Master Thesis

Institute of Human Resource Management and Organizations

Hamburg University of Technology (TUHH)

Forecasting and time series analysis: methods and application in business research

by

Jayesh Raghavendra Bhat

Matriculation number: 48810

Address: Marienstraße 90, 21073, Hamburg

E-mail: jayesh.r.bhat@gmail.com

Telephone number: +49-176-41786041

Examiner and Supervisor: Prof. Dr. Christian M. Ringle

Hamburg, 16/01/2019

Abstract

Research in forecasting has evolved at a tremendous pace since the latter half of this century. A proliferation of methods from multiple disciplines has been detected. But despite all the effort invested in research till date, there still seems to be an increasing gap between the advancements made in research and their transition into practical business applications. This thesis aims at narrowing the gap by defining a systematic procedure for forecasting, which can be applied to discrete univariate time series dataset observed in day-to-day business processes. Useful methods for conducting data analysis are proposed and the key criteria for selecting quantitative forecasting methods are recommended. Commonly used forecasting methods are discussed by maintaining a distinction between heuristic methods and advanced statistical models. An attempt is also made to capture the relationship between state-of-the-art research and well-established methods in this field of research. A robust procedure for the evaluation of forecast accuracy will be demonstrated using simple summary statistics and benchmarks. Due to its increasing popularity, the use of artificial neural networks for forecasting is also introduced. All demonstrations and visualizations are created using the open source statistical software R.

Keywords: time series analysis, quantitative forecasting methods, forecasting methodology, forecast accuracy, ARIMA, ETS, naïve benchmark, time series decomposition, R, summary statistics, neural networks, business forecasting, out-of-sample-forecasts, time series cross-validation.

List of Abbreviations

ACF	Autocorrelation Function
ANN	Artificial Neural Nets
AR	Auto-regressive
ARIMA	Autoregressive Integrated Moving Average
CI	Computational Intelligence
ETS	Error, Trend, Seasonality
GUI	Graphical User Interface
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
ME	Mean Error
ML	Machine Learning
MSE	Mean Squared Error
NNAR	Autoregressive Neural Nets
PACF	Partial Autocorrelation function
PE	Percentage Error
RW	Random Walk
SSE	Sum of Squared Errors
STL	Seasonal and Trend decomposition using Loess

List of Symbols

t	Time period
Y_t	Actual value at time t
S_t	Seasonal component (or seasonal index) at time t
T_t	Trend-cycle component at time t
E_t	Irregular or random (error) component at time t
λ	Box-Cox transformation parameter
y	Time series
$\hat{y}$	Forecast
$y_1 \dots y_T$	Past observations
$\hat{y}_{T+h\|T}$	Forecast (y) made using actual observation y_T at time t, for $T+h$ time steps ahead
y_1	First observation of a time series
y_T	Most recent or last observation in a time series
T	Number of observations
h	Forecast horizon
m	Seasonal period; frequency of seasonality
k	Number of complete years in the forecast period prior to time $T+h$
α	Smoothing parameter for the level component
l_t	Unobserved level component (or smoothed value) of the series at time t
b_t	Trend estimate (slope) at time t
β^*	Smoothing parameter for the trend
δ	Damping parameter
s_t	Seasonal component
γ	Smoothing parameter for seasonality
e_t	Residual at time t; also represented by $\varepsilon \sim NID(0, \sigma^2)$ or white noise
L	Likelihood of the model
K	Total number of estimated parameters and initial states
$\bar{y}$	Mean

k	Number of lags
ϕ	AR model parameter
$\acute{y}_t$	Differenced series
θ	MA model parameter
p, P	Number of lagged observations in non-seasonal and seasonal AR models
q, Q	Number of error terms in non-seasonal and seasonal MA models
d, D	Number of differencing needed to achieve stationarity in an ARIMA models
$w_{i\ j}$	Weights linearly combined with the inputs in a neural network
b_j	Bias induced in the neural network
z_j	Sum of the linear combination and bias b_j
$s(z)$	Activation function

List of Figures

List of Tables

Table of contents

Chapter 1. Introduction

We make numerous decisions every day, some are simple and can be made in a matter of few seconds, on the other hand, there are those that heavily impact our lives and might change the direction in which our lives are headed. The latter involves decisions that we struggle to take since there is always a certain level of risk and uncertainty attached to them. In such cases, we usually tend to mitigate the uncertainty by gathering information and making elaborate plans pertaining to the choices we have at our disposal. Only after a thorough assessment of the risks and evaluation of all perceivable alternatives, we are finally satisfied and find the confidence to make that important decision. But even after all that effort, we cannot be entirely sure about whether the right choice was made and sometimes we end up with unanticipated results. From an organizational perspective, managers and executives face a similar dilemma while taking decisions related to their work. Most of these decisions affect the entire company and usually involve multiple stakeholders. Due to the high stakes involved, it is critical to get it right the first time and to avoid major pitfalls that would ultimately result in massive losses to the organization. This urge to make better decisions is satisfied by planning. Planning helps managers and executives to systematically assess the situation and consider all relevant factors that influence it.

It is a well-known fact that meticulous planning always precedes decision making. Important decisions usually demand a considerable amount of time and effort in making plans that range from a few days, weeks, months or sometimes years depending on the organizational needs and objectives. However, planning depends extensively on the availability of information, which can be found externally and within the organization. Even though planning is usually oriented towards the future, it requires information that already exists.

Due to the enormous amount of information out there it becomes increasingly important to find the right information from the overabundance of data out there. In other words, we must separate the signal from the noise. Particularly we must look for information that aligns well with the planning needs of the organization. This usually demands some sort of detailed analysis to isolate the useful information from the irrelevant. After filtering out the signal from the noise, the existing patterns and relationships become clearer. The resulting signal may either take the form of an underlying pattern that repeats itself or it may indicate an underlying relationship between variables. However, to prove beneficial to planning, these observed patterns and deduced relationships need to be projected to the future. The systematic procedure that extends these patterns and relationships into the future is known as forecasting. Forecasting and planning are closely associated (Armstrong, 1985; Ord and Fildes, 2013, pp. 14-15; Makridakis, 1996) and when performed well, can lead to effective decision making. This paper will discuss the process of forecasting and the accompanying data analysis procedures in greater detail.

Forecasting, unlike fortune telling and unaided subjective judgments, is a methodical procedure to make predictions. It is a systematic process based on objective evidence and concrete empirical results (Makridakis and Wheelwright, 1989, pp. 12-13). The field of forecasting is vast and has a wide range of applications that stem from different fields of science and management. Moreover, over the course of history, there has been a tremendous amount of research conducted in this field. Many novel techniques have been formulated (Ord and Fildes, 2013) and due to its potential in dealing with risks and uncertainty (Makridakis, Spiliotis and Assimakopoulos, 2018), it has been a popular topic of discussion in many areas of business research. This paper investigates the business implications of forecasting by mainly dwelling on the nature of time series data observed in business and the opportunities it offers for forecasting.

One may be tempted to believe that after all these years of research, this abundance of theoretical findings to have made a smooth transition into practical business applications. Unfortunately, this is not the case when we look at how forecasting is actually practiced in business today. It has been found that there still exists a gap between theoretical findings and practical applications (Morlidge and Player, 2010, pp. 12-13). A major barrier to the smooth transition of research findings to business application is not just the abundance of information, but also a large amount of variability present in the research. Generalizing from existing literature becomes difficult when the papers vary in scope, data frequencies, modelling frameworks, forecast horizons, estimation techniques, results and conclusions (Athanasopoulos *et al.*, 2011). Furthermore, employees in many firms still use outdated forecasting methods. Businesses are either unaware of the latest developments in this field or find it too complicated to understand (Gilliland, Tashman and Sglavo, 2015). To make matters worse, the turbulent nature of the business environment makes it difficult to forecast and has often led to skepticism (Makridakis and Wheelwright, 1989, pp. 3-4). On the other end of the spectrum, however, problems such as the lack of accuracy and large forecasting errors still prevail despite using sophisticated state of the art forecasting methods (Makridakis and Hibon, 2000). Furthermore, in today's business environment, it is not enough to just find the right solution to a forecasting problem. For a business to remain competitive in such an environment it must constantly strive to improve and find new ways to solve problems faster and in a cost-effective manner. This means that automatic and inexpensive forecasting is of prime importance for business applications today (Athanasopoulos *et al.*, 2011).

This thesis aims to address these problems by fulfilling the following objectives.

- Presentation of research that is up to date and relevant to today's business scenario.
- Definition of a methodology for forecasting and time series analysis.
- Recommendation of tools for time series data visualization to improve data analysis and forecasting.
- Demonstration of popular heuristic methods and statistical models for conducting time series analysis and forecasting.
- A brief overview of the application of advanced machine learning algorithms like the neural networks to forecasting.
- Showcase of the right way to measure forecast accuracy.
- Proof that forecasting can also be automated and performed inexpensively using the free open-source statistical software called R.

Due to the vastness of this field it is not possible to cover the entire range of topics and methods that forecasting brings along with it. To comply with research specifications, this paper will restrict its scope to forecasting only univariate time series data with equally spaced intervals and discrete data points. Furthermore, the forecasts will be limited to forecasting methods that work well for forecasting within the short-medium term horizon. The effectiveness and accuracy of these methods decrease as the forecast horizon increases. This paper strives to present its findings using a clear and simple approach. Hence, detailed mathematical proofs are avoided and focus is mainly given to understanding the basic underlying concept. However, in some sections, it was necessary to provide mathematical definitions to aid a better understanding of some concepts. For those interested in investigating deeper into the mathematical details, references are provided wherever necessary.

This thesis is divided into seven chapters that mainly focus on the topic of time series analysis and forecasting. Chapter 1 starts by laying the foundation for this thesis. It consists of the introduction, problem statement, scope and few definitions of important terms used in this field. Chapter 2 gives an overview of forecasting applications in business. It includes the classification of quantitative forecasting methods and talks about factors that affect method selection while operating in a business scenario. It also informs about the various forecasting pre-requisites which need to be covered and recommends a systematic methodology for conducting forecasting in an organization. Chapter 3 dives into the topic of time series analysis, where different ways of analyzing time series data using the statistical software R will be shown. From Chapter 4 onwards, well-established forecasting methods are narrowed down from the methods mix. A distinction will be made between a heuristic method and a statistical model; their interrelationships studied, and current state-of-the-art methods based on these well-established methods and models will be demonstrated using R. In Chapter 5, the methods of measuring forecast accuracy using summary statistics will be discussed. This will be followed by Chapter 6, where validation methods and accuracy measures will be put to test by demonstrating a forecasting scenario using R. Lastly, Chapter 7 concludes the thesis by summarizing the key points highlighted in this thesis and provides recommendations for further research in this field.

Before we begin, there are a few terms and concepts that need clarification since they form the basis of this research. Sometimes, we come across few terms in the forecasting literature that are used interchangeably in different contexts, hence it is of utmost importance that their differences be stated in advance to prevent any misinterpretations later.

Forecasting vs. Prediction

Usually, the terms ‘prediction’ and ‘forecasting’ are used interchangeably in many sources and this has led to them being considered similar in meaning. However, they do possess a subtle difference that is important to know. According to Brown (2004, pp.467-468) prediction involves subjective estimations whereas forecasting consists of objective estimations. Prediction can be used while estimating something novel, e.g. the impact of a new product on the market. In contrast, forecasting requires the possession of some prior information which is then projected or “thrown” into the future. This prior information can be any type of past data like sales, demand, production or prices, which can then be projected into the future using a systematic procedure. Hence, it must be noted that the central theme of this paper is forecasting and should not be confused with prediction.

Cross-sectional vs. Time series data

When the data are collected consistently for the same variable over a period, let's say - monthly, quarterly or yearly, it is called time series data. Most data found in business applications can be classified as time series data. A good example of a time series could be the data collected monthly for maintaining a company's inventory level. On the other hand, when data are collected for multiple variables at the same point in time, they are referred to as cross-sectional data. For example, if we collect data for variables like sales volume, sales revenue, number of customers and expenses for the past month only, this would constitute cross-sectional data (Evans, 2003, pp.30-31). It is important to note that this paper will only deal with time series data and discuss forecasting methods that accompany them.

Univariate vs. Multivariate time series

A time series is said to be univariate if it depicts the change of only one variable with time. A typical example of a univariate time series would be the monthly sales of a company, where the variable 'sales' varies each month. In contrast, a time series is said to be multivariate if it depicts the change of two or more variables with time. Usually, multivariate time series data are used for determining causality or identifying relationships between variables (Makridakis, Wheelwright and Hyndman, 1998, pp. 29-38). This paper will only use univariate data sets.

Discrete vs. Continuous data

Discrete data contains finite values that are scattered across a definite interval and have nothing in-between. In contrast, continuous data are observed in an uninterrupted endless fashion and usually contain data that can be measured, which may include fractions and decimals (Box *et al.*, 2016, pp. 21-22). The data used in this research will be discrete with observations collected at regular intervals of time.

Chapter 2. Business forecasting methods

Forecasting methods can be broadly split into two groups – quantitative (statistical) and qualitative (judgmental) methods. If the methods use past numerical data to identify and forecast existing patterns or relationships, then such methods are known as quantitative methods. Whereas, when little or no numerical data is available, but sufficient subjective knowledge exists, then forecasting is generally done using qualitative methods. Drawing conclusions purely from judgment and accumulated knowledge makes qualitative forecasting methods perform low in terms of accuracy when compared to their quantitative counterparts (Makridakis and Wheelwright, 1989, pp. 4-5). Nevertheless, these methods are quite heavily used in business since they provide an easy, intuitive and quick way to forecast. Even though these methods can be used separately, they are often used in combination with quantitative forecasting methods which make them more effective (Fildes *et al.*, 2009).

As we mentioned earlier, a higher level of accuracy clearly differentiates quantitative methods from judgmental methods. The reason for such a distinction lies in the fact that quantitative methods rely on numerical data and statistical procedures which usually tend to be precise. But this does not imply that quantitative methods are well suited to all situations that demand a high level of accuracy. In certain situations, quantitative methods perform worse and one would benefit by using judgmental methods instead. In general, recurring situations that allow the orderly collection of statistical data are ideally suited for the application of quantitative methods in order to achieve higher accuracy (Makridakis and Wheelwright, 1989, pp. 12-13). To further simplify this decision regarding whether to use quantitative methods or not, Makridakis, Wheelwright and Hyndman (1998, pp. 8-9) proposed three conditions that should be satisfied while using quantitative forecasting methods.

- Sufficient information from the past must be available.
- The information must be quantifiable into numerical data.
- The assumption of continuity - existing patterns or relationships in the data will continue into the future.

This paper will mainly focus on quantitative forecasting methods. However, those interested in learning more about qualitative forecasting methods, (Armstrong, 2001) is a comprehensive resource, in which these methods, their dependencies and their combined use with quantitative methods are discussed in more detail.

2.1 Classification of forecasting methods

Quantitative forecasting can be further subdivided according to the type of the basic characteristic that is being extrapolated from the historical data. If it involves the projection of the *underlying pattern* into the future, then it is categorized under time series forecasting. On the other hand, if it is the *relationship between variables* that need to be taken into consideration, then this type of forecasting is known as casual forecasting. To better understand these forecasting methods, let's have a look at each of their underlying concepts respectively.

Causal forecasting methods are best represented by explanatory models. This model works with the assumption that the variable to be forecast maintains an explanatory relationship with one or more independent variables. This can be further clarified using the following functional relationship.

$$Ice\ cream\ sales\ =\ f\ (climate, population, flavors, age, error) \tag{1}$$

The inclusion of the error term to the end, indicates that the relationship is not deterministic and random variation in the ice cream sales may exist. This random fluctuation is unaccounted for by the variables present in the model. This model captures the form of relationship and then uses it to forecast the future.(Makridakis, Wheelwright and Hyndman, 1998, pp. 10-11).

In contrast, time series models are normally meant for forecasting the behavior of historical patterns in time series data. This type of model ignores the relationship between the variables responsible for influencing the behavior of the system. Instead, it gives more importance to past values or errors rather than explanatory factors. These methods mainly focus on finding the underlying pattern and then extrapolating it into the future as indicated by the following example.

$$\begin{aligned} Monthly\ Ice\ Cream\ Sales\ (ICSt + 1) \\ = f\ (ICSt, ICSt - 1, ICSt - 2, ICSt - 3, \dots, error) \end{aligned} \tag{2}$$

Here, *t* indicates the present month, *t*+1 the next month, *t*-1 the previous month, *t*-2 two months ago and so on. The time series approach is used for cases where it is difficult to understand the system or when the relationships are unclear. They can also be used if the main goal of forecasting involves only knowing what will happen and not paying much attention to why it happens (Makridakis, Wheelwright and Hyndman, 1998, pp. 11-12). Note that this research will be restricted to the discussion of time series forecasting methods only. For those interested in investigating further about causal forecasting methods, the recommended source is Makridakis, Wheelwright and Hyndman (1998).

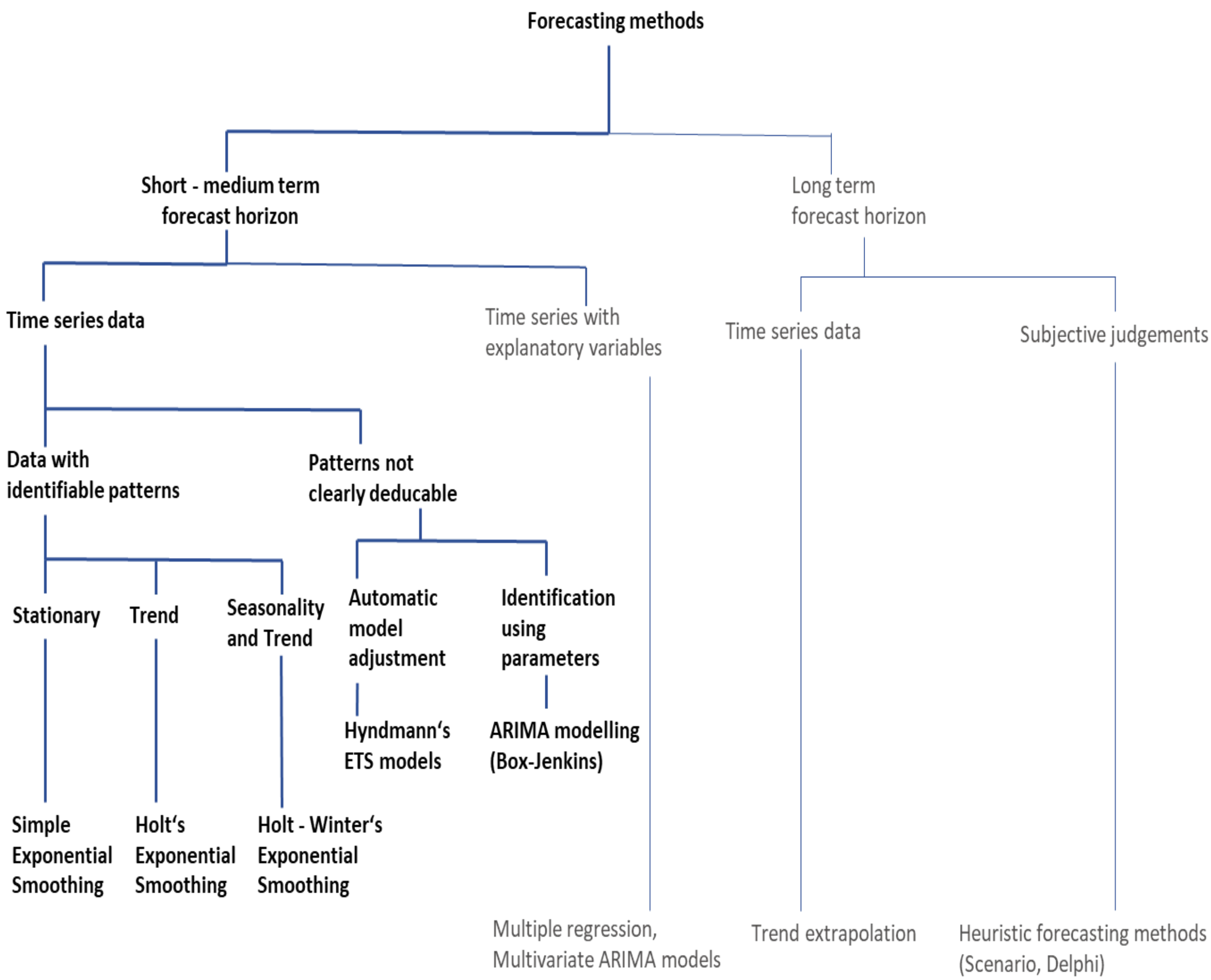


Figure 1. Forecasting methods classification tree.[1]

There are many different ways to classify forecasting methods and a number of different typologies have already been proposed by Gentry, Calantone and Cui (2006), in their research. However, the classification can be made from another perspective as well, as seen in Hansmann (1983, pp. 141-

[1] Modified from Hansmann (1983).

143). Here the methods are classified according to the length of the forecast horizon, the type of data and their underlying patterns. A modified version of this classification is presented in Figure 1.

This classification can be regarded as an extension to the scope of this research. As mentioned earlier, this paper deals with only quantitative forecasting methods, particularly those that are highlighted on the classification tree shown in Figure 1. In other words, these methods are well suited in dealing with forecasting problems in the short and medium-term horizon. Additionally, this paper will comprise of a brief introduction to forecasting using advanced machine learning models such as Artificial Neural Nets (ANN) as these methods are increasingly gaining popularity nowadays. The forecasting ability of all the discussed methods will be demonstrated by applying them to real-life datasets using the features offered by the open-source software R. The greyed-out methods as seen in the classification tree will be deliberately left out due to restrictions of scope.

2.2 Business application of forecasting

Time series forecasting can be applied in all organizations that deal with quantitative data that are collected at regular intervals while carrying out their day to day functions. To give an idea about the different forms of quantitative data and where they are collected from, Table 1 lists the typical forecasting needs of different types of organizations. From this table, it is evident that forecasting is not just restricted to one area in an organization. Almost every business sector can benefit from it. Moreover, its plethora of methods caters to all kinds of needs. However, it is also crucial to understand that certain forecasting methods do well only in particular situations. Due to the different kinds of organizational needs and the huge dissimilarities present within data, as they vary from one organizational setting to another, it is impossible to find one forecasting method that can perform accurately in all organizational scenarios (Makridakis and Wheelwright, 1989, p. 5). Hence, it is crucial to understand which method performs well in which scenario, what do the data look like in that situation and what were the forecasting needs pertaining to that forecasting problem where the methods were used. This chapter mainly focuses on discussing some of the key aspects that directly influence and if used properly, can potentially improve the method selection process during forecasting. Since the scope restricts itself to just short to medium-term forecasting and its accompanying methods during the entirety of this paper, the discussion of these factors will also be concentrated to the discussion of mainly short and medium-term forecasting only.

Organization	Forecasting needs
Retail Stores	Point of sales
Energy companies	Reserves, production, demand, prices
Educational Institutions	Enrollment
Government	Tax receipts and spending
International financial organizations	Inflation, economic activity
Transport companies	Travel behavior, demand
Banks	New home purchases
Venture capital firms	Market potential

Table 1. Typical forecasting needs of different organization types.[2]

Short-term forecasts are frequently used in production planning and scheduling, equipment, financial and personal planning, ordering of raw materials and for setting appropriate levels of inventory. Usually, a short time-frame requires using extrapolative methods like the ones discussed in this paper. Changes in the existing patterns occur rarely, making it easier for extrapolative methods to forecast accurately and in a reliable manner. Another aspect of forecasting in the short-term horizon is the simplicity involved in capturing the seasonality. Substantial research shows that seasonality does not change much at all. But when it does change, it changes less frequently and in a predictable fashion. (Makridakis, Wheelwright and Hyndman, 1998, pp. 553-554).

Forecasting is relatively easy when patterns and relationships do not change, as one would expect in short-term horizons. But while making the transition from short-term to medium-term horizons, one can witness an increase in the frequency and magnitude of such changes. Medium-term forecasts are usually needed for budgeting purposes and require estimates of sales, prices, and costs for the entire organization, as well as the divisions, geographical areas, product lines and so forth. Moreover, they sometimes require the predictions of economic and industry variables that influence company sales, price fluctuations and cash flows. Economic variables are particularly more susceptible to recessions which may result in large errors. Thus, the forecasting methods must deal with greater uncertainty that follows as the forecast horizon increases. However, due to the dynamic nature of business processes,

[2] Information adapted from Ord and Fildes (2013).

one must never forget that a slight uncertainty will always exist due to unprecedented events that are almost impossible to detect, thereby making business forecasting a stochastic process (Makridakis, Wheelwright and Hyndman, 1998, pp. 554- 557).

Forecasting pre-requisites from a business perspective

Before initiating the forecasting process, it is considered as good practice to collect information regarding certain aspects related to the forecasting problem at hand. There are some pre-requisites that should be met to prevent drastic forecasting problems in the long run. These pre-requisites are essentially important questions that need to be answered before an organization decides to set aside its budget for a project requiring the use of forecasting. The *PHIVE* approach developed by Ord and Fildes (2013), discusses such requirements of forecasting in greater detail. It is fundamentally composed of five elements – Purpose, Horizon, Information, Value, and Evaluation.

a. Purpose

As its title suggests, the organization must clearly know why they require forecasts and for what will it be used for. The definition of the problem is considered the most difficult part of forecasting, so it is worth spending time at this stage to prevent shortcomings later. Plans and their respective supporting forecasts need to be mapped in detail to make the definition of needs easier. For example, a sales forecast may be needed by different departments in the organization, each department needing a certain level of detail and accuracy. Therefore, it is also important to design the forecasting process to cater to the individual needs of that department.

b. Horizon

How far ahead into the future do we wish to forecast is another requirement that needs to be satisfied earlier. This is also more commonly referred to as the forecasting horizon. In fact, the horizon depends on the purpose of forecasting and as we have seen from the classification table in the previous section, the forecasting methods change as we move from short to long-term horizons. As a rule of thumb, short-term horizons are usually three months long, medium-term horizons lie anywhere between three months and two years. Any horizon longer than two years is considered as long-term horizon (Hansmann, 1983, pp. 12-13). But again, this can vary depending on the needs of the organization too. In general, for longer horizons, the focus shifts to discovering the causal relationships between variables rather than depending solely on pattern recognition. This happens due to the possibility that existing patterns may change over longer forecast horizons, in turn aggravating pattern recognition. In contrast, causal relationships hold for longer periods of time and even if they do change, they'll do so at a much slower and deducible pace. This signifies the importance of the forecast horizon on the accuracy of forecasts as well. Because of this one can safely say that forecast accuracy dwindles as the forecast horizon increases. The forecast horizon is sometimes also referred to as the forecasting lead time and the point where the forecast horizon starts is known as the forecast origin. This point also contains the most recent information, or in other words, the last observation of the historical data.

c. Information

In an era where information is found in abundance, it is of utmost importance to determine the type of data that forecasting methods require and to check whether the data are available in a timely manner. The quantity and quality of information greatly influence the choice of the forecasting method and eventually the accuracy of the forecast. There are usually two kinds of information – statistical (usually numerical data) and the accumulated expertise (usually subjective) obtained from employees and experts. It is good practice to acquire both types of information. Data also needs to be checked for missing values and outliers before it is used for forecasting using established data preprocessing tools. This step is important since the presence of outliers can greatly affect the forecast performance (Koehler *et al.*, 2012). The data requirements of different methods also vary. Advanced forecasting models such as neural nets that utilize machine learning algorithms usually have to be trained initially and usually demand large amounts of data. This makes them inapplicable to situations that keep changing or where large datasets are not available. (Ord and Fildes, 2013, pp. 333-334).

d. Value

The question one needs to ask here is how valuable the forecast is to the organization and how much does it cost. Is the organization willing to pay a lot for accurate information or is basic information about a scenario sufficient. The level of detail the forecasts are expected to maintain also influence the costs directly and the organization needs to be aware of this. More importantly, these peculiarities must be taken into account before initiating any forecasting tasks in order to keep the costs associated with forecasting minimal. Moreover, an organization must also be able to support its forecaster by providing a basis for obtaining such information.

e. Evaluation

Let us say that we have already satisfied the previous four factors and now have a forecasting method in place. But how do we know that this forecasting method will perform effectively. This final factor is the most important as this validates the performance of these methods. It does not matter how sophisticated the forecasting method is if it produces bad forecasts. The forecaster must test the accuracy of the methods by comparing its performance against other methods. Only then can major problems such as overfitting can be identified and avoided before it is too late. Chapter 5 is dedicated entirely to the discussion of forecast accuracy measurement techniques.

Basic steps involved in forecasting

Once the prerequisites are in place, next let us focus on how the forecasting can be performed. At this stage, the organization can begin assigning resources to this area and the forecaster can initiate the forecasting process. Being a systematic procedure, it contains a series of steps that help the forecaster reach the objective in an effective and efficient manner. Shmueli and Lichtendahl (2016, pp.16-17) propose a simple and intuitive methodology for performing forecasting as represented in Figure 2.

As stated earlier in the *PHIVE* approach, forecasting begins with defining the goal or the forecasting problem. This phase deals with setting up the pre-requisites and usually involves information regarding the purpose of the forecast, who will use the forecast in which area and how valuable it is. As mentioned earlier, this is the stage that demands a great deal of work and is considered the most difficult task of forecasting (Makridakis, Wheelwright and Hyndman, 1998, pp. 13-14). The next step concerns itself with data collection. Efforts need to be made to collect the right type of data. The data are cleaned (pre-processing) and analyzed using various visualization tools. A set of potential forecasting methods are selected according to the nature of the data. These methods are then applied and compared against a benchmark, and their performance and accuracy are evaluated using a set of accuracy measures. Consequently, the best method is chosen and implemented to generate forecasts. Note that this process is not complete once the forecasts are generated. Instead, the accuracy is continually monitored on a periodic basis and in some situations, the methods are adapted to accommodate any changes that the data may have gone through. These iterative sub-processes are clearly shown by the two sets of arrows creating the iterative loops in Figure 2.

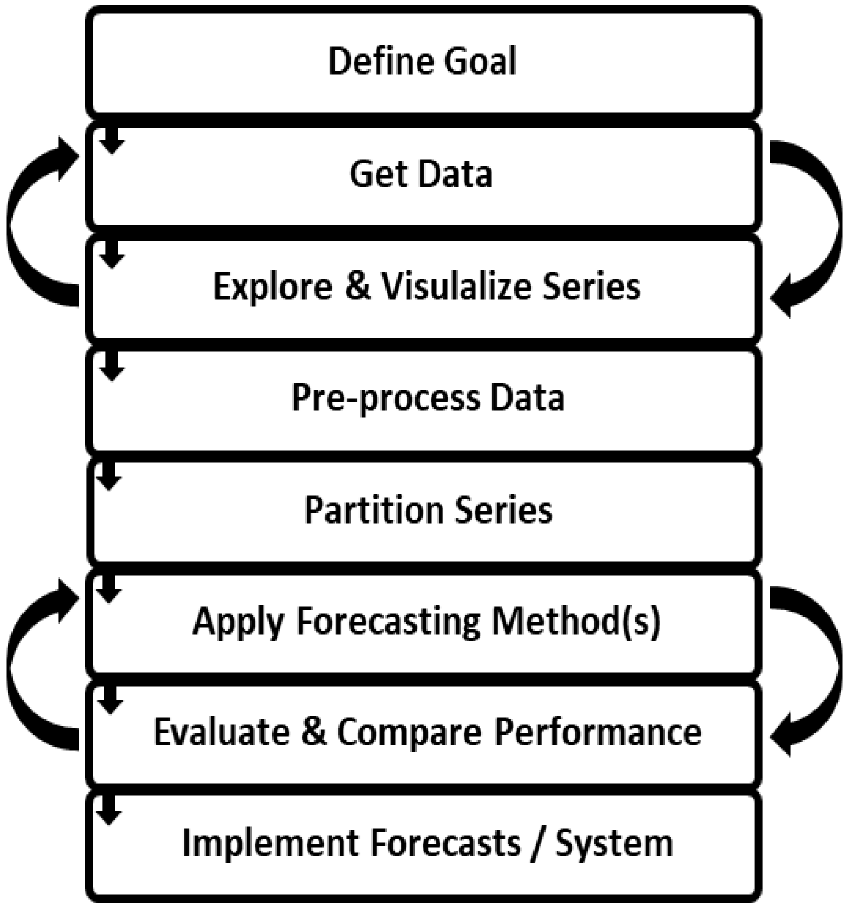


Figure 2. Basic steps involved in forecasting.[3]

2.3 Factors affecting method selection

As we have seen so far, forecasting involves a plethora of methods with a wide range of applications. But selecting a suitable method is not an easy task due to different factors affecting it, both internally and externally from an organizational perspective. Although organizational factors are important, they only indirectly influence the selection of methods. Such factors can sometimes be insidious and lie outside the control of the forecaster. Nevertheless, it is the responsibility of the top-level management to ensure that these factors do not negatively impact the forecasting process. In contrast, the factors that directly influence the forecasting method selection are more explicit and usually fall under the forecaster's control. The following sections categorize these factors based on the type of influence they have on the selection of methods.

[3] Taken from Shmueli and Lichtendahl (2016)

Factors directly influencing method selection

a. Time series characteristics

We already know by now that time series forecasting mainly deals with extrapolating the underlying pattern that exists in a data series. Often, this pattern is broken down into its components for simplifying analysis. The process of breaking down time series into its components is known as time series decomposition. This concept will be discussed in greater detail when we deal with time series analysis in the next chapter. But for now, let us focus only on the fundamental components of a time series – *Level, Trend, Cycle* (sometimes considered as a single component: trend - cycle), *Seasonality* and *Noise* (Randomness). From another perspective, a time series can also be generalized into two parts - systematic and non-systematic. Then the systematic part includes the level, trend, cycle and seasonality components. On the contrary, the non- systematic part consists of the noise or randomness. It is also common for trend and seasonality to occur in a combined form. Figure 3 shows the common modes of occurrences of trend and seasonality (Shmueli and Lichtendahl, 2016, pp. 28-29).

A series is said to be exhibiting a level pattern when the observations fluctuate in a consistent manner around a constant mean. This series is also referred to as stationary series. On the contrary, a trend component depicts long-term increase or decrease in the level of the series. It is also important to note the influence of the element of time on the occurrence of these patterns. The trend and cycle components are usually more prominent in long-term time series. However, it is quite possible for a trended series to exhibit stationarity in the short-term horizon. Seasonality exists when the rise and fall occur periodically due to some seasonal factors (month, quarter, day, week, winter/summer, day/night, etc.). Cyclic behavior, unlike seasonality, is observed when the ups and downs in the data do not follow a fixed period and have varying magnitudes. Cycles also occur in longer but unpredictable intervals of time. Lastly, the noise is the random variation that may be caused due to the measurement errors or other unknown causes. These unknown causes are mainly responsible for inducing uncertainty in the forecasts and must be accounted for during forecasting.

After decomposing of the series and identifying the respective components, the forecaster can then proceed to shortlist few forecasting methods that would best describe these components during extrapolation.

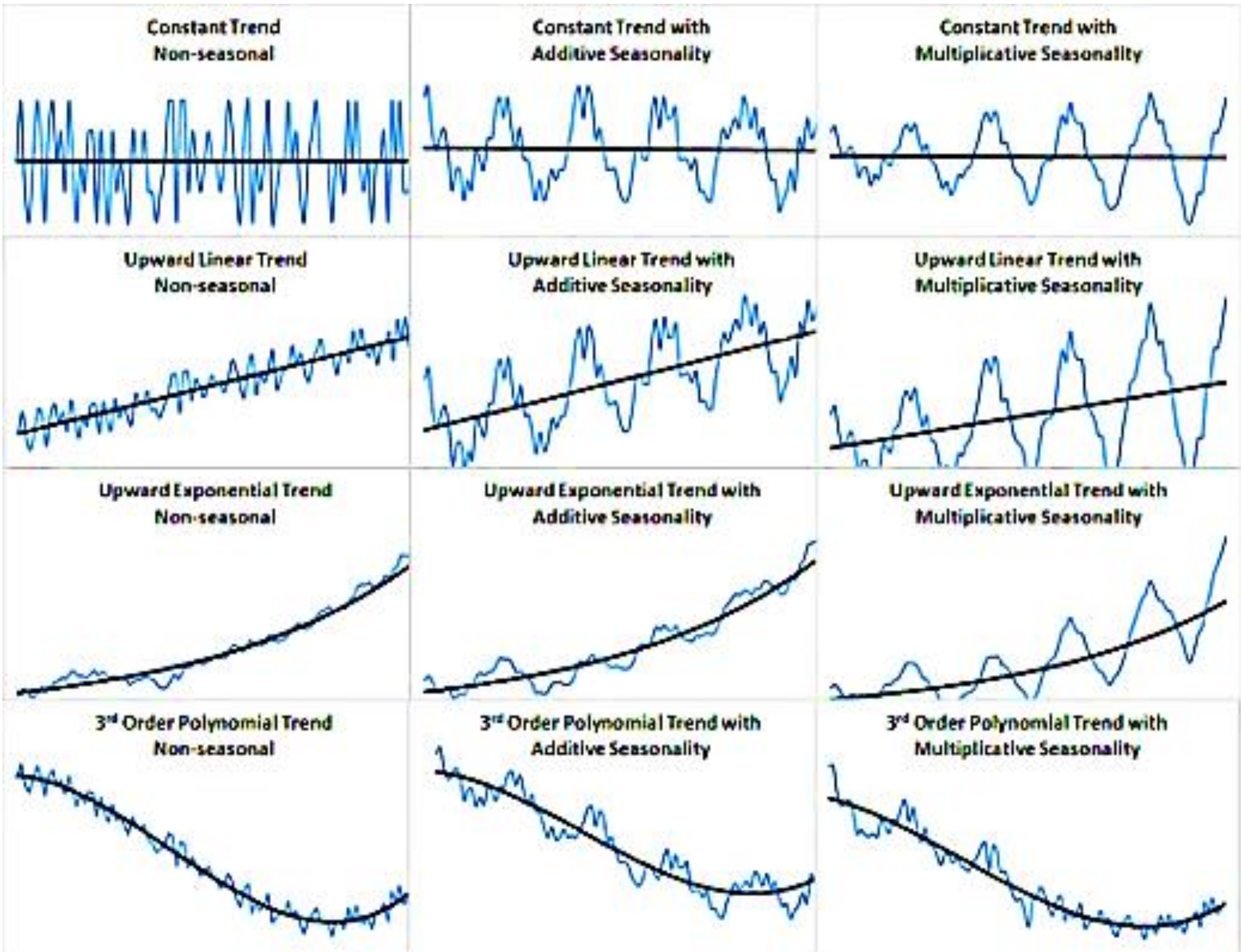


Figure 3. Common modes of occurrences of trend and seasonality.[4]

In some forecasting procedures, the seasonality is separated by using moving averages. In the case of ARMA models, it is suggested to remove seasonality before selecting the model to simplify the model selection process. The process of removal or separating seasonality is known as seasonal adjustment. Research also shows that more accurate forecasts can be obtained by forecasting after seasonally adjusting data rather than considering the seasonal element during forecasting (Makridakis *et al.*, 1993). It is also proven that a method like the Holt's exponential smoothing applied to seasonally adjusted data performs well in post sample accuracy compared to its counterpart – Holt Winter's exponential smoothing, where seasonality is considered and extrapolated directly (Hyndman, 2018).

[4] Figure taken from Shmueli and Lichtendahl (2016, p. 29).

After having dealt with seasonality, let us now shift our focus to the magnitude of randomness and behavior of the trend-cycle components. In general, it has been found that simpler methods are more suitable when the randomness is high (Makridakis *et al.*, 1993). If a high amount of randomness persists in the trend-cycles, which is typical of short-term horizons, single exponential smoothing is the best-suited method in terms of post sample accuracy due to its simplicity. But when the trend-cycle is more prominent, and randomness is low, the trend persistence becomes clear and can be extrapolated using more advanced methods.

b. Type of data

This refers to the effect of periodicity (yearly, quarterly, monthly, weekly, daily) on the data characteristics. It is important to note that the randomness decreases as the level of aggregation increases. Thus, it is evident that yearly data is less random than monthly because an average of 12 months balances all the monthly random fluctuations to zero and makes the trend more evident. Hence, a method which can clearly identify and extrapolate the trend must be chosen in this case. At the other end of the spectrum, in daily data, the randomness usually overpowers the trend. Due to this behavior, the trend is barely visible or non-existential. In such cases, the use of exponential smoothing methods is recommended. Likewise, quarterly data contain variations too, but they lie somewhere in the middle of the spectrum of randomness. Moreover, quarterly data can express high levels of seasonality and cyclic behavior too. However, with limited randomness and a moderately persistent trend, patterns belonging to this kind of series do not change very often.

c. Number and frequency of forecasts

The number of forecasts needed, and their frequency also plays an important role in determining the suitability of the methods. The number of forecasts needed increase when forecasting daily rather than monthly and decrease when moving from a monthly to a yearly basis. The number and frequency of the forecasts directly relate to the effort required for forecasting, which in turn has implications towards the cost. Thus, if the methods are complex, statistically demanding, and require a lot of effort and data to operate, they would not be considered as a reasonable choice. A good example of such a scenario can be - forecasts for maintaining inventory levels of thousands of products, which are required on a monthly or weekly basis. In such situations, the use of simpler and automated methods is recommended.

Factors indirectly influencing method selection

According to the survey by Sanders and Manrodt (2003) as cited in Ord and Fildes (2013, pp. 456-457), there are many organizational factors that influence the forecasting method selection too. These factors are depicted pictorially in Figure 4.

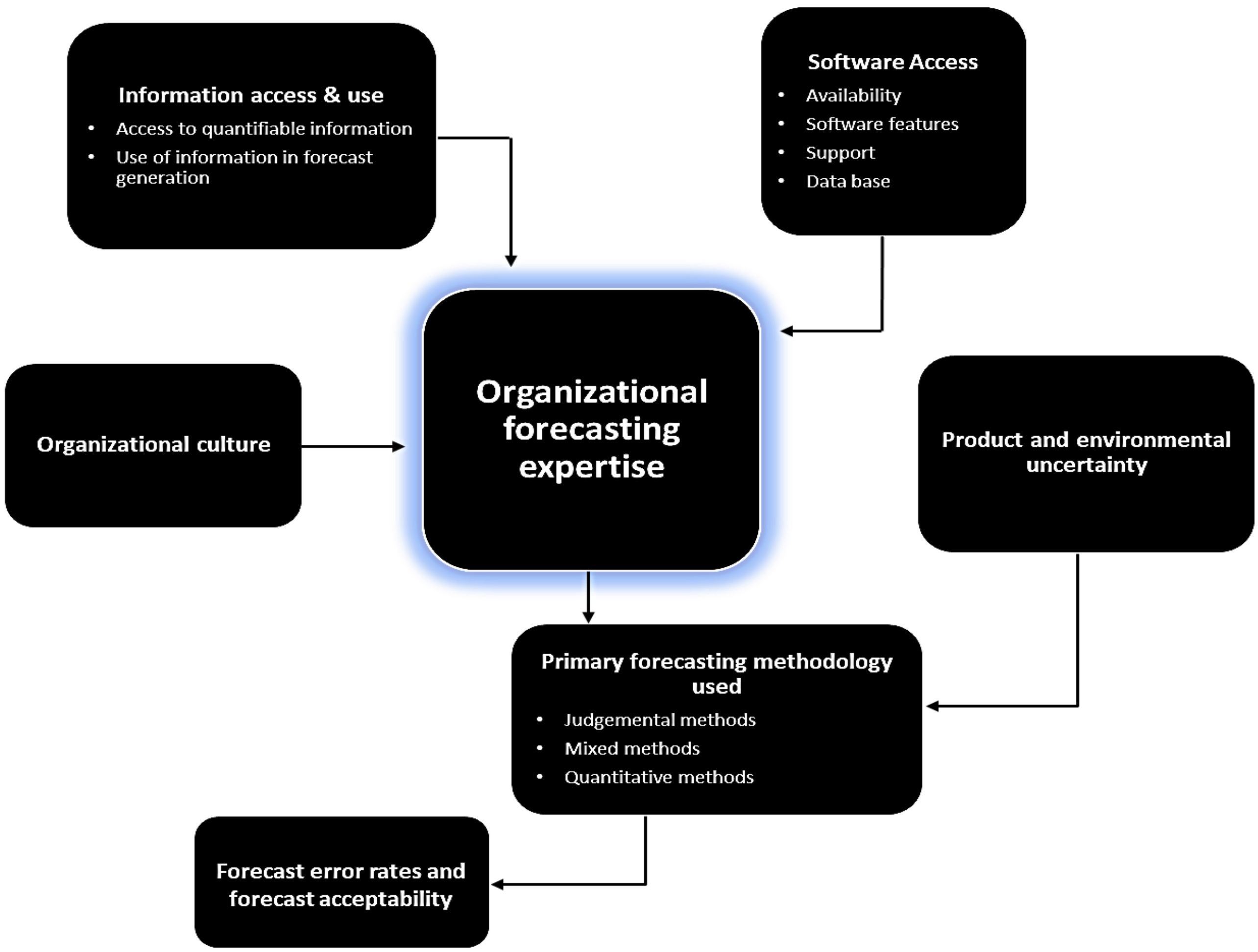


Figure 4. Organizational factors influencing method selection.[5]

Uncertainties prevalent in the market and external environment need to be identified. The preference for quantitative methods diminishes in highly dynamic environments with high market uncertainty due to the complexity it induces into the forecasting efforts. Surprisingly, the organizational culture also influences method selection drastically, but it is often ignored due to its obscurity and subjective

[5] Original source: Sanders and Manrodt (2003); as adapted by Ord and Fildes (2013).

nature. Adding transparency into the business model would greatly help. Quantitative methods are rarely adopted when the organizational paradigm distances itself from its usage. When employees show reluctance in using quantitative forecasting methods, forecasting will seem to be a burden to them rather than a useful tool (Fildes, 2015). Moreover, the organizational culture also indirectly affects the access to relevant software and the availability of data. If the organization is content with spreadsheets and ad-hoc processes based on subjective opinions, then it would not opt for heavy investments in setting up of databases, developing data pipelines and purchasing relevant software. But unfortunately, the consequences of these choices also affect the quality of forecasts, which in turn adds to the existing skepticism that still lingers in business forecasting practice today.

Chapter 3. Time series analysis

In the previous chapter, we realized that time series data and its characteristics play a critical role in the forecasting method selection procedure. In this chapter, we will discuss about the various ways to analyze the data and its underlying patterns. The subsequent sections will discuss about few useful tools that are commonly used for visualizing time series data. As a reminder, we already spoke about the various components that characterize a time series and briefly touched upon the concept of time series decomposition. These concepts are revisited in this chapter again, only this time they will be explained in more detail and their use will be demonstrated using the analytical tools offered by the R statistical software. Note that R contains a multitude of packages that enable the creation of beautiful plots and visualization. For the plots generated in this chapter, the base R and the ggplot package were used.

3.1 Data exploration

After having collected enough data, the next step in forecasting is to analyze and visualize the dataset. The forecaster's task is to identify the different features of data using the various visualization tools available and then choose a forecasting technique which best represents this pattern or characteristics exhibited by the data. We also discussed the various factors influencing the selection of a forecasting method, where it was evident that most of these factors are data related. Thus, we can imply that data analysis plays a major role in forecasting. Since time series data represent the behavior of a variable over time, the best way to analyze this is by plotting the data using graphs. This step marks the beginning of the effective data visualization process and helps us gain an intuitive understanding about data components.

<u>The time series graph (ts plot)</u>

Graphs are usually the first tool of choice when it comes to analyzing time series data. Using them allows a forecaster to visualize many subtle characteristics of the data. The selection of the type of graph depends on the type of pattern that needs to be studied. The examples in this chapter will provide a better insight into data visualization and show why data visualization should always remain the first step in any kind of data analysis.

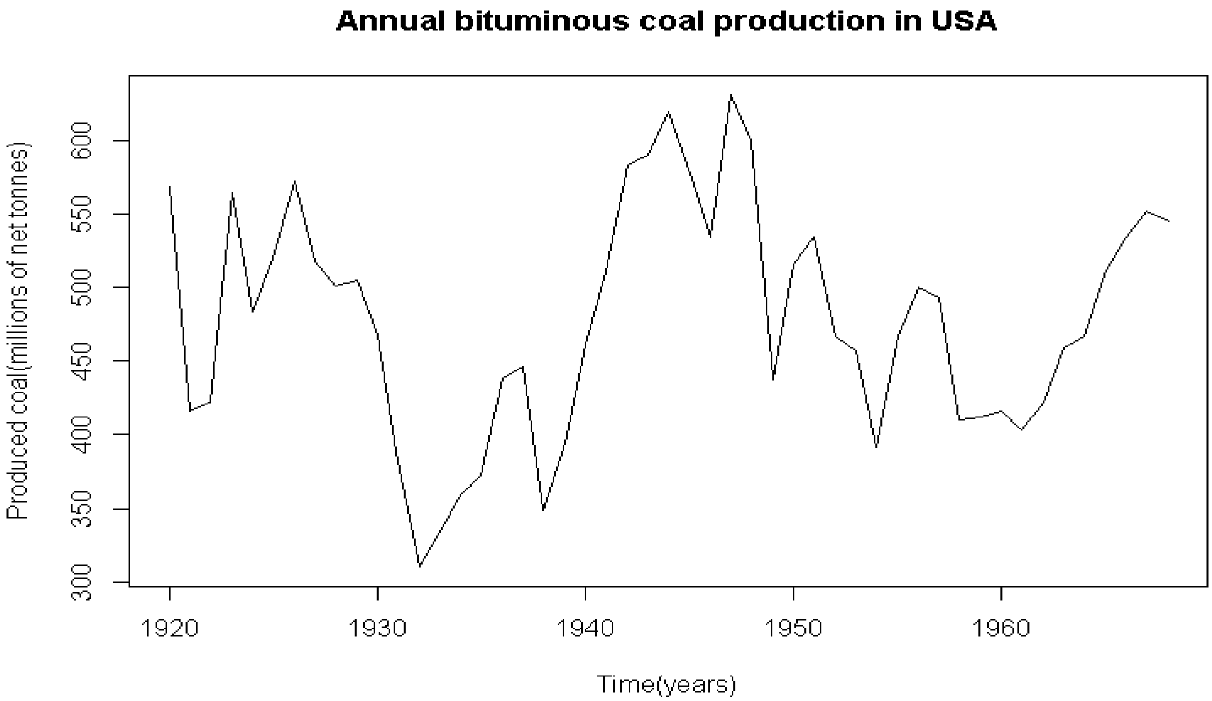


Figure 5. A typical time series plot.[6]

Figure 5 shows a typical time series plot of the annual bituminous coal production in the USA during the years (1920-1968) in millions of net tons. The procedure to upload a dataset into R and generate this plot and all subsequent plots in this chapter can be found in Appendix A1 attached at the end of the thesis. This system of representation will be followed throughout the entire paper. R supports datasets having many different formats. More details can be found in the help pages of the R documentation in their official website. Time series data are always stored as ts objects in R. Hence, it is necessary to convert all datasets after being imported into ts class objects before any analysis can be performed on them.

After the initial inspection of the generated graph, the forecaster will have already obtained a rough idea about the type of components that the time series consists. If further ambiguity persists, then the forecaster can proceed with a more detailed investigation using other plot types, which help to detect specific components. By using such plots, the presence of any time series components - seasonality, cyclic behavior, trend, stationarity or their combination can be further confirmed.

[6] Source: Taken from time series data library available at Datamarket.com.

Visual tools like the seasonal plots, subseries plots, lag plots and auto-correlograms (ACF plots) are basically plots dedicated for visualizing specific kinds of pattern. The examples chosen in Figure 6 will show the practicality and importance of using these plots for detecting patterns. The following four data sets will be used as examples; each representing a certain characteristic pattern in the time series.

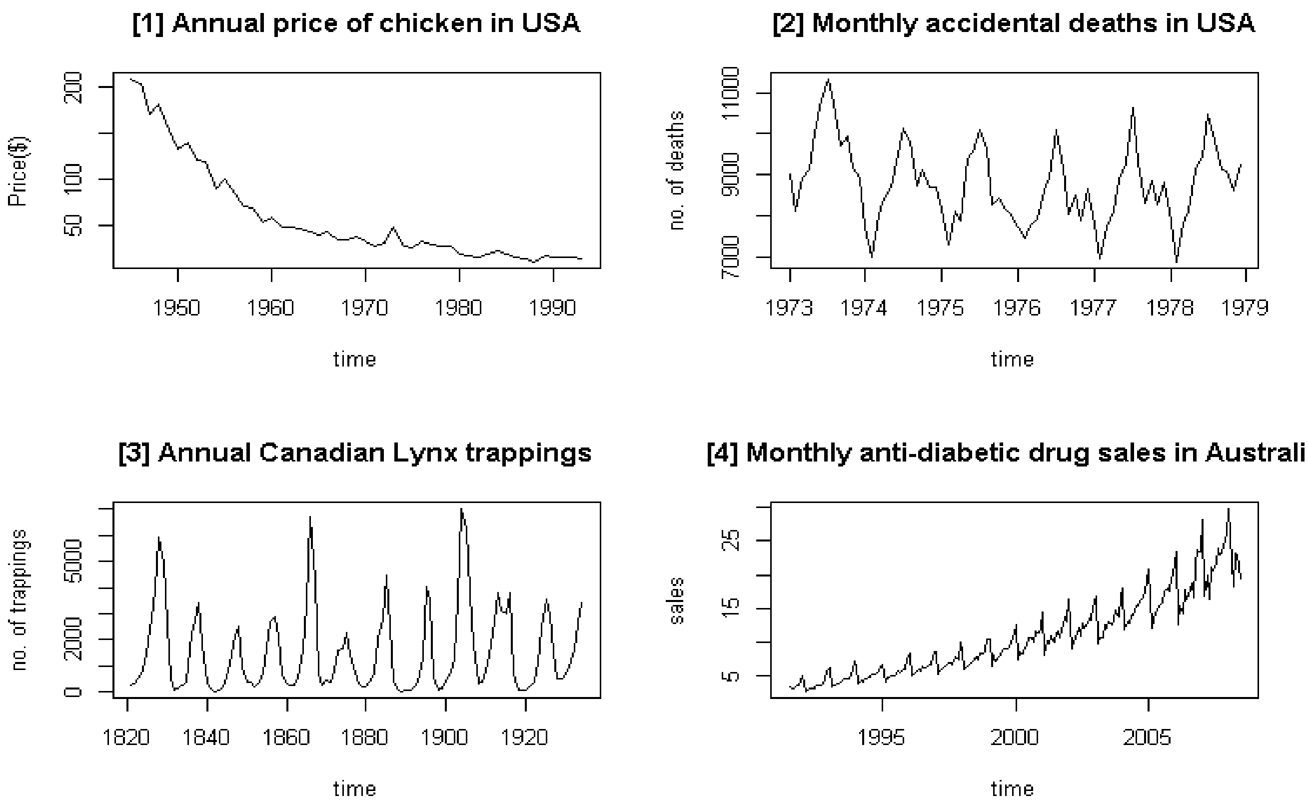


Figure 6. Time series plots chosen for further investigation.[7]

From the plots in Figure 6, a trained eye can immediately notice that the time series [1] has a persistent trend with no traces of seasonality. Whereas time series [3] looks like it is seasonal at first, but upon closer observation, one can notice that the magnitudes of the spikes are uneven, indicating cyclic behavior. Moreover, the occurrence of the cyclic patterns in random intervals is yet another aspect that sets apart cyclicity from seasonality. The assumptions of trend and cyclic behavior will be tested

---

[7] Source: Plots generated using pre-installed datasets available in the fpp2 package in R.

later using lag and ACF plots. Contrastingly, the series [2] and [4] seem to display symptoms of seasonality. We can study this seasonal presence in greater clarity using seasonal and subseries plots.

Seasonal and sub-seasonal plots

From the seasonal plot [2] depicted in Figure 7 shown below, we can clearly make out that the number of deaths were high for the year 1973 and then decrease during the subsequent years. This also signifies a very slight downward trend, which was hard to notice because of obstructions due to seasonal effects in the simple plot earlier in Figure 6, plot [2].

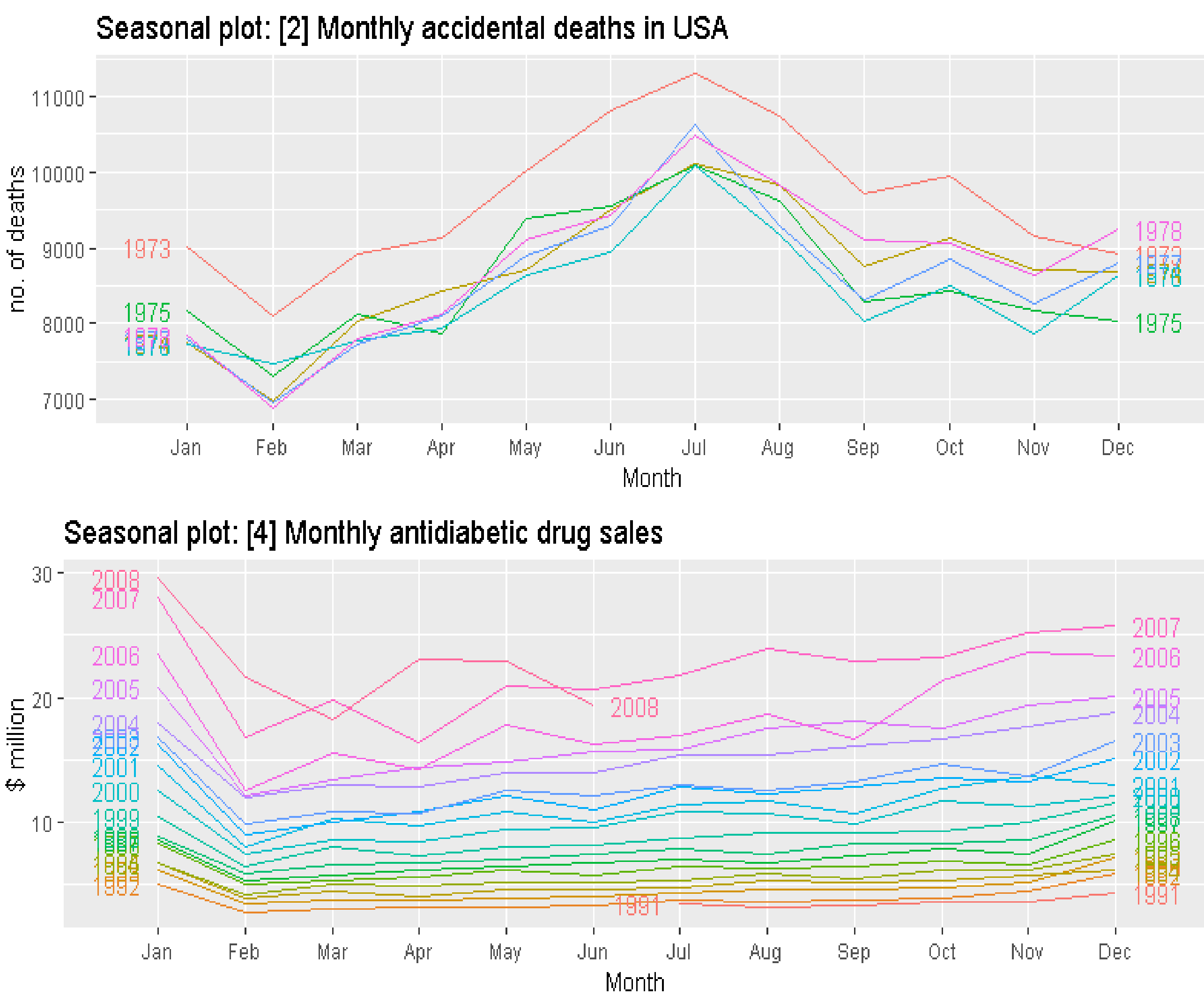


Figure 7. Seasonal plots for studying seasonality changes.

In the seasonal plot [4] however, not only is the rising trend evident but also shows the increasing magnitude of seasonal cycles can be noticeable. Particularly, the curves representing the years 1992

– 2000 are relatively short and flat when compared to the long and staggered curves that occur later during the years following 2001.

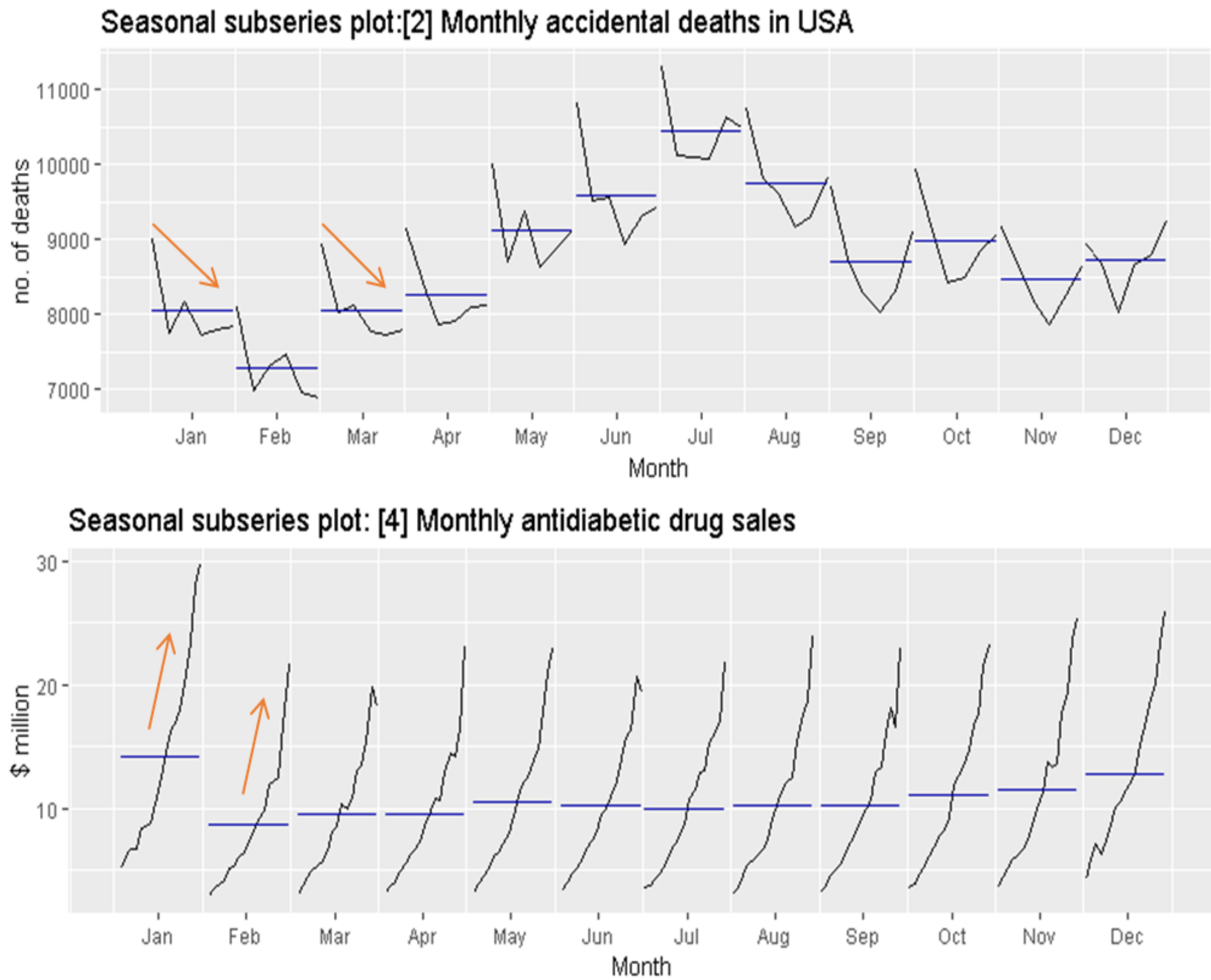


Figure 8. Seasonal subseries plots to study seasonality.

Figure 8. above, shows typical sub-seasonal plots. This is another way of representing the seasonality present in the series. In this kind of plot, the mean of all the seasons each year is indicated separately by the short horizontal blue line seen each month. In seasonal sub-series plot [2] we can see how the seasonality is behaving. It starts increasing in March, peaks in July and then starts decreasing until it finally stabilizes in November. The orange arrows show the development of the trend over time. Notice how the arrow slopes down gradually signifying a slight trend depicting the decrease in the accidental deaths. Whereas, the steep upward slope of the arrow in [4] indicates a strong upward trend representing a healthy growth in sales. It is also important to note for obvious reasons, that the

seasonal and subseries plots only work well for time series with seasonality. When applied on time series lacking seasonality, R throws an error informing the user that the data is devoid of seasonality.

<u>Lag plots</u>

Another visualization method for detecting the presence of the time series components can be realized by plotting the subsequent time lags of the variable against each other. They comprise a series of scatterplots arranged in increasing order of time lags. The variables that constitute the coordinates of the scatterplot are the lagged values. The correlation of lagged values in time is known as autocorrelation. This concept will be discussed in detail when we deal with ARIMA models later.

In the lag plots shown in Figures 9 - 12, each scatterplot shows y_t plotted against its subsequent lags y_{t-j} for j number of lags. We can easily detect the presence of trend, seasonality, cyclic and a combined effect by simply looking at these plots. For instance, Figure 9. represents a lag plot of the time series (Figure 6, plot [1]), characterized by a distinct downward trend. In this lag plot, notice that the most recent lags show a relatively high autocorrelation compared to the later lags. This effect is highlighted by the red rectangle over lags 1 and 2.

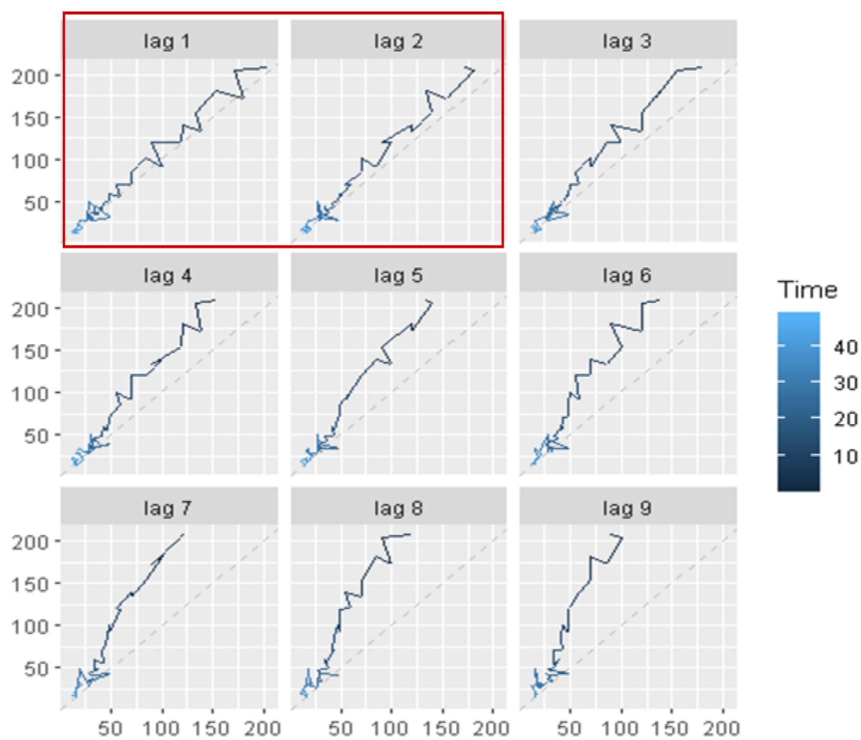


Figure 9. Annual prices of chicken in the USA (Lag plot [1]).

However, since monthly seasonality is predominant in time series (Figure 6, plot [2]), high autocorrelation occurs periodically for every 12th lag as shown by the orange rectangle in Figure 10. Analogously, the cyclic behavior present in time series (Figure 6, plot [3]) is depicted by completely random circular curves in the lag plots shown in Figure 11. This peculiar behavior seen in the lag plots is typical when dealing with time series containing cyclicity.

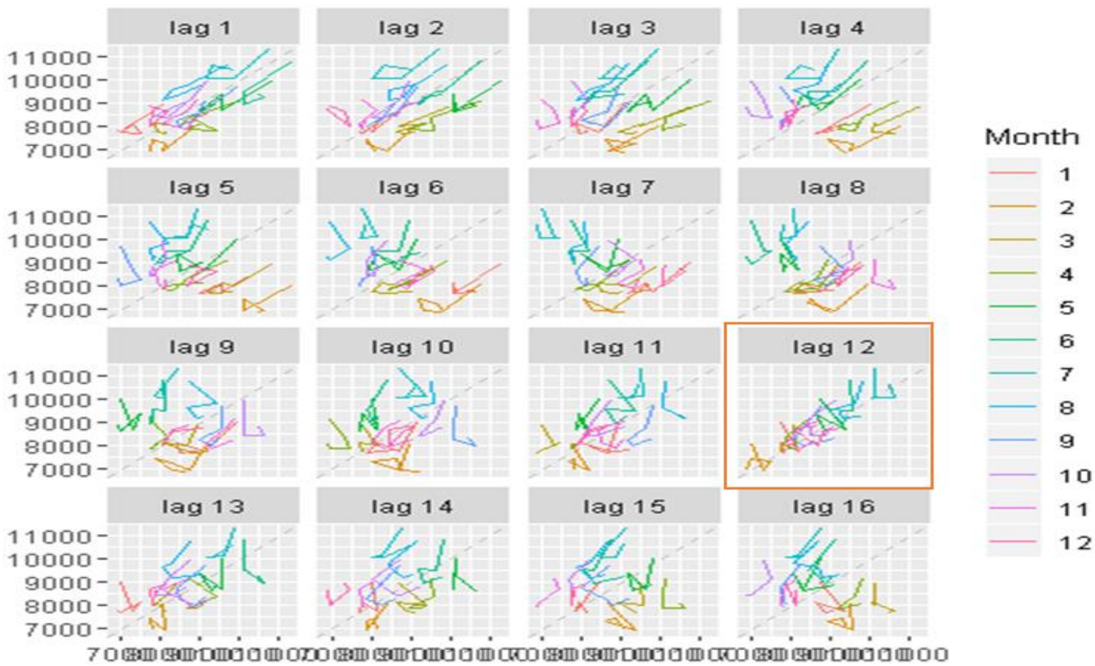


Figure 10. Monthly accidental deaths in USA (Lag plot: [2]).

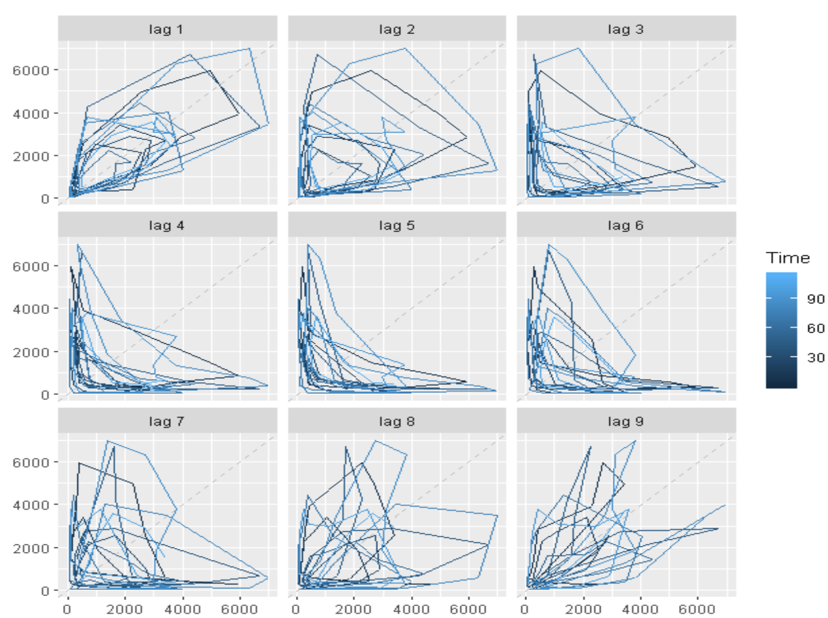


Figure 11. Annual Canadian Lynx trappings (Lag plot [3]).

Finally, the presence of both seasonality and trend in time series (Figure 6, plot [4]) is signaled using the same colored rectangles used earlier to represent trend and seasonality (red for trend and orange for seasonality) as seen in Figure 12.

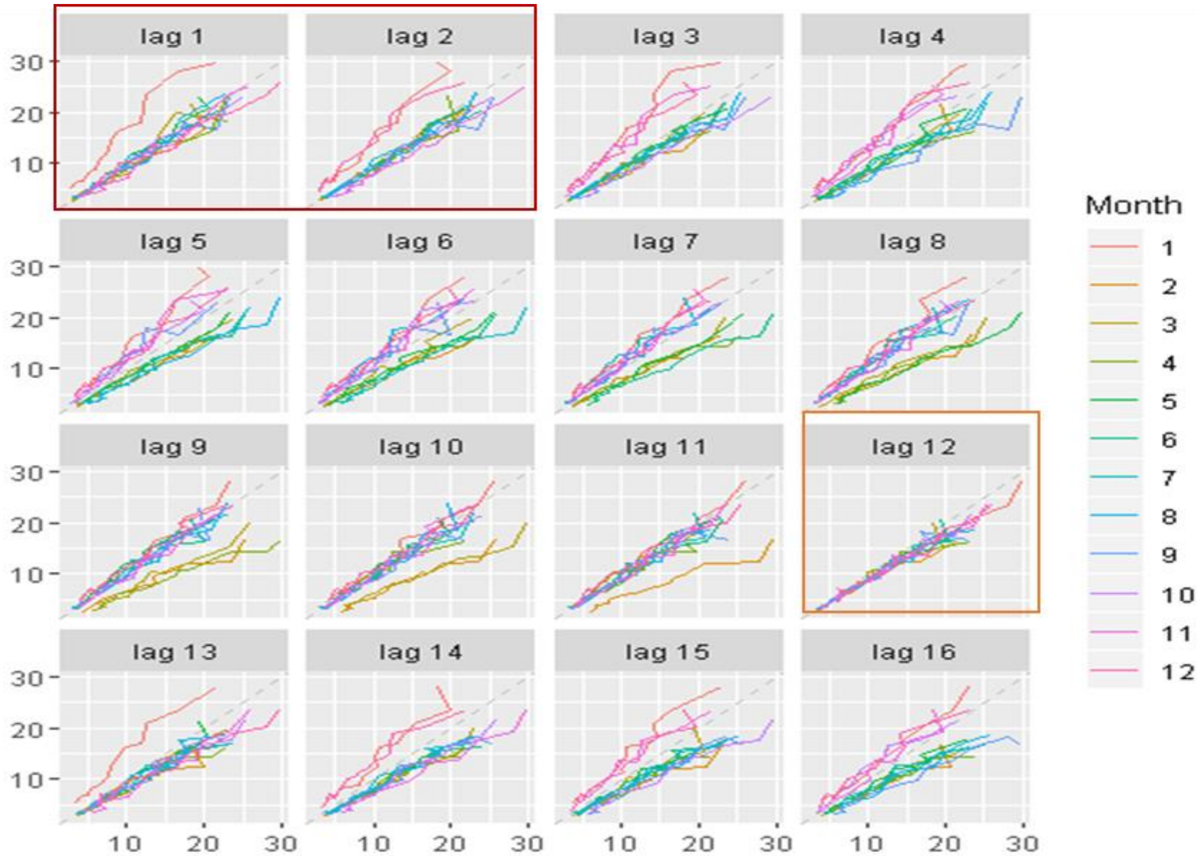


Figure 12. Monthly antidiabetic drug sales (Lag plot [4]).

<u>Auto-correlogram or ACF plots</u>

The ACF plots shown in Figure 13 are commonly known as auto-correlograms that are based on the property of autocorrelation. If correlation determines the relationship between two variables, then autocorrelation is simply the relationship between its the lagged observations. We have already seen this concept used previously in lag plots. In addition to the revelation of time series components, these plots also reveal the underlying autocorrelation structure (time-dependence) in the series. Due to this property, they are widely used in selecting appropriate ARIMA models. Note that the underlying autocorrelation is usually hard to detect during the presence of dominating effects induced by non-stationary elements (trend, seasonality, etc.). These elements overshadow the pure autocorrelation structure by inducing large positive systematic autocorrelations as seen in the plots that follow. We will discuss ways of removing these non-stationary elements later when we deal with ARIMA modelling in Chapter 4. For now, we will keep our focus restricted to the analysis of time series components as shown in the ACF plots that follow.

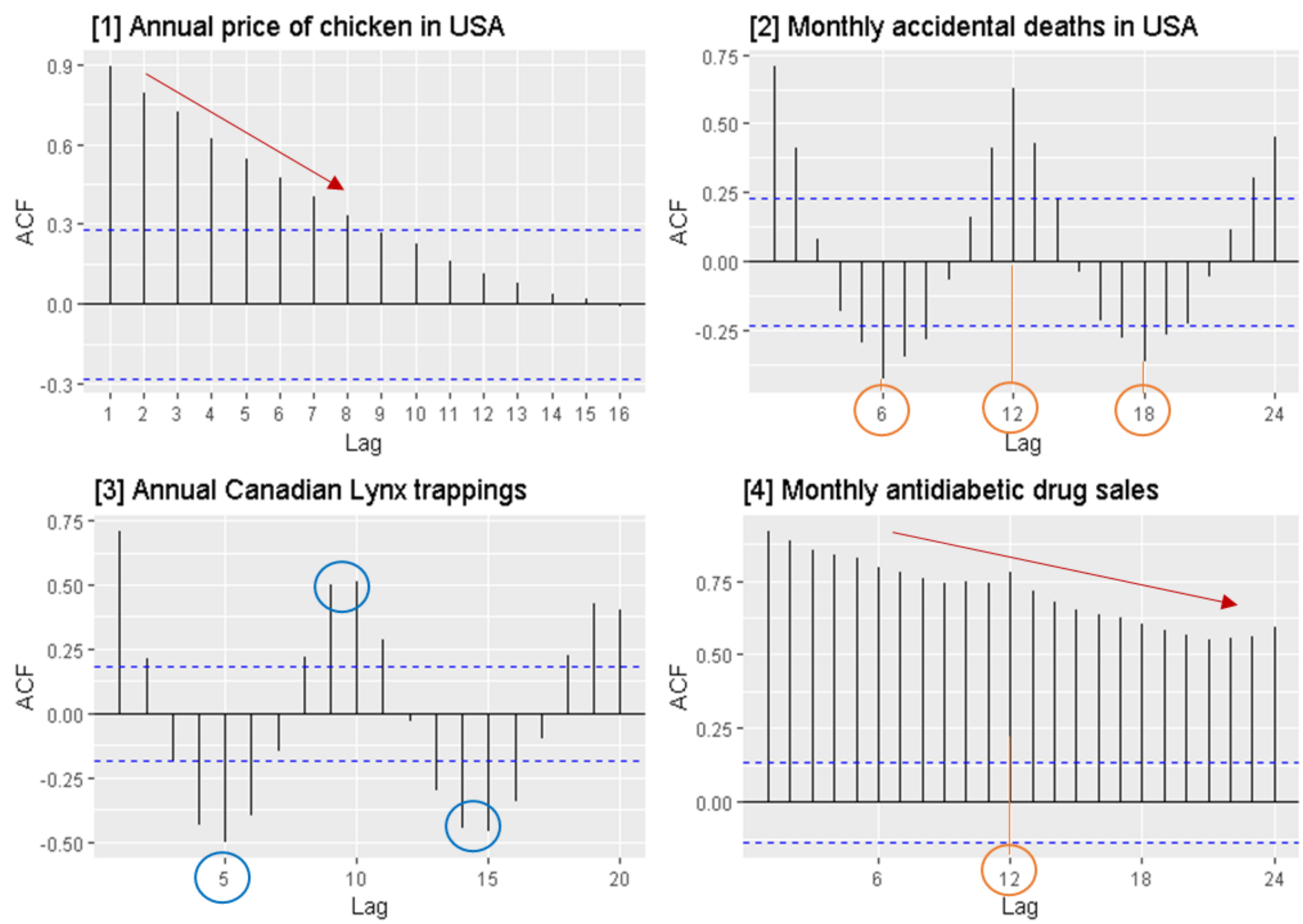


Figure 13. ACF plots for time series analysis.

In the ACF plot labeled [1] belonging to a series with a downward trend, it is common for the autocorrelation function (ACF) to gradually decrease as shown by the red arrow. If we recall, we witnessed the same type of behavior while using lag plots previously, where the autocorrelations gradually decreased. This also applies to the ACF plot [2], where the periodic peaks occurring at lags 6, 12 and 18 indicate seasonal behavior. This phenomenon is augmented by the orange circles. For cyclical behavior, however, these peaks and troughs occur at irregular intervals as indicated by the blue circles in ACF plot [3]. Lastly, the presence of both trend and seasonality is indicated by the rippled shape, which is a combination of the respective indicators for trend and seasonality. This can be clearly seen in the last ACF plot labeled [4].

3.2 Time series decomposition

As we have seen in the plots before, a time series can sometimes occur as a combination of these fundamental components– trend, cycle, seasonality, and randomness. In such cases, it would be much easier if these components can be separated and studied individually to better understand the pattern behavior. This can be performed by using time series decomposition. It is a technique which still retains its popularity, even after decades have passed since it was first developed. Methods of decomposition are considered as one of the oldest techniques used for time series analysis. It was initially developed to eliminate the influence of the trend while studying the effects of serial correlation between or within variable(s). Later, attempts were also made to separate the seasonal fluctuations from the data using these methods. During this time, economists were also frantically searching for ways to isolate the economic activity from the influence of the business cycle. Thus, decomposition methods quickly gained attention and were implemented for economic purposes. (Makridakis, Wheelwright and Hyndman, 1998, pp.82-83).

The oldest technique among the lot is known as classical decomposition. Later, many other forms of decomposition were developed by gradually eliminating the disadvantages of the classical form. Following the same chronological order, we will initially start our discussion with the classical decomposition since this forms the basis for all other techniques. Other decomposition methods will be subsequently introduced as this section proceeds. We start by defining the basic underlying concept that makes all kinds of decomposition methods work. When data consists of an underlying pattern, its underlying components can be represented by the following equation.

$$Data\ =\ pattern\ +\ error\ =\ f\ (trend - cycle, seasonality, error) \tag{3}$$

By substituting each component with a particular symbol, we get the following.

$$Yt\ \ =\ f\ (\ St\ \ , Tt\ \ , Et\ ) \tag{4}$$

Where $\boldsymbol{Y_t}$ is the actual value at time *t*;

$\boldsymbol{S_t}$ is the seasonal component (or index) at time *t*;

$\mathbf{T_t}$ is the trend-cycle component at time *t*;

$\mathbf{E_t}$ is the irregular (error) component at time *t*.

Decomposition methods are normally split into two forms – the additive and multiplicative forms. The additive approach is more common though. Which means that the seasonal, trend-cycle and the irregular components are simply added together to produce the observed series. The additive and multiplicative approaches can be mathematically represented by the following equations.

$$Yt = St + Tt + Et \tag{5}$$

$$Yt = St * Tt * Et \tag{6}$$

The only difference in the multiplicative form is that the components are multiplied together instead of addition. Although these equations don't differ much apart from their signs, they do differ in their usage. An additive form of decomposition is suitable for data whose seasonal fluctuations do not vary with the level of the series. If the seasonality varies along with time and deviates from the level, then a multiplicative model is well suited to capture this form of variation. If the choice between an additive and multiplicative model is too difficult, then we can initially transform the series and then use additive decomposition. The decomposed series can later be transformed back to the original condition. We will discuss transformations in the next section. Note that the various decomposition methods described in the paper are as mentioned in Hyndman (2018).

Now let us look at the general steps of classical decomposition. An attempt to showcase the similarity and the subtle differences between the additive and multiplicative models is made as seen in Table 2. While using `R` for decomposition, these calculations are performed automatically. Then the results i.e. separated time series components, are displayed in the form of a visual representation such as the one shown in Figure 14 and Figure 15. As an example, the `a10` dataset[8] that is available by default in `R`'s `fpp2` package is used. An important characteristic of this time series is that it contains seasonality that varies with the level of the series. Its corresponding time plot can be found in (Figure 6, plot[4]). By applying both additive and multiplicative seasonality we get the following results. Notice that the steps are grouped using different color codes. Each color represents an operation on a particular time series component. For example, step 1 and step 2 are color-coded with blue indicating operations that handle the trend. In a similar fashion, red and green correspond to seasonality and the remainder component respectively.

[8] Monthly anti-diabetic drug sales in Australia from 1991 to 2008.

Classical decomposition

Additive	Multiplicative
Step 1: Calculate the trend-cycle component $\hat{T}$ by using a $2 \times m$-MA. If m is an odd number then just use m-MA.	Step1: Calculate the trend-cycle component $\hat{T}$ by using a $2 \times m$-MA. If m is an odd number then just use m-MA.
Step 2: Calculate the detrended series: $y_t - \hat{T}_t$.	Step 2: Calculate the detrended series: $\frac{y_t}{\hat{T}_t}$.
Step 3: Then calculate the average of all detrended values for that season. e.g. For monthly data, find the average of all detrended values belonging to July in that data.	Step 3: Then calculate the average of all detrended values for that season. e.g. For monthly data, find the average of all detrended values belonging to July in that data.
Step 4: The obtained average values are then adjusted to ensure that they add to zero.	Step 4: The obtained average values are then adjusted to ensure that they add to m.
Step 5: The seasonal component ($\hat{S}_t$) is then obtained by putting together the adjusted values for each year.	Step 5: The seasonal component ($\hat{S}_t$) is then obtained by putting together the adjusted values for each year.
Step 6: The remainder component is calculated by subtracting the obtained seasonal and trend-cycle components. $\hat{R}_t = y_t - \hat{T}_t - \hat{S}_t$	Step 6: The remainder component is calculated by dividing the obtained seasonal and trend-cycle components. $\hat{R}_t = y_t \,/\, (\hat{T}_t * \hat{S}_t)$

Table 2. Steps involved in classical decomposition using the additive and multiplicative approach.

Performing decomposition using R returns a visual plot consisting of the various components as shown in Figure 14 and Figure 15. When comparing the remainder component of the additive with that of the multiplicative form, the inability of the additive model to capture the increasing seasonal variation adequately can be clearly noticed. The red rectangle shows how the uncaptured variation adds to the residuals making it larger, whereas the green rectangle shows only residuals, which in turn indicates that the seasonality was adequately separated. The respective R codes for performing this procedure can be found in Appendix A1.

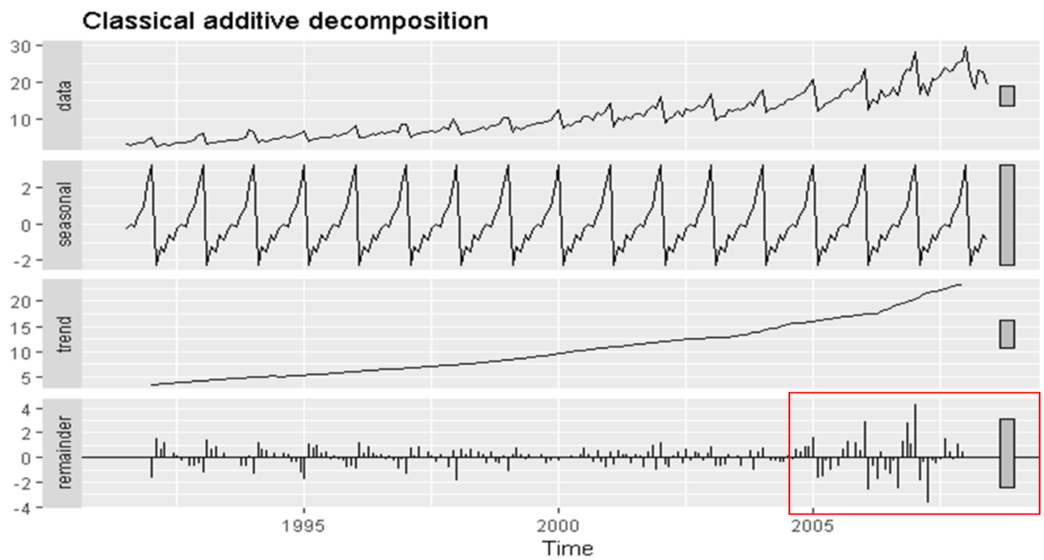


Figure 14. Results of a classical decomposition (additive).

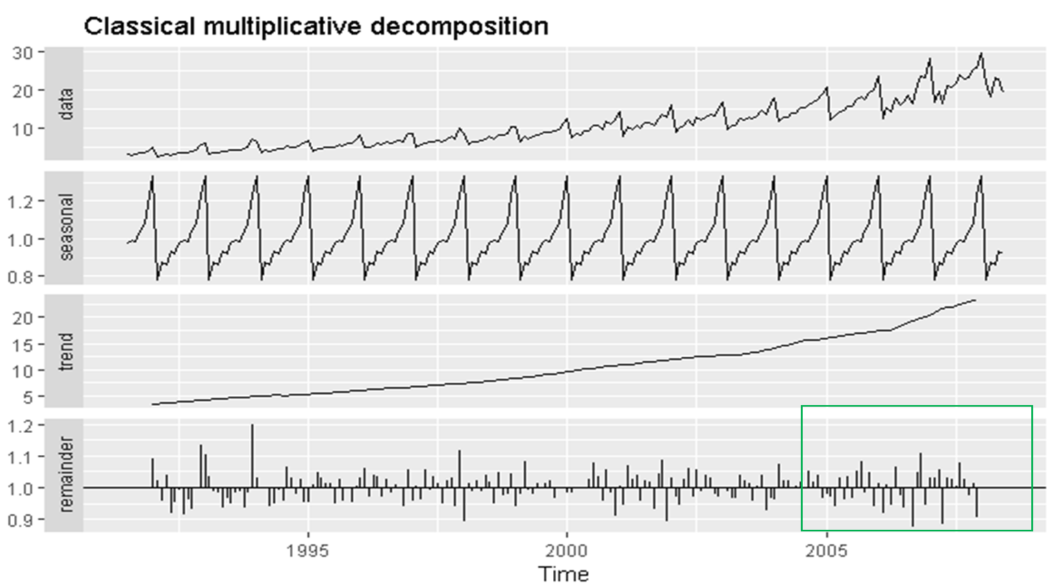


Figure 15. Results of a classical decomposition (multiplicative).

But the use of classical decomposition has gradually declined with time due to some of its inherent shortcomings, which in turn have led to the development of new methods to address these problems. The underlying problems with the classical form are listed as follows.

- Missing values on both ends of the trend-cycle estimate, particularly the recent values, which are very important for forecasting purposes.
- The smoothing of the trend component cannot be controlled, which increases the risk of over-smoothing. Important data characteristics can be lost.
- Classical methods assume a seasonal pattern to stay the same each year. For short-term series this is true but for some longer series this assumption does not hold; it is highly likely that the seasonality will change over longer horizons.
- Not immune to outliers, sudden shifts, and special values, whose presence may distort the original decomposed patterns.

To overcome these problems several new decomposition methods have been developed. The remaining part of this section is dedicated to the discussion of these modified versions of decomposition. However, a detailed discussion is avoided as it requires stepping out of the scope of this research. References are provided accordingly for those who are interested to investigate further.

X-11 and SEATS decomposition

This method was first used by the US Census Bureau and Statistics Canada. It is based on the classical decomposition. This method allows both the additive and multiplicative type of classical decomposition and was designed solely to overcome its disadvantages - It successfully captures the trend-cycle component without missing any information and does not suffer from losses due to over-smoothing. Moreover, this method uses a sophisticated method that allows the seasonal pattern to change slowly over time. It is robust to sudden shifts, calendar adjustments, and outliers. The only downside to this method is that it can only decompose quarterly and monthly data. This method is also the oldest and commonly used method to extract seasonality from the time series before forecasting (Gooijer and Hyndman, 2006, p.10).

Another decomposition method that resembles the X- 11 method is called ‘SEATS’ decomposition, which stands for Seasonal Extraction in ARIMA Time Series. This method was developed by the Bank of Spain and is widely used in many fields of business. This method also limits its usage to only monthly and quarterly seasonality. Please refer Dagum and Bianconcini (2016)[9] for more details regarding these methods.

STL

The STL decomposition is the most versatile and robust method of the lot. It stands for “Seasonal and Trend decomposition using Loess”, where Loess is a form of regression used for estimating non-linear relationships. Developed by Cleveland *et al*. (1990)[9]; this method contains many advantages over the X-11 and the SEATS methods as stated below.

- It can handle any type of seasonality and not restricted to monthly and quarterly.
- Not only is seasonality allowed to change over time, but also its rate of change can be controlled by the user.
- The smoothness of trend can also be controlled by the user in a similar fashion.
- The user can specify a robust decomposition if necessary so that outliers and unusual observations do not affect the trend and seasonality.

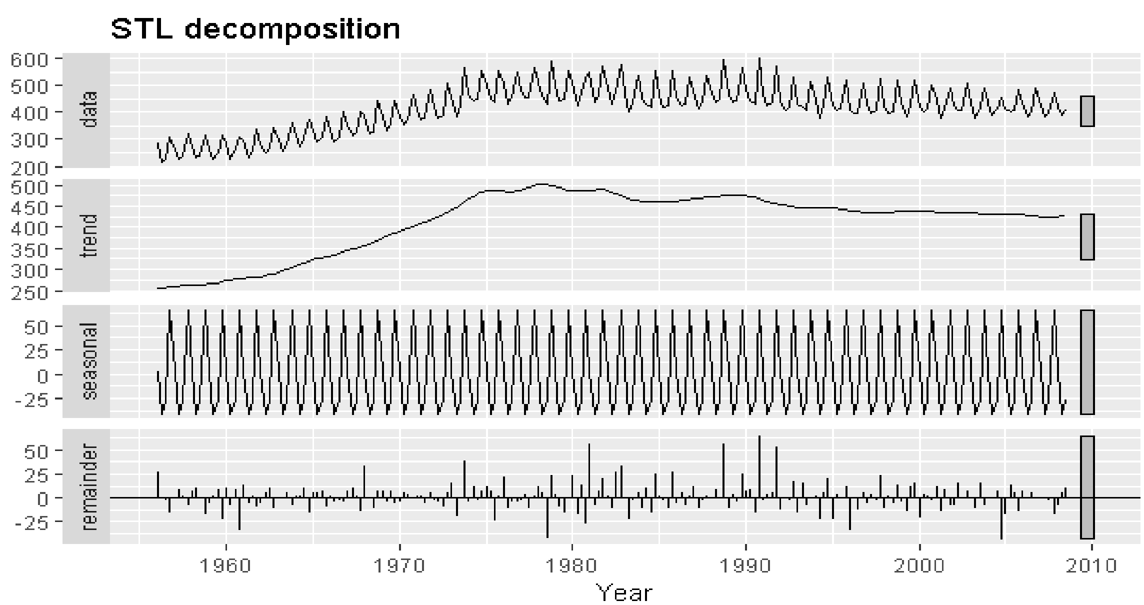


Figure 16. Manual STL decomposition of Quarterly Australian beer production.[10]

---

[9] As cited in Hyndman (2018).

[10] Source: Preloaded dataset ausbeer – Total quarterly beer production in Australia (In megaliters) available in fpp2 package in R.

Although STL performs relatively better than other methods, it does have a few minor disadvantages. i.e. it does not handle trading day or calendar day variations and allows only the additive type of decomposition. However, it is possible to perform a multiplicative decomposition by initially taking the log of the data and then performing the STL decomposition. Finally, back transforming the components brings them back to their original form. The next section deals with such transformation procedures and discusses the wide range of applications they entail. Figure 16 and Figure 17 depict the results of the STL decomposition performed using R. When certain parameters need adjustment, the manual mode of STL offers more customizability. On the other hand, if speed is the priority then the automated STL decomposition mode is preferred.

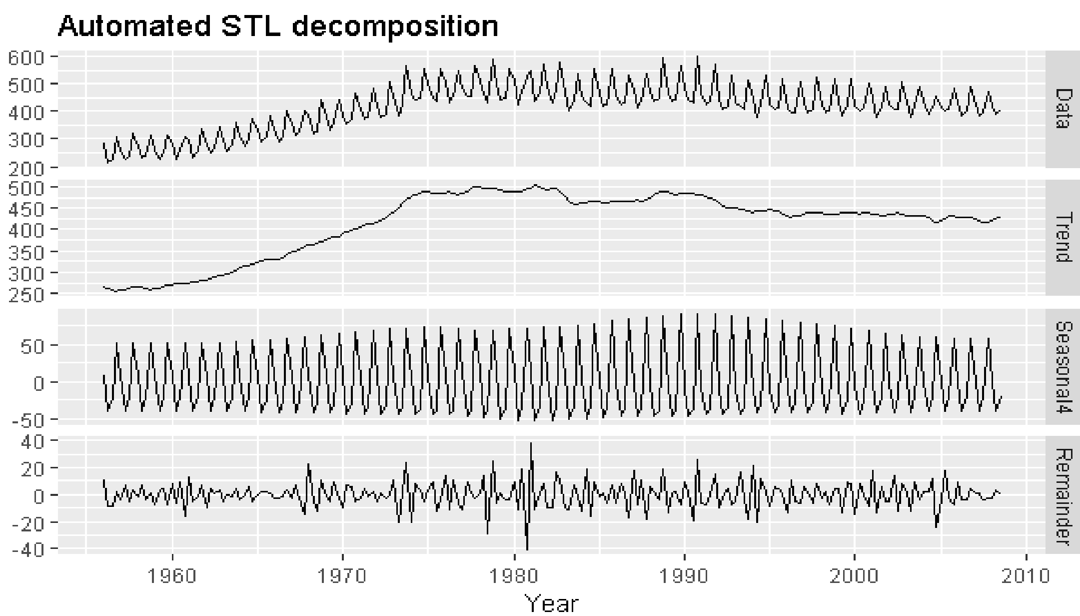


Figure 17. Automated STL decomposition of Quarterly Australian beer production.[11]

3.3 Transformations and adjustments

Previously, an alternative approach for performing STL decomposition on time series exhibiting the multiplicative form of variation was mentioned. Essentially, it made use of log transformations, which is a simple method of data adjustments. In this section, we will look at the different types of transformations and demonstrate how we can carry out such procedures using R. Such adjustments simplify the time series patterns and makes further data analysis easier. More importantly, using

[11] Data source: Preloaded dataset ausbeer taken from the fpp2 package in R.

simplified data leads to more accurate forecasts allowing the use of simple models, thereby reducing the need for more complex forecasting models.

In this paper, we will specifically deal with a class of adjustments known as mathematical transformations. These transformations can be made in many ways, but the most commonly implemented way involves the use of power and logarithmic transformations. A special family of transformations that use both the power and logarithmic modes is known as the Box-Cox transformations. The selection of the transformation type is controlled by the value assigned to the parameter λ, applied to the time series y undergoing a transformation. The entire relationship can be represented mathematically as follows (Hyndman, 2018).

$$w_t = \begin{cases} \log y_t , & if\ \lambda = 0; \\ (y^t - 1)/\lambda, & otherwise. \end{cases} \tag{7}$$

Where $\boldsymbol{\lambda}$ is the Box-Cox transformation parameter;

$\boldsymbol{y}$ is the time series.

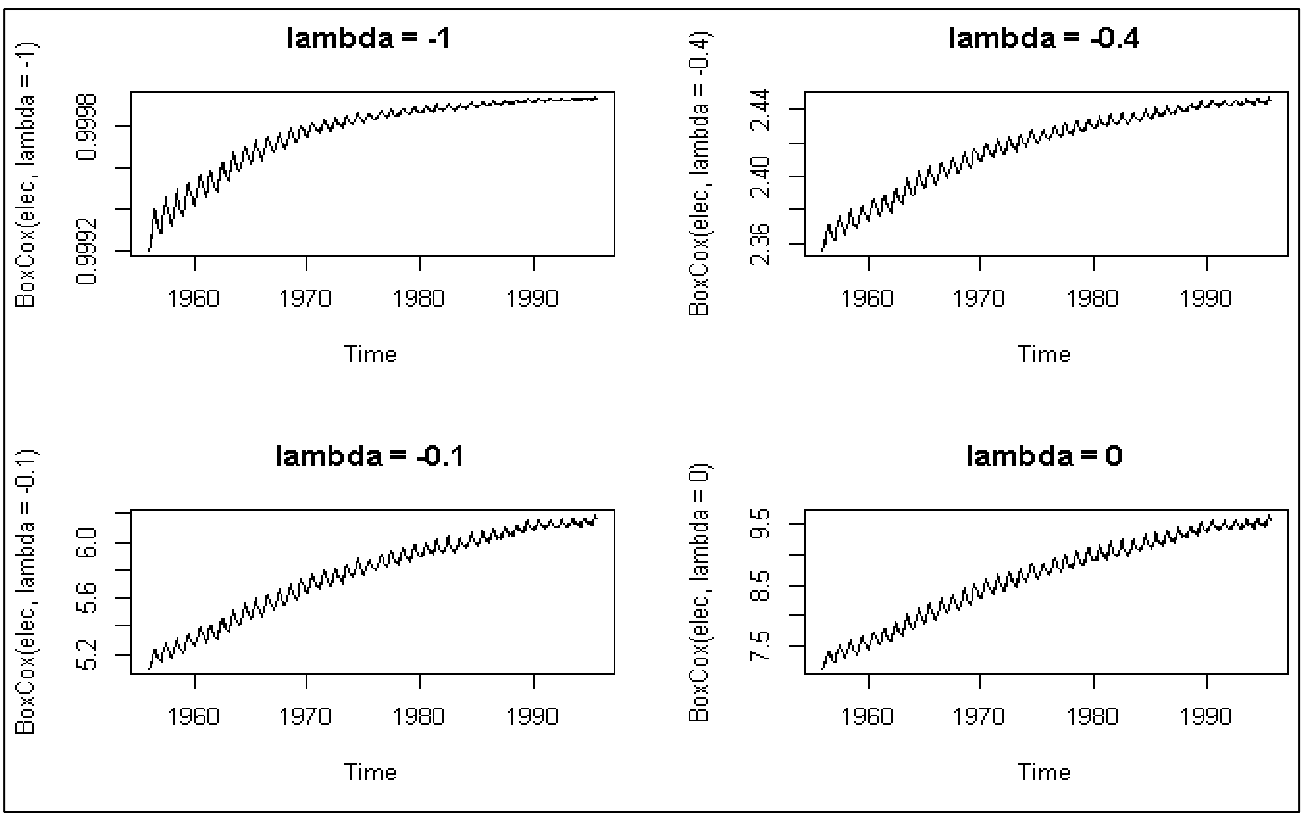


Figure 18. Box-Cox transformations for $-1 \leq \lambda \leq 0$.

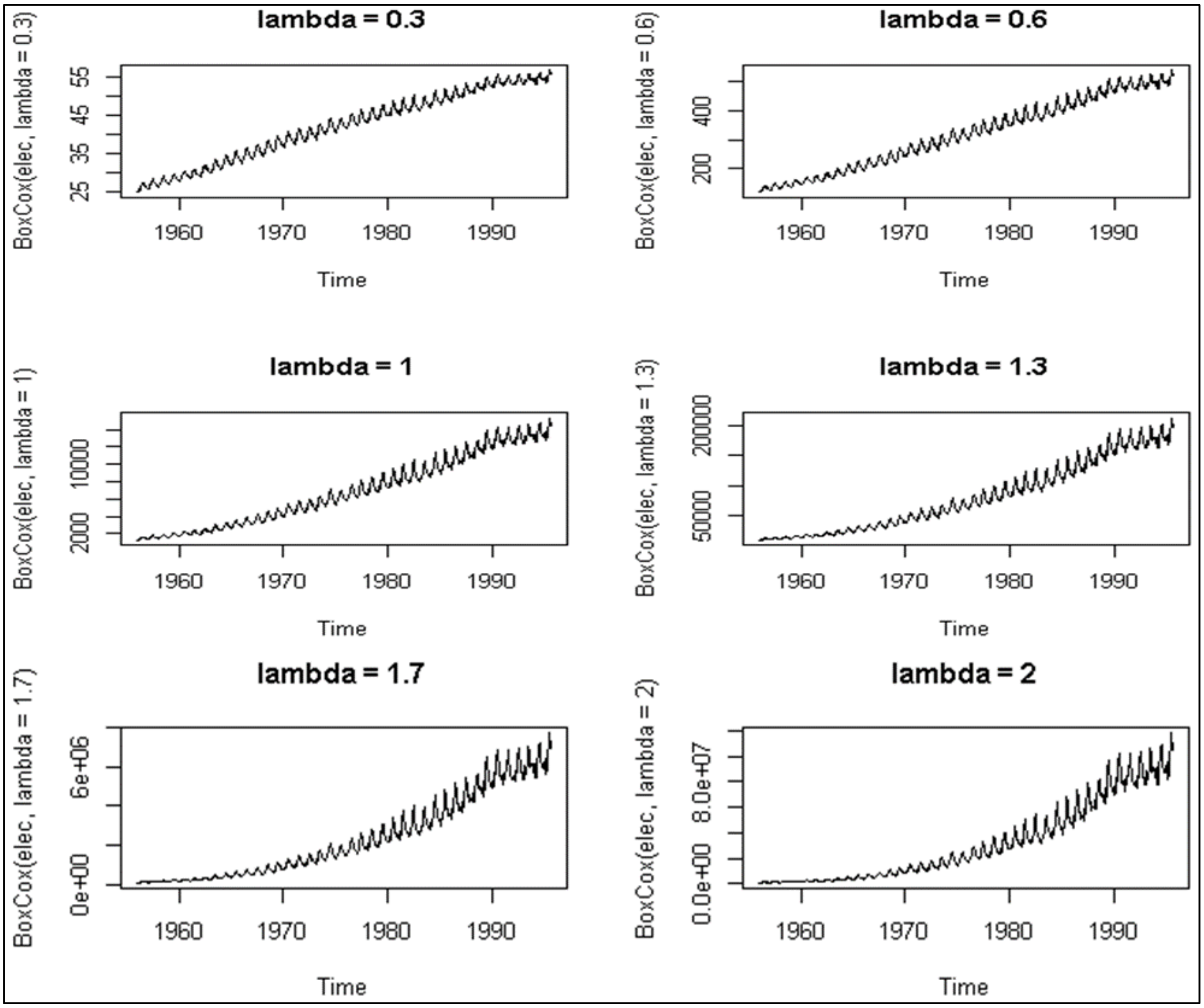


Figure 19. Box-Cox transformations for $0 \leq \lambda \leq 2$.

A natural logarithm (to the base *e*) is always used in a Box-Cox transformation for cases when $\lambda = 0$. For cases when $\lambda \neq 0$, a power transformation is applied. The influence of the parameter λ can be better understood by visualizing it using R. The parameters can be either specified manually or chosen automatically by R's built-in BoxCox() function as shown by the R codes provided in Appendix A1. For this example, the elec dataset taken from R's fma package is used. The data represents the Monthly electricity production in Australia. On applying this function to this dataset, the optimal value of λ is computed to be 0.2654076. This can be confirmed by looking at each plot provided in Figure 18 and Figure 19, where the behavior of the time series as values of λ change from -1 to 2 is shown.

Chapter 4. Commonly used forecasting techniques

After all these years of research that this field of research has been through, it is obvious to witness a large accumulation of forecasting methods. Moreover, with the advent of powerful supercomputers, cheap and speedy computing empowered a proliferation of many novel forecasting methods too. Especially, with data-analysis related problems gaining popularity in recent times, there has been an increase in the influx of novel methods developed by computer scientists, statisticians, and mathematicians all over the world, making forecasting an interdisciplinary field.

Although there is a lot of literature that exists in academia that endorses new methods and their complexity, very little research exists that validate their out-of-sample accuracy by comparing them to naïve benchmarks. The out of sample accuracy studies discussed in Table 15 confirm this argument. Nevertheless, there are a few well-established methods which have stood the test of time and have been vigorously researched in academia repeatedly. Novel methods have also been developed based on these well-established procedures. This section particularly aims to discuss the key features describing state-of-the-art methods and tries to determine their relationships to the well-established methods, which were prolifically utilized in the past. As an added benefit, using this holistic approach for discussing about methods aims at simplifying the general understanding about these techniques, in turn bridging the gap between theory and practice.

But before we proceed with this chapter, an important distinction must be made at this point between the usage of the terms - *method* and *model*. Until now, we used the term 'methods' to signify all types of forecasting techniques in general. But in this chapter, the term 'methods' will be used solely for referring to *heuristic methods*. More complex techniques that use an underlying *statistical model* will be referred to as 'models' from this point onwards in this chapter.

According to Hyndman *et al.*(2008, pp. 4-5), a forecasting method is basically an algorithm for generating a point forecast. A point forecast is simply a single forecasted value. In contrast, a statistical model is characterized by a stochastic data generating process consisting of a range of future values scattered across a probability distribution. An important motivation for choosing models for forecasting is that they allow setting up of prediction intervals within a certain level of confidence. The specification of prediction intervals is a crucial aspect that affects forecast accuracy. The main reason for using them is to quantify the uncertainty involved in the forecasts. It is considered as good forecasting practice to present a range of forecasts supported by prediction intervals rather than blindly relying on point forecasts. Prediction intervals will be dealt with separately in Chapter 5.

To further clarify the distinction between a method and a model, we can refer to the explanation given by Ord and Fildes (2013, pp. 59-60). According to the authors, every forecasting process is extrapolative in nature. This extrapolation is performed for a definite time horizon. Usually, extrapolation requires a forecasting function, which is simply a mathematical equation that forms the fundamental entity governing a quantitative forecasting technique. Thus, the basic difference between a method and a model can be noticed by observing how this forecast function is used. If a forecasting technique uses the forecast function directly and is not based on an underlying statistical framework, then it is referred to as *heuristic methods*. This type is commonly used as ad hoc procedures in business forecasting. But when the forecast function is governed by an underlying statistical model, then we refer to it as a forecasting *model*. As mentioned earlier, statistical models are simply a mathematical description of a data generating process from which a forecasting method can be formulated. For the same reason, it is not surprising to find that many heuristic methods described in this paper are linked to certain statistical models as well.

4.1 Heuristic methods

The most rudimentary approach to generate a forecast would simply comprise of the average(mean) of all the past observations. It can be mathematically represented as shown in the equation below.

$$\hat{y}_{T+h\,|\,T} = \bar{y} = (y_1 + \dots + y_T)\,/\,T \qquad (8)$$

Where $\boldsymbol{y_1 \dots y_T}$ represent the past observations;

$\boldsymbol{\hat{y}}$ represents the forecast;

$\boldsymbol{\hat{y}_{T+h|T}}$ is the estimate of y_{T+h} based on the data $y_1 \dots y_T$; in other words, it is the forecast made using actual observation y_T at time t, for $T + h$ time steps ahead;

$\boldsymbol{y_1}$ is the first observation and y_T is the most recent or last observation in the series;

$\boldsymbol{T}$ represents the number of observations;

$\boldsymbol{h}$ represents the forecast horizon.

The simple mean method as described above makes for an ideal model for initiating the discussion on heuristic methods. Note that all equations used in this section are taken from Hyndman (2018). Setting this as the starting point (baseline for this study), we first take a look at some of the commonly used heuristic forecasting methods.

4.1.1 Naïve method

We begin with the simplest heuristic method of the lot – the *naïve* method. In this technique, the forecasting is purely done by considering the current period's actual observation as the forecast for the next period. This can be depicted mathematically using the following equation.

$$\hat{y}_{T+h\,|\,T} = y_T \tag{9}$$

Despite being very basic and uncomplicated, this method is still widely used in forecasting today. In fact, this is the first method of choice for forecasting stock price data due to the dynamic nature of the stock market. But it is more commonly used as a benchmarking tool against which other forecasting methods can be checked for accuracy. The reason for its selection as a benchmark can be traced back to its simplicity. One can argue that when a forecasting method cannot perform better than even the most basic and simplest method then it is not worth implementing it. And often, it has been found that this method is very hard to beat (Makridakis *et al.*, 1993).

Another variation to this method is also very popular and can be used for forecasting highly seasonal data. In this method, instead of using the most recent observation in the series, the forecast takes the value corresponding to the previous season of the year. (e.g. same month of the previous year). The mathematical representation is given below.

$$\hat{y}_{T+h\,|\,T} = \hat{y}_{T+h-m(k+1)} \tag{10}$$

Where $\boldsymbol{m}$ represents the seasonal period;
$\boldsymbol{k}$ represents the number of complete years in the forecast period prior to time $T + h$.

More simply said, in case of monthly data - all future forecasts for a particular month (say March) is equal to the last observed value for that month (i.e. previous year's March value). The same applies for quarterly data as well. The forecast for all future Quarter 1 values equal to the previous years' observed Quarter 1 values. This rule can be extended to other seasonal periods as well.

4.1.2 Drift method

When the forecasts are allowed to change (i.e. increase or decrease) over time then this is a characteristic of the drift method. To put it in simple words, the rate of change is known as the drift and is equivalent to the average change observed in the data. This relationship can be depicted mathematically by the following equation.

$$\hat{y}_{T+h\,|\,T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_T + h\left(\frac{y_T - y_1}{T-1}\right) \tag{11}$$

To better understand the working of the methods discussed until now, it would be a good idea at this instance to plot their forecasts against each other and visualize their behavior. We use the past three years' worth of data containing monthly sales for a certain shampoo as an example.

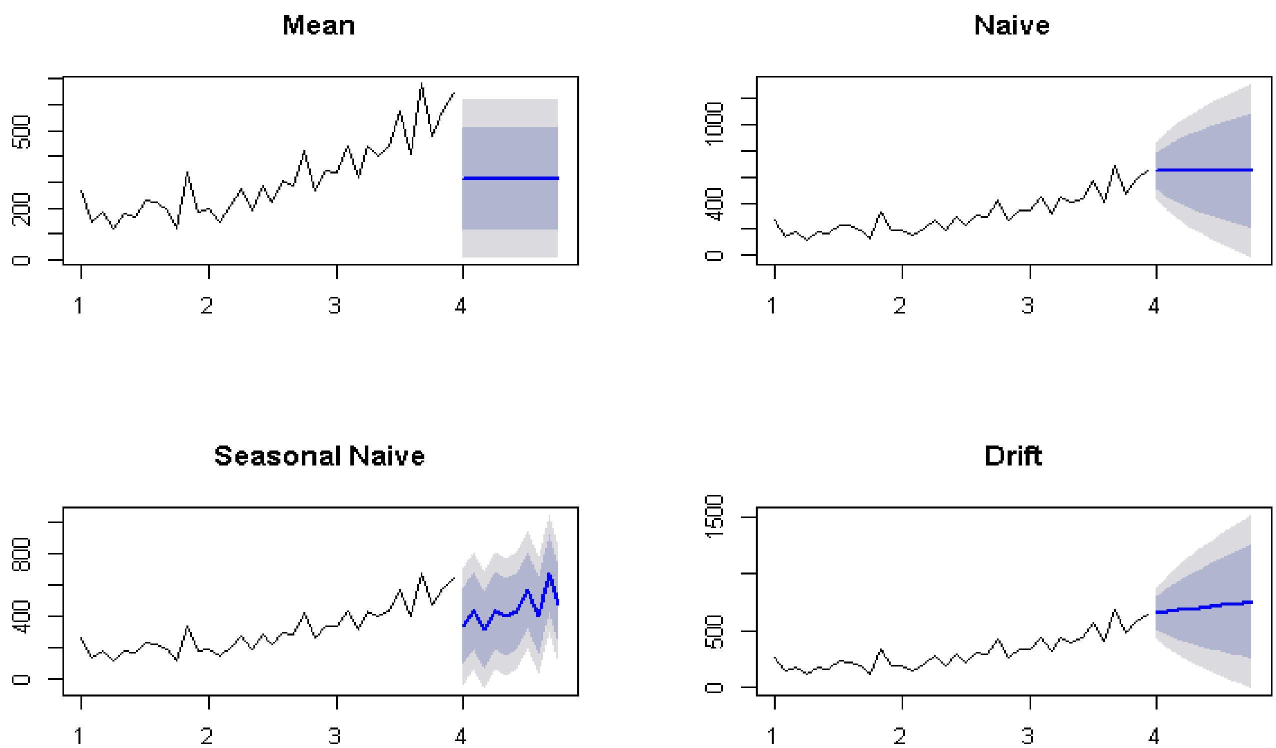


Figure 20. Forecasts generated using the mean, naive, seasonal naive and drift methods.[12]

The plots in Figure 20 show the forecasts with their prediction intervals, whereas in the superimposed plot in Figure 21, the prediction interval is switched off (`PI = FALSE`) to allow for better clarity. The

[12] Data source: Preloaded dataset `shampoo` taken from the `fma` package in `R`.

respective R codes used to generate these plots can be found in Appendix A1. Notice how the forecasts generated by the naïve and drift methods behave in comparison to the seasonal naïve forecasts. The naïve method simply copies the most recent observation generating a straight line. Whereas the drift method can be imagined as extending a line through the first and last observations of the time series. Contrastingly, the seasonal naïve combines the seasonal component to the naïve forecast. Most of these methods are used for benchmarking purposes and in certain cases, they can also be the best method for that forecasting scenario beating all other methods.

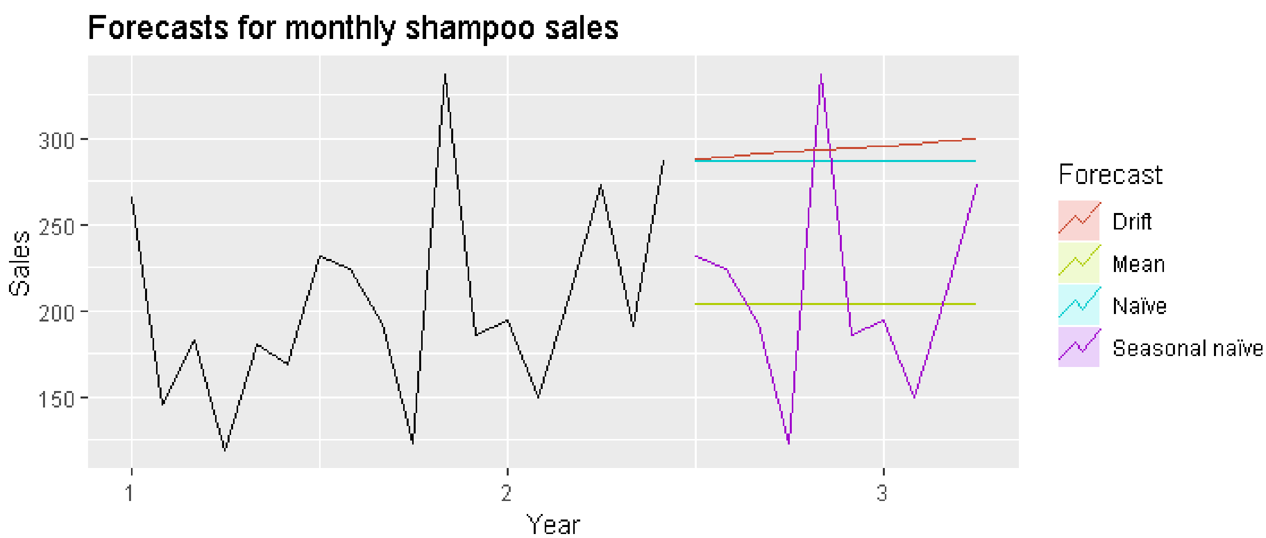


Figure 21. Superimposed plot.

4.1.3 Exponentially weighted forecasting method

As we saw previously, the mean and the naïve methods make up the simplest form of forecasting methods. In the naïve method, only the most recent observation makes up the forecast for all future periods. In contrast, the mean method uses the average of all observations as the forecast for all future periods. A more useful method would be something between these two extremes. A method whose forecast is based on all observations, but only the most recent observations are heavily weighted. This is the idea behind the exponentially weighted forecasts, commonly known as exponential smoothing methods (Hyndman, 2018).

Exponential smoothing methods first originated in the mid-1950s and were used by the military but gradually made their way into business applications due to the simplicity they offered in comprehension and computation. However, the simplicity also brought along with it a lot of skepticism by many academics and forecasters of that time. But the introduction to computing and the works of Robert G. Brown allowed this method to stand the test of time making it one of the most popular forecasting methods till date with substantial literature validating its utility (Hyndman *et al.*, 2008; Makridakis and Wheelwright, 1989, pp.5-6). With time, numerous variations and extensions of these techniques have been developed, the most important ones being the works of Brown (1956)[13], Holt (1952)[13] and Winters (1960)[13]. A relatively recent advancement includes the development of a modelling framework for exponential smoothing methods by the works of Ord *et al.*(1997)[14] and Hyndmann *et al.*(2002)[15], which extended the application of this concept from merely an ad-hoc method to a robust modelling approach incorporating stochastic models, likelihood calculations, prediction intervals and procedures for model selection. This state-of-the-art method will be discussed later in subsequent sections dedicated to state-space models.

Simple Exponential Smoothing (SES)

The simplest of all exponential smoothing methods are called Simple Exponential Smoothing (SES). This method is mainly suitable for data that do not contain a clear trend or seasonality. As mentioned earlier, this method is a balance between the naïve and the mean method. It is commonly used as a method to clear out the signal from a noisy data, hence the name - smoothing. As we know, the naïve method considers the most recent observation as the most significant. All other previous observations are ignored. This can also be perceived as a weighted average in which the entire weight is given to the most recent observation. On the other hand, the mean method treats all observations equally, which can be perceived as a weighted average with each observation given equal weights while generating forecasts.

Since we are aiming for something in between, the underlying concept behind exponential smoothing can be thought of as a weighted average where more recent observations are assigned larger weights than observations from the past. These weights decrease exponentially as more observations are realized from the future. This process can be mathematically represented by the following equation.

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots, \tag{12}$$

---

[13] As cited in Makridakis and Wheelwright (1989, pp.23 -24)

[14,15] As cited in Hyndman *et al.* (2008, pp.8 – 9)

Where $\boldsymbol{\alpha}$ is the smoothing parameter that varies between $0 \leq \alpha \leq 1$; it controls the rate at which the weights decrease (decay);

$\{\, \boldsymbol{\alpha}, \boldsymbol{\alpha}(\mathbf{1}-\boldsymbol{\alpha}), \boldsymbol{\alpha}(\mathbf{1}-\boldsymbol{\alpha})^{\mathbf{2}} \ldots \}$ constitute the weights assigned to the observations as they move from y_T, y_{T-1}, y_{T-2}, and so on.

The one step ahead forecast generated at time $T+1$ is a weighted average of all the observations in the series $y_1, \ldots, y_T$. The table below depicts the decay of weights for different values of α. Interestingly, the sum of the weights even for a small value of α will be approximately equal to one for any reasonable sample.

	$\boldsymbol{\alpha = 0.2}$	$\boldsymbol{\alpha = 0.4}$	$\boldsymbol{\alpha = 0.6}$	$\boldsymbol{\alpha = 0.8}$
$\boldsymbol{y_T}$	0.2	0.4	0.6	0.8
$\boldsymbol{y_{T-1}}$	0.16	0.24	0.24	0.16
$\boldsymbol{y_{T-2}}$	0.128	0.144	0.096	0.032
$\boldsymbol{y_{T-3}}$	0.1024	0.0864	0.0384	0.0064
$\boldsymbol{y_{T-4}}$	0.0819	0.0518	0.0154	0.0013
$\boldsymbol{y_{T-5}}$	0.0655	0.0311	0.0061	0.0003

decay

slow decay $\longrightarrow$ **fast decay**

Table 3. The relationship between smoothing parameter $\boldsymbol{\alpha}$ and rate of decay.

As one can notice from Table 3, when α takes any value between 0 and 1, the weights attached to the observations decay exponentially while going back in time. If α is small (close to 0), more weight is given to the past observations. On the other end of the spectrum, if α is large (close to 1), then more weight is given to the recent observations. Moreover, inputs of $\alpha = 1$ and $\alpha = 0$ represent the extremities, i.e. the naïve and mean forecasts respectively. This makes the assignment of α an important step in simple exponential smoothing.

Hyndman (2018) proposed two ways of representing the simple exponential smoothing mathematically. Both lead to the same equation (eq.12). The first approach is known as the weighted average form and is often found in many of the existing literature on exponential smoothing. It involves a simple mathematical derivation as shown below.

Let us assume that the forecast at time $T+1$ is equal to a weighted average between the most recent observation y_T and the previous forecast $\hat{y}_{T|T-1}$. It can be shown mathematically as follows.

$$\hat{y}_{T+1|t} = \alpha y_T + (1-\alpha)\hat{y}_{T|T-1} \tag{13}$$

Similarly, we can slightly modify the equation to accommodate fitted values. In simple terms, fitted values are one step forecasts at each time $t = 1, \dots, T$. The corresponding equation develops into the following expression.

$$\hat{y}_{t+1|t} = \alpha y_t + (1-\alpha)\hat{y}_{t|t-1} \tag{14}$$

Since the process has to start somewhere, let's assume the fitted value at $t = 1$ is represented by l_0. This makes up the initial value (initialization), which also needs to be estimated in addition to α. This creates the following chain of equations.

$$\begin{aligned}
\hat{y}_{2|1} &= \alpha y_1 + (1-\alpha)l_0 \\
\hat{y}_{3|2} &= \alpha y_2 + (1-\alpha)\hat{y}_{2|1} \\
\hat{y}_{4|3} &= \alpha y_3 + (1-\alpha)\hat{y}_{3|2} \\
&\dots \\
\hat{y}_{T|T-1} &= \alpha y_{T-1} + (1-\alpha)\hat{y}_{T-1|T-2} \\
\hat{y}_{T+1|T} &= \alpha y_T + (1-\alpha)\hat{y}_{T|T-1}
\end{aligned}$$

Substituting each equation into its next following equation, we get

$$\begin{aligned}
\hat{y}_{3|2} &= \alpha y_2 + (1-\alpha)[\alpha y_1 + (1-\alpha)l_0] \\
&= \alpha y_2 + \alpha(1-\alpha)y_1 + (1-\alpha)^2 l_0 \\
\hat{y}_{4|3} &= \alpha y_3 + (1-\alpha)[\alpha y_2 + \alpha(1-\alpha)y_1 + (1-\alpha)^2 l_0] \\
&= \alpha y_3 + \alpha(1-\alpha)2 + (1-\alpha)^2 y_1 + (1-\alpha)^3 l_0 \\
&\dots
\end{aligned}$$

$$\hat{y}_{T+1|T} = \sum_{j=0}^{T-1} \alpha(1-\alpha)^j y_{T-j} + (1-\alpha)^T l_0 \tag{15}$$

As T increases, the last term becomes infinitely small. So, this is how the weighted average form leads to the same forecast equation (eq.12). The second and more convenient approach is known as the component form and is useful only when the exponential smoothing is extended to handle the trend and seasonality. Using this approach particularly reduces the complexity created when more components need to be added to the equation. We will see this approach being used in the subsequent sections. It can be represented by the following sets of equations.

Forecast equation: $\hat{y}_{t+h|t} = l_t$ (16)

Smoothing equation (Level): $l_t = \alpha y_t + (1 - \alpha) l_{t-1}$ (17)

Where $\boldsymbol{\alpha}$ the smoothing parameter for the level ($0 \leq \alpha \leq 1$);

$\boldsymbol{l_t}$ is known as the unobserved level component (or smoothed value) of the series at time t.

We will discuss why it is called 'unobserved level component' when we deal with state-space models in the subsequent sections. The forecast equation (eq.16) indicates the similarity between the forecast value at time $t + h$ and the estimated level at time t. On the other hand, the smoothing equation (eq.17) represents a particular state, which in this case is the estimated level of the series observed at period t.

An important aspect of forecasting using exponential smoothing is the initialization step, whereby an appropriate value for l_0 must be chosen. It must be noted that initialization gains importance only if α takes a value close to zero. As α begins closing towards 1, initialization quickly becomes less significant. Many approaches have been proposed for initialization, such as simply choosing the first observed value as l_0. Another approach would be to take the average of the first four or five values in the data set and set it as the initial value (Makridakis, Wheelwright and Hyndman, 1998, pp.150-151). Since the manual estimation of the initial value is not an entirely reliable method, automatic estimation procedures like the least squares or maximum likelihood estimation are preferred for initialization.

Hence, we can also say that the parameter selection of α and l_0 can be made either subjectively or objectively. A forecaster may choose the values subjectively based on previous accumulated experience. However, the objective approach is more reliable because the estimation of these unknown parameters is data-driven. One such data-driven estimation method is usually used during regression, where the parameters are estimated by minimizing the sum of the squared residuals. This method is known as the SSE or "sum of squares errors" approach and can be represented mathematically as follows.

$$SSE = \sum_{t=1}^{T} (y_t - \hat{y}_{t|t-1})^2 \tag{18}$$

But unlike regression, there is no definite formula for determining the optimal parameters for the exponential smoothing equations. Therefore, R uses a nonlinear optimization algorithm instead. Fortunately, R handles all estimation procedures automatically by using the ses() function. The output of this function, when used for forecasting, is shown below following the corresponding plot shown in Figure 22.

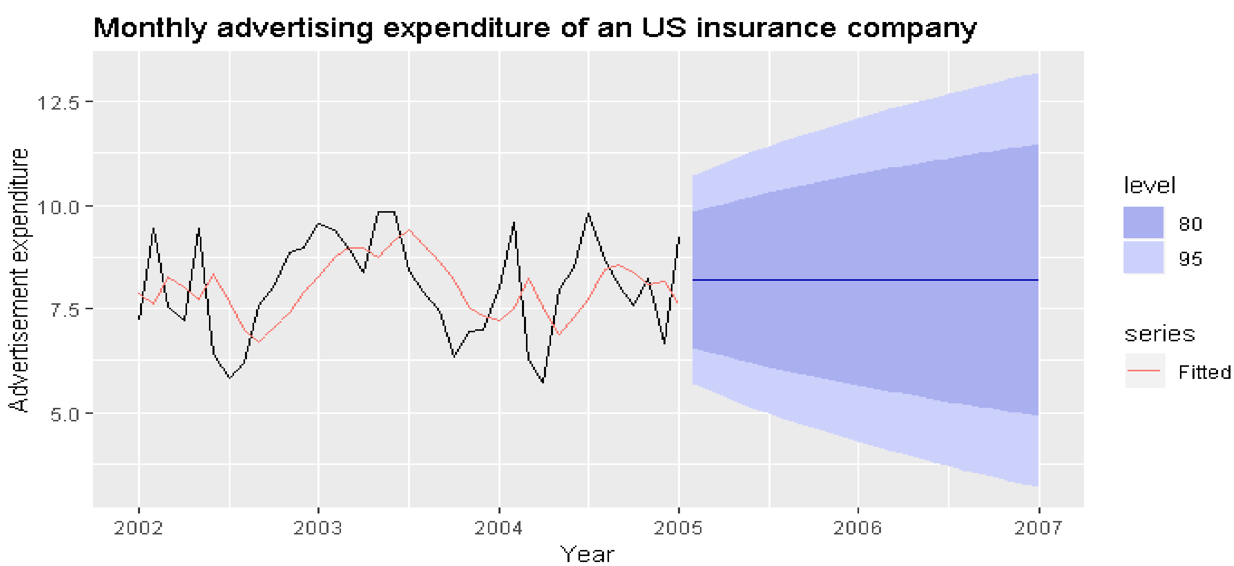


Figure 22. Forecasts generated using the simple exponential smoothing method.[15]

Output showing estimated parameters:

```
Forecast method: Simple exponential smoothing
Model Information:
Simple exponential smoothing
Call:
 ses(y = advertdata, h = 24)

 Smoothing parameters:
    alpha = 0.3611
 Initial states:
      l = 7.8569
```

[15] Source: Dataset insurance- Monthly television expenditure for a U.S. insurance company taken from the fpp2 package in R.

Holt's Linear Trend method

The simple exponential smoothing can be extended to handle trends and seasonality as well. The variation that handles trend is commonly known as the Holt's linear method, as it was developed by Holt (1957)[16]. Following the second approach we dealt with earlier, this method involves three equations- One forecast equation and two smoothing equations as shown below.

$$\text{Forecast Equation: } \hat{y}_{t+h|t} = l_t + hb_t \tag{19}$$

$$\text{Level Equation: } l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$\text{Trend Equation: } b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}$$

Where $\boldsymbol{b_t}$ is the trend estimate (slope) at time t;

$\boldsymbol{\beta}^*$ is the smoothing parameter for the trend, $0 \leq \beta^* \leq 1$.

The forecast equation now gives trended forecasts with slope b_t. The level equation is similar to that of the simple exponential smoothing. However, in order to handle trended data, the level equation (l_t) here adjusts the level considering the trend of the previous period (b_{t-1}) and adds it to the previously smoothed value (l_{t-1}). This eliminates any lag and brings l_t approximately in level with the current data value. The trend equation then updates the trend by taking the difference between the last two smoothed values. Any remaining randomness is removed by smoothing the trend using β^*in the last period ($l_t - l_{t-1}$) and adding it to the previous estimate of trend multiplied by $(1 - \beta^*)$. A small β^*indicates a trend that hardly changes in which case the data will have a trend close to linear throughout the series. Whereas a large β^*value indicates a rapidly changing trend which represents a series with a non-linear trend.

Sometimes the forecasts generated by this method shows a trend that extends infinitely, it possesses a tendency to over forecast. It is particularly common for forecasts produced in longer horizons. To solve this problem, another variation to the Holt's method was developed by Gardner and Mckenzie, (1985)[17] through the introduction of a damping parameter as seen in the following equations. This method is called the Damped trend method for exponential smoothing.

[16] As cited in Hyndman (2018).

[17] As cited in Hyndman (2018).

$$\text{Forecast Equation: } \hat{y}_{t+h|t} = l_t + (\delta + \delta^2 + \cdots + \delta^h) b_t \tag{20}$$

$$\text{Level Equation: } l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + \delta b_{t-1})$$

$$\text{Trend Equation: } b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)\delta b_{t-1}$$

Where, $\boldsymbol{\delta}$ is the damping parameter, $0 \leq \delta \leq 1$.

If $\delta = 1$, the method is similar to the Holt's linear trend method. The dampening effect can be realized by varying δ between 0 and 1. The larger the value of δ, the lesser the damping. Under this method, the short run forecasts are trended, and the long run forecasts flatten off in the end.

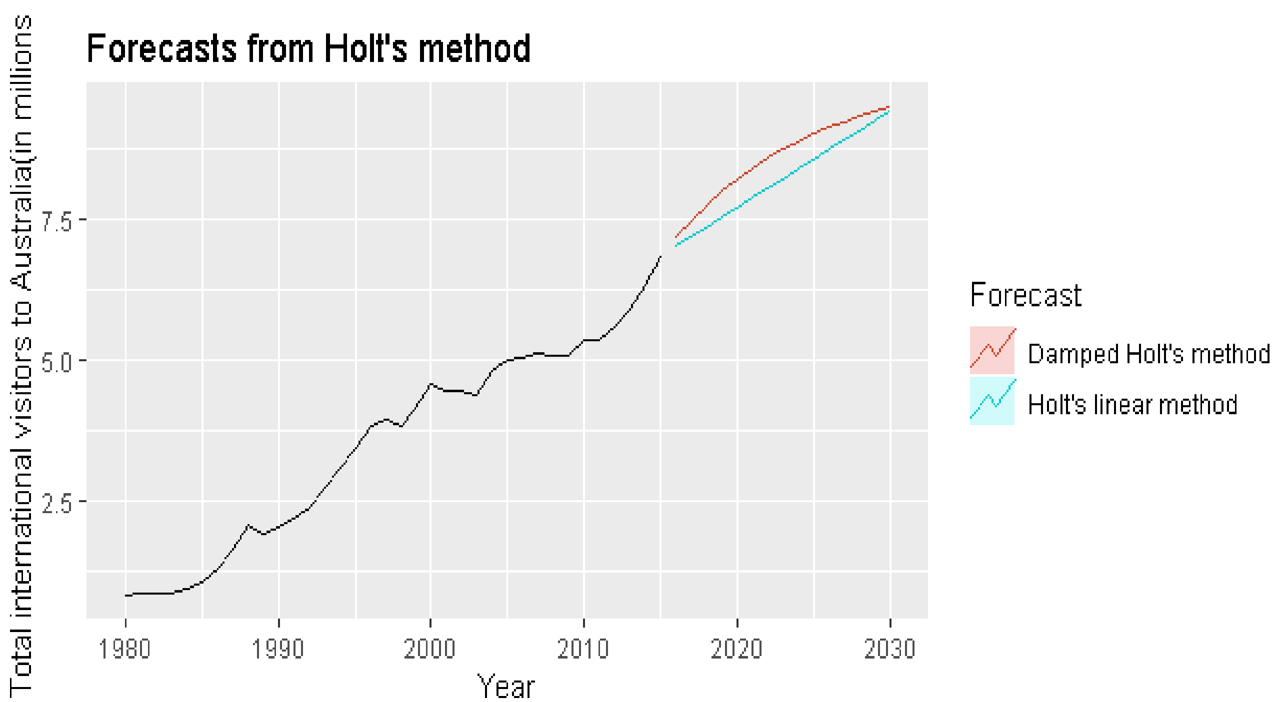


Figure 23. Forecasting using the Holt's linear and damped methods.[18]

There are now four parameters to estimate i.e. l_0., b_0, α, β^*. This is automatically handled by the holt() function in R. The parameter δ can either be set manually or estimated automatically and the corresponding forecasts generated using this method is shown below in Figure 23.

[18] Data source: Preloaded dataset austa taken from the fpp2 package in R.

Holt – Winter's seasonal method

The Holt's method was further extended to handle seasonality by Winters (1960)[19]. The equations representing this method look similar to the Holt's method with only a few minor changes. This time there are four equations – three smoothing equations in addition to one forecasting equation. As we have seen earlier during decomposition, seasonality can either take up the additive form or the multiplicative form. So, this method consists of two variations to cater to the two forms of seasonality.

Additive Seasonality: (21)

Forecast equation: $\hat{y}_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)}$

→ Level equation: $l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} - b_{t-1})$

→ Trend equation: $b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}$

→ Seasonal equation: $s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m}$

Multiplicative Seasonality: (22)

Forecast equation: $\hat{y}_{t+h|t} = (l_t + hb_t)\, s_{t+h-m(k+1)}$

→ Level equation: $l_t = \alpha\left(\frac{y_t}{s_{t-m}}\right) + (1 - \alpha)(l_{t-1} + b_{t-1})$

→ Trend equation: $b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}$

→ Seasonal equation: $s_t = \gamma\left(\frac{y_t}{l_{t-1} + b_{t-1}}\right) + (1 - \gamma)s_{t-m}$

Where,

$\boldsymbol{s_t}$ represents the seasonal component;

$\boldsymbol{\gamma}$ represents the smoothing parameter for seasonality, $0 \leq \gamma \leq 1$;

$\boldsymbol{m}$ is used to denote the frequency of seasonality, i.e. for yearly (m = 1), quarterly (m = 4) etc.

Shown above (eq.21 and eq.22) are the equations representing the additive form and the multiplicative forms of the Holt-Winter's method. Notice how these two sets of equations are essentially the same if addition/subtraction signs (in additive) are replaced by multiplication/division signs (in multiplicative). Here l_t consists of the weighted average between the seasonally adjusted part of the series $(y_t - s_{t-m})$ and the non-seasonal forecast $(l_{t-1} + b_{t-1})$ for time t. The trend equation is

[19] As cited in Hyndman (2018).

similar to the Holt's linear method. The seasonal equation shows the weighted average between the current seasonal index and the seasonal index of the same season in the previous year observed m time periods ago. Any randomness present in y_t is smoothed by the varying weights of γ.

Both the additive and multiplicative forms can be easily utilized for the purposes of forecasting. R automatically handles the initialization and smoothing parameter estimation. The hw() function can be used for forecasting using this method. From the plot below in Figure 24, we can clearly see that the multiplicative model does a better job at capturing seasonality as it changes with the level of the series compared to the additive model which maintains the same seasonality corresponding to the level as it moves ahead in time.

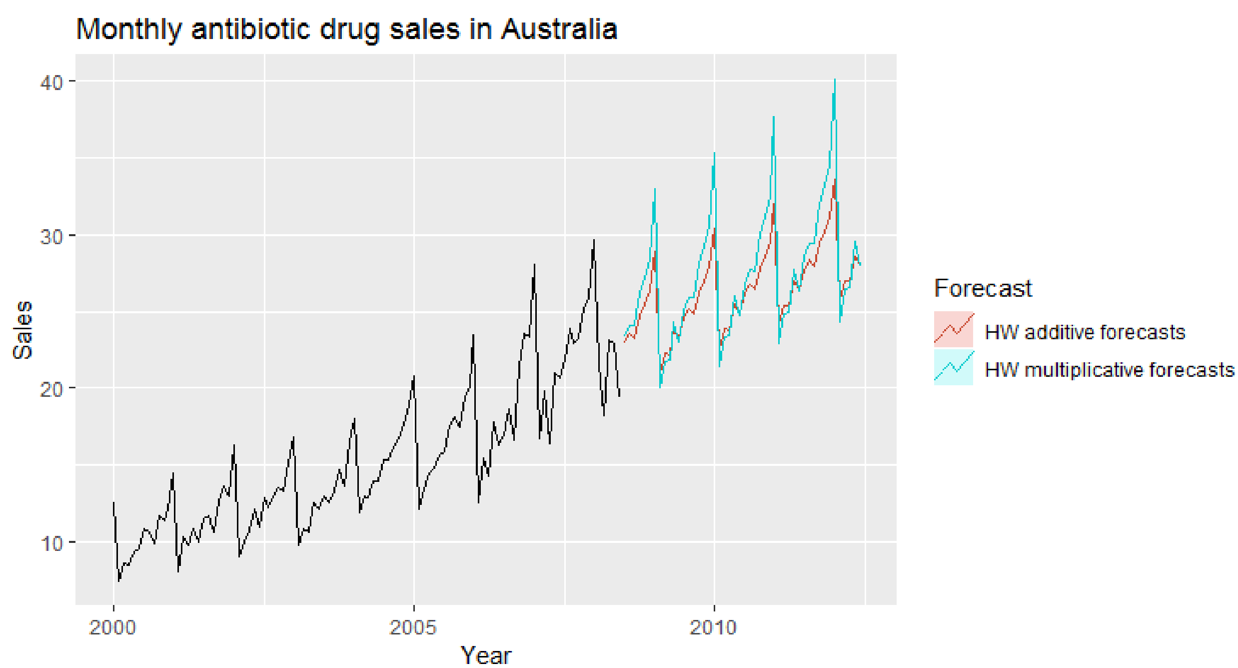


Figure 24. Forecasts generated using the Holt-Winter's method of exponential smoothing.[20]

<u>The Family of Exponential smoothing methods</u>

Exponential smoothing methods are not limited to just the Simple, Holt's linear trend and Holt-Winter's methods. As we have seen earlier in Figure 3, time series can also contain a combination of different trend and seasonal variations. Due to the popularity and simplicity of the exponential

---

[20] Source: Dataset a10 – Monthly antibiotic drug sales in Australia taken from fpp2 package in R.

smoothing, these methods have been extended to handle these combinations of variations in trend and seasonality. This led to the development of a system for the classification of these methods. This type of classification was first proposed by Pegels (1969) and referred to as Pegel's taxonomy. This classification included methods handling Multiplicative trend. This classification was later extended by Gardner (1985) and Taylor (2003) to include methods that handle the damped versions of the additive and multiplicative trends respectively.

The latest version of this classification exists in Hyndman *et al.* (2008, pp.12). as shown in Table 4 below. The trend can be classified into - none (no trend), additive and additive (damped), multiplicative and multiplicative (damped) whereas the seasonality can be classified into none (no seasonality), additive and multiplicative and their damped versions. Notice that the multiplicative and additive versions are just modified versions of the respective damped counterparts and are obtained by simply substituting $\phi = 1$. Each of these methods included in this table consist of a recursive formula for computing point forecasts h periods ahead. The recursive formulas for the heuristic methods of simple exponential smoothing, Holt's linear trend and Holt Winter's methods were already discussed in the previous section. The remaining equations can be found in Appendix 2.1a at the end of this thesis.

	Seasonal Component		
Trend Component	**N**(None)	**A**(Additive)	**M**(Multiplicative)
N (None)	**N N**	**N A**	**N M**
A (Additive)	**A N**	**A A**	**A M**
M(Multiplicative)	**M N**	**M A**	**M M**
A (Additive Damped)	**A N**	**A A**	**A M**
M(Multiplicative Damped)	**M N**	**M A**	**M M**

Table 4. Taxonomy of Exponential smoothing methods.[21]

It would be interesting to know which component mix resembles which exponential smoothing method. Therefore, the following table clearly portrays this clearly by relating the components to the

[21] Adapted from Hyndman *et al.* (2008, pp.12).

methods discussed so far (Hyndman *et al.*, 2008). The corresponding R functions are provided next to these methods to show R's competence in handling the different variations of these methods.

Component mix	Method	R function
N N	Simple Exponential Smoothing	ses()
A N	Holt's linear Method	holt()
Ad N	Holt's damped trend method	hw()
A A	Additive Holt-Winter's method	hw()
A M	Multiplicative Holt-Winter's Method	hw()
Ad M	Damped multiplicative Holt-Winter's method	hw()

Table 5. The resemblance to previously discussed heuristic methods.[22]

4.2 Advanced models

Due to the probabilistic nature of many business problems, it is difficult to predict the behavior of such time-dependent processes, which are usually affected by many unknown factors. Due to this reason, deterministic models are not well suited for forecasting. However, it is possible to build models based on the probabilities of future values that lie within specified limits (Box *et al.*, 2016, pp.6-7). These models are known as stochastic models and are can be used to model the behavior of many different time series. This paper will focus mainly on such type of models. Until now we only dealt with forecasting methods for time series that can handle variations in trend and seasonality by allowing them to change over time. As mentioned earlier, these methods lacked any theoretical foundations and so were aptly called heuristic methods. But the main drawback of such methods is that they fail to capture the uncertainty involved in their forecasts. To achieve this, we must use a stochastic model which usually governed by a predefined set of assumptions. In this section, we will discuss two popular modelling procedures which allow the representation of forecast uncertainty. The first procedure is called the state space modelling approach and the other is called ARIMA modelling approach. We will first discuss the state space approach and showcase how these models find their way back to the well-established heuristic methods that we discussed earlier.

[22] Hyndman (2018).

4.2.1 State space (ETS) models

State space models are extremely flexible and provide a simple modelling framework to connect methods and models. Any time series model can be described using this framework (Gooijer and Hyndman, 2006). These models usually consist of a measurement equation that defines the behavior of the observed data, and a couple of state equations that represent the change in states (unobservable components) over time. Hence, they are widely referred to as state space models (Hyndman, 2018). The flexibility offered by this modelling approach is particularly helpful during the specification of the parametric structure of a model (Hyndman *et al.*, 2008, pp. 6).

In this paper, we will particularly deal with the innovations form of state space models and use them to define the statistical framework for exponential smoothing methods that we discussed earlier. Here 'innovation' indicates that all equations assume the same random error process implying that the errors follow a normal distribution with zero mean and variance one. Further details regarding the sources of error can be found in Hyndman *et al.* (2008, pp.6-7). Earlier, these methods had ignored the error component. But in the realm of state space modelling, errors play a substantial role and their effect must be accounted for by making certain assumptions. Usually, the behavior of the error component is ignored, because the distinction between additive and multiplicative errors makes no difference to point forecasts (Hyndman *et al.*, 2008, pp.11). But even though the point forecasts for these models remain the same for a given set of parameters, their prediction intervals will differ greatly.

To better understand the intricacies involved in building this model, we use an example. Here, we will formulate the state space variations for the simple exponential smoothing method. Each method will consist of two models, one representing additive errors and the other for multiplicative errors. The mathematical derivations and equations that are used in this section are as proposed in Hyndman, (2018).

<u>Simple Exponential Smoothing (Additive errors)</u>

By using the component form of simple exponential smoothing (eq. 16 and 17), we get as follows.

Forecast equation: $\hat{y}_{t+h|t} = l_t$ (23)

Smoothing equation (level): $l_t = \alpha y_t + (1-\alpha) l_{t-1}$ (24)

Rearranging the smoothing equation for the level (eq. 24), we get as follows.

$$l_t = l_{t-1} + \alpha(y_t - l_{t-1})$$

$$l_t = l_{t-1} + \alpha e_t \tag{25}$$

$$e_t = y_t - l_{t-1} = y_t - \hat{y}_{t|t-1} \tag{26}$$

Where $\boldsymbol{e_t}$ is referred to as the residual at time t.

The eqn. (25) is called the "error correction" form. The residual errors constantly keep adjusting the estimated level throughout the smoothing process for t = 1…. T. For example, for a negative error at a time 't', the actual observation will be less than the forecast ($y_t < \hat{y}_{t|t-1}$). This implies that the level at time $t - 1$ is over-estimated. The new level l_t will then adjust itself lower than the previous level l_{t-1} . Like before, the value of α plays an important role in controlling the rate at which the level changes. For α closer to 1, the level estimates appear to be jagged, implying sudden adjustments to the level. On the contrary, as α moves away from 1, smaller adjustments result in a smoother level.

By rearranging terms in (eq. 26), each observation can be described by the previous level plus some error as follows.

$$y_t = l_{t-1} + e_t \tag{27}$$

To convert this into an innovations state space model, we need to simply specify the probability distribution for e_t. These residual errors are hence characterized by the following assumptions as shown in Table 6.

Statistical Assumption	Reason
The expected value (mean of the distribution) is zero.	We need to assume that there is no bias in the measurement process; otherwise, the observed values would not reflect the true value of that process (e.g. Risk of some sales going unreported or returned items not being added to inventory Figures).
The errors for different time periods are independent of (or at least uncorrelated with) one another and independent of past states.	If errors are related, it indicates the presence of still some uncaptured pattern, we can improve the forecast by using this information. If we have the correct model, such correlations will be zero.

The variance of the errors is constant. The common variance is denoted by σ^2.	If errors are increasing [decreasing] in absolute magnitude over time, the stated prediction intervals for future time periods would become too narrow [wide].
The errors are drawn from a normal distribution	A distributional assumption is necessary in order to make inferences. The normal distribution is by far the most common choice, but the use of an empirical distribution of errors that is based on the observed errors in the past is becoming increasingly popular.

Table 6. The statistical assumptions describing the residual errors.[23]

Only when these assumptions are justified (at least approximately) we can use the model to make valid inferences (such as setting up prediction intervals). Also, note that these assumptions imply that now the random errors are independent and follow a normally distributed Gaussian white noise series with zero mean and a common variance, symbolized by $e_t = \varepsilon \sim NID(0, \sigma^2)$. Moreover, note that these assumptions also hold for the random error terms that occur in ARIMA models discussed in the next section. Once the assumptions hold true (at least approximately), the equations for the model can then be generated as follows.

$$y_t = l_{t-1} + \varepsilon_t \tag{28}$$

$$l_t = l_{t-1} + \alpha\varepsilon_t \tag{29}$$

The (eq. 28) makes up the measurement equation (observation) and (eq. 29) makes up the state (transition) equation. The measurement equation shows how the observations and unobserved states relate to each other. These two equations in combination to the statistical distribution of errors make up a fully specified innovation state space model for the simple exponential smoothing with additive errors. In this case, observation y_t is a linear function of the level l_{t-1}, (the predictable part of y_t), and the error e_t, forms the unpredictable part of y_t. For other innovations state space models, this relationship may also be nonlinear (Hyndman, 2018).

[23]Source: Adapted from Ord and Fildes (2013, pp.132-133) and Makridakis *et al.* (1987).

Random Walk model

The state space representation for the naïve method can also be easily created from the previously obtained equations as follows (Ord and Fildes, 2013, pp.133-134).

When $\alpha = 1$, (eq. 29) becomes,

$$l_t = l_{t-1} + \varepsilon_t \tag{30}$$

We also know from (eq. 28) that,

$$y_t = l_{t-1} + \varepsilon_t \tag{31}$$

Comparing (eq. 30) and (eq. 31) we notice that the R.H.S are equal for both equations. Fusing both equations into a single equation we get the following.

$$y_t = y_{t-1} + \varepsilon_t \tag{32}$$

The above equation represents a time series where the mean level is equivalent to the previously observed value. This is known as the *random walk* model.

Simple Exponential Smoothing (Multiplicative errors)

Just like the additive form, we can specify the model for the multiplicative form by representing the residual errors as relative errors.

$$\varepsilon_t = \frac{y_t - \hat{y}_{t|t-1}}{\hat{y}_{t|t-1}} \tag{33}$$

Where $\varepsilon \sim NID(0, \sigma^2)$. Substituting $\hat{y}_{t|t-1} = l_{t-1}$ gives the following.

$$y_t = l_{t-1} + l_{t-1}\varepsilon_t \tag{34}$$

$$e_t = y_t - \hat{y}_{t|t-1} = l_{t-1}\varepsilon_t \tag{35}$$

Then the equations for the multiplicative form of innovations state space models become the following.

Observation/measurement equation: $y_t = l_{t-1}(1 + \varepsilon_t)$ (36)

State equation: $l_t = l_{t-1}(1 + \alpha\varepsilon_t)$ (37)

Similarly, we can write the innovation state space models for all exponential smoothing methods. This can be found in appendices – (2.1b) representing additive errors and (2.1c) representing multiplicative errors, found at the end of this thesis.

<u>Innovations form of state space models (ETS models)</u>

ADDITIVE ERROR MODELS

Trend	**Seasonal**		
	N	**A**	**M**
N	$y_t = \ell_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/\ell_{t-1}$
A	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + b_{t-1})$
$\mathbf{A_d}$	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + \phi b_{t-1})$

MULTIPLICATIVE ERROR MODELS

Trend	**Seasonal**		
	N	**A**	**M**
N	$y_t = \ell_{t-1}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$	$y_t = (\ell_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \alpha(\ell_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + s_{t-m})\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m}(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1}(1 + \alpha\varepsilon_t)$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
A	$y_t = (\ell_{t-1} + b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
$\mathbf{A_d}$	$y_t = (\ell_{t-1} + \phi b_{t-1})(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$

Figure 25. The ETS framework and the corresponding state space models.[24]

If we extend the taxonomy of exponential smoothing methods to their state space form, we will need to have a separate notation for additive and multiplicative errors. To achieve this, a third letter is added

[24] Hyndman (2018).

to the classification shown in Table 4 and each model is labeled according to the notation E, T, S (Error, Trend, and Seasonality). Now the possibilities for each component become Error {A,M}; Trend{N,A,A_d,M, M_d} and Seasonal{N,A,M}, generating a total of 30 combinations (refer Appendix 2.1b and 2.1c). However, we won't consider the multiplicative trend and its damped version since they tend to produce inaccurate forecasts. This leaves us with 18 combinations as shown in Figure 25.

The parameters of the state space model are estimated using maximum likelihood estimation (Hyndman *et al.*, 2002). It is a concept that is based on the probability of the data arising from the model. We will not discuss it further in this paper since R handles the parameter estimation automatically. For those interested in its mathematical details, refer to Hyndman *et al.* (2008, p.24). Alternatively, the parameters and initial states could also be estimated by minimizing the least square error as discussed previously, but it is rarely used nowadays.

One advantage of the ETS framework is the simplicity it provides using an information criterion for model selection. The best model is chosen by using the Akaike's Information Criterion (AIC). For a detailed explanation of this concept and other similar information criteria, refer to Hyndman *et al.* (2008,pp.105-106). These selection criteria are also commonly referred to as the in-sample model selection criteria (Sharma *et al.*, 2018). It can be mathematically represented by the following equation as proposed by Akaike (1974).

$$AIC = -2\, log(L) + 2K \tag{38}$$

Where ***L*** is the likelihood of the model;

K is the total number of parameters and initial states that have been estimated; this includes the residual variance.

A more common approach is to use a tweaked version of this information criteria, which is basically AIC corrected for small sample bias. It is usually referred to as AIC_c and can be mathematically represented by the following equation (Sugiura, 1978).

$$AIC_c = AIC + \frac{k(k+1)}{T-k-1} \tag{39}$$

Where ***T*** is the number of observations used for the estimation.

Forecasting using the ETS function in R

Now that by now substantial insight into the theoretical underpinnings of ETS models have been provided in this paper. But the main intention is to show how it can be applied to forecasting. This can be quite easily performed by using the ets() function in R's forecast package. Its ability to handle different types of variations in trend and seasonality, and its automated parameter estimation and model selection features makes this type of modelling very powerful while maintaining simplicity when used to generate forecasts. It is also very accurate and has a proven track record. This automatic forecasting function was used during the IJF-M3 competition data (Makridakis and Hibon, 2000), where it was found that this modelling procedure did very well in forecasting the short- term, beating all other methods in the competition (Hyndman *et al.*, 2008, p.28). Therefore, accurate forecasting is now no longer restricted to researchers and mathematicians only. Now it is within the grasp of business managers as well, who will benefit from accurate forecasts the most during their planning operations.

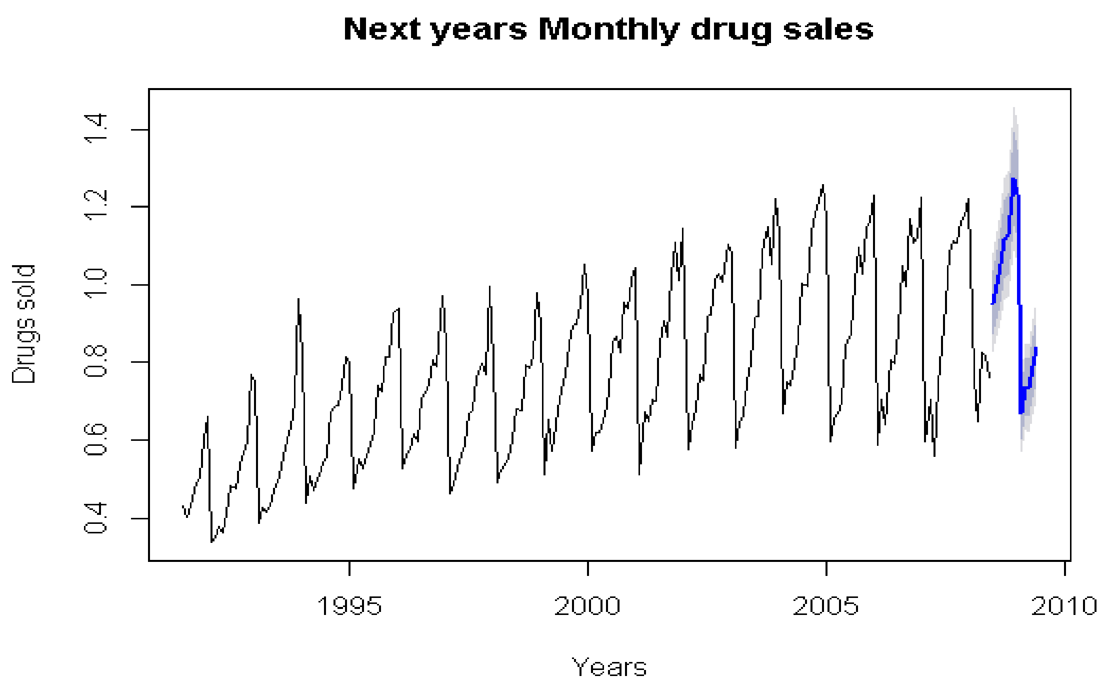


Figure 26. Time series plot showing forecasts generated by the ETS model.[25]

The difference between the ets() function when compared to other functions like holt() or hw() is that the ets() function does not perform forecasting by itself. It only returns the best fitting model instead.

[25] Source: Dataset h02- Monthly cortisteroid drug sales in Australia taken form the fpp2 package in R.

This model must then be passed to the forecast() function to generate forecasts. The R code used for this operation can be found in the Appendix A1 at the end of this thesis. Additionally, note that R's built-in pipe operator %> % makes it easier to see this passing on of arguments to functions. The resulting forecasts can be depicted in the following plot.

The ets() function also throws the following output that contains the estimated smoothing parameters and initial values for the corresponding model chosen according to the lowest AICc. This is shown in the sample snippet below.

<u>Output:</u>

```
ets(h02)
ETS(M,Ad,M)

Call:
 ets(y = h02)


  Smoothing parameters:
    alpha = 0.1953
    beta  = 1e-04
    gamma = 1e-04
    phi   = 0.9798


  Initial states:
    l = 0.3945
    b = 0.0085
    s = 0.874 0.8197 0.7644 0.7693 0.6941 1.2838
           1.326 1.1765 1.1621 1.0955 1.0422 0.9924

  sigma:  0.0676

      AIC       AICc        BIC
-122.90601 -119.20871  -63.17985
```

4.2.2 Box-Jenkins (ARIMA) models

Autoregressive Integrated Moving Average models or ARIMA class of models have been discussed extensively in forecasting literature. They were developed during the 1970s by George Box and Gwilym Jenkins, whose names frequently appear in tandem whenever ARIMA modelling is discussed. Their main contribution to this field was the development of a systematic procedure to understand and use these kinds of models for forecasting a vast variety of time series. This approach is commonly known as the Box-Jenkins approach.

Unlike the state space modelling approach that we discussed previously while dealing with exponential smoothing methods, this approach does not divide the model into different 'states' for representing the corresponding time series components. Here, the modelling methodology is more data-driven and the resulting models use the underlying interdependencies (autocorrelation) between the observations in a time series to describe the behavior of that data generating process (Ord and Fildes, 2013, p.152).

Autocorrelation function (ACF)

This interdependency displayed by observations is a rather common phenomenon in many time series that occur in the business world, making ARIMA models well suited for this application. However, it needs to be measured in order to prove useful to modelling tasks. The statistic used to measure the autocorrelation is known as the autocorrelation function (ACF). The autocorrelations of a time series can be graphically summarized using the ACF plots, which we used previously when we discussed various data exploration methods available for time series analysis in Chapter 3. It is very similar to the correlation used in regression, except here, instead of determining the strength of the relationship between the dependent and independent variables, it measures the correlation of the time series with its own periodically lagged values. The following formula describes the mathematical constructs of this statistical measure (Makridakis, Wheelwright and Hyndman, 1998, pp.313-314).

$$r_k = \frac{\sum_{t=k+1}^{n}(y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{n}(y_t - \bar{y})^2} \tag{40}$$

Where $\boldsymbol{\bar{y}}$ is the mean;

$\boldsymbol{k}$ represents the number of lags.

Stationarity

An important prerequisite for building ARIMA models is that the assumption of stationarity must hold. A time series is said to be stationary when the following three conditions are satisfied. (Ord and Fildes, 2013, pp.154-155).

- It needs to have a constant mean (stationarity in the mean).
- It needs to have a constant variance (stationarity in the variance) or homoscedasticity.
- The autocorrelation structure remains constant over time. i.e. The autocorrelations between observations n lags apart must be the same, no matter where they occur in the series. For

example – The correlation between the months January and May should be the same as the autocorrelation between the same months during the previous year.

A white noise series is considered as a special case of stationarity. It is a time series that contains no systematic patterns like trend or seasonality and is mostly filled with random fluctuations (noise). It contains random, independent and identically distributed observations. The autocorrelations are minimum and fluctuate close to zero. So basically, a white noise series is a pure random time series with absolutely no systematic structure.

The following plots depicted in Figure 27 show the typical white noise process and demonstrate its properties. Notice the blue dotted lines in the ACF plot. These lines represent the 95% significance levels. The minimal or almost nil autocorrelations of a white noise process always lie within these lines and tend to fluctuate around zero, which implies an absence of any meaningful autocorrelation. The series itself follows a horizontal level and the variance remains constant throughout the series.

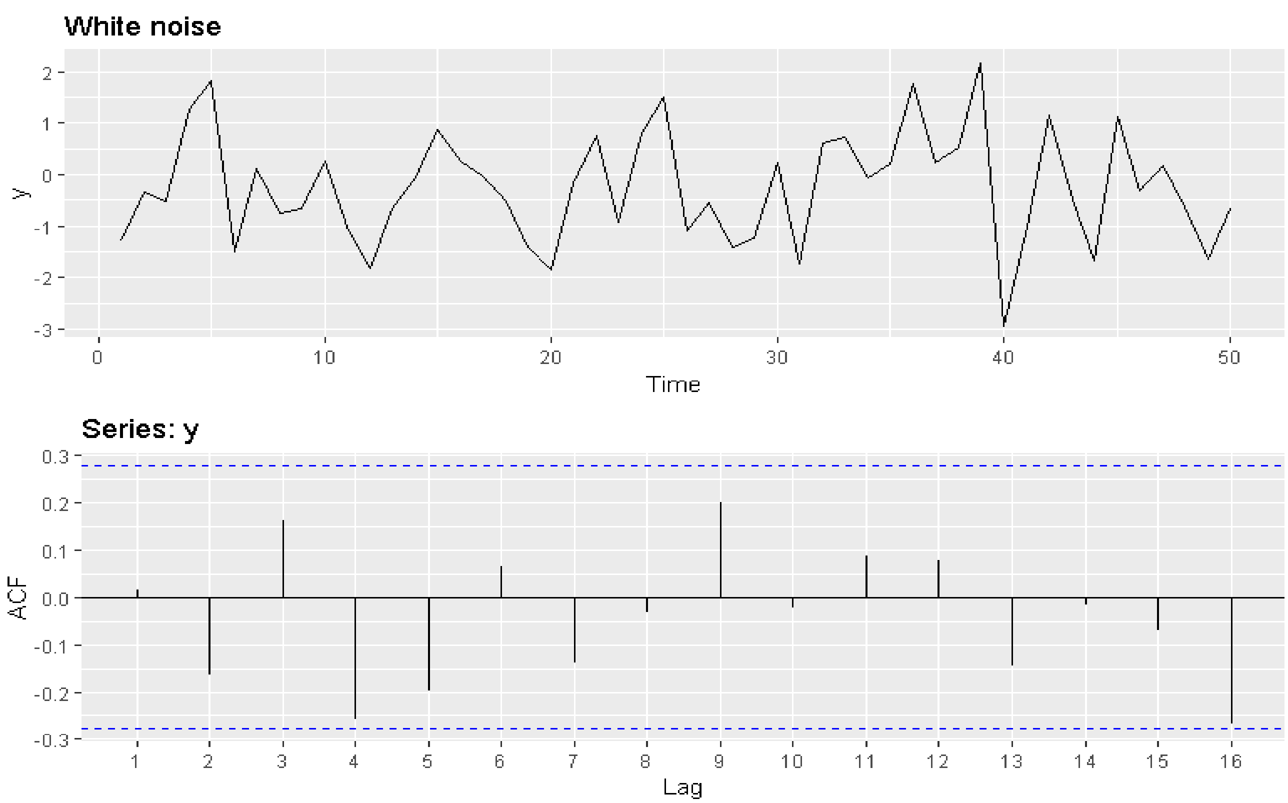


Figure 27. Time plot and ACF plot of a white noise series.

But unfortunately, most time series do not exhibit stationarity naturally. In fact, most time series observed in business are non-stationary since they don't satisfy one or more of the conditions of stationarity. This behavior can be demonstrated with the help of a few examples. Shown below in Figure 28 are three datasets. Each dataset disregards few or all conditions of stationary.

In plots A and B, one can easily notice that the mean is not constant due to the presence of the trend and seasonality respectively. In plot C, in addition to a varying mean, there is a change in variance with time (heteroscedasticity).

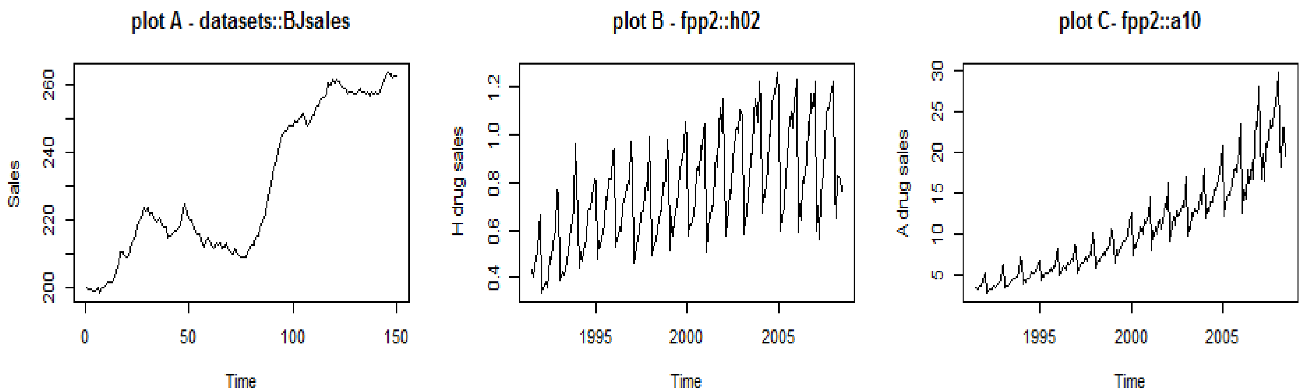


Figure 28. Time plots showing non-stationary characteristics of time series.[26]

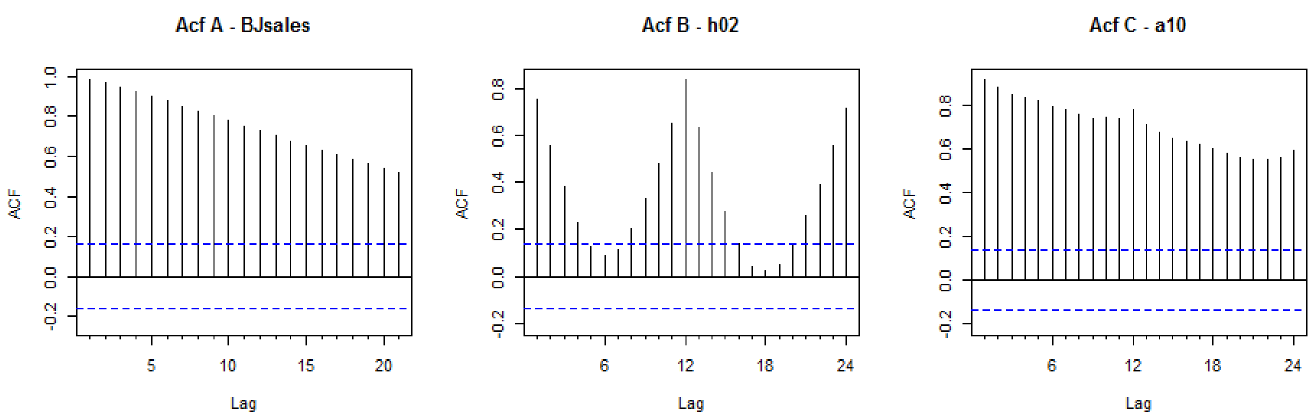


Figure 29. Typical ACF plots for non-stationary series.

When we compare the ACF plots of the above datasets with that of a white noise series, we can immediately notice their non-conformity to the white noise behavior. Furthermore, it is also important

[26] Source: Dataset BJsales – Sales data with leading indicator are taken from the datasets package in R; Dataset h02 – Monthly corticosteroid drug sales in Australia and Dataset a10 – Monthly antibiotic drug sales in Australia taken from the fpp2 package in R.

to note that non-stationary patterns such as the trend or seasonality induce large autocorrelations themselves, which overpower the true autocorrelation structure that we are interested in for modelling purposes. Thus, the removal of such dominating autocorrelations becomes a mandatory step before we begin fitting models to the series (Makridakis, Wheelwright and Hyndman, 1998, p. 326). Fortunately, there are many ways with which these superimposing non-stationary patterns can be stabilized resulting in either a stationary (white noise) series or revealing the underlying autocorrelation structure, which when subjected to further modelling using the AR, MA or ARMA modelling procedure, returns a stationary series. We will witness this procedure later in more detail.

<u>Differencing</u>

The most popular way to stabilize the non-stationarity is through *differencing* the series. Differencing involves computing differences between consecutive observations (Hyndman, 2018). It can be represented by the following equation as formulated by Makridakis, Wheelwright and Hyndman (1998, pp.366-327).

$$\acute{y}_t = y_t - y_{t-1} \tag{41}$$

Where $\boldsymbol{\acute{y}_t}$ represents the differenced series.

The differenced series will have one observation less $(n-1)$ than the original series(n) since it is impossible to calculate the difference otherwise. The resulting series after being differenced is also referred to as the 'lag'. An example of this procedure is shown in Table 7 as follows. Notice how the number of observations decreases as the series is lagged from Y_1 to Y_3.

Time	Original series Y	Series lagged by 1 period (Y_1)	Series lagged by 2 periods (Y_2)	Series lagged by 3 periods (Y_3)
t = 1	4	-3	6	8
t = 2	-3	6	8	6
t = 3	6	8	6	8
t = 4	8	6	8	13
t = 5	6	8	13	11
t = 6	8	13	11	2
t = 7	13	11	2	4
t = 8	11	2	4	
t = 9	2	4		
t = 10	4			

Table 7. Constructing time lagged series for differencing.

Differencing can be a tedious task if performed manually. Especially for large datasets and for cases when differencing once is not enough. Computation using a statistical computational software is recommended, which in our case happens to be R. Due to the software's versatility, it offers many different functions to determine the form (first or seasonal differencing) and the order of differencing (once, twice, etc.). The seasonal differencing is performed exactly like the lag1 differencing (lag 1,2,3, etc.), dissimilar only in terms of period when the lagged differences are taken. For example, for monthly data, the lags used are 12, 24, 32, etc. Similarly, for quarterly data, the lags used are 4, 8,12, 16, and so on.

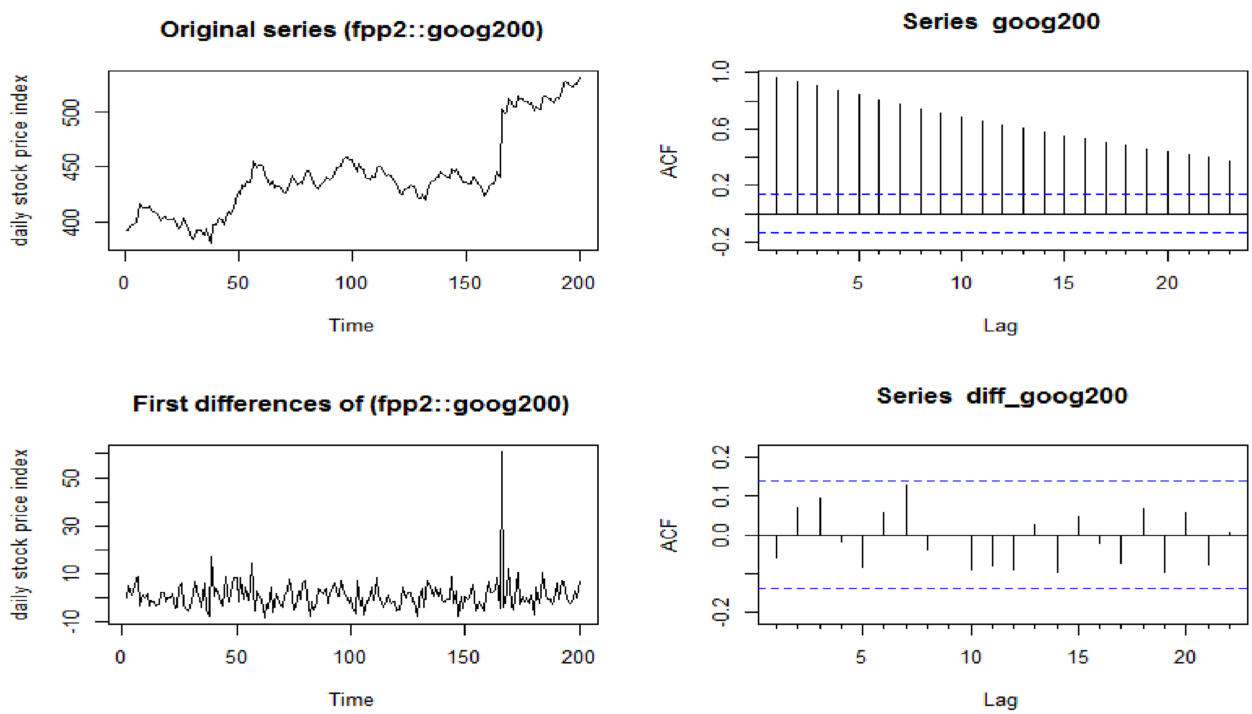


Figure 30. Characteristics of a random walk (RW) model and white noise model (after differencing RW).[27]

After the induced autocorrelations due to non-stationarity are removed, the true autocorrelation structure of the time series is unveiled. This is the autocorrelation that indicates the interdependencies between observations in the series. This interdependency is exactly what needs to be captured and it forms the basis for model building. Before moving on to discussing the types of models, it is important to note that in some special cases, it is possible that differencing directly leads to stationary (white

[27] Source: Dataset goog200- Daily closing stock prices of Google taken from the fpp2 package in R.

noise) series. This results in situations where further modelling is not necessary. Let us look at an example demonstrating such a scenario.

The time series depicted in the upper left plot in Figure 30 represents the daily closing stock prices of Google. It is quite evident that the series is non-stationary. The ACF plot also shows large positive autocorrelations due to the presence of long periods of trends, which behave peculiarly by changing directions unpredictably. However, by taking the first difference of such a series, it directly leads to the attainment of a stationary (white noise) series without any residual autocorrelation being leftover. Notice that the ACF plot resembles that of a white noise series. A model that is used to represent this behavior is commonly known as the random walk model. We can express it mathematically by designating the goog200 data with the notation y_t. Then we get,

$$y_t - y_{t-1} = e_t \tag{42}$$

Where $\boldsymbol{e_t}$ is the white noise. This can be rewritten as follows.

$$y_t = y_{t-1} + e_t \tag{43}$$

The above equation is a classical representation of the random walk model as proposed by Makridakis, Wheelwright and Hyndman (1998, p. 329). We will now continue referring to the previous example from Figure 30 above. The plots (A-C) are used again, and a procedure to systematically convert the different types of non-stationarity inherent to these time series into stationarity will be presented in the Figure 31.

R offers many useful functions to perform differencing easily. Functions such as ndiffs() and nsdiffs() that return the number of lag 1 and lag 12 differences (for monthly data) can be used. To perform differencing, we can use the diff(x) and diff (x, lag = 12) functions for lag 1 and seasonal differencing respectively. Here, x denotes the time series that is being modeled. The following Figure depicts how the procedure of differencing changes according to the type of data. The blue arrows represent the sequence that the transformations and differencing operations follow. The type of operation is specified in red text located above the blue arrows. The corresponding R codes used to create these graphs can be found in Appendix A1 found at the end of this thesis.

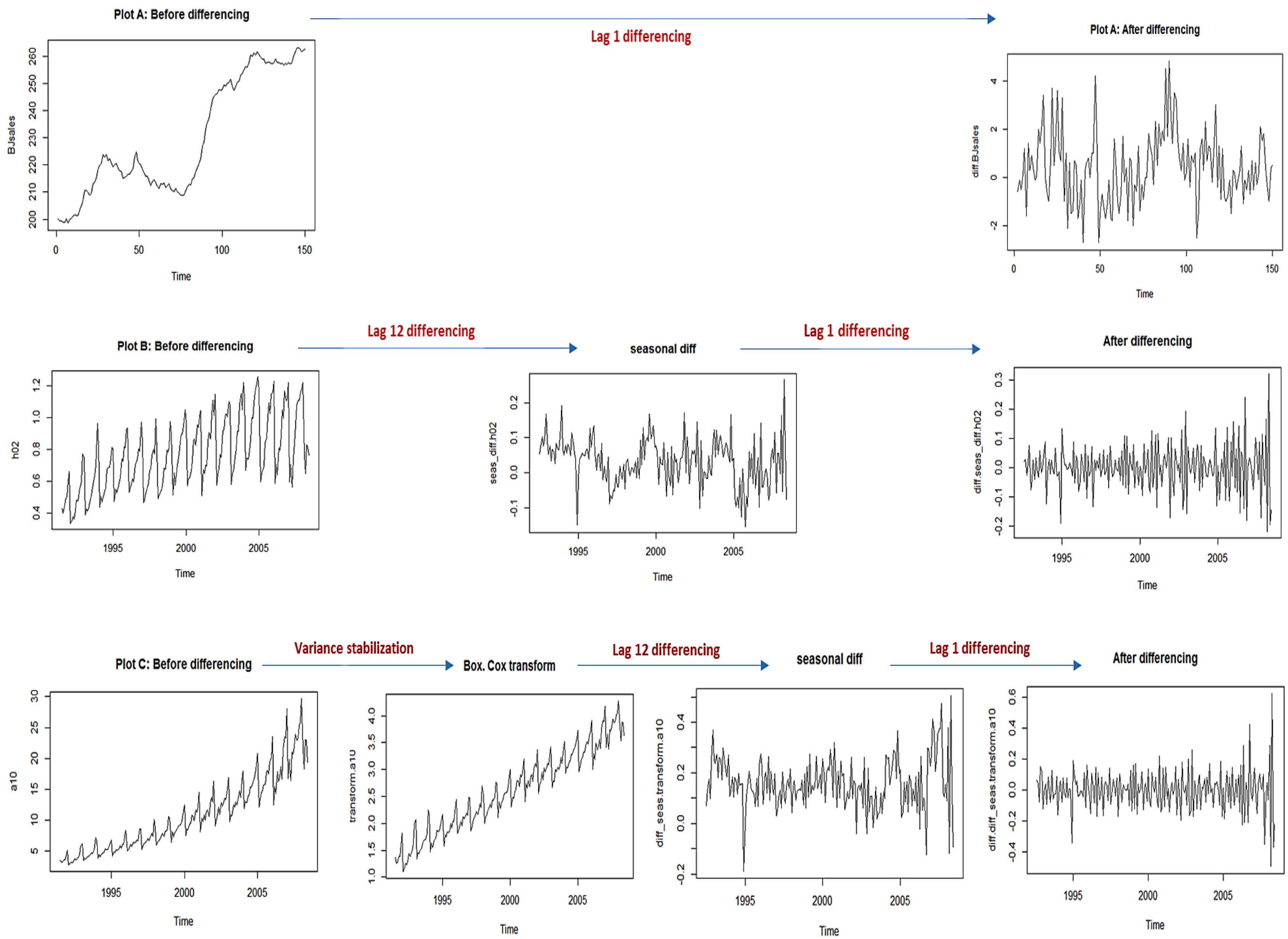


Figure 31. Time plots before and after differencing.

In Figure 31, Plot A needs to attain stationarity in its mean, which was formerly deterred by the presence of a trend. By differencing the series once, the mean stabilizes, leaving behind the pure autocorrelation structure unaltered by the effects of non-stationarity. Similarly, in Plot B a seasonal differencing can initially scrape off the seasonality, which is then followed by first differencing to remove the trend effects. Because of these two sets of differencing, stationarity in the mean is restored. However, the series depicted by Plot C is a bit complicated. In addition, to the large deviations in the autocorrelation induced by the trend and seasonality, it also shows heteroscedastic behavior, noticeable from the increasing variance as the series progresses with time.

This requires a combination of two types of differencing procedures (1 normal and 1 seasonal), and a transformation (as discussed in Chapter 3). The Box-Cox transformation will be used in this case to stabilize the variance. The resulting series then attains stationarity in its mean and variance, validating the first two conditions of stationarity. After the non-stationary effects due to trend, seasonality, and heteroscedasticity are completely removed by differencing efforts, any prevailing autocorrelations between the observations will then be clearly discernable and the series is said to attain stationarity.

Normally, the transformation is performed before the differencing takes place. And note that while performing differencing on a series with strong seasonality, it is recommended to take the seasonal (lag 12) difference before the lag 1 difference. The reason being that the initial seasonal difference may immediately transform the series into stationary, thereby eliminating the need for further lag 1 differencing cutting down on overall effort needed (Hyndman, 2018). The resulting ACF plots after differencing are also shown below in Figure 32.

When comparing these ACF plots to that of the white noise series one can notice that there are still spikes that overshoot the blue dotted line on all three ACF plots shown in Figure 32. Naturally, it indicates the presence of autocorrelation but from a more general perspective, this implies that there is still some useful information in the data that can be forecast. In contrast, in a white noise series, there will be absolutely nothing to forecast. Thus, a white noise series is used as a basis to check for any residual information that may have been left out during the analysis. Furthermore, many tests for residual analysis are also based on it. The dotted blue lines make it easier to determine whether the autocorrelation structure follows that of a white noise series.

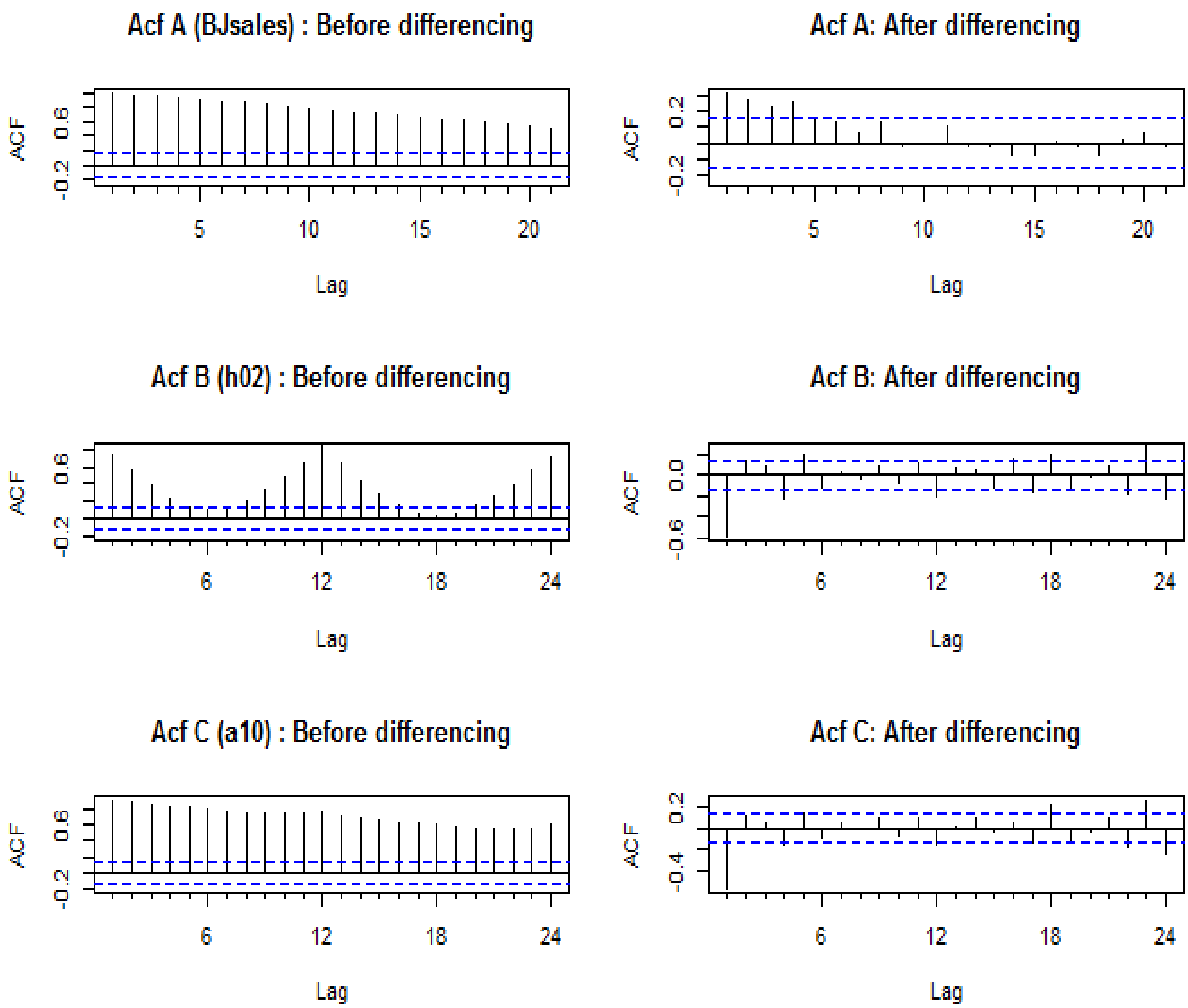


Figure 32. ACF plots before and after differencing.

ARIMA models

The ARIMA models can now be used to model the autocorrelation structure that is revealed after the differencing operation. This model is basically made up of AR (Auto-Regressive models) and MA (Moving Average) models. The equations used here are as proposed by (Makridakis, Wheelwright and Hyndman, 1998).

An *Auto-Regressive (AR)* model is simply a regression of the time series against the lagged values of that series. The last p observations are used as predictors in the regression equation. It can be mathematically represented by the following multiple regression equation.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t, \quad (44)$$

Where $\boldsymbol{e_t} \sim$ white noise;

$\boldsymbol{\phi}$ is the AR model parameter.

A *Moving Average(MA)* model can also be thought of as a regression. But instead of regression against lagged observations, we regress against lagged errors. The last q errors are used as predictors in the equation. This can be mathematically represented as following multiple regression equation.

$$y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q} \quad (45)$$

Where $\boldsymbol{\theta}$ is the MA model parameter.

When these two models are together, we get an *Auto-Regressive Moving Average (ARMA)* model with the last p observations and the last q errors, which are used simultaneously as predictors in the multiple regression equation as shown below.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q} + e_t \quad (46)$$

ARMA models can only work with stationary data so differencing will be needed if the data shows non-stationary behavior. This is represented by the 'I' in ARIMA. It denotes the number of times differencing is carried out to achieve stationary data. If an ARIMA model needs to be differenced 'd' times (lag 1 differences) then that model is referred to as an ARIMA (p, d, q) model.

Similarly, for a seasonal ARIMA model we have another set of $(P, D, Q)_M$ referring to the seasonal versions listed as follows.

D represents the number of seasonal differences needed to achieve stationarity.

P represents the number of seasonal AR lags $(y_{t-m}, y_{t-2m}, \dots, y_{t-Pm})$.

Q represents the number of seasonal MA lags $(\varepsilon_{t-m}, \varepsilon_{t-2m}, \dots, \varepsilon_{t-Qm})$.

M represents number of observations per year.

<u>Partial Autocorrelation function(PACF)</u>

In order to fit these models to data, it becomes increasingly important to find the values for p, d, q, and P, D, Q (in the case of seasonal data). These parameters cannot be simply determined by using time plots. Therefore, we will have to use the ACF plot together with its modified version - the PACF plot. PACF is based on the partial autocorrelations, which builds over the drawbacks of the autocorrelation function. For further details regarding this refer to Hyndman (2018). For our purposes, it is it would suffice to know that when the parameters are not easily discernable from the ACF plot, they can then be determined by checking the PACF plot. Moreover, the ACF and PACF usually take up some standard forms.

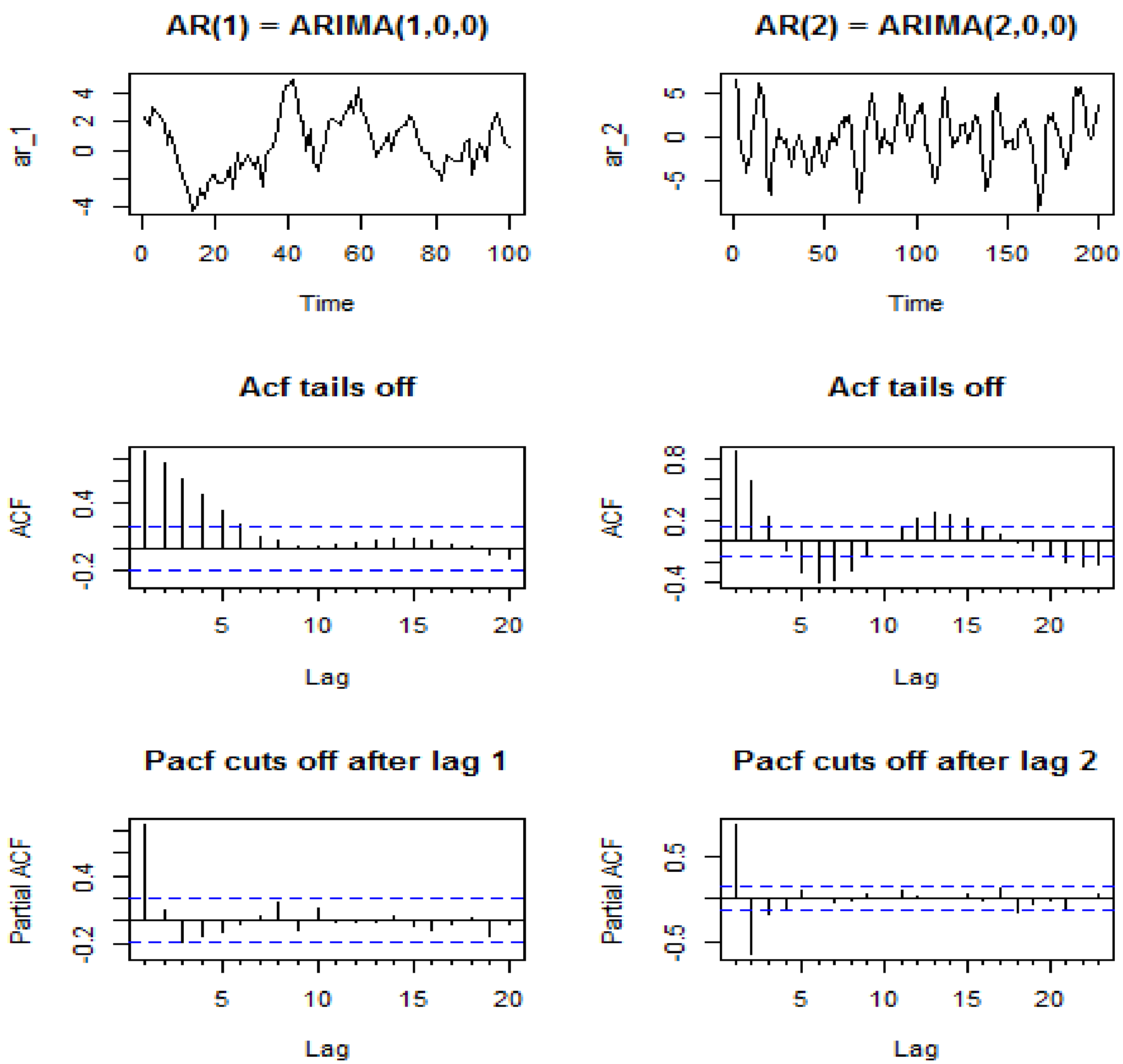


Figure 33. ACF and PACF behavior for AR models.

We shall demonstrate each of these standard forms by using a set of pure (without differencing) AR and MA models that were simulated in `R` using the `astsa` package. The `R` code can be found in

Appendix A1. For pure AR (p) models, the corresponding ACF and PACF plots look like in Figure 33. The spikes in the ACF plot tail off slowly, whereas the spikes observed in the PACF plot usually tend to cut off suddenly at lag p. For pure MA (q) models, the ACF and PACF look like Figure 34. Here the spikes in the ACF and PACF plots behave inversely to that of the AR models. In the case of MA models, the spikes in the ACF plot tend to cut off at lag q and the spikes in the PACF plot tends to gradually tail off.

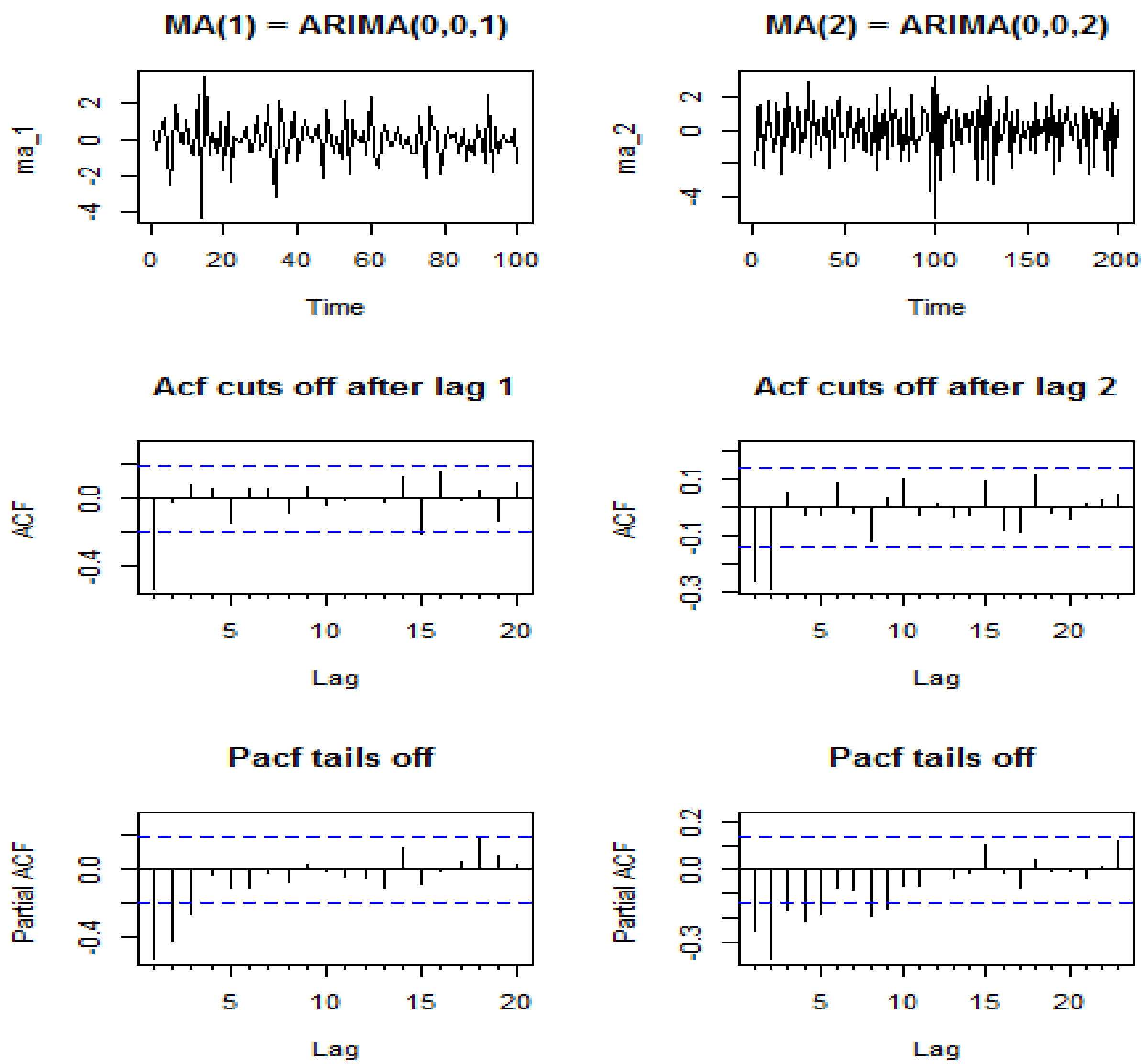


Figure 34. ACF and PACF behavior for pure MA models.

However, in the case of a pure ARMA(p, q) model, the ACF and PACF plots do not help in the selection of the model since it is not possible to clearly determine the right values for p and q. As seen in Figure 35 below, one model is an ARMA (1,1) and the other is ARMA (2,1). Both its ACF and PACF plots show their spikes ‘tailing off, thereby confirming the ARMA model type. But this behavior also makes it hard to deduce the correct p and q values. One way to overcome this problem is to initially start small by fitting an ARMA (1,1) model. This can then be followed by a residual analysis to check if the fitted model is good enough. If that model won’t suffice, then repeat the same process again, gradually adding more AR and MA parameters until the residuals look like white noise.

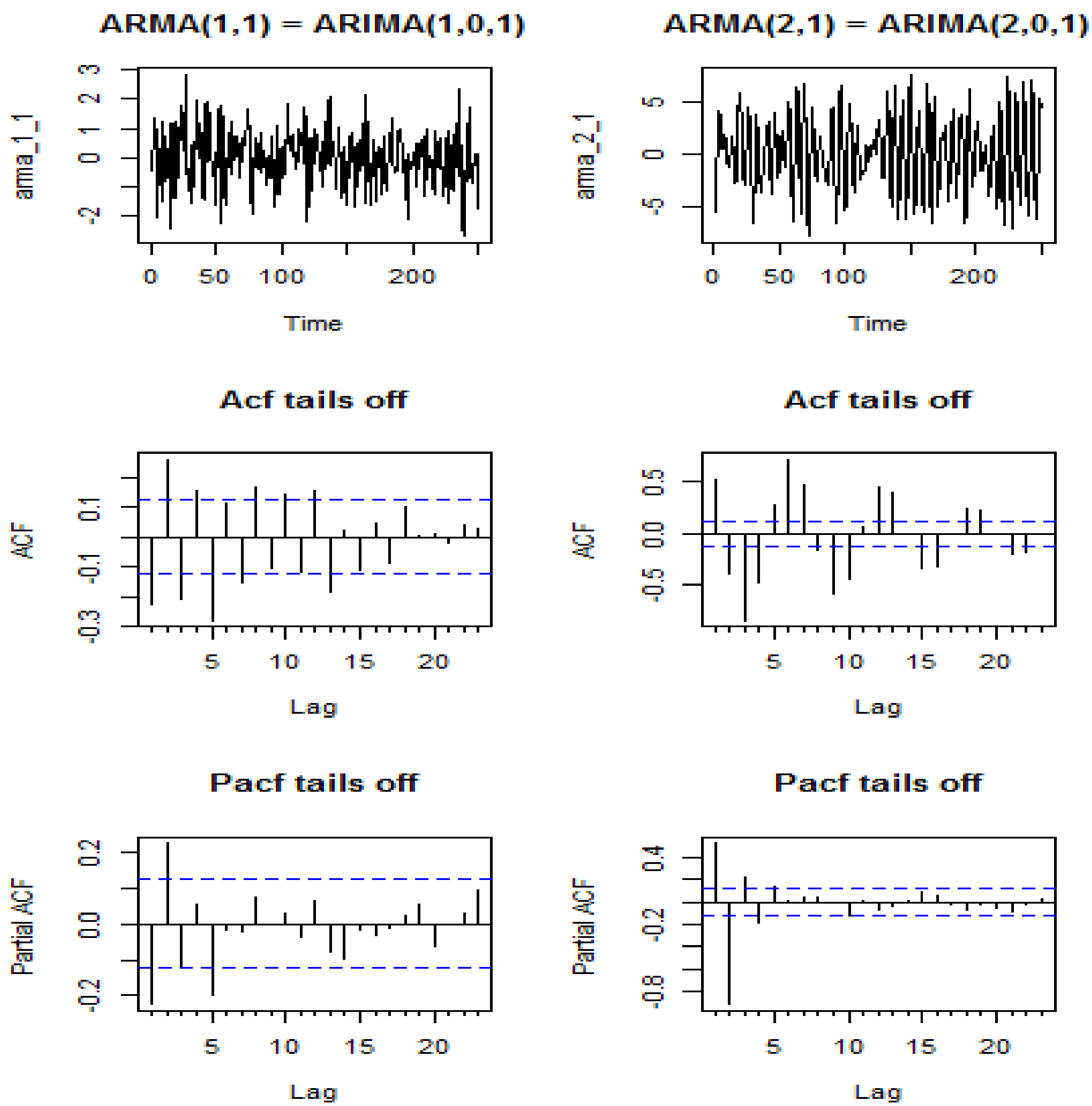


Figure 35. Typical ACF and PACF behavior for pure ARMA models.

The following table summarizes the behavior of ACF and PACF in terms of its corresponding ARMA model type.

	AR(p)	MA(q)	ARMA(p,q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off
For seasonal ARIMA models: Just replace the p and q with P and Q.			

Table 8. ACF and PACF behavior for AR, MA and ARMA models.[28]

The ACF and PACF behave similarly for seasonal ARIMA models, i.e. ARIMA $(\mathrm{P}, \mathrm{D}, \mathrm{Q})_m$ as well. The only difference being that the spikes occur at seasonal periods since the lags are seasonal. Often the seasonal models occur in combination with non-seasonal models. So, it would be beneficial to understand the procedure of model-order deduction using their combined form, i.e. ARIMA $(\mathrm{p}, \mathrm{d}, \mathrm{q})(\mathrm{P}, \mathrm{D}, \mathrm{Q})_m$, where m refers to the number of seasons per year. For example, we can use monthly data (where m = 12 or quarterly m = 4). To demonstrate this, we will use the following sample dataset as plotted in Figure 36.

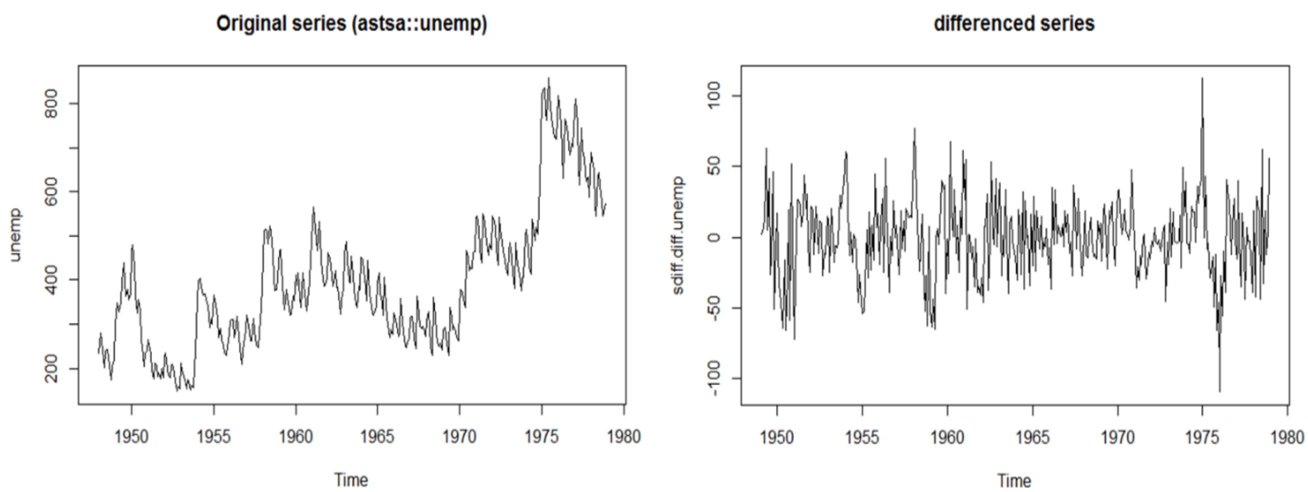


Figure 36. Differencing operation for seasonal series.[29]

[28] Simplified version of the table as adopted from Makridakis, Wheelwright and Hyndman (1998, p.342)

[29] Source: Dataset unemp - Monthly U.S. unemployment series taken from the astsa library in R.

It was found that the original time series contained both trend and seasonality. Thus, the time series had to first undergo one round of seasonal and lag 1 differencing to convert it into a stationary series The differenced series is shown beside the original series in Figure 36. The ACF and PACF plotted for the differenced series are shown below in Figure 37.

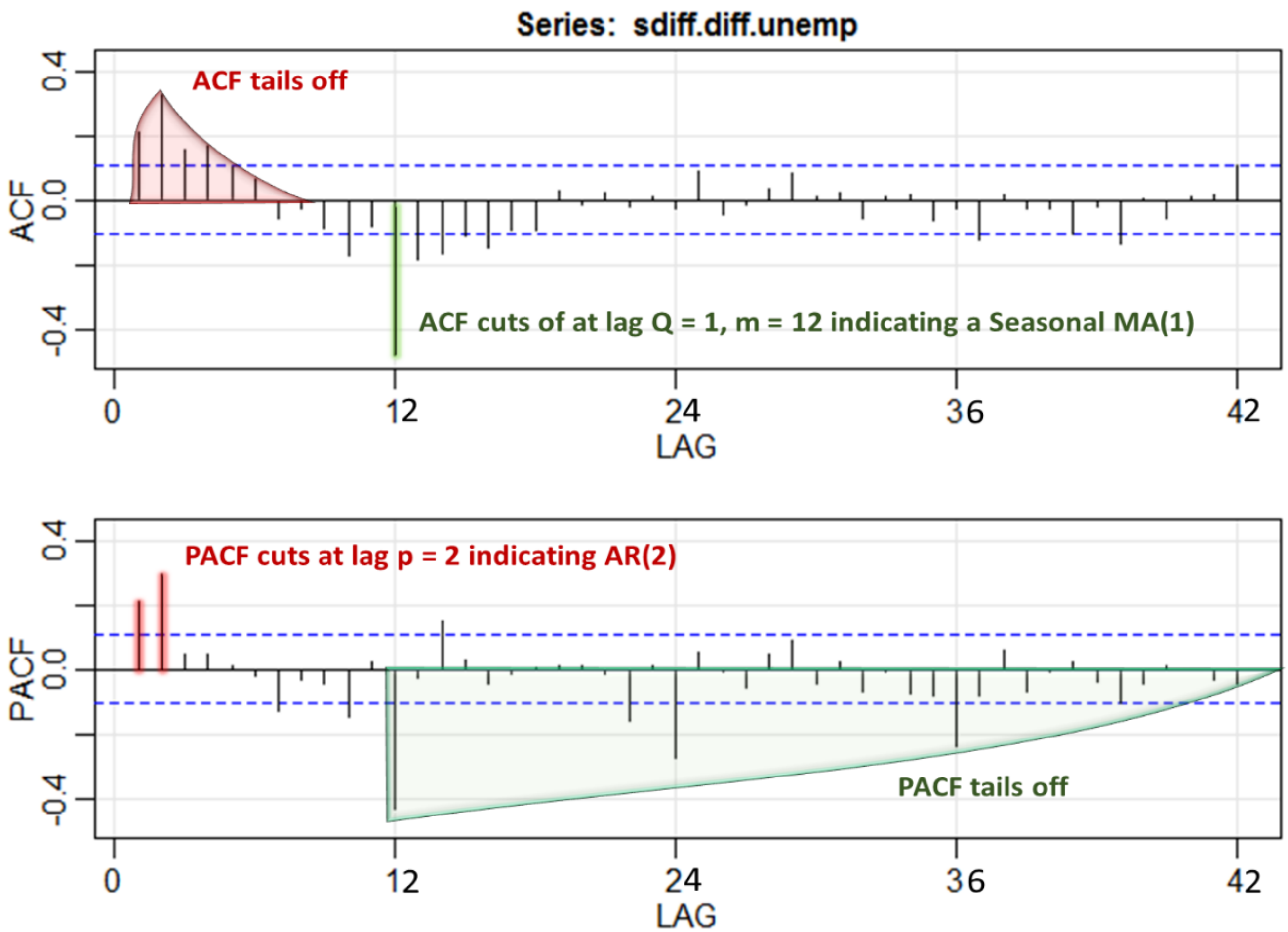


Figure 37. The ACF and PACF plots for seasonal ARIMA models.

<u>Residual Analysis</u>

From the plots depicted in Figure 36, we can deduce an ARIMA (2,1,0) (0,1,1) [12] model. To check if this is the right model, we should perform a residual analysis. It checks whether the residuals correspond to that of a white noise series. The residual analysis can be easily performed using R's checkresiduals() function. This function is part of the forecast package and provides the user with a large variety of tools to accurately examine the residuals after a model has been fit to the data. In particular, a call to the checkresiduals() function will generate a time plot, ACF plot and a histogram (to check for normality) of the residuals. In addition to that, it also performs a Ljung-Box portmanteau test whose output is provided below. The results of this analysis are as depicted in Figure 38 below. The p-value is just above the 0.05 threshold confirming the white noise behavior. Furthermore, it is clear from the plots that there is no substantial pattern left which would be useful; the autocorrelations also seem to lie within limits. The histogram indicates that the residuals behave normally and vary randomly around zero, imitating the behavior of a typical white noise series.

Output:

```
Ljung-Box test

data:  Residuals from ARIMA(2,1,0)(0,1,1)[12]
Q* = 31.327, df = 21, p-value = 0.06839

Model df: 3.   Total lags used: 24
```

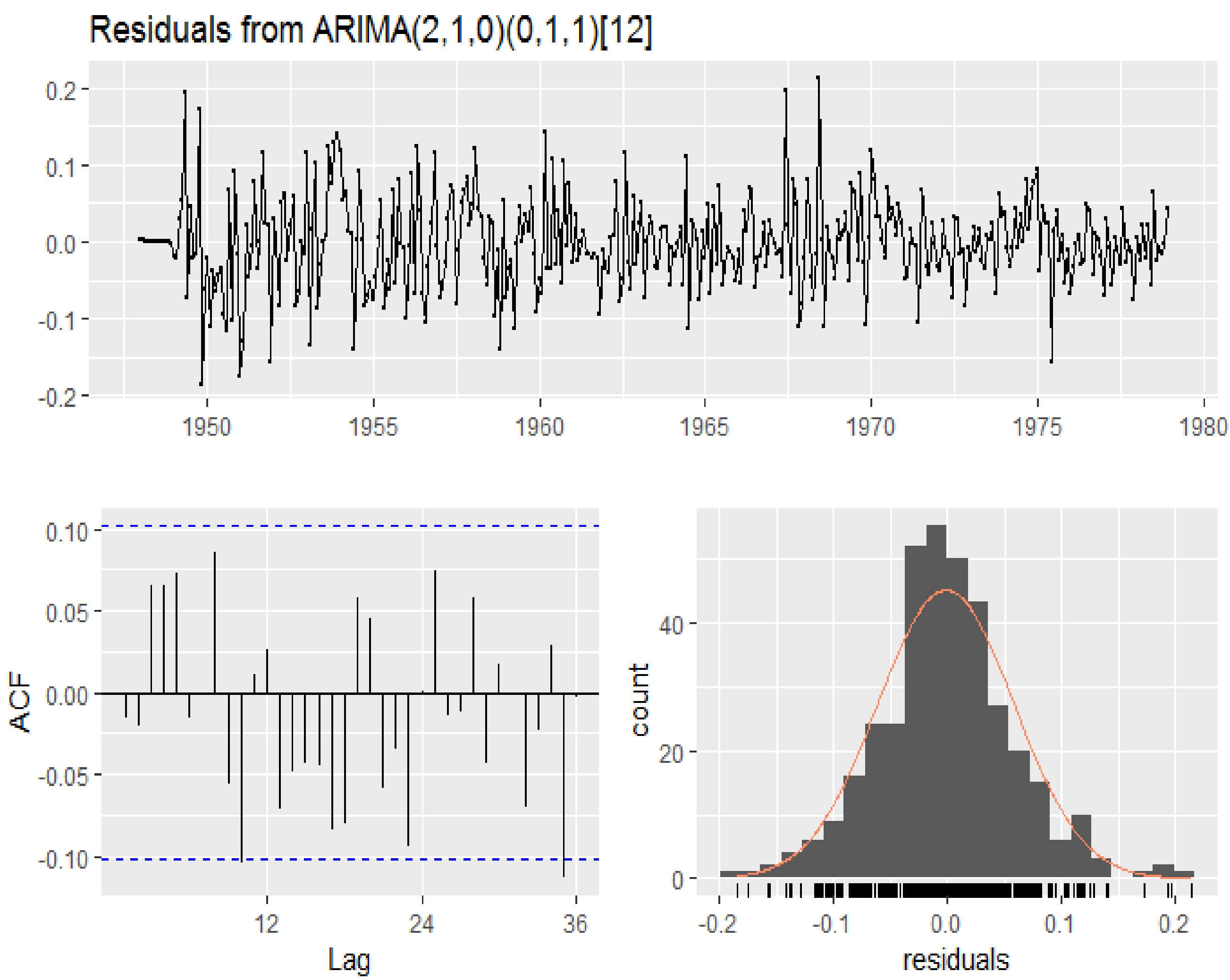


Figure 38. Resulting plots from the residual analysis.

Automatic model parameterization using R

From the previous example it is evident that using the manual model order selection process is very time consuming and mentally demanding, especially for mixed ARIMA models. In fact, in many cases the model determination will not be as simple as it was in the previous example. This was one of the problems that made ARIMA modelling using the Box-Jenkin's approach very unpopular among forecasters. But now there exists another approach that is faster and much more accurate than the manual method. Using R, this procedure is carried out automatically by using Information Criteria such as Akaike's Information Criteria (AIC) and its bias corrected version (AICc) to select the best model order. The estimation of the parameters of the ARIMA model, i.e. c, $\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$ is carried out by using the maximum likelihood estimation algorithm, just like in the ETS models.

This automated procedure is included in the forecast package in R. It handles the entire procedure of fitting an ARIMA model by calling just a single function auto.arima(). It uses a variation of the Hyndman-Khandakar algorithm for fulfilling this purpose. For a detailed explanation of the functioning of this algorithm refer to Hyndman and Khandakar (2008).

The following example as shown in Figure 39 demonstrates the use of the auto.arima() function for fitting an appropriate model to the selected time series. After fitting an appropriate model, the algorithm returns the results containing the specifications of the fitted ARIMA model. This result is shown below and contains the model order, coefficients of its parameters and the Information Criterion for that model.

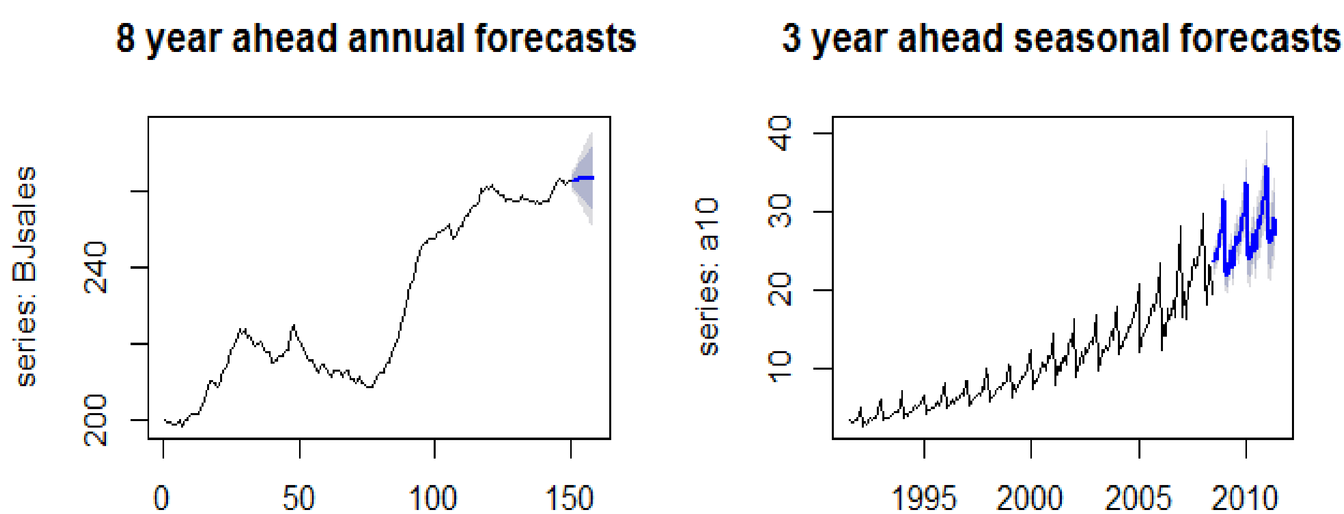


Figure 39. Forecasts generated using ARIMA modelling procedure.

Just like the `ets()` function, the `auto.arima()` function is limited to fitting the model to the data. In order to perform forecasting, this model has to be called using the `forecast()` function. The resulting forecasts are shown in the plots found in Figure 39. The `R` code and all relevant details regarding the forecast can be found in Appendix A1. The output consisting of the model parameter estimates are as shown below.

Output:

```
Series: BJsales
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.8800  -0.6415
s.e.  0.0644   0.1035

sigma^2 estimated as 1.8:  log likelihood=-254.37
AIC=514.74   AICc=514.9   BIC=523.75

Series: a10
ARIMA(1,1,1)(0,1,1)[12]

Coefficients:
          ar1      ma1      sma1
      -0.2504  -0.6674  -0.4725
s.e.   0.1007   0.0870   0.0641

sigma^2 estimated as 0.8756:  log likelihood=-258.82
AIC=525.63   AICc=525.85   BIC=538.64
```

The flowchart depicted in Figure 40 below, summarizes the steps involved in the ARIMA modelling procedure. It is important to note that the ARIMA modelling procedure used by `R`'s forecast package performs better than the old procedure developed by Box Jenkins. This updated version of the ARIMA model is considered as state-of-the-art and performs better than most heuristic exponential smoothing methods out there(Gooijer and Hyndman, 2006). It uses a stepwise search to select the best model from a class of models (Fildes and Petropoulos, 2015). This was not the case with the old ARIMA modelling procedure. Thus, it is important not to confuse this newly developed procedure with its predecessor that performs poorly. Moreover, this ARIMA based algorithm has been found to produce forecasts as accurate as, or more accurate than, those of its competitors such as ETS models (Athanasopoulos *et al.*, 2011, p. 842).

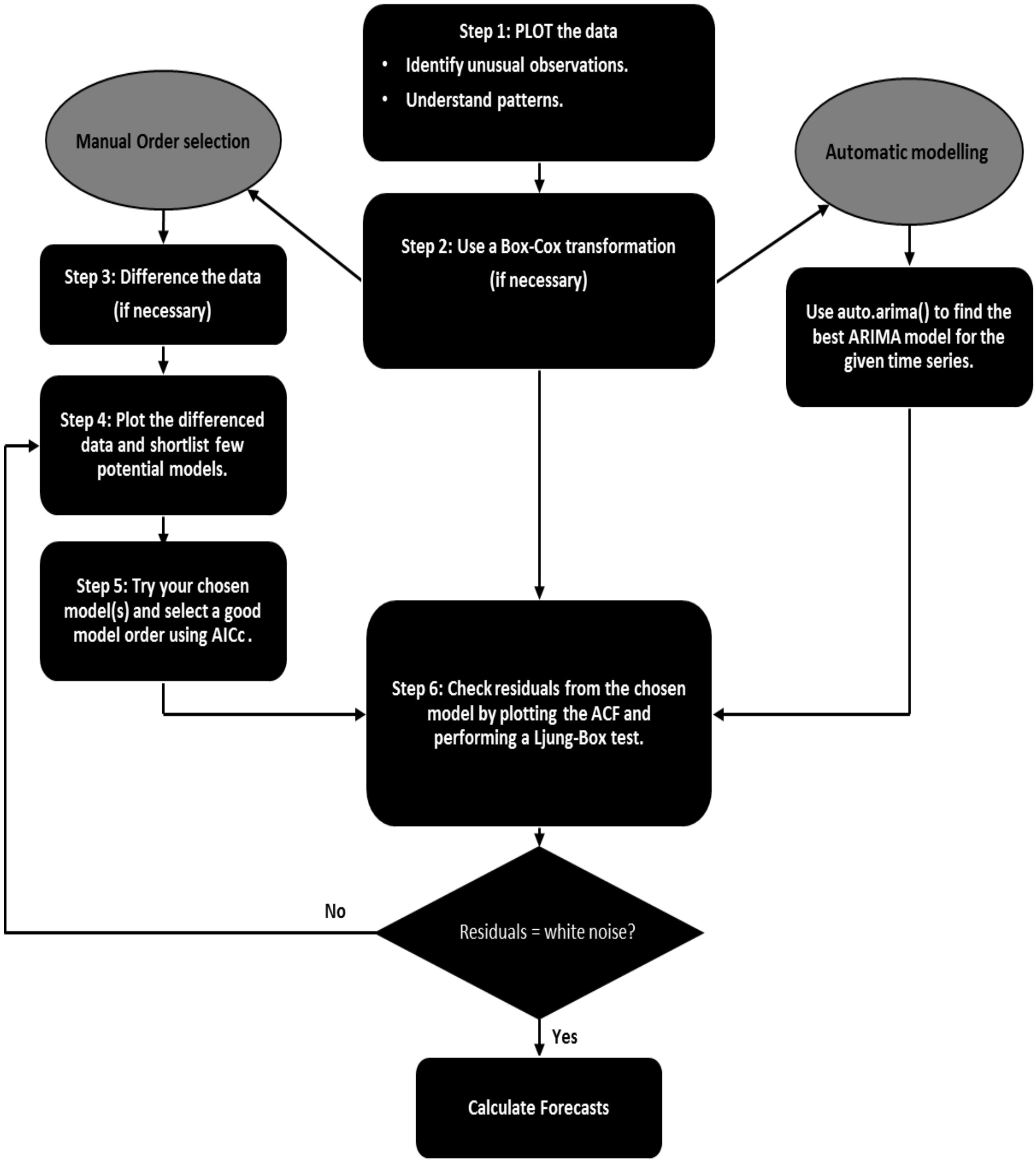


Figure 40. The general process of forecasting using ARIMA modelling in R. [30]

[30] Hyndman (2018).

The relationship between ARIMA and ETS models

Interestingly, some similarities also exist between the ETS and ARIMA models. It is believed that ARIMA models are a more general form of ETS models. But this notion is only partially true. In fact, all forms of ETS models are non - stationary and need to be differenced a few numbers of times to be compared to their stationary ARIMA counterparts. Furthermore, only the linear ETS models have an ARIMA counterpart. The non-linear ETS models do not have any ARIMA equivalents. On the other hand, ARIMA models that do not have an ETS counterpart exist too. Table 9 depicts only those models that do have a relationship with the ETS models (Hyndman,2018).

ETS models	ARIMA models
ETS (A, N, N)	ARIMA (0,1,1)
ETS (A, A, N)	ARIMA (0,2,2)
ETS (A, A_d, N)	ARIMA (1,1,2)
ETS (A, N, A)	ARIMA (0,1, m) $(0,1,0)_m$
ETS (A, A, A)	ARIMA (0,1, m+1) $(0,1,0)_m$
ETS (A, A_d, A)	ARIMA (0,1, m+1) $(0,1,0)_m$

Table 9. ETS vs ARIMA models.[31]

4.2.3 Artificial neural nets (ANNs)

Artificial neural nets are beautiful constructs that derive their biological inspiration from the behavior of the intertwined framework of neurons present in the human brain. Just like the neurons in our brain, ANNs are comprised of a large number of interconnected artificial nodes (neurons) that interact with one another to process information. This highly interconnected group of neurons work in unison to solve a problem and do it automatically by learning through the examples fed into them. From a statistical perspective, an analogy can be drawn between the neural network and a non-linear model, wherein the learning process is similar to estimating the model parameters (Balkin and Ord, 2000).

The strength of NN lies in its ability to solve many complex problems, which otherwise would not have been possible by using conventional problem-solving algorithms. In contrast to the traditional

[31] Hyndman (2018).

model building approach, the ANNs are purely data-driven. Furthermore, they also exhibit a strong self - learning capability that allows them to adapt and correct themselves autonomously without the need for human intervention. Due to these useful characteristics, ANNs lie at the heart of machine learning and artificial intelligence models. These models have gained enormous attention in solving many real-world problems during the past decade. ANNs are well suited for problems whose models are difficult to specify but possess high data availability. They learn from the input examples fed to them and can capture subtle patterns and functional relationships, which normally would not be feasible using traditional models.

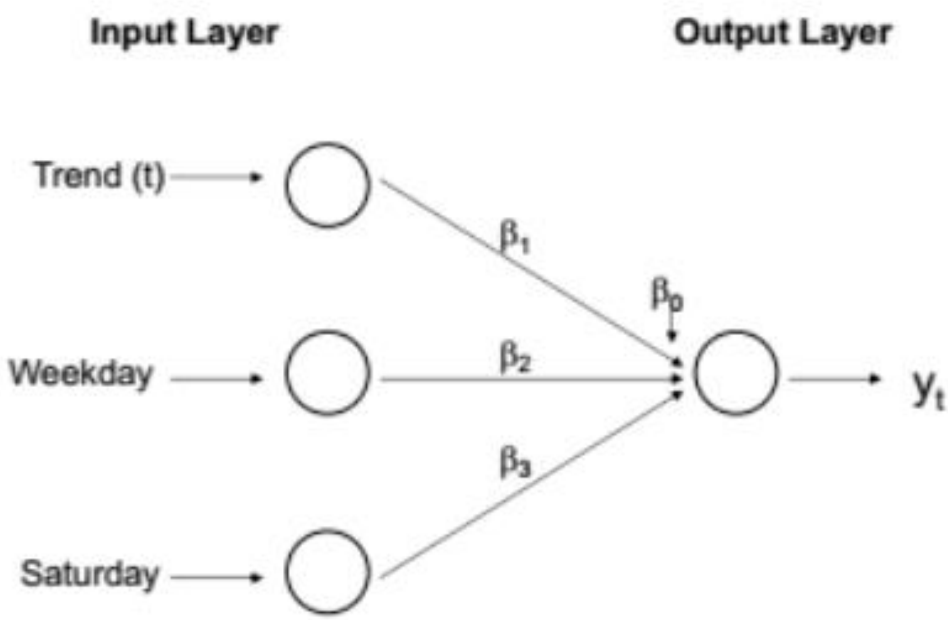


Figure 41. ANN equivalent to a linear regression model.[32]

ANNs contain a series of layers, each populated with numerous neurons, which in turn are connected to other neurons in the neighboring layers as shown in Figure 41. Each time series forecasting model is built on the assumption that there exists an underlying functional relationship between the inputs (past lagged values of the series) and the outputs (future forecast values). This relationship can be linear or non-linear depending on the process expected to be modeled. In fact, most traditional statistical modelling approaches assume that the time series generated is derived from a linear data

[32] Image taken from Shmueli and Lichtendahl (2016, pp. 192).

generating process. This is because linear models are comprehensible, simple to specify and are backed by well-defined statistical frameworks.

A popular example of a linear model is that of a linear regression, which is a suitable model for describing such scenarios. Interestingly, the simplest form of ANNs do not contain a hidden layer and behave very similar to a linear regression model, containing a certain number of predictors (input layer: Trend, Weekday, Saturday) producing a response (Output layer, y_t) as shown in Figure 41. Each predictor is has some weight (w_i) attached to it and the forecast is obtained by a linear combination of these inputs and their associated weights. In the context of time series forecasting, the predictors are simply replaced by the lagged variables as shown in Figure 41 and then the model is converted into a linear autoregressive model.

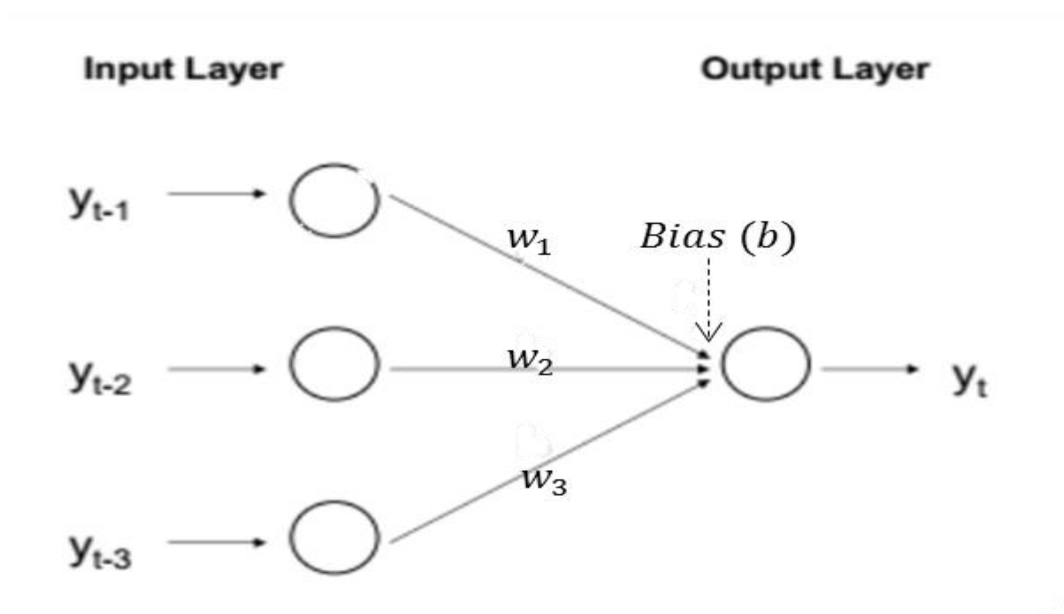


Figure 42. ANN equivalent to a linear autoregressive model.[33]

However, traditional statistical models usually have major limitations in estimating the underlying functional relationship of time series if it originates from a non-linear system. In cases such as these, it becomes extremely difficult to specify a linear model and even if it could be specified, the model might fail to capture the patterns adequately. By using ANNs, modelling of non-linear systems becomes possible. ANNs are also commonly known as 'universal functional approximators', a title

[33] Image adapted from Shmueli and Lichtendahl (2016, p 193).

given to them due to their ability to approximate any continues function very accurately. But this popular belief does not entirely hold true to time series forecasting due to their inability to accurately handle seasonality as proven by Nelson *et al.* (1999). Nevertheless, given sufficient amount of data, the ANNs shall remain the most appropriate choice for modelling complex non-linear functional relationships and capturing patterns that are hard to specify using traditional modelling approaches.

Non-linearity can be induced into the modelling framework through the addition of a hidden layer as shown in Figure 43. If a network consists of more than one hidden layer, then it is called a deep learning ANN. The ANNs that we will deal with in this thesis is the feedforward types consisting of just one hidden layer. These are more commonly known as a multi-layer perceptron (MLP).

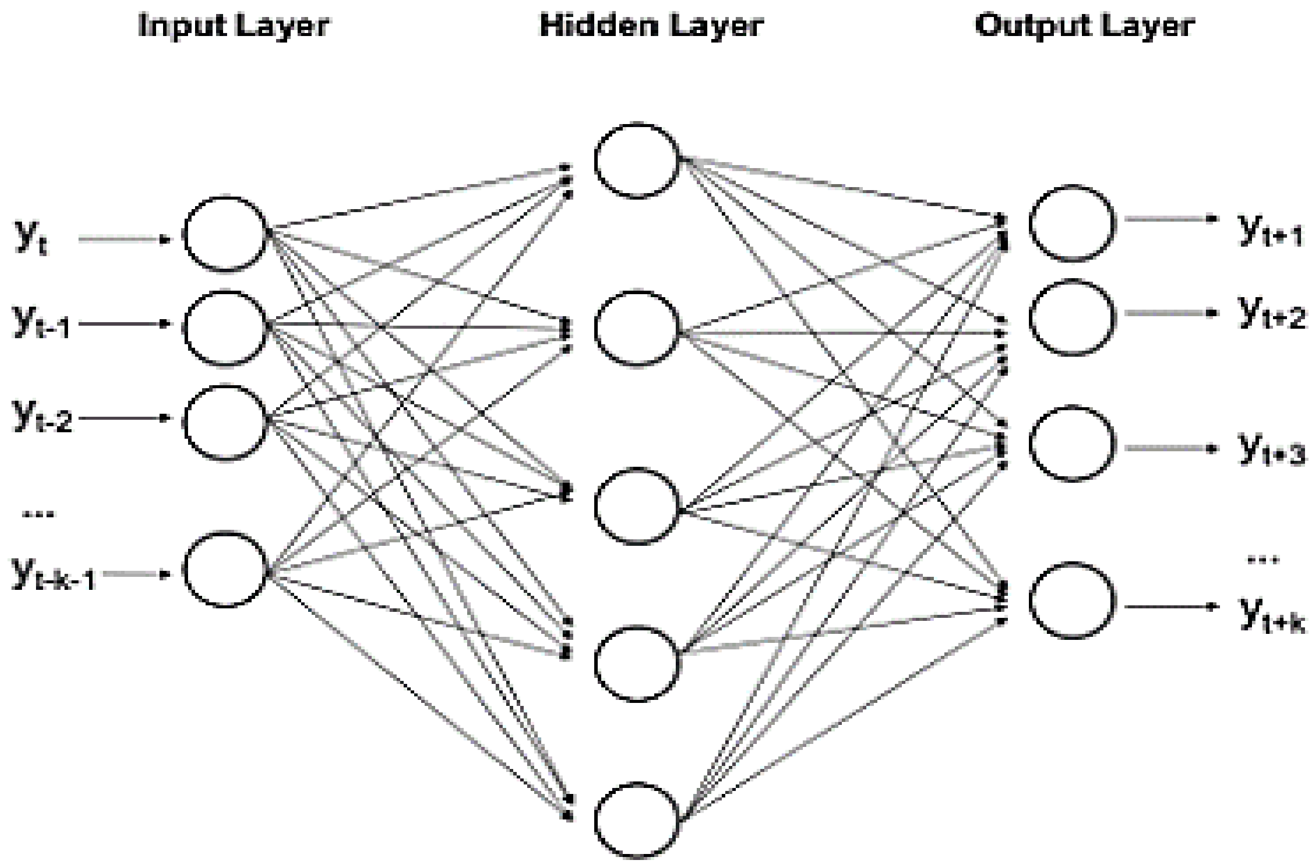


Figure 43. A non-linear ANN after hidden layer introduction.[34]

---

[34] Image taken from Shmueli and Lichtendahl (2016, pp. 193)

As their name suggests, the outputs from each neuron from each layer is ‘fed forward’ as inputs to the neurons in the next layer. Just before being received by a node in the subsequent layer, the inputs are combined using a weighted linear combination and a *bias* (b_j) is added. This bias allows a degree of flexibility while fitting a model to the data. The process is expressed mathematically by the following equation.

$$z_j = b_j + \sum_{i=1}^{4} w_{i\,j} x_i \tag{47}$$

The sum z_j then becomes the input to the hidden layer. But before being passed on to the hidden layer it is transformed by a *non-linear activation function* such as the sigmoid (logistic) function. It is mathematically represented as follows.

$$s(z) = \frac{1}{1 + e^{-z}} \tag{48}$$

The activation function is also sometimes referred to a transfer function and the type of function determines the fundamental relationship between the input and output of a neuron and the entire network. Although the sigmoid function is the most commonly used function, there are many other non-linear functions that are used in certain situations. Since the choice of the activation function depends on the problem at hand, there is no unified agreement on which function is the best. A ‘squashing effect’ is also induced by these functions wherein the outputs are restricted to a range of values. This is useful if, for example, one desires the outputs to be probabilities i.e. between 0 and 1 (Zhang, Eddy Patuwo and Y. Hu, 1998).

Like any other model, ANNs also need to be trained before they can be used for forecasting purposes. The most popular training algorithm used for training ANNs is known as the *back-propagation* algorithm. It uses a gradient descent method to minimize the cost function in a computationally feasible time. Training involves the iterative adjustment of weights and biases until the *cost function* is minimized. For time series forecasting, performance measure such as the MSE can be used as the cost function. But it is important to note that unlike the models discussed so far, the ANN requires a relatively larger dataset for training.

Although it offers a lot of benefits while solving complicated non-linear forecasting problems that linear models struggle to solve, setting up an ANN is not easy an easy task. An important aspect of modelling an ANN is the definition of its network architecture (Balkin and Ord, 2000). Many factors must be considered, and numerous settings tweaked. Moreover, the absence of any established standard for selection of these parameters makes ANN modelling a tedious task. However, many heuristic methods do exist in literature, which helps reduce the complexity involved in network design. But such heuristic shortcuts prove useful only when applied to the specific problem they were designed to solve and cannot be applied in a generic manner. Moreover, for making decisions regarding the selection of the optimum number of nodes in the hidden layer, heuristic methods currently do not exist. Therefore, in such cases, the right number of nodes can be determined only through experiments or trial and error methods.

Elements	Commonly used configuration
No. of hidden layers	> One layer sufficient for most time series forecasting problems > Sometimes two layers chosen > More than two layers does not improve performance
No. of nodes (hidden layer)	Determined through trial and error
No. of nodes (input layer)	> Training set containing input values must adequately describe pattern > Equivalent to the number of lagged observations
No. of nodes (output layer)	Equivalent to the length of forecast horizon
No. of links	All nodes in a layer connect to every node in its neighbouring layers
Activation function (hidden layer)	Sigmoid/logistical function is most common
Training algorithm	Back propagation algorithm with gradient descent is commonly used
Training and test sets division	> Determined through trial and error > Heuristic rules exists > Ensure both sets are representative of the population
Performance measure	> MSE or MASE > Use of multiple measures recommended

Table 10. Main elements of the ANN network architecture and common configuration.[35]

[35] As mentioned in Zhang, Eddy Patuwo and Y. Hu (1998).

Table 10 shown above lists the important elements constituting a typical ANN architecture. Also included in this table are the most common configuration that is used by most authors in the field of univariate time series forecasting.

As we have seen so far, a well-defined network architecture will greatly influence the forecasting performance of the ANN. Table 10 lists all the main elements that comprise the ANN. Since the forecast performance of the ANN model is a crucial criterion during model building, it would be useful to know which elements of the network architecture influence the forecast performance significantly. This can be visualized by the radial plot shown in Figure 44 below, where the effect of each element on the forecast performance is denoted by a scale ranging from 0 – lowest effect to 4 – highest effect. From this depiction, it is evident that the selection of the training-test sets and the number of hidden layers highly affect the predictive performance of an ANN.

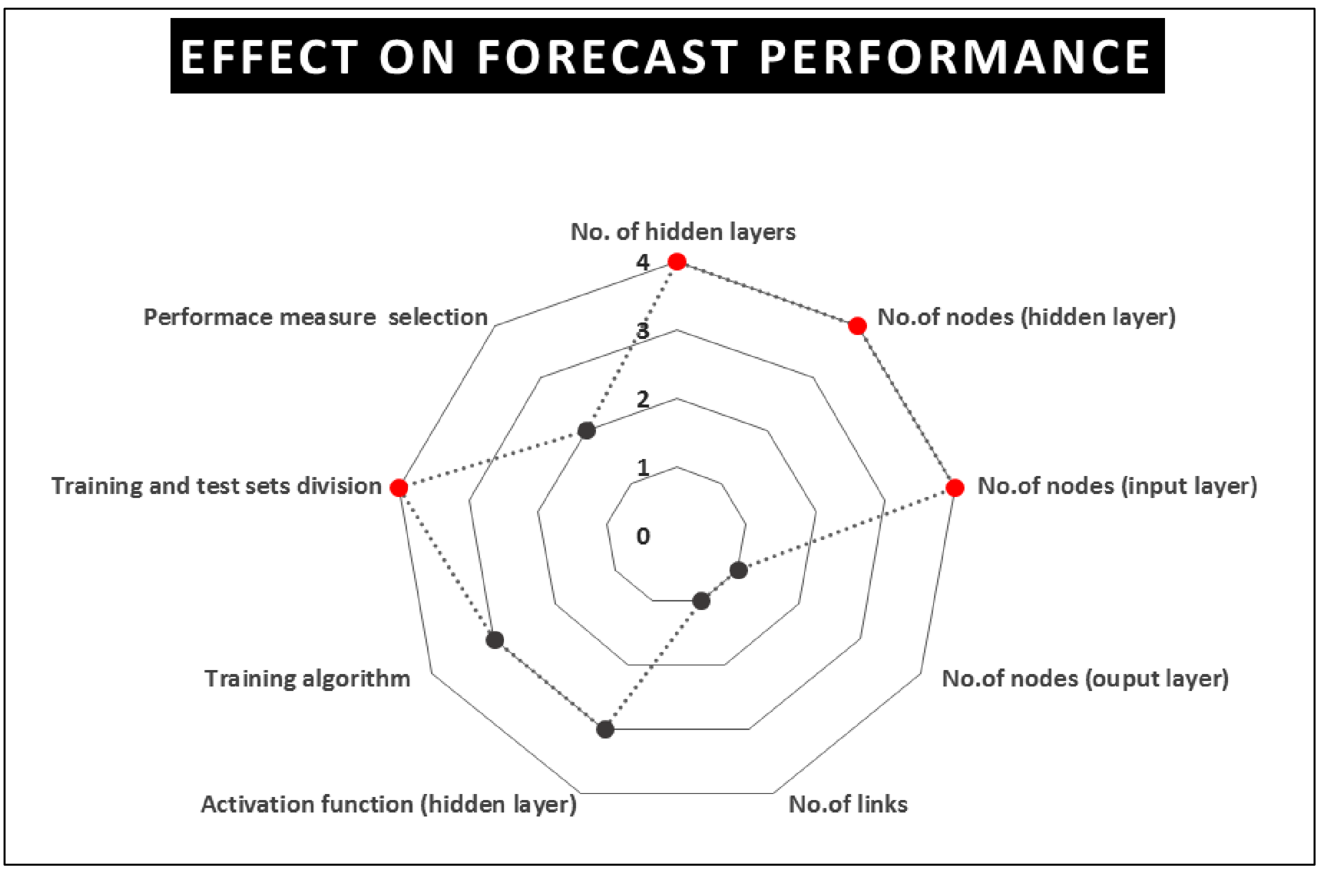


Figure 44. Elements of ANN architecture and their effect on forecast performance.

For univariate time series forecasting using the R software, the ANNs are called autoregressive Neural Nets (NNAR). It is composed of lagged values of the time series, which are used as inputs (predictors in the regression model). They can be represented by the notation NNAR (p, k), where p represents the number of lagged inputs and k represents the number of nodes in the hidden layer. So, if we want to forecast y_t using nine lagged input values ranging from $(y_{t-1}, y_{t-2}, \dots, y_{t-9})$ with 5 hidden neurons in the hidden layer, then we represent the model by NNAR (9,5). Note that an NNAR $(p, 0)$ model resembles an ARIMA $(p, 0, 0)$ model. The only difference being that the condition for stationarity need not be satisfied in case of ANN models.

For seasonal data, the lagged seasonal periods are also used in addition to the lagged time periods. In notational form it can be represented as NNAR$(p, P, k)_m$, which represents a NNAR model with $(y_{t-1}, y_{t-2}, \dots, y_{t-p})$ lagged time periods, $(y_{t-m}, y_{t-2m}, \dots, y_{t-Pm})$ seasonal lags and k neurons in the hidden layer. Similarly, for example, a NNAR $(4, 1, 2)_{12}$ would indicate a NNAR model consisting of four consecutive time lags, followed by one monthly seasonal lag and two neurons in the hidden layer. Again, the NNAR $(p, P, 0)_m$ model is equivalent to the ARIMA $(p, 0,0)(P, 0, 0)_m$ model, but without restrictions of stationarity.

NNARs can be easily applied to forecasting using the `nnetar()` function that fits an ANN model to a time series. This function is available in the `forecast` package too. For non-seasonal time series, the values of p are selected automatically by choosing the optimal number of lags found by minimizing the AIC by considering a linear AR (p) model. Whereas for seasonal time series, the default values are $P = 1$ and p is chosen by fitting an optimal linear model fitted to the seasonally adjusted data. In case k is not provided, then it is calculated according to the formula.

$$k = (p + P + 1)/2 \tag{49}$$

But unlike the stochastic models discussed earlier, ANNs are not built on a well-defined statistical framework and hence do not have any specific assumptions in place to account for errors. This makes it difficult to derive the prediction intervals to substantiate the forecasts. However, `R` allows the calculation of prediction intervals by simulating future sample paths of this model iteratively using bootstrapped residuals. For further details on bootstrapping residuals mathematically, refer to Hyndman (2018). By default, the `nnetar()` function does not produce the prediction intervals since it makes the computation somewhat slower. But if required, it can be easily generated by setting the argument `PI = TRUE` while calling the function. The respective `R` codes used to generate this plot can be found in appendix A1. The forecasts can be generated by passing the fitted model to the forecast function, just like we did when dealing with ETS and ARIMA models earlier in this chapter.

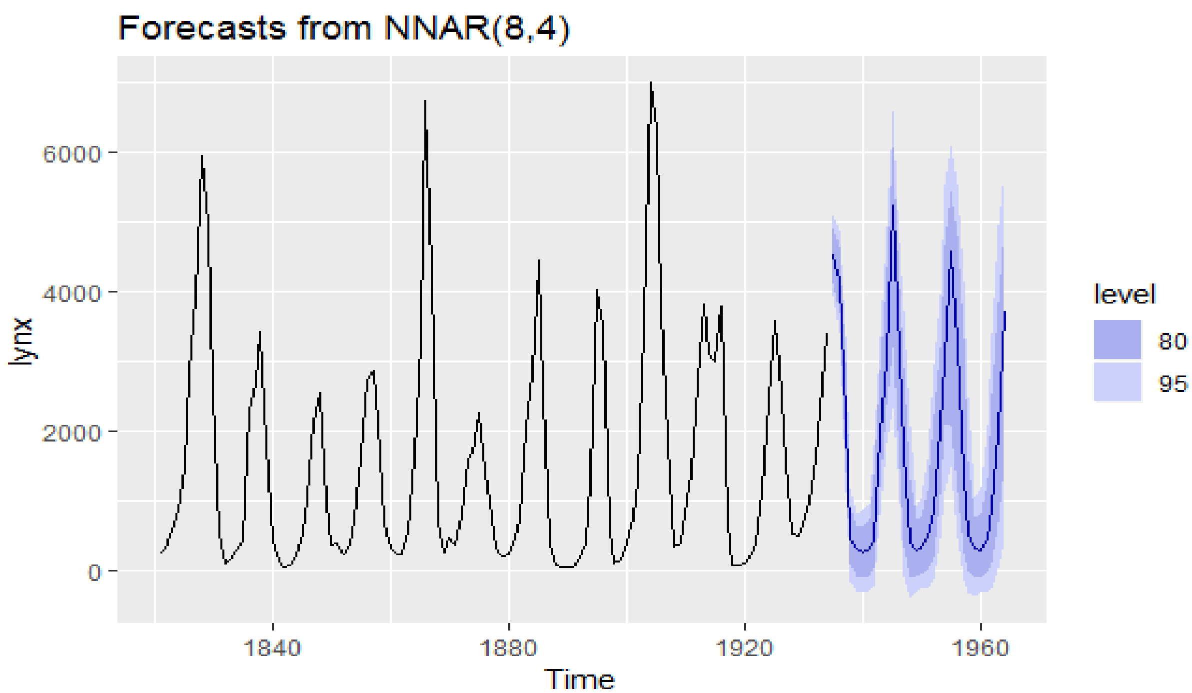


Figure 45. Forecasts generated using the neural nets.[36]

[36] Source: Dataset lynx: Annual lynx trappings taken from the fpp2 package in R.

Chapter 5. Forecast accuracy and measurements

Until now, we discussed several forecasting methods and their underlying statistical models. Moreover, we also saw how these models can be implemented to generate forecasts. But despite being able to generate forecasts, how does one decide which forecasting model to choose among the vast array of methods that exist. An all-purpose forecasting method that can generate accurate forecasts irrespective of the data does not exist. Furthermore, a highly experienced forecaster can, however, only shortlist a couple of methods depending on the inferences made from the time series analysis (trended, seasonal, etc.) and based on accumulated experience. But even then, the forecaster cannot discern the accurate method out of the bunch unless there is a system of measurement in place. This means that the output of such a measurement system should indicate the best forecasting method, or in other words, allow the forecaster to identify the method with the highest accuracy. Moreover, since there isn't a forecasting method that can be 100% accurate, the underlying uncertainty must also be presented by the measurement system. Thus, a good forecast accuracy measurement system should not only help in determining the most accurate forecasting method, but also indicate the uncertainty attached with the forecasts this method generates. Only then can the generated forecasts be utilized feasibly. This section will discuss the different aspects involved in the measurement of forecast accuracy. It begins by discussing the various indicators used in accuracy measurement. This will be followed by a discussion pertaining to forecast accuracy validation methods.

5.1 Forecast accuracy measurement indicators

Summary statistics

Every measurement system needs to have a common evaluation metric to be even called a measurement system in the first place. For example, the mean serves as a metric for measuring the central tendency and the standard deviation is the preferred metric to measure the spread across various data distributions. In a similar fashion, it only makes sense to measure forecast accuracy if there exists a suitable metric, which would allow the comparison of forecasting performance across various forecasting methods. These metrics are generally referred to as *summary statistics* in forecasting jargon. This section discusses few such metrics that are commonly used for the measurement of forecast accuracy. Each metric has its own set of pros and cons, making it suitable for some situations, but at the same time unsuitable for others. Let us begin by introducing these useful forecasting metrics. The equations used in the following section and their accompanying descriptions are as proposed by Makridakis, Wheelwright and Hyndman (1998, pp.42-54).

As we know by now, forecast error is the difference between the forecast and the actual observation. It can be represented by the following equation.

$$e_t = Y_t - F_t \tag{50}$$

Where $\boldsymbol{e_t}$ is the forecast error.

Here, e_t indicates a one-step forecast error since it is forecasting one period ahead of the last observation Y_t . The forecast F_t is the forecast calculated using data $Y_1, \dots, Y_{t-1}$. Now if observations and forecasts are available for n time periods (steps), then the number of error terms will also be equal to n. Taking this into consideration, now we can define the following standard statistical measures of accuracy. The simplest indicator is the one that can be developed by averaging all the errors for the n periods. The resulting average is referred to as the *Mean Error(ME)*. It can be represented mathematically as follows.

$$\text{ME} = \frac{1}{n}\sum_{t=1}^{n} e_t \tag{51}$$

But forecast errors can either be positive or negative depending on whether the forecasts overshoot or undercut the actual value. This may cause the resulting ME to be very small. This would be helpful if the forecaster just wants to know whether the forecasts have under-forecasted or over-forecasted. But this indicator provides no information regarding the general size of the error. To overcome this problem, we take the absolute value of the errors and then take the average. This indicator is referred to as the *Mean Absolute Error(MAE)*. It can be mathematically represented as follows.

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} |e_t| \tag{52}$$

Although MAE serves the purpose theoretically, in practice another indicator is used that is very similar to the MAE. This indicator is known as the *Mean Squared Error(MSE)*, where the errors are squared rather than taking the absolute value. This makes this indicator much easier to handle mathematically making its usage ubiquitous in statistical optimization operations. It can be represented mathematically as follows.

$$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2 \tag{53}$$

However, the magnitude of the resulting values that the aforementioned indicators take depend largely on the scale of the data used. This makes them unsuitable for comparing forecast accuracy for time series with different scales or different time intervals. For example, an error of five units obtained while forecasting monthly magazine sales is dissimilar to an error of five units obtained while forecasting annual magazine sales. To facilitate such comparisons, the relative or percentage can be used. For this purpose, we should first compute the *percentage error(PE)* as shown below.

$$\text{PE}_t = \left(\frac{Y_t - F_t}{Y_t}\right) \times 100 \tag{54}$$

Taking the mean across all n time periods gives the *Mean Percentage Error (MPE)*. This can be used to compare the accuracy of methods across different time series irrespective of the scale. It can be mathematically represented by the following equation.

$$\text{MPE} = \frac{1}{n}\sum_{t=1}^{n} \text{PE}_t \tag{55}$$

However, just like the ME, the MPE may produce small values since positive and negative PEs tend to offset each other. Thus, we take its absolute value of the PEs that gives us the *Mean Absolute Percentage Error (MAPE)*. It can be represented by the following equation.

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n} |\text{PE}_t| \tag{56}$$

Using the MAPE, the forecaster can easily compare the percentage errors across series with different scales. In such cases, a MAPE of 8% would be considered more informative than an MSE of 170. But the main disadvantage of the MAPE is that it remains valid only if the scale on which the errors are compared has a meaningful origin. For example, temperature scales such as the Celsius or Fahrenheit have different origins (zero points) relative to each other. Using the MAPE for such situations results in invalid calculations. Moreover, percentage errors tend to output infinite or undefined values if zeros are present in the time series values being measured.

Benchmarking

Another commonly used approach for measuring the forecast accuracy is to set up a *benchmark*. Usually, a simple forecasting method such as the naïve method is used as a yardstick for comparing the performance of more sophisticated methods. The underlying argument for doing so is that if a forecasting method cannot beat a simple forecasting method, then a more complex method is not worth considering for that forecasting scenario. Thus, the naïve method sets a benchmark for forecast accuracy and surprisingly, it is very hard to beat in some cases (Makridakis *et al.*, 1993).

Other simple methods such as seasonal naïve, drift and the mean method can also be used for benchmarking purposes; their selection depends on the time series characteristics shown by the dataset. For seasonal data, usually, a seasonal naïve benchmark is preferred over the naïve. We have already discussed these simple methods and demonstrated their application using `R` in Chapter 3. We will use the seasonal naïve method as our preferred benchmarking method during the demonstration conducted in the next chapter. This benchmark can be used for both seasonal and non-seasonal data.

Theil's U-Statistic

The MSE has another feature which is worth discussing. As we know, the errors get squared while applying this statistic. But this squaring not only helps to maintain positive values but also exaggerates large errors by giving it more weight, resulting in a large overall MSE. This helps penalize forecasts that contain large errors from those that have relatively small errors. This is a useful property for a forecast accuracy measurement indicator. If this feature can somehow be combined with the advantage of the relative measures (i.e. scale independence) while allowing its comparison to a benchmarking method, then such a measurement indicator can serve as a multi-purpose indicator. These are exactly the characteristics that describe the Theil's U-statistic indicator. This is a relative error metric, which allows the comparison with a naïve method. While doing so, it also penalizes large errors by squaring them. It can be mathematically represented by the following equation.

$$U = \sqrt{\frac{\sum_{t=1}^{n-1}(FPE_{t+1} - APE_{t+1})^2}{\sum_{t=1}^{n-1}(APE_{t+1})^2}} \tag{57}$$

Where $FPE_{t+1} = \frac{F_{t+1} - Y_t}{Y_t}$ is called the *forecast relative change*

and $APE_{t+1} = \frac{Y_{t+1} - Y_t}{Y_t}$ is called the *actual relative change*

When compared to the statistical indicators discussed so far, the Theil's U-statistic is slightly more complicated from a mathematical perspective, making it slightly less intuitive. To make things simpler, it would suffice to understand that the numerator is equivalent to the MAPE of the forecasting method under evaluation. Whereas the denominator is similar to the MAPE of that of a naïve method. Taking the squares allows for the penalization of large errors. In this way, the Theil's U-Statistic combines the positives of the MAPE and the naïve benchmark. For more details pertaining to the underlying mathematical structure of this indicator, refer to Makridakis, Wheelwright and Hyndman (1998, pp. 49-50) and Theil (1975). The interpretation of the values that this indicator outputs can be understood using the following rule.

- If $\mathrm{U} = 1$: The accuracy of the forecasting method under evaluation is similar to the naïve method.
- If $\mathrm{U} < 1$: The accuracy of the forecasting method under evaluation is better than the naïve method. A smaller U statistic is an indication that the method performs relatively better than the naïve method in general.
- If $\mathrm{U} > 1$: The accuracy of the forecasting method under evaluation is worse than the naïve method. The new method must be discarded since the naïve method will produce better results.

However, when the errors are very small or equal to zero as seen while using an intermittent demand series used in Kim and Kim (2016), then a relative error metric would no longer work. To tackle this issue, another similar method was proposed by Hyndman and Koehler (2006). This is more commonly referred to as the *Mean Absolute Scaled Error(MASE).* This method is also a scale-free error but uses a different approach to become scale-independent. Unlike the relative indicators that use 'division' by a naïve method to get rid of the scale, the MASE becomes scale independent by 'scaling the forecast error' based on the in-sample mean absolute error using the naïve method. This also tackles the issue involving infinite or undefined values. We first define a scaled error using the following formula.

$$q_t = \frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|} \tag{58}$$

If the forecast performs better than the average one-step naïve forecast, then the scaled error $q_t < 1$. On the other hand, if the forecast performs worse than the naïve forecast then $q_t > 1$.

Once the scaled error is computed then the MASE can simply be calculated as shown in (eq.61) below.

$$MASE = mean\ |q_t|) \tag{59}$$

5.2 Validating forecast accuracy

Using the summary statistics mentioned in the previous section only indicates how well a model has performed for that particular sample dataset. The statistics fail to give us any information regarding how accurate the forecasts are really going to be in the future. True forecast errors can be found only when the actual observations are available from the future. But if the forecaster keeps waiting for future observations, then it would not be feasible for business and the clients might find somebody else to solve their forecasting problems. To save time, the most sensible thing to do is to make use of the data already available in hand to validate the forecasting method. This section will discuss two commonly used procedures using which the forecaster can measure the accuracy and validate the selected methods.

Data Partitioning

Data partitioning basically involves splitting the time series into *training* and *test* sets. The corresponding partitions are pictorially depicted in Figure 46 using a sample dataset. As its name suggests, the training set is mainly used to fit (train) the chosen forecasting model to the data belonging to that training set. In contrast, the test set (holdout/validation set) is generally used for validating the model or for determining its out-of-sample forecast accuracy. This is done by comparing the forecasts generated from the training set against the observations belonging to the test set. But the main purpose of data partitioning is to generate out-of-sample forecast errors without needing to wait for the real observations to occur in the future. In other words, this validation procedure assumes that the training data are the only data available at the time of forecasting. The data in the test set is ignored or 'hidden', so that its values don't influence the forecasts and model fit. Intentionally hiding the test data makes its absence behave like the real observations that are yet to be observed in the future. This process can also be imagined as pushing the forecast origin back to a certain period.

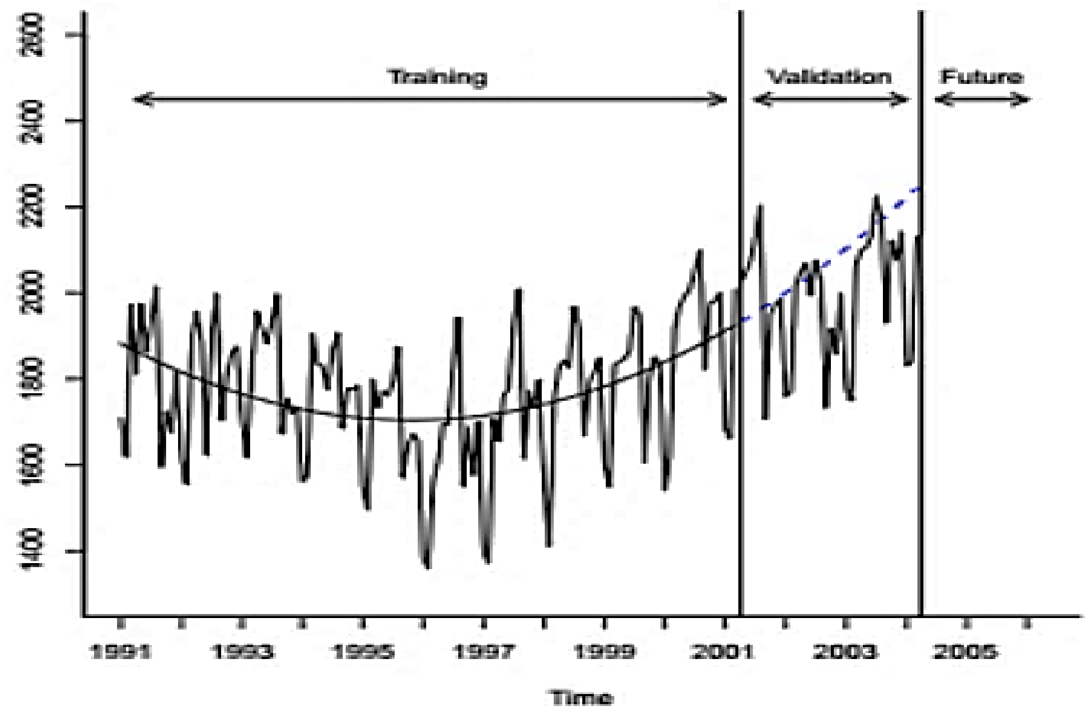


Figure 46. Data partitioning into training and test (validation) sets.[37]

Once the model is trained on the training data, then forecasts should be generated for a forecast horizon that is equal to the length of the test set. Only after these forecasts are produced, the test set should be revealed and analyzed for computing the forecast errors. It is important to not confuse forecast errors with residuals. Residuals are in-sample (training set) errors obtained by taking the difference between the training set data and the fitted values generated by fitting the model in the same set. They help assesses the goodness of fit of a forecasting model. They can quickly be examined by using the `checkresiduals()` function in `R` as demonstrated previously when we dealt with fitting ARIMA models. But since the modelling functions of the `forecast` package use automatic model selection and parametric estimation procedures for fitting the best model to the data, the check for residuals is not very crucial. Although residuals provide crucial information regarding how well the model replicates the systematic patterns in the data, they do not provide enough information regarding the accuracy of future forecasts. To achieve this, we must calculate forecast errors.

[37] Shmueli and Lichtendahl (2016, p.49).

Forecast errors can be further subdivided based on the type of forecasts that generate them. Two commonly used terms to assess forecast performance are ex-post (meaning “Before the event” in Latin) and ex-ante (meaning “After the fact” in Latin) forecasts. In the context of univariate time series forecasting, an ex-post forecast refers to the forecasts that were generated from the model estimated using the training set; where the forecast model is evaluated using a holdout set (test set) created using the data already available in hand. Terms such as ‘out-of-sample’ or ‘out-of-estimation sample’ forecasts are generally referring to ex-post forecasts. Note that the ‘sample’ here relates to the training set. In comparison, an ex-ante forecast refers to the forecast that is generated using the entire dataset (without partitioning), where future observations are unknown at the time of forecasting (Chan, 2010, pp. 71-72; Yaffa, 2010, pp. 4-6). To put things into perspective, the main goal of the data-partitioning procedure should be to minimize the ex-post forecast errors. By maintaining a minimum ex-post forecast error, we can expect the ex-ante forecast errors to be small as well, which in turn indicates high forecast accuracy (Ledolter, 2006, p.189).

There is yet another important reason why the use of out-of-sample accuracy measurements is essential. Residuals can be kept to a minimum level simply by fitting the data using a complicated model containing many parameters, thereby giving a false implication of a good model. Despite fitting the data well, usually, such a model will produce terrible forecasts. This phenomenon is generally referred to as *overfitting*. It generally occurs when the sophisticated model captures the unnecessary components of the noise in addition to the normal systemic components. If left unchecked, this might pose a serious problem for forecast accuracy due to the bias it induces into the forecasts (Shmueli and Lichtendahl, 2016, p.45). But this problem can be prevented by performing an out-of-sample accuracy evaluation using data-partitioning. Here, the test set does not influence the model fitting process, thereby resulting in unbiased forecast errors.

<u>Time series cross-validation</u>

The splitting of the data into training and tests sets is a traditional way of evaluating the performance of a forecasting method. Although it works well for large test sets, a problem arises when the test set tends to be small. Since the accuracy of the forecasts depends purely on the out-of-sample forecast errors that are essentially derived from the test set, having a small test set might mislead the forecaster to draw conclusions that work well only for that test set but may not be reliable for future times. A smaller test set is also more susceptible to outliers which can give rise to large forecast errors during the out-of-sample test, inadvertently accusing a good forecasting model as a faulty model. Moreover, forecast accuracy using the traditional data partitioning method only generates a single set of forecast errors per forecast horizon, which does not characterize a robust forecast accuracy measurement system (Tashman, 2000).

As an alternative, we can use *time series cross-validation* as the forecast accuracy validation system. Unlike the fixed origin used in data partitioning, this method consists of a moving origin that constantly moves forward with time, simultaneously generating a series of training and test sets. Instead of just splitting the data into a single pair of training and test sets by maintaining a fixed origin, a rolling window allows the generation of multiple training-test set pair, allowing the addition of a new observation to the training set every time the origin rolls forward. This leads to a robust forecast accuracy measurement since a set of forecast errors are now generated from multiple test sets, compared to the single pair of training and test sets generated while using the conventional data partitioning procedure.

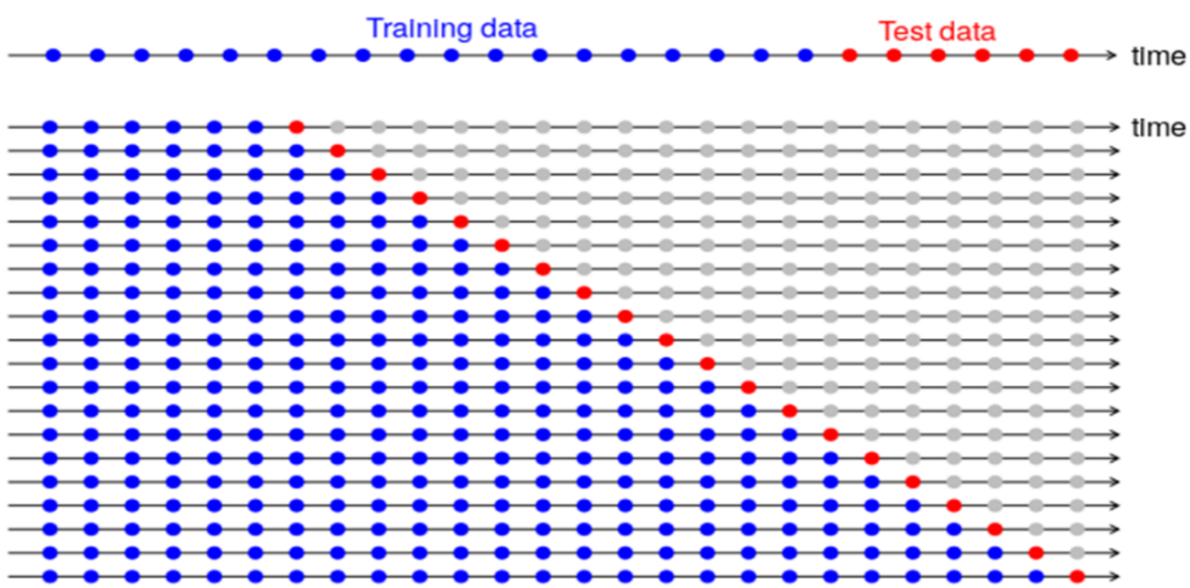


Figure 46. Time series cross-validation (single step point forecasts).[38]

The setup would look like the one shown in Figure 46 above, where one-step forecasts are considered. The blue dots represent the training set and the red dots represent the test set. The white dots are

[38] Adapted from Hyndman (2018).

ignored for that particular run. Notice that each training set consists of one more observation than the previous training set. Doing this allows many more observations to be used in the test set during each run. Now we can evaluate the forecasting accuracy of a model by averaging the errors over all those tiny test sets. Note that there may be some missing values at the start of the series since it is simply not possible to compute the forecasts when the training set is too small (Hyndman, 2018).

Time series cross-validation can be easily performed in `R` using the `tsCV()` function. A better understanding of the application of this method can be made later, while demonstrating the measurement of forecast accuracy in the next chapter. For multi-step forecasts, the argument h representing the forecast horizon must be set accordingly while calling the function as shown in Figure 47 below. As an output of this procedure, an error matrix consisting of all the computed errors corresponding to each time step is generated by `R`.

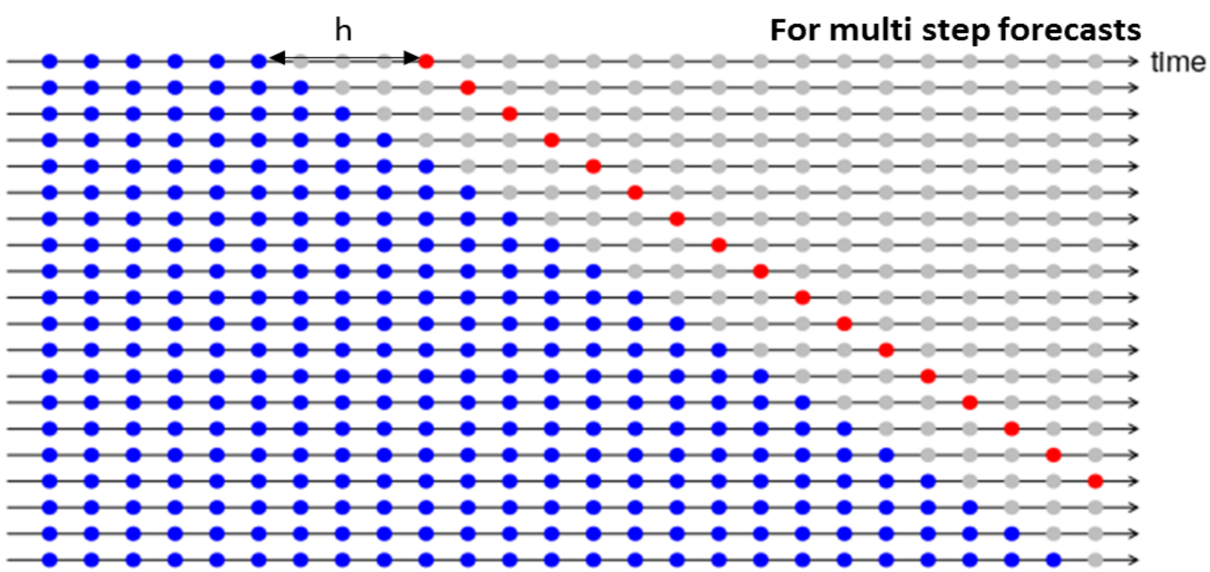


Figure 47. Time series cross-validation setup (for multi-step forecasts).[39]

<u>Prediction Intervals</u>

Prediction intervals are commonly used in statistics, particularly in regression analysis. They serve as a good system for the representation of uncertainty/variability, which is intrinsic to stochastic measurements. From a forecasting perspective, enhancing the forecasts with prediction intervals make

[39] Adapted from Hyndman (2018).

the forecasts more reliable. It does this by attaching probabilities to a certain range of expected values. For example, with the assumption that the errors are normally distributed, if a probability of 0.95 is given to a certain range containing a set of forecast values, it means the probability of the actual observation falling within that range is 95%. In other words, the forecaster can confidently promise that the true value would exist within this prediction interval, 95 out of 100 times.

Prediction intervals can also be imagined as a range containing many future values as shown by the multi-colored lines in Figure 48.

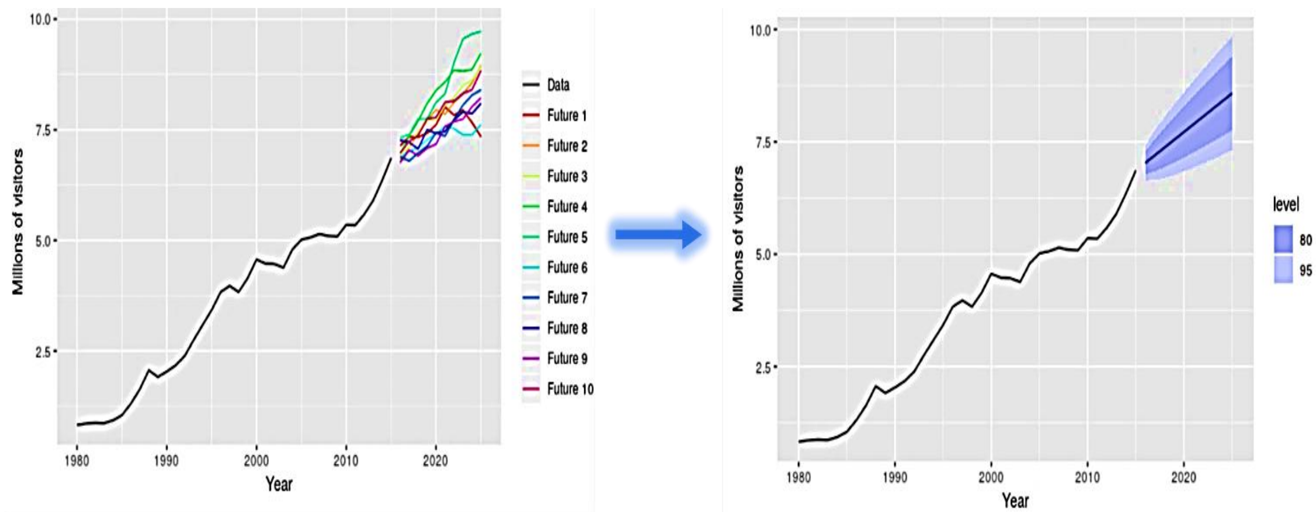


Figure 48. Prediction intervals in R plots.[40]

R also calculates the prediction intervals in a similar way, but instead of representing each individual future as a separate curve, it plots this forecast distribution as a blue shaded strip. The thick dark blue line passing through the center of this strip indicates the mean of all the future values. This is what makes up the forecast. Prediction intervals are automatically calculated based on the standard deviation obtained from the residuals after fitting the model to the data.

For naïve forecasts, the standard deviation for the fitted values and forecast errors remain the same. But for models with many parameters (unlike the naive), the forecast errors' standard deviation is slightly larger compared to the residual standard deviation. This can be observed from their respective prediction intervals too. Also notice that the blue strip is sub-divided into a dark blue and a light blue

[40] Taken from Hyndman (2018).

strip, representing the 80% and 95% prediction interval levels respectively. For an h-step forecast distribution, the prediction intervals can be mathematically represented as follows. Note that the errors are assumed to be normally distributed.

$$\hat{y}_{T+h|T} \pm 1.28\hat{\sigma}_h \text{ for 80\% prediction intervals} \tag{60}$$

$$\hat{y}_{T+h|T} \pm 1.96\hat{\sigma}_h \text{ for 95\% prediction intervals} \tag{61}$$

Representation of the forecasts as a range of values instead of just a single value (point forecasts) helps distribute the uncertainty over many forecast values rather than just one forecast value. So, prediction intervals provide a more realistic representation of the risks involved and help overcome the limitations of the point forecast (Song and Hyndman, 2011). As discussed earlier, the accuracy of forecasts decreases as the time horizon increases. Particularly at longer time horizons, the forecasts are affected by more uncertainty than that existing in the short-term horizons. Due to this reason, it is common for prediction intervals to increase in size for multi-step forecasts that span a longer horizon. For more details regarding how the prediction intervals are calculated and plotted in R, refer to Hyndman (2018).

Chapter 6. Forecasting using the R software

In this chapter, the methodology of forecasting will be demonstrated by using an example dataset. This methodology follows the one shown in Figure 2. The forecasting methods discussed earlier and the statistical modelling approaches that govern these methods will be put to test. Since the main goal of forecasting is to obtain accurate and reliable forecasts, the accuracy of the forecasts generated by the selected models will be tested according to the statistical measures and the validation procedures discussed in Chapter 5. We start by describing the setup and assumptions made for this demonstration. Once the setup is specified, the subsequent sections break down the actual forecasting methodology into further detail by providing step by step explanation about performing forecasting using R. For better understanding, code snippets are provided during each step. The final section of this chapter deals with the evaluation of results obtained from this demonstration. Moreover, established forecasting paradigms based on findings from forecasting competitions conducted in the past will be discussed then.

Setup specifications / Assumptions for demonstration

- In our forecasting demonstration, we will use a real-world dataset available in the time series data library[41] created by Rob J. Hyndman. Particularly, the dataset chosen for this demonstration is titled as Industrial Production, Spain: Monthly[42] containing monthly data.
- The work environment and graphical user interface (GUI) used is R studio.
- The base R package is loaded by default into the application when the GUI is started. In addition to that, the following R packages must also be loaded - forecast, ggplot2, readxl.
- The R version number used in this thesis is 3.5.1; (2018-07-02) - “Feather Spray”.
- All models and plots used in this demonstration are available in the forecast package. No additional packages need to be installed.
- The accuracy of the models will be tested by using the data partitioning and time series cross-validation procedures separately (see steps 8.1 and 8.2 respectively). Note that since this is

[41] Hyndman, R.J. (2018) - *Time Series Data Library*

[42] Original Source: Brockwell and Davis (1991, Series E, p. 556)

just a demonstration, importance is mainly given to describing the 'process' of validation (data partitioning and time series cross-validation) rather than proving their appropriateness. i.e. Determining which validation procedure is better – data partitioning vs. time series cross-validation – is not the main goal of this demonstration.

- The summary statistics used are the MSE, MASE, MASE, RMSE, Theil's U coefficient, ACF plot. These measures are automatically computed when the accuracy() function is called.

6.1 Demonstration of the forecasting methodology

Step 1: Define the goal

Let us assume that we have previous n periods of monthly production data and now we wish to forecast the demand for the next 12 months (1 year). Once the goal is established, then this is followed by setting up the program and loading the required packages that contain the functions pertaining to time series analysis and forecasting. In our case, we need the forecast package for forecasting, the readxl package for loading datasets into R and the ggplot2 package for visualizations. Packages can be loaded into R using the library() function as shown below.

```
library("forecast", lib.loc="~/R/R-3.5.1/library")
library("ggplot2", lib.loc="~/R/R-3.5.1/library")
library("readxl", lib.loc="~/R/R-3.5.1/library")
```

Step 2: Get data

Normally, the data must be pre-processed and cleaned before it can be used for modelling purposes. Any lurking missing values should be detected and the reason for their presence must be identified. But for our demonstration purposes, we shall assume that the data has already undergone pre-processing, and all missing values are filled-in using interpolation.

Now we need to load the dataset into R. Since we are loading an excel worksheet, the read.excel() function from the readr package must be used. Other types of formats can also be loaded into R using other functions and packages. After loading the data set, it needs to be converted into ts (time series) object. Time series related functions can be applied only after converting the data into ts objects. The R code used to perform this step is as shown below.

Import excel data

```
industrial_production_spain_montOK <- read_excel("D:/MASTER THESIS/R files/SIM - data and results/Tested/Monthly/with horizon tests/industrial-production-spain-montOK.xlsx")

View(industrial_production_spain_montOK)
```

```
# Convert to ts class

indprod_spain <- industrial_production_spain_montOK[, 2]

str(indprod_spain)

indprod_spain.ts <- ts(indprod_spain, start = c(1977, 12), end = c(1982, 12), frequency = 12)
```

Step 3: Explore the data

After the data is loaded and available for use in its ts form, only then data visualization can be performed. This step uncovers the basic structure and time series components underlying the data. Obvious patterns or relationships can be broadly identified. The presence of outliers can be easily be detected during this step. Forecasting relevant time series components can be readily separated using STL decomposition by executing the following R code.

```
#Plot the data

autoplot(indprod_spain.ts)
```

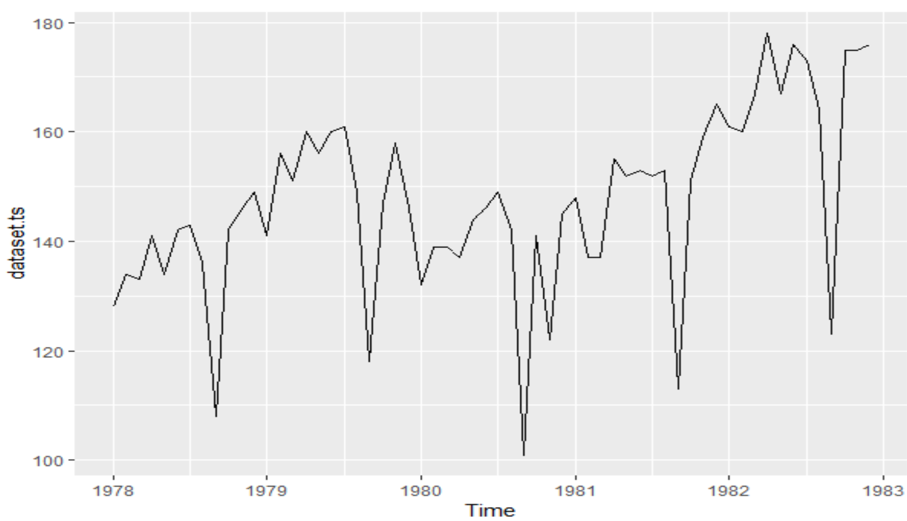


Figure 49. Plot of monthly industrial production in Spain.[43]

[43]Source: Time series data library created by Rob. J. Hyndmann; Available at Datamarket.com.

```
# Cross-checking length of periods; To ensure the correct number of periods have been specified.
length(indprod_spain.ts)
length(indprod_spain.ts)/12 # Division by 12 since it is monthly data

# STL Decomposition
autoplot(mstl(indprod_spain.ts))
```

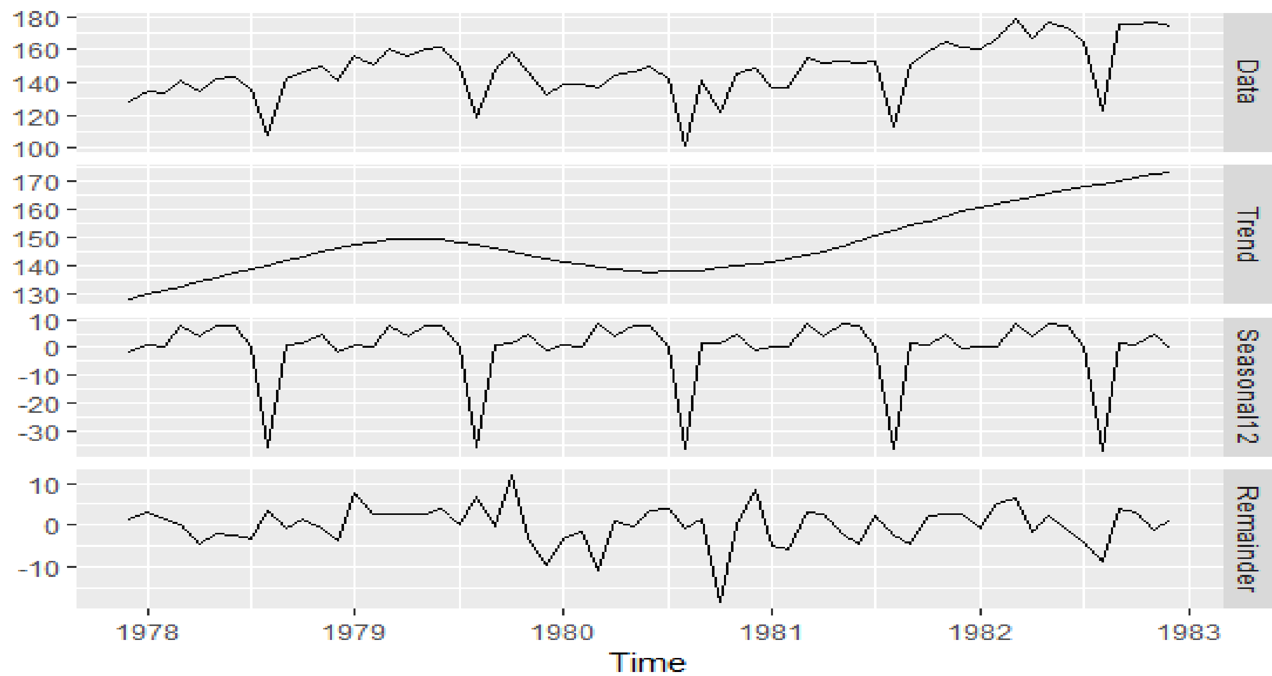


Figure 50. Splitting into time series components using decomposition.

<u>Step 4: Select models</u>

After analyzing the time series from the decomposed time plots shown in Figure 50, any distinct patterns belonging to the various time series components now become evident. Based on the method selection criterion discussed in Chapter 2, the ETS model, ARIMA and the ANN models are shortlisted as the preferred models for forecasting this kind of data generating process. For benchmarking purposes, the seasonal naïve method is chosen since the data appears to be exhibiting seasonality.

<u>Step 5: Data partitioning</u>

If data partitioning is the preferred validation procedure, then the data needs to be manually split into training and test sets. Note that the length of the test set should be maintained equivalent to that of the forecast horizon. In this case, it is set to 12 months since that is how far we intend to forecast. The window() function can be utilized to partition data at the respective origin as shown below.

```
# Partition data into test and training sets; Necessary for data partitioning.
indprod_spain_test <- window(indprod_spain.ts, start = c(1982,1))
indprod_spain_train <- window(indprod_spain.ts, end = c(1981, 12))
```

Step 6: Fit the models

After partitioning the data, the next step is to fit the selected models to the training set. This can be achieved by calling the model-specific functions provided by the forecast package. To ease computation and reduce redundancy of code, these models are grouped together by calling the list() function and naming this list as models as shown below.

```
# Fit the models to the training set.
models <- list(
  mod_ets = ets(indprod_spain_train, ic='aicc', restrict = FALSE),
  mod_arima = auto.arima(indprod_spain_train, stepwise = TRUE),
  mod_neural = nnetar(indprod_spain_train)
)
```

Step 7: Generate preliminary forecasts

Once the models are fitted (trained) on the training set, we can then use them to produce forecasts. Forecasts are generated by calling the forecast() function. This function is applied holistically to the model list by applying the lapply() function. Since we intend to forecast a year ahead, h must be set to 12. The seasonal naïve forecasts are generated separately since it is a heuristic method and does not need a model fitting step. These forecasts can then be appended to the variable forecast in this way - forecasts$naive. The following R code shows these steps in greater detail.

```
# Generate forecasts
forecasts <- lapply(models, forecast, 12, PI = TRUE)

#attach the snaive forecast to    the models' list and plot them together
forecasts$snaive <- snaive(indprod_spain_train, 12)

#Plot forecasts
par(mfrow = c(2, 2))
```

```
for (f in forecasts) {
  plot(f)
  lines(indprod_spain_test, col = 'red')
}
```

Step 8.1: Evaluate the forecast accuracy

After the forecasts have been generated, the most important step lies in evaluating their accuracy. This includes checking the in-sample and out -of. sample accuracy measurement. The check the in-sample model fit, the residuals for each model should be computed and compared to that of a white noise series (Gaussian white noise). However, we know by now that residuals are not considered as true forecast errors. Moreover, even though automatic algorithms are used for model fitting, an in-sample accuracy measurement is not exempt from the risk of overfitting. Thus, to prevent overfitting and to find the true forecast errors, we must perform an out-of-sample accuracy test.

To demonstrate this, the test set that was withheld until now will be used. The forecasts generated will be compared to the test set and their difference will give the true forecast error. This can be calculated using the `accuracy()` function in `R`. It outputs a set of summary statistics as discussed in Chapter 5. Due to its advantages, we will choose the MASE metric for our evaluation since it provides a relative assessment of the models with respect to the naïve benchmark. The discussion of the results obtained from this demonstration will be dealt with in the following section. The following `R` code summarizes the various functions and arguments used to perform this step.

```
#Check accuracy and compare models based on MASE metric
acc <- lapply(forecasts, function(f){
  accuracy(f, indprod_spain_test)[2,,drop = FALSE]#drops the training set
residuals
})
#Reduce()applies the rbind function iteratively to elements of 'acc'.
acc <- Reduce(rbind, acc)

#naming each row
row.names(acc) <- names(forecasts)

#sort rows according to minimum MASE value
acc <- acc[order(acc[,'MASE']),]
```

```
#rounding off to two decimal places
round(acc, 2)
```

Step 8.2 (alternative approach): Time series cross-validation

As an alternative approach to data partitioning; used in cases where the data set is too small to generate a test set of a decent size, we can also use time series cross-validation. Unlike data partitioning, where the origin is fixed, and the sizes of the training and test set remain constant, this procedure generates a series of training and test set pairs, which vary in size with each validation step, resulting in a more robust accuracy measurement. The following R code shows one possible way of implementing this procedure in R.

However, it is important to mention that due to a large number of training and test sets generated by this procedure, obtaining a visual overview of the result is difficult. Moreover, this procedure was found to be time consuming for this dataset; especially, if used for large data sets which create many training and test set pairs as the origin rolls forward. Nevertheless, this is the recommended procedure when datasets are small and inadequate for partitioning purposes. But it must be noted that it takes preposterous computational effort and time when used on large datasets.

```
# Set up forecast functions for ETS and ARIMA models
fets <- function(x, h) {
  forecast(ets(x), h = h)
}
farima <- function(x, h) {
  forecast(auto.arima(x), h = h)
}
fneural <- function(x, h){
  forecast(nnetar(x), h = h)
}
fsnaive <- function(x, h){
  forecast(snaive(x), h = h)
}

# Compute CV errors for ETS as e1
e1 <- tsCV(indprod_spain.ts, fets, 1)
```

```r
# Compute CV errors for ARIMA as e2
e2 <- tsCV(indprod_spain.ts, farima, 1)

# Compute CV errors for NEURAL as e3
e3 <- tsCV(indprod_spain.ts, fneural, 1)

# Compute CV errors for SNAIVE as e4
e4 <- tsCV(indprod_spain.ts, fsnaive, 1)

# Find MSE of each model class
mse_ETS <- mean(e1^2, na.rm = TRUE)
mse_ETS
mse_ARIMA <- mean(e2^2, na.rm = TRUE)
mse_ARIMA
mse_NEURAL <- mean(e3^2, na.rm = TRUE)
mse_NEURAL
mse_NAIVE <- mean(e4^2, na.rm = TRUE)
mse_NAIVE
```

Step 9: Generate forecasts using the best model

After evaluating the results of the forecast from the previous validation steps, it was found that the ETS model generates the most accurate forecasts. The best forecasting model among the lot must then be used to produce the actual forecasts by considering the entire dataset. Note that the result of the forecast accuracy evaluation is directly assumed in this step without providing clarifications about the evaluation procedure. The actual evaluation procedure is dealt with in the next section.

To perform this step in `R`, simply pass the entire data set to the `ets()` function which fits the appropriate model for the entire dataset `indprod_spain.ts` as shown in the code snippet. Forecasts can then be produced for the next year, by passing this model to the `forecast` function and setting `h = 12`. Note that in this step, the origin has shifted to the end of the dataset and the actual observations are not known beyond the forecast origin. This contradicts the assumptions made during the data partitioning procedure.

This step marks the end of the forecasting methodology. But it must be kept in mind that the model performance should be constantly monitored to ensure forecast accuracy as the new observations keep occurring. Forecast errors must be calculated at regular intervals with the updated data set. This means that steps 5-9 need to be repeated regularly at pre-specified intervals to maintain forecast accuracy. In case there is a large deviation in errors, then the entire process must be repeated again from step 1.

```
# Find MSE of each model class
indprod_spain.ts %>% ets() %>% forecast(h= 12)%>% autoplot()
```

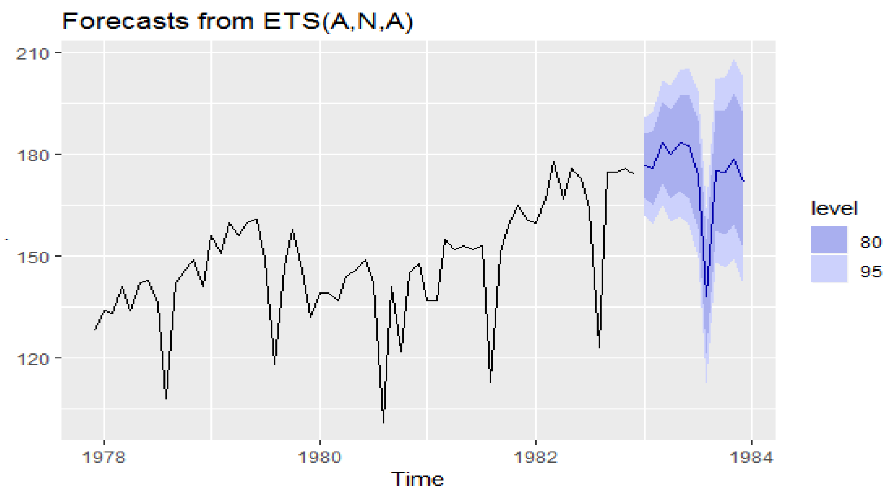


Figure 51. Forecasts using the best model.

6.2 Evaluation of results and discussion

After fitting the models to the training set data, the residuals of the training set can be calculated from its fitted values. By representing these residuals in terms of the summary statistics, the in-sample evaluation can be carried out as shown in Table 11 below. But when the forecasts generated from the training set data are compared against the values of the test set as shown in Figure 52, the evaluation of forecast accuracy becomes an out-of-sample evaluation. The latter produces true forecast errors and provides an accurate representation of forecast accuracy. The out-of-sample forecast errors, when represented in terms of the summary statistics, will look like the values shown in Table 12.

Methods	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
ETS	0.76	6.45	4.85	0.38	3.44	0.36	-0.03	NA
ARIMA	1.14	7.39	4.95	0.51	3.44	3.48	-0.03	NA
ANN	-0.003	6.75	5.54	-0.33	3.88	0.41	0.03	NA
SNAIVE	4.10	15.7	13.45	2.22	9.34	1	0.66	NA

Table 11. In-sample forecast accuracy based on residuals.

Total length of dataset = 61months; When test set = 12 months; training set = 49 months								
Methods	**ME**	**RMSE**	**MAE**	**MPE**	**MAPE**	**MASE**	**ACF1**	**Theil's U**
ETS	**5.41**	**8.41**	**6.70**	**2.99**	**3.95**	**0.50**	**0.25**	**0.36**
ARIMA	15.05	16.54	15.05	8.87	8.87	1.12	0.19	0.70
SNAIVE	18.33	19.36	18.33	10.91	10.91	1.36	0.28	0.78
ANN	20.02	23.07	20.07	11.53	11.57	1.49	0.34	0.98

Table 12. Out-of-sample forecast accuracy measurement.

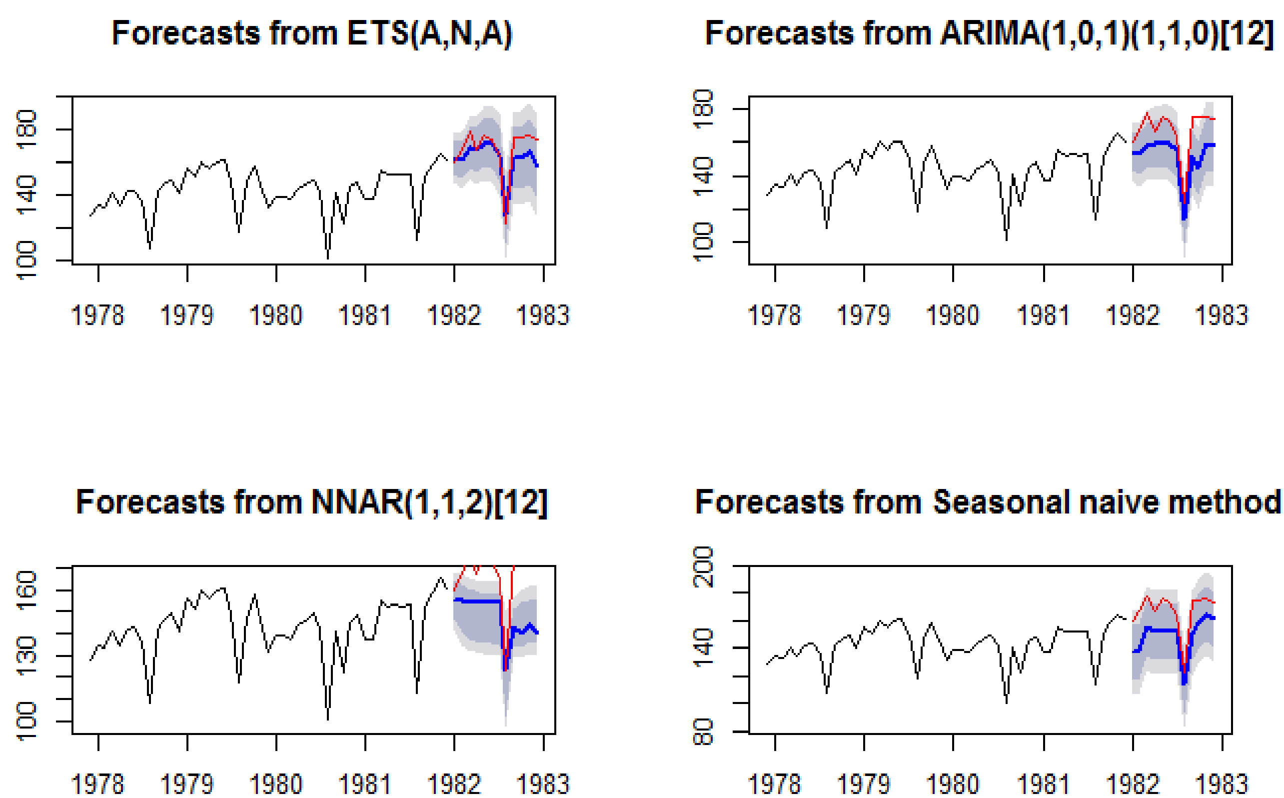


Figure 52. Plots comparing out-of-sample forecasts(blue) with hidden test set data(red).

But note that this out-of-sample evaluation is only valid for the chosen test set. To avoid any bias, the same steps were repeated by considering different lengths of test sets. The out-of-sample evaluation for each test set is displayed in Table 13. Just like the previous example, the methods are ranked in ascending order with the most accurate method on the top highlighted in green. The forecast plots supporting Table 13 can be found in the Appendix 2.2. Additionally, the output of the time series cross-validation procedure can be found in table 14.

Total length of dataset = 61months; When test set = 6 months; training set = 55 months								
Methods	**ME**	**RMSE**	**MAE**	**MPE**	**MAPE**	**MASE**	**ACF1**	**Theil's U**
ARIMA	**0.35**	**7.13**	**6.11**	**-0.13**	**3.82**	**0.41**	**0.19**	**0.21**
ETS	3.63	7.94	7.39	1.70	4.60	0.50	0.13	0.26
ANN	-5.91	15.29	11.47	-4.76	7.94	0.78	-0.12	0.46
SNAIVE	14.17	14.96	14.17	8.57	8.57	0.96	-0.09	0.50
Total length of dataset = 61months; When test set = 18 months; training set = 43 months								
Methods	**ME**	**RMSE**	**MAE**	**MPE**	**MAPE**	**MASE**	**ACF1**	**Theil's U**
ETS	**20.18**	**22.10**	**20.18**	**12.09**	**12.09**	**1.58**	**0.55**	**0.87**
SNAIVE	23.67	25.85	23.67	14.50	14.50	1.86	0.34	1.03
ANN	20.36	26.07	24.16	11.48	14.72	1.90	0.37	1.02
ARIMA	25.55	27.78	25.55	15.41	15.41	2.01	0.66	1.09
Total length of dataset = 61months; When test set = 24 months; training set = 37 months								
Methods	**ME**	**RMSE**	**MAE**	**MPE**	**MAPE**	**MASE**	**ACF1**	**Theil's U**
ARIMA	**3.36**	**10.67**	**7.20**	**1.78**	**4.46**	**0.51**	**0.27**	**0.48**
ETS	11.85	17.05	14.25	6.93	8.64	1.00	0.71	0.76
SNAIVE	20.42	24.36	20.75	12.48	12.73	1.46	0.54	1.10
ANN	20.73	25.25	22.01	12.47	13.41	1.55	0.49	1.16

Table 13. Out-of-sample-accuracy measurement for different test sets.

The output from time series cross-validation tsCV()

Method	**MSE**
ETS	**227.1025**
ARIMA	325.9822
ANN	410.0056

Table 14. Results of the time series cross-validation procedure.

The main findings obtained from this demonstration can be summarized as follows.

- Residuals (Table 11) were found to be clearly less than forecast errors (Table 12). This indicates that the models do a good job fitting the data but do not reciprocate their superior performance while forecasting.
- The naïve benchmark beats the ANN model in many instances during the out-of-sample accuracy measurement. (see Table 12 and 13).
- The out-of-sample evaluation presented in Table 12 indicates that ETS has the least forecast error, hence it is the most accurate model. It was closely followed by the ARIMA model.
- For different lengths of the test set, the results from the evaluation also vary as seen in Table 13. There can be numerous reasons responsible for this variation which needs further investigation. Hence, this aspect shall be considered out of the scope of this thesis and will not be discussed further. But more importantly, this phenomenon can be viewed as a shortcoming of the data partitioning procedure.
- As an alternative approach, the time series cross-validation procedure was also used, generating forecast evaluations over a series of test and training sets by means of a rolling origin. According to the cross-validations results shown in Table 14, the most accurate model was found to be ETS with an MSE of 227.1025 compared to the ARIMA and Neural Nets. The resulting error matrix is shown in Appendix 3.2 at the end of this thesis.

This thesis intends to only demonstrate the methods of forecasting and accuracy measurement. Any inferences or generalizations regarding the forecasting methods cannot be made solely depending on the results pertaining to a single dataset. To make any kind of generic inference normally demands rigorous empirical testing of forecasting methods applied to many time series, which vary in frequency, the number of observations and their nature. This kind of empirical testing is out of the scope of this thesis.

However, in the past, there have been numerous forecasting competitions that have tried to do this kind of empirical testing. The following table summarizes the main findings from such prestigious competitions. Only the findings relevant to this thesis are listed to maintain coherence with the research objectives. Relevant sources are accordingly provided for those interested in further details. Integrating these main findings into any forecast evaluation process will reduce costly mistakes and improve the overall effectiveness of forecasts in business applications.

<table>
<tr><th>Competition</th><th>Year</th><th>Remarks</th></tr>
<tr><td>M competition (Makridakis et al., 1982; Fildes et al., 1998)</td><td>1982</td><td>- First large-scale forecasting competition.
- 1001 time series used.
- Conducted by Spyros Makridakis.
- Many significant findings were established.</td></tr>
<tr><td colspan="3">Relevant findings:
• Statistically complex methods generally do not forecast more accurately than simpler ones.
• The accuracy of the methods varies with the type of accuracy measure being used.
• The accuracy of a combination of methods is higher than the combined method's individual accuracy; beating other individual methods as well.
• Performance of methods varies with the length of the forecast horizon.
• Time series data characteristics were found to be an important factor influencing method performance.</td></tr>
<tr><td>M2 competition (Makridakis et al., 1993)</td><td>1993</td><td>- Second competition of the M series.
- 29 time series (23 taken from major companies).
- Consolidation of findings from the previous competition.
- Incorporation of additional judgmental information into forecasts.</td></tr>
<tr><td colspan="3">Relevant findings:
• Exponential smoothing methods generate forecasts with the highest accuracy.
• The naïve method was found to be very hard to beat, particularly the naïve2 (Seasonal naive).
• Combining forecasters did not beat an individual forecaster's performance.</td></tr>
<tr><td>M3 competition (Makridakis and Hibon, 2000)</td><td>2000</td><td>- Results were similar to the M competition.
- 3003 time series used.
- Author's final attempt at settling the accuracy issue in time series.
- Machine learning (ML) methods participated for the first time.</td></tr>
<tr><td colspan="3">Relevant findings:
• Statistically sophisticated methods do not necessarily outperform simple methods on average i.e. it is hard to generalize on situations where sophisticated methods beat simple methods since it is difficult to categorize methods as simple or complex. However, in most cases, it was found that simple methods that are developed and established by practicing forecasters perform equally or sometimes even better than the complicated ones.</td></tr>
</table>

• Reconfirms the fact that the best method (most accurate method in that competition) varies not only with the accuracy measure being used, but also due to the type of data used (micro, industry, macro). • Combined methods score higher in terms of accuracy compared to individual methods. • The accuracy of the best method varies with the forecast horizon.		
NN3 competition (Crone, Hibon and Nikolopoulos, 2011)	2011	- Like the M3 competition but with a focus on computational intelligence(CI) and machine learning methods. - Two masked subsets from the M3 competition were chosen (111 and 11). - Intended to clear speculations regarding the forecast accuracy of such methods in comparison to statistical methods.
Relevant findings: • Just like in previous competitions, it was reconfirmed that time series data characteristics heavily influence the performance of ML and CI methods as well. • The accuracy of various methods relies heavily on the forecast horizon and method selection. • ML and CI methods showed varied results when different accuracy measures were applied. • ML and CI methods can compete well with well-established statistical methods but still cannot outperform them. • Reconfirming previous competition results - combinations (involving statistical methods) and 'Ensembles' (combined CI and ML methods) perform better than their individual counterparts. • Although the CI and ML methods failed to beat the established statistical methods that were praised in the M3 competition, these complicated methods should not be regarded as inferior to simple statistical methods in general. Many simple statistical methods do perform worse and there are many desirable benefits obtainable from sophistication.		
Tourist competition (Athanasopoulos *et al.*, 2011; Song and Hyndman, 2011)	2010	- Focused mainly on Tourism data (366 monthly, 427 quarterly and 518 annual series). - Both point forecasts and forecast intervals were evaluated. - MASE was used as an alternative measure.
Relevant findings: • It was found that pure time series models perform better than models with additional explanatory variables (exogenous variables) in the context of tourism data. • The naïve method proves to be hard to beat for annual data. • For seasonal data, it was found that the automated models like Forecast Pro, ARIMA and ETS produced accurate forecasts consistently compared to the other methods.		

• The superior performance of the MASE measure compared to the MAPE.		
M4 competition (Makridakis, Spiliotis and Assimakopoulos, 2018)	2018	- The most recent and up to date competition of the M series. - Methods tested on 100000-time series data sets. - More submissions of machine learning methods compared to the M3 (only one ML method participated).
Relevant findings: • Combination methods again dominated the rankings. • The best method was a hybrid that utilized both statistical and ML characteristics. • The best and second-best method predicted 95% prediction intervals correctly. • The six participating ML methods performed poorly (none of them beat the combined methods and only one was able to beat the Naive2 benchmark.)		

Table 15. Summary of past forecasting competitions.

The summary above clearly indicates that there has been a lot of empirical research conducted in evaluating the accuracy of forecasting methods. Strong evidence suggests that merely improving the model fit is often not enough to improve the post sample forecasting accuracy. For this reason, it is necessary to consider more information from the past by utilizing larger datasets. Another alternative for minimizing out-of-sample accuracy is through combining different models or methods. The advantages derived using combined methods can be persistently seen in the results from almost every forecasting competition. Furthermore, research has proven that quantitative forecasting methods are well-suited for identifying and extrapolating systematic patterns. In addition to that, they offer the possibility for forecasting large number of items, which is crucial to keep up with the rate at which data is generated today. However, their biggest limitation presents itself in their inability to accurately predict when a systematic pattern will change its course. Due to this reason, it is advisable to start with simple models initially and then add extra layers of complexity if needed. The tendency for complex forecasting should be avoided (Goldstein and Gigerenzer, 2009; Green and Armstrong, 2015). This uncertainty about the change is also the main reason behind the increase in forecast errors with increasing forecast horizons. Accurate forecasting hence becomes difficult and the uncertainty in the future can never be completely controlled necessitating a conservative forecasting approach (Armstrong, Green and Graefe, 2015). Therefore, every forecaster must be aware of these peculiarities in forecasting (Makridakis, 1986; Makridakis and Taleb, 2009).

Chapter 7. Conclusion

The main intention of this thesis was to bridge the gap between the vast amounts of research that is currently being conducted in this field and the way forecasting is being practiced in business today. By following the demonstrated forecasting methodology and utilizing the various data analysis and manipulation tools presented in this thesis, users can now potentially create robust and accurate forecasting procedures in a considerably less amount of development time, while maintaining an optimum utilization of resources. It is also evident that the availability of open-source statistical software such as R has allowed the dissemination of powerful forecasting tools to every individual and organization for free. This has drastic ramifications on the costs allocated for forecasting tasks. Furthermore, R offers many more visualization tools and packages which can improve data analysis that can immensely simplify model building. In this thesis, importance was given to ensure that only well-established, state-of-the-art forecasting methods relevant to today's business scenario are included in this study. To ease the method selection process, key factors that affect choosing of methods were emphasized. Given the importance of good forecasts, the right way to perform an evaluation of forecast accuracy was also presented in this thesis. Commonly used metrics for the evaluation of forecast accuracy were provided and their relative drawbacks mentioned. The entire forecasting procedure and its accuracy evaluation was demonstrated using R.

The distinction made between heuristic methods and statistical models is an essential step to eliminate the prevailing confusion that exists in academia due to the proliferation of forecasting methods that has occurred over time. Knowing that certain heuristic methods could find their way back to specific state space and ARIMA models, helps alleviate the obscurity induced by forecasting methods. Moreover, it becomes much easier to understand the intricacies of state-of-the-art models such as the ETS model, if it is modeled under a statistical framework such as that of the state space models and has well-established heuristics methods like the simple exponential smoothing and holt's winter's method traced back to it. The same applies for ARIMA models and its link to ETS models. This process of mapping novel methods to other well-established methods and their structural representation using a statistical modelling framework can greatly improve generalization and standardization efforts. This can potentially reduce problems with implementation, thereby building a foundation for a normalized forecasting methodology. This would allow forecasts to be effectively integrated into a business model to improve planning and decision making. Therefore, it holds potential implications for further research.

Due to the vastness of this field of research, many aspects were intentionally left out in this thesis due to restrictions of scope. This leaves plenty of room for further development in this field. Causal and multivariate forecasting are few such topics that show tremendous potential for further research. Another topic concerns itself with Neural Networks. Although ANNs were briefly introduced in this thesis, at this moment there is substantial research still going on in this field. Its steep rate of development and beneficial ability to deal with non-linearity autonomously will certainly keep them at the forefronts of forecasting research in the years to come. On the other hand, the consistent superior performance of combined forecasts and ensembles during many of the forecasting competitions conducted till now, makes them an attractive research prospect as well.

Bibliography

Akaike, H. (1974) ‘A new look at the statistical model identification’, *IEEE Transactions on Automatic Control*, 19(6), pp. 716–723. doi: 10.1109/TAC.1974.1100705.

Armstrong, J.S. (1985) *Long-range forecasting: From crystal ball to computer*. 2nd edn. New York: Wiley.

Armstrong, J.S. (2001) *Principles of Forecasting: A Handbook for Researchers and Practitioners (International series in operations research & management science)*: Springer.

Armstrong, J.S., Green, K.C. and Graefe, A. (2015) ‘Golden rule of forecasting: Be conservative’, *Journal of Business Research*, 68(8), pp. 1717–1731. doi: 10.1016/j.jbusres.2015.03.031.

Athanasopoulos, G., Hyndman, R.J., Song, H. and Wu, D.C. (2011) ‘The tourism forecasting competition’, *International Journal of Forecasting*, 27(3), pp. 822–844. doi: 10.1016/j.ijforecast.2010.04.009.

Balkin, S.D. and Ord, J.K. (2000) ‘Automatic neural network modelling for univariate time series’, *International Journal of Forecasting*, 16(4), pp. 509–515. doi: 10.1016/S0169-2070(00)00072-8.

Box, G.E.P., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M. (2016) *Time Series Analysis: Forecasting and Control*. 5th edn. (Wiley Series in Probability and Statistics Ser). Hoboken: Wiley; John Wiley & Sons, Incorporated.

Chan, N.H. (2010) *Time Series*. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Cleveland, R.B., Cleveland, W.S., McRae, J.E. and Terpenning, I. (1990) ‘STL: A Seasonal-Trend Decomposition. Journal of Official Statistics’, *Journal of Official Statistics*, 6(1), 3-73.

Crone, S.F., Hibon, M. and Nikolopoulos, K. (2011) ‘Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction’, *International Journal of Forecasting*, 27(3), pp. 635–660. doi: 10.1016/j.ijforecast.2011.04.001.

Dagum, E.B. and Bianconcini, S. (2016) *Seasonal adjustment methods and real time trend-cycle estimation*. New York NY: Springer Berlin Heidelberg.

Evans, M.K. (2003) *Practical Business Forecasting*. Oxford, UK: Blackwell Publishing Ltd.

Fildes, R. (2015) ‘Forecasters and rationality—A comment on Fritsche et al., Forecasting the Brazilian Real and Mexican Peso: Asymmetric loss, forecast rationality and forecaster herding’, *International Journal of Forecasting*, 31(1), pp. 140–143. doi: 10.1016/j.ijforecast.2014.01.003.

Fildes, R. and Petropoulos, F. (2015) ‘Simple versus complex selection rules for forecasting many time series’, *Journal of Business Research*, 68(8), pp. 1692–1701. doi: 10.1016/j.jbusres.2015.03.028.

Fildes, R., Goodwin, P., Lawrence, M. and Nikolopoulos, K. (2009) ‘Effective forecasting and judgmental adjustments: an empirical evaluation and strategies for improvement in supply-chain planning’, *International Journal of Forecasting*, 25(1), pp. 3–23. doi: 10.1016/j.ijforecast.2008.11.010.

Fildes, R., Hibon, M., Makridakis, S. and Meade, N. (1998) ‘Generalising about univariate forecasting methods: further empirical evidence’, *International Journal of Forecasting*, 14(3), pp. 339–358. doi: 10.1016/S0169-2070(98)00009-0.

Gardner, E.S. and Mckenzie, E. (1985) ‘Forecasting Trends in Time Series’, *Management Science*, 31(10), pp. 1237–1246. doi: 10.1287/mnsc.31.10.1237.

Gentry, L., Calantone, R.J. and Cui, S.A. (2006) ‘The forecasting classification grid: a typology for method selection’, *Journal of Global Business Management*, 2(1), pp. 48–60.

Gilliland, M., Tashman, L. and Sglavo, U. (2015) *Business Forecasting*. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Goldstein, D.G. and Gigerenzer, G. (2009) ‘Fast and frugal forecasting’, *International Journal of Forecasting*, 25(4), pp. 760–772. doi: 10.1016/j.ijforecast.2009.05.010.

Gooijer, J.G. de and Hyndman, R.J. (2006) ‘25 years of time series forecasting’, *International Journal of Forecasting*, 22(3), pp. 443–473. doi: 10.1016/j.ijforecast.2006.01.001.

Green, K.C. and Armstrong, J.S. (2015) ‘Simple versus complex forecasting: The evidence’, *Journal of Business Research*, 68(8), pp. 1678–1685. doi: 10.1016/j.jbusres.2015.03.026.

Hansmann, K.-W. (1983) *Kurzlehrbuch Prognoseverfahren: Mit Aufgaben u. Lösungen*. (Gabler-Lehrbuch). Wiesbaden: Gabler.

Hyndman, R.J. (2018) *Forecasting: Principles and practice ; [a comprehensive introduction to the latest forecasting methods using R ; learn to improve your forecast accuracy using dozens of real data examples.]*. Lexington, Ky.: Otexts.

Hyndman, R.J. (2018) *Time Series Data Library*, 2 April. Available at: https://datamarket.com/data/list/?q=provider:tsdl (Accessed: 1 January 2019).

Hyndman, R.J. and Khandakar, Y. (2008) 'Automatic Time Series Forecasting: The forecast Package for R', *Journal of Statistical Software*, 27(3). doi: 10.18637/jss.v027.i03.

Hyndman, R.J. and Koehler, A.B. (2006) 'Another look at measures of forecast accuracy', *International Journal of Forecasting*, 22(4), pp. 679–688. doi: 10.1016/j.ijforecast.2006.03.001.

Hyndman, R.J., Koehler, A., Ord, K. and Snyder, R. (2008) *Forecasting with Exponential Smoothing: The State Space Approach*. (Springer Series in Statistics). Berlin: Springer. Available at: http://d-nb.info/989924998/34.

Hyndman, R.J., Koehler, A.B., Snyder, R.D. and Grose, S. (2002) 'A state space framework for automatic forecasting using exponential smoothing methods', *International Journal of Forecasting*, 18(3), pp. 439–454. doi: 10.1016/S0169-2070(01)00110-8.

Hyndman R, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O'Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F (2018). *forecast: Forecasting functions for time series and linear models*. R package version 8.4, http://pkg.robjhyndman.com/forecast.

Kim, S. and Kim, H. (2016) 'A new metric of absolute percentage error for intermittent demand forecasts', *International Journal of Forecasting*, 32(3), pp. 669–679. doi: 10.1016/j.ijforecast.2015.12.003.

Koehler, A.B., Snyder, R.D., Ord, J.K. and Beaumont, A. (2012) 'A study of outliers in the exponential smoothing approach to forecasting', *International Journal of Forecasting*, 28(2), pp. 477–484. doi: 10.1016/j.ijforecast.2011.05.001.

Ledolter, J. (2006) 'Prediction and Forecasting', in Kotz, S., Read, C.B., Balakrishnan, N., Vidakovic, B. and Johnson, N.L. (eds.) *Encyclopedia of statistical sciences*, 2nd edn. Hoboken, N.J.: Wiley-Interscience, 189p.

Makridakis, S. (1986) ‘The art and science of forecasting An assessment and future directions’, *International Journal of Forecasting*, 2(1), pp. 15–39. doi: 10.1016/0169-2070(86)90028-2.

Makridakis, S. (1996) ‘Forecasting: its role and value for planning and strategy’, *International Journal of Forecasting*, 12(4), pp. 513–537. doi: 10.1016/S0169-2070(96)00677-2.

Makridakis, S. and Hibon, M. (2000) ‘The M3-Competition: results, conclusions and implications’, *International Journal of Forecasting*, 16(4), pp. 451–476. doi: 10.1016/S0169-2070(00)00057-1.

Makridakis, S. and Taleb, N. (2009) ‘Living in a world of low levels of predictability’, *International Journal of Forecasting*, 25(4), pp. 840–844. doi: 10.1016/j.ijforecast.2009.05.008.

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E. and Winkler, R. (1982) ‘The accuracy of extrapolation (time series) methods: Results of a forecasting competition’, *Journal of Forecasting*, 1(2), pp. 111–153. doi: 10.1002/for.3980010202.

Makridakis, S., Chatfield, C., Hibon, M., Lawrence, M., Mills, T., Ord, K. and Simmons, L.F. (1993) ‘The M2-competition: A real-time judgmentally based forecasting study’, *International Journal of Forecasting*, 9(1), pp. 5–22. doi: 10.1016/0169-2070(93)90044-N.

Makridakis, S., Hibon, M., Lusk, E. and Belhadjali, M. (1987) ‘Confidence intervals’, *International Journal of Forecasting*, 3(3-4), pp. 489–508. doi: 10.1016/0169-2070(87)90045-8.

Makridakis, S., Spiliotis, E. and Assimakopoulos, V. (2018) ‘The M4 Competition: Results, findings, conclusion and way forward’, *International Journal of Forecasting*, 34(4), pp. 802–808. doi: 10.1016/j.ijforecast.2018.06.001.

Makridakis, S., Wheelwright, S.C. and Hyndman, R.J. (1998) *Forecasting: Methods and applications*. 3rd edn. New York: Wiley.

Makridakis, S., Williams, T., Kirkham, R. and Papadaki, M. (2018) ‘Forecasting, uncertainty and risk management’, *International Journal of Forecasting*. doi: 10.1016/j.ijforecast.2018.10.002.

Makridakis, S.G. and Wheelwright, S.C. (1989) *Forecasting methods for management*. 5th edn. New York: Wiley.

Morlidge, S. and Player, S. (2010) *Future ready: How to master business forecasting*. Hoboken, N.J: Wiley.

Nelson, M., Hill, T., Remus, W. and O'Connor, M. (1999) ‘Time series forecasting using neural networks: should the data be deseasonalized first?’ *Journal of Forecasting*, 18(5), pp. 359–367. doi: 10.1002/(SICI)1099-131X(199909)18:5<359::AID-FOR746>3.0.CO;2-P.

Ord, J.K. and Fildes, R. (2013) *Principles of business forecasting*. Mason OH: South-Western Cengage Learning.

Pegels, C.C. (1969) ‘Exponential Forecasting: Some New Variations’, *Management Science* (15), 311–315.

Robert A. Yaffee (2010) *Forecast evaluation with Stata*, United Kingdom Stata Users' Group Meetings 2010 10, Stata Users Group.

R Core Team (2013). R: A language and environment for statistical computing.

Sharma, P.N., Shmueli, G., Sarstedt, M., Danks, N. and Ray, S. (2018) ‘Prediction-Oriented Model Selection in Partial Least Squares Path Modelling’, *Decision Sciences*, 21(1), p. 243. doi: 10.1111/deci.12329.

Shmueli, G. and Lichtendahl, K.C. (2016) *Practical time series forecasting with R: A hands-on guide* [Place of publication not identified]: Axelrod Schnall Publishers.

Song, H. and Hyndman, R.J. (2011) ‘Tourism forecasting: An introduction’, *International Journal of Forecasting*, 27(3), pp. 817–821. doi: 10.1016/j.ijforecast.2011.03.001.

Sugiura, N. (1978) ‘Further analysts of the data by akaike' s information criterion and the finite corrections’, *Communications in Statistics - Theory and Methods*, 7(1), pp. 13–26. doi: 10.1080/03610927808827599.

Tashman, L.J. (2000) ‘Out-of-sample tests of forecasting accuracy: an analysis and review’, *International Journal of Forecasting*, 16(4), pp. 437–450. doi: 10.1016/S0169-2070(00)00065-0.

Theil, H. (1975) *Applied Economic Forecasting*. (Studies in Mathematical and Managerial Economics, vol 4). Amsterdam: North-Holland Publishing Company.

Winters, P.R. (1960) ‘Forecasting Sales by Exponentially Weighted Moving Averages’, *Management Science*, 6(3), pp. 324–342. doi: 10.1287/mnsc.6.3.324.

Zhang, G., Eddy Patuwo, B. and Y. Hu, M. (1998) ‘Forecasting with artificial neural networks’, *International Journal of Forecasting*, 14(1), pp. 35–62. doi: 10.1016/S0169-2070(97)00044-7.

Appendices

A1. Contains all R codes used in this thesis. (functions are marked in blue)

Figure 5. A typical time series plot.

```
#Importing excel file into R.
library(readxl)
annual_bituminous_coal_production<-read_excel("F:/MASTERTHESIS/R
files/Datamarket datasets/annual-bituminous-coal-production.xlsx")
view(annual_bituminous_coal_production)

#Removing the first column and using only the values column.
bit_coal <- annual_bituminous_coal_production[ , -1]

#Procedure to convert into time series object.
bit_coal.ts <- ts(bit_coal, start = 1920, frequency = 1)
plot(bit_coal.ts, xlab = "Time(years)", ylab = "Produced coal(millions of net
tonnes)", main = "Annual bituminous coal production in USA")
```

Figure 6: R code for generating time series plots

```
#separating only the trend from the original series using window() function.
chickentrend <- window(chicken, start = 1945)

#stacking function par()
par(mfrow = c(2,2))

#plotting the four time series
plot(chickentrend, xlab = "time", ylab = "Price($)", main = "[1] Annual price of
chicken in USA")
plot(usdeaths, xlab = "time", ylab = "no. of deaths", main = "[2] Monthly
accidental deaths in USA")
plot(lynx, xlab = "time", ylab = "no. of trappings", main = "[3] Annual Canadian
Lynx trappings")
plot(a10, xlab = "time", ylab = "sales", main = "[4] Monthly anti-diabetic drug
sales in Australia")
```

Figure 7. R code for generating seasonal plots

```
autoplot(usdeaths)
p1  <-  ggseasonplot(usdeaths, year.labels=TRUE, year.labels.left=TRUE) +
ylab("no. of deaths") +
```

```
ggtitle("Seasonal plot: [2] Monthly accidental deaths in USA")

p2  <-  ggseasonplot(a10, year.labels=TRUE, year.labels.left=TRUE) +
ylab("$ million") +
ggtitle("Seasonal plot: [4] Monthly antidiabetic drug sales")
grid.arrange(p1, p2, nrow = 2)
```

Figure 8. R code for generating seasonal sub-series plots

```
#Subseries plot
q1 <- ggsubseriesplot(usdeaths) +
  ylab("no. of deaths") +
  ggtitle("Seasonal subseries plot:[2] Monthly accidental deaths in USA")
q2 <- ggsubseriesplot(a10) +
  ylab("$ million") +
  ggtitle("Seasonal subseries plot: [4] Monthly antidiabetic drug sales")

 #Stacking function
grid.arrange(q1, q2, nrow = 2)
```

Figure 9, Figure 10, Figure 11, Figure 12. R codes for Lag plots

```
#Splitting the trend from the original chicken series
chickentrend <- window(chicken, start = 1945)


#Generating the lag plots
gglagplot(chickentrend) +
ggtitle("Lag plot:[1] Annual price of chicken in USA") #Fig.9
gglagplot(usdeaths) +
ggtitle("Lag plot:[2] Monthly accidental deaths in USA") #Fig.10
gglagplot(lynx) +
  ggtitle("Lag plot:[3] Annual Canadian Lynx trappings") #Fig.11
gglagplot(a10) +
  ggtitle("Lag plot:[4] Monthly antidiabetic drug sales") #Fig.12
```

Figure 13. R codes generating ACF plots

```
#Splitting the trend fromt the original chicken series
chickentrend <- window(chicken, start = 1945)

#Generating Acf plots
g1 <- ggAcf(chickentrend) +
  ggtitle(" [1] Annual price of chicken in USA")
g2 <- ggAcf(usdeaths) +
```

```
  ggtitle("[2] Monthly accidental deaths in USA")
g3 <- ggAcf(lynx) +
  ggtitle("[3] Annual Canadian Lynx trappings")
g4 <- ggAcf(a10) +
  ggtitle("[4] Monthly antidiabetic drug sales")

#Stacking function
grid.arrange(g1, g2, g3, g4 , nrow = 2)
```

Figure 14. R code for performing classical decomposition (additive)

```
decomp.add <- decompose(a10,  type = "additive")
autoplot(decomp.add) + ggtitle("Classical additive decomposition")
```

Figure 15. R code for performing classical decomposition (multiplicative)

```
decomp.multi <- decompose(a10,  type = "multiplicative")
autoplot(decomp.multi) + ggtitle("Classical multiplicative decomposition")
```

Figure 16. R code for performing manual STL decomposition.

```
decompSTL <- stl(ausbeer, t.window = 13, s.window = "periodic", robust = TRUE)
autoplot(decompSTL)+ xlab("Year")+ ggtitle("STL decomposition")
```

Figure 17. R code for Automated STL decomposition

```
decompmstl <- mstl(ausbeer)
autoplot(decompmstl) + xlab("Year") + ggtitle("Automated STL decomposition")
```

Figure 18 & Figure 19. R codes for performing Box-Cox transformaitons.

```
#R code for Box-Cox transformation (Manual parameter selection)
autoplot(BoxCox(elec, lambda = -1 )

#R code for Box-Cox transformation ( Automatic parameter selection)
autoplot(BoxCox(elec, lambda = BoxCox.lambda(elec)))

>Output optimum λ:  0.2654076

par(mfrow = c(2,2))
plot(BoxCox(elec, lambda = -1), main = "lambda = -1")
plot(BoxCox(elec, lambda = -0.4), main = "lambda = -0.4")
plot(BoxCox(elec, lambda = -0.1), main = "lambda = -0.1")
plot(BoxCox(elec, lambda = 0), main = "lambda = 0")
```

```
par(mfrow = c(2,2))
plot(BoxCox(elec, lambda = 0.3), main = "lambda = 0.3")
plot(BoxCox(elec, lambda = 0.6), main = "lambda = 0.6")
plot(BoxCox(elec, lambda = 1), main = "lambda = 1")
plot(BoxCox(elec, lambda = 1.3), main = "lambda = 1.3")

par(mfrow = c(2,2))
plot(BoxCox(elec, lambda = 1.7), main = "lambda = 1.7")
plot(BoxCox(elec, lambda = 2), main = "lambda = 2")

BoxCox.lambda(elec)
```

Figure 20. R code for generating forecasts using the mean, naïve and seasonal naïve and drift.

```
# Partitioning plots
par(mfrow = c(2,2))
plot(meanf(shampoo, 10), main = "Mean")
plot(naive(shampoo, 10), main = "Naive")
plot(snaive(shampoo, 10), main = "Seasonal Naive")
plot(rwf(shampoo, 10, drift = TRUE),main = "Drift")
```

Figure 21. R code for superimposed plot.

```
# Set training data from 1992 to 2007
shampoo_train <- window(shampoo,start=1,end=c(2,6))

# Plot forecasts
autoplot(shampoo_train) +
  autolayer(meanf(shampoo_train, h=10),
            series="Mean", PI=FALSE) +
  autolayer(naive(shampoo_train, h=10),
            series="Naïve", PI=FALSE) +
  autolayer(snaive(shampoo_train, h=10),
            series="Seasonal naïve", PI=FALSE) +
  autolayer(rwf(shampoo_train, h=10, drift = TRUE), series = "Drift", PI =
FALSE)+
  ggtitle("Forecasts for monthly shampoo sales") +
  xlab("Year") + ylab("Sales") +
  guides(colour=guide_legend(title="Forecast"))
```

Figure 22. R code for estimating parameters for SES method.

```
#separate advertisement data set from insurance dataset
advert<- insurance [ , 2]
advertdata<-  window(advert, end=2005)
```

```r
autoplot(advertdata)
# Estimate parameters

fc <- ses(advertdata, h=24)
autoplot(fc) +
autolayer(fitted(fc), series="Fitted") +
ylab("Advertisement expenditure") + xlab("Year") + ggtitle("Monthly advertising expenditure of an US insurance company")
summary(fc)
```

Figure 23. R code for forecasting using the Holt's method

```r
tourists_linear <- holt(austa, h=15)
tourists_damped <- holt(austa, damped = TRUE, phi = 0.9, h = 15)

autoplot(austa) +
  autolayer(tourists_linear, series = "Holt's linear method", PI = FALSE) +
  autolayer(tourists_damped, series = "Damped Holt's method", PI = FALSE) +
  ggtitle("Forecasts from Holt's method") +  xlab("Year") + ylab("Air passengers in Australia (millions)") +  guides(colour = guide_legend(title = "Forecast"))
```

Figure 24. R code generates forecasts using the Holt Winter's exponential smoothing

```r
drug <- window(a10,start=2000)
fit1 <- hw(drug,seasonal="additive", h = 48)
fit2 <- hw(drug,seasonal="multiplicative", h = 48)
autoplot(drug) +
  autolayer(fit1, series="HW additive forecasts", PI=FALSE) +
  autolayer(fit2, series="HW multiplicative forecasts", PI=FALSE) +
  xlab("Year") +
  ylab("Sales") +
  ggtitle("Monthly antibiotic drug sales in Australia") +
  guides(colour=guide_legend(title="Forecast"))
```

Figure 26. Time series plot showing forecasts generated by the ETS model.

```r
#For our example we choose the dataset containing Monthly corticosteroid drug sales in Australia from July 1991 to June 2008

#Choosing the dataset containing the monthly corticosteroid drug sales in Australia
plot(h02, main = "Monthly drug sales", xlab = "Years", ylab = "Drugs sold")

#Fitting the ETS model to the data
ets(h02)
drug_sales_model <- ets(h02)
```

```
#Forecasting for the next years (For monthly data h = 12)
forecast_drug_sales <- forecast (drug_sales_model, h = 12)
plot(forecast_drug_sales, main = "Next years Monthly drug sales", xlab =
"Years", ylab = "Drugs sold")
```

Figure 27. R codes generate Time plot and ACF plot for a white noise series.

```
#White noise series
set.seed(30)
y <- ts(rnorm(50))
w1 <- autoplot(y) + ggtitle("White noise")

# plot Acf
w2 <- ggAcf(y)

#Stacking function
grid.arrange(w1, w2, nrow = 2)
```

Figure 28. Time plots showing non-stationary characteristics of time series

```
par(mfrow = c(1,3))
plot(BJsales, main = "plot A - datasets::BJsales ", ylab = "Sales" )
plot(h02, main = "plot B - fpp2::h02", ylab = "H drug sales" )
plot(a10, main = "plot C - fpp2::a10", ylab = "A drug sales")
```

Figure 30. R codes for generating a Random walk model

```
diff_goog200 <- diff(goog200)
par(mfrow = c(2,2))
plot(goog200, main = "Original series (fpp2::goog200)", ylab = "daily stock
price index")
Acf(goog200, ylab = "ACF")
plot(diff_goog200, main = "First differences of (fpp2::goog200)", ylab = "daily
stock price index")
Acf(diff_goog200, ylab = "ACF")
```

Figure 31. Time plots before and after differencing

```
plot(BJsales, main = "Plot A: Before differencing")
diff.BJsales <- diff(BJsales)
plot(diff.BJsales, main = "Plot A: After differencing")
```

```
plot(h02, main = "Plot B: Before differencing")
seas_diff.h02 <- diff(h02, lag = 12)
plot(seas_diff.h02, main = "seasonal diff")
diff.seas_diff.h02 <- diff(seas_diff.h02)
plot(diff.seas_diff.h02, main ="After differencing")
plot(a10, main = "Plot C: Before differencing")
transform.a10 <- BoxCox(a10, lambda = BoxCox.lambda(a10))
plot(transform.a10, main = "Box. Cox transform")
diff_seas.transform.a10 <- diff(transform.a10, lag = 12)
plot(diff_seas.transform.a10, main = "seasonal diff")
diff.diff_seas.transform.a10 <- diff(diff_seas.transform.a10)
plot(diff.diff_seas.transform.a10, main = "After differencing")
```

Figure 32. ACF plots before and after differencing.

```
par(mfrow = c(3,2))
Acf(BJsales, main = "Acf A (BJsales) : Before differencing")
Acf(diff.BJsales, main = "Acf A: After differencing")
Acf(h02, main ="Acf B (h02) : Before differencing")
Acf(diff.seas_diff.h02, main = "Acf B: After differencing")
Acf(a10, main = "Acf C (a10) : Before differencing")
Acf(diff.diff_seas.transform.a10, main = "Acf C: After differencing")
```

Figure 33. ACF and PACF behavior for AR models.

```
ar_1 <- arima.sim(model = list(order = c(1, 0, 0), ar = .9), n = 100)
ar_2 <- arima.sim(model = list(order = c(2, 0, 0), ar = c(1.5, -.75)), n = 200)
ma_1 <- arima.sim(model = list(order = c(0, 0, 1), ma = -.8), n = 100)
ma_2 <- arima.sim(model = list(order = c(0, 0, 2), ma = c(-.8, -.4)), n = 200)
arma_1_1 <- arima.sim(model = list(order = c(1, 0, 1), ar = -.9, ma = .8), n =
250)
arma_2_1 <- arima.sim(model = list(order = c(2, 0, 1), ar = c(1, -.9), ma = .8),
n = 250)

par(mfcol = c(3,3))
plot(ar_1, main = "AR(1) = ARIMA(1,0,0)")
Acf(ar_1, main = "Acf tails off")
Pacf(ar_1, main = "Pacf cuts off after lag 1")
plot(ar_2, main = "AR(2) = ARIMA(2,0,0)")
Acf(ar_2, main = "Acf tails off")
Pacf(ar_2, main = "Pacf cuts off after lag 2")
```

Figure 34. ACF and PACF behavior for pure MA models.

```
par(mfcol = c(3,3))
plot(ma_1, main = "MA(1) = ARIMA(0,0,1)")
Acf(ma_1, main = "Acf cuts off after lag 1")
```

```
Pacf(ma_1, main = "Pacf tails off ")
plot(ma_2, main = "MA(2) = ARIMA(0,0,2)")
Acf(ma_2, main = "Acf cuts off after lag 2")
Pacf(ma_2, main = "Pacf tails off ")
```

Figure 35. Typical ACF and PACF behavior for pure ARMA models.

```
par(mfcol = c(3,3))
plot(arma_1_1, main = "ARMA(1,1) = ARIMA(1,0,1)")
Acf(arma_1_1, main = "Acf tails off")
Pacf(arma_1_1, main = " Pacf tails off ")
plot(arma_2_1, main = "ARMA(2,1) = ARIMA(2,0,1)")
Acf(arma_2_1, main = "Acf tails off")
Pacf(arma_2_1, main = " Pacf tails off ")
```

Figure 36. Differencing operation for seasonal series.

```
auto.arima(unemp)
plot(unemp, main = "Original series (astsa::unemp)")
diff.unemp <- diff(unemp)
sdiff.diff.unemp <- diff(diff.unemp, lag = 12)
plot(sdiff.diff.unemp, main = "differenced series")
acf2(sdiff.diff.unemp)
par(mfrow = c(2,1))
```

Figure 37. The ACF and PACF plots for seasonal ARIMA models.

```
Acf(sdiff.diff.unemp)
Pacf(sdiff.diff.unemp)
fit <- Arima(unemp, order=c(2,1,0), seasonal=c(0,1,1), lambda=0)
```

Figure 38. Resulting plots from the residual analysis. code for performing residual analysis

```
checkresiduals(fit)
```

Figure 39. Forecasts generated using ARIMA modelling procedure.

```
fit1 <- auto.arima(BJsales)
fit2 <- auto.arima(a10)

forecast1 <- forecast(fit1, 8)
forecast2 <- forecast(fit2, 36)

par(mfrow = c(1,2))
plot(forecast1, main ="8 year ahead annual forecasts", ylab = "series: BJsales")
plot(forecast2, main ="3 year ahead seasonal forecasts", ylab = "series: a10")
```

Figure 45. Forecasts generated using the neural nets.

```
#fitting the neural nets model to the data
fit_nn <- nnetar(lynx)

#forecast generated for the next 30 years; only for demonstration purposes!
future_nn <- forecast(fit_nn, h = 30, PI = TRUE)#Prediction intervals on!

#plotting
autoplot(future_nn)
```

A2. Additional Figures

(2.1a) Recursive formulas for all exponential smoothing methods.

Trend	Seasonal: N	Seasonal: A	Seasonal: M
N	$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$ $\hat{y}_{t+h\|t} = \ell_t$	$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t + s_{t-m+h_m^+}$	$\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1}) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t s_{t-m+h_m^+}$
A	$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $\hat{y}_{t+h\|t} = \ell_t + hb_t$	$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t-m+h_m^+}$	$\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = (\ell_t + hb_t)s_{t-m+h_m^+}$
A_d	$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t$	$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t + s_{t-m+h_m^+}$	$\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1})) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = (\ell_t + \phi_h b_t)s_{t-m+h_m^+}$
M	$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$ $\hat{y}_{t+h\|t} = \ell_t b_t^h$	$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t b_t^h + s_{t-m+h_m^+}$	$\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1}b_{t-1})) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t b_t^h s_{t-m+h_m^+}$
M_d	$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}^\phi$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^\phi$ $\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h}$	$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^\phi$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^\phi$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}^\phi) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} + s_{t-m+h_m^+}$	$\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^\phi$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^\phi$ $s_t = \gamma(y_t/(\ell_{t-1}b_{t-1}^\phi)) + (1-\gamma)s_{t-m}$ $\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} s_{t-m+h_m^+}$

In each case, ℓ_t denotes the series level at time t, b_t denotes the slope at time t, s_t denotes the seasonal component of the series at time t, and m denotes the number of seasons in a year; α, β^*, γ and ϕ are constants, $\phi_h = \phi + \phi^2 + \cdots + \phi^h$ and $h_m^+ = \lfloor (h-1) \bmod m \rfloor + 1$.

(2.1b) ETS models for additive errors

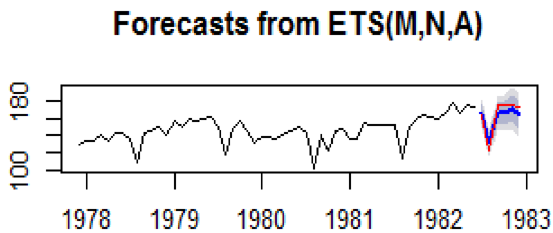
Trend	Seasonal: N	Seasonal: A	Seasonal: M
N	$\mu_t = \ell_{t-1}$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$	$\mu_t = \ell_{t-1} + s_{t-m}$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$\mu_t = \ell_{t-1}s_{t-m}$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/\ell_{t-1}$
A	$\mu_t = \ell_{t-1} + b_{t-1}$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$	$\mu_t = \ell_{t-1} + b_{t-1} + s_{t-m}$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$\mu_t = (\ell_{t-1} + b_{t-1})s_{t-m}$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + b_{t-1})$
A_d	$\mu_t = \ell_{t-1} + \phi b_{t-1}$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$	$\mu_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m}$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$\mu_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m}$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + \phi b_{t-1})$
M	$\mu_t = \ell_{t-1}b_{t-1}$ $\ell_t = \ell_{t-1}b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t/\ell_{t-1}$	$\mu_t = \ell_{t-1}b_{t-1} + s_{t-m}$ $\ell_t = \ell_{t-1}b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t/\ell_{t-1}$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$\mu_t = \ell_{t-1}b_{t-1}s_{t-m}$ $\ell_t = \ell_{t-1}b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/(s_{t-m}\ell_{t-1})$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1}b_{t-1})$
M_d	$\mu_t = \ell_{t-1}b_{t-1}^{\phi}$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi} + \alpha\varepsilon_t$ $b_t = b_{t-1}^{\phi} + \beta\varepsilon_t/\ell_{t-1}$	$\mu_t = \ell_{t-1}b_{t-1}^{\phi} + s_{t-m}$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi} + \alpha\varepsilon_t$ $b_t = b_{t-1}^{\phi} + \beta\varepsilon_t/\ell_{t-1}$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$\mu_t = \ell_{t-1}b_{t-1}^{\phi}s_{t-m}$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1}^{\phi} + \beta\varepsilon_t/(s_{t-m}\ell_{t-1})$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1}b_{t-1}^{\phi})$

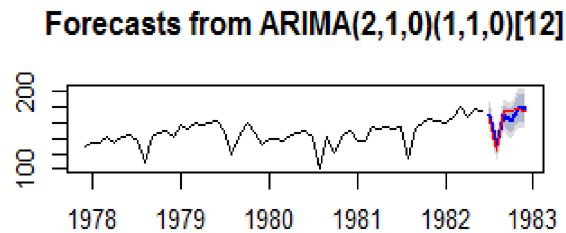
2.1c ETS models for multiplicative errors

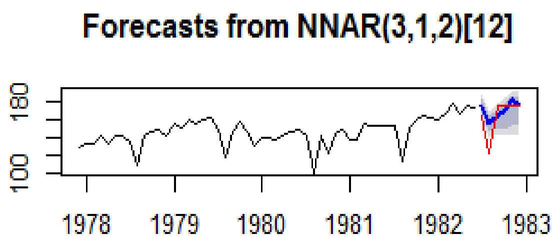
Trend	Seasonal: N	Seasonal: A	Seasonal: M
N	$\mu_t = \ell_{t-1}$ $\ell_t = \ell_{t-1}(1+\alpha\varepsilon_t)$	$\mu_t = \ell_{t-1} + s_{t-m}$ $\ell_t = \ell_{t-1} + \alpha(\ell_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + s_{t-m})\varepsilon_t$	$\mu_t = \ell_{t-1}s_{t-m}$ $\ell_t = \ell_{t-1}(1+\alpha\varepsilon_t)$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$
A	$\mu_t = \ell_{t-1} + b_{t-1}$ $\ell_t = (\ell_{t-1} + b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$	$\mu_t = \ell_{t-1} + b_{t-1} + s_{t-m}$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$	$\mu_t = (\ell_{t-1} + b_{t-1})s_{t-m}$ $\ell_t = (\ell_{t-1} + b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$
A_d	$\mu_t = \ell_{t-1} + \phi b_{t-1}$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$	$\mu_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m}$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$	$\mu_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m}$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$
M	$\mu_t = \ell_{t-1}b_{t-1}$ $\ell_t = \ell_{t-1}b_{t-1}(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}(1+\beta\varepsilon_t)$	$\mu_t = \ell_{t-1}b_{t-1} + s_{t-m}$ $\ell_t = \ell_{t-1}b_{t-1} + \alpha(\ell_{t-1}b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1} + \beta(\ell_{t-1}b_{t-1} + s_{t-m})\varepsilon_t/\ell_{t-1}$ $s_t = s_{t-m} + \gamma(\ell_{t-1}b_{t-1} + s_{t-m})\varepsilon_t$	$\mu_t = \ell_{t-1}b_{t-1}s_{t-m}$ $\ell_t = \ell_{t-1}b_{t-1}(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}(1+\beta\varepsilon_t)$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$
M_d	$\mu_t = \ell_{t-1}b_{t-1}^{\phi}$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi}(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}^{\phi}(1+\beta\varepsilon_t)$	$\mu_t = \ell_{t-1}b_{t-1}^{\phi} + s_{t-m}$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi} + \alpha(\ell_{t-1}b_{t-1}^{\phi} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1}^{\phi} + \beta(\ell_{t-1}b_{t-1}^{\phi} + s_{t-m})\varepsilon_t/\ell_{t-1}$ $s_t = s_{t-m} + \gamma(\ell_{t-1}b_{t-1}^{\phi} + s_{t-m})\varepsilon_t$	$\mu_t = \ell_{t-1}b_{t-1}^{\phi}s_{t-m}$ $\ell_t = \ell_{t-1}b_{t-1}^{\phi}(1+\alpha\varepsilon_t)$ $b_t = b_{t-1}^{\phi}(1+\beta\varepsilon_t)$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$

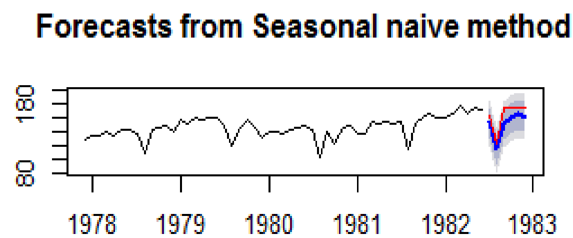
2.2 Plots supporting the results shown in Table 13.

(2.2a) Total length of dataset = 61months; When test set = 6 months; training set = 55 months

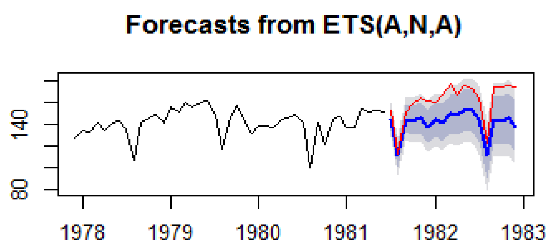


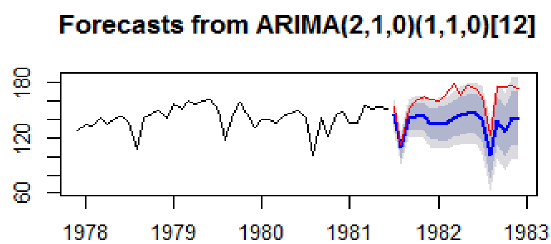


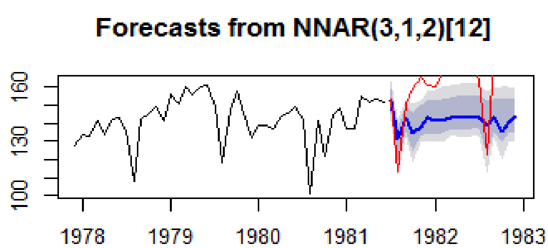


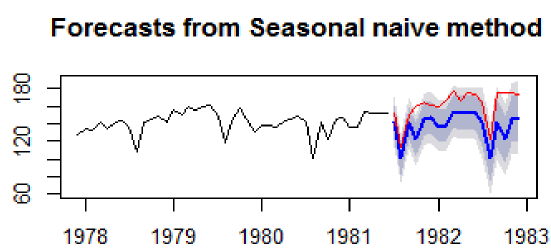


(2.2b) Total length of dataset = 61months; When test set = 18 months; training set = 43 months

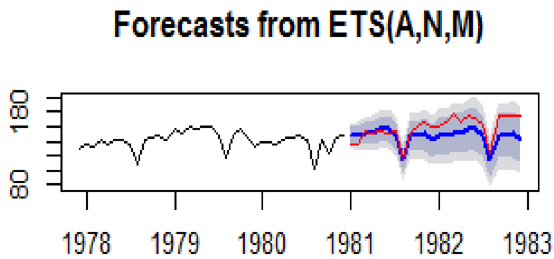


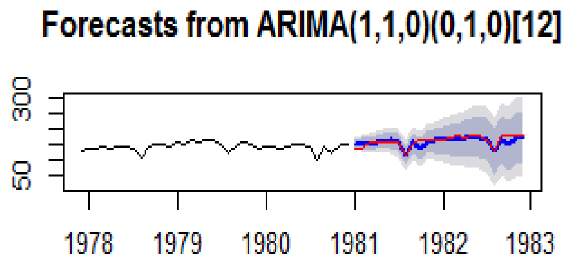


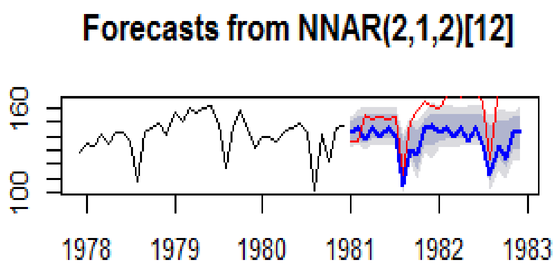


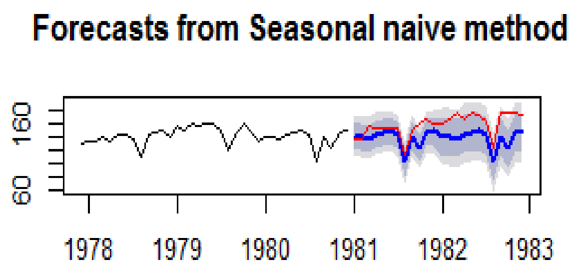


(2.2c) Total length of dataset = 61months; When test set = 18 months; training set = 43 months









A3. Outputs from R functions.

(A3.1) Results from decomposition. The different time series components and the values.

```
> trendcycle(decompSTL)
         Qtr1     Qtr2     Qtr3     Qtr4
1956 255.3971 256.3194 257.2417 258.1811
1957 259.1205 260.4679 261.8153 262.5781
1958 263.3408 263.5405 263.7402 264.5475
1959 265.3548 266.8020 268.2493 271.3309
1960 274.4126 276.2582 278.1037 278.9775
1961 279.8512 280.7421 281.6331 282.6222
1962 283.6114 285.6749 287.7383 290.8280
1963 293.9176 297.3448 300.7719 304.7711
1964 308.7703 312.7183 316.6662 319.7714
1965 322.8766 324.9378 326.9990 329.2554
1966 331.5118 334.5053 337.4988 340.8877
1967 344.2765 347.5354 350.7944 354.3333
1968 357.8722 361.3964 364.9207 369.2385
1969 373.5563 377.8087 382.0611 386.0325
1970 390.0039 393.8748 397.7456 401.6539
1971 405.5622 408.7511 411.9400 415.8901
1972 419.8402 424.6194 429.3986 434.9853
1973 440.5719 447.5303 454.4886 461.0068
1974 467.5250 473.7610 479.9971 483.2715
1975 486.5460 486.9935 487.4411 486.4180
1976 485.3950 485.0249 484.6548 486.0524
1977 487.4499 491.1351 494.8203 498.0715
1978 501.3226 500.8849 500.4471 498.6234
1979 496.7997 493.9144 491.0290 488.8458
1980 486.6626 486.6500 486.6375 487.6200
1981 488.6026 488.9707 489.3388 489.1630
1982 488.9872 485.1773 481.3674 477.0831
1983 472.7988 469.1936 465.5885 463.6994
1984 461.8103 461.3902 460.9700 460.7466
1985 460.5232 460.8413 461.1593 461.7757
1986 462.3920 463.5195 464.6469 465.6456
1987 466.6443 468.2189 469.7936 470.8418
1988 471.8900 472.7697 473.6494 474.9383
1989 476.2271 476.6764 477.1257 476.3280
1990 475.5302 473.6273 471.7244 468.1151
1991 464.5059 461.1180 457.7301 455.7947
1992 453.8593 451.7883 449.7172 448.7373
1993 447.7574 446.8663 445.9753 445.5833
1994 445.1913 445.1262 445.0611 445.5029
1995 445.9448 445.0748 444.2049 442.3102
1996 440.4154 438.9847 437.5539 436.5573
1997 435.5606 435.5990 435.6374 436.0055
1998 436.3735 436.4287 436.4838 437.3641
1999 438.2445 438.8329 439.4214 439.6862
2000 439.9510 439.6271 439.3033 437.8020
2001 436.3007 435.6799 435.0591 434.8027
2002 434.5463 434.3784 434.2105 434.1068
2003 434.0030 433.4228 432.8425 432.4764
2004 432.1104 432.1717 432.2330 431.9675
2005 431.7020 431.1921 430.6822 430.1702
```

```
2006 429.6583 428.5046 427.3510 425.7993
2007 424.2476 424.3162 424.3848 424.7594
2008 425.1340 425.7083 426.2826
```

> seasonal(decompSTL)

```
          Qtr1       Qtr2       Qtr3       Qtr4
1956   2.362696 -41.228039 -27.100247  65.965593
1957   2.362696 -41.228039 -27.100247  65.965593
1958   2.362696 -41.228039 -27.100247  65.965593
1959   2.362696 -41.228039 -27.100247  65.965593
1960   2.362696 -41.228039 -27.100247  65.965593
1961   2.362696 -41.228039 -27.100247  65.965593
1962   2.362696 -41.228039 -27.100247  65.965593
1963   2.362696 -41.228039 -27.100247  65.965593
1964   2.362696 -41.228039 -27.100247  65.965593
1965   2.362696 -41.228039 -27.100247  65.965593
1966   2.362696 -41.228039 -27.100247  65.965593
1967   2.362696 -41.228039 -27.100247  65.965593
1968   2.362696 -41.228039 -27.100247  65.965593
1969   2.362696 -41.228039 -27.100247  65.965593
1970   2.362696 -41.228039 -27.100247  65.965593
1971   2.362696 -41.228039 -27.100247  65.965593
1972   2.362696 -41.228039 -27.100247  65.965593
1973   2.362696 -41.228039 -27.100247  65.965593
1974   2.362696 -41.228039 -27.100247  65.965593
1975   2.362696 -41.228039 -27.100247  65.965593
1976   2.362696 -41.228039 -27.100247  65.965593
1977   2.362696 -41.228039 -27.100247  65.965593
1978   2.362696 -41.228039 -27.100247  65.965593
1979   2.362696 -41.228039 -27.100247  65.965593
1980   2.362696 -41.228039 -27.100247  65.965593
1981   2.362696 -41.228039 -27.100247  65.965593
1982   2.362696 -41.228039 -27.100247  65.965593
1983   2.362696 -41.228039 -27.100247  65.965593
1984   2.362696 -41.228039 -27.100247  65.965593
1985   2.362696 -41.228039 -27.100247  65.965593
1986   2.362696 -41.228039 -27.100247  65.965593
1987   2.362696 -41.228039 -27.100247  65.965593
1988   2.362696 -41.228039 -27.100247  65.965593
1989   2.362696 -41.228039 -27.100247  65.965593
1990   2.362696 -41.228039 -27.100247  65.965593
1991   2.362696 -41.228039 -27.100247  65.965593
1992   2.362696 -41.228039 -27.100247  65.965593
1993   2.362696 -41.228039 -27.100247  65.965593
1994   2.362696 -41.228039 -27.100247  65.965593
1995   2.362696 -41.228039 -27.100247  65.965593
1996   2.362696 -41.228039 -27.100247  65.965593
1997   2.362696 -41.228039 -27.100247  65.965593
1998   2.362696 -41.228039 -27.100247  65.965593
1999   2.362696 -41.228039 -27.100247  65.965593
2000   2.362696 -41.228039 -27.100247  65.965593
2001   2.362696 -41.228039 -27.100247  65.965593
2002   2.362696 -41.228039 -27.100247  65.965593
2003   2.362696 -41.228039 -27.100247  65.965593
2004   2.362696 -41.228039 -27.100247  65.965593
2005   2.362696 -41.228039 -27.100247  65.965593
```

```
2006   2.362696 -41.228039 -27.100247  65.965593
2007   2.362696 -41.228039 -27.100247  65.965593
2008   2.362696 -41.228039 -27.100247

> remainder(decompSTL)
           Qtr1        Qtr2        Qtr3        Qtr4
1956  26.2402082  -2.0913781  -3.1414917 -16.1467098
1957   0.5168102   8.7601218   1.2849060  -8.5436616
1958   6.2965087  10.6875373   0.3600384 -17.5131106
1959  -6.7175216   1.4259894   8.8509730 -23.2965132
1960   9.2247386  -8.0301126   8.9965086 -33.9430767
1961  12.7860760  -6.5141002   2.4671963  -9.5878021
1962  -6.9740624   5.5531839   9.3619028 -10.7935658
1963  -2.2802964  -1.1167107   4.3283475  -7.7367030
1964   1.8669844   1.5097562  10.4340005 -15.7370059
1965   5.7607257   4.2902394   6.1012258  -9.2209802
1966   1.1255517  -5.2772534  -2.3985860  -4.8532455
1967   6.3608330   9.6926219   1.3058834 -15.2988555
1968  32.7651437  -1.1683665 -10.8204041   6.7959270
1969   7.0809960  -4.5806434   6.0391897  -5.9980695
1970  -5.3665907   4.3532723   3.3546080  -1.6195091
1971   2.0751118   2.4769437  -5.8397519   5.1443235
1972  -3.2028631  -5.3913509  -9.2983661   5.0491535
1973  15.0654111 -19.3022180  -0.3883746  38.0276088
1974  -4.8876698  12.4670136  -2.8968305   6.7628721
1975  11.0913127   6.2344933 -25.3408534   1.6163706
1976  22.2423327 -10.7968630  -4.5545861  -4.0179584
1977  -3.8125928   3.0929218 -10.7200911   1.9629200
1978  11.3146691   4.3431877 -42.3468211  23.4109999
1979   3.8375589  -9.6863252 -15.9287368   0.1886276
1980  23.9747300 -18.4219984  13.4627456 -27.5856264
1981  57.0347395  -7.7426612   6.7614107  19.8713904
1982   1.6501080 -10.9492546  25.7328554  32.9513301
1983  -0.1614572 -22.9655930  -3.4882563   5.3349823
1984 -11.1730410   9.8378543 -16.8697778  25.2877843
1985   1.1140844  -2.6132359 -11.0590836  26.2587396
1986  -5.7546992   5.7085799  -8.5466685   2.3888092
1987  11.9930248 -10.9908857  -2.6933237   1.1926427
1988  -0.2526529   8.4583545   0.4508344  57.0961438
1989 -11.5898089   3.5516109  -4.0254967  24.7064270
1990   7.1070887   8.6007199 -15.6241764  64.9192688
1991  -2.8685480   4.1100526   5.3701257  52.2396821
1992 -13.2220235  -0.5602485  -2.6170010  17.2970967
1993 -17.1200676  15.3617074  -8.8750451   0.4511087
1994   1.4460006 -22.8981482   5.0391756  19.5314958
1995 -22.3074460   4.1532028  -1.1046758  11.7242286
1996 -33.7781291   0.2433736 -12.4536511   4.4771424
1997  -5.9233261   3.6290120  -2.5371774  24.0289397
1998 -10.7362053   1.7993644  -6.3835934  13.6702586
1999  -5.6071514 -14.6048998  11.6788243  15.3482137
2000 -21.3136590   3.6009146   1.7969608  -3.7675865
2001  12.3366041 -14.4518699   8.0411286  -8.7682891
2002  -8.9089688  14.8496349  -1.1102889   5.9276304
2003  -1.3657124 -12.1947400  15.2577048  -8.4420402
2004   0.5269528  -0.9436311   6.8672575 -43.9331108
2005 -18.0647411  13.0358927   4.4179991 -14.1358425
```

```
2006   5.9790539  -1.2765809   4.7492569  -0.7648810
2007   0.3897190  -0.0881512  -3.2845489 -17.7250144
2008  -7.4967421   5.5197347  10.8176840
```

<u>(A3.2.) Results from time series cross-validation tcCV(); output is in the form of an **error matrix**.</u>

```
$`ETS`
                 h=1         h=2         h=3         h=4         h=5
Dec 1977   6.0000000   5.0000000  13.0000000   6.0000000  14.0000000
Jan 1978   4.9000000  12.9000000   5.9000000  13.9000000  14.9000000
Feb 1978   8.0000000   1.0000000   9.0000000  10.0000000   3.0000000
Mar 1978  -8.2448471  -3.4109201  -5.5769932 -15.7430663 -46.9091394
Apr 1978   6.5657334   7.5657334   0.5657334 -27.4342666   6.5657334
May 1978   2.0849084  -6.0491930 -35.1832944  -2.3173957   0.5485029
Jun 1978  -5.6045524 -33.6045524   0.3954476   4.3954476   7.3954476
Jul 1978 -28.3746761   5.6253239   9.6253239  12.6253239   4.6253239
Aug 1978   8.7788391  12.7788391  15.7788391   7.7788391  22.7788391
Sep 1978  11.8987284  14.8987284   6.8987284  21.8987284  16.8987284
Oct 1978  13.6070189   5.6070189  20.6070189  15.6070189  24.6070189
Nov 1978   4.6674844  19.6674844  14.6674844  23.6674844  19.6674844
Dec 1978  19.3095011  14.3095011  23.3095011  19.3095011  23.3095011
Jan 1979  12.9070418  21.9070418  17.9070418  21.9070418  22.9070418
Feb 1979  14.4134979  10.4134979  14.4134979  15.4134979   3.4134979
Mar 1979   4.2306081   8.2306081   9.2306081  -2.7693919 -33.7693919
Apr 1979   5.5327319   6.5327319  -5.4672681 -36.4672681  -7.4672681
May 1979   3.9638629  -8.0361371 -39.0361371 -10.0361371   0.9638629
Jun 1979  -9.8996700 -40.8996700 -11.8996700  -0.8996700 -12.8996700
Jul 1979 -36.6102122  -7.6102122   3.3897878  -8.6102122 -22.6102122
Aug 1979   4.4301049  15.4301049   3.4301049 -10.5698951  -3.5698951
Sep 1979  13.6092535   1.6092535 -12.3907465  -5.3907465  -5.3907465
Oct 1979  -2.5026119 -16.5026119  -9.5026119  -9.5026119 -11.5026119
Nov 1979 -15.8246505  -8.8246505  -8.8246505 -10.8246505  -3.8246505
Dec 1979  -4.4952569  -4.4952569  -6.4952569   0.5047431   2.5047431
Jan 1980  -3.0499116  -5.0499116   1.9500884   3.9500884   6.9500884
Feb 1980  -8.7306062   3.3377348  -2.5783988  -0.3528957   3.3006594
Mar 1980   9.9342522   6.0510528   8.6706857  11.2681271  -2.9666896
Apr 1980  -1.1218971   3.2797105   5.4119986  -6.8807558   1.1454288
May 1980   6.4674191  -0.5325809 -41.5325809  -1.5325809 -20.5325809
Jun 1980  -0.2584242 -10.1491140  -2.9835052 -28.0241269   0.7575542
Jul 1980 -12.2557503  -4.8127696 -30.4881956  -2.4690288  10.1978440
Aug 1980   7.7148208 -24.2534918   2.0339161  18.5465506   3.6781417
Sep 1980 -28.1375909  -0.5575751  12.7566859  -5.5297741  -3.3041074
Oct 1980  20.1671944  33.1809325  15.0441946  17.9620442  32.3286740
Nov 1980  22.8324489   1.3231248   4.0954191  17.0868069  16.4811212
Dec 1980 -12.2550843 -10.1219662   1.3659384   0.0307215  -3.1747499
Jan 1981  -2.9395878   9.2250794   8.4022289   4.3828170   3.0100898
Feb 1981  10.9088731  11.1345329   7.4584189   5.4595547  14.9288905
Mar 1981   5.2444985   1.9619294   0.2447313   9.4286561   3.5893196
Apr 1981  -1.4631001  -3.9156379   7.2010720   1.8460830   5.1704318
May 1981  -2.1831599   7.9383461   1.3220388   4.7622542  13.7156494
Jun 1981   8.7171868   1.9050594   6.1198906  15.3878591  18.6485701
Jul 1981  -1.6591881   2.3359724  11.7360932  14.2847581  18.4090909
Aug 1981   3.7043715  12.8417745  14.5752425  21.3229980  14.0535124
Sep 1981  10.8819588  14.5596592  19.1925077  12.4997261  21.4655419
Oct 1981   8.3867801  13.6250778   8.7476388  16.7201172  19.0899820
Nov 1981   7.8740127   2.6381257  11.1103241  12.6382388   4.3366805
```

Dec 1981	-2.5174499	5.8413374	9.1032459	-0.2083929	4.5715839
Jan 1982	7.1519065	9.8012476	0.6492343	5.7500271	3.1980450
Feb 1982	6.2578851	-1.8945716	2.0846391	-0.5848235	-1.7789964
Mar 1982	-6.8062544	-1.0776015	-2.9236599	-4.7130991	-12.4081442
Apr 1982	5.1002547	2.2151547	-0.6735201	-6.4242822	9.7793516
May 1982	-1.5581797	-2.1487864	-8.8457028	8.1473546	8.2184596
Jun 1982	-2.4054841	-8.8722173	9.7959198	8.8518109	4.3449171
Jul 1982	-5.7963570	11.9159155	12.8583158	9.8291055	15.4967622
Aug 1982	13.8516395	13.2435600	9.8443242	16.1680856	NA
Sep 1982	6.8214061	4.8590919	11.1147221	NA	NA
Oct 1982	-1.6920835	5.3934514	NA	NA	NA
Nov 1982	6.2757432	NA	NA	NA	NA
Dec 1982	NA	NA	NA	NA	NA

	h=6	**h=7**	**h=8**	**h=9**	**h=10**
Dec 1977	15.00000000	8.0000000	-20.0000000	14.0000000	18.0000000
Jan 1978	7.90000000	-20.1000000	13.9000000	17.9000000	20.9000000
Feb 1978	-25.00000000	9.0000000	13.0000000	16.0000000	8.0000000
Mar 1978	-16.07521247	-15.2412855	-15.4073586	-26.5734317	-14.7395048
Apr 1978	10.56573340	13.5657334	5.5657334	20.5657334	15.5657334
May 1978	2.41440151	-6.7196999	7.1461988	1.0120974	8.8779960
Jun 1978	-0.60455244	14.3954476	9.3954476	18.3954476	14.3954476
Jul 1978	19.62532387	14.6253239	23.6253239	19.6253239	23.6253239
Aug 1978	17.77883913	26.7788391	22.7788391	26.7788391	27.7788391
Sep 1978	25.89872839	21.8987284	25.8987284	26.8987284	14.8987284
Oct 1978	20.60701893	24.6070189	25.6070189	13.6070189	-17.3929811
Nov 1978	23.66748437	24.6674844	12.6674844	-18.3325156	10.6674844
Dec 1978	24.30950115	12.3095011	-18.6904989	10.3095011	21.3095011
Jan 1979	10.90704185	-20.0929582	8.9070418	19.9070418	7.9070418
Feb 1979	-27.58650209	1.4134979	12.4134979	0.4134979	-13.5865021
Mar 1979	-4.76939194	6.2306081	-5.7693919	-19.7693919	-12.7693919
Apr 1979	3.53273189	-8.4672681	-22.4672681	-15.4672681	-15.4672681
May 1979	-11.03613711	-25.0361371	-18.0361371	-18.0361371	-20.0361371
Jun 1979	-26.89966995	-19.8996700	-19.8996700	-21.8996700	-14.8996700
Jul 1979	-15.61021224	-15.6102122	-17.6102122	-10.6102122	-8.6102122
Aug 1979	-3.56989515	-5.5698951	1.4301049	3.4301049	6.4301049
Sep 1979	-7.39074647	-0.3907465	1.6092535	4.6092535	-2.3907465
Oct 1979	-4.50261194	-2.5026119	0.4973881	-6.5026119	-47.5026119
Nov 1979	-1.82465052	1.1753495	-5.8246505	-46.8246505	-6.8246505
Dec 1979	5.50474315	-1.4952569	-42.4952569	-2.4952569	-21.4952569
Jan 1980	-0.04991157	-41.0499116	-1.0499116	-20.0499116	2.9500884
Feb 1980	-8.89341890	-0.9151415	-27.3720737	-0.5846016	13.5307180
Mar 1980	6.80313374	-18.5465365	7.9551373	21.6757965	0.2470367
Apr 1980	-24.95947476	1.7686151	15.9180518	-4.0570741	-1.9531298
May 1980	2.46741910	5.4674191	-5.5325809	-5.5325809	12.4674191
Jun 1980	15.07139910	-3.3292039	-2.8786018	12.9368306	9.7932368
Jul 1980	-10.49843080	-7.3849596	6.8011846	2.9226802	2.4812340
Aug 1980	2.32841799	14.8712625	12.2364085	8.3053219	5.5005529
Sep 1980	12.32068840	9.8620115	7.4927640	1.8153528	13.4708228
Oct 1980	35.75725804	30.1286036	26.9038456	33.8615355	24.8053438
Nov 1980	11.21863302	8.1386917	16.9935003	10.2454537	14.5610120
Dec 1980	-4.94986721	5.0850539	0.2058637	3.9164663	13.8701082
Jan 1981	11.92739156	5.5847891	9.6507875	19.7520746	20.8147801
Feb 1981	8.09228978	12.5817015	21.2315103	22.6511260	26.3922404
Mar 1981	7.24320395	17.0629650	18.1403180	22.0817008	16.4332860
Apr 1981	15.30345015	15.3209681	21.6400613	14.3545475	22.4151114

May 1981	15.52738365	20.4598254	14.8536241	23.7195354	25.3599313
Jun 1981	23.02156817	15.2886351	25.2107226	28.3667869	18.5906039
Jul 1981	12.59265804	20.2678377	23.7707943	14.3633719	20.1988288
Aug 1981	22.91428731	25.2984788	15.4793246	20.8792321	17.7793555
Sep 1981	23.44495827	14.0590440	19.7037409	16.4948841	14.0131676
Oct 1981	9.85069909	13.5488927	11.1492284	9.1225035	3.9050399
Nov 1981	11.19453773	7.5556836	5.5380276	1.2115985	17.9382263
Dec 1981	1.86922953	-0.2282823	-4.7713498	12.4366386	12.7309871
Jan 1982	1.36866849	-4.5023596	13.5812205	14.8236035	11.2291145
Feb 1982	-6.74407047	9.0884386	9.5251486	7.4701827	13.8590628
Mar 1982	7.13792064	5.5221896	3.2029561	9.5084081	NA
Apr 1982	10.06321734	8.1516990	13.2799562	NA	NA
May 1982	5.40405195	11.4920727	NA	NA	NA
Jun 1982	10.04997729	NA	NA	NA	NA
Jul 1982	NA	NA	NA	NA	NA
Aug 1982	NA	NA	NA	NA	NA
Sep 1982	NA	NA	NA	NA	NA
Oct 1982	NA	NA	NA	NA	NA
Nov 1982	NA	NA	NA	NA	NA
Dec 1982	NA	NA	NA	NA	NA

	h=11	**h=12**
Dec 1977	21.0000000	13.0000000
Jan 1978	12.9000000	27.9000000
Feb 1978	23.0000000	18.0000000
Mar 1978	-22.9055779	-17.0716509
Apr 1978	24.5657334	20.5657334
May 1978	3.7438946	6.6097933
Jun 1978	18.3954476	19.3954476
Jul 1978	24.6253239	12.6253239
Aug 1978	15.7788391	-15.2211609
Sep 1978	-16.1012716	12.8987284
Oct 1978	11.6070189	22.6070189
Nov 1978	21.6674844	9.6674844
Dec 1978	9.3095011	-4.6904989
Jan 1979	-6.0929582	0.9070418
Feb 1979	-6.5865021	-6.5865021
Mar 1979	-12.7693919	-14.7693919
Apr 1979	-17.4672681	-10.4672681
May 1979	-13.0361371	-11.0361371
Jun 1979	-12.8996700	-9.8996700
Jul 1979	-5.6102122	-12.6102122
Aug 1979	-0.5698951	-41.5698951
Sep 1979	-43.3907465	-3.3907465
Oct 1979	-7.5026119	-26.5026119
Nov 1979	-25.8246505	-2.8246505
Dec 1979	1.5047431	4.5047431
Jan 1980	5.9500884	-5.0499116
Feb 1980	-5.7730061	-1.9907422
Mar 1980	2.7211492	16.7070130
Apr 1980	9.9813303	9.7780092
May 1980	9.4674191	10.4674191
Jun 1980	6.0218193	3.3119453
Jul 1980	0.8679241	11.0001498
Aug 1980	18.7804113	9.6864436
Sep 1980	3.6076334	12.8172176

```
Oct 1980  26.2801445  34.3625080
Nov 1980  23.4266949  25.0633508
Dec 1980  15.5300134  19.3049185
Jan 1981  24.7017028  19.3097622
Feb 1981  20.5469931  28.5337015
Mar 1981  24.7594347  27.4261664
Apr 1981  26.2413440  16.2478718
May 1981  17.3784783  22.5065692
Jun 1981  23.6360419  19.6132833
Jul 1981  15.9284532  14.7330165
Aug 1981  14.7497985   9.3138089
Sep 1981   8.8694930  24.7629391
Oct 1981  20.7132135  20.4304070
Nov 1981  16.7481115  13.9469244
Dec 1981  10.0098264  16.1417118
Jan 1982  16.2123202          NA
Feb 1982          NA          NA
Mar 1982          NA          NA
Apr 1982          NA          NA
May 1982          NA          NA
Jun 1982          NA          NA
Jul 1982          NA          NA
Aug 1982          NA          NA
Sep 1982          NA          NA
Oct 1982          NA          NA
Nov 1982          NA          NA
Dec 1982          NA          NA
```

```
$ARIMA
                  h=1           h=2        h=3           h=4        h=5
Dec 1977   6.00000000  5.000000e+00  13.000000  6.000000e+00  14.0000000
Jan 1978   2.00000000  1.000000e+01   3.000000  1.100000e+01  12.0000000
Feb 1978   9.33333333  2.333333e+00  10.333333  1.133333e+01   4.3333333
Mar 1978   0.00000000  8.000000e+00   9.000000  2.000000e+00 -26.0000000
Apr 1978   8.00000000  9.000000e+00   2.000000 -2.600000e+01   8.0000000
May 1978   7.66666667  6.666667e-01 -27.333333  6.666667e+00  10.6666667
Jun 1978  -7.00000000 -3.500000e+01  -1.000000  3.000000e+00   6.0000000
Jul 1978 -28.37500000  5.625000e+00   9.625000  1.262500e+01   4.6250000
Aug 1978   8.77777778  1.277778e+01  15.777778  7.777778e+00  22.7777778
Sep 1978  11.90000000  1.490000e+01   6.900000  2.190000e+01  16.9000000
Oct 1978  13.81818182  5.818182e+00  20.818182  1.581818e+01  24.8181818
Nov 1978   4.66666667  1.966667e+01  14.666667  2.366667e+01  19.6666667
Dec 1978  19.30769231  1.430769e+01  23.307692  1.930769e+01  23.3076923
Jan 1979  12.92857143  2.192857e+01  17.928571  2.192857e+01  22.9285714
Feb 1979  21.06666667  1.706667e+01  21.066667  2.206667e+01  10.0666667
Mar 1979  -4.00000000  0.000000e+00   1.000000 -1.100000e+01 -42.0000000
Apr 1979   4.00000000  5.000000e+00  -7.000000 -3.800000e+01  -9.0000000
May 1979   1.00000000 -1.100000e+01 -42.000000 -1.300000e+01  -2.0000000
Jun 1979 -12.00000000 -4.300000e+01 -14.000000 -3.000000e+00 -15.0000000
Jul 1979 -31.00000000 -2.000000e+00   9.000000 -3.000000e+00 -17.0000000
Aug 1979  17.20689049  2.253470e+01   7.806513 -7.505676e+00  -1.1368068
Sep 1979  13.99990014  3.106030e+00 -10.486115 -3.335730e+00  -3.2802798
Oct 1979 -12.00000000 -2.600000e+01 -19.000000 -1.900000e+01 -21.0000000
Nov 1979 -14.00000000 -7.000000e+00  -7.000000 -9.000000e+00  -2.0000000
Dec 1979  -8.00000000 -3.000000e+00 -14.000000 -3.000000e+00  -5.0000000
```

Jan 1980	5.00000000	-6.000000e+00	5.000000	3.000000e+00	5.0000000
Feb 1980	-11.00000000	2.842171e-14	-2.000000	2.842171e-14	5.0000000
Mar 1980	11.00000000	9.000000e+00	11.000000	1.600000e+01	6.0000000
Apr 1980	-2.00000000	2.842171e-14	5.000000	-5.000000e+00	6.0000000
May 1980	2.00000000	7.000000e+00	-3.000000	8.000000e+00	-22.0000000
Jun 1980	5.00000000	-5.000000e+00	6.000000	-2.400000e+01	11.0000000
Jul 1980	-10.00000000	1.000000e+00	-29.000000	6.000000e+00	23.0000000
Aug 1980	7.32026980	-2.132569e+01	13.176061	3.035940e+01	12.2919381
Sep 1980	-25.10139590	7.717120e+00	25.688595	7.255971e+00	7.4486304
Oct 1980	14.97403570	4.534201e+01	18.418458	2.437522e+01	40.3988884
Nov 1980	46.52836237	3.616243e+00	24.633788	2.690197e+01	31.8617285
Dec 1980	-11.76602233	-1.405205e+01	6.786247	-3.521161e+00	-4.4084331
Jan 1981	-3.22844600	9.943919e+00	8.050999	4.471890e+00	1.7713740
Feb 1981	14.18187654	1.168225e+01	8.345837	5.548179e+00	15.1971696
Mar 1981	6.17641855	3.811163e+00	-2.755215	1.075801e+01	5.6356488
Apr 1981	1.18616924	-4.812891e+00	6.741043	2.602347e+00	6.6003818
May 1981	-5.26150877	6.412641e+00	1.834678	6.116972e+00	14.4118254
Jun 1981	8.77816535	3.677615e+00	9.504170	1.670241e+01	22.1678870
Jul 1981	0.05276353	6.313685e+00	11.624802	1.766278e+01	21.8151358
Aug 1981	6.38988787	1.133263e+01	17.770363	2.211472e+01	19.8904842
Sep 1981	9.17501093	1.528139e+01	17.752402	1.668373e+01	24.0300828
Oct 1981	14.49199704	2.057218e+01	18.063624	2.557426e+01	33.5732967
Nov 1981	12.01334100	1.175891e+01	20.008769	2.708373e+01	13.5480079
Dec 1981	5.59746357	1.421021e+01	20.167144	7.723588e+00	16.3270783
Jan 1982	10.95229832	1.714710e+01	4.713691	1.340977e+01	10.2466662
Feb 1982	11.47922627	-8.411403e-01	9.379710	7.022493e+00	3.6858419
Mar 1982	-9.42218289	-2.683367e+00	-7.787605	-1.274921e+01	-13.2328078
Apr 1982	2.07064382	-3.936944e+00	-6.861411	-8.051567e+00	4.3177680
May 1982	-4.95837325	-7.678805e+00	-9.331237	3.210640e+00	10.6364431
Jun 1982	-5.38525032	-7.309748e+00	6.265849	1.309578e+01	-1.4382212
Jul 1982	-4.66376981	8.533829e+00	16.680310	1.565565e+00	-0.3387155
Aug 1982	10.89238919	1.881517e+01	4.539154	2.252556e+00	NA
Sep 1982	13.27732172	3.689480e-02	-4.564479	NA	NA
Oct 1982	-6.58046515	-9.348499e+00	NA	NA	NA
Nov 1982	-5.85460783	NA	NA	NA	NA
Dec 1982	NA	NA	NA	NA	NA

	h=6	**h=7**	**h=8**	**h=9**	**h=10**
Dec 1977	15.0000000	8.0000000	-20.0000000	14.000000	18.0000000
Jan 1978	5.0000000	-23.0000000	11.0000000	15.000000	18.0000000
Feb 1978	-23.6666667	10.3333333	14.3333333	17.333333	9.3333333
Mar 1978	8.0000000	12.0000000	15.0000000	7.000000	22.0000000
Apr 1978	12.0000000	15.0000000	7.0000000	22.000000	17.0000000
May 1978	13.6666667	5.6666667	20.6666667	15.666667	24.6666667
Jun 1978	-2.0000000	13.0000000	8.0000000	17.000000	13.0000000
Jul 1978	19.6250000	14.6250000	23.6250000	19.625000	23.6250000
Aug 1978	17.7777778	26.7777778	22.7777778	26.777778	27.7777778
Sep 1978	25.9000000	21.9000000	25.9000000	26.900000	14.9000000
Oct 1978	20.8181818	24.8181818	25.8181818	13.818182	-17.1818182
Nov 1978	23.6666667	24.6666667	12.6666667	-18.333333	10.6666667
Dec 1978	24.3076923	12.3076923	-18.6923077	10.307692	21.3076923
Jan 1979	10.9285714	-20.0714286	8.9285714	19.928571	7.9285714
Feb 1979	-20.9333333	8.0666667	19.0666667	7.066667	-6.9333333
Mar 1979	-13.0000000	-2.0000000	-14.0000000	-28.000000	-21.0000000
Apr 1979	2.0000000	-10.0000000	-24.0000000	-17.000000	-17.0000000
May 1979	-14.0000000	-28.0000000	-21.0000000	-21.000000	-23.0000000

Jun 1979	-29.0000000	-22.0000000	-22.0000000	-24.000000	-17.0000000
Jul 1979	-10.0000000	-10.0000000	-12.0000000	-5.000000	-3.0000000
Aug 1979	-1.4403649	-3.5863688	3.3434069	5.309631	8.2933853
Sep 1979	-5.2598340	1.7477048	3.7504845	6.751509	-0.2481126
Oct 1979	-14.0000000	-12.0000000	-9.0000000	-16.000000	-57.0000000
Nov 1979	0.0000000	3.0000000	-4.0000000	-45.000000	-5.0000000
Dec 1979	-3.0000000	2.0000000	-8.0000000	3.000000	-27.0000000
Jan 1980	10.0000000	0.0000000	11.0000000	-19.000000	16.0000000
Feb 1980	-5.0000000	6.0000000	-24.0000000	11.000000	28.0000000
Mar 1980	17.0000000	-13.0000000	22.0000000	39.000000	21.0000000
Apr 1980	-24.0000000	11.0000000	28.0000000	10.000000	10.0000000
May 1980	13.0000000	30.0000000	12.0000000	12.000000	32.0000000
Jun 1980	28.0000000	10.0000000	10.0000000	30.000000	20.0000000
Jul 1980	5.0000000	5.0000000	25.0000000	15.000000	14.0000000
Aug 1980	12.3167635	32.3076284	22.3109899	21.309753	17.3102081
Sep 1980	27.3628337	17.4010413	16.3840264	12.391604	20.3882293
Oct 1980	33.0532149	30.2813666	27.4641323	34.674598	36.2016376
Nov 1980	18.2406651	24.8886461	23.9052906	32.484256	24.0901292
Dec 1980	-8.4497709	-0.4346121	0.5598291	-1.438132	25.5611200
Jan 1981	11.4207635	4.9633339	8.5113417	18.257275	19.2009920
Feb 1981	8.9903367	12.3865937	22.6609414	23.070624	25.0722259
Mar 1981	9.4636250	18.1572922	23.3107335	26.707813	25.5138171
Apr 1981	15.1907130	19.9927828	23.2690178	22.189519	29.2666364
May 1981	19.5175809	23.0185107	21.7536879	28.807557	34.2956391
Jun 1981	26.2817815	24.6114324	31.6001073	37.483829	23.0427621
Jul 1981	19.8042869	27.0285124	32.7158692	18.270276	25.6845159
Aug 1981	27.1229792	33.0293794	18.4694278	25.875110	21.3810515
Sep 1981	29.2137992	15.0095944	22.4749343	17.935894	12.9049147
Oct 1981	18.6634032	26.2524596	21.6104691	17.666763	17.5920005
Nov 1981	21.7341858	17.8680322	13.6994072	13.906746	27.0965814
Dec 1981	13.0879697	8.6460339	9.0827167	22.708280	30.9379591
Jan 1982	5.9823855	6.5438700	20.2719251	28.837503	14.6703794
Feb 1982	4.9215905	19.2004304	28.4756687	14.537162	13.3858002
Mar 1982	-0.5165416	6.3695544	-7.9297164	-9.799183	NA
Apr 1982	11.5793593	-2.7818945	-4.7395352	NA	NA
May 1982	-3.8832435	-5.8384552	NA	NA	NA
Jun 1982	-3.1486939	NA	NA	NA	NA
Jul 1982	NA	NA	NA	NA	NA
Aug 1982	NA	NA	NA	NA	NA
Sep 1982	NA	NA	NA	NA	NA
Oct 1982	NA	NA	NA	NA	NA
Nov 1982	NA	NA	NA	NA	NA
Dec 1982	NA	NA	NA	NA	NA

	h=11	**h=12**
Dec 1977	21.00000000	13.00000000
Jan 1978	10.00000000	25.00000000
Feb 1978	24.33333333	19.33333333
Mar 1978	17.00000000	26.00000000
Apr 1978	26.00000000	22.00000000
May 1978	20.66666667	24.66666667
Jun 1978	17.00000000	18.00000000
Jul 1978	24.62500000	12.62500000
Aug 1978	15.77777778	-15.22222222
Sep 1978	-16.10000000	12.90000000
Oct 1978	11.81818182	22.81818182

```
Nov 1978  21.66666667   9.66666667
Dec 1978   9.30769231  -4.69230769
Jan 1979  -6.07142857   0.92857143
Feb 1979   0.06666667   0.06666667
Mar 1979 -21.00000000 -23.00000000
Apr 1979 -19.00000000 -12.00000000
May 1979 -16.00000000 -14.00000000
Jun 1979 -15.00000000 -12.00000000
Jul 1979   0.00000000  -7.00000000
Aug 1979   1.28557165 -39.71818654
Sep 1979 -41.24797324  -1.24792186
Oct 1979 -17.00000000 -36.00000000
Nov 1979 -24.00000000  -1.00000000
Dec 1979   8.00000000  25.00000000
Jan 1980  33.00000000  15.00000000
Feb 1980  10.00000000  10.00000000
Mar 1980  21.00000000  41.00000000
Apr 1980  30.00000000  20.00000000
May 1980  22.00000000  21.00000000
Jun 1980  19.00000000  15.00000000
Jul 1980  10.00000000  18.00000000
Aug 1980  25.31004064  26.31010227
Sep 1980  21.38973194  19.38906276
Oct 1980  33.84982185  61.08467016
Nov 1980  56.48464584  34.93346749
Dec 1980   8.56139412   1.56129360
Jan 1981  21.59520097  16.88737989
Feb 1981  21.00693489  29.34477338
Mar 1981  32.55704633  38.06919491
Apr 1981  34.52389798  20.36729808
May 1981  20.03257701  27.42092913
Jun 1981  30.39496462  25.84325868
Jul 1981  21.23089932  16.19409686
Aug 1981  16.42727376  16.07028996
Sep 1981  12.54764731  25.29331365
Oct 1981  30.30382129  42.10780457
Nov 1981  37.04927746  21.38746204
Dec 1981  16.78899137  15.32965465
Jan 1982  13.27168962           NA
Feb 1982           NA           NA
Mar 1982           NA           NA
Apr 1982           NA           NA
May 1982           NA           NA
Jun 1982           NA           NA
Jul 1982           NA           NA
Aug 1982           NA           NA
Sep 1982           NA           NA
Oct 1982           NA           NA
Nov 1982           NA           NA
Dec 1982           NA           NA
```

$NEURAL

	h=1	h=2	h=3	h=4	h=5
Dec 1977	NA	NA	NA	NA	NA
Jan 1978	NA	NA	NA	NA	NA

Feb 1978	7.84501830	0.8694410	8.8656226	9.8662204	2.8661268
Mar 1978	-2.99964679	5.0003536	6.0003536	-0.9996464	-28.9996464
Apr 1978	7.77709937	9.0713924	1.6201934	-25.8262204	7.3167371
May 1978	9.81524890	-1.5000719	-29.3726226	4.6076022	8.6104625
Jun 1978	-2.60186868	-30.6018687	3.3981313	7.3981313	10.3981313
Jul 1978	-30.16720489	3.8327840	7.8327840	10.8327840	2.8327840
Aug 1978	10.45540385	14.4553851	17.4553851	9.4553851	24.4553851
Sep 1978	12.30764061	15.1604746	7.1679233	22.1675548	17.1675731
Oct 1978	13.77939110	5.7786995	20.7786997	15.7786997	24.7786997
Nov 1978	-7.27068315	7.7803966	2.7852291	11.7856967	7.7857420
Dec 1978	19.97282174	17.8258425	26.8258521	22.8258521	26.8258521
Jan 1979	11.33024879	-9.7153644	18.6220470	-13.4543211	23.2321322
Feb 1979	-11.84024097	0.8579390	-12.9193402	-11.2708379	-9.8867016
Mar 1979	4.36854154	-0.3362685	3.0592670	-5.3239727	3.9698169
Apr 1979	-1.64967528	-2.4345240	-11.7616802	-19.5304094	-17.0173373
May 1979	0.96639622	-10.7389269	-18.3764359	-19.1381495	-0.8016578
Jun 1979	-11.21768788	-15.9696475	-18.3852843	-2.3665026	-13.0759393
Jul 1979	-4.46351819	-31.5905603	1.3304837	-25.7976351	-22.1810051
Aug 1979	-25.11362905	-2.0826306	-20.5428733	-23.5133103	-34.5552622
Sep 1979	-4.94462375	-15.3744021	-26.3035385	-23.8312582	-22.6365232
Oct 1979	-14.05046442	-26.7722403	-15.1944883	-18.6575191	-13.8573927
Nov 1979	NA	NA	NA	NA	NA
Dec 1979	-21.68861800	-13.5974737	-26.5680834	-13.4617833	-16.7418542
Jan 1980	-4.11789773	-18.3798189	-9.9443907	-8.0185683	-5.0175475
Feb 1980	-9.30064683	-25.7388022	-0.8887091	-26.4561156	29.5604679
Mar 1980	0.49607647	-3.4304947	-16.6043395	-11.1400932	-45.4880215
Apr 1980	-6.17845059	-5.3614662	-12.1259297	-40.6161648	1.1131458
May 1980	0.30036586	-10.0183870	-30.1516699	2.5446109	-16.1194980
Jun 1980	-14.56919727	-31.3381770	-4.4660731	-32.9285467	-6.1958665
Jul 1980	-23.78500280	1.5304401	-16.4188432	14.6413713	20.5353545
Aug 1980	4.95610758	-20.6274186	3.0519832	-0.8805782	-17.2978993
Sep 1980	-29.93751811	-4.8889105	4.3898930	-10.3085136	-3.3014005
Oct 1980	-4.10806959	24.9458990	-10.8932951	-12.8137582	11.3726951
Nov 1980	48.62878232	8.3225630	44.7191497	35.7567125	46.0926407
Dec 1980	-5.30925079	-9.7583366	15.3536814	7.4426768	15.4083601
Jan 1981	-5.69190504	15.1845374	9.4691480	12.2080554	9.5853788
Feb 1981	8.46877715	8.0511387	7.4887688	6.7905305	9.3226563
Mar 1981	-3.75210617	-5.3317926	-7.7995922	-6.0317761	-13.5522186
Apr 1981	-0.08652913	-1.2462151	-0.8239406	-7.9907359	11.2891859
May 1981	-15.96147135	-2.6341800	-22.6735580	9.7358806	15.5602678
Jun 1981	3.56444606	-16.0683934	8.4096535	23.6756477	22.6090344
Jul 1981	-6.16467411	5.7085744	16.0336017	22.4586015	18.1997550
Aug 1981	17.93529947	22.3273764	14.7961108	15.1170657	14.6575113
Sep 1981	20.55000079	27.2690253	16.9475709	12.6357208	18.3100615
Oct 1981	33.52065487	15.3452182	10.5308538	20.2802307	31.1459468
Nov 1981	3.87840596	10.5224547	16.0505853	18.2448532	10.3753109
Dec 1981	4.48059785	11.7540973	23.4815737	12.4825721	21.4901509
Jan 1982	12.27286447	25.8715344	15.1041589	24.1355695	21.1679958
Feb 1982	18.91784911	6.6056922	16.2514428	14.3745015	4.3387268
Mar 1982	-14.68954569	-10.8602209	-18.0874374	-31.7657643	-61.9512718
Apr 1982	9.89346964	7.3430360	-1.3946383	-36.0067475	16.9399034
May 1982	5.10378251	-6.5474428	-23.0320783	26.0496022	23.9628166
Jun 1982	-10.63515510	-30.9452888	15.2810432	9.3515569	-10.9392283
Jul 1982	-20.72478534	30.9477884	30.1166741	30.5531036	28.6222786
Aug 1982	34.18182081	28.5182979	28.1437476	26.6240998	NA
Sep 1982	24.15704969	13.9517828	13.1238250	NA	NA

Oct 1982	18.34722072	7.1735280	NA	NA	NA
Nov 1982	0.93312694	NA	NA	NA	NA
Dec 1982	NA	NA	NA	NA	NA

	h=6	**h=7**	**h=8**	**h=9**	**h=10**
Dec 1977	NA	NA	NA	NA	NA
Jan 1978	NA	NA	NA	NA	NA
Feb 1978	-25.133859	8.8661392	12.86613952	15.86613946	7.866139
Mar 1978	5.000354	9.0003536	12.00035363	4.00035363	19.000354
Apr 1978	12.271656	13.9224576	7.32906184	20.64389342	17.353240
May 1978	11.610044	3.6101053	18.61009643	13.61009773	22.610098
Jun 1978	2.398131	17.3981313	12.39813132	21.39813132	17.398131
Jul 1978	17.832784	12.8327840	21.83278405	17.83278405	21.832784
Aug 1978	19.455385	28.4553851	24.45538514	28.45538514	29.455385
Sep 1978	26.167572	22.1675722	26.16757224	27.16757224	15.167572
Oct 1978	20.778700	24.7786997	25.77869966	13.77869966	-17.221300
Nov 1978	11.785746	12.7857469	0.78574690	-30.21425310	-1.214253
Dec 1978	27.825852	15.8258521	-15.17414787	13.82585213	24.825852
Jan 1979	-17.855223	-18.5203107	-27.26565020	19.99417146	-30.608785
Feb 1979	-17.588376	-41.8246099	-15.57965983	-33.40262940	-35.737176
Mar 1979	-20.969066	0.8841829	-15.01442603	-24.91353299	-24.186371
Apr 1979	-5.883331	-17.2849883	-31.58314878	-23.27946304	-23.512107
May 1979	-10.237119	-28.5507585	-15.20510854	-16.53520147	-16.666686
Jun 1979	-29.241575	-19.8228492	-20.02896220	-23.13088299	-15.069690
Jul 1979	-37.824307	-21.3749330	-41.08008430	-20.05380269	-33.293707
Aug 1979	-25.697934	-34.8853360	-24.31161295	-24.75132274	-21.497130
Sep 1979	-25.896202	-18.2187741	-16.80645752	-13.84813099	-19.378969
Oct 1979	-7.934166	-3.9543022	-0.36253929	-15.87967094	-19.961402
Nov 1979	NA	NA	NA	NA	NA
Dec 1979	-13.880181	-13.0793313	-28.83110734	-8.27480194	-40.053663
Jan 1980	-11.900476	-31.8139766	-3.99640110	-31.42840106	-8.961641
Feb 1980	-42.365027	-1.9259659	-28.58110783	-17.73796944	25.880538
Mar 1980	-10.960133	-30.8258683	-9.32395011	-5.29599214	-18.061184
Apr 1980	-17.943725	2.1243859	5.14240880	-3.05040404	-6.141962
May 1980	15.949286	29.7359897	16.81716010	21.31782494	38.493579
Jun 1980	23.238771	-2.8744814	-9.48862595	26.25173588	17.261470
Jul 1980	4.535027	12.9802824	28.85072334	16.74075623	16.527988
Aug 1980	-18.798939	-0.8275987	-2.49004562	0.05645242	-1.034075
Sep 1980	7.995393	10.6074949	8.26316063	7.74138808	3.847796
Oct 1980	7.048066	9.8397198	8.14422533	7.41414464	-10.301608
Nov 1980	32.850895	46.9420973	32.43933276	8.00752165	30.679077
Dec 1980	8.907315	15.5466132	4.54157471	11.57378417	42.729396
Jan 1981	13.915085	6.9935227	12.02640133	37.48345317	22.756270
Feb 1981	4.754988	11.3625938	43.68094630	23.78405607	26.158804
Mar 1981	11.781185	22.6907449	22.28589099	21.09242713	16.789931
Apr 1981	20.997615	21.1722269	17.84057496	14.77312502	26.052738
May 1981	21.673775	18.3703043	15.00174057	25.60255849	31.405055
Jun 1981	17.828834	23.5337543	24.17972498	34.76477922	26.169246
Jul 1981	21.892403	24.9863166	36.02755156	24.38665167	33.447776
Aug 1981	18.953322	32.1505000	21.78434683	29.26038822	26.712120
Sep 1981	33.839516	22.7320522	31.93634734	28.86746346	19.938909
Oct 1981	20.205039	29.3427991	26.47348627	17.45241077	6.493809
Nov 1981	5.762915	6.2360793	0.05456658	-13.13989844	29.561439
Dec 1981	18.482853	9.4901599	-2.32710725	32.24819205	34.425762
Jan 1982	12.162514	-1.2268397	32.73316418	35.86934806	32.991930
Feb 1982	-16.087162	34.0344001	32.41962468	31.83104502	26.878560

Mar 1982	-13.417831	-25.4363844	-42.30559302	-49.85703354	NA
Apr 1982	17.664854	19.1824442	17.91336195	NA	NA
May 1982	21.855815	17.7837469	NA	NA	NA
Jun 1982	6.979541	NA	NA	NA	NA
Jul 1982	NA	NA	NA	NA	NA
Aug 1982	NA	NA	NA	NA	NA
Sep 1982	NA	NA	NA	NA	NA
Oct 1982	NA	NA	NA	NA	NA
Nov 1982	NA	NA	NA	NA	NA
Dec 1982	NA	NA	NA	NA	NA

	h=11	**h=12**
Dec 1977	NA	NA
Jan 1978	NA	NA
Feb 1978	22.866139	17.8661395
Mar 1978	14.000354	23.0003536
Apr 1978	24.518793	22.3619292
May 1978	18.610098	22.6100976
Jun 1978	21.398131	22.3981313
Jul 1978	22.832784	10.8327840
Aug 1978	17.455385	-13.5446149
Sep 1978	-15.832428	13.1675722
Oct 1978	11.778700	22.7786997
Nov 1978	9.785747	-2.2142531
Dec 1978	12.825852	-1.1741479
Jan 1979	-5.274851	-39.6385293
Feb 1979	-47.494838	-39.8983337
Mar 1979	-20.548211	-26.2033037
Apr 1979	-25.056896	-18.0586796
May 1979	-10.518754	-7.7158934
Jun 1979	-14.114715	-10.9257215
Jul 1979	-23.929165	-18.2377428
Aug 1979	-21.628636	-23.8180070
Sep 1979	-22.202345	-16.8530781
Oct 1979	-13.835068	-29.4659824
Nov 1979	NA	NA
Dec 1979	-15.087850	1.9409332
Jan 1980	4.644590	-7.1636625
Feb 1980	-6.365027	-5.9411328
Mar 1980	-17.893728	0.2500108
Apr 1980	14.502051	9.2753808
May 1980	22.842517	17.1305149
Jun 1980	13.762767	11.0334185
Jul 1980	10.606771	16.1870218
Aug 1980	-2.341567	2.4092125
Sep 1980	1.306030	9.6640057
Oct 1980	4.102335	44.5152732
Nov 1980	42.337017	44.2141653
Dec 1980	22.855381	28.2785235
Jan 1981	24.263273	16.0817770
Feb 1981	15.566491	22.7948320
Mar 1981	26.818302	35.4704557
Apr 1981	35.456568	23.7926550
May 1981	18.695495	25.5263112
Jun 1981	33.461862	29.8029277
Jul 1981	30.342183	21.5186668

Aug 1981	18.416857	9.8986082
Sep 1981	11.808052	32.5866047
Oct 1981	32.024595	30.9817665
Nov 1981	28.358500	28.1414253
Dec 1981	36.162743	31.6312916
Jan 1982	30.593021	NA
Feb 1982	NA	NA
Mar 1982	NA	NA
Apr 1982	NA	NA
May 1982	NA	NA
Jun 1982	NA	NA
Jul 1982	NA	NA
Aug 1982	NA	NA
Sep 1982	NA	NA
Oct 1982	NA	NA
Nov 1982	NA	NA
Dec 1982	NA	NA

$NAIVE

	h=1	**h=2**	**h=3**	**h=4**	**h=5**	**h=6**	**h=7**	**h=8**	**h=9**	**h=10**	**h=11**	**h=12**
Dec 1977	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jan 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Feb 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Mar 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Apr 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
May 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jun 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jul 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Aug 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Sep 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Oct 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Nov 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Dec 1978	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jan 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Feb 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Mar 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Apr 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
May 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jun 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jul 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Aug 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Sep 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Oct 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Nov 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Dec 1979	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jan 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Feb 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Mar 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Apr 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
May 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jun 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jul 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Aug 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Sep 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Oct 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Nov 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Dec 1980	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jan 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Feb 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Mar 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Apr 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
May 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jun 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jul 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Aug 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Sep 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Oct 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Nov 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Dec 1981	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jan 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Feb 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Mar 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Apr 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
May 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jun 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Jul 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Aug 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Sep 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Oct 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Nov 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Dec 1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Affidavit

I hereby swear that I single-handedly compiled this document. I did not make use of any other sources than those listed here. Furthermore, I swear that this document was never handed in to another department in this or any deviated manner. Every passage referring to or quoted from another source is identifiable as such.

Hamburg, 16.01.2019