is generated when prices increase from a previous low. Such a low may be defined as the lowest closing price in a particular period. In more recent years this technique has been replaced by moving averages.
2. moving averages - signals generated when short MA cross long MA.
3. support and resistance trading strategy (SR) - a buy is initiative when prices rise above a local maximum, and vice versa for a local minimum price.
4. Channel breakout - related to SR, a buy/sell is triggered when a price moves outside a channel generated from highs and lows of a certain period.

The author applied a variety of parameters in each model resulting in a total of 12937 models being tested. It was reported that TA produced positive results in the first half of the time period tested, but not in the latter half. In the second half of the time period studied TA provided inferior performance than a buy-and-hold approach, i.e. a trader simply buys a particular asset and waits. The author concludes these differences in performance are due to equity markets having become more efficient in recent years which has reduced the short term predictive powers of TA.

The use of technical analysis in the finance community was studied by Menkhoff (2010) who looked into its use by professional fund managers. This study is note worthy as it used data from experienced and educated market professionals and not a wider cross-section of traders. With the advent of the internet and the explosive growth in on-line financial charting and trading sites, financial trading became accessible to the general public, resulting in huge numbers of amateur traders entering the market. All of the web sites that cater for this segment of traders offer a huge number of technical analysis indicators built into their respective charting packages and even a rudimentary visit to any of the discussion forums will demonstrate the popularity and wide spread use of technical analysis.

The author surveyed 692 fund managers in several countries, with funds of various sizes under management. The vast majority of these fund managers reported using technical analysis to some degree and particular faith was put in TA for predicting price movements in the short term of up to a few weeks, beyond which focus shifts to fundamental analysis. Further, the workers found that smaller asset manager firms make greater use of TA, possibly because deriving the information for fundamental analysis is beyond their resources. Finally, most respondents to the survey believe that human psychology is the reason TA works. In particular they suggest psychological biases in the market participants are the root cause of market trends and that TA is able to identify and follow them.

2.1.4 Moving Average Indicators

A study of moving average convergence divergence (MACD) is reported by Ulku and Prodan (2013). MACD is a technique which attempts to detect the early stage of a trend as it forms, and is widely used by market participants. It is described in more detail in Appendix B. Ulku and Prodan (2013) apply MACD to a wide range of national stock market indices comprising developed as well as emerging markets. The authors compare the MACD signals against entry signals generated from simple break out systems (described previously). The comparison systems would generate a buy signal if the price moved higher than a moving average (MA), set at either 22, 56 and 200 days. The MACD and the comparison system using 22 day moving averages are classified as short horizon signals, while the break out of the 56 and 200 day MA are considered long horizon signals. The workers reported that the MACD indicators provide for profitable returns on 23 of 30 national indices, but that the 22 day MA performs better being positive in 27 of the 30 markets.

2.1.5 Candlesticks Patterns

Probably the oldest form of technical analysis in use today is the so-called candlestick analysis, so named because daily open and close prices are plotted such that they resemble candlesticks (Morris, 2006). Figure 2.1 is an example of daily prices being plotted as a candlestick, this plotting method is today ubiquitous in trading software. Typically the colour in which the candlestick is plotted indicates whether the price went up or down over the course of the day. Many charts that are plotted in colour use green to represent days that close up and red for days that close down. The main body of the candlestick represents the movement from open to close, and the protruding lines mark the high and low of the day.

Technical analysis via candlesticks is reputed to have been developed by Munelusa Homma, a legendary trader of rice in Osaka, Japan who made a fortune analysing rice prices with candlesticks in the seventeenth century (Nison, 2001). Candlestick patterns with supposed predictive qualities can be derived from a single day or from considering a few days, usually 2 or 3, together (Bigalow, 2011). There are a huge number of patterns recorded in the literature and usually assigned exotic names such as “White Marubozu”,

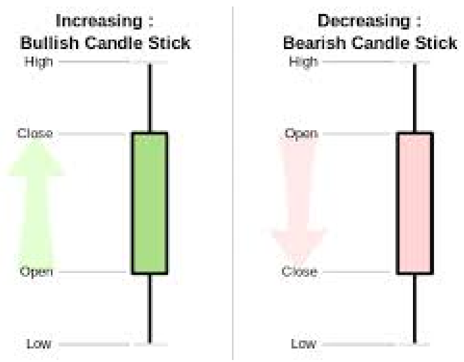

FIGURE 2.1: Candlestick representation of daily open and close prices. Days when markets close higher than opening prices (up days) are distinguished from days that close down by use of either different colour or alternatively being filled or not filled.

”Black Shooting Star” and ”Hanging Man”. Examples of such named patterns can be seen in Figure 2.2.

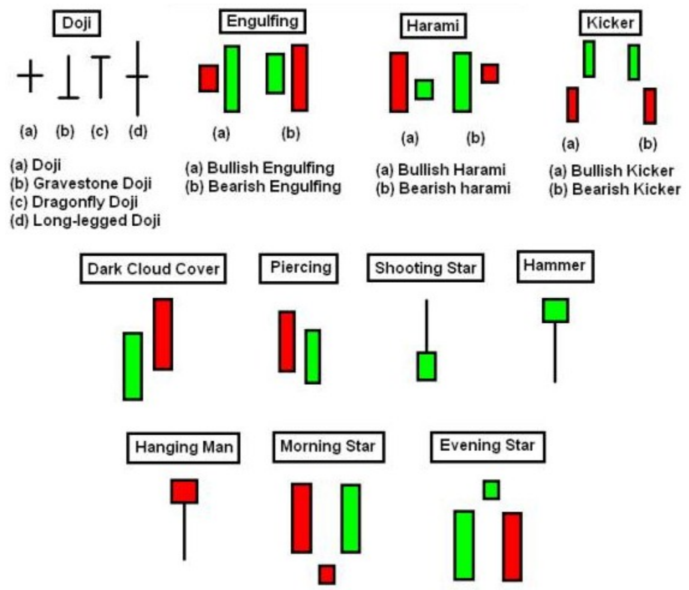

FIGURE 2.2: Examples of well known patterns encountered in candlestick analysis together with the their commonly accepted predicted powers.

Candlestick patterns are essentially visualisation tools providing an easy to comprehend view of the market movements in a particular day. However there is some vital information which is not conveyed in a candlestick. In particular the order of events isn’t displayed. Figure 2.3 shows how two days can produce the same candlestick but in actuality the price movements and volatility in them was very different. Depending upon the type of trading system being employed this could have important effects.

As always with technical analysis there is doubt as to the validity of the methods despite its almost universal employment. An in depth study of the predictive power of a range of candlestick patterns on stock prices between 1992 and 2002 from the Dow Jones

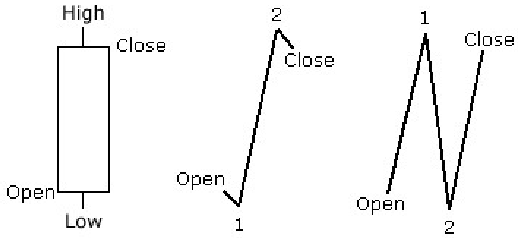

FIGURE 2.3: Candlesticks don't provide information regarding the order of price movements. Both these daily price movements would be represented with the same candlestick pattern.

Industrial Average (DJIA) was carried out by (Marshall et al., 2006) in which doubt was cast on the validity of candlestick patterns to predict market movements. The workers used a range of bullish (signals that indicate a trader should buy) and bearish (signals that indicate a trader should sell) candlestick patterns to initiate trades on the various stocks. Trades were held for ten days as it was assumed that these patterns reflect short terms trends and thus have a predictive power in a similar time frame. In order to quantify the results generated from the use candlestick patterns they were compared to results observed from four alternative null models. Simulated stock data was generated using a bootstrapping methodology (Efron, 1979) and then four null models were applied to the data, random walk, an autoregressive process of order one (AR(1)), a GARCH in-Mean (GARCH-M) model and an Exponential GARCH (EGARCH) model.

From the comparison of the results generated from the candlestick patterns and the four null models the workers concluded that the variety of candlestick patterns tested had no predictive power on the stocks at all. The returns from making buying and selling decisions based on candlestick patterns didn't outperform the null models on the simulated data. As always one has to be slightly careful with results of this nature as the trading period was fixed at ten days, in other words the candlestick patterns were used as an entry signal for the trade but there wasn't an exit signal. Further in reality use of candlesticks analysis would be incorporated into a trading system, which typically consists entry and exit signal, position sizing rules and money management strategies (Faith, 2007).

2.1.6 Trend Reversal Oscillators

Tanaka-Yamawaki and Tokuoka (2007) reported the use of several technical analysis techniques in the successful prediction of price movements in eight stocks found on the New York Stock Exchange (NYSE) by analysing tick data. The predictions were in the very short term as tick data is the most granular level reported in financial data. The workers used ten technical analysis indicators from three broad classes, namely trend indicators, oscillators to find market reversals and momentum indicators to measure the strength of the market. Combinations of indicators, typically from the different categories are usually combined by market participants into a variety of systems. In this study the ten indicators can form a possible 1023 combinations. A genetic algorithm was used to determine the best combination of indicators for each stock, resulting in a customised combination for each. Using each stock's indicators, the next ten ticks of data were modelled with very high accuracy, with predictions for IBM's stock being the best at a very impressive up to 82%.

2.2 Time Series

The study of forecasting time series data has been an active area of study for several decades and an overview of work over 25 years has been documented by Gooijer and Hyndman (2006). Series data is ordered such that the ordering is an important if not critical aspect of the data and the requirement to maintain this ordering enforces certain requirements on its processing. Series data can be ordered by factors such as distance or height but typically time is the ordering encountered, and thus such collections are referred to as time series. Analysis of time series data is found in a wide range of areas including, Sales Forecasting, Speech Recognition, Economic Forecasting, Stock Market Analysis, Process and Quality Control and Seismic Recordings.

In general with non-series data we are interested in the relationships between the attributes of any particular row of data and perhaps how they affect the parameter we are interested in. Frequently some kind of regression technique is used in this kind of analysis in order to answer questions such as how is rainfall in an area affected by altitude or how does fuel consumption vary with car engine size (Han et al., 2011).

However with time series data there is an additional consideration, the relationship between the attribute's current value to that of its previous or later values. This is known as auto-correlation (Mills, 2011) and more details can be seen in section 2.2.2.1. Typically with financial data we are interested in previous values, in other words how is today's closing price affected by the closing prices one, two or three days ago?

As illustrated in Figure 2.4 a time series can contain some or all of the following components:

1. Trend - the overall direction of the series, is it increasing or decreasing over time?

2. Seasonality - regular variations in the time series that is caused by re-occurring events, for example a spike in sales during the Christmas period (So and Chung, 2014).

3. Random component - additional fluctuations in the series that may be attributed to noise or other random events.

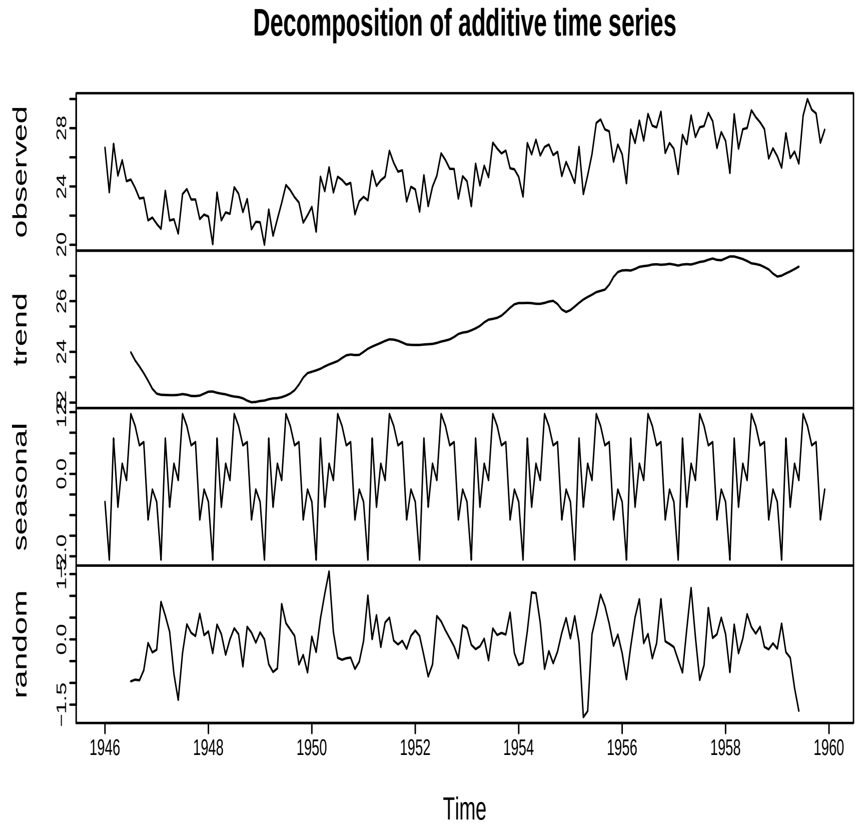

FIGURE 2.4: A time series decomposed into its three primary components (Coghlan).

There are three primary types of time series, stationary, additive and multiplicative. Stationary series have constant amplitude without a trend element and an example can be seen in Figure 2.5. Often stationary time series are repetitive, in other words showing constant auto-correlation and are considered the easiest type to model. A stationary time series can be composed of a seasonal element and/or a random component, thus:

$$\textit{stationary time series} = \textit{seasonality} + /\textit{or noise}$$

The second type of time series is the additive type. In this type all three components of the series are present, trend, seasonality and noise. The distinguishing feature here is the amplitude of the seasonal component in that it is quite regular being static over time.

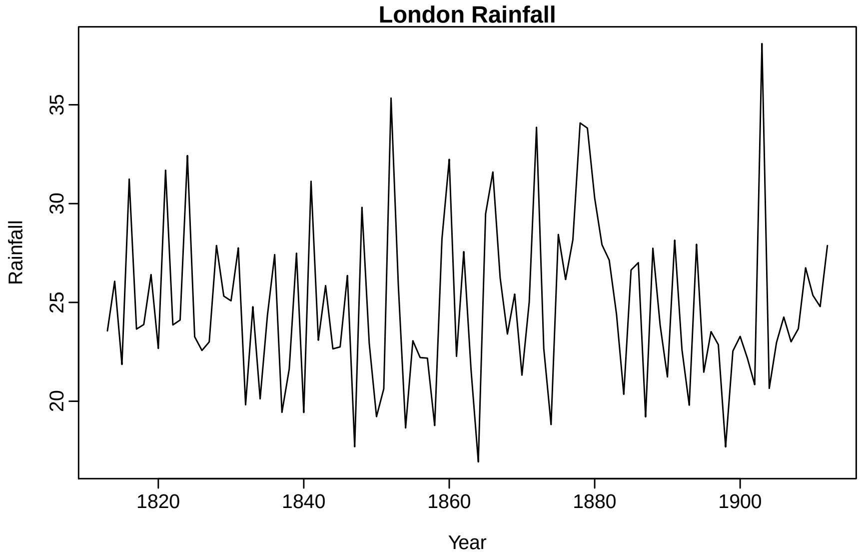

FIGURE 2.5: Example of a stationary time series which can be made up from noise and/or a seasonal component.

An example of an additive series can be seen in Figure 2.6. This time series is trending upwards overall but there is a clear repetitive pattern of peaks and troughs caused by the seasonality, with the heights of the peaks all being similar. We can consider an additive time series as:

$$\textit{additive time series} = \textit{trend} + \textit{seasonality} + \textit{noise}$$

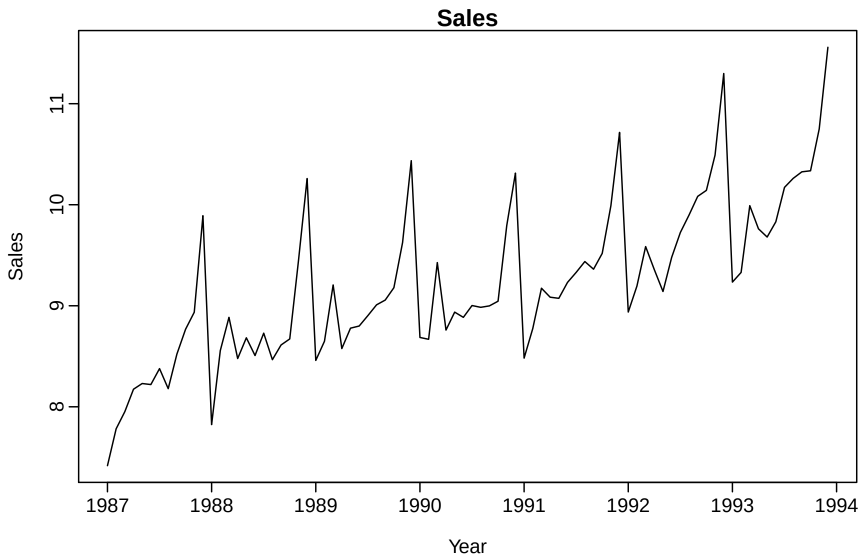

FIGURE 2.6: Example of an additive time series which results from all three components trend, noise and seasonality.

The third type of time series, as seen in Figure 2.7 is multiplicative. This is similar to the additive version except the amplitude of the seasonality increases over time. It can be considered as:

$$multiplicative\ time\ series = trend * seasonality * noise$$

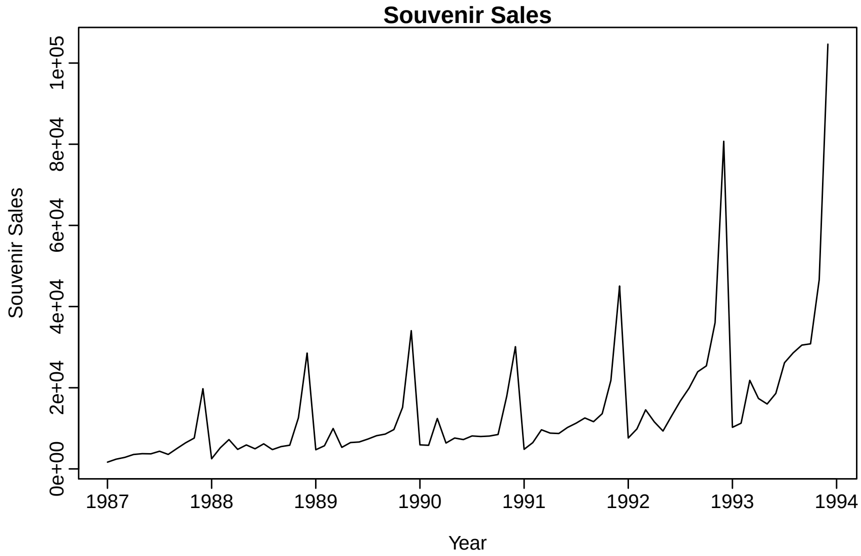

FIGURE 2.7: Example of a multiplicative time series resulting from the effects of trend, noise and seasonality.

Financial time series can be considered as containing all three elements of a time series. They can show properties of a stationary time series when they are range bound and only move between two values. At other times, markets trend strongly consistently, making new highs or lows and exhibit properties of an additive and occasionally a multiplicative series.

2.2.1 Time Series Smoothing

Smoothing is an important and widely adopted method to predict financial markets. Recent work on smoothing time series data has its origins in Brown (1959), Brown (1963), Holt (2004) and Winters (1960). Typically, the various smoothing techniques encountered are based around the concept of moving averages. This section will introduce a variety of smoothing methods commonly encountered in forecasting financial data.

2.2.1.1 Simple Moving Average (SMA)

A simple moving average is calculated from the value itself and its neighbours, which can be ahead or behind in the series. In this study values behind the current value are considered. The number of previous values included is often referred to as the “window”, so if one was to consider the current value and four previous ones this would be considered a simple moving average of lag 5 (SMA5). An example of a simple moving average can be seen in Table 2.1, where a SMA5 of the closing price has been added.

TABLE 2.1: Example of a simple moving average of the closing price with a lag of 5 periods.

Date	Open	High	Low	Close	SMA5
02/01/14	9598	9621	9394	9400	NA
03/01/14	9410	9453	9368	9435	NA
06/01/14	9419	9469	9400	9428	NA
07/01/14	9446	9519	9417	9506	NA
08/01/14	9513	9516	9468	9498	9453
09/01/14	9492	9550	9403	9422	9458
10/01/14	9474	9530	9441	9473	9465
13/01/14	9498	9519	9457	9510	9482

2.2.1.2 Weighted Moving Average (WMA)

A simple moving average assigns equal importance to all data points being averaged, however if this is considered unsuitable a higher weighting can be applied to certain data points elevating their importance in the average and thus generating a weighted moving average (Devcic, 2010). Typically the more recent data points in a time series would be given higher importance. One common version of a WMA is to decrease the weighting by one for each period in the average. The formula for calculating a weighted moving average is:

$$((n * P_n) + (n - 1 * P_{n-1}) + ...(n - (n-1) * P_{n-(n-1)})) \div (n + (n-1) + ...n - (n-1))$$

where:
n = the number of periods used in calculating the moving average
Pn = the price of the most recent period used to calculate the moving average

An extra column has been added to the data in Table 2.1 which contains the WMA for the last five close values. The current value was multiplied by 5, the previous one by 4, the previous one to that by 3 and so on. These five values were added together and divided by 5+4+3+2+1 to generate the WMA as shown in Table 2.2.

2.2.1.3 Exponential Moving Average (EMA)

An exponential moving average (EMA) is an extension of the weighted moving average (Ord, 2004). In comparison to the simple moving average, greater emphasis is given to the most recent data points and the resulting averaged values are closer to the actual

Table 2.2: Example of a weighted moving average.

Date	Open	High	Low	Close	SMA5	WMA5
02/01/14	9598	9621	9394	9400	NA	NA
03/01/14	9410	9453	9368	9435	NA	NA
06/01/14	9419	9469	9400	9428	NA	NA
07/01/14	9446	9519	9417	9506	NA	NA
08/01/14	9513	9516	9468	9498	9453	9471
09/01/14	9492	9550	9403	9422	9458	9461
10/01/14	9474	9530	9441	9473	9465	9466
13/01/14	9498	9519	9457	9510	9482	9481

observations of the data set. Weighting factors decay exponentially resulting in the emphasis falling on the recent values though not discarding the older ones totally. See Appendix B for full details.

2.2.1.4 Moving Averages in Practical Use

Moving averages are widely used in the financial world to predict the start of trends which is important as trends are considered the best opportunity to make profits from the markets. By their nature moving averages are lagged indicators in that they reflect market action from the past (recent or distant depending on the lag variable) and this can be considered a drawback. The lag period offers a trade off in terms of prediction. If the lag is short and/or weighting is applied the average is affected strongly by recent prices and trends can be detected in the early stages and trading profits can be enhanced. However when the average is close to the current price they have a tendency to generate "false signals" (see section 2.1.1 for an explanation of entry and exit signals), in other words prices may start to rise (or fall) but they are not actually in a trend, it is just the natural wax and wane of the market, and traders are said to be "whipsawed". When the lag variable is long a different problem is encountered. For example, if a price moves above a long moving average the indicated trend is usually genuine, however by the time this is reflected in the average a lot of the trend has developed and the trader has lost a lot of potential profits. Thus there are pros and cons associated with using the different types of moving average.

2.2.1.5 Holt-Winters Smoothing Models

The exponential smoothing of a time series containing noise, trend and seasonality was developed by Winters (1960) who as a student of Holt, built upon his previous work, and is today called the Holt-Winters method. This method defines three parameters alpha, beta and gamma which define the degree of smoothing to be applied to the three components of the time series. Firstly, a value of alpha is used to dictate the amount of smoothing to apply, with high smoothing factors placing more emphasis on recent data points at the expense of those further away. In a data set with trend this simple exponential moving average doesn't perform well and a second order of smoothing is needed, so called "double exponential smoothing". The parameter beta in Holt-Winters defines this second order smoothing. Finally if a seasonal component is also present in the data set a third level of smoothing is introduced making the process a triple exponential smoothing. It is this third level of smoothing that the parameter gamma refers to. Depending upon the nature of the time series one, two or all three of the parameters may be defined in the Holt-Winters methodology.

If researching a time series with no seasonality or trend use of the Holt-Winters model with the beta and gamma parameters set to false, in other words not used, is appropriate. Figure 2.8 shows the addition of an exponential smoothing line to the stationary data set introduced in Figure 2.5.

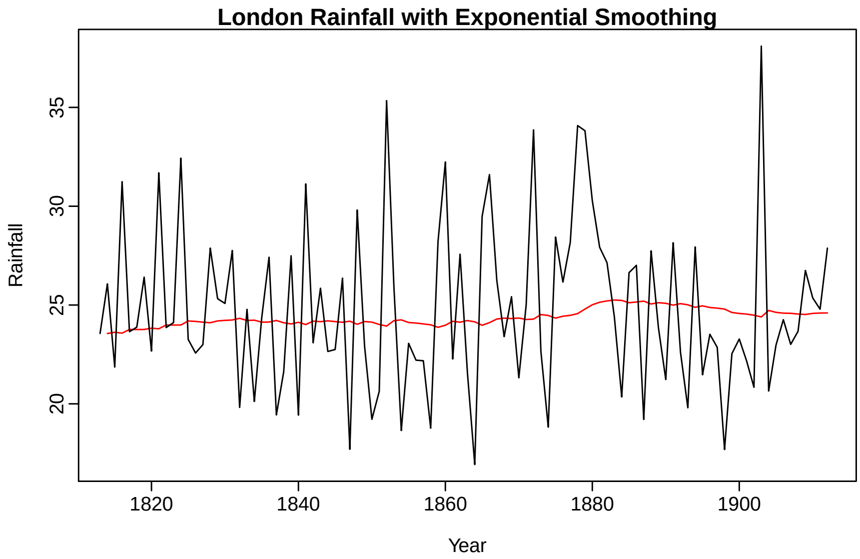

FIGURE 2.8: A time series with no seasonality or trend, showing the fitted line generated from Holt-Winters exponential smoothing with the beta and gamma parameters set to false.

If the time series is additive with a trend but without seasonality the use of Holt-Winters with values used for alpha and beta but with the gamma parameter set to false is appropriate. Such a time series can be seen in Figure 2.9 with the exponential smoothing. Finally if the time series contains all three components a smoothing line can be fitted using Holt-Winters exponential smoothing in which there are values for all three terms alpha, beta and gamma. Figure 2.10 is an example of a time series with both trend and seasonality and overlaid with Holt-Winters smoothing generated by using values for all three terms in the smoothing algorithm.

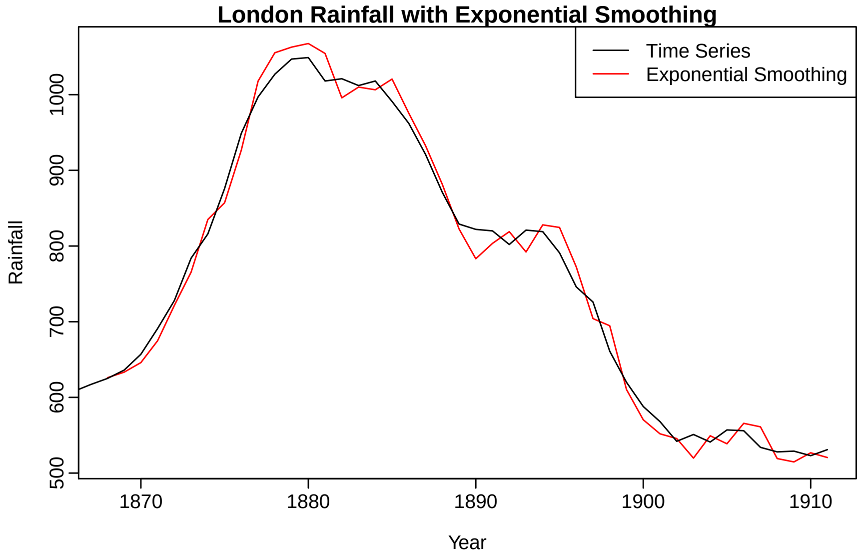

FIGURE 2.9: A time series with trend though no seasonality, showing the fitted Holt-Winters exponential smoothing with the gamma parameter set to false.

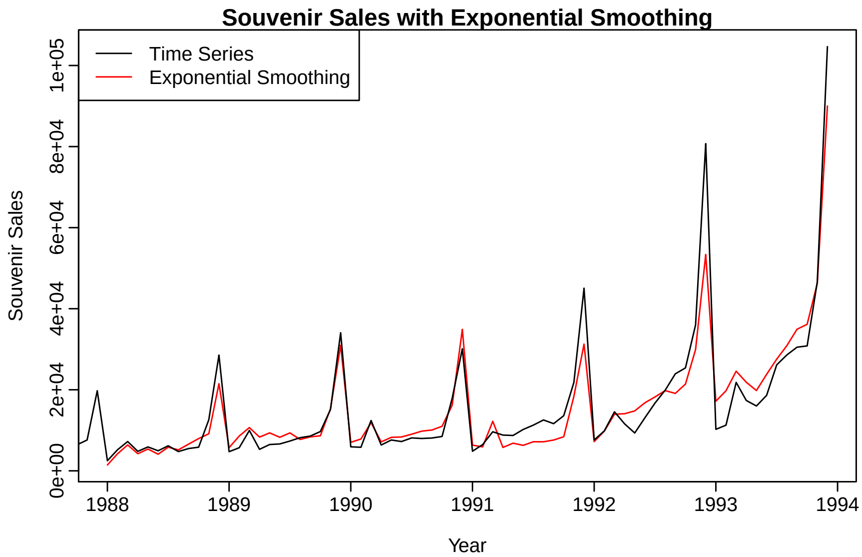

FIGURE 2.10: A time series with trend and seasonality, showing the fitted Holt-Winters exponential smoothing.

2.2.2 Auto-Regression Family of Models

2.2.2.1 Auto-Regression

Regression is the study of the impact of known variables (independent) on an unknown (dependent) variable and addresses questions such as how does a persons income vary with their years of education. The general equation for linear regression is given by:

$$y = a + bx + \varepsilon$$

where:
a is the intercept.
b is the co-efficient.
x is the independent variable.
ε is the error term.

In reality there are often a large number of independent variables that affect the unknown under study and thus multiple regression, shown below, is usually of interest.

$$y_1 = a + b_1x_{1i} + b_2x_{2i} + ... + b_nx_{ni} + \varepsilon$$

In a time series the preceding values often have a bearing on the current data point, and this is especially important in financial time series data. Thus auto-regression is the prediction of the current point from the use of previous values of the data point itself, and is given by:

$$t_t = c + b_1 * r_{t-1} + b_2 * r_{t-2}...b_p * r_{t-p} + \varepsilon$$

where:
c is the intercept, is often zero and the mean of the time series.
$b_1 - b_p$ are the independent variables, the previous values.
ε is random noise.

2.2.3 Auto-Regressive Moving Average (ARMA)

The auto-regressive moving average (ARMA)model, also known as Box-Jenkins (Box and Jenkins, 1970), combines moving averages with auto-regression. A model that uses moving averages to predict current values is given by:

$$-r_t = c + a_1 * ma_{t-1} + a_2 * ma_{t-2}...a_q * ma_{t-q} + err_t$$

ARMA combines the moving average model with auto-regressive terms to generate:

$$\begin{aligned} r(t) = c+ & \\ & b_1 * r_{t-1} + b_2 * r_{t-2}...b_p * r_{t-p}+ \\ & a_1 * ma_{t-1} + a_2 * ma_{t-2}...a_q * ma_{t-q} \\ & + err \end{aligned}$$

where:
c is the intercept, which is often zero and the mean of the time series.
$b_1 - b_p$ are the independent variables, the previous values in the auto-regression term.
$a_1 - a_p$ are parameters of the moving average model.
ε is random noise.

An ARMA(1,1) model uses the previous value in the auto-regression term and the previous value's moving average. Thus in general terms an ARMA(p,q) model uses the previous p values in the auto-regression term and the moving averages derived from the last q values. There are therefore three steps in in developing an ARMA model:

1. identification step in which the order of AR and MA components is determined
2. parameter estimation
3. forecasting

ARMA models have certain intrinsic properties that may be considered drawbacks, namely the requirement for the time series to be stationary with no trend and also linear and the difficulty in deriving the correct parameters to use in the model. In order to overcome these restrictions researchers have tried a number of approaches to enhance the effectiveness of ARMA models.

The problem of model and parameter selection in ARMA models has also been addressed by Rojas et al. (2008). The authors make the point that in traditional research choosing the correct model is time consuming and requires a large degree of expertise. In order to circumvent these issues they propose an automatic model selection method to speed up the process, remove the need for expert intervention and allow the processing of a large number of time series. In a similar study Qian and Zhao (2007) investigate how to determine model selection where there are potentially millions of candidate ARMA models available for the time series. Again, the authors propose an automatic selection algorithm centred on the Gibbs sampler. The proposed method allows for various problems typically encountered in selecting ARMA models and the resulting choice was used to generate a prediction of China's Consumer Price Index (CPI).

2.2.4 Auto-Regressive Integrated Moving Average (ARIMA)

One limitation with the ARMA model and indeed other approaches is that it is assumed that the time series is stationary, it doesn't have trend and has constant variance and mean (Shumway and Stoffer, 2010). In reality of course many time series data sets have trend, and in the world of financial data this is also true. In order to account for trend in a time series it is often transformed into a stationary data set, modelling is then performed on this adapted data after which it is returned to its original state. In effect the trend aspect is removed, modelling is done, then the trend component is added back into the data.

One such method for removing trend is differencing (Mills, 2011). Differencing is the technique of replacing the actual values of the observations with the values of the differences between them. This is represented as:

$$Diff1_t = r_t - r_{t-1}$$

Differencing is the same as calculating the derivative of the series, thus a time series that has under gone differencing is considered “integrated”. If taking this so-called first difference doesn’t remove the trend one can go further and use the second difference:

$$Diff2_t = (r_t - r_{t-1}) - (r_{t-1} - r_{t-2})$$

Addition of an integration step to the ARMA model results in an auto-regressive integrated moving average (ARIMA) model, with the general formula:

$$\begin{aligned} r(t) = c+ \\ & b_1 * r_{t-1} + b_2 * r_{t-2} ... b_p * r_{t-p} + \\ & a_1 * ma_{t-1} + a_2 * ma_{t-2} ... a_q * ma_{t-q} \\ & d_1 * diff_{t-1} + d_2 * diff_{t-2} ... d_d * diff_{t-d} \\ & + err \end{aligned}$$

where:

c is the intercept, which is often zero and the mean of the time series.

$b_1 - b_p$ are the independent variables, the previous values in the auto-regression term.

$a_1 - a_p$ are parameters of the moving average model.

$d_1 -_p$ are the parameters of the differencing term. ε is random noise.

ARIMA models are usually referenced as ARIMA(p,d,q) with p the number of terms used in the auto-regression, d the number of differencing terms and q the number of terms used in the moving average. A summary of which model (Holt-Winters, ARMA or ARIMA) to use with which type of time series can be seen in Table 2.3.

Table 2.3: Appropriate models for use with time series data.

Model	**Time Series Required**	**Assumes Correlation**	**Trend**	**Seasonality**
Holt-Winters	Short Term	N	Y	Y
ARMA	Stationary	Y	N	Y
ARIMA	Non-stationary: Additive or Multiplicative	Y	Y	Y

2.2.5 ARIMA Parameter Selection

An important aspect of building time series models with ARIMA techniques is the choice of parameters to use. Auto-correlation (AC) and partial auto-correlation (PAC) are important measures in the selection process of these parameters (Mills, 2011).

Correlation is the measure of how one variable changes with a second one. For example if variable A increases while variable B increases they are positively correlated and conversely they are negatively correlated when one decreases as the other increases. Further, correlations are measured by degree on a scale of 1 to -1, with 1 being perfectly correlated. A value of 1 indicates that the two variables increase together perfectly in sync, whereas a value of -1 suggests that as one variable increases the other decreases by the same amount. Finally a value of 0 is indicative of no correlation at all between the two variables.

Auto-correlation is the correlation between an attributes value now and the same attribute's value in the past or future (Shumway and Stoffer, 2010). Typically with financial data we are interested in the correlation with values in the past. The interval between the value of interest and the previous observation used in determining the correlation is known as the lag. Thus the correlation between the current observation and the previous one may be of interest, and this is a lag of +1, while a value five time intervals previous is +5. Non-intuitively positive values for lags refer to the past while negative values are in the future.

A correlogram is a matrix plot of auto-correlations over a series of time lags. Correlograms are used in checking data for randomness and in the model identification stage of the ARMA methodology (see section 2.2.3). Data is considered random if the auto-correlation value is close to zero. In general a data set's randomness needs to be checked in order to confirm the validity of many statistical tests. Thus a correlogram helps to determine if data is random or if an observation is related to an earlier one, thereby helping in the determination of an appropriate ARMA model.

Figure 2.11 is the correlogram of a data set built-in to R, called AirPassenger, over a range of lags from 1 to 80 and Figure 2.12 is the correlogram of the seasonal component of this data set.

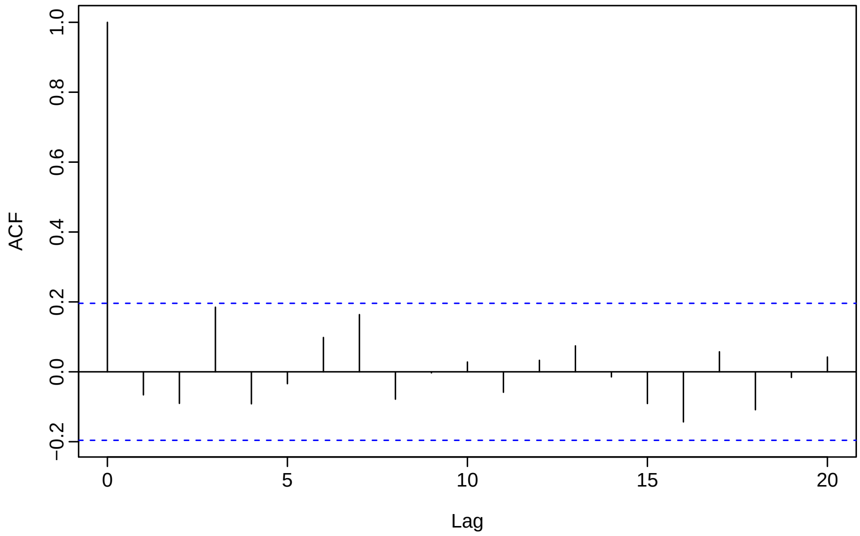

FIGURE 2.11: Correlogram of AirPassenger data a built-in data set of R, over a range of time lags from 1 to 80.

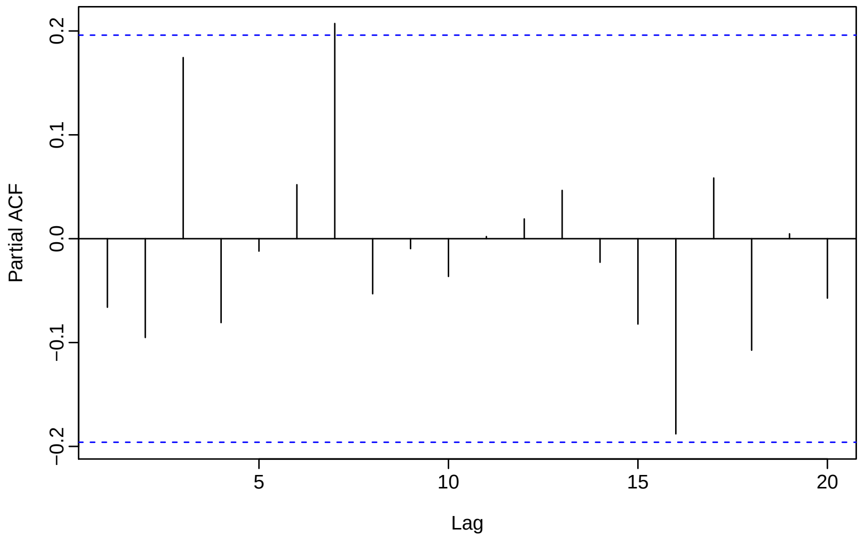

FIGURE 2.12: Correlogram of the seasonal component of the AirPassenger data, a built-in data set of R, over a range of time lags from 1 to 80.

The partial correlation is defined as the degree of correlation not already explained by the correlations previously measured. If the regression of variable A on variables B1, B2 and B3 is considered the partial correlation between variables A and B3 is the degree of correlation not accounted for by their common correlations with variables B1 and B2. In a similar manner the partial autocorrelation is the unexplained correlation after considering the variable and itself at an earlier time period. In a series, if a variable A at time t is correlated with an earlier lag at time t-1 it follows that the variable at t-1 itself is correlated with the previous variable at lag t-2. By extension the variable at time t should also be correlated with the variable at lag t-2, as the correlation will propagate through the series. The partial autocorrelation is the difference the expected correlations due the propagating factors and the actual correlation measured.

The lag at which the PAC tails off can be a good candidate for the AR aspect, looking at Figure **??** this is three - ARMA(3,0)

The AC indicates potential values for the MA component, looking at Figure 2.11 this is one - ARMA(0,1)

A third alternative is an ARMA(p,q) as both the AC and PAC tail off to zero.

Generally one would start with the simplest model so an ARMA(0,1) should be a good place to start the modelling.

2.2.6 Hybrid Models

Auto-regressive (integrated) moving average models have shown themselves to be important modelling methods for time series data, including financial time series data. However the techniques have limitations that have detracted from their popularity, namely their assumption of a linear relationship and the need for a lot of data to produce accurate results. In order to address these limitations a variety of hybrid solutions have been proposed in which ARIMA models are combined with other techniques, often non-linear prediction algorithms.

One combination that has found a lot of attention in the literature is the combination of Artificial Neural Networks (ANNs) with ARIMA. Khashei et al. (2009) report on the use of this combination in a attempt to predict the future price movement in gold and US dollar / Iran rials financial markets. The workers report favourable results in comparison to the techniques alone and suggest the method as having potential for accurate predictions of non-linear time series data. In a similar study Zhang (2003) applied a combination of ARIMA and ANN to various data sets including the British pound / US dollar exchange rate. They observe that in the literature in general these two popular techniques are frequently compared in terms of predictive power with the reported results non-conclusive. Results from the three data sets modelled show that the combination of the two methods outperform the individual ones when the mean squared error (MSE) and mean absolute deviation (MAD) are used as the measure of forecasting accuracy.

Fatima and Hussain (2008) also investigated the impact of a hybrid approach in modelling short term predictions for the Karachi Stock Exchange index (KSE100). The authors reported comparison results for ANN versus ARIMA and a hybrid of ANN/ARIMA. The hybrid solution out-performed the individual ARIMA and ANN models. It is postulated that a rationale for this is that at any point in time financial markets are subject to linear, non-linear and volatility patterns as the cumulative effects of government fiscal and monetary policies and general rumour and political instabilities feed into the market. Under these complex conditions simple models can only capture one aspect of the underlying factors affecting the price series. A hybrid combination approach is more successful as more of the market variance is modelled.

Kriechbaumer et al. (2014) reports on a further hybrid approach to forecast the prices of aluminum, copper, lead and zinc. Previous research has indicated that these markets exhibit a strong cyclic behaviour. In an attempt to factor this into the predictive model ARIMA was coupled with a wavelet approach. Wavelet analysis decomposes a time series into its frequency and time domains in an attempt to isolate this cyclic behaviour. The performance of the ARIMA modelling was shown to be enhanced substantially by the addition of wavelet based multi-resolution analysis (MRA) before performing the ARIMA analysis.

Tan et al. (2010) have also reported the combination of wavelet analysis and ARIMA in the prediction of electricity prices. The general method employed is to transform the original time series data set into a collection of sub-series through the application of wavelet analysis. Subsequent to the transformation a prediction for each sub-series can be made with ARIMA modelling. The final forecasted result is obtained by reforming the sub-series back into the original time series. The authors report resulting showing the enhanced predictive power of the ARIMA wavelet hybrid approach compared to ARIMA and GARCH models used in isolation.

Pai and Lin (2005) reported on attempts to overcome the limitation of ARIMA models in that the time series must be linear by use of an hybrid ARIMA / Support vector machine (SVMs) combination. SVMs have been successfully applied to to non-linear regression problems and the authors have harnessed the strengths of both methodologies in order to predict the prices of a selection of fifty stocks. Results from the work show that the hybrid method out-performs the ARIMA and SVM methods individually.

Rout et al. (2014) report the use of ARMA models in the prediction of exchange rates. The workers note the limitations of ARMA in that the time series data must be linear and stationary, a condition often not met in practical situations and the difficulty in deriving steps one and two (listed previously) in developing the ARMA model. In order to overcome these limitations ARMA is combined with differential evolution (DE) in order to determine the models feed-forward and feed-back parameters. The results from the prediction models generated are compared with models resulting from ARMA in conjunction with particle swarm optimisation (PSO), cat swarm optimisation (CSO), bacterial foraging optimization (BFO) and forward backward least mean square (FBLMS). The workers conclude that the ARMA - DE model produces the best short and long-range predictions from the options tested and is a potentially valuable method in predicting exchange rates on the international finance markets.

Chapter 3

Methodology

3.1 Data Collection

The data used in this study was freely collected from the Yahoo finance web site (www.yahoo.com).

3.2 Data Quality

The data is of high quality with no missing values. It represents the opening, high, low and closing prices for each day that the particular market indice was open for trading.

3.3 Data Description

Data from a variety of national stock market indices was employed in this study. The indices were from a variety of geographic locations with FTSE (UK), DAX (Germany) and CAC (France) all being in Europe, the Dow is from the US, the Nikkei from Japan and AORD from Australia. The data is in the form of so-called daily OHLC (daily open, high, low and close prices) for Monday to Friday (excluding appropriate national holidays) for the period 2000 until the end of 2013. A schematic representation of daily OHLC data can be seen in Figure 3.1. The data sets are freely available from the finance section of Yahoo's website (www.yahoo.com). The first six observations from the DAX data set (German national indice) can be seen in Table 3.1.

The final six observations from the DAX data set can be seen in Table 3.2. Over the period of the data (2000 until the end of 2013) the Dax started at 6691 and finished at

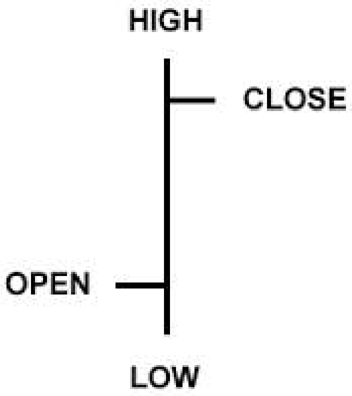

Figure 3.1: A schematic representation of open, high, low and closing prices (OHLC).

9552. Summary statistics for the Dax data set can be seen in Table 3.3. The data set contains 3621 observations and the closing price has ranged between 2202 and 9742 over the period. A graph of the closing prices from 2000 to 2013 and be seen in Figure 3.2 and a graph for 2013 can be seen in Figure 3.3.

Table 3.1: First 6 rows of the Dax data set

Date	Open	High	Low	Close
03/01/2000	6962	7159	6721	6751
04/01/2000	6747	6755	6510	6587
05/01/2000	6586	6586	6389	6502
06/01/2000	6501	6539	6403	6475
07/01/2000	6490	6792	6470	6781
10/01/2000	6785	6975	6785	6926

Table 3.2: Final 6 rows of the Dax data set

Date	Open	High	Low	Close
13/12/2013	9017	9047	8991	9006
16/12/2013	9005	9188	8998	9164
17/12/2013	9143	9162	9085	9085
18/12/2013	9145	9191	9122	9182
19/12/2013	9280	9352	9257	9336
20/12/2013	9371	9413	9353	9400

Table 3.3: Summary statistics of the Dax data set.

Statistic	N	Mean	St. Dev	Min	Max
Open	3,621	5,858.36	1,559.40	2,203.97	9,752.11
High	3,621	5,906.70	1,561.17	2,319.65	9,794.05
Low	3,621	5,804.85	1,557.49	2,188.75	9,714.02
Close	3,621	5,857.74	1,559.39	2,202.96	9,742.96

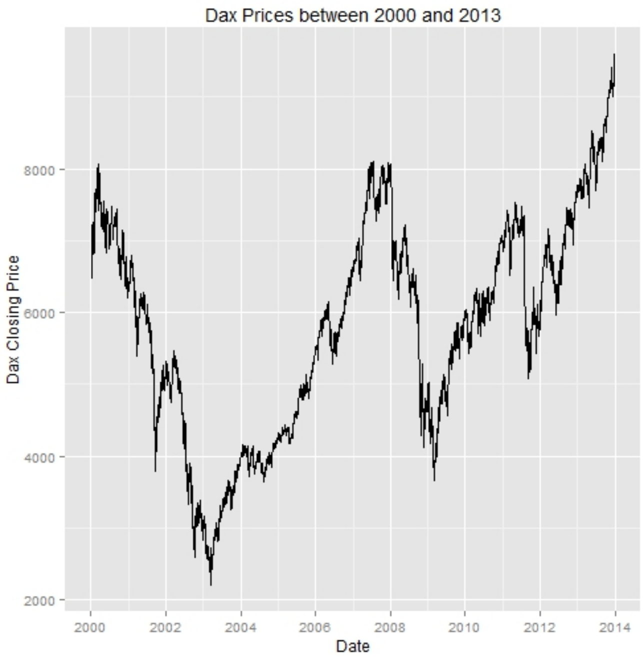

FIGURE 3.2: Graph of German Dax between 2000 and 2013.

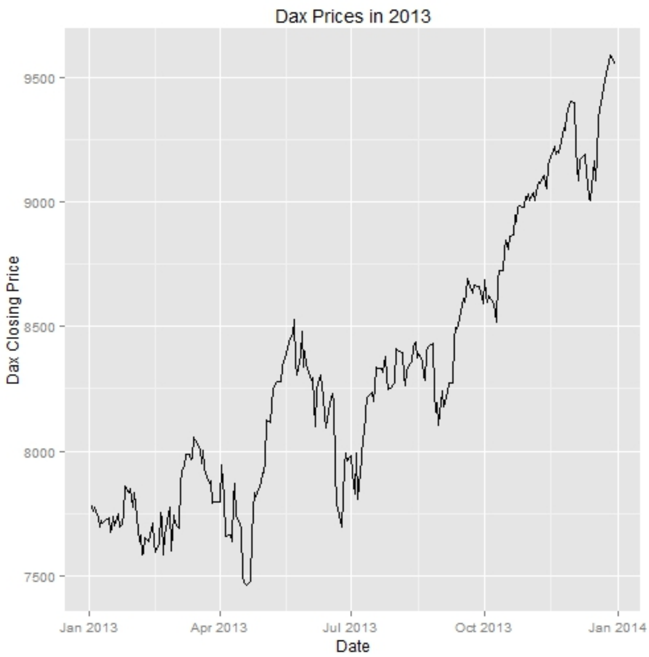

FIGURE 3.3: Graph of German Dax in 2013.

Each data set has a particular set of characteristics and these are important when technical analysis and other analytical techniques are applied to the data set. A variety of these are explored in the following sections. The average amount a market moves is investigated and the term Average True Range is introduced and defined for the data sets. Where the opening and closing prices are in relation to the previous day's high and low values are also considered. Finally the distance between the day's opening and high prices and opening to low prices are investigated. The relative ratios of these values are important when considering which technical analysis may be best suited to a particular market.

3.3.1 Average True Range (ATR)

Wilder (1978) introduced the concept of Average True Range (ATR) as a way to measure a market's volatility or the amount the price is likely to move in any one day. Initially the True Range (TR) is calculated as the maximum of:

1. the today's high price minus today's low price.
2. the absolute value of the today's high minus the previous day's closing price.
3. the absolute value of the today's low minus the previous day's closing price.

Having calculated the TR, an average of a previous number of days is used to derive the ATR. Typically the TR values from the previous 14 days are used.

Absolute values are used in the calculation of the ATR as we are not concerned with the market direction but rather the the amount the market is likely to move. ATRs are typically quoted as absolute values and as such markets trading at higher prices will have higher ATRs. For example the Japanese Nikkei with a value of 14000 will move more in a day than the French CAC with a value in the 4000's.

Dividing the ATR by the closing price is a useful way to see how a security's volatility varies over time. Table 3.4 are the ATR and ATR divided by closing price for the Dax between 2000 and 2013. In absolute terms the ATR varied between 36 and 316, however the value of the indice itself varied a lot. Looking at the ATR value divided by the closing period it can be seen that over the period of 2000 to 2014 the mean value is approximately 2. Thus on average the market can be expected to move 2% of the closing price in any one day. However this value has varied between 0.7% in periods of low volatility to a value of 6.7%.

Table 3.4: ATR and ATR divided by closing price for the Dax between 2000 and 2013

Statistic	N	Mean	St. Dev	Min	Max
ATR	3,556	108.29	45.53	36.07	316.04
ATR/Close	3,556	1.995	1.065	0.700	6.740

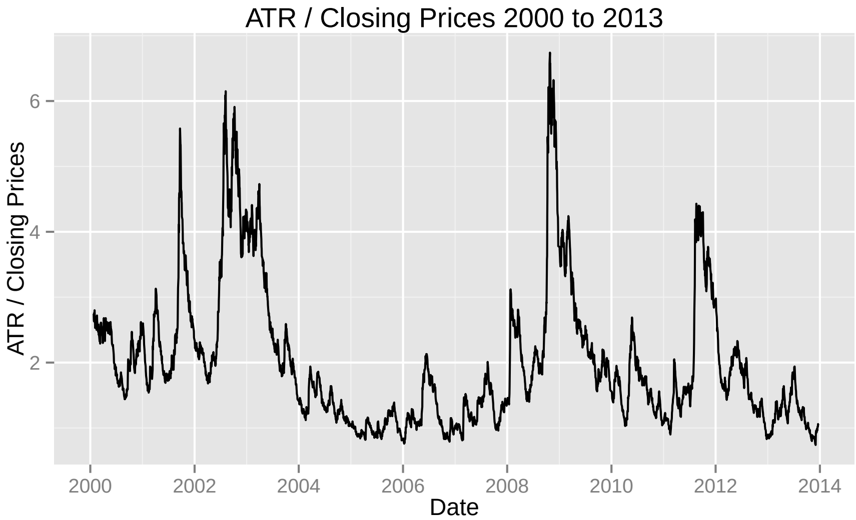

Figure 3.4: Graph of Dax ATR / Closing Price between 2000 and 2013.

3.3.2 Opening Price

Where a market opens in relation to the previous day's high and low price varies across the data sets. This is important and can influence the technical analysis indicator or trading system to utilise. Table 3.5 lists opening price statistics for a variety of national indices. The table lists the number of times that opening prices are between the previous day's high and low prices. These statistics are useful in characterising a market in terms how they move out of hours and can have an impact when choosing a trading system.

Table 3.5: Opening prices in relation to the previous day's high and low values.

Market	Opening Price between Previous Day's High and Low (%)
Dax	75
FTSE	90
CAC	60
Dow	98
Nikkei	53
AORD	79

3.3.3 Closing Price

In a similar fashion to the opening prices the position of the closing prices in relation to the previous day's high and low price are also of interest. In this case, the percentage of closes outside the previous high / low price may indicate that the market may be a good choice for a breakout type of trading system (see section 4.6.1 for details of breakout systems). Likewise the opposite situation occurs if a market frequently finishes within the previous day's high and low levels and may be a candidate for a reversal type of system. The statistics for various national indices can be seen in Table 3.6. Looking at these figures it would suggest that the Dow with a low ratio of finishing outside the previous period's high low values may be a candidate for a reversal type of system and conversely the Japanese Nikkei has a high percentage value and potentially a candidate for a break-out system.

TABLE 3.6: Number of occasions when closing prices finished outside the previous day's high or low values.

Market	Closing Price outside Previous Day's High and Low (%)
Dax	56
FTSE	56
CAC	58
Dow	39
Nikkei	63
AORD	60

The range from opening price to closing price, either up or down, is of interest. Table 3.7 lists the minimum and maximum values in this range and Table 3.8 shows the quantiles for this price range.

TABLE 3.7: Minimum and maximum values for the open to close price range.

Market	Min Value	Max Value
Dax	0	508
FTSE	0	431
CAC	0	313
Dow	0	950
Nikkei	0	1333
AORD	0	347

3.3.4 High / Low Price

Table 3.9 shows the percentage of times that either today's high price crosses yesterday's high or today's low prices dips below the yesterday's low value. The final closing price

Table 3.8: Quantile values for the open to close price range.

Market	25%	50%	75%	90%
Dax	16	39	75	508
FTSE	15	33	63	431
CAC	11	26	49	313
Dow	27	61	119	950
Nikkei	32	71	133	1333
AORD	8	19	36	347

may be between yesterday's high and low or outside of it. The second column of Table 3.9 is the number of times when today's values crossed both the previous low and the previous high in the same day. This is also known as an Engulfing Candlestick (see section 4.7.2). In all the indices the previous day's high or low value is reached the following day in a a large number of instances, in the case of the Nikkei 90% of the time. Conversely, the like hood of both the previous day's high and low values being touched are low, only 5% of occasions in the Australian AORD.

Table 3.9: Number of occasions when today's high or low prices crossed the previous day's high or low values.

Market	Crosses either previous day's High or Low (%)	Crosses both the previous day's High and Low (%)
Dax	89	9
FTSE	87	8
CAC	90	10
Dow	88	9
Nikkei	90	8
AORD	86	5

3.3.5 OH/OL Price Fluctuations

The movements in prices between the open and high (OH) and open to low (OL) are interesting and can have an influence on any trading systems developed. On any given day prices open, move to their lowest point, move to their highest point and then close (not in any particular order). From the OHLC data used in this study the order of these events can not be determined or even the number of times in a day these price points are reached.

In this section we are concerned with the relative sizes of these two price movements, the day's high price minus the opening price (OH) and the opening price minus the low price (OL), one of which is usually greater than the other. We will define the daily "minor" price fluctuation as the smaller of the two price movements. Likewise we will define the larger value as the “mayor” price fluctuation.

Considering the minor price fluctuation, the range of values encountered in the indice markets under study can be seen in Table 3.10. In all cases the minimum value is zero, in other words the market opening price and either the day’s high or low price were the same, the market didn’t dip below or above this level. The second column in Table 3.10 is the maximum value. In the case of the German Dax, there was a day when the market moved 189 points away from its opening price but also moved further in the opposite direction away from the opening price. Clearly this was a highly volatile day on the German markets.

Table 3.10: Minimum and maximum values for the smaller of the daily OH or OL price movement - the “minor” move.

Market	Min Value	Max Value
Dax	0	189
FTSE	0	186
CAC	0	134
Dow	0	379
Nikkei	0	310
AORD	0	114

The quantiles of the minor price movements can be seen in Table 3.11. The 90% quantile is the level at which 90% of the time the minor move is less than this level. This value may be important to know and understand when considering break-out type of systems (see section 4.6). Looking at the value of the Dax we can see that the 90% quantile level occurs at 46, which indicates that if the market has moved to this level it is unlikely to be the day’s minor move (whose level 90% of the time is below this). Perhaps a break-out type of system may be profitable at this point, as once the market has moved this far it is usually a major move and may be expected to continue further in the same direction.

Table 3.11: Quantile values for the smaller of the days OH or OL price movement - the “minor” move.

Market	25%	50%	75%	90%
Dax	5	15	29	46
FTSE	0	7	20	33
CAC	4	11	19	31
Dow	12	43	75	113
Nikkei	5	21	43	72
AORD	0	1	7	13

In contrast to the minor daily price fluctuation, the “major” price fluctuation is defined as the largest of the OH or OL values. The range of values encountered in this price fluctuation in the indice markets can be seen in Table 3.12 and the quantiles of the major price movements can be seen in Table 3.13. Considering the Dax once more, it can be seen that the 25% quantile is approximately equal to the 90% quantile of the minor fluctuation. Thus if the Dax moves approximately 50 points away from the opening it is unlikely to be the smaller of the price movements and much more likely to be part of the larger movement. Knowledge of the minor and major price fluctuations may be useful in developing trading systems.

TABLE 3.12: Minimum and maximum values for the larger of the days OH or OL price movement - the "major" daily price fluctuation.

Market	Min Value	Max Value
Dax	0	530
FTSE	0	471
CAC	0	359
Dow	0	992
Nikkei	0	1737
AORD	0	347

TABLE 3.13: Quantile levels of the maximum values for the larger of the days OH or OL price movement - the "major" daily price fluctuation.

Market	25%	50%	75%
Dax	43	69	106
FTSE	37	56	86
CAC	30	45	69
Dow	92	131	190
Nikkei	76	118	184
AORD	18	30	48

A final consideration in this section is the range of the open to close prices detailed in section 3.3.3. Again considering the German Dax it can be seen that the 50% quantile value is 39 (see Table 3.14 - NOTE repeated from Table 3.8) , which is below the 90% minor fluctuation level.

TABLE 3.14: Quantile values for the open to close price range.

Market	25%	50%	75%	90%
Dax	16	39	75	508
FTSE	15	33	63	431
CAC	11	26	49	313
Dow	27	61	119	950
Nikkei	32	71	133	1333
AORD	8	19	36	347

3.4 Software Tools

3.4.1 R and R Studio

A lot of the experimental results and graphs were produced with the open source programming language R version 3.0.2. For help in the creation and organisation of the R code for this thesis the open-source development environment R Studio version 0.98.490 was used extensively. The following packages were immensely helpful in the preparation of this thesis:

- TTR - provided technical analysis functions
- xts - irregularly spaced time series
- forecast - time series forecasting
- candlestick - Japanese candlestick patterns

3.4.2 Sweave

Sweave (Leisch, 2002) was employed extensively in the preparation of this manuscript. Use of this software enabled tex code to be prepared that embedded R code within it. Thus it was possible to embed results tables and graphs directly and dynamically into the published document. Changes to the code or underlying data is reflected in the manuscript directly.

3.4.3 Rapid Miner

3.4.4 Microsoft Excel and VBA

A lot of data was manipulated in Microsoft Excel and much proofing and testing done with the Visual Basic for Applications programming language built into the Microsoft Office suite of products.

Chapter 4

Technical Analysis

4.1 Introduction

This chapter investigates whether technical analysis can provide a positive expectanacy for financial traders. A variety of technical analysis indicators are employed including MACD, Aroon, Stochastics Oscillator and Rate of Change (ROC) indicator. The experimental reslts from using these indicators are presented in groupings based on the general category of indicator such as trend identification or market reversal indicators. Some technical indicators have a role to play in more than one area, such as MACD, and as such the categorisation is quite general.

The effectiveness of a particular indicator or system is measured in terms of "points" gained, which is also referred to as "PL" (which stands for to profit and loss). The results presented in this chapter are mainly based around systems in which a trade is opened and closed each day, producing a daily PL either positive or negative. The sum of all the individual days produces the total system PL and these values are reported in the results tables. For example, if the market moved from 6000 to 6200 in any one day a PL of either 200 (6200 - 6000) or -200 (6000 - 6200) depending upon which way the trade was placed, would be added to the overall system results.

In addition the results are presented such that returns from "going Long" (expecting the market to rise) are presented seperately from the opposite scenario of "going Short". This is because market behaviour is often diffrent while it is rising than it is while falling and systems may be more adept at predicitng price movements in one of the directions. Further, transactions costs are not taken into account in the results and these would typically be 1 point per trade for the European markets, 2 points for the Dow and 10 for the Nikkei. Thus if a system made a PL of 1000 but it required 2000 trades at 2 points per trade, in reality the system would have lost money.

The results presented in this chapter and the following one are based around trading systems. Essentially the methodology concerned, technical analysis in this chapter and time series analysis in the next, attempt to predict future market direction. The values from the various indicators and forecast techniques are fed into a variety of trading algorithms which use the forecast information to decide whether to make long (expect the market to rise) or short (expect the market to fall) trades. For consistency the algorithms all return the same data object containing the following results:

1. Mkt - the name of the financial market such as Dax, FTSE etc.
2. S Loss - the value of any stop loss applied
3. LongPL - the profit or loss generated from just the "Long" trades.
4. ShortPL - the profit or loss generated from just the "Short" trades.
5. L Win % - the percentage of time the Long trades win.
6. L Trades - the number of Long trades executed.
7. Av L PL - the average profit or loss generated from each Long trade.
8. S Win % - the percentage of time the Short trades win.
9. S Trades - the number of Short trades executed.
10. Av S PL - the average profit or loss generated from each Short trade.
11. misc - miscellaneous information such as the SMA used in the algorithm.

The results from Long and Short trades in particular trading algorithm are considered separately as frequently markets behave differently as they move up as opposed to as they fall. Further the percentage of times the algorithm results in winning trades, the number of trades and the average profit or loss (PL) for each trade is reported for both Long and Short trades. The average PL is primarily reported in the following results tables because this allows comparisons between systems that generate a lot of trades with those such as the algorithms based on candlestick patterns that results in only a small number of trades.

4.2 Base Systems - Naive Methods

Initially two very simple ideas were explored in order for the results to be used as baselines against which the technical indicators explored in the rest of the chapter can be compared. There is an expectation that the use of technical indicators will produce systems that provide much better results than these two so-called naive systems.

The first system simply uses the idea that markets tend to increase in value over time. The algorithm applies a naive approach and simply enters a trade each day expecting the market to rise. The well-known method of ”Buy and Hold” applies the same principles. The total PL of the resulting system is the the sum of all the daily close minus open prices. This approach has been named a “Naive Long System”.

The second approach is equally simplistic, and again is based around opening and closing a trade each day. A notable difference from the first naive system is that the algorithm can result in either a buy or a sell (expecting the market to decline in value) occurring. If a market increased in price the previous day the algorithm “reverses” it and expects the market to fall today. Likewise if the market had fallen the previous day the system buys the market today. This idea has been named “Naive Reversing System”.

4.2.1 Naive Long System

The results of the naive long system can be seen in Table 4.1. The R code for the algorithm which generates the results shown in Table 4.1 can be seen in Appendix A section A.1.2.1. For comparison purposes, the opening prices of the indices in January 2000 along with the closing prices in 2013 can be seen in Table 4.2. In this period three of them increased in value (Dax, Dow and AORD) and three decreased (FTSE, CAC and Nikkei).

Interestingly the PL produced from this simple system doesn’t match the price differentials seen in Table 4.2. The German Dax indice produced a marked loss in the naive system even though it increased 37% during this period. The Japanese Nikkei declined by over 2600 points in this period, whereas the system reported a loss of over 18000 points in the same period. On the other hand the US Dow increased by around 5000 points during the period of the study but the trading algorithm produced a positive result of almost 10000. These discrepancies can be explained by the fact that the system was using prices from the market’s opening to closing times, which represents approximately eight hours of trading between 8am and 4pm local time. These price movements don’t account for the rest of the hours, the so-called out of market hours, when the market prices also change. Clearly the markets show different characteristics in the amount they move during market hours compared to out of market hours. The Nikkei, Dax and CAC have a tendency to fall during market hours and rise during out of market hours. The opposite situation occurs for the Dow.

Table 4.1: Naive Long System. A very simple system in which the algorithm assumes the market will rise and enters a long trade each day.

Mkt	LongPL	L Win %	Av L PL
Dax	-1714	52	0
CAC	-6725	50	-2
FTSE	149	51	0
Dow	9816	53	3
Nikkei	-18125	49	-5
AORD	972	52	0

Table 4.2: Prices of six national indices in January 2000 and December 2013.

Date	Start 2000	End 2013	Difference	% Change
Dax	6961	9552	+2591	+37
CAC	6024	4250	-1774	-29
FTSE	6930	6749	-181	+-3
Dow	11501	16576	+5075	+44
Nik	18937	16291	-2646	-14
AORD	3152	5353	+2201	+70

Altering the algorithm slightly so that a trade represents the difference between the previous close and today's close affects the results markedly. A full 24 hour period is now accounted for and the system reflects the overall market movement during this period. These results can be seen in Table 4.3 and the amended R code can be seen in Appendix A section A.1.2.2.

Table 4.3: Naive Long System changed such that the trading period is the previous close price minus today's close.

Mkt	LongPL	L Win %	Av L PL
Dax	2649	53	1
CAC	-1667	51	0
FTSE	86	51	0
Dow	5219	53	1
Nikkei	-2712	51	-1
AORD	2229	53	1

4.2.2 Naive Reversing Previous

The second naive method is to reverse the previous day's movement. For example, if the market closed up the previous day the algorithm follows this by trading short for the current day (the R code for this algorithm can be see in Appendix A section A.1.2.3) . The results from this system can be seen in Table 4.4.

Table 4.4: Naive system which reverses the previous day's trade direction.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	947	3131	53	1	49	2
CAC	940	7810	53	1	53	4
FTSE	4284	4115	53	3	50	2
Dow	15799	6047	56	10	49	3
Nikkei	2324	20486	51	1	54	12
AORD	1264	237	53	1	48	0

For all the markets tested, this second naive system produces positive results especially for the Nikkei and CAC trading short and the Dow trading long. These results demonstrate that markets have a tendency to reverse direction each day, they move up one day then down the next. This behaviour is also observed in trending markets, and market "pull-backs" are a well-known phenomena.

4.2.3 Summary of Naive Baseline Systems

Of the two naive systems tested, the "reversing" methodology produces the best results in terms of profit and loss by quite a margin. Thus the results from the "Naive Reversing System" will be used to compare the performance of technical indicators being tested in the following sections.

4.3 Trend Detection Indicators

One of the most widely used phrases in financial trading is "the trend is your friend". Thus, most market participants are interested in identifying the start of trends, their direction and strength. In this section a variety of technical indicators that purport to assist in this important task are tested.

4.3.1 Simple Moving Average (SMA) System

One of the most popular and widely utilised technical indicators is the simple moving average (as detailed in Chapter 2 section 2.2.1.1). The effectiveness of SMA as an aid to predicting future market movements has been widely debated, with views mixed. A system based on simple moving averages is presented here, and the R code used to generate the results can be seen in Appendix A section A.1.3.1. The algorithm trades daily, opening and closing a trade each day. If the market opens above the SMA the algorithm trades long and trades short when the market opens below the SMA.

Table 4.5 lists the results from passing a variety of national index data sets (see Chapter 3 for details) to the algorithm. For each indice the algorithm is run with values of 5, 25, 50, 100 and 200 for the SMA period. In general the results are poor, especially after consideration is given to any transaction costs. The CAC and Nikkei produce negative results for long trades, the FTSE negative results across the board, and the Dow negative returns on the short side.

TABLE 4.5: Results from SMA system.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL	SMA
Dax	2113	3278	54	1	50	2	5
Dax	1367	3427	54	1	50	2	25
Dax	779	3447	54	0	51	3	50
Dax	714	2339	54	0	51	2	100
Dax	3401	4416	55	2	52	4	200
CAC	-3952	2338	49	-2	49	1	5
CAC	-5058	1615	49	-2	49	1	25
CAC	-5323	1029	49	-3	49	1	50
CAC	-2363	3188	50	-1	50	2	100
CAC	-1219	3923	50	-1	50	3	200
FTSE	-4724	-5331	49	-2	46	-3	5
FTSE	-1013	-1940	51	0	47	-1	25
FTSE	-2226	-2769	50	-1	47	-2	50
FTSE	-889	-1692	51	0	48	-1	100
FTSE	-158	-835	52	0	49	-1	200
Dow	408	-9630	52	0	46	-6	5
Dow	1138	-9204	53	1	46	-7	25
Dow	5478	-5876	53	3	47	-4	50
Dow	2576	-8220	53	1	47	-6	100
Dow	6378	-4567	54	3	48	-4	200
Nikkei	3078	20401	51	2	54	13	5
Nikkei	-7878	10770	48	-4	52	7	25
Nikkei	-6054	11408	49	-4	52	7	50
Nikkei	-6235	8381	49	-4	52	5	100
Nikkei	-5928	6836	49	-4	52	4	200
AORD	5009	3929	55	3	51	3	5
AORD	3701	2674	54	2	50	2	25
AORD	2804	1864	54	1	50	1	50
AORD	2688	1521	54	1	50	1	100
AORD	2574	1616	54	1	51	2	200

One aspect of a trading system of this nature worth considering is the risk / reward profile. As written in its current form the SMA algorithm has an unlimited profit potential (trades are left to run until the end of the day) and an unlimited potential loss for the same reason. Often traders employ what is known as a “stop loss”. This is a level in the market that if reached during a trade will cause the trade to close. The risk is now therefore reduced to this value while the profit is still potentially uncapped. Table 4.6 lists the results of using a stop loss with the SMA system.

The logic of the stop loss was coded as follows. Considering a long trade (the opposite holds true for trading short), where there is an expectation that the market to rise, a the stop loss would be triggered if the market fell to a certain level. Thus in the algorithm for a long trade the distance from the opening price to the low is calculated and this is compared to the stop loss value. If the open to low value exceeds the stop loss value the PL for this particular trade is set at the stop loss value, for example a loss of 100 points. One point of note is the fact that after hitting this low level the market may well recover and move upwards as originally expected. In many cases a trade that ultimately would have been profitable may be "stopped out" by the natural wax and wane of the markets. Therefore the impact of a stop loss is the balance btween lost good trades and the reduction in the lost PL from losing trades. The size of the stop loss determines the impact of the two competing situations.

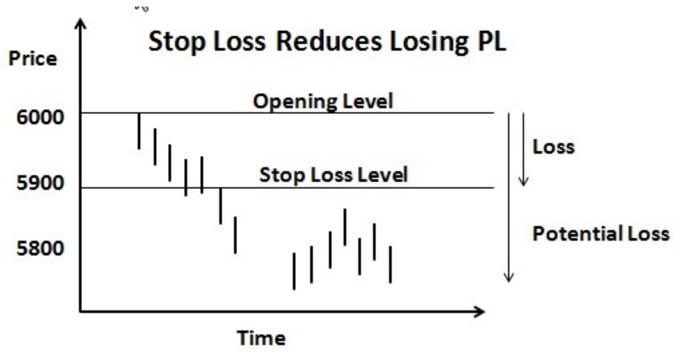

FIGURE 4.1: Situation in which using a Stop Loss is beneficial, with a losing PL being reduced..

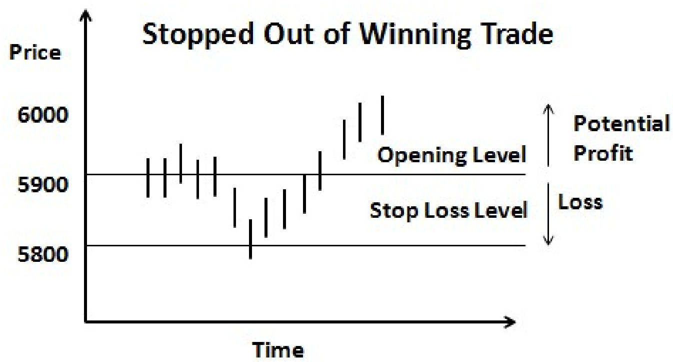

FIGURE 4.2: Situation in which using a Stop Loss is detrimental, being "Stopped Out" of an ultimately winning trade.

Figure 4.1 shows the situation in which a stop loss is benefiical. The potential large loss is reduced to the value of the stop loss value. Figure 4.2 illustrates the alternative scenario of being "Stopped Out" of an ultimately winning trade, an undesirable outcome.

It is the ratio of these scenarios that ultimately determines whether using a stop loss is a sound strategy.

Table 4.6: Results from SMA system with Stop Loss.

Mkt	S Loss	LongPL	ShortPL	L Win %	L Trades	S Win %	S Trades	SMA
Dax	-50	3652	6618	51	2069	42	1360	100
Dax	-100	1392	5272	54	2069	50	1360	100
CAC	-50	-172	5178	50	2012	47	1475	100
CAC	-100	-1822	4658	50	2012	50	1475	100
FTSE	-50	1114	6303	50	2044	43	1389	100
FTSE	-100	-885	1892	51	2044	47	1389	100
Dow	-50	-18212	-8229	32	2125	22	1297	100
Dow	-100	-11771	-14696	49	2125	36	1297	100
Nikkei	-50	8258	33882	38	1643	39	1696	100
Nikkei	-100	2550	25582	47	1643	48	1696	100
AORD	-50	4008	3730	54	2230	49	1219	100
AORD	-100	2881	2149	54	2230	50	1219	100

Comparing Tables 4.5 and 4.6 it can be seen that applying the stop loss has been on the whole beneficial to the results obtained, with the exception of those from the Dow which were markedly negatively impacted.. Essentially losing trades have been truncated while winning trades have been left to develop. One question that needs to be addressed is what value is appropriate for a stop loss. If the value is large the benefits of cutting losses is lost, whereas if it is too small a large number of trades will be "stopped out". Many traders use a value based on the Average True Range (see Chapter 3 section 3.3.1 for details) as this allows for the volatility of a particular market.

4.3.2 Moving Average Convergence/Divergence (MACD)

Moving Average Convergence/Divergence (MACD) is a trend following indicator, developed by Appel (2005), that is formed from the relationship of two moving averages, see Appendix B section B.1 for more details. The value of MACD itself is the difference between two exponential moving averages (EMA), a "slower" e.g. 26 day value and a "faster" e.g. 12 day value. In addition an EMA of the MACD value is calculated, which is set to 9 days in the following algorithm, which acts as a "signal" line.

The MACD is generally used two ways. Firstly, it can be used to derive the general trend of the security so that the market participant can trade with the trend. Secondly, it can be employed to identify periods when the market is "over-bought" or "over-sold" and can be expected to reverse direction (Achelis, 2014).

In order to identify the trend of a market using the MACD indicator, the relative values of the MACD itself and the signal line are used. If the value of the MACD exceeds the signal it is considered "bullish" and the market is expected to rise in price. Similarly in the opposite situation where the value of the signal is greater than the MACD the trend of the market is expected to be down.

Table 4.7 lists the results of using the MACD indicator in just such a way. The MACD value itself is generated using the EMA of the opening prices with values of 26 and 12 for the slow and long averages and a value of 9 days for the indicator line.

The trading algorithm splits the results into two values, days when the system expected the market to rise and days when a market decline were predicted (see Appendix A section A.1.3.2 for details of the R code used and Appendix B section B.1 for background information on the MACD indicator). At the start of each day if the MACD value exceeds the signal line the algorithm adds the value of the close price minus the opening price to the "Long PL" running total. Likewise in the opposite situation with the signal line greater than the MACD, the value of the open price minus the close price is added to the "Short PL". Table 4.7 lists the results of the algorithm run against a variety of national indices.

Table 4.7: Results from system using MACD as a Trend Indicator.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	-791	1424	53	0	48	1
CAC	-4153	2188	49	-2	49	1
FTSE	63	-839	51	0	48	0
Dow	5592	-5190	53	3	46	-3
Nikkei	-4078	14064	49	-2	52	8
AORD	2563	1569	54	1	49	1

4.3.3 Aroon Indicator

Developed by Tushar Chande in 1995, the Aroon indicator was designed to identify trending markets (aro). The word aroon means "dawn's early light" in Sanskrit and this indicator tries to pin point the dawning of a new trend. Essentially it is a measure of the time since the occurrence of a high/low price in a particular period. Further details can be seen in Appendix B section B.2.

Table 4.8 shows the results of applying the Aroon algorithm (shown in Appendix A section A.1.3.3) on the data of the national indices. The results are promising with the indicator making positive predicitions in most of the markets and doing particularly well in declining markets.

Table 4.8: Aroon trend indicator.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	5308	5257	56	3	51	4
CAC	-1638	4919	50	-1	52	4
FTSE	3042	5715	52	2	51	5
Dow	12131	3811	55	7	49	3
Nikkei	-4852	12013	49	-3	52	10
AORD	3735	3540	55	2	50	3

Table 4.9: Aroon trend indicator with stop loss.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	5410	7465	56	3	50	6
CAC	-1224	6086	50	-1	52	5
FTSE	3091	8015	52	2	51	7
Dow	-5922	-9341	49	-3	37	-8
Nikkei	3153	22177	46	2	47	18
AORD	3786	4159	55	2	50	4

The affects of using a stop loss was investigated and the results shown in Table 4.9. The use of a stop loss was beneficial in all cases except the Dow, in which case it had a catastrophic impact turning a winning system into a losing one. The impact of the stop loss is shown in Table 4.10 which lists the difference in PL between the original results without a stop loss and the revised ones with it.

Table 4.10: Impact of stop loss on Aroon.

Market	Long Difference	Short Difference
Dax	102	2208
CAC	414	1167
FTSE	49	2300
Dow	-18053	-13152
Nikkei	8005	10164
AORD	51	619

4.4 Market Reversal Indicators

The alternative to trend detection indicators are market reversal indicators, designed to identify when a trend may be ending and the market will start to move in the opposite direction. Many traders advocate that this type of trading should be avoided and cite the old phrase “never try to catch a falling knife”. Nevertheless a variety of market reversal technical indicators are explored and their effectiveness noted.

4.4.1 Parabolic Stop-and-Reverse (SAR)

The parabolic stop-and-reverse (SAR) is a method to calculate a trailing stop. This technical indicator was developed by J. Welles Wilder and is detailed in his book New Concepts in Technical Trading Systems (Wilder, 1978). A trailing stop is related to the stop loss explored previously but differs in that it is adjusted as the market moves. If the market rises it is adjusted upwards, for example being set a certain distance from the highest high of a particular period. The parabolic SAR calculates the point at which a long trade would be closed and a short position entered, the assumption being that the market participant is always in the market either short or long. More details on the theory and calculations to generate the parabolic SAR can be found in Appendix B section B.3.

Table 4.11 lists the results from passing a variety of national index data sets to an algorithm using the parabolic SAR. The R code used to generate these results can be seen in See Appendix A section A.1.4.1. On the whole the results from these initial tests are very disappointing. Only three of the national indices generated positive results and only the Japanese Nikkei provided reasonable returns.

Table 4.11: Results from SAR system.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	-3856	-2353	53	-2	48	-2
CAC	-5584	1034	49	-3	49	1
FTSE	-1141	-1663	51	-1	48	-1
Dow	-1301	-11112	52	-1	46	-7
Nikkei	-5767	12424	49	-3	52	8
AORD	2071	1097	53	1	49	1

4.4.2 MACD as reversal Indicator

MACD can also be used as a reversal indicator. Recalling that the MACD is formed from the relationship of two moving averages, when the faster one moves sharply away from the slower one (i.e.the value of MACD rises) this could be an indication of an "over-bought" market and that a reversal is approaching. In this situation the trader would place a sell trade. The opposite is true for a large negative MACD, and it is postulated that the market may well reverse upwards.

Table 4.12 shows the results of applying the algorithm shown in Appendix A section A.1.4.2 on the data of the national indices. In the algorithm the 15% and 85% quantile of the MACD value is calculated and this is used to decide on the reversal point. Once the 85% value is exceeded the algorithm predicts a reversal will occur and trades short, the opposite is true for the 15% level which triggers a long trade. Overall the results are very modest, with small positive gains being seen in 5 of the 6 national indices.

TABLE 4.12: MACD can also be used as a trend reveral indicator.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	391	407	49	1	48	1
CAC	-545	2657	51	-1	55	5
FTSE	2080	1649	53	4	53	3
Dow	3882	-807	52	7	48	-2
Nikkei	199	2828	51	0	52	6
AORD	-319	-584	50	-1	49	-1

4.5 Momentum Indicators

Momentum indicators are closely related to the trend indicators introduced in section 4.3. They are concerned with trending markets but differ in that the strength of the trend is also included in the information the indicator attempts to portray.

4.5.1 Stochastic Oscillator

The stochastic indicator is one of the oldest in widespread use today having been developed by George Lane in the 1950s (Lane, 1986). It measures the relative position of a market's closing price in the range between the low and high of the period of interest. This is of interest as some market participants believe that financial markets essentially swing between price boundaries marked by where the market closes in this range (Williams, 2011). Thus markets increase until the close is at the top of this range before changing direction and moving down until it is at the bottom of the high low range.

The stochastic is usually represented by two lines %K which is the position of the price within this high low envelope described above, and %D a moving average of %K (see Appendix B.4 for more details). It can be used a number of ways and one popular technique is to go long when the %K crosses above %D and to go short in the opposite situation. Table 4.13 lists the results from passing a variety of national index data sets to an algorithm which uses the relative position of %K and %D to decide which way to trade. The R code used to generate these results can be seen in See Appendix A section A.1.4.3.

TABLE 4.13: Results from Stochastics system.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	-28	1673	53	0	49	1
CAC	-4540	1817	48	-3	48	1
FTSE	-73	-744	51	0	48	0
Dow	867	-9414	53	0	46	-5
Nikkei	-10591	7802	48	-6	51	5
AORD	2839	1780	54	2	49	1

The results from Table 4.13 for this system are very modest with only the Australian ORD showing positive values for both long and short trades. Adding a stop loss of 100 points increases the PL across the board except for the case of the Dow where again the stop loss has had a detrimental affect. The results from using a stochastic based system with a stop loss can be seen in Table 4.14.

TABLE 4.14: Results from Stochastics system and using a Stop Loss.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	1173	3889	52	1	48	2
CAC	-3493	2730	48	-2	48	2
FTSE	1640	1424	51	1	48	1
Dow	-13969	-27388	45	-8	37	-16
Nikkei	1647	17977	45	1	46	10
AORD	3028	1974	54	2	49	1

4.5.2 Rate of Change (ROC)

The Rate of Change (ROC) indicator is a simple and widely observed technical indicator. It is the difference between the current price and the price several observations ago. See Appendix B section B.5 for details. If this value is large, either positive or negative it is indicative of a strongly trending market with a lot of momentum either upwards or downwards. The R code for a trading system exploiting these ideas can be seen in Appendix A section A.1.4.4. The results can be seen in Table 4.13 which lists the results from passing a variety of national index data sets to the algorithm.

4.6 Break-out systems

This section explores some trading systems that use a particular price as the indicator to place a trade. The first system uses the simple idea of trading when the previous day's high or low is passed. The second idea is related to the results generated in Chapter

TABLE 4.15: ROC.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	930	148	50	2	50	0
CAC	952	956	53	2	51	2
FTSE	1147	1880	51	2	51	4
Dow	8517	3396	58	16	49	6
Nikkei	2971	2546	50	6	52	5
AORD	271	1325	51	1	52	2

3, where the 90% quantile for the day's minor move was calculated. The system tested here is to simply trade long or short when this point is reached in a day.

4.6.1 Daily High / Low Breakout System

Table 4.16 lists the results from a trading system based around the idea of trading after the previous day's high or low price has been breached. The R code used to generate these results can be seen in See Appendix A section A.1.5.1.

TABLE 4.16: Results from Daily High / Low Breakout System.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	21073	21229	58	11	56	13
CAC	14252	20176	58	8	59	12
FTSE	13239	18614	59	7	59	12
Dow	-19355	-27334	42	-11	38	-17
Nikkei	74600	81645	64	44	64	49
AORD	19347	21244	67	11	65	14

Referring to Table 4.16 we can see that this system produces good results, with the exception of the US Dow. This ties in with the data in Chapter 3 Table 3.6 which shows that the Dow only closes outside of the previous low or high price a relatively low number of times. Likewise good results are seen with the Japanese Nikkei from the breakout system and this tallies with the high proportion of the time in which it closes above or below the previous day's high or low.

4.6.2 Break Out of 90% Quantile Level

A second system utilising the break-out concept is presented in this section. In Chapter 3 one characteristic of the markets was noted, namely that each day the market moves from its opening price to a low price and then to a high price (not necessarily in any particular order). One of these moves (O-H vs O-L) is greater than the other was termed the major move and the smaller move was called the minor move. The algorithm generating the results in this section (see Appendix A section A.1.5.2) makes a long or short trade after the market has passed the 90% quantile of the minor move. Table 4.17 lists the results from this algorithm.

TABLE 4.17: Results from a trading system using 90% Quantile level.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	7841	6371	56	6	53	4
CAC	2647	5085	54	2	52	3
FTSE	10758	15295	56	7	54	10
Dow	-30262	-34854	39	-24	37	-28
Nikkei	23606	31830	58	16	56	20
AORD	16730	19357	63	9	62	12

4.7 Candlestick Patterns

As previously noted in Chapter 2 section 2.1.5 candlestick patterns are visual representations of price movements over the course of a particular time period (often a day) in terms of the market's opening, closing, high and low prices. The pattern generated from these price markets are categorised and named depending upon the visual shape they produce. Thus candlestick patterns represent the counter forces of buyers and sellers throughout the trading period. This section analyses some well known candlestick patterns for predictive power in making trading decisions.

4.7.1 Hanging Man, Hammer, Inverted Hanging Man and Shooting Star

Four well-known patterns that are generally considered to indicate the possible end of a trend and the start of a reversal are the so-called Hanging Man, Hammer, Inverted Hanging Man and Shooting Star candlestick patterns.

Figure 4.3 is a diagram of a Hammer and Inverted Hammer patterns. Both patterns have a small "body" (the distance between the open and close prices) and a long "shadow" (the distance between the high and low prices). In the diagrams presented here a white candlestick means the market price increased over the course of the day while a black one means the market fell. The body of the candlestick is white in this case, indicating that the market moved up (the closing price was above the opening price), although by only a small amount. Hammer and Inverted Hammer differ in that the long shadow in

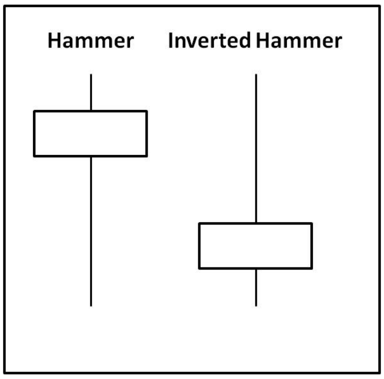

FIGURE 4.3: Hammer and Inverted Hammer.

hammer is generated from a low price whereas the shadow of Inverted Hammer goes upwards as it is indicative of the period's high price.

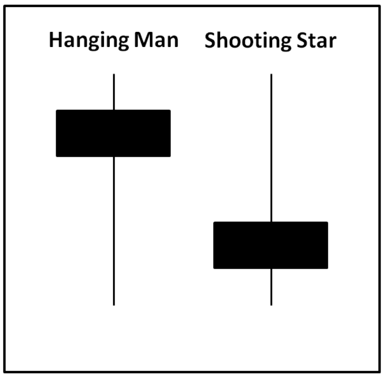

FIGURE 4.4: Shooting Star and Hanging Man.

Figure 4.4 is a diagram of Hanging Man and Shooting Star, these being the opposite to Hammer and Inverted Hammer. In this case the market direction is down, albeit only by a small amount, and thus the body of the candlestick is a different colour, in this case black. Again both patterns have long shadows, the direction of which determines if the pattern is Hanging Man or Shooting Star.

Both sets of patterns Hammer/Inverted Hammer and Hanging Man/Shooting Star are considered to indicate that a trend is coming to a close and a reversal could be looming. In the case of Hammer/Inverted Hammer if they are encountered during a down trend they could indicate that the selling pressure is easing and a market move to the upside could happen soon. The opposite is true for Hanging Man/Shooting Star. When these are encountered in an up trend they often indicate that the trend is ending and a reversal may occur. Figure 4.5 shows daily candlestick patterns for the German Dax over 22 days in April 2014. A Shooting Star is circled on the 6th April and a Hanging Man on the 23rd April. In each case they occur while the market is rising and in each case it reverses immediately afterwards.

FIGURE 4.5: Daily candlestick patterns from the German Dax over 22 days in April 2014 with Shooting Star and Hanging Man circled.

In order to have a system based on candlestick patterns, the pattern itself must be identified in code. A Hammer and Hanging Man are essentially the same pattern except Hammer has a close higher than the open whereas Hanging Man represents a decline in the price. For these patterns three components are defined, the length of the upper shadow (short), the size of the body (short) and the length of the lower shadow. In the trading system that follows these were defined as:

1. Upper Shadow - the value of the day's high minus the high of the body is less than 10% the total High-Low range.
2. Body - is larger than 10% the total High-Low range.
3. Lower Shadow - the value of the day's low minus the low of the body is greater than 66% of the High-Low range.

Analysing the Dax data set running from 2000 to 2013 with 3570 observations, and using the criteria described above 35 Hammer and 48 Hanging Man patterns can be detected.

Inverted Hammer and Shooting Star are again the same pattern except in Inverted Hammer the price rose. In the later system these are defined as:

1. Upper Shadow - the value of the day's high minus the high of the body is at least 66% the total High-Low range.
2. Body - is larger than 10% the total High-Low range.
3. Lower Shadow - the value of the day's low minus the low of the body is less than 10% of the High-Low range.

Considering the Dax data set again, occurrences of these patterns are quite rare with 30 Inverted Hammers and 17 Shooting Stars in 3570 observations.

Results from a trading system based on the Hammer / Inverted Hammer can be seen in Table 4.18 and the R code in Appendix A section A.1.6.1. The algorithm simply places a buy the day after a Hammer or Inverted Hammer occur, the assumption being that these patterns indicate that the market is about to rise.

Table 4.18: Results from Hammer / Inverted Hammer.

Mkt	LongPL	L Win %	L Trades	Av L PL
Dax	594	53	126	5
CAC	-793	44	149	-5
FTSE	834	58	188	4
Dow	2097	59	88	24
Nikkei	-2202	48	147	-15
AORD	-809	46	236	-3

An alternative approach is to look for Hammer and Inverted Hammer patterns occurring in a down trend, in which case it could signal the end of the down trend and the start of a reversal. Table 4.19 shows the results of using the Hammer and Inverted Hammer to predict a price rise during a down trend. An aroon down value of greater than 65 (with a 20 day look back period) is used to define the down trend. The algorithm can be seen in Appendix A section A.1.6.1.

Table 4.19: Results from Hammer / Inverted Hammer occurring in a downtrend as defined by the aroon value.

Mkt	LongPL	L Win %	L Trades	Av L PL
Dax	594	53	126	5
CAC	-793	44	149	-5
FTSE	834	58	188	4
Dow	2097	59	88	24
Nikkei	-2202	48	147	-15
AORD	-809	46	236	-3

4.7.2 Engulfing Candlestick

The "Engulfing" pattern, either Bull or Bear is another widely considered candlestick pattern and is depicted in Figure 4.6. This pattern has a lower low and a higher high than the preceding candlestick and is usually interpreted as indicating a change in direction of the trend. Engulfing candlesticks can be either bullish, where the closing price is above the opening price or bearish when the market moves down.

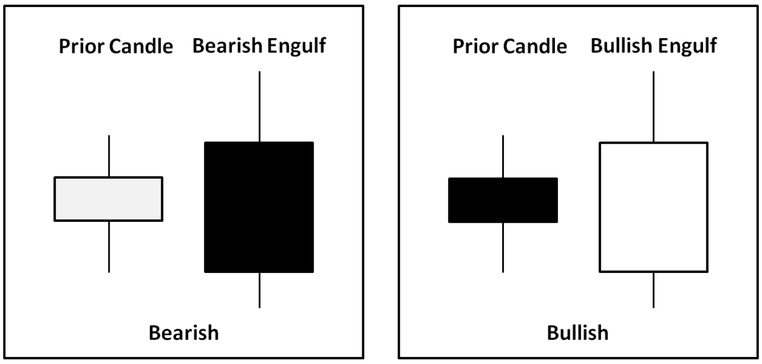

Figure 4.6: Engulfing Pattern.

Table 4.20 lists the results from passing a variety of national index data sets (see Appendix A section A.1.6.1 for details) to an algorithm that buys or sells the market depending on the presence of an Engulfing pattern.

Table 4.20: Results from Engulfing Candlestick.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	-920	-258	44	-7	46	-2
CAC	-319	228	45	-2	50	1
FTSE	-1721	1185	51	-4	50	3
Dow	-770	-3662	48	-4	35	-28
Nikkei	-3823	-1166	37	-39	44	-11
AORD	-6	-600	53	0	46	-3

Table 4.21 lists the results from extending the algorithm such that trades are only taken in either up or down trends, as defined by the aroon indicator. The R code for the amended algorithm can be see Appendix A section A.1.6.1.

Table 4.21: Results from Engulfing Candlestick with Aroon.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	-874	-513	38	-20	43	-7
CAC	-118	-666	49	-3	30	-11
FTSE	-1217	-782	47	-8	48	-3
Dow	202	-1154	45	4	44	-11
Nikkei	-1522	-1733	38	-59	37	-32
AORD	-49	-27	53	-1	50	0

4.7.3 Doji

Doji is a well-known candlestick pattern that can appear on its own or as a component of a pattern. A Doji forms when the open and close price are similar and there is an upper and lower shadow, thus they often resemble a cross. Variations within Doji include the Dragonfly and Gravestone Doji, see Figure 4.7. In an up trend Doji (especially Gravestone) can indicate a reversal could occur and likewise in a down trend a Dragonfly could suggest an upward move is about to start.

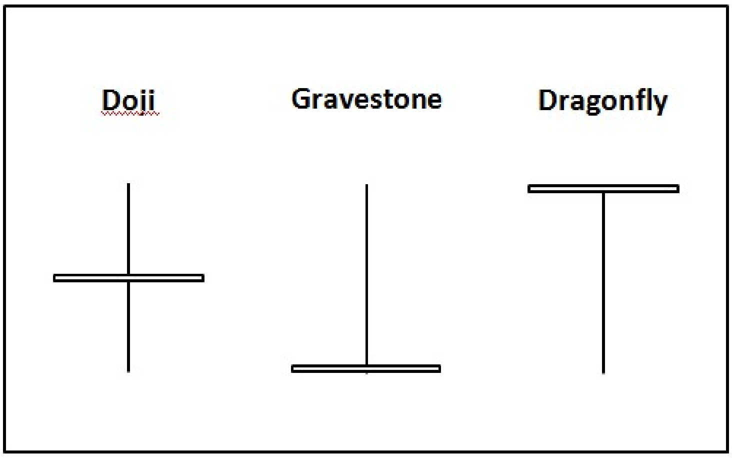

FIGURE 4.7: Doji Pattern.

Table 4.22 lists the results from passing a variety of national index data sets (see Appendix A section A.1.6.1 for details) to an algorithm that buys or sells the market depending on the presence of a Doji. In an up trend, as identified by the aroon indicator, a Doji or Gravestone is used to initiate a sell and conversely in down trend a Doji or Dragonfly is used as a signal to buy.

TABLE 4.22: Results from Doji Candlestick with aroon.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	-826	-1132	53	-8	52	-6
CAC	-747	-326	46	-6	49	-2
FTSE	-697	418	53	-8	52	3
Dow	-763	-2869	51	-5	50	-10
Nikkei	1296	-2944	55	12	45	-22
AORD	-115	195	54	-1	54	2

Chapter 5

Time Series

This chapter will explore the use of time series analysis techniques to generate models for forecasting futures prices in various national stock market indices. Usually, in trying to predict the future behaviour of financial markets the direction they will move, either up or down, is of more interest than the actual value itself. Thus in this chapter predictions of the future direction as well as the actual value itself are attempted. Further when considering the general direction of the market it can be represented as a categorical variable, i.e. "up" or "down" or as a variable for example 1 for "up" and 2 for "down". The choice has implications for which algorithm can be employed in the forecasting technique because of the differing data types. A variety of time series models are developed in this chapter using ARIMA and hybrid ARIMA methods.

5.1 ARIMA Models

The use of Auto-Regressive Integrated Moving Average (ARIMA) models, see section 2.2.4 for details, was explored in order to forecast future prices for financial markets. The process of fitting an ARIMA model to a time series is quite challenging and involves the following general steps:

1. Plot the data to get a general feel for the time series and to establish if it is stationary.

2. Stabilize any variance in the data with a transformation process such as the Box-Cox method.

3. ARIMA models work with stationary data, so if necessary, take differences of the data until it is stationary.

4. Examine the auto-correlation and partial auto-correlation (ACF/PACF) plots in order to determine if an AR(p) or MA(q) model is appropriate.

5. Test the chosen model(s), using the AICc to determine if a better model is available.

6. Check the residuals from the best model by plotting the ACF, and doing a portmanteau test on them. If the results from these tests do not look like white noise, a modified model may be required.

7. Finally, once the residuals have a similar pattern to white noise, the model can be used to generate forecasts.

In recent years automatic forecasting algorithms have become available and are widely used (Hyndman and Kh, 2008). These are necessary in a variety of circumstances, especially when organisations are faced with the need to repeatedly carry out a large number of forecasts and the human effort required renders manual means impractical. The auto.arima() function found in R's "forecast" package is an example of an automatic algorithm for ARIMA models. This function automates steps 3, 4, and 5 of those outlined previously, in the general steps required for ARIMA modelling. In the following sections, the general steps are used to generate an ARIMA model manually, and then the automatic algorithm is utilised to build one.

5.2 Manual Generation ARIMA of Models

5.2.1 Data Exploration

The first step, as always is to explore the data. Figure 5.1 shows the UK's FTSE 100 index between the years 2000 to 2013. Over this time period the series has shown strong trends to move up and down and a uniform variance. Because the time series is non-stationary it will need to be transformed into a stationary series before ARIMA modelling can be undertaken.

5.2.2 Adjusting for non-uniform variance and non-stationariness

The variance within the FTSE time series is relatively uniform and thus this data set doesn't need stabilizing with regard to this. If it did, a Box-Cox transformation could be used. However, over this time period the FTSE 100 exhibits marked non-stationariness and requires adjusting accordingly. One such technique to make a data set stationary is differencing. Instead of using the actual observations the differences between two

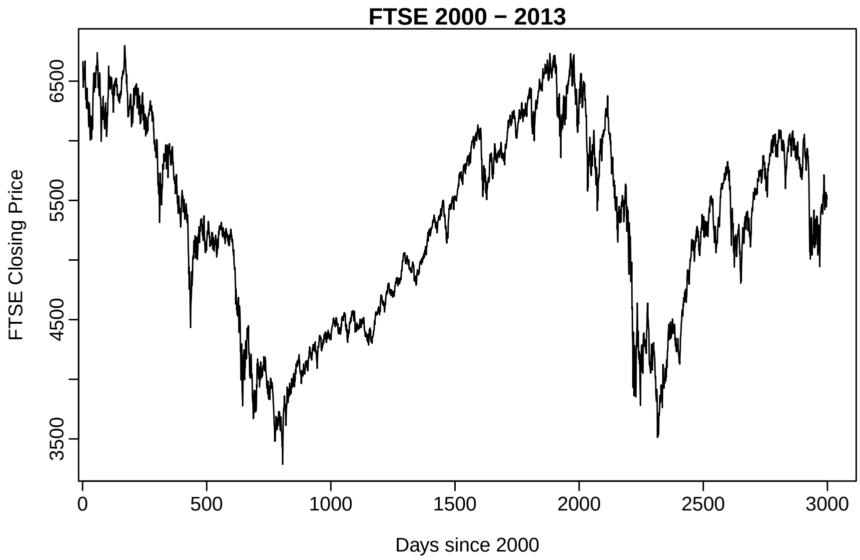

Figure 5.1: UK's FTSE 100 index between the years 2000 to 2013.

adjacent points are used and this is known as the first difference. If the data set still isn't stationary the difference between consecutive points in the differenced data set can used, this is the difference of the differences and is known as the second difference. Figure 5.2 shows the FTSE data set after the first differences have been taken. The resulting data set is now stationary.

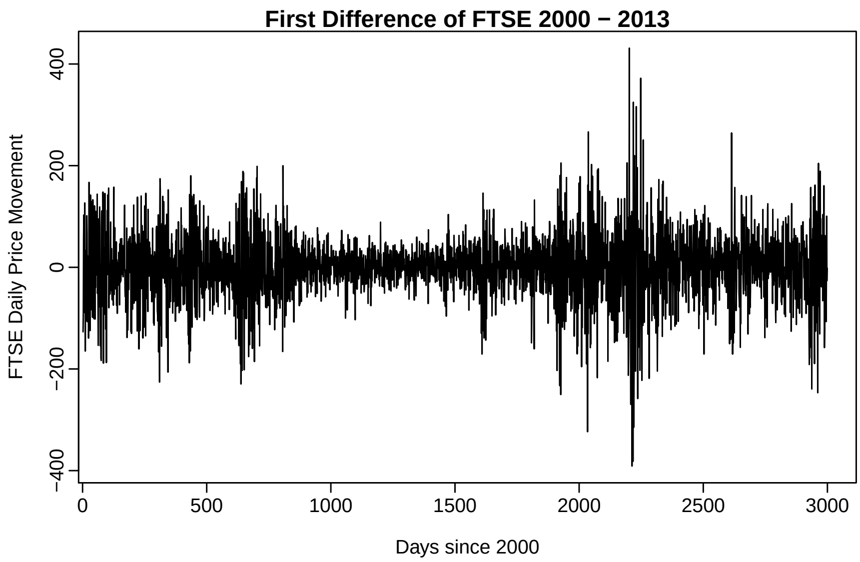

Figure 5.2: First difference of FTSE between 2000-13.

5.2.3 Examine ACF / PACF

With a stationary data set, the next stage is to investigate the auto-correlation and partial auto-correlation (ACF/PACF) plots in order to help in the model selection process (see section 2.2.5 for details of ACF and PACF). The ACF and PACF for the FTSE data set can be seen in Figures 5.3 and 5.4.

If ultimately the ARIMA model is of the form ARIMA(p,d,0) or ARIMA(0,d,q) then the ACF and PACF plots are useful in helping to define values for p or q. In the event that both p and q are positive, the ACF and PACF are not helpful in deducing the values for p and q. An ARIMA(p,d,0) model may be appropriate if the ACF and PACF plots of the stationary data exhibit an exponentially decaying pattern in the ACF and a large spike at lag p in PACF plot. Conversely an ARIMA(0,d,q) model may be appropriate if the PACF is decaying exponentially and there is there is a significant spike in the ACF plot at lag q. Considering the ACF and PACF plots in Figures 5.3 and 5.4, neither of the two patterns are observed and thus an ARIMA model where both p and q are positive is likely.

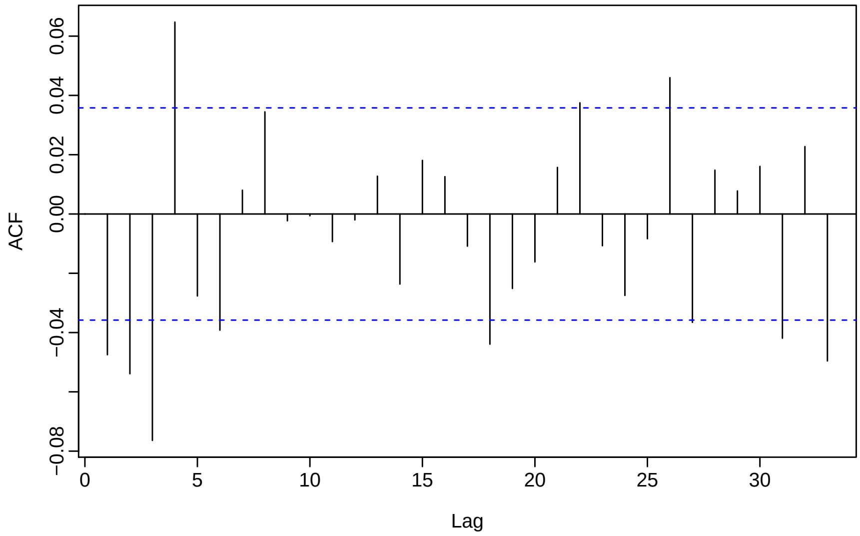

FIGURE 5.3: Auto-correlation plot of differenced data from FTSE 100 between 2000-2013.

'

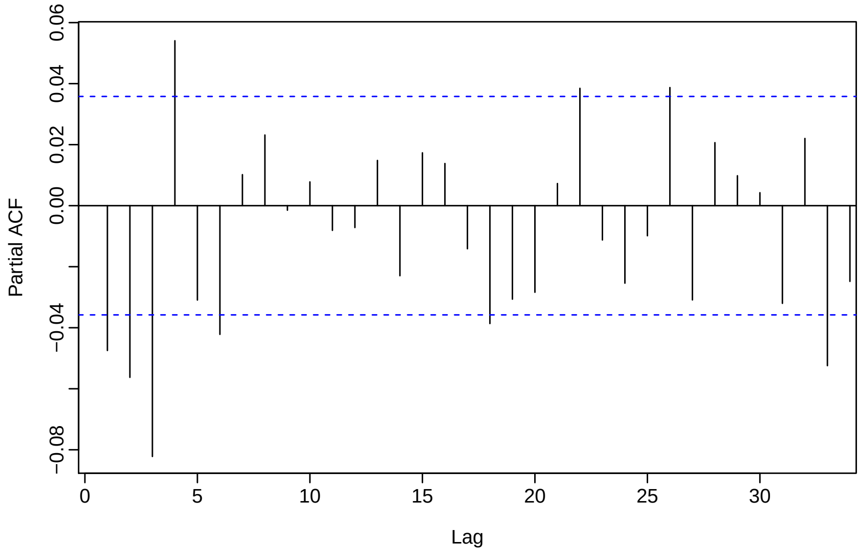

FIGURE 5.4: Partial auto-correlation plot of differenced data from FTSE 100 between 2000-2013.

5.2.4 Try the chosen model(s)

The next step is to try the chosen model along with a few viable alternatives. Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC) are useful for determining the optimum order of an ARIMA model, and are typically used as a measure of how well the model fits the data. AIC can be given by:

$$AIC = -2log(L) + 2(p + q + k + 1)$$

where:
L is the likelihood of the data and $k = 1$ if $c \neq 0$ and $k = 0$ if $c \neq 0$, the last term in parentheses is the number of parameters in the model.

For ARIMA models, the corrected AIC can be written as:

$$AIC_c = AIC + \frac{2(p + q + k + 1)(p + q + k + 2)}{T - p - q - k - 2}$$

The Bayesian Information Criterion can be expressed as:

$$BIC = AIC + log(T)(p + q + k + 1)$$

Table 5.1 shows the AIC, AICc and BIC accuracy measures for a selection of ARIMA models applied to the FTSE data set. On all three measures the ARIMA(2,1,3) model has the lowest value.

Table 5.1: AIC, AIc and BIC results from alternative ARIMA models.

Model	AIC	AICc	BIC
Arima(3,1,1)	39357.4	39357.4	39388.2
Arima(3,1,2)	39354.9	39354.9	39391.9
Arima(3,1,3)	39355.7	39355.7	39398.9
Arima(2,1,1)	39371.1	39371.1	39395.7
Arima(2,1,2)	39372.5	39372.5	39403.4
Arima(2,1,3)	39354.2	39354.2	39391.2

5.2.5 Model Residuals

A so-called residual is the difference between an observation and its forecast. In forecasting a time series, residuals are calculated from a one-step forecast. A one-step forecast is based on all observations from the start of the series until the previous observation to which the forecast applies to. Thus the number of data points used to calculate the one-step forecast increases as the forecast proceeds through the time series. An alternative is cross-sectional forecasting which uses all the points in the data set except the observation being predicted.

Knowledge of the residuals from the application of a model is important in establishing the validity of the model. There are two essential and two valuable properties that can be established by inspecting the model residuals. A good method of forecasting will produce a model in which the residuals are uncorrelated and have a zero mean. If a forecasting method doesn't comply with these two properties it can be improved upon. Correlation in residuals means that information is present in them that the model has missed and a non-zero mean is evidence of bias in the forecast. Adjusting for bias is straight forward, the mean value observed in the residuals can simply be added to all forecasts. Looking at Figure 5.5 it can be seen that the mean of the residuals is close to zero and this model doesn't have any bias. Figure 5.6 is the plot of the residuals of the ARIMA model applied to the FTSE data set. The lower order lags are all within the confidence boundaries and is indicative of a good model.

Two additional properties of the residuals that are desirable, though not necessary, are constant variance and normal distribution. If these two conditions are met, the calculation of the prediction interval in the forecast step is easier. From Figure 5.5 it can be seen that the residuals have relatively constant variance and from Figure 5.7, a histogram of the residuals, it can be seen that they are normally distributed.

Consideration of the ACF plots provides evidence for auto-correlation. However a more formal approach is to consider auto-correlation values together as a group as opposed to individually. The Box-Ljung portmanteau test is just one such approach and Table 5.2

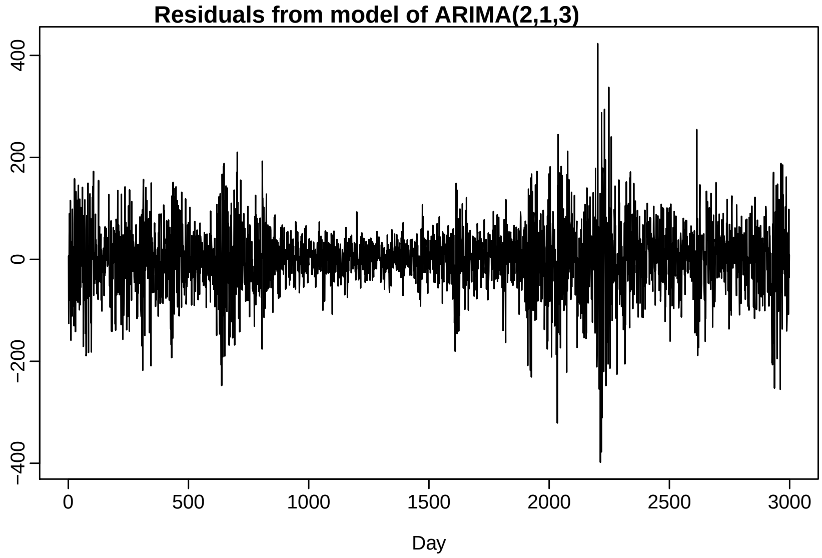

FIGURE 5.5: The residuals from applying the ARIMA(2,1,3) model to the FTSE data set.

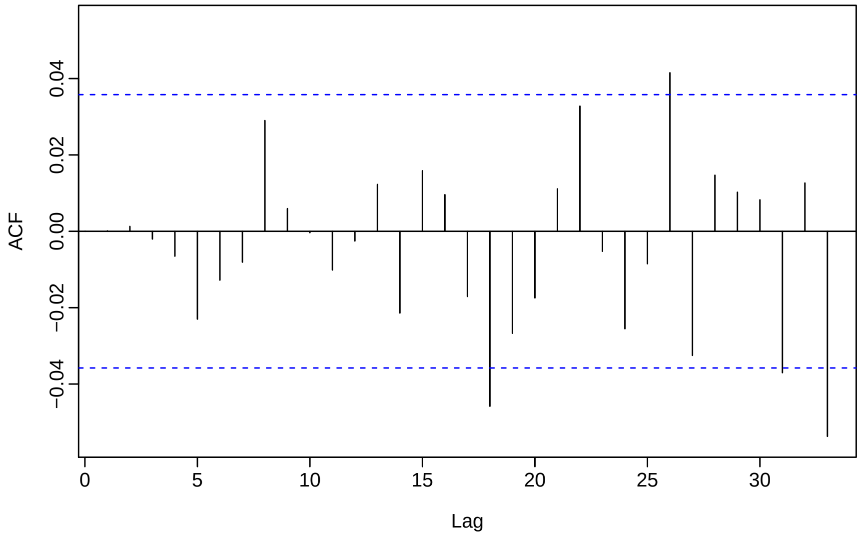

FIGURE 5.6: ACF plot of the residuals from applying the ARIMA(2,1,3) model to the FTSE data set.

lists the results of the Box-Ljung portmanteau test being applied to the residuals of the ARIMA(2,1,3) model. A large p-value is indicative of white noise and is the desirable situation for a good ARIMA model. Taking all the evidence together the ARIMA(2,1,3) model appears a good option for the FTSE data set.

TABLE 5.2: Box Ljung test.

	p-value	x-squared	df
ARIMA(2,1,3)	0.2328	20	24

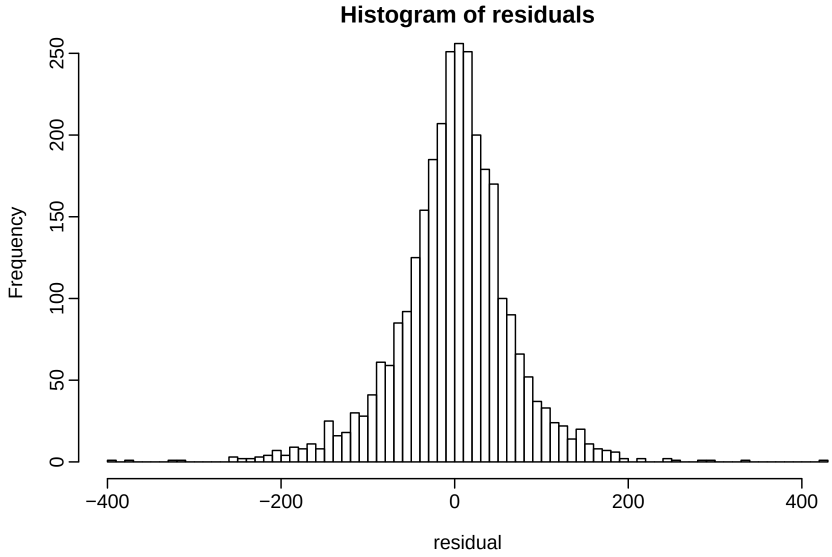

FIGURE 5.7: Histogram of the residuals from applying the ARIMA(2,1,3) model to the FTSE data set.

5.2.6 Calculate forecast

Finally, after developing a model that meets the previous criteria a forecast can be generated. Table 5.3 shows the one-step forecast produced when the ARIMA(2,1,3) model developed in the previous section is applied to the FTSE data set.

TABLE 5.3: FTSE 100 foecast.

Date	Open	High	Low	Close	Forecast
20/12/2013	6585	6617	6577	6607	6560
23/12/2013	6607	6679	6606	6679	6598
24/12/2013	6679	6712	6672	6694	6666
27/12/2013	6694	6754	6694	6751	6692
30/12/2013	6751	6768	6718	6731	6743
31/12/2013	6731	6757	6731	6749	6730

5.3 Automatic Generation of ARIMA Models

As explained previously the automatic ARIMA modelling algorithm in the R forecast package, auto.arima(), automates steps 3 to 5 in the general steps used in the modelling process as outlined in section 5.1. The function uses a variation of the Hyndman and Khandakar algorithm which obtains an ARIMA model by the minimisation of the AICc and combination with unit root tests. KPSS tests are used to establish the number of differences, d, required to get a stationary time series. The p and q values are then obtained by choosing the model that minimises the AICc for the differenced data.

The results of passing the indice data sets to the auto.arima() function can be seen in Table 6.1. For the FTSE data set the automatic procedure selects the ARIMA(2,1,3) as being the most appropriate, which matches the conclusion of the work from the manual model selection process described earlier in section 5.2.

TABLE 5.4: Arima models from national indices.

Market	Arima Model
Dax	ARIMA(3,1,3)
CAC	ARIMA(2,1,3)
FTSE	ARIMA(2,1,3)
Dow	ARIMA(1,1,2)
Nikkei	ARIMA(2,1,3)
AORD	ARIMA(1,1,0)

5.4 Trading the ARIMA Models

Having developed forecasts based on ARIMA models these can be passed into a trading system. Two ideas are presented here, in the first the previous closing price is compared against the prediction and if it is lower than the forecast a long trade is entered. This first system will be referred to as System 1. In the second algorithm the current forecast is compared with the previous prediction. When the previous forecast value is lower than the current prediction the system trades long. This algorithm will be referred to as System 2.

5.4.1 System 1 - Close Price vs Forecast

Using the ARIMA models listed in Table 6.1 a series of amended data sets were generated by applying the models to the national indice data sets used throughout this study. The amended data sets contained the original Date, Open, High, Low and Close attributes plus a new one called Forecast, in a similar manner to the data seen in Table 5.3. Table 5.5 are results produced from passing the newly generated data sets to the algorithm listed in Appendix A section A.2.1. This system uses the relative position of the close price and the forecast to determine the direction of the trade. If the forecast is higher than the close a long trade is made and when the prediction is lower than the close price a short trade is made.

Table 5.5: Auto.arima models passed to the System 1 trading algorithm

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	-644	-1881	50	-3	41	-7
CAC	1555	850	59	6	51	3
FTSE	531	-708	53	2	46	-2
Dow	3130	-1766	58	14	48	-6
Nikkei	41	-1157	48	0	45	-5
AORD	679	-204	55	3	49	-1

5.4.2 System 2 - Forecast vs Previous Forecast

Table 5.6 lists the results from passing the amended indice data sets with the forecasts generated from the auto.arima() function, described in the previous section, to the System 2 algorithm. The R code of this system can be seen in Appendix A section A.2.2. System 2 uses the relative values of the forecasts themselves to decide which direction to trade. If the prediction is higher than the previous day's prediction a long trade is initiated and in the opposite circumstances when the previous forecast is higher than the current forecast a short trade is made.

Table 5.6: Auto.arima models passed to the System 2 trading algorithm

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	2226	989	57	8	49	4
CAC	-76	-781	50	0	43	-3
FTSE	173	-1066	54	1	47	-4
Dow	2910	-1985	53	10	43	-9
Nikkei	-3269	-4467	50	-14	46	-22
AORD	247	-635	51	1	45	-2

5.5 Hybrid ARIMA Models

A hybrid ARIMA model is one in which the moving averages of a stationary data set (possibly a non-stationary data set that has been differenced) are combined with data mining learners other than regression. Possible learners include k nearest neighbour algorithms, artificial neural networks and support vector machines. RapidMiner, the open source data mining tool is a powerful solution for building hybrid ARIMA models. Figure 5.8 shows the RapidMiner process used to generate hybrid ARIMA models. The Validation operator in the model below can hold a variety of learners depending upon the task and data types involved. The various components in Figure 5.8 are as follows:

- Read CSV - reads in the appropriate data set.

- Select Attribute (1) - selects the attribute that will be processed in the following steps.
- Rename - renames the attribute selected in Select Attribute (1) to “attr1” which is then used in the est of the steps. This component is used to make it easy to change the attribute without having to rename all the subsequent steps.
- Moving Average - calculates a moving average of the time series (see section 2.2.1.1 for details.) This provides the q in ARIMA(p,d,q) models.
- Differentiate - calculates the difference in the time series and provides the d in ARIMA(p,d,q) models.
- Lag - creates lag variables which are values of the attribute (the attribute itself, the moving average or the difference value) at earlier points in the time series.
- Select Attribute (2) - selects the attributes that will be passed to the validation block. Attributes regarding today’s values are removed because we are building a model to calculate them and don’t want to “peak” at them before the model is built.
- Set Role - sets an attribute as the label to be predicted.

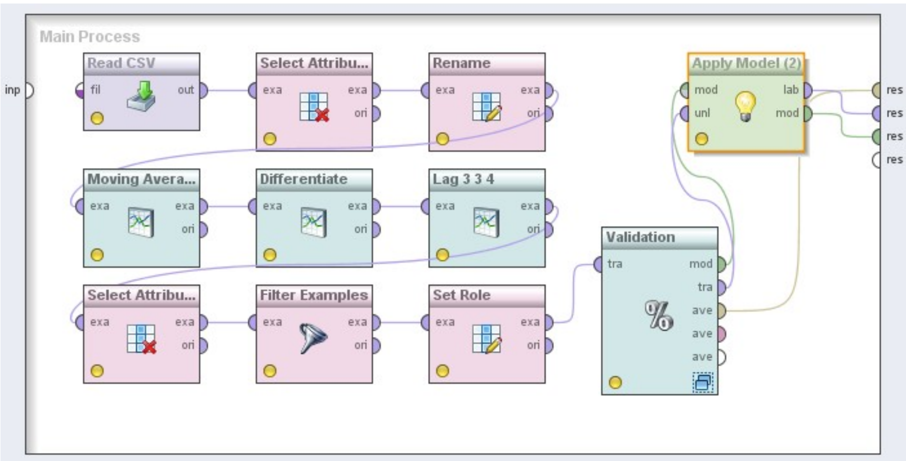

FIGURE 5.8: Rapid Miner Hybrid ARIMA Process.

Figure 5.9 shows the cross-validation operator of the hybrid ARIMA Rapid miner process. This operator can hold alternative learners other than the standard regression operator found in ARIMA models. In the diagram there is an Artificial Neural Network (ANN) operator shown, other options include k-Nearest Neighbour (k-NN) and Support Vector Machine (SVM) operators.

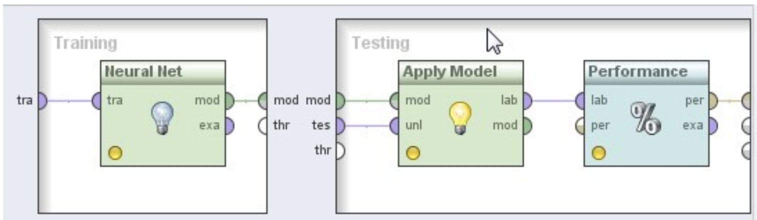

FIGURE 5.9: Rapid Miner cross-validation operator of Hybrid ARIMA process.

5.6 Predicting Closing Price

As mentioned previously ARIMA and hybrid ARIMA models were used to predict either the value of the one-step ahead close price or the binary value of whether the market moved up or down. In this section the ability of hybrid ARIMA models to forecast the future price of financial markets (as opposed to the general direction up or down) is explored.

5.6.1 ARIMA/Artificial Neural Networks (ANN)

An ARIMA/ANN method was used to generate predictions for the closing price of the indice data sets under study. For each data set applying the hybrid model produces a new one-step forecast attribute which can be used in the System 1 and 2 algorithms previously introduced in section 5.4. Table 5.7 are the results generated by passing the output of the ARIMA/ANN models to trading System 1 (which compares the previous closing price with the current forecast).

TODO

TABLE 5.7: Predicting Close Price - Arima/ANN predictions passed to System 1

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	-1305	325	52	0	100	325
CAC	-1018	5295	51	0	52	5
FTSE	1987	1408	58	32	49	0
Dow	11685	1904	53	4	48	4
Nikkei	373	18365	46	4	51	6
AORD	2171	1151	53	3	48	0

Table 5.8 are the results of passing the output of the ARIMA/ANN models to trading System 2, which compares the value of the current forecast with the previous one.

TABLE 5.8: Predicting Close Price - Arima/ANN predictions passed to System 2

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	193	1823	52	0	47	1
CAC	-5544	769	48	-3	48	0
FTSE	-3565	-4144	50	-2	47	-2
Dow	-3417	-13198	51	-2	44	-8
Nikkei	-18852	-861	47	-11	50	-1
AORD	-101	-1121	52	0	47	-1

5.6.2 ARIMA/k-Nearest Neighbour (k-NN)

An ARIMA/k-NN method was used to generate predictions for the closing price of the indice data sets. Table 5.9 shows the results of passing data sets containing forecasts generated with hybrid ARIMA/k-NN to trading System 1.

TABLE 5.9: Predicting Close Price - Arima/k-NN predictions passed to System 1.

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	8270	9900	56	4	52	6
CAC	6284	12597	54	3	55	7
FTSE	17605	17026	58	9	56	10
Dow	30330	20549	59	17	53	12
Nikkei	15374	33366	54	9	57	20
AORD	7658	6638	57	4	53	4

Table 5.10 shows the results of passing data sets containing forecasts generated with hybrid ARIMA/k-NN to trading System 2.

TABLE 5.10: Predicting Close Price - Arima/knn predictions passed to System 2

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	6131	7750	54	3	50	5
CAC	-567	5746	50	0	50	3
FTSE	2571	1992	52	1	49	1
Dow	8466	-1269	54	4	48	-1
Nikkei	-5066	12577	49	-3	52	8
AORD	3153	2013	54	2	50	1

5.7 Predicting Up or Down - Categorical Label

In this section the ability of hybrid ARIMA models to forecast whether a financial market will rise or fall is investigated. A categorical attribute taking values "U" and "D", representing whether the market moved up ("U") or down ("D") was introduced into the indice data sets depending upon which way the market moved that day. Hybrid ARIMA models were used to forecast this categorical label.

5.7.1 ARIMA/Artificial Neural Networks (ANN)

The R code for a trading system using the forecasts from a hybrid model can be seen in Appendix A section A.2.3. The algorithm simply uses the prediction from the hybrid ARIMA model ("U" or "D") to decide whether to trade long or short. Table 5.11 lists the results from using this hybrid ARIMA/ANN model to make the forecasts.

TABLE 5.11: Predicting UpDn CAT - Arima/ANN predictions passed to System 4

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	49	1714	56	2	48	0
CAC	0	6426	NaN	NaN	50	2
FTSE	7399	6806	55	5	51	3
Dow	12434	2711	56	8	49	1
Nikkei	-14054	3771	49	-4	56	24
AORD	3938	2978	53	1	59	13

5.7.2 ARIMA/k-Nearest Neighbour (k-NN)

An Arima/k-NN model was also employed in an attempt to predict the categorical label indicating whether the financial markets would move up or down. The forecasts produced from these hybrid models were also applied to the trading algorithms listed in A section A.2.3. Table 5.12 lists the results from this combination.

TABLE 5.12: Predicting UpDn CAT - Arima/k-NN predictions passed to System 4

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	15692	17357	61	8	60	12
CAC	10161	16587	60	6	59	9
FTSE	15553	14960	60	8	60	10
Dow	30347	20624	62	14	60	15
Nikkei	27206	45031	60	18	60	24
AORD	9711	8751	60	5	59	6

As the results from Table 5.12 were good, the algorithm was re-run but this time a stop loss was introduced. A stop loss of 100 points was applied to all the markets and the amended results can be seen in Table 5.13. In a similar manner as encountered previously, the use of the stop loss was beneficial for all the markets except the Dow in which case it had a large detrimental affect.

Table 5.13: Predicting UpDn CAT - Arima/k-NN predictions passed to System 4 - SLoss

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	15767	17826	60	8	59	13
CAC	10524	17378	59	6	59	9
FTSE	16562	16020	59	8	59	11
Dow	7152	-671	52	3	48	0
Nikkei	29132	48387	54	19	56	25
AORD	9743	8978	60	5	59	6

5.7.3 ARIMA/Support Vector Machine (SVN)

ARIMA was also married with a SVM learner in order to predict the categorical value, "U" or "D". Table 5.14 lists the results of passing forecasts made using this combination to the trading algorithm listed in A section A.2.3.

Table 5.14: Predicting UpDn CAT - Arima/SVm predictions passed to System 4

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	-3817	-2152	53	-2	49	-1
CAC	1044	7470	53	1	53	4
FTSE	6944	6351	54	4	51	3
Dow	4659	-5065	54	3	48	-3
Nikkei	2881	20706	57	28	52	6
AORD	972	0	52	0	NaN	NaN

5.8 Predicting Up or Down - Continuous Label

An alternative to using a categorical variable represented by "U" or "D" to indicate if the market rose of fell is to use the numeric range 0 to 1. Here 0 is used to indicate the market fell and 1 to indicate that it increased in value. Thus an additional attribute was introduced into the national indice data sets which took the value of 0 or 1.

5.8.1 ARIMA/Artificial Neural Networks (ANN)

An ARIMA/ANN model was employed in an attempt to predict the value of 0 or 1, which indicates the directional movement of the market. Table 5.15 are the results of passing the indice data sets augmented with the ARIMA/ANN forecasts to the trading algorithm listed in Appendix A section A.2.5.

Table 5.15: Predicting UpDn 01 - Arima/ANN predictions passed to System 3

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	449	2114	100	224	48	1
CAC	-5761	665	50	-2	100	111
FTSE	747	154	52	0	100	51
Dow	2728	-6995	63	52	47	-2
Nikkei	-17151	674	49	-5	100	169
AORD	870	-90	52	0	60	-18

5.8.2 ARIMA/k-Nearest Neighbour (k-NN)

An ARIMA/k-NN model was used to make forecasts for the continuous variable that represents whether the market will move up or down. The output of the model is a value between 0 and 1. The trading algorithm that uses this prediction can be seen in Appendix A section A.2.4 and the results generated in Table 5.16. The trading algorithm looks at the forecast value and trades long if the values are over 0.5 and short if they are below this value.

Table 5.16: Predicting UpDn 01 - Arima/knn predictions passed to System 3

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	15692	17357	61	8	60	12
CAC	10161	16587	60	6	59	9
FTSE	15553	14960	60	8	60	10
Dow	30347	20624	62	14	60	15
Nikkei	27206	45031	60	18	60	24
AORD	9711	8751	60	5	59	6

Chapter 6

Analysis

6.1 Introduction

In chapters 4 and 5 a wide variety of analytical techniques were applied to a variety of time series data sets. In Chapter 4 a range of trading algorithms were developed based on technical analysis indicators. The intention was to automate the decision of whether to buy or sell a market based on the value of the indicator. For comparison purposes, two simple so called ”naive” systems were explored to set a base line against which the technical analysis indicators could be compared. The technical indicators were grouped together in their general area of applicability, namely trend detection indicators, reversal, momentum and candlestick indicators.

Chapter 5 continued the exploration of financial time series through the use of exponential smoothing, ARIMA and hybrid ARIMA techniques. The generated models were used to create one-step forecasts which were then combined with the original data set. These data sets were then fed into trading algorithms which used the forecast values to make trading decisions. Again a series of simple forecast techniques were used as a baseline against which the trading system algorithms could be compared.

6.2 Technical Analysis

Initially two simple, naive systems were explored to set a baseline for further analysis. These systems were the Naive Long System which mirrors a buy and hold strategy and a Naive Follow Previous system which simply repeats the previous days market direction.

6.2.1 Naive Systems

The first base line system tried was the Naive Long system in which a market buy is placed each day and is similar to the so-called ”Buy and Hold” technique. The assumption here is that the market rises over time and if an investor simply holds a security it will eventually generate a profit. The total profit is simply the price at the start, in this case the data set started in 2000, subtracted from the price at the end of the period which in this case was the end of 2013.

The first iteration of the algorithm placed a buy at the start of the trading session and closed it at the end and thus the system was out of the market overnight. This resulted in significant discrepancies from the returns expected from a buy and hold system. With a buy and hold system the returns would have been as follows:

- Dax: +2591
- CAC:-1774
- FTSE: -181
- Dow: +11501
- Nikkei: -2649
- AORD: +2201

From simply trading long during market hours the Dax generated a loss as opposed to the 2591 profit expected, likewise the CAC showed a much larger loss than expected and the Nikkei resulted in a large loss when a small loss was expected. The Dow, FTSE and AORD were similar to the expected values. Changing the algorithm such that the trades ran close to close and covered the full 24 hour period resulted in system results that matched the expected results from a buy and hold approach. Clearly the discrepancies from the first algorithm were due to the relative amounts the markets moved during the day as opposed to during the “out of hours” trading. There is a slight bias for the markets to move upwards overnight and over the course of the study (14 years) adds up to significant values.

The second naive system was termed ”Naive Follow Previous” and simply places a trade today consistent with the market direction from the previous day. This idea produced very poor results, with every market losing money. Clearly if the trades were reversed so that the algorithm traded in the opposite direction to the previous day the exact opposite results would have occurred (in a real scenario this wouldn’t be true because of trading charges). From these results it can be concluded that the markets have a tendency to ”flip flop” and reverse back on themselves, and the phenomena of market reverses is well understood.

6.2.2 Trend Detection

The first group of the technical analysis indicators studied were the trend detection indicators. Identification of trend direction and strength is very important in the world of financial trading and one of the most widely encountered phrases is ”the trend is your friend”, as most authorities advocate trading in the direction of the trend. (In fact on a recent webinar it was claimed that 80% of all money made is made trading in the direction of the trend.) Well known indicators that purport to assist the trader in identifying trends are the simple moving average (SMA), the moving average convergence/divergence indicator (MACD) and the Aroon indicator.

The use of simple moving average is wide-spread in the financial markets. Market participants track moving averages or even more than one and make a decision which way to trade based on the position of the current price relative to it. Popular values to use in the SMA are 25, 50 and 200. The results of a trading based on SMA is presented in Table 4.5. The algorithm places a buy trade if the current price is above the SMA and a sell trade if it is below it. The results from this algorithm an be seen in

Results are mixed with some markets producing positive results when trades are directed by the SMA values and some result in losses. The German Dax produces positive results across all the SMA values with values from trading short (predicting the market will decline doing best). The French CAC displays different results in that all the SMA values produce negative results in trying to predict long trades but positive results when trying to predict short results. The UK’s FTSE 100 displays different behaviour again, producing negative results across the board. The Dow produces a different set of results again, trades on the long side produce a profit whereas trading short results in losses. The Japanese Nikkei exhibits similar results to the CAC in that short trades are profitable whereas long trades aren’t and finally the Australian AORD is similar to the DAX in producing positive results across the board.

b. applied SL – winners and losers, need to assess . . . . Do we want the SLoss discussion moved here?

b. MACD – no SL applied here MACD can generally be used 2 ways, as a trend detection indicator and as an over-bought / over-sold indicator in which case traders use it to identify potential market reversals. In this section the indicator was used as a trend detector and the results from a system based on the MACD indicator can be seen in Table **??**. The algorithm trades long when the value of MACD is greater than the value of the signal line, see section **??** for more details of the implementation of the MACD indicator. (Is it in the App?). The results are not very impressive, only the Nikkei producing reasonable profits – though wouldn't beat the flip flop naive system.

c. Aroon – alone and SL -> sort ref out ... The final trend detection indicator examined was Aroon. This indicator measures the time since the previous high or low within a certain time window. The algorithms presented here used a time window of 20. If today was highest price in the last 20 days trading the indicator would take a value of 100 and for each day that proceeds that doesn't make a new high the indicator falls by 5 (100 divided by the lag period which is 20). Thus if the highest price was four days ago the AroonUp value would be 80. The opposite situation occurs with regard to the low price. A value of 70 or above for the AroonUp is indicative of a upward trending market and likewise a value of 70 and above for AroonDn suggests a falling market.

Naive Reversed PL: CAC - 7800 (L) -> FTSE - 4000 (LS) Dow - 6000/15800 Nik - 20500(L)

The results from an algorithm using these concepts can be seen in Table **??**. Overall the results are encouraging. The Dax, FTSE, Dow, AORD produce positive returns for both Long and Short. The CAC and Nikkei are positive short trading.

The use of a Stop Loss improves the returns from all markets except the Dow. One again using a SL with the Dow shows very marked negative impacts on profits. These results can be seen in Table **??**

MACD – no SL applied here The results from a system based on the moving average convergence/divergence (MACD) indicator can be seen in Table **??**. The results are not very impressive, only the Nikkei producing reasonable profits – though wouldn't beat the flip flop system.

Aroon – alone and SL Alone – promising results on short side, long side CAC and Nik show loss With SL – results improve, except Dow -> results get worse!! See diff table.

6.2.3 Market Reversal Indicators

– no SL used In this section two indicators that purport to assist in identifying market reversals are examined.

1. Parabolic Stop-and-Reverse (SAR) The first market reversal indicator used was the Parabolic Stop-and-Reverse (SAR) an indicator initially developed for traders who were

always in the market either long or short. The SAR was used to judge when the position should be reversed from long to short or vice versa. The trading algorithm using the SAR trades each day (i.e opens a trade at the start of the trading session and closes it out at the end) and makes a decision regarding the direction of the trade based on the SAR indicator. If the market opening is above the SAR a long trade is initiated and vice versa if the market is below the SAR value.

The results from the trading system based on the SAR can be seen in Table **??** and are very poor. Only the Nikkei trading short produce reasonable results, but these are much worse than the naive flip flop method of section **??**.

2. MACD As previously mentioned the MACD indicator can be used as a market reversal indicator. Once the MACD value reaches its extreme values the market is considered over-bought or over-sold. The trading algorithm using this concept expects a market reversal once the MACD crosses above the 85% quantile (of the MACD range) or the 15% quantile. Short trades are initiated once the MACD crosses above the 85% quantile value and short trades once it has passed below the 15% quantile.

The results from this trading system can be seen in Table **??** and are very unimpressive.

V modest results, but we are deffo not in trend at this point ... Cross-ref to a trend indicator?

6.2.4 Momentum Indicators

A third type of technical indicators are the momentum indicators, which are related to the trend detection indicators. Two such indicators are studied here, the stochastic and Rate of Change (ROC). 1. Stoch – plus SL The stochastic oscillator is one of the oldest and most widely used of the technical indicators. Essentially it measures the percentage position the current close in relation to the high low range of period of interest. Thus it has conceptual similarities to the Aroon indicator. The stochastic is usually represented by two lines %K which is the position of the price within this high low envelope described above, and %D a moving average of %K (see Chapter 4. Appendix **??** for more details). Long trades are initiated when %K is above %D and vice versa for short trades. Results from an algorithm implementing these ideas can be seen in Table **??**. the results are poor being significantly worse than the naive flip flop system.

NB – written “of interest” twice ... also ”This is of interest as ... ” , within this high”

Use of the SL improves things - except the Dow as usual.

1. Stoch – plus SL NB – written “of interest” twice ... a. Alone – results v. modest

2. ROC - Needs finishing in Chp 4 - Roc1 and 2 – what's Diff ? let's go with ROC2 ...

ROC1 - Mkt$Long < -ifelse(Mkt$prevROC $<$ lw,Mkt$Close - Mkt$Open,NA)

ROC2 - Mkt$Long < -ifelse(Mkt$prevROC $>$ 0,Mkt$Close - Mkt$Open,NA)

The second momentum indicators is the Rate Of Change (ROC) indicator, and this is simply the difference between the current price and a price a certain number of days previously. If this value is positive the market is considered to be trending up and the larger the value the greater the trending momentum. The results from an algorithm using these ideas is presented in Table **??**. Most of the results are negative and in fact doing the opposite would produce reasonable results.

6.2.5 Break-out systems

The fourth section explored the idea of trade signals being generated by a particular value from the previous day, so-called breakout systems. Two particular values are used as the trigger price for a trade, the previous day's high/low or the 90% quantile of the minor move (see section x).

2. 90The second break-out stem used the minor fluctuation 90% quantile value as the level to buy or sell. Once the market moved this amount above - Good results, not as good as b/out system ... Dow poor again ... Oz v. good given price of mkt, almost as good as b/out sys

2. 90- Good results, not as good as b/out system ... Dow poor again ... Oz v. good given price of mkt, almost as good as b/out sys ...

6.2.6 Candlestick Patterns

1. Hammer 2. Hammer plus aroon

3. Engulfing 4. Engulfing plus aroon To Do – name of tables are the same ...

5. Doji

Summary SL – sma, aroon and stoch – why these? Add to b/out and naïve systems

6.3 Time Series Analysis

ARIMA and hybrid ARIMA models were used to generate forecasts of the closing prices and the more general situation of whether the market would rise or fall. In modelling the more general situation of market direction a categorical and a continuous label was employed. The categorical label used "U" to represented occasions when the market prices increased and "D" for when it decreased in value. Alternatively the values 1 and 0 were also used to represent up and down respectively. The primary difference between the two labels was in the values returned from the hybrid ARIMA models. When using 1 and 0 for the class label the models return a value in the range of 1 to 0, whereas for the categorical value there was only the choice of the two values.

6.3.1 Automatically generated ARIMA Models

The auto.arima() function of the R forecast package was used to generate ARIMA models for the national indice data sets used in this study. For convenience the models picked are listed in Table 6.1.

Table 6.1: Arima models from national indices.

Market	Arima Model
Dax	ARIMA(3,1,3)
CAC	ARIMA(2,1,3)
FTSE	ARIMA(2,1,3)
Dow	ARIMA(1,1,2)
Nikkei	ARIMA(2,1,3)
AORD	ARIMA(1,1,0)

The one-step forecasts generated from these models were then used in two trading systems. In the first algorithm the decision to trade long or short was dependant upon on the relative values of the previous close price and the forecast. If the forecast was was higher than the close price a long trade wa entered in the expectation that the market would rise towards the prediction. The opposite situation was expected for when the forecast was lower than the close price. The second trading algorithm used the relative values of the predictions themselves in order to decide whether to trade long or short. If the current forecast was higher than the previous one a long trade was made and vice versa.

The results from both systems were poor. The difference in mean PL per trade between the first system based on the auto.arima models (previous close in comparison to forecast) and the mean PL for the Naive Reversing system from section 4.2.2 Chapter 4 can be seen in Table 6.2. Most of the results are worse than the naive baseline system except for the French CAC and US Dow when trading long.

Table 6.2: Mean Long/Short PL from Naive Reverse system subtracted from PL generated by auto.arima models

Mkt	Diff in Mean Long PL	Diff in Mean Short PL
Dax	-4	-9
CAC	5	-1
FTSE	-1	-4
Dow	4	-9
Nikkei	-1	-17
AORD	2	-1

6.3.2 ARIMA Hybrids - Predicting Closing Price

Hybrid ARIMA models in which Artificial Neural Networks and k-Nearest Neighbour algorithms were used instead of regression in the ARIMA algorithm were used to predict the closing prices of financial markets.

6.3.2.1 ARIMA/Artificial Neural Networks (ANN)

Overall the use of the models generated from hybrid ARIMA/ANN algorithms to create trading systems was not very successful. The results from passing the indice data sets augmented with a forecast attribute generated by the hybrid ARIMA models can be seen in Tables 5.7 and 5.8 of Chapter 5. System 1 compares the price of the forecast with the price of the previous and in the event that the prediction is higher than the previous closing price a long trade is entered. The opposite is true when the forecast is lower than the closing price and a short trade is made. System 2 is similar but compares the forecast with the last forecast. In the event that the current prediction is greater than the previous one a long trade is initiated.

Considering the results in Tables 5.7 and 5.8 it can be seen that System 1 outperforms System 2 quite markedly. Even so, the results are quite modest across most of the indices and especially poor for the Dax. The results prove inferior to the baseline Naive Reversing System introduced in 4.2.2 Chapter 4 as shown in Table 6.3.

6.3.2.2 ARIMA/k-Nearest Neighbour (k-NN)

An alternative to the ARIMA/ANN methodology is to replace ANN with a k-Nearest Neighbour learner. Results from using the forecasts generated in the two trading systems introduced in section 5.4 can be seen in Tables 5.9 and 5.10. The results from System 1 are very good and exceed the baseline Naive Reversing approach. Table 6.4 lists the difference in results between those generated with System 1 and the ARIMA/k-NN

TABLE 6.3: Results from a trading system based on forecasts of closing price generated by the Arima/ANN model compared to baseline Naive Reversing methodology.

Mkt	Diff in Mean Long PL	Diff in Mean Short PL
Dax	-1	323
CAC	-1	1
FTSE	29	-2
Dow	-6	1
Nikkei	3	-6
AORD	2	0

models and the baseline system. In all cases the hybrid ARIMA model produces superior results.

TABLE 6.4: Predicting Close Price - Arima/k-NN predictions passed to System 1 diff.

Mkt	Diff in Mean Long PL	Diff in Mean Short PL
Dax	3	4
CAC	2	3
FTSE	6	8
Dow	7	9
Nikkei	8	8
AORD	3	4

6.3.3 ARIMA Hybrids - Predicting Up Down with Categorical Label

An alternative to forecasting the closing price of a financial market is to predict the general direction it will move in the short term either up or down. To this end an additional categorical label to indicate whether the market increased or fell in value over the course of the day was introduced into the data sets. This new attribute had the value “U” if the market increased and “D” if it decreased. Hybrid ARIMA models were then employed to predict this label.

6.3.3.1 ARIMA/Artificial Neural Networks (ANN)

The first methodology employed was to combine ARIMA with Artificial Neural Networks (ANN) in order to generate a forecast of the categorical label that indicated whether the market increased in value or fell over the course of the day. Once the forecast was generated and added to the data set in the form of a new attribute it was passed to a trading algorithm which based the decision whether to trade long or short on the forecast generated. The R code for the trading algorithm can be seen in Appendix A section

A.2.3 and the results generated in Table 5.11. Overall the results were poor and inferior to the baseline system used for comparison.

6.3.3.2 ARIMA/k-Nearest Neighbour (k-NN)

Replacing the ANN learner from the previous section with a k-NN method resulted in far better results. Table 5.12 lists the results of passing the forecasts from this combination to the trading algorithm in Appendix A section A.2.3. Across all the data sets large positive results are recorded. Table 6.5 lists the difference in results between using this hybrid ARIMA approach and the usual baseline returns. Clearly this methodology produces superior results. Using a stop loss with this system increases the returns from all the markets except the US Dow and these results are listed in Table 5.13.

Table 6.5: Predicting UpDn CAT - Arima/k-NN predictions passed to System 4 - diff.

Mkt	Diff in Mean Long PL	Diff in Mean Short PL
Dax	7	10
CAC	5	5
FTSE	5	8
Dow	4	12
Nikkei	17	12
AORD	4	6

6.3.4 ARIMA Hybrids - Predicting Up Down with Numeric Label

The final approach adopted was to represent whether a financial market moved up or down by using 1 to signify that the market moved up and 0 that it moved down. The implications of using a numeric value is that the forecasts were in a range between these two values. In such circumstances the trading algorithms picked long trades when the prediction were in the upper half of the range.

6.3.4.1 ARIMA/Artificial Neural Networks (ANN)

An hybrid approach using ARIMA and ANN was used to make one-step forecasts for the future direction of the market, either up (1) or down (0). Table 5.15 in Chapter 5 lists the results of passing the indice data sets augmented with the ARIMA/ANN forecasts to the trading algorithm listed in Appendix A section A.2.5. Overall the results are poor, especially for the Japanese Nikkei trading long and inferior to the Naive Reversing system that is used as a comparative baseline.

6.3.4.2 ARIMA/k-Nearest Neighbour (k-NN)

Finally a k-Nearest Neighbour (k-NN) learner was used instead of the ANN algorithm. The models were used to calculate the one-step ahead forecast as represented by a numeric value between 0 and 1. The forecast was added to the data sets and passed to the trading listed in Appendix A section A.2.4. In common with other forecast using the hybrid k-NN approach good results were obtained and these can be seen in Table 5.16 of Chapter 5. The results are much better than the baseline system which simply trades based on doing the opposite of what happened yesterday. Table 6.6 lists the difference in terms of performance between the two systems and is simply the values in Table 4.4 from Chapter 4 subtracted from the results in Table 5.16 from Chapter 5.

TABLE 6.6: Predicting UpDn 01 - Arima/k-NN predictions passed to System 3 res diff

Mkt	LongPL	ShortPL	L Win %	Av L PL	S Win %	Av S PL
Dax	15692	17357	61	8	60	12
CAC	10161	16587	60	6	59	9
FTSE	15553	14960	60	8	60	10
Dow	30347	20624	62	14	60	15
Nikkei	27206	45031	60	18	60	24
AORD	9711	8751	60	5	59	6

6.4 Conclusion

TA - not much cop -> b/out good

6.5 Future Work

candlestick systems -> price 2,,3,4 days ahead? combining systems k-NN seems promising additional markets

Chapter 7

To Do - Not for Thesis

- - remove date from opening page ...
- - consistency - long / short
- - consistency - Nik, FTSE, Oz

7.1 Chp2

- R Code for graphs in own Script
- 2.14 and 2.15 correlogrmams - check and sort out...
- - 2.2.5 - sort out chapter2
- -> stationary series - coghlan: no seasonality ot trend ... further additive -> without trend ...

7.2 Chp3

- R Code for graphs in own Script
- section 1.1.2 - Add refs for TA methods mentioned.
- table check list - cap top, llcccc etc, big cap, sm cap
- Lane Lane (1986) and williams Williams (2011) Williams (1989) - Chp5c - stoch

7.3 Chp4

- aroon ref,
- Candlestick details - move to Appendix for consistency?
- candlestick systems -> price 2,,3,4 days ahead?

7.4 Chp5

- - reftodo-Examine ACF / PACF - - interpret acf graph
- - 5.6.1 - results?
- - 5.7.1 - results?

7.5 App A

- - app A - sub titles

“participant

Appendix A

R Code

A.1 Chapter 4

The R code used to generate the results and tables in Chapter 4 is shown in listing A.1.1. This is followed by the individual files containing the algorithms used in the chapter.

A.1.1 Chapter 4 Results Generation

```
# Chapter 4
setwd("D:/Allan/DropBox/MSc/Dissertation/Thesis/RCode")

# Housekeeping
library(xtable)
library(TTR)
library(candlesticks)

source("../RCode//Utils.R")
source("../RCode//NaiveLongSystem.R")
source("../RCode//NaiveLongSystem2.R")
source("../RCode//NaiveFollowPrev.R")
source("../RCode//SMA_sys.R")
source("../RCode//MACD_XO.R")
source("../RCode//Aroon.R")
source("../RCode//SAR.R")
source("../RCode//Stoch.R")
source("../RCode//ROC.R")
source("../RCode//ROC2.R")
source("../RCode//MACD_OB.R")
source("../RCode//Bout_sys.R")
source("../RCode//Quant90_sys.R")
source("../RCode//Candle_Hammer.R")
source("../RCode//Candle_Hammer_aroon.R")
source("../RCode//Candle_Engulf.R")
source("../RCode//Candle_Engulf_aroon.R")
```

```
source("../RCode//Candle_Doji_aroon.R")

fil <- c("../Data/Dax_2000_d.csv",
         "../Data/CAC_2000_d.csv",
         "../Data/F100_2000_d.csv",
         "../Data/Dow_2000_d.csv",
         "../Data/N225_2000_d.csv",
         "../Data/Oz_2000.csv")
#nm <- c("Dax", "CAC", "FTSE", "Dow", "Nikkei", "AORD")
df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11)) # to hold results

std6 <- c(1,3,4,5,7,8,10)

#s <- read.csv('../Data/Dax_2000_d.csv')

# ----------------------------------------
# ---------  1. Naive Long (Sub Chapter) --------

run_NaiveLongSystem <- function(fil, SLoss, nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i])
    a <- NaiveLongSystem(Mkt, SLoss, nm[i])
    df10 <- rbind(df10, a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res1 <- run_NaiveLongSystem(fil,0,nm)

#res1 <- debug(run_NaiveLongSystem)
#res1 <- run_NaiveLongSystem(fil,0,nm)
#undebug(run_NaiveLongSystem(fil,0,nm))

# produce latex table
dat <- res1[,c(1,3,5,7)]
dig <- 2
cap = c('Naive Long System. A very simple system in which the algorithm assumes
    the market will rise and enters a long trade each day.',
            'Naive Long System')
lab = 'tab:nlng_results'
filname ='../Tables/chp_ta_naive_long.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# ---------------------------------------------
# --------- previous close and today's close

run_NaiveLongSystem2 <- function(fil,SLoss, nm){
df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
for(i in 1:length(fil)){
  Dax <- read.csv(fil[i])
```

```
  a <- NaiveLongSystem2(Dax, 0, nm[i])
  df10 <- rbind(df10, a)
}
df.name <- names(a)
names(df10) <- df.name
df10 <- df10[-1,]
return(df10)
}

res2 <- run_NaiveLongSystem2(fil,0,nm)

# produce latex table
dat <- res2[,c(1,3,5,7)]
dig <- 2
cap = c('Naive Long System changed such that the trading period is the previous
    close price minus today\'s close.',
        'Naive Long System - Close to Close')
lab = 'tab:nlng_results_2'
filname ='../Tables/chp_ta_naive_long_ctoc.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# ------------------------------------------------------------------------------
# ----------- Follow Previous ---------------------------------------------------
# ------------------------------------------------------------------------------

source("../RCode/NaiveFollowPrev.R")
source("../RCode/Utils.R")
res3 <- run_NaiveFollowPrev(fil, 0, nm)

# produce latex table
dat <- res3[,c(1,3,4,5,7,8,10)]
dig <- 2
cap = c('Naive system which reverses the previous day\'s trade direction.',
                  'Naive Following System.')
lab = 'tab:ntfresults'
filname ='../Tables/chp_ta_naive_follow_prev.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# rpeat with a stop loss
res3a <- run_NaiveFollowPrev(fil, -75, nm)
#tt <- sub_df(res3a,res3);tt

# produce latex table
dat <- res3a[,std6]
dig <- 2
cap = c('Naive system which reverses the previous day\'s trade direction with
    stop loss.',
      'Naive Following System.')
lab = 'tab:ntfresults_sl'
filname ='../Tables/chp_ta_naive_follow_prev_sl.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)
```

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

# section{Trend Detection Indicators}

# SMA
run_BaseSystem1SMA <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Dax <- read.csv(fil[i])
    a <- BaseSystem1SMA(Dax, 5, SLoss, nm[i])
    b <- BaseSystem1SMA(Dax, 25, SLoss, nm[i])
    c <- BaseSystem1SMA(Dax, 50, SLoss, nm[i])
    d <- BaseSystem1SMA(Dax, 100, SLoss, nm[i])
    e <- BaseSystem1SMA(Dax, 200, SLoss, nm[i])
    df10 <- rbind(df10, a, b, c, d, e)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res4 <- run_BaseSystem1SMA(fil,0,nm)

dat <- res4[,c(1,3,4,5,7,8,10,11)]
dig <- 2
cap = c('Results from SMA system.','SMA Base System')
lab = 'tab:sma_results'
filname ='../Tables/chp_ta_sma.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# SMA SLoss
run_BaseSystem1SMA2 <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Dax <- read.csv(fil[i])
    #f <- BaseSystem1SMA(Dax, 5, -50, nm[i])
    #g <- BaseSystem1SMA(Dax, 5, -100, nm[i])
    h <- BaseSystem1SMA(Dax, 100, -50, nm[i])
    hh <- BaseSystem1SMA(Dax, 100, -100, nm[i])  #don't use i !!!!!
    df10 <- rbind(df10,h,hh)
  }
  df.name <- names(hh)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res5 <- run_BaseSystem1SMA2(fil,0,nm)

```

```
dat <- res5[,c(1,2,3,4,5,6,8,9,11)]
dig <- 2
cap =  c('Results from SMA system with Stop Loss.',
                      'SMA Base System with Stop Loss')
lab = 'tab:sma_results_Sloss'
filname ='../Tables/chp_ta_sma_sloss.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# ----------------------------------------------------------------------
# subsection{Moving Average Convergence/Divergence (MACD)}
# subsubsection{MACD as trend Indicator}

run_MACD_XO <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i])
    ma <- MACD( Mkt[,"Open"], 12, 26, 9, maType="EMA" ) #calc MACD values
    Mkt <- cbind(Mkt, ma)
    a <- MACD_XO(Mkt, SLoss, nm[i])
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res6 <- run_MACD_XO(fil,0,nm)

dat <- res6[,std6]
dig <- 2
cap =  c('Results from system using MACD as a Trend Indicator.','Results from
    system using MACD as a Trend Indicator')
lab = 'tab:mac_trend_results'
filname ='../Tables/chp_ta_macd.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# ----------------------------------------------------------
# ---------------------------  Aroon -----------------------

run_aroon_sys <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i])
    ar <- aroon(Mkt[c(3,4)], n=20)                #calc Aroon values
    Mkt <- cbind(Mkt, ar)                          #Add Aroon values to orig
    data set
    a <- aroon_sys(Mkt, SLoss, nm[i])
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
```

```
  df10 <- df10[-1,]
  return(df10)
}

res7 <- run_aroon_sys(fil,0,nm)

dat <- res7[,std6]
dig <- 2
cap =  c('Aroon trend indicator.',
                      'Aroon trend indicator')
lab = 'tab:aroon_results'
filname ='../Tables/chp_ta_aroon.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# Aroon with SLoss
aroondfsl <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
for(i in 1:length(fil)){
  Dax <- read.csv(fil[i])                       #read data
  ar <- aroon(Dax[c(3,4)], n=20)                #calc Aroon values
  Dax <- cbind(Dax, ar)                          #Add Aroon values to orig data
    set
  a <- aroon_sys(Dax, -100, nm[i])                 #Call fnc
  aroondfsl <- rbind(aroondfsl, a)
}
df.name <- names(a)
names(aroondfsl) <- df.name

res7a <- run_aroon_sys(fil,-100,nm)
aroondfsl <- res7a

dat <- res7a[,std6]
dig <- 2
cap =  c('Aroon trend indicator with stop loss.',
                      'Aroon trend indicator with Stop Loss')
lab = 'tab:aroon_results_sloss'
filname ='../Tables/chp_ta_aroon_sloss.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# Aroon - Diffs

aroondfsldf <- as.data.frame(matrix(seq(3),nrow=1,ncol=3))
ln <- nrow(aroondfsl)
res <- 1:3
for(i in 1:ln){
  res[1] <- aroondfsl[i,1]
  res[2] <- as.numeric(res7a[i,3]) - as.numeric(res7[i,3])
  res[3] <- as.numeric(res7a[i,4]) - as.numeric(res7[i,4])
  aroondfsldf <- rbind(aroondfsldf,res)
}
df.name <- c("Market", "Long Difference", "Short Difference")
names(aroondfsldf) <- df.name
```

```
aroondfsldf <- aroondfsldf[-1,]

dat <- aroondfsldf[,c(1,2,3)]
dig <- 2
cap =  c('Impact of stop loss on Aroon.',
                      'Impact of using stop loss with Aroon trend indicator.')
lab = 'tab:aroon_results_sloss_diff'
filname ='../Tables/chp_ta_aroon_diff.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# ---------------------------------------------------------------------------
# ----------- Trend REversal ------------------------

# ---------- SAR
run_sar_sys <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i])
    sar <- SAR(Mkt[c(3,4)]) #HL
    Mkt <- cbind(Mkt,sar)
    a <- sar_sys(Mkt,SLoss, nm[i])
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res8 <- run_sar_sys(fil,0,nm)

dat <- res8[,std6]
dig <- 2
cap =  c('Results from SAR system.','SAR Base System')
lab = 'tab:sar_results'
filname ='../Tables/chp_ta_sar.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# ----------------------------------------------------------
# --------------------------  MACD OB -----------------------

run_MACD_OB <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i])
    ma <- MACD( Mkt[,"Open"], 12, 26, 9, maType="EMA" ) #calc MACD values
    Mkt <- cbind(Mkt, ma)                               #Add MACD values to orig
    data set
    lw <- quantile(Mkt$macd, na.rm=T, probs=0.15)       #Calc low val for algo
    up <- quantile(Mkt$macd, na.rm=T, probs=0.85)       #Calc up val for algo
    a <- MACD_OB(Mkt, 0, nm[i], lw, up)
```

```
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res9 <- run_MACD_OB(fil,0,nm)

dat <- res9[,std6]
dig <- 2
cap =  c('MACD can also be used as a trend reveral indicator.',
                      'MACD as Trend Reversal Indicator')
lab = 'tab:mac_ob_results'
filname ='../Tables/chp_ta_macd_ob.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


#-----------------------------------------------------------
# ---------------------------  stoch ------------------------

ln <- nrow(df10)
for(i in 1:length(fil)){
  Dax <- read.csv(fil[i])
  st <- stoch(Dax[c(3,4,5)]) #HL
  Dax <- cbind(Dax,st)
  a <- stoch_sys(Dax, 0, nm[i])
  df10 <- rbind(df10, a)
}
df10 <- df10[-c(1:ln-1),]

run_stoch_sys <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i])
    st <- stoch(Mkt[c(3,4,5)]) #HL
    Mkt <- cbind(Mkt,st)
    a <- stoch_sys(Mkt, SLoss, nm[i])
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res10 <- run_stoch_sys(fil,0,nm)

dat <- res10[,std6]
dig <- 2
cap =  c('Results from Stochastics system.',
                      'Stochastics system')
lab = 'tab:stoch_results'
```

```
filname ='../Tables/chp_ta_stoch.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# Stock plus SLoss
res10a <- run_stoch_sys(fil,-100,nm)

dat <- res10a[,std6]
dig <- 2
cap =  c('Results from Stochastics system and using a Stop Loss.',
                      'Stochastics system with stop loss')
lab = 'tab:stoch_results_sloss'
filname ='../Tables/chp_ta_stoch_sloss.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

#-----------------------------------------------------------
# ---------------------------  ROC ------------------------

run_roc_sys <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i])
    roc <- ROC( Mkt$Close )                        #calc MACD values
    Mkt <- cbind(Mkt, roc)                         #Add MACD values to orig
    data set
    lw <- quantile(Mkt$roc, na.rm=T, probs=0.15)     #Calc low val for algo
    up <- quantile(Mkt$roc, na.rm=T, probs=0.85)     #Calc up val for algo
    a <- roc_sys(Mkt, SLoss, nm[i], lw, up)
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res11 <- run_roc_sys(fil,0,nm)

dat <- res11[,std6]
dig <- 2
cap =  c('ROC.',
                                'ROC')
lab = 'tab:mac_roc_results'
filname ='../Tables/chp_ta_roc.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# ROC 2
#If previous ROC was greater or smaller than 0:
#source("../RCode//ROC2.R")
# ln <- nrow(df10)
# #results <- 1:11
# for(i in 1:length(fil)){
```

```
#   Mkt <- read.csv(fil[i])                              #read data
#   roc <- ROC( Mkt$Close )                              #calc MACD values
#   Mkt <- cbind(Mkt, roc)                               #Add MACD values to orig
    data set
#   lw <- quantile(Mkt$roc, na.rm=T, probs=0.15)        #Calc low val for algo
#   up <- quantile(Mkt$roc, na.rm=T, probs=0.85)        #Calc up val for algo
#   a <- roc_sys2(Mkt, 0, nm[i])                #Call fnc
#   df10 <- rbind(df10, a)               #add results
# }
# df10 <- df10[-c(1:ln-1),]                #NOTE ln-1 !!!!!
#
# dat <- df10[-1,std6]
# dig <- 2
# cap =  c('ROC2.',
#                                  'ROC2')
# lab = 'tab:mac_roc2_results'
# filname ='../Tables/chp_ta_roc2.tex'
# inclrnam=FALSE
# print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# ----------------------------------------------------------------
# ------------section{Break-out systems}

#----------------------------------------------------------------
# --------------------------  Break Out -----------------------
run_BaseSystem2Bout <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i])
    a <- BaseSystem2Bout(Mkt, SLoss, nm[i])
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res12 <- run_BaseSystem2Bout(fil,0,nm)

dat <- res12[,std6]
dig <- 2
cap =  c('Results from Daily High / Low Breakout System.',
                                'Daily High / Low Breakout System')
lab = 'tab:hl_bout_sys'
filname ='../Tables/chp_ta_b_out.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


#----------------------------------------------------------------
# --------------------------  90% Quant -----------------------

run_BaseSystem3Quant902 <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
```

```
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i])
    a <- BaseSystem3Quant902(Mkt, SLoss, nm[i])
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res14 <- run_BaseSystem3Quant902(fil,0,nm)

dat <- res14[,std6]
dig <- 2
cap =  c('Results from a trading system using 90\\% Quantile level.',
         'Break-out of 90\\% Quantile')
lab = 'tab:q_90_results'
filname ='../Tables/chp_ta_90q.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# ----------------------------------------------------------------------------
# -------section{Candlestick Patterns}

run_candle_hammer <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i],stringsAsFactors = FALSE)
    Mkt <- Mkt[,c(1,2,3,4,5)]
    Mkt$Date <- as.POSIXct(Mkt$Date,format='%d/%m/%Y')
    Mkt_xts <- xts(Mkt[,c(2,3,4,5)],Mkt$Date)
    hh <- as.data.frame(CSPHammer(Mkt_xts))
    hi <- as.data.frame(CSPInvertedHammer(Mkt_xts))
    Mkt <- cbind(Mkt,hh)
    Mkt <- cbind(Mkt,hi)
    a <- candle_hammer(Mkt,SLoss, nm[i])
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res14 <- run_candle_hammer(fil,0,nm)

# latex table
dat <- res14[,c(1,3,5,6,7)]
dig <- 2
cap = c('Results from Hammer / Inverted Hammer.','Hammer System')
lab = 'tab:hammer_results'
filname ='../Tables/chp_ta_hammer.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)
```

```

# plus aroon
run_candle_hammer_aroon <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i],stringsAsFactors = FALSE)
    Mkt <- Mkt[,c(1,2,3,4,5)]
    Mkt$Date <- as.POSIXct(Mkt$Date,format='%d/%m/%Y')
    Mkt_xts <- xts(Mkt[,c(2,3,4,5)],Mkt$Date)
    hh <- as.data.frame(CSPHammer(Mkt_xts))
    hi <- as.data.frame(CSPInvertedHammer(Mkt_xts))
    Mkt <- cbind(Mkt,hh)
    Mkt <- cbind(Mkt,hi)
    ar <- aroon(Mkt$Close,n=20)
    Mkt <- cbind(Mkt,ar)
    a <- candle_hammer(Mkt,SLoss, nm[i])
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res14a <- run_candle_hammer_aroon(fil,0,nm)

# latex table
dat <- res14a[,c(1,3,5,6,7)]
dig <- 2
cap =  c('Results from Hammer / Inverted Hammer occurring in a downtrend as
    defined by the aroon value.',
                                'Hammer System in downtrend.')
lab = 'tab:hammer_aroon_results'
filname ='../Tables/chp_ta_hammer_d_trend.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# ------------------------------------------------------------
# -----------------  Engulfing Candlestick -------------------

run_candle_engulf <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i],stringsAsFactors = FALSE)
    #create xts obj
    Mkt$Date <- as.POSIXct(Mkt$Date,format='%d/%m/%Y')
    Mkt_xts <- xts(Mkt[,c(2,3,4,5)],Mkt$Date)
    en <- as.data.frame(CSPEngulfing(Mkt_xts))
    #use data frame again
    Mkt <- cbind(Mkt,en)
    a <- candle_engulf(Mkt,SLoss, nm[i])
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
```

```
  df10 <- df10[-1,]
  return(df10)
}

res15 <- run_candle_engulf(fil,0,nm)

# latex table
dat <- res15[,std6]
dig <- 2
cap =  c('Results from Engulfing Candlestick.',
                                 'Engulfing Candlestick System')
lab = 'tab:engulf_results'
filname ='../Tables/chp_ta_englf.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# with Aroon
run_candle_engulf_aroon <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i],stringsAsFactors = FALSE)
    #create xts obj
    Mkt$Date <- as.POSIXct(Mkt$Date,format='%d/%m/%Y')
    Mkt_xts <- xts(Mkt[,c(2,3,4,5)],Mkt$Date)
    en <- as.data.frame(CSPEngulfing(Mkt_xts))
    #use data frame again
    Mkt <- cbind(Mkt,en)
    ar <- aroon(Mkt$Close,n=20)
    Mkt <- cbind(Mkt,ar)
    a <- candle_engulf_aroon(Mkt,SLoss, nm[i])
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res15a <- run_candle_engulf_aroon(fil,0,nm)

# latex table
dat <- res15a[,std6]
dig <- 2
cap =  c('Results from Engulfing Candlestick with Aroon.',
                                 'Engulfing Candlestick System with Aroon')
lab = 'tab:engulf_aroon_results'
filname ='../Tables/chp_ta_englf_aroon.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# -------------------------------------------------------
# ---------------------------  Doji ----------------------
run_candle_doji_aroon <- function(fil,SLoss,nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
```

```
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i],stringsAsFactors = FALSE)
    #create xts obj
    Mkt$Date <- as.POSIXct(Mkt$Date,format='%d/%m/%Y')
    Mkt_xts <- xts(Mkt[,c(2,3,4,5)],Mkt$Date)
    dj <- as.data.frame(CSPDoji(Mkt_xts))
    #back to data fram
    Mkt <- cbind(Mkt,dj)
    ar <- aroon(Mkt$Close,n=20)
    Mkt <- cbind(Mkt,ar)
    a <- candle_doji_aroon(Mkt,SLoss, nm[i])
    df10 <- rbind(df10,a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}

res16 <- run_candle_doji_aroon(fil,0,nm)

# latex table
dat <- res16[,std6]
dig <- 2
cap = c('Results from Doji Candlestick with aroon.',
                                 'Doji Candlestick System with aroon')
lab = 'tab:doji_aroon_results'
filname ='../Tables/chp_ta_doji.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# END
```

RCode/Chapter4.R

A.1.2 Naive Systems

A.1.2.1 Naive Long

```
NaiveLongSystem <- function(Mkt, SLoss, MktName){
  # Calculates the profit/loss from simply trading long.
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  # Buy Long
```

```
  Mkt$Long <- Mkt$Close - Mkt$Open
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  return(results)
}
```

RCode/NaiveLongSystem.R

A.1.2.2 Naive Long - Close to Close

```
NaiveLongSystem2 <- function(Mkt, SLoss, MktName){
  # Calculates the profit/loss from simply trading long each day.
  # Opening price is previous day's close price.
  #
  # Args:
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: name of market data
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  Mkt$prevCl <- c(NA,Mkt$Close[ - length(Mkt$Close) ])

  # Buy Long
  Mkt$Long <- Mkt$Close - Mkt$prevCl
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  return(results)
}
```

RCode/NaiveLongSystem2.R

A.1.2.3 Naive Follow Prev

```
NaiveFollowPrev <- function(Mkt, SLoss, MktName){
  # Calculates the profit/loss from trading according to a naive follow previous
    day idea.
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   profit/loss from trading according to SMA.

  results <- createResultsVector(MktName, SLoss)

  Mkt$pl <- Mkt$Close - Mkt$Open
  #Mkt$prevPL <- c( NA, Mkt$Close[ - length(Mkt$Close) ] - Mkt$Open[ - length(Mkt
    $Open) ] )
  Mkt$prevPL <- c( NA, Mkt$pl[ - length(Mkt$pl) ] )


  # Trade Long
  Mkt$Long <- ifelse(Mkt$prevPL<0,Mkt$Close-Mkt$Open,NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$prevPL<0,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)

    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$prevPL>0,Mkt$Open-Mkt$Close,NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Short <- ifelse(Mkt$prevPL>0,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/NaiveFollowPrev.R

A.1.3 Trend Detection Systems

A.1.3.1 SMA

```r
BaseSystem1SMA <- function(Mkt, sma, SLoss, MktName){
  # Calculates the profit/loss from trading according to SMA.
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   profit/loss from trading according to SMA.

  results <- createResultsVector(MktName, SLoss)

  sma.value <- SMA(Mkt["Open"], sma)  #create sma vector
  Mkt <- cbind(Mkt, sma.value)        #add sma vector as new col

  # Trade Long
  #browser()
  Mkt$Long <- ifelse(Mkt$Open > Mkt$sma.value, Mkt$Close - Mkt$Open, NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=T))
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$Open > Mkt$sma.value,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=T))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$Open < Mkt$sma.value, Mkt$Open - Mkt$Close, NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=T))
  if (SLoss < 0) {
    Mkt$Short <- ifelse(Mkt$Open < Mkt$sma.value,
                               ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                               Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=T))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  results[11] <- sma
  nm <- c("Mkt",          # 1. Name of Mkt
          "S Loss",       # 1. Name of Mkt
          "LongPL",       # 1. Name of Mkt
          "ShortPL",      # 1. Name of Mkt
          "L Win %",      # 1. Name of Mkt
          "L Trades",    # 1. Name of Mkt
```

```
          "Av L PL",      # 1. Name of Mkt
          "S Win %",      # 1. Name of Mkt
          "S Trades",    # 1. Name of Mkt
          "Av S PL",
          "SMA")
  names(results) <- nm

  #write.csv(Mkt, 'smatest.csv')

  return(results)
}
```

RCode/SMA_sys.R

A.1.3.2 MACD - trend indicator

```
MACD_XO <- function(Mkt, SLoss, MktName){
  # MACD cross-over system.
  #
  # Args:
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  # Trade Long
  Mkt$Long <- ifelse(Mkt$macd>Mkt$signal,Mkt$Close-Mkt$Open,NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$macd>Mkt$signal,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$macd<Mkt$signal,Mkt$Open-Mkt$Close,NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Short <- ifelse(Mkt$macd<Mkt$signal,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats
```

```

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/MACD_XO.R

A.1.3.3 Aroon trend indicator

```
aroon_sys <- function(Mkt, SLoss, MktName){
  # uses Aroon indicator to trigger rades
  #
  # Args:
  #   Mkt:      Data
  #   SLoss:    Stop Loss (if 0 not used)
  #   MktName:  Name of market
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

    # Trade Long
  Mkt$Long <- ifelse(Mkt$aroonUp >= 70,Mkt$Close-Mkt$Open,NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$aroonUp >= 70,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$aroonDn >= 70,Mkt$Open-Mkt$Close,NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  if (SLoss < 0) {
      Mkt$Short <- ifelse(Mkt$aroonDn >= 70,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/Aroon.R

A.1.4 Market Reversal Indicator

A.1.4.1 SAR reversal indicator

```
sar_sys <- function(Mkt, SLoss, MktName){
  # uses Aroon indicator to trigger rades
  #
  # Args:
  #   Mkt:      Data
  #   SLoss:    Stop Loss (if 0 not used)
  #   MktName:  Name of market
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  Mkt$prevsar <- c( NA, Mkt$sar[ - length(Mkt$sar) ])

  # Trade Long
  Mkt$Long <- ifelse(Mkt$Open > Mkt$prevsar,Mkt$Close-Mkt$Open,NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$Open > Mkt$prevsar,
                      ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                      Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$Open < Mkt$prevsar,Mkt$Open-Mkt$Close,NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  if (SLoss < 0) {
    Mkt$Short <- ifelse(Mkt$Open < Mkt$prevsar,
                       ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                       Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/SAR.R

A.1.4.2 MACD as Reversal Indicator

```
MACD_OB <- function(Mkt, SLoss, MktName,lw, up){
  # MACD over-bought/sold system.
  #
  # Args:
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #   lw: value of MACD that signals end of bear runs and rev
  #   up: value of MACD that signals end of bull runs and rev
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  # Break out high
  Mkt$Long <- ifelse(Mkt$macd < lw,Mkt$Close-Mkt$Open,NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$macd < lw,
                      ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                      Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Break out low
  Mkt$Short <- ifelse(Mkt$macd > up,Mkt$Open-Mkt$Close,NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  if (SLoss < 0) {
    Mkt$Short <- ifelse(Mkt$macd > up,
                       ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                       Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/MACD_OB.R

A.1.4.3 Stochastic reversal indicator

```
stoch_sys <- function(Mkt, SLoss, MktName){
  # uses Stochastic Oscillator to trigger rades
  #
  # Args:
  #   Mkt:      Data
  #   SLoss:    Stop Loss (if 0 not used)
  #   MktName:  Name of market
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  Mkt$PrevfastD <- c( NA, Mkt$fastD[ - length(Mkt$fastD) ])
  Mkt$PrevslowD <- c( NA, Mkt$slowD[ - length(Mkt$slowD) ])

  # Trade Long
  Mkt$Long <- ifelse(Mkt$PrevfastD > Mkt$PrevslowD,Mkt$Close-Mkt$Open,NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$PrevfastD > Mkt$PrevslowD,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$PrevfastD < Mkt$PrevslowD,Mkt$Open-Mkt$Close,NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Short <- ifelse(Mkt$PrevfastD < Mkt$PrevslowD,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/Stoch.R

A.1.4.4 Rate of Change(ROC)

```
roc_sys <- function(Mkt, SLoss, MktName,lw, up){
  # Rate of Change (ROC) system.
  #
  # Args:
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #   lw: value of MACD that signals end of bear runs and rev
  #   up: value of MACD that signals end of bull runs and rev
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  Mkt$prevROC <- c( NA, Mkt$roc[ - length(Mkt$roc) ] )

  # Trade Long
  Mkt$Long <- ifelse(Mkt$prevROC < lw,Mkt$Close-Mkt$Open,NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$prevROC < lw,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$prevROC > up,Mkt$Open-Mkt$Close,NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Short <- ifelse(Mkt$prevROC > up,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/ROC.R

A.1.5 Break Out Systems

A.1.5.1 Break Out

```
BaseSystem2Bout <- function(Mkt, SLoss, MktName){
  # Calculates the profit/loss from a break out system.
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  Mkt$prevHigh <- c( NA, Mkt$High[ - length(Mkt$High) ] )
  Mkt$prevLow <- c( NA, Mkt$Low[ - length(Mkt$Low) ] )

  # Break out high
  Mkt$Long <- ifelse(Mkt$High>Mkt$prevHigh,Mkt$Close-Mkt$prevHigh,NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$High>Mkt$prevHigh,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Break out low
  Mkt$Short <- ifelse(Mkt$Low<Mkt$prevLow,Mkt$prevLow-Mkt$Close,NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  if (SLoss < 0) {
    Mkt$Short <- ifelse(Mkt$Low<Mkt$prevLow,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/Bout_sys.R

A.1.5.2 90% Quantile

```
BaseSystem3Quant902 <- function(Mkt, SLoss, MktName){
  # Calculates the profit/loss from trading a breakout of a 90% quantile move.
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  Mkt$OH <- Mkt$High - Mkt$Open
  Mkt$OL <- Mkt$Open - Mkt$Low
  Mkt$mn <- ifelse(Mkt$OH>Mkt$OL,Mkt$OL,Mkt$OH)
  Mkt$mx <- ifelse(Mkt$OH>Mkt$OL,Mkt$OH,Mkt$OL)
  qq <- quantile(Mkt$mn, probs=0.90)

  # Trade Long
  Mkt$Long <- ifelse((Mkt$High - Mkt$Open) > qq, Mkt$Close - (Mkt$Open + qq), NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse((Mkt$High - Mkt$Open) > qq,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse((Mkt$Open - Mkt$Low) > qq, (Mkt$Open - qq) - Mkt$Close, NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0){
    Mkt$Short <- ifelse((Mkt$Open - Mkt$Low) > qq,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/Quant90_sys.R

A.1.6 Candlestick Systems

A.1.6.1 Hammer and Inverted Hammer

```
candle_hammer <- function(Mkt, SLoss, MktName){
  # Calculates the profit/loss from trading a based on candelstick Hammer.
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  Mkt$prev_Hammer   <- c( NA, Mkt$Hammer[ - length(Mkt$Hammer) ] )
  Mkt$prev_Inv_Hammer   <- c( NA, Mkt$InvertedHammer[ - length(Mkt$InvertedHammer
    ) ] )

  # Trade Long
  Mkt$Long <- ifelse(Mkt$prev_Hammer==TRUE | Mkt$prev_Inv_Hammer==TRUE, Mkt$Close
    -Mkt$Open, NA)
   results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse((Mkt$prev_Hammer==TRUE | Mkt$prev_Inv_Hammer==TRUE) > 0,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  return(results)
}
```

RCode/Candle_Hammer.R

```
candle_hammer_aroon <- function(Mkt, SLoss, MktName){
  # Calculates the profit/loss from trading a based on candelstick Hammer in a
    trend.
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  #browser()
```

```
  Mkt$prev_Aroon_UP <- c( NA, Mkt$aroonUp[ - length(Mkt$aroonUp) ] )
  Mkt$prev_Aroon_DN <- c( NA, Mkt$aroonDn[ - length(Mkt$aroonDn) ] )
  Mkt$prev_Hammer   <- c( NA, Mkt$Hammer[ - length(Mkt$Hammer) ] )
  Mkt$prev_Inv_Hammer   <- c( NA, Mkt$InvertedHammer[ - length(Mkt$InvertedHammer
    ) ] )

  # Trade Long
  Mkt$Long <- ifelse(Mkt$prev_Aroon_DN >= 70, ifelse(Mkt$prev_Hammer==T | Mkt$
    prev_Inv_Hammer==T, Mkt$Close-Mkt$Open, NA) ,NA)

  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))

  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse((Mkt$High - Mkt$Open) > 0,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  return(results)
}
```

RCode/Candle_Hammer_aroon.R

```
candle_engulf <- function(Mkt, SLoss, MktName){
  # Calculates the profit/loss from trading a based on an Engulfing candelstick.
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  Mkt$prev_Bull_Engulf <- c( NA, Mkt$Bull.Engulfing[ - length(Mkt$Bull.Engulfing)
     ] )
  Mkt$prev_Bear_Engulf <- c( NA, Mkt$Bear.Engulfing[ - length(Mkt$Bear.Engulfing)
     ] )

  # Trade Long
   Mkt$Long <- ifelse(Mkt$prev_Bull_Engulf==TRUE, Mkt$Close-Mkt$Open, NA)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
#   #Adj for SLoss
   if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$prev_Bull_Engulf == TRUE,
                       ifelse( (Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
     results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
   }
```

```

  # Trade Short
  Mkt$Short <- ifelse(Mkt$prev_Bear_Engulf == TRUE,Mkt$Open-Mkt$Close,NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Short <- ifelse(Mkt$prev_Bear_Engulf == TRUE,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/Candle_Engulf.R

```
candle_engulf_aroon <- function(Mkt, SLoss, MktName){
  # Calculates the profit/loss from trading a based on an Engulfing candelstick in a trending market.
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  #browser()
  Mkt$prev_Aroon_UP <- c( NA, Mkt$aroonUp[ - length(Mkt$aroonUp) ] )
  Mkt$prev_Aroon_DN <- c( NA, Mkt$aroonDn[ - length(Mkt$aroonDn) ] )
  Mkt$prev_Bull_Engulf   <- c( NA, Mkt$Bull.Engulfing[ - length(Mkt$Bull.Engulfing) ] )
  Mkt$prev_Bear_Engulf   <- c( NA, Mkt$Bear.Engulfing[ - length(Mkt$Bear.Engulfing) ] )

  # Trade Long
  Mkt$Long <- ifelse(Mkt$prev_Aroon_DN >= 70, ifelse(Mkt$prev_Bull_Engulf==T, Mkt$Close-Mkt$Open, NA) ,NA)

  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))

  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse((Mkt$High - Mkt$Open) > 0,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
```

```
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  #Trade Short
  Mkt$Short <- ifelse(Mkt$prev_Aroon_UP >= 70, ifelse(Mkt$prev_Bull_Engulf==T,
    Mkt$Close-Mkt$Open, NA) ,NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0){
    Mkt$Short <- ifelse((Mkt$Open - Mkt$Low) > 0,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/Candle_Engulf_aroon.R

```
candle_doji_aroon <- function(Mkt, SLoss, MktName){
  # Calculates the profit/loss from using Doji candlestick pattern.
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  #browser()
  Mkt$prev_Aroon_UP <- c( NA, Mkt$aroonUp[ - length(Mkt$aroonUp) ] )
  Mkt$prev_Aroon_DN <- c( NA, Mkt$aroonDn[ - length(Mkt$aroonDn) ] )
  Mkt$prev_Doji   <- c( NA, Mkt$Doji[ - length(Mkt$Doji) ] )
  Mkt$prev_Dragonfly   <- c( NA, Mkt$DragonflyDoji[ - length(Mkt$DragonflyDoji) ]
     )
  Mkt$prev_Gravestone   <- c( NA, Mkt$GravestoneDoji[ - length(Mkt$GravestoneDoji
    ) ] )

  # Trade Long
  Mkt$Long <- ifelse(Mkt$prev_Aroon_DN >= 70, ifelse(Mkt$prev_Doji==TRUE | Mkt$
    prev_Dragonfly == TRUE, Mkt$Close-Mkt$Open, NA) ,NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))

  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse((Mkt$High - Mkt$Open) > 0,
```

```
                        ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                        Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  #Trade Short
  Mkt$Short <- ifelse(Mkt$prev_Aroon_UP >= 70, ifelse(Mkt$prev_Doji==TRUE | Mkt$
    prev_Gravestone == TRUE, Mkt$Close-Mkt$Open, NA) ,NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0){
    Mkt$Short <- ifelse((Mkt$Open - Mkt$Low) > 0,
                         ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                         Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/Candle_Doji_aroon.R

A.2 Chapter 5

The R code used to generate the results and tables in Chapter 5 is shown in listing A.2. This is followed by the individual files containing the algorithms used in the chapter.

```
# Chapter 5 - test
setwd("D:/Allan/DropBox/MSc/Dissertation/Thesis/RCode")

# libraries
library(forecast)
library(xtable)

#source
source("../RCode/Utils.R")
source("../RCode/ts_1.R")
source("../RCode/ts_2.R")
source("../RCode/ts_3.R")
source("../RCode/ts_3a.R")
source("../RCode/ts_4.R")
source("../RCode//NaiveFollowPrev.R")
#source("ts_1.R")
#source("ts_2.R")
#source("Utils.R")

```

```
fil <- c("../Data/Dax_2000_d.csv",
         "../Data/CAC_2000_d.csv",
         "../Data/F100_2000_d.csv",
         "../Data/Dow_2000_d.csv",
         "../Data/N225_2000_d.csv",
         "../Data/Oz_2000.csv")
#nm <- c("Dax", "CAC", "FTSE", "Dow", "Nikkei", "AORD")

# Add Naive follow prev for comparison purposes
# data frame will be fed into sub_df

df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
NaiveRev <- run_NaiveFollowPrev(fil, 0, nm)

# -----------------------------------------------------------
# ---------- Base Systems
# Mkt <- read.csv("../Data/Dax_2000_d.csv")
# nrow(Mkt)
# Mkt$Date[2999]
# Mkt_ts <- ts(Mkt$Close)
# #Mkt_ts <- ts(Mkt$Close,frequency=252, start=c(2000,1))
# #Mkt_train <- window(Mkt_ts, start=2000, end=2009.99)
# Mkt_train <- window(Mkt_ts, end=2999.99)
# Mkt_test <- window(Mkt_ts, start=3000)
#
# # a.build the  mean model
# mean_model <- meanf(Mkt_train, h=5)
# a <- accuracy(mean_model, Mkt_test) #out of sample
# rownames(a) <- c('Mean Training Set', 'Mean Test Set')
#
# # b. build the mean model
# naive_model <- naive(Mkt_train, h=5)
# b <- accuracy(naive_model, Mkt_test) #out of sample
# rownames(b) <- c('Naive Training Set', 'Naive Test Set')
#
#
# # c. build the drift model
# drift_model <- rwf(Mkt_train,drift=TRUE,h=5)
# c <- accuracy(drift_model, Mkt_test) #out of sample
# rownames(c) <- c('Drift Training Set', 'Drift Test Set')
#
# # combine results
# d <- rbind(a,b,c)
#
# # produce latex table
# dat <- d[,c(2,3,4,5,6)]
# dig <- 0
# cap <- c("Mean, Naive and Drift methods applied to
#          to the Dax.","Simple forecasting methods.")
# lab = 'tab:chp_ts:sma'
# filname ='../Tables/chp_ts_sma.tex'
# inclrnam=TRUE
# print_xt(dat,dig,cap,lab,al,filname,inclrnam)
#
# # --- plot all three base systems on Dow
```

```
# savepdf("chp_ts_dax1")
# Mkt_act <- window(Mkt_ts, start=3020, end=3200)
# plot.ts(Mkt_train,
#         main="Simple Forecasting Methods",
#         xlab="Days since 2000", ylab="Dax Closing Price",
#         xlim=c(2, 3200))
# lines(meanf(Mkt_train, h=350) $mean, col=4)
# lines(rwf(Mkt_train,h=350)$mean,col=2)
# lines(rwf(Mkt_train,drift=TRUE,h=350)$mean,col=3)
# legend("bottomright",lty=1,col=c(4,2,3),
#        legend=c("Mean method","Naive method","Drift method"))
# dev.off() #savepdf end
#
# # --- plot all three base systems on Dow PLUS actual data
# savepdf("chp_ts_dax1_plus_act_data")
# Mkt_act <- window(Mkt_ts, start=3020, end=3200)
# plot.ts(Mkt_train,
#         main="Simple Forecasting Methods",
#         xlab="Days since 2000", ylab="Dax Closing Price",
#         xlim=c(2, 3200))
# lines(meanf(Mkt_train, h=350) $mean, col=4)
# lines(rwf(Mkt_train,h=350)$mean,col=2)
# lines(rwf(Mkt_train,drift=TRUE,h=350)$mean,col=3)
# legend("bottomright",lty=1,col=c(4,2,3),
#        legend=c("Mean method","Naive method","Drift method"))
# lines(Mkt_act, col=6)
# dev.off() #savepdf end

# --------------- NOT USED AT MO -------------------
# plot diff range
# Mkt_test2 <- window(Mkt_ts, start=1510, end=1600)
# Mkt_train2 <- window(Mkt_ts, start=1000, end=1500)
# plot.ts(Mkt_train2,
#         main="Dax over 300 Days",
#         xlab="Day", ylab="",
#         xlim=c(1000, 1600),
#         ylim=c(3500, 6350))
# lines(meanf(Mkt_train2, h=150) $mean, col=4)
# lines(rwf(Mkt_train2,h=150)$mean,col=2)
# lines(rwf(Mkt_train2,drift=TRUE,h=150)$mean,col=3)
# legend("topleft",lty=1,col=c(4,2,3),
#        legend=c("Mean method","Naive method","Drift method"))
#
# # plot diff range PLUS actual data
# Mkt_test2 <- window(Mkt_ts, start=1510, end=1600)
# Mkt_train2 <- window(Mkt_ts, start=1000, end=1500)
# plot.ts(Mkt_train2,
#         main="Dax over 300 Days",
#         xlab="Day", ylab="",
#         xlim=c(1000, 1600),
#         ylim=c(3500, 6350))
# lines(meanf(Mkt_train2, h=150) $mean, col=4)
# lines(rwf(Mkt_train2,h=150)$mean,col=2)
# lines(rwf(Mkt_train2,drift=TRUE,h=150)$mean,col=3)
# legend("topleft",lty=1,col=c(4,2,3),
```

```
#        legend=c("Mean method","Naive method","Drift method"))
# lines(Mkt_test2,col=6)

# ----------------------------------------

# 1. Exp Smoothing

# Mkt <- read.csv("../Data/Dax_2000_d.csv")
# nrow(Mkt)
# Mkt$Date[2999]
# Mkt_ts <- ts(Mkt$Close)
# #Mkt_ts <- ts(Mkt$Close,frequency=252, start=c(2000,1))
# #Mkt_train <- window(Mkt_ts, start=2000, end=2009.99)
# Mkt_train <- window(Mkt_ts, end=2999.99)
# Mkt_test <- window(Mkt_ts, start=3000)
#
# # a.build the  mean model
# mean_model <- ets(Mkt_train)
# a <- accuracy(mean_model, Mkt_test) #out of sample
# rownames(a) <- c('Mean Training Set', 'Mean Test Set')
# a

# exp_sm <- function(Mkt_ts, Mkt, st){
#   #browser()
#   Mkta <- Mkt
#   cc <- Mkta[1,]
#   cc$a <- 0
#   ln <- nrow(Mkt)
#   lb <- 300 #lookback
#   for(i in 301:ed){
#     st <- i-300
#     Mkt_slice <- window(Mkt_ts,start=st,end=i)
#     mod <- ets(Mkt_slice, model="AAN")
#     fcast <- forecast.ets(mod)
#     a <- fcast$fitted[300]
#     b <- Mkta[i,]
#     ab <- cbind(b,a)
#     cc <- rbind(cc,ab)
#   }
#   cc <- cc[-1,]
#   return(cc)
# }
#
# Mkt <- read.csv("../Data/Dax_2000_d.csv")
# Mkt_ts <- ts(Mkt$Close)
# res <- exp_sm(Mkt_ts, Mkt, 3500)
# write.csv(res,'../Data/Dax_ets_aan_300.csv')

# ----------------------------------------
# 2. ARIMA ---------------------
Mkt <- read.csv("../Data/F100_2000_d.csv")
Mkt_ts <- ts(Mkt$Close)
Mkt_train <- window(Mkt_ts, end=2999.99)
Mkt_test <- window(Mkt_ts, start=3000)
```

```
# ------------------------------
# 2.1. Plot the data. Identify any unusual observations.
savepdf("chp_ts_ftse_2000-13")
plot.ts(Mkt_train,
        main="FTSE 2000 - 2013",
        xlab="Days since 2000",
        ylab="FTSE Closing Price",
        xlim=c(100, 3000))
dev.off()

# 2.2. If necessary, transform the data (using a Box-Cox transformation)
#to stabilize the variance.

# 2.3. If the data are non-stationary: take first differences of the
#data until the data are stationary.
savepdf("chp_ts_ftse_2000-13_diff")
plot(diff(Mkt_train),
          main="First Difference of FTSE 2000 - 2013",
          xlab="Days since 2000",
          ylab="FTSE Daily Price Movement",
          xlim=c(100, 3000))
dev.off()

# ------------------------------------
# 2.4. Examine the ACF/PACF: Is an AR(p) or MA(q) model appropriate?

# all 3 incl diff
savepdf("chp_ts_ftse_2000-13_diff_acf_tsd")
tsdisplay(diff(Mkt_train),main="FTSE 100 between 2000 and 2013",
          xlab="Days since 2000",
          ylab="FTSE Daily Price Movement")
dev.off()

# a ACF
savepdf("chp_ts_ftse_2000-13_diff_acf")
plot(Acf(diff(Mkt_train)),
     main="ACF of FTSE 100 between 2000 and 2013",
     ,ylim=c(-0.08, 0.08))
dev.off()

# a PACF
savepdf("chp_ts_ftse_2000-13_diff_pacf")
plot(Pacf(diff(Mkt_train)),
     main="PACF of FTSE 100 between 2000 and 2013",
     ylim=c(-0.08, 0.08))
dev.off()

# ---------------------------------------------------
# 2.5. Try your chosen model(s), and use the AICc to search for a better model.

mod_ar <- function(Mkt, ord, nm){
  res <- t(as.data.frame(rep(0,4)))
  mod <- Arima(Mkt_ts, order=ord)
  res[1,1] <- nm
  res[1,2] <- round(mod$aic,1)
```

```
  res[1,3] <- round(mod$aicc,1)
  res[1,4] <- round(mod$bic,1)
  return(res)
}

results <- t(as.data.frame(rep(0,4)))
colnames(results) <- c('Model','AIC','AICc','BIC')

r2 <- mod_ar(Mkt_train, c(3,1,1), 'Arima(3,1,1)')
results <- rbind(results,r2)
r2 <- mod_ar(Mkt_train, c(3,1,2), 'Arima(3,1,2)')
results <- rbind(results,r2)
r2 <- mod_ar(Mkt_train, c(3,1,3), 'Arima(3,1,3)')
results <- rbind(results,r2)
r2 <- mod_ar(Mkt_train, c(2,1,1), 'Arima(2,1,1)')
results <- rbind(results,r2)
r2 <- mod_ar(Mkt_train, c(2,1,2), 'Arima(2,1,2)')
results <- rbind(results,r2)
r2 <- mod_ar(Mkt_train, c(2,1,3), 'Arima(2,1,3)')
results <- rbind(results,r2)
results <- results[-1,]

# produce latex table
dat <- results
dig <- c(0,0,2,2,2)
cap <- c("AIC, AIc and BIC results from alternative ARIMA models.","alternative
    ARIMA models")
lab = 'tab:chp_ts:arima_res_r'
filname ='../Tables/chp_ts_arima_res_r.tex'
inclrnam=F
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# model_311 <- Arima(Mkt_ts, order=c(3,1,1))
# results[1,1] <- model_311$aic
# results[2,1] <- model_311$aicc
# results[3,1] <- model_311$bic

# ------------------------------------------------------
# 2.6. Check the residuals from your chosen model by plotting the ACF of the
    residuals,
#and doing a portmanteau test of the residuals.
#If they do not look like white noise, try a modified model.

model_used_for_res <- Arima(Mkt_train, order=c(2,1,3))
model_name <- forecast(model_used_for_res)$method

# a mean of residual
residual <- model_used_for_res$residuals
savepdf("chp_ts_ftse_2000-13_mean_residuals")
plot(residual, main = paste("Residuals from model of", model_name),
     ylab="", xlab="Day")
dev.off()

# b. acf of residual
savepdf("chp_ts_ftse_2000-13_acf_residuals")
```

```
Acf(residuals(model_used_for_res),
    main= paste("ACF of Residuals of", model_name))
dev.off()

# c. variance - use plot from a

# d. histogram of residuals - normal distribution
savepdf("chp_ts_ftse_2000-13_hist_residuals")
hist(residual, nclass="FD", main="Histogram of residuals")
dev.off()

# e. portmanteau tests
bb <- Box.test(residuals(model_used_for_res), lag=24, fitdf=4, type="Ljung")
results_bc <- as.data.frame(rep(0,3))
results_bc[1,1] <- round(bb$p.value,4)
results_bc[2,1] <- round(bb$parameter)
results_bc[3,1] <- round(bb$statistic)
#colnames(results_bc) <- c(paste(bb$method,forecast(model_311)$method))
colnames(results_bc) <- c(forecast(model_used_for_res)$method)
rownames(results_bc) <- c('p-value','x-squared','df')
#results_bc[1,1]
results_bc_t <- t(results_bc)

dat <- results_bc_t
dig <- c(0,4,0,0)
cap <- c("Box Ljung test.","Box Ljung test.")
lab = 'tab:chp_ts:arima_res_rbox_l'
filname ='../Tables/chp_ts_arima_res_r_box_l.tex'
inclrnam=TRUE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# 2.7 Once the residuals look like white noise, calculate forecasts.
model_used_for_res <- Arima(Mkt_ts, order=c(2,1,3))
model_name <- forecast(model_used_for_res)$method

arima_man_fcast <- forecast.Arima(model_used_for_res,Mkt_test)
fitted.data <- as.data.frame(arima_man_fcast$fitted);
#ln <- nrow(Mkt)
#lw <- nrow(fitted.data)
#Mkt_test_df <- Mkt[(ln-lw+1):ln,]
Mkt_test_df <- cbind(Mkt,fitted.data)
colnames(Mkt_test_df) <- c('Date','Open','High','Low','Close','Forecast')

# plot the results
dat <- tail(Mkt_test_df)
dig <- 0
cap <- c("FTSE 100 foecast.","FTSE 100 forecast.")
lab = 'tab:chp_ts:ftse_100_fcast'
filname ='../Tables/chp_ts_ftse_100_fcast.tex'
inclrnam=F
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


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

```
# 2.8 auto.arima

arim_mod_fnc <- function(fil,nm){
  #browser()
  dfres <- dfres <- t(c('a','b'))
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i])
    Mkt_train <- ts(Mkt$Close)
    #Mkt_train <- window(Mkt_ts, end=2999.99)
    #Mkt_test <- window(Mkt_ts, start=3000)
    arima_train_mod <- auto.arima(Mkt_train)
    #arima_fcast <- forecast.Arima(arima_train_mod)
    dfres <- rbind(dfres,c(nm[i], forecast(arima_train_mod)$method))
  }
  return(dfres)
}

fg <- arim_mod_fnc(fil,nm)
fg <- fg[-1,]
colnames(fg) <- c('Market','Arima Model')

# plot the results
dat <- fg
dig <- 0
cap <- c("Arima models  from national indices.","Arima models.")
lab = 'tab:chp_ts_arima_models'
filname ='../Tables/chp_ts_arima_models.tex'
inclrnam=F
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# ------------------------------------------------------
# 3. Trading System
# using the models generated from the auto.arima function

df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))

ts_1_fnc <- function(fil,nm,ts1){
  for(i in 1:length(fil)){

    Mkt <- read.csv(fil[i])
    Mkt_ts <- ts(Mkt$Close)
    Mkt_train <- window(Mkt_ts, end=2999.99)
    Mkt_test <- window(Mkt_ts, start=3000)
    arima_train_mod <- auto.arima(Mkt_train)
    arima_fcast <- forecast.Arima(arima_train_mod,Mkt_test)
    arima_test_mod <- Arima(Mkt_test, model = arima_train_mod) # 1 step fcast on
    future data ...
    arima_test_fcast <- forecast(arima_test_mod)
    fitted.data <- as.data.frame(arima_test_fcast$fitted);
    ln <- nrow(Mkt)
    lw <- nrow(fitted.data)
    Mkt_test_df <- Mkt[(ln-lw+1):ln,]
    Mkt_test_df <- cbind(Mkt_test_df,fitted.data)
    colnames(Mkt_test_df) <- c("Date","Open", "High","Low","Close","p")
```

```
    if(ts1 == TRUE){
      a <- ts_1(Mkt_test_df, 0, nm[i]) # System 1
    } else {
      a <- ts_2(Mkt_test_df, 0, nm[i]) # System 2
    }
    df10 <- rbind(df10, a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-c(1),]
  return(df10)
}

# run the fnc ts_1
# apply Sys 1 to the auto.arima data
res <- ts_1_fnc(fil,nm,TRUE)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Auto.arima models passed to the System 1 trading algorithm",
         "Sysytem 1 and auto.arima models")
lab = 'tab:chp_ts:arima1'
filname ='../Tables/chp_ts_arima1.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# compare to Naive reverse
diff_df <- sub_df_av_pl(res,NaiveRev)
# produce latex table from ts_1
#dat <- diff[,c(1,7,10)]
dat <- diff_df
dig <- 0
cap <- c("Mean Long/Short PL from Naive Reverse system subtracted from PL
    generated by auto.arima models",
         "Mean PL from Auto.arima models inus mean PL from Naive Reverse system")
lab = 'tab:chp_ts:arima1_diff'
filname ='../Tables/chp_ts_arima1_diff.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# ----------------------------------------------------------------
# run the fnc ts_2
# apply system 2 to auto.arima data
res <- ts_1_fnc(fil,nm,FALSE) # F = ts_2

# produce latex table from ts_2
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Auto.arima models passed to the System 2 trading algorithm",
         "Sysytem 1 and auto.arima models.")
lab = 'tab:chp_ts:arima2'
filname ='../Tables/chp_ts_arima2.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)
```

```


# ----------------------------------------------------------------------
# --------- RM Generated Files ----------------------------------------
# --------- HYBRID ARIMA SYSTEMS --------------------------------------

source("../RCode/ts_1.R")
source("../RCode/ts_2.R")
Mkt <- read.csv("../Data/rm_ar334_reg.csv",stringsAsFactors=F)

ts_1_2_fnc_ar <- function(fil,nm,ts1){
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i],stringsAsFactors=F)
    Mkt_p <- Mkt[,c(1,2,3,4,5,18)]
    colnames(Mkt_p) <- c("Date","Open", "High","Low","Close","p")
    if(ts1 == TRUE){
      a <- ts_1(Mkt_p, 0, nm[i])
    } else {
      a <- ts_2(Mkt_p, 0, nm[i])
    }
    df10 <- rbind(df10, a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-c(1),]
  return(df10)
}

# ---------- Predicting Closing Price -----------------
# 1. ------ Arima Ann Predicting Closing Price -----------------
fil <- c("../Data/ARIMA/Predict_Close/ar334_ann_DAX.csv",
         "../Data/ARIMA/Predict_Close/ar334_ann_CAC.csv",
         "../Data/ARIMA/Predict_Close/ar334_ann_FTSE.csv",
         "../Data/ARIMA/Predict_Close/ar334_ann_Dow.csv",
         "../Data/ARIMA/Predict_Close/ar334_ann_Nik.csv",
         "../Data/ARIMA/Predict_Close/ar334_ann_Oz.csv")

#nm <- c("Dax","CAC","FTSE","Dow","Nikkei","AORD")
df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))

# a. System 1
res <- ts_1_2_fnc_ar(fil,nm,TRUE)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting Close Price - Arima/ANN predictions passed to System 1",
         "Predicting Close Price - Arima/ANN predictions passed to System 1.")
lab = 'tab:chp_ts:arima_ann_sys1'
filname ='../Tables/chp_ts_arima_ann_sys1.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# comp aring to Naive Prev
res_diff <- sub_df_av_pl(res,NaiveRev)
```

```

dat <- res_diff
dig <- 0
cap <- c("Results from a trading system based on forecasts of closing price
    generated by the Arima/ANN model compared to baseline Naive Reversing
    methodology.",
         "Arima/ANN predictions passed to System 1 compared to Naive Reversing
    methodology")
lab = 'tab:chp_ts:arima_ann_sys1_diff'
filname ='../Tables/chp_ts_arima_ann_sys1_diff.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# a. System 2
res <- ts_1_2_fnc_ar(fil,nm,FALSE)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting Close Price - Arima/ANN predictions passed to System 2",
         "Predicting Close Price - Arima/ANN predictions passed to System 2.")
lab = 'tab:chp_ts:arima_ann_sys2'
filname ='../Tables/chp_ts_arima_ann_sys2.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# 2. ------ Arima knn Predicting Closing Price -----------------
fil <- c("../Data/ARIMA/Predict_Close/ar334_knn_Dax.csv",
         "../Data/ARIMA/Predict_Close/ar334_knn_CAC.csv",
         "../Data/ARIMA/Predict_Close/ar334_knn_F100.csv",
         "../Data/ARIMA/Predict_Close/ar334_knn_Dow.csv",
         "../Data/ARIMA/Predict_Close/ar334_knn_Nik.csv",
         "../Data/ARIMA/Predict_Close/ar334_knn_Oz.csv")

# a. System 1
#nm <- c("Dax","CAC","FTSE","Dow","Nikkei","AORD")
df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
# a. System 1
res <- ts_1_2_fnc_ar(fil,nm,TRUE)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting Close Price - Arima/k-NN predictions passed to System 1.",
         "Predicting Close Price - Arima/k-NN predictions passed to System 1.")
lab = 'tab:chp_ts:pred_close_arima_knn_sys1'
filname ='../Tables/chp_ts_pred_close_arima_knn_sys1.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# comp aring to Naive Prev
res_diff <- sub_df_av_pl(res,NaiveRev)

# produce latex table from ts_1
dat <- res_diff
```

```
dig <- 0
cap <- c("Predicting Close Price - Arima/k-NN predictions passed to System 1 diff
    .",
         "Predicting Close Price - Arima/k-NN predictions passed to System 1.")
lab = 'tab:chp_ts:pred_close_arima_knn_sys1_diff'
filname ='../Tables/chp_ts_pred_close_arima_knn_sys1_diff.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# a. System
res <- ts_1_2_fnc_ar(fil,nm,FALSE)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting Close Price - Arima/k-NN predictions passed to System 2",
         "Predicting Close Price - Arima/k-NN predictions passed to System 2")
lab = 'tab:chp_ts:pred_close_arima_knn_sys2'
filname ='../Tables/chp_ts_pred_close_arima_knn_sys2.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# -----------------------------------------------------------------
#  ------ Arima Ann Predicting Up/Dn - Categorical -----------------
# a. Categorical

# 1. ARMA / ANN (Predicting Up/Dn - Categorical)
#source("../RCode/ts_4.R")
source("../RCode/Utils.R")
fil <- c("../Data/ARIMA/PredUpDn_CAT/ar_334_UD_ANN_Dax.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_ANN_CAC.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_ANN_F100.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_ANN_Dow.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_ANN_N225.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_ANN_Oz.csv")

#nm <- c("Dax","CAC","FTSE","Dow","Nik","AORD")
df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))

res <- ts_4_fnc_ar(fil,0, nm)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting UpDn CAT - Arima/ANN predictions passed to System 4",
         "Predicting UpDn CAT - Arima/ANN predictions passed to System 4.")
lab = 'tab:chp_ts:pUD_CAT_arima_ann_sys'
filname ='../Tables/chp_ts_predUpDn_CAT_arima_ann_sys.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# ----------------------
# 2. ARMA / knn (Predicting Up/Dn - Categorical)
#source("../RCode/ts_4.R")
#source("../RCode/Utils.R")
```

```
fil <- c("../Data/ARIMA/PredUpDn_CAT/ar_334_UD_knn_Dax.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_knn_CAC.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_knn_F100.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_knn_Dow.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_knn_N225.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_knn_Oz.csv")

#nm <- c("Dax","CAC","FTSE","Dow","Nikkei","AORD")
df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))

res <- ts_4_fnc_ar(fil, 0, nm)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting UpDn CAT - Arima/k-NN predictions passed to System 4",
         "Predicting UpDn CAT - Arima/k-NN predictions passed to System 4.")
lab = 'tab:chp_ts:pUD_CAT_arima_knn_sys'
filname ='../Tables/chp_ts_predUpDn_CAT_arima_knn_sys.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# 2. ARMA / knn (Predicting Up/Dn - Categorical) - SLoss
res <- ts_4_fnc_ar(fil, -100, nm)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting UpDn CAT - Arima/k-NN predictions passed to System 4 - SLoss
    ",
         "Predicting UpDn CAT - Arima/k-NN predictions passed to System 4 - SLoss
    ")
lab = 'tab:chp_ts:pUD_CAT_arima_knn_sys_SL'
filname ='../Tables/chp_ts_predUpDn_CAT_arima_knn_sys_SL.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# -------------------
# comp aring to Naive Prev
res_diff <- sub_df_av_pl(res,NaiveRev)

# produce latex table from ts_1
dat <- res_diff
dig <- 0
cap <- c("Predicting UpDn CAT - Arima/k-NN predictions passed to System 4 - diff.
    ",
         "Predicting UpDn CAT - Arima/k-NN predictions passed to System 4 - ")
lab = 'tab:chp_ts:pUD_CAT_arima_knn_sys_diff'
filname ='../Tables/chp_ts_predUpDn_CAT_arima_knn_sys_diff.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# 3. ARMA / Reg (Logistic) (Predicting Up/Dn - Categorical)
```

```


# 4. ARMA / SVM (Predicting Up/Dn - Categorical)
fil <- c("../Data/ARIMA/PredUpDn_CAT/ar_334_UD_svm_Dax.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_svm_CAC.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_svm_F100.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_svm_Dow2.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_svm_N225.csv",
         "../Data/ARIMA/PredUpDn_CAT/ar_334_UD_svm_Oz.csv")

#nm <- c("Dax","CAC","FTSE","Dow","Nikkei","AORD")
df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))

res <- ts_4_fnc_ar(fil,0, nm)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting UpDn CAT - Arima/SVM predictions passed to System 4",
         "Predicting UpDn CAT - Arima/SVM predictions passed to System 4.")
lab = 'tab:chp_ts:pUD_CAT_arima_svm_sys'
filname ='../Tables/chp_ts_predUpDn_CAT_arima_svm_sys.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# ------------------------------------------------
#  ------ Arima Ann Predicting Up/Dn - 01 ---------
source("../RCode/ts_3a.R")
#source("../RCode/Utils.R")
# 1. ARMA / ANN - (Predicting Up/Dn - 01)
fil <- c("../Data/ARIMA/PredUpDn_01/ar_334_01_ANN_Dax.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_ANN_CAC.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_ANN_FTSE.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_ANN_Dow.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_ANN_N225.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_ANN_Oz.csv")

#nm <- c("Dax","CAC","FTSE","Dow","Nikkei","AORD")
df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))

res <- ts_3a_fnc_ar(fil, nm)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting UpDn 01 - Arima/ANN predictions passed to System 3",
         "Predicting UpDn 01 - Arima/ANN predictions passed to System 3.")
lab = 'tab:chp_ts:pUD_01_arima_ann_sys'
filname ='../Tables/chp_ts_predUpDn_01_arima_ann_sys.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# 2. ARMA / knn (Predicting Up/Dn - 01)

```

```
#source("../RCode/ts_3.R")
fil <- c("../Data/ARIMA/PredUpDn_01/ar_334_01_knn_Dax.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_knn_CAC.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_knn_FTSE.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_knn_Dow.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_knn_Nik.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_knn_Oz.csv")
#nm <- c("Dax","CAC","FTSE","Dow","Nikkei","AORD")
df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))

res <- ts_3_fnc_ar(fil, nm)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting UpDn 01 - Arima/k-NN predictions passed to System 3",
         "Predicting UpDn 01 - Arima/k-NN predictions passed to System 3.")
lab = 'tab:chp_ts:pUD_01_arima_knn_sys'
filname ='../Tables/chp_ts_predUpDn_01_arima_knn_sys.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)

# comp to Naive
res_diff <- sub_df_av_pl(NaiveRev, res)

dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting UpDn 01 - Arima/k-NN predictions passed to System 3 res diff
    ",
         "Predicting UpDn 01 - Arima/k-NN predictions passed to System 3.")
lab = 'tab:chp_ts:pUD_01_arima_knn_sys_diff'
filname ='../Tables/chp_ts_predUpDn_01_arima_knn_sys_diff.tex'
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# b3. ARMA / Reg  (Predicting Up/Dn - 01)
fil <- c("../Data/ARIMA/PredUpDn_01/ar_334_01_Reg_Dax.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_Reg_CAC.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_Reg_FTSE.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_Reg_Dow.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_Reg_Nik.csv",
         "../Data/ARIMA/PredUpDn_01/ar_334_01_Reg_Oz.csv")
#nm <- c("Dax","CAC","FTSE","Dow","Nikkei","AORD")
df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))

res <- ts_3_fnc_ar(fil, nm)

# produce latex table from ts_1
dat <- res[,c(1,3,4,5,7,8,10)]
dig <- 0
cap <- c("Predicting UpDn 01 - Arima/Reg predictions passed to System 3",
         "Predicting UpDn 01 - Arima/Reg predictions passed to System 3.")
lab = 'tab:chp_ts:01_arima_reg_sys'
filname ='../Tables/chp_ts_predUpDn_01_arima_reg_sys.tex'
```

```
inclrnam=FALSE
print_xt(dat,dig,cap,lab,al,filname,inclrnam)


# END
```

RCode/Chapter5.R

A.2.1 System 1

```
ts_1 <- function(Mkt, SLoss, MktName){
  # Calculates the profit/loss from trading a breakout of a 90% quantile move.
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  #Mkt$p_p <- c( NA, Mkt$p[ - length(Mkt$p) ] ) # prev prediction
  Mkt$p_c <- c( NA, Mkt$Close[ - length(Mkt$Close) ] ) # prev close

  # Trade Long
  Mkt$Long <- ifelse(Mkt$p > Mkt$p_c, Mkt$Close - Mkt$Open, NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$p > Mkt$p_c,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$p < Mkt$p_c, Mkt$Open - Mkt$Close, NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0){
    Mkt$Short <- ifelse(Mkt$p < Mkt$p_c,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats2(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats2(Mkt$Short)
  results[8:10] <- Stats

```

```
  return(results)
}
```

RCode/ts_1.R

A.2.2 System 2

```
ts_2 <- function(Mkt, SLoss, MktName){
  #
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  Mkt$p_p <- c( NA, Mkt$p[ - length(Mkt$p) ] ) # prev prediction
  #Mkt$p_c <- c( NA, Mkt$Close[ - length(Mkt$Close) ] ) # prev close

  # Trade Long
  Mkt$Long <- ifelse(Mkt$p > Mkt$p_p, Mkt$Close - Mkt$Open, NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$p > Mkt$p_p,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$p < Mkt$p_p, Mkt$Open - Mkt$Close, NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0){
    Mkt$Short <- ifelse(Mkt$p < Mkt$p_p,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats2(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats2(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/ts_2.R

A.2.3 Categorical Label

```
ts_4 <- function(Mkt, SLoss, MktName){
  #   trading system based on prediction fron ANN working with categorical
  #   label with valued U or D
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  # Trade Long
  Mkt$Long <- ifelse(Mkt$pred == "U", Mkt$Close - Mkt$Open, NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$pred == "U",
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$pred == "D", Mkt$Open - Mkt$Close, NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0){
    Mkt$Short <- ifelse(Mkt$pred == "D",
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats2(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats2(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/ts_4.R

A.2.4 Continuous Label

```
ts_3 <- function(Mkt, SLoss, MktName){
  #
  #
  #   Mkt: market data
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)

  # Trade Long
  Mkt$Long <- ifelse(Mkt$p > 0.5, Mkt$Close - Mkt$Open, NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$p > 0.5,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$p < 0.5, Mkt$Open - Mkt$Close, NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0){
    Mkt$Short <- ifelse(Mkt$p < 0.5,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats2(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats2(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/ts_3.R

A.2.5 Continuous Label - ARIMA/ANN

```
ts_3a <- function(Mkt, SLoss, MktName){
  #
  #
  #   Mkt: market data
```

```
  #   SLoss: stop loss
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- createResultsVector(MktName, SLoss)
  #browser()
  Mkt$v <- as.numeric(Mkt$p)
  lvl <- min(Mkt$v) + ((max(Mkt$v) - min(Mkt$v))/2)

  # Trade Long
  Mkt$Long <- ifelse(Mkt$v > lvl, Mkt$Close - Mkt$Open, NA)
  results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0) {
    Mkt$Long <- ifelse(Mkt$v > lvl,
                       ifelse((Mkt$Low-Mkt$Open) < SLoss, SLoss, Mkt$Long),
                       Mkt$Long)
    results["LongPL"] <- round(sum(Mkt$Long, na.rm=TRUE))
  }

  # Trade Short
  Mkt$Short <- ifelse(Mkt$v < lvl, Mkt$Open - Mkt$Close, NA)
  results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  #Adj for SLoss
  if (SLoss < 0){
    Mkt$Short <- ifelse(Mkt$v < lvl,
                        ifelse((Mkt$Open-Mkt$High) < SLoss, SLoss, Mkt$Short),
                        Mkt$Short)
    results["ShortPL"] <- round(sum(Mkt$Short, na.rm=TRUE))
  }

  Stats <- calcStats2(Mkt$Long)
  results[5:7] <- Stats

  Stats <- calcStats2(Mkt$Short)
  results[8:10] <- Stats

  return(results)
}
```

RCode/ts_3a.R

A.3 Utility Code

```
nm <- c("Dax", "CAC", "FTSE", "Dow", "Nikkei", "AORD")

createResultsVector <- function(MktName, SLossValue){
  # Function to create results vector
  #
  # Args:
```

```
  #   SLoss: stop loss value
  #   MktName: market's name for print out
  #
  # Returns:
  #   results vector.

  results <- rep(0,11)
  nam <- c("Mkt",          # 1. Name of Mkt
           "S Loss",       # 1. Name of Mkt
           "LongPL",       # 1. Name of Mkt
           "ShortPL",      # 1. Name of Mkt
           "L Win %",      # 1. Name of Mkt
           "L Trades",    # 1. Name of Mkt
           "Av L PL",      # 1. Name of Mkt
           "S Win %",      # 1. Name of Mkt
           "S Trades",    # 1. Name of Mkt
           "Av S PL",
           "misc")      # 1. Name of Mkt
  names(results) <- nam
  results["Mkt"] <- MktName
  results["S Loss"] <- SLossValue
  return(results)
}

calcStats <- function(x){
  # Function to calculate trade stats
  #
  # Args:
  #   x - data set
  #
  # Returns:
  #   results vector.

  results <- 1:3
  v <- na.omit(x)

  # Win %
  wins <- length(v[v>0])
  losses <- length(v[v<0])
  results[1] <- round(wins/(wins+losses)*100)

  # Num Trades
  results[2] <- length(v)

  # Av Long PL
  results[3] <- round(sum(v) / length(v))

  return(results)
}

calcStats2 <- function(x){
  # Function to calculate trade stats
  #
  # Args:
  #   x - data set
```

```
  #
  # Returns:
  #   results vector.
  #browser()
  results <- 1:3
  #v <- na.omit(x)
  v <- x

  # Win %
  wins <- sum(v>0,na.rm=T)
  losses <- sum(v<0,na.rm=T)
  results[1] <- round(wins/(wins+losses)*100)

  # Num Trades
  results[2] <- wins+losses

  # Av Long PL
  results[3] <- round(sum(v,na.rm=T) / (wins+losses))

  return(results)
}

calcWinPer <- function(x){
  wins <- length(x[x>0])
  losses <- length(x[x<0])
  return(wins/(wins+losses)*100)
}

calcAverageWin <- function(x){
  wins <- length(x)
  winpl <- sum(x, na.rm=T)
  return((winpl/wins))
}

calcNumTrades <- function(x){
  return(length(na.omit(x)))
}

savepdf <- function(file, width=16, height=10)
{
  fname <- paste("../Figures/",file,".pdf",sep="")
  pdf(fname, width=width/2.54, height=height/2.54,
      pointsize=10)
  par(mgp=c(2.2,0.45,0), tcl=-0.4, mar=c(3.3,3.6,1.1,1.1))
}


print_xt <- function(dat,dig,cap,lab,al,filname,inclrnam){
  xt <- xtable(
    dat,
    digits = dig,
    caption = cap,
    label = lab
  )
  al <- c('l','l')
```

```
  al <- c(al, rep('c',ncol(dat)-1))
  align(xt) <- al
  print(xt,
        file=filname,
        include.rownames=inclrnam,
        caption.placement = "top",
        hline.after=NULL,
        add.to.row=list(pos=list(-1,0, nrow(xt)),
                        command=c('\\toprule ', '\\midrule ', '\\bottomrule ')))

}


# subtract 2 data frames
# df2 from df1
sub_df <- function(df1, df2){

  nc <- ncol(df1)
  ln <- nrow(df1)
  dfres <- df1

  for(i in 1:ln){
    for(j in 2:nc){
      dfres[i,j] <- as.numeric(df1[i,j]) - as.numeric(df2[i,j])
    }
  }
  return(dfres)
}

sub_df_av_pl <- function(df1, df2){

  nc <- ncol(df1)
  ln <- nrow(df1)
  dfres <- df1
  for(i in 1:ln){
    for(j in 2:nc){
      dfres[i,j] <- as.numeric(df1[i,j]) - as.numeric(df2[i,j])
    }
  }
  dfres <- dfres[,c(1,7,10)]
  colnames(dfres) <- c('Mkt','Diff in Mean Long PL','Diff in Mean Short PL')
  return(dfres)
}


# ----------------------------------------------------------------
# ------------ CHAPTER 4 -----------------------------------------
# ----------------------------------------------------------------

# ----------- Follow Previous -------------
run_NaiveFollowPrev <- function(fil,SLoss, nm){
  df10 <- as.data.frame(matrix(seq(11),nrow=1,ncol=11))
  for(i in 1:length(fil)){
    Dax <- read.csv(fil[i],stringsAsFactors=F)
    a <- NaiveFollowPrev(Dax, SLoss, nm[i])
```

```
    df10 <- rbind(df10, a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-1,]
  return(df10)
}


# ----------------------------------------------------------------
# ----------- CHAPTER 5 ----------------------------------------
# ----------------------------------------------------------------
#  ------ Arima Ann Predicting Up/Dn - Categorical -----------------
# a. Categorical
ts_4_fnc_ar <- function(fil,SLoss,nm){

  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i],stringsAsFactors=F)
    Mkt_p <- Mkt[,c(1,2,3,4,5)]
    Mkt_p$pred <- Mkt$pred
    colnames(Mkt_p) <- c("Date","Open", "High","Low","Close","pred")
    a <- ts_4(Mkt_p, SLoss,nm[i])
    df10 <- rbind(df10, a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-c(1),]
  return(df10)
}


# ------------------------------------------------
#  ------ Arima Ann Predicting Up/Dn - 01 ---------
ts_3_fnc_ar <- function(fil,nm,ts1){
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i],stringsAsFactors=F)
    Mkt_p <- Mkt[,c(1,2,3,4,5,18)]
    colnames(Mkt_p) <- c("Date","Open", "High","Low","Close","p")
    a <- ts_3(Mkt_p, 0, nm[i])
    df10 <- rbind(df10, a)
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-c(1),]
  return(df10)
}

# bit of fiddling for ANN
ts_3a_fnc_ar <- function(fil,nm,ts1){
  for(i in 1:length(fil)){
    Mkt <- read.csv(fil[i],stringsAsFactors=F)
    Mkt_p <- Mkt[,c(1,2,3,4,5,18)]
    colnames(Mkt_p) <- c("Date","Open", "High","Low","Close","p")
    a <- ts_3a(Mkt_p, 0, nm[i])
    df10 <- rbind(df10, a)
```

```
  }
  df.name <- names(a)
  names(df10) <- df.name
  df10 <- df10[-c(1),]
  return(df10)
}
```

RCode/Utils.R

Appendix B

Technical Indicators

B.1 Moving Average Convergence Divergence (MACD)

MACD is a widely used technical indicator which attempts to detect the early stage of a market trend. It is calculated by subtracting a long exponential moving average (EMA) from a shorter one. The EMA is calculated as follows:

$$EMA(n)_t = \frac{2}{n+1}(P_t - EMA_{t-1}) + EMA_{t-1}$$

Where P_t is the closing price of a market on day t and n is the number of periods used in calculating the moving average. MACD itself is calculated as:

$$MACD_t = EMA(s)_t - EMA(l)_t$$

where $EMA(s)_t$ is the short moving average and $EMA(l)_t$ is the long one. In addition an EMA of the MACD itself is calculated in order to generate trade signals and is often referred to as the "trigger line". Thus a particular MACD trading rule is often expressed in the form $MACD(s, l, k)$ where s is the number of periods of the short EMA, l the number of periods of the long EMA and k the period used to average the MACD for the trigger line.

B.2 Aroon Indicator

TTR: Aroon up (down) is the elapsed time, expressed as a percentage, between today and the highest (lowest) price in the last n periods. If today's price is a new high (low)

Aroon up (down) will be 100. Each subsequent period without another new high (low) causes Aroon up (down) to decrease by (1 / n) x 100.

http://www.fmlabs.com/reference/default.htm?url=Aroon.htm: The word aroon is Sanskrit for "dawn's early light." The Aroon indicator attempts to show when a new trend is dawning. The indicator consists of two lines (Up and Down) that measure how long it has been since the highest high/lowest low has occurred within an n period range.

When the Aroon Up is staying between 70 and 100 then it indicates an upward trend. When the Aroon Down is staying between 70 and 100 then it indicates an downward trend. A strong upward trend is indicated when the Aroon Up is above 70 while the Aroon Down is below 30. Likewise, a strong downward trend is indicated when the Aroon Down is above 70 while the Aroon Up is below 30. Also look for crossovers. When the Aroon Down crosses above the Aroon Up, it indicates a weakening of the upward trend (and vice versa).

http://www.linnsoft.com/tour/techind/aroon.htm: Chande states that when AroonUp and AroonDown are moving lower in close proximity, it signals a consolidation phase is under way and no strong trend is evident. When AroonUp dips below 50, it indicates that the current trend has lost its upwards momentum. Similarly, when AroonDown dips below 50, the current downtrend has lost its momentum. Values above 70 indicate a strong trend in the same direction as the Aroon (up or down) is under way. Values below 30 indicate that a strong trend in the opposite direction is underway.

The Aroon Oscillator signals an upward trend is underway when it is above zero and a downward trend is underway when it falls below zero. The farther away the oscillator is from the zero line, the stronger the trend.

$$AroonUp = 100 * \left(\frac{n - PeriodSinceHighestHigh}{n} \right)$$

$$AroonDown = 100 * \left(\frac{n - PeriodSinceLowestLow}{n} \right)$$

B.3 Parabolic Stop-and-Reverse (SAR)

The Parabolic Stop-and-Reverse (SAR) is a quite complex indicator developed by Welles Wilder in 1978 (Wilder, 1978). The calculation for SAR in rising and falling markets are different and are usually presented separately.

If the market is rising SAR is calculated as:

$$\text{Current SAR} = \text{Prior SAR} + \text{Prior AF(Prior EP - Prior SAR)}$$

where:

- Prior SAR is the SAR value for the previous time period, for example the previous day's value.
- Extreme Point (EP) is the highest high of the current trend.
- Acceleration Factor (AF) starts at 0.02, and increases by 0.02 each time the market makes a new high (Extreme Point). The maximum value the AF can reach is 0.20, at which point it is capped.

Note: SAR can never be greater than the value of the previous two periods' lows. Should SAR be above one of those lows, it is set to the lowest of the two.

If the market is falling SAR is calculated as:

$$\text{Current SAR} = \text{Prior SAR - Prior AF(Prior SAR - Prior EP)}$$

Note: SAR can never be less than the value of the previous two periods' highs. Should SAR be less than one of those highs, it is set to the lowest of the two.

B.4 Stochastic

The stochastic oscillator measures where a particular close price is in relation to the highest high and lowest low in the range under study. It is usually drawn on a chart as two lines, one is %K and the other is its moving average usually called %D.

The calculation of the stochastic involves four variables:

1. %K Period - the number of periods used in the calculation (see below).
2. %K Slowing Period - smoothing period applied to %K.
3. %D Period - the number of time periods used in the moving average of %K to generate %D.
4. %D Method - the moving average method used to calculate %D.

%K is calculated as follows:

$$\%K = 100 * \left(\frac{\text{Today's Close} - \text{Lowest Low in n Periods}}{\text{Highest High in n Periods} - \text{Lowest Low in n Periods}} \right)$$

The stochastic is used in a variety of ways. One popular method is to buy when the stochastic falls below a particular level then rises back above that level (and vice versa for a short trade). An alternative technique is to buy when the %K rises above %D and sell when it falls under %K.

B.5 Rate of Change(ROC)

The Rate of Change or ROC indicator highlights the difference between a particular price (e.g. closing price) and the same price a number of periods previously. This value can be expressed in absolute terms or a percentage rise or fall. The calculation is as follows:

$$ROC = 100 * \left(\frac{\text{Today's Close} - \text{Today's Close n Periods Ago}}{\text{Today's Close n Periods Ago}} \right)$$

The ROC can be calculated from a wide range of time periods, with 12 and 25 days being the most common. The ROC is typically used as an over-bought / over-sold indicator to provide evidence for when a market turn maybe expected.

Bibliography

Aroon. http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:aroon. Accessed April 22, 2014.

Steven B. Achelis. Macd, 2014. URL http://www.metastock.com/Customer/Resources/TAAZ/default.aspx?c=3&p=66.

G. Appel. *Technical Analysis: Power Tools for Active Investors : [new Techniques for Active Trading in the Stock Market!]*. Financial Times Prentice Hall books. Financial Times/Prentice Hall, 2005. ISBN 9780131479029.

G. Appel and E. Dobson. *Understanding Moving Average Convergence-divergence*. Traders Press, 2007. ISBN 9780934380928.

George S. Atsalakis and Kimon P. Valavanis. Surveying stock market forecasting techniques – part ii: Soft computing methods. *Expert Systems with Applications*, 36(3): 5932 – 5941, 2009.

C.L. Beau and D.W. Lucas. *Day Trading Systems and Methods*. Marketplace Books, 1999. ISBN 9781883272272.

S.W. Bigalow. *Profitable Candlestick Trading: Pinpointing Market Opportunities to Maximize Profits*. Wiley Trading. Wiley, 2011. ISBN 9781118005804.

G.E.P. Box and G.M. Jenkins. *Time series analysis: forecasting and control*. Holden-Day, 1970.

William Brock, Josef Lakonishok, and Blake LeBaron. Two simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance*, 47(5):1731–64, 1992.

R.G. Brown. *Statistical forecasting for inventory control*. McGraw-Hill, 1959.

R.G. Brown. *Smoothing, forecasting and prediction of discrete time series*. Prentice-Hall, 1963.

T.N. Bulkowski. *Encyclopedia of Chart Patterns*. Wiley Trading. Wiley, 2011. ISBN 9781118045855.

N. Chopra, J. Lakonishok, and J. Ritter. Measuring abnormal performance do stocks overreact? *Journal of Financial Economics*, 31:235 – 268, 1992.

Avril Coghlan. *Little Book of R for Time Series!* URL http://http://a-little-book-of-r-for-time-series.readthedocs.org/en/latest/.

M.W. Covel. *Trend Following (Updated Edition): Learn to Make Millions in Up or Down Markets*. Pearson Education, 2009. ISBN 9780135094402.

Thaler R. De Bondt, W. Does the stock market overreact? *Journal of Finance*, 40(0): 793 – 805, 1985.

John Devcic. Weighted moving averages: The basics, 2010. URL http://www.investopedia.com/articles/technical/060401.asp. [Online; accessed 1-March-2014].

M. Dueker and C.J. Neely. Can markov switching rules predict foreign exchange returns? *Journal of Banking and Finance*, 31:297 – 296, 2007.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 71:1–26, 1979.

C. Faith. *Way of the Turtle: The Secret Methods that Turned Ordinary People into Legendary Traders*. McGraw Hill professional. McGraw-Hill Education, 2007. ISBN 9780071509466.

Samreen Fatima and Ghulam Hussain. Statistical models of kse100 index using hybrid financial systems. *Neurocomputing*, 71(13–15):2742 – 2746, 2008.

Geoffrey C. Friesen, Paul A. Weller, and Lee M. Dunham. Price trends and patterns in technical analysis - a theoretical and empirical examination. *Journal of Banking and Finance*, 33(6):1089 – 1100, 2009.

Jan G. De Gooijer and Rob J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443 – 473, 2006.

Kelley E. Gutierrez Jr, R. The long-lasting momentum in weekly returns. *Journal of Finance*, 63:415 – 447, 2008.

J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques: Concepts and Techniques*. Elsevier Science, 2011. ISBN 9780123814807.

Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5 – 10, 2004.

Chih-Ming Hsu. A hybrid procedure for stock price prediction by integrating self-organizing map and genetic programming. *Expert Systems with Applications*, 38(11): 14026 – 14036, 2011.

Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Computers and Operations Research*, 32 (10):2513 – 2522, 2005.

Rob J Hyndman and Yeasmin Kh. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, 2008.

N. Jegadeesh. Evidence of predictable behavior of security returns. *Journal of Finance*, 45:881 – 898, 1990.

N. Jegadeesh and S. Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance*, 48:65 – 91, 1993.

N. Jegadeesh and S. Titman. Overreaction, delayed reaction and contrarian profits. *Review of Financial Studies*, 8:973 – 993, 1995.

P.J. Kaufman. *Trading Systems and Methods*. Wiley, 2013. ISBN 9781118222249.

Mehdi Khashei, Mehdi Bijari, and Gholam Ali Raissi Ardali. Improvement of auto-regressive integrated moving average models using fuzzy logic and artificial neural networks (anns). *Neurocomputing*, 72(4–6):956 – 967, 2009.

Thomas Kriechbaumer, Andrew Angus, David Parsons, and Monica Rivas Casado. An improved wavelet–arima approach for forecasting metal prices. *Resources Policy*, 39 (0):32 – 41, 2014.

G.C. Lane. *Using Stochastics, Cycles and R.S.I.* 1986.

B. Lehmann. Fads, martingales and market efficiency. *Quarterly Journal of Economics*, 105:1 – 28, 1990.

Friedrich Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. In Wolfgang Härdle and Bernd Rönz, editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, Heidelberg, 2002. URL http://www.stat.uni-muenchen.de/~leisch/Sweave. ISBN 3-7908-1517-9.

R. Levich and L. Thomas. The significance of technical trading rule profits in the foreign exchange market: A bootstrap approach. *Journal of International Money and Finance*, 12:451 – 474, 1993.

B.G. Malkiel. *A Random Walk Down Wall Street: Including a Life-cycle Guide to Personal Investing*. Norton, 1999. ISBN 9780393047813.

Ben R. Marshall, Martin R. Young, and Lawrence C. Rose. Candlestick technical trading strategies: Can they create value for investors? *Journal of Banking and Finance*, 30 (8):2303 – 2323, 2006.

Lukas Menkhoff. The use of technical analysis by fund managers: International evidence. *Journal of Banking and Finance*, 34(11):2573 – 2586, 2010.

T.C. Mills. *The Foundations of Modern Time Series Analysis*. Palgrave Macmillan, 2011. ISBN 9780230290181.

G.L. Morris. *Candlestick Charting Explained: Timeless Techniques for Trading stocks and Sutures*. McGraw-Hill Education, 2006. ISBN 9780071632171.

J.J. Murphy. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. Penguin Group US, 1999. ISBN 9781101659199.

C. J. Neely, P. Weller, and R. Dittmar. Is technical analysis in the foreign exchange market profitable? a genetic programming approach. *Journal of Financial and Quantitative Analysis*, 32:405 – 426, 1997.

S. Nison. *Japanese Candlestick Charting Techniques: A Contemporary Guide to the Ancient Investment Techniques of the Far East*. New York Institute of Finance, 2001. ISBN 9780735201811.

Keith Ord. Charles holt's report on exponentially weighted moving averages: an introduction and appreciation. *International Journal of Forecasting*, 20(1):1 – 3, 2004.

Ping-Feng Pai and Chih-Sheng Lin. A hybrid arima and support vector machines model in stock price forecasting. *Omega*, 33(6):497 – 505, 2005.

Guoqi Qian and Xindong Zhao. On time series model selection involving many candidate arma models. *Computational Statistics and Data Analysis*, 51(12):6180 – 6196, 2007.

I. Rojas, O. Valenzuela, F. Rojas, A. Guillen, L.J. Herrera, H. Pomares, L. Marquez, and M. Pasadas. Soft-computing techniques and arma model for time series prediction. *Neurocomputing*, 71(4-6):519 – 537, 2008.

Minakhi Rout, Babita Majhi, Ritanjali Majhi, and Ganapati Panda. Forecasting of currency exchange rates using an adaptive arma model with differential evolution based training. *Journal of King Saud University - Computer and Information Sciences*, 26 (1):7 – 18, 2014.

J.D. Schwager. *Market Wizards: Interviews With Top Traders*. HarperCollins, 1988. ISBN 978-0887306105.

J.D. Schwager. *The New Market Wizards: Conversations with America's Top Traders*. HarperBusiness, 1994. ISBN 978-0887306679.

R.H. Shumway and D.S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer, 2010. ISBN 9781441978653.

Andrei Shynkevich. Performance of technical analysis in growth and small cap segments of the us equity market. *Journal of Banking and Finance*, 36(1):193 – 208, 2012.

Mike K.P. So and Ray S.W. Chung. Dynamic seasonality in time series. *Computational Statistics and Data Analysis*, 70(0):212 – 226, 2014.

R. Sweeney. Beating the foreign exchange market. *Journal of Finance*, 41:163 – 182, 1986.

Zhongfu Tan, Jinliang Zhang, Jianhui Wang, and Jun Xu. Day-ahead electricity price forecasting using wavelet transform combined with arima and garch models. *Applied Energy*, 87(11):3606 – 3610, 2010.

Mieko Tanaka-Yamawaki and Seiji Tokuoka. Adaptive use of technical indicators for the prediction of intra-day stock prices. *Physica A: Statistical Mechanics and its Applications*, 383(1):125 – 133, 2007.

Numan Ulku and Eugeniu Prodan. Drivers of technical trend-following rules profitability in world stock markets. *International Review of Financial Analysis*, 30:214 – 229, 2013.

R.L. Weissman. *Mechanical Trading Systems: Pairing Trader Psychology with Technical Analysis*. Wiley, 2005. ISBN 9780471654353.

J.W. Wilder. *New concepts in technical trading systems*. Trend Research, 1978. ISBN 9780894590276.

L. Williams. *The definitive guide to futures trading*. Number v. 2 in The Definitive Guide to Futures Trading. Windsor Books, 1989. ISBN 9780930233365.

L. Williams. *Long-Term Secrets to Short-Term Trading*. Wiley Trading. Wiley, 2011. ISBN 9781118184684.

Peter R. Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342, 1960.

G.Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50(0):159 – 175, 2003.